

DOCTORAL DISSERTATION

STATISTICAL MODELS
BASED ON MULTIPLE MODEL STRUCTURES
FOR SPEECH RECOGNITION

DOCTOR OF ENGINEERING

JANUARY 2012

Sayaka SHIOTA

Supervisor : Dr. Keiichi TOKUDA

Department of Scientific and Engineering Simulation
Nagoya Institute of Technology

Abstract

The topics of automatic speech recognition (ASR) has been active areas of research focus. Hidden Markov models (HMMs) are one of widely used statistical models for representing time series by well-defined algorithms. They have successfully been applied to acoustic modeling in speech recognition when training data can be sufficiently available. Recently, although I can easily obtain large and many databases from the Internet, the databases contain noises or miss transcriptions and the quality are low. Thus, acoustic modeling technique without considering a quality of given data is important for improving speech recognition performance. In this paper, frameworks of improving acoustic modeling were proposed for HMM-based speech recognition.

First, I propose a simultaneously optimization of model structure and model parameters. In the use of context-dependent models, decision-tree-based context clustering is applied to find an appropriate parameter tying structure. However, context clustering is usually performed on the basis of unreliable statistics of hidden Markov model (HMM) state sequences because the estimation of reliable state sequences requires an appropriate model structures, that cannot be obtained prior to context clustering. Therefore, context clustering and the estimation of state sequences essentially cannot be performed independently. To overcome this problem, I propose an optimization technique of state sequences based on an annealing process using multiple decision trees. In this technique, a new likelihood function is defined in order to treat multiple model structures, and the deterministic annealing expectation maximization (DAEM) algorithm is used as the training algorithm. Speech recognition experiments show that the proposed method achieved a higher performance than the conventional methods.

Next, training criterion has been focused. The maximum likelihood (ML) criterion has usually been used for training statistical models for HMM-based speech recognition systems. However, since the ML criterion produces a point estimate of model parameters, the estimation accuracy may degrade when little training data is available. The Bayesian method is a statistical technique for estimating reliable predictive distributions by marginalizing model parameters, and it can accurately estimate observation distribu-

tions even if the amount of training data is small. However, the local maxima problem in the Bayesian method is more serious than in the ML-based approach, because the Bayesian method treats not only state sequences but also model parameters as latent variables. The deterministic annealing EM (DAEM) algorithm has been proposed to improve the local maxima problem in the EM algorithm, and its effectiveness has been reported in HMM-based speech recognition using ML criterion. In this paper, the DAEM algorithm is applied to Bayesian speech recognition to relax the local maxima problem.

In speech recognition based on generative models, there are many efforts to find appropriate model structures to predict observation vector sequences (e.g., multi-mixture models, clustering techniques and more complicated models). Even though a better prediction obtained by these methods leads to improve recognition performance, they still aim to find only one model structure. However, in most practical cases, it is insufficient to represent a true model distribution using only “one” model structure, because a family of such models usually does not include a true distribution. Therefore, it is necessary to increase model complexity efficiently without inaccurate estimation caused by the over-fitting problem. Thus, I focus on model structure integration based on the Bayesian framework. In the previous work, I proposed the marginalization of model parameters based on the Bayesian framework. Next, the model structures should be marginalized. Therefore, I proposed a new likelihood function for using multiple model structures. Since the proposed framework regards model structures as latent variables, the local maxima problem is caused. The basic idea of the Bayesian approach is to treat all parameters as random variables. Therefore, I proposed a novel framework of using multiple model structures based on the Bayesian framework. The conventional VB method sometimes suffers from the local maxima problem, because the conventional VB method treats not only state sequences but also model parameters as latent variables, that makes the estimation problem complicated. To overcome this problem, I have proposed the training algorithm applying the deterministic annealing framework to the Bayesian speech recognition, and reported the effectiveness for the local maxima problem. Since the proposed technique also treats the multiple model structures as a latent variable, the local maxima problem is more serious than in the conventional VB method. Therefore, the DAEM algorithm is applied to the proposed technique as a training algorithm. The proposed method can consistently perform model estimation and model selection based on the VB method.

Keywords: Speech recognition, Hidden Markov Model, Deterministic Annealing, Local Maxima problem, Multiple Model Structures, Training Algorithm, Bayesian approach

Abstract in Japanese

近年、音声を情報伝達の手段としたシステムの需要が高まっており、音声認識・音声合成などといった音声に関する研究が盛んに行われている。音声認識における代表的な枠組みとして、音響モデルに統計モデルの一種である隠れマルコフモデル (Hidden Markov Model; HMM) を用いる枠組みがある。HMM は声の強さ、速さ、明瞭さといった音声パターンの変動を確率モデルで捉えることから統計的に処理できることや、比較的簡単なモデルパラメータの推定法が知られていること、現実的な計算量で学習・認識を行えることといった特徴があるため、モデルの推定に十分な学習データ量が与えられれば高い認識性能を示すことが知られている。しかし、高精度な音響モデルを学習するためには学習データとして雑音や言い間違いなどが存在しない音声データと音声データと対になる発話内容のテキストデータが必要となる。また、より精度の高い音響モデルを推定するためには発話内容に関する正確な時間情報が必要となる。インターネットの発達に伴いウェブ上から大量の音声データを入手することは容易になってきているが、高精度な音響モデルを構築するための最適なデータを揃えることは困難であり、限られた量の学習データや情報が不正確なデータから汎化性能の高い音響モデルを推定することは重要な課題であると言える。そこで、本論文は音響モデルの性能を向上する枠組みを提案することで HMM 音声認識システムの認識性能を改善することを目的とする。

まず、音響モデルのモデル構造とモデルパラメータ推定の同時最適化について提案を行う。従来の HMM 音声認識システムでは、音声の最小単位となる音素をコンテキスト依存モデルが広く用いられている。コンテキスト依存モデルは前後の音素などの音素文脈 (コンテキスト) を考慮した詳細なモデル表現ができるという利点がある一方で、モデル数が非常に膨大になるために各モデルに十分なデータ量を割り当てることが困難となるという問題を抱えている。この問題に対処するために、決定木に基づくコンテキストクラスタリングという手法が提案され、音声認識の分野において広く用いられている。コンテキストクラスタリングでは決定木構造と呼ばれるモデル構造を構築し、各モデルに割り当てられる学習データ量を増やすことでより信頼性の高いモデルパラメータの推定を行う。しかし、高精度な決定木構造を構築するためには初期のモデルパラメータとして用いられるモデルパラメータの信頼性が高い必要がある。逆に、信頼性の高いモデルパラメータを推定するためには

高精度な決定木構造が必要となる．このようにモデルパラメータの推定と決定木構造の構築には相互に強い依存関係があるため，同時に最適化されることが望ましい．しかし，決定木構造の構築とモデルパラメータの推定の間に強い依存関係を持っているために同時最適化は計算量的に困難である．そこで，提案法では，複数のモデル構造を用いてモデルパラメータの推定を行うことでモデル構造とモデルパラメータ推定の同時最適化の近似を表現する枠組みを提案し，連続音声認識実験においても複数のモデル構造を考慮することの有効性を示した．

次に，音響モデルの学習基準について考察する．HMM に基づく音響モデリングでは，尤度最大化 (Maximum likelihood; ML) 基準が広く用いられている．しかし，ML 基準は学習データが十分に得られない場合，モデルの推定精度が低下するという問題がある．これに対し，ベイズ基準では学習データが少ない場合においても高い汎化性能が得られることが知られている．さらに，近年変分ベイズ法が提案され音声認識においてもその有効性が確認されている．しかし，ベイズ基準を用いた学習では，隠れ変数が増加することから ML 基準よりも初期値に依存する局所最適性問題の影響を受けると考えられる．そのため，学習アルゴリズムを改善するために確定的アニーリング EM (Deterministic Annealing EM; DAEM) アルゴリズムを学習アルゴリズムとして導入することで重要な課題である局所最適性問題に対処することが出来ることを示した．

さらに，ベイズ基準による音響モデリングにおいてモデル構造に関する提案を行う．従来の生成モデルによる音声認識システムでは，適切なモデル構造を観測系列から推定するために混合正規分布モデル (Gaussian Mixture Model; GMM) やクラスタリング手法の改善など様々な提案が行われてきた．これらの手法により，モデル構造をより複雑に表現することができが，音声信号の真の分布を表現するための表現としては不十分であった．そこでベイズ基準において複数のモデル構造を用いることを提案する．ベイズ基準において複数のモデル構造を扱うということはつまり，モデル構造に関しても周辺化を行うということを意味している．ベイズ基準の基本概念は全てのパラメータを周辺化することであるため，モデルパラメータだけでなくモデル構造についても周辺化することは順当な考え方である．連続音声認識実験において，ベイズ基準という統一的な枠組みにおいてモデルパラメータの推定を行い，複数のモデル構造を用いることの有効性を示した．

以上の様に，本論文では，統計的手法による音声認識のためのより高精度なモデル化手法を提案し，これらの手法の有効性を示す．

Acknowledgement

First of all, I would like to express my sincere gratitude to Keiichi Tokuda, my advisor, for his support, encouragement, and guidance.

I would like to thank Akinobu Lee, Yoshihiko Nankaku, Heiga Zen (currently with Research scientist at Google), Keiichiro Oura, and Kei Hashimoto for their technical supports and helpful discussions. Special thanks go to all the members of Tokuda and Lee laboratories for their technical support and encouragement. If somebody was missed among them, my work would not be completed. I would be remiss if I did not thank Natsuki Kuromiya and Masayo Fujimura, secretaries of the laboratory, for their kind assistance.

I am grateful to Satoshi Nakamura (currently with Nara Institute of Science and Technology), Hisashi Kawai (with NICT), Shinsuke Sakai (currently with Kyoto University), Yoshinori Shiga (with NICT), and and Tomoki Toda (with Nara Institute of Science and Technology) for giving me the opportunity to work in National Institute of Information and Communications Technology Spoken Language Communication Group and for their valuable advice.

Finally, I would sincerely like to thank my parents and my friends for their encouragement.

Contents

List of Tables	ix
List of Figures	x
1 Introduction	1
2 Hidden Markov Models	3
2.1 Definition of HMM	3
2.2 Calculation of output probability	5
2.2.1 Total output probability of an observation vector sequence	5
2.2.2 Forward-Backward algorithm	6
2.3 Searching optimal state sequence	8
2.4 Maximum likelihood estimation of HMM parameters	9
2.4.1 Q -function	9
2.4.2 Maximization of Q -function	10
2.5 Summary	12
3 HMM-based speech recognition	13
3.1 Statistical speech recognition	13
3.2 Front-ends	14
3.3 HMM-based acoustic modeling	15

3.4	Word N -gram-based language modeling	17
3.5	Pronunciation lexicon	18
3.6	Search algorithms	18
3.7	Summary	18
4	Speech recognition based on statistical models including multiple phonetic decision trees	19
4.1	Deterministic annealing EM algorithm in parameter estimation	19
4.1.1	EM algorithm	19
4.1.2	Deterministic annealing EM algorithm	20
4.1.3	Optimization of state sequences	21
4.2	Speech recognition based on multiple phonetic decision trees	22
4.2.1	Acoustic modeling based on model structure annealing	22
4.2.2	Speech decoding based on multiple model structures	24
4.3	Experiments	25
4.3.1	Speaker dependent phoneme recognition	25
4.3.2	Speaker independent phoneme recognition	27
4.4	Summary	28
5	Speech recognition based on variational Bayesian method	35
5.1	Speech recognition based on variational Bayesian method	35
5.1.1	Bayesian approach	35
5.1.2	Variational Bayesian method	36
5.1.3	Bayesian context clustering using cross validation	37
5.2	DAEM algorithm for variational Bayes method	37
5.3	Experiments	38
5.3.1	Experimental conditions	38

5.3.2	Experimental results	40
5.4	Summary	42
6	Integration of multiple model structures based on Bayesian framework	43
6.1	Bayesian speech recognition using multiple model structures	43
6.1.1	Marginalized likelihood function including multiple model structures	43
6.1.2	Training algorithm based on deterministic annealing	44
6.2	Related approach	45
6.2.1	Random Forest	45
6.2.2	Non-parametric Bayes	46
6.2.3	Discriminative approaches	46
6.3	Experiments	46
6.3.1	Speaker independent speech recognition (small training data) . . .	46
6.3.2	Speaker independent speech recognition (large training data) . . .	51
6.4	Summary	55
	List of Publications	62
	Journal papers	62
	International conference proceedings	62
	Technical reports	63
	Domestic conference proceedings	64
	Appendix A Derivation of Bayesian framework using multiple model structures	65
A.1	Parameter estimation based Bayesian framework using multiple model structures (diagonal matrix)	65
	Appendix B Software	75

List of Tables

5.1	Experimental conditions	39
6.1	Experimental condition	47

List of Figures

2.1	Examples of HMM structure.	4
2.2	Implementation of the computation using forward-backward algorithm in terms of a trellis of observations and states.	7
3.1	Example of a phonetic decision tree for triphone models.	16
4.1	Joint optimization process	30
4.2	Schedule of temperature parameter β_q	31
4.3	Schedule of update $Q(m)$	31
4.4	Log-likelihood of training data (Speaker dependent)	32
4.5	Phoneme accuracy for each temperature schedule	32
4.6	Log-likelihood of training data (Speaker independent)	33
4.7	Word accuracy for each temperature schedule	33
4.8	Word accuracy using multiple trees in decoding	34
5.1	Log marginal likelihood	41
5.2	Phoneme accuracy	42
6.1	Upper bound of log marginal likelihood (the temperature parameter of DAEM and Mtree are set to $\alpha = 2$)	49
6.2	Phoneme accuracy	50
6.3	The posterior distributions of the model structures. Monophone has 129 leaf nodes and CV-Bayes has 7,755 leaf nodes.	51

6.4	Upper bound of log marginal likelihood $\bar{\mathcal{F}}_{\beta}$	54
6.5	Phoneme accuracy	55
6.6	Posterior distributions of model structures.	56
B.7	HTS: http://hts.sp.nitech.ac.jp/	75

Chapter 1

Introduction

Speech is the most important ways for human communication, and a number of research topic for human-machine communication have been proposed. Automatic speech recognition (ASR) and text-to-speech synthesis (TTS) are fundamental technologies for human-machine communication. In recent years, they are used in many application such as car navigation system, information retrieval over the telephone, voice mail, speech-to-speech translation (S2ST) system, and so on. The goal of ASR and TTS systems is perfect speech recognition and speech synthesis with natural human voice characteristics.

Most state-of-art speech recognition is based on large amounts of speech data. This type of approach is generally called corpus-based systems. In these days statistical approaches based on hidden Markov models (HMMs) have been dominant in ASR [1], due to their ease of implementation and modeling flexibility. In this approach, the HMMs are used for modeling sequences of speech spectra. In this paper, improved techniques for acoustic modeling are proposed for HMM-based speech recognition.

First, I focused on improvement of training algorithm of speech recognition. For conventional speech recognition system based on statistical models, the maximum likelihood (ML) criterion has usually been used. However, since the ML criterion produces a point estimate of model parameters, the estimation accuracy may degrade when little training data is available. The Bayesian approach is a statistical technique for estimating reliable predictive distributions by marginalizing model parameters, and it can accurately estimate observation distributions even if the amount of training data is small. However, the calculation becomes complicated due to the combination of latent variables, i.e., state sequences and model parameters. To solve this problem, the variational Bayesian (VB) method has been proposed as an effective approximation method of the Bayesian approach [2] [3], and it shows a good performance in HMM-based speech recognition [4] [5] [6] [7]. Although the Bayesian approach achieves higher performance than

the ML approach, the local maxima problem in the Bayesian method is more serious than in the ML-based approach, because the Bayesian method treats not only state sequences but also model parameters as latent variables. The combination of many latent variables makes the likelihood function complicated. Therefore, the optimization algorithm is important for the Bayesian approach. Furthermore, the VB method assumes the independence between the posterior distributions of state sequences and model parameters, and these factorized distributions are iteratively updated. This means that the VB method requires reliable initial posterior distributions. To overcome this problem, some approaches have been reported [8] [9], and I have also been reported the training algorithm applying the deterministic annealing EM (DAEM) algorithm [10] to the the effectiveness for the local maxima problem for speech recognition system [11].

Next, a framework of using multiple model structure is proposed. In conventional speech recognition based on generative models, there are many efforts to find appropriate model structures to predict observation vector sequences (e.g., multi-mixture models, clustering techniques and more complicated models). Even though a better prediction obtained by these methods leads to improve recognition performance, they still aim to find only one model structure. However, in most practical cases, it is insufficient to represent a true model distribution using only “one” model structure, because a family of such models usually does not include a true distribution. Therefore, it is necessary to increase model complexity efficiently without inaccurate estimation caused by the over-fitting problem. Recently, to overcome this problem, some approaches were reported using multiple model structures (e.g., random forest [12] and ROVER [13]). Although various integration techniques and criteria can be considered, I focuses on model structure integration based on the Bayesian framework in acoustic modeling. The proposed method can consistently perform model estimation and model selection based on the VB method. The conventional VB method sometimes suffers from the local maxima problem, because the conventional VB method treats not only state sequences but also model parameters as latent variables, that makes the estimation problem complicated. To overcome this problem, I have proposed the training algorithm applying the deterministic annealing EM (DAEM) algorithm [10] to the Bayesian speech recognition, and reported the effectiveness for the local maxima problem [11]. Since the proposed technique also treats the multiple model structures as a latent variable, the local maxima problem is more serious than in the conventional VB method. Therefore, the DAEM algorithm is applied to the proposed technique as a training algorithm.

Chapter 2

Hidden Markov Models

Recently, hidden Markov models (HMMs) are widely used as statistical models for speech recognition. The advantages of using the HMM are that i) it can represent speech as probability distributions, ii) it is robust, iii) efficient algorithms for estimating its model parameters are provided. Parameter estimation and calculation of output probability distributions are described in this chapter.

2.1 Definition of HMM

An HMM [14–16] is a finite state machine which generates a sequence of discrete time observations. At each frame it changes states according to its state transition probability distributions, and then generates an observation at time t , \mathbf{O}_t , according to its output probability distribution of the current state. Therefore, the HMM is a doubly stochastic random process model.

An N -state HMM consist of state transition probability distributions $\{a_{ij}\}_{i,j=1}^N$, output probability distributions $\{b_j(\mathbf{O}_t)\}_{j=1}^N$, and initial state probability distributions $\{\pi_i\}_{i=1}^N$. For convenience, the compact notation is used to indicate the parameter set of the model Λ as follows:

$$\Lambda = \left[\{a_{ij}\}_{i,j=1}^N, \{b_j(\cdot)\}_{j=1}^N, \{\pi_i\}_{i=1}^N \right] \quad (2.1)$$

Figure 2.1 shows examples of the HMM structure. Figure 2.1(a) shows a 3-state ergodic model, in which every state of the model could be reached from every state of the model in a single step, and Figure 2.1(b) shows a 3-state left-to-right model, in which the state

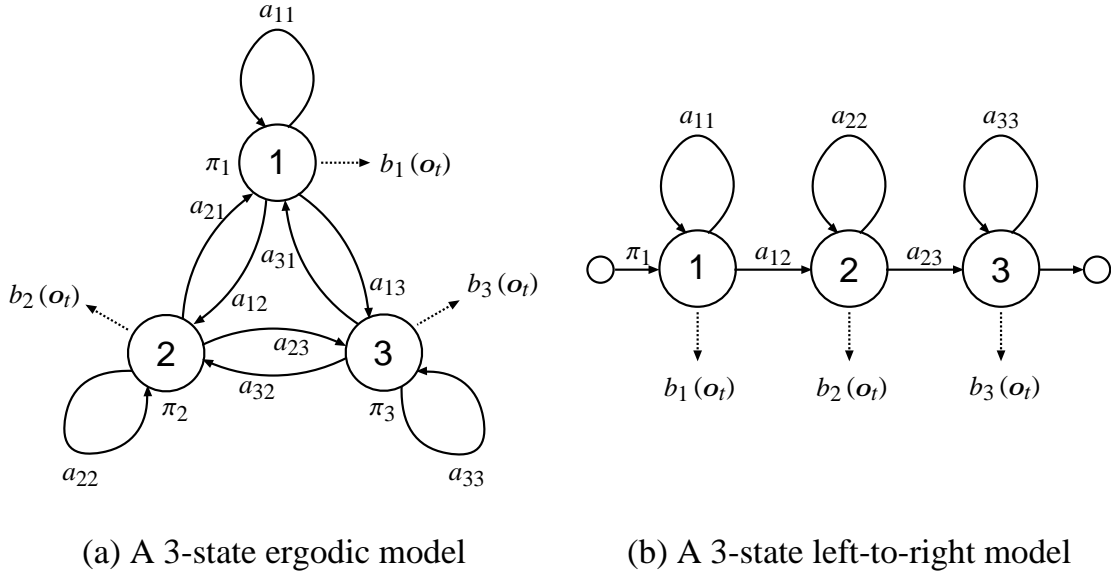


Figure 2.1: Examples of HMM structure.

index increases or stays the same state as time increases. The left-to-right HMMs are generally used to model speech parameter sequences, since they can appropriately model signals.

The output probability distributions $\{b_j(\cdot)\}_{j=1}^N$ can be discrete or continuous depending on the observations. In continuous distribution HMM (CD-HMM), each output probability distribution is usually modeled by a mixture of multivariate Gaussian components [17] as follows:

$$b_j(\mathbf{O}_t) = \sum_{m=1}^M w_{jm} \cdot \mathcal{N}(\mathbf{O}_t \mid \boldsymbol{\mu}_{jm}, \boldsymbol{\sigma}_{jm}), \quad (2.2)$$

where M , w_{jm} , $\boldsymbol{\mu}_{jm}$, and $\boldsymbol{\sigma}_{jm}$ are the number of Gaussian components, the mixture weight, mean vector, and covariance matrix of the m -th Gaussian component of the j -th state, respectively. Each Gaussian component is defined by

$$\mathcal{N}(\mathbf{O}_t \mid \boldsymbol{\mu}_{jm}, \boldsymbol{\sigma}_{jm}) = \frac{1}{\sqrt{(2\pi)^K |\boldsymbol{\sigma}_{jm}|}} \exp \left\{ -\frac{1}{2} (\mathbf{O}_t - \boldsymbol{\mu}_{jm})^\top \boldsymbol{\sigma}_{jm}^{-1} (\mathbf{O}_t - \boldsymbol{\mu}_{jm}) \right\}, \quad (2.3)$$

where symbol \top means transpose of vector or matrix, and K is the dimensionality of an observation vector \mathbf{O}_t . For each state, $\{w_{jm}\}_{m=1}^M$ should satisfy the stochastic constraint

$$\sum_{m=1}^M w_{jm} = 1, \quad 1 \leq j \leq N \quad (2.4)$$

$$w_{jm} \geq 0, \quad \begin{matrix} 1 \leq j \leq N \\ 1 \leq m \leq M \end{matrix} \quad (2.5)$$

so that $\{b_j(\cdot)\}_{j=1}^N$ are properly normalized, i.e.,

$$\int_{\mathbb{R}^K} b_j(\mathbf{O}_t) d\mathbf{O}_t = 1. \quad 1 \leq j \leq N \quad (2.6)$$

2.2 Calculation of output probability

2.2.1 Total output probability of an observation vector sequence

When a state sequence is determined, a joint probability of an observation vector sequence $\mathbf{O} = \{\mathbf{O}_1, \mathbf{O}_2, \dots, \mathbf{O}_T\}$ and a state sequence $\mathbf{q} = \{q_1, q_2, \dots, q_T\}$ is calculated by multiplying the state transition probabilities and state output probabilities for each state, that is,

$$p(\mathbf{O}, \mathbf{q} \mid \Lambda) = \prod_{t=1}^T a_{q_{t-1}q_t} b_{q_t}(\mathbf{O}_t), \quad (2.7)$$

where $a_{q_0q_1}$ denotes π_{q_1} . The total output probability of the observation vector sequence from the HMM is calculated by marginalizing Eq. (2.7) over all possible state sequences,

$$p(\mathbf{O} \mid \Lambda) = \sum_{\text{all } \mathbf{q}} \prod_{t=1}^T a_{q_{t-1}q_t} b_{q_t}(\mathbf{O}_t). \quad (2.8)$$

The order of $2T \cdot N^T$ calculation is required, since at every $t = 1, 2, \dots, T$ there are N possible states that can be reached (i.e., there are N^T possible state sequences). This calculation is computationally infeasible, even for small values of N and T ; e.g., for $N = 5$ (states), $T = 100$ (observations), there are on the order of $2 \cdot 100 \cdot 5^{100} \approx 10^{72}$ computations. Fortunately, there is an efficient algorithm to calculate Eq. (2.8) using forward and backward procedures.

2.2.2 Forward-Backward algorithm

The forward-backward algorithm is generally used to calculate $p(\mathbf{O} \mid \Lambda)$, which is the probability of the observation sequence \mathbf{O} given the model Λ . If I directly calculate $p(\mathbf{O} \mid \Lambda)$, it requires on the order of $2T \cdot N^T$ calculation. The detail of the forward-backward algorithm is described in the following part.

The probability of a partial observation vector sequence from time 1 to t and the i -th state at time t , given the HMM Λ is defined as

$$\alpha_t(i) = p(\mathbf{O}_1, \mathbf{O}_2, \dots, \mathbf{O}_t, q_t = i \mid \Lambda). \quad (2.9)$$

$\alpha_t(i)$ is calculated recursively as follows:

1. Initialization

$$\alpha_1(i) = \pi_i b_i(\mathbf{O}_1), \quad 1 \leq i \leq N \quad (2.10)$$

2. Recursion

$$\alpha_t(j) = \left[\sum_{i=1}^N \alpha_{t-1}(i) a_{ij} \right] b_j(\mathbf{O}_t), \quad \begin{matrix} 1 \leq j \leq N \\ t = 2, \dots, T \end{matrix} \quad (2.11)$$

3. Termination

$$p(\mathbf{O} \mid \Lambda) = \sum_{i=1}^N \alpha_T(i). \quad (2.12)$$

As the same way as the forward algorithm, backward variables $\beta_t(i)$ are defined as

$$\beta_t(i) = p(\mathbf{O}_{t+1}, \mathbf{O}_{t+2}, \dots, \mathbf{O}_T \mid s_t = i, \Lambda), \quad (2.13)$$

that is, the probability of a partial vector observation sequence from time t to T , given the i -th state at time t and the HMM Λ . The backward variables can also be calculated in a recursive manner as follows:

1. Initialization

$$\beta_T(i) = 1, \quad 1 \leq i \leq N \quad (2.14)$$

2. Recursion

$$\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(\mathbf{O}_{t+1}) \beta_{t+1}(j), \quad \begin{matrix} 1 \leq i \leq N \\ t = T-1, \dots, 1. \end{matrix} \quad (2.15)$$

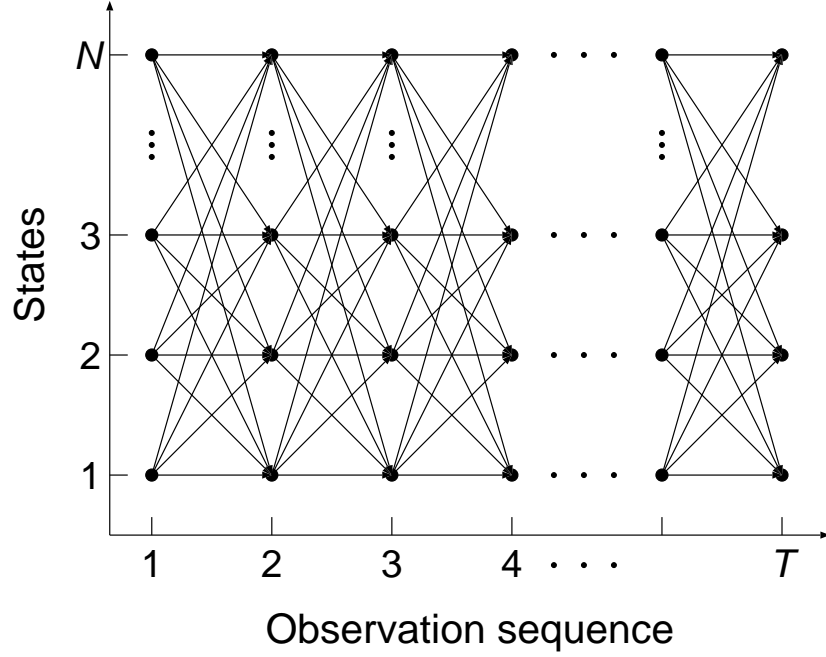


Figure 2.2: Implementation of the computation using forward-backward algorithm in terms of a trellis of observations and states.

3. Termination

$$p(\mathbf{O} \mid \Lambda) = \sum_{i=1}^N \beta_1(i). \quad (2.16)$$

The forward and backward variables can be used to compute the total output probability as follows:

$$p(\mathbf{O} \mid \Lambda) = \sum_{j=1}^N \alpha_t(j) \beta_t(j), \quad 1 \leq t \leq T \quad (2.17)$$

The forward-backward algorithm is based on the trellis structure shown in Figure 2.2. In this figure, the x-axis and y-axis represent observations and states of an HMM, respectively. On the trellis, all possible state sequences will re-merge into these N nodes no matter how long the observation sequence. In the case of the forward algorithm, at time $t = 1$, I need to calculate values of $\alpha_1(i)$, $1 \leq i \leq N$. At times $t = 2, 3, \dots, T$, I need only calculate values of $\alpha_t(j)$, $1 \leq j \leq N$, where each calculation involves only the N previous values of $\alpha_{t-1}(i)$ because each of the N grid points can be reached from only the N grid points at the previous time slot. As a result, the forward-backward algorithm can reduce order of probability calculation.

2.3 Searching optimal state sequence

The single optimal state sequence $\hat{\mathbf{q}} = \{\hat{q}_1, \hat{q}_2, \dots, \hat{q}_T\}$ for a given observation vector sequence $\mathbf{O} = \{\mathbf{O}_1, \mathbf{O}_2, \dots, \mathbf{O}_T\}$ is useful for various applications (e.g., decoding, initializing HMM parameters). By using a manner similar to the forward algorithm, which is often referred to as the Viterbi algorithm [18], I can obtain the optimal state sequence $\hat{\mathbf{q}}$. Let $\delta_t(i)$ be the likelihood of the most likely state sequence ending in the i -th state at time t

$$\delta_t(i) = \max_{q_1, \dots, q_{t-1}} p(q_1, \dots, q_{t-1}, q_t = i, \mathbf{O}_1, \dots, \mathbf{O}_t \mid \Lambda), \quad (2.18)$$

and $\psi_t(i)$ be the array to keep track. The complete procedure for finding the optimal state sequence can be written as follows:

1. Initialization

$$\delta_1(i) = \pi_i b_i(\mathbf{O}_1), \quad 1 \leq i \leq N \quad (2.19)$$

$$\psi_1(i) = 0, \quad 1 \leq i \leq N \quad (2.20)$$

2. Recursion

$$\delta_t(j) = \max_i [\delta_{t-1}(i) a_{ij}] b_j(\mathbf{O}_t), \quad \begin{matrix} 1 \leq i \leq N \\ t = 2, 3, \dots, T \end{matrix} \quad (2.21)$$

$$\psi_t(j) = \arg \max_i [\delta_{t-1}(i) a_{ij}], \quad \begin{matrix} 1 \leq i \leq N \\ t = 2, 3, \dots, T \end{matrix} \quad (2.22)$$

3. Termination

$$\hat{P} = \max_i [\delta_T(i)], \quad (2.23)$$

$$\hat{q}_T = \arg \max_i [\delta_T(i)]. \quad (2.24)$$

4. Back tracking

$$\hat{q}_t = \psi_{t+1}(\hat{q}_{t+1}), \quad t = T-1, \dots, 1. \quad (2.25)$$

It should be noted that the Viterbi algorithm is similar to the forward calculation of Eqs. (2.10)–(2.12). The major difference is the maximization in Eq. (2.21) over previous states, which is used in place of the summation in Eq. (2.11). It also should be clear that a trellis structure efficiently implements the computation of the Viterbi procedure.

2.4 Maximum likelihood estimation of HMM parameters

There is no known method to analytically obtain the model parameter set based on the maximum likelihood (ML) criterion to obtain Λ which maximizes its likelihood $p(\mathbf{O} \mid \Lambda)$ for a given observation sequence \mathbf{O} , in a closed form. Since this problem is a high dimensional nonlinear optimization problem, and there will be a number of local maxima, it is difficult to obtain Λ which globally maximizes $p(\mathbf{O} \mid \Lambda)$. However, the model parameter set Λ locally maximizes $p(\mathbf{O} \mid \Lambda)$ can be obtained using an iterative procedure such as the expectation-maximization (EM) algorithm [19], and the obtained parameter set will be appropriately estimated if a good initial estimate is provided.

In the following, the EM algorithm for the CD-HMM is described. The algorithm for the HMM with discrete output distributions can also be derived in a straightforward manner.

2.4.1 Q -function

In the EM algorithm, an auxiliary function $Q(\Lambda, \hat{\Lambda})$ of the current parameter set Λ and the new parameter set $\hat{\Lambda}$ is defined as follows:

$$Q(\Lambda, \hat{\Lambda}) = \sum_{\text{all } \mathbf{q}} p(\mathbf{q} \mid \mathbf{O}, \Lambda) \log p(\mathbf{O}, \mathbf{q} \mid \hat{\Lambda}). \quad (2.26)$$

Each mixture of Gaussian components is decomposed into a substate, and \mathbf{q} is redefined as a substate sequence,

$$\mathbf{q} = \{(q_1, s_1), (q_2, s_2), \dots, (q_T, s_T)\}, \quad (2.27)$$

where (q_t, s_t) represents being in the s_t -th substate (Gaussian component) of the q_t -th state at time t .

At each iteration of the procedure, the current parameter set Λ is replaced by the new parameter set $\hat{\Lambda}$ which maximizes $Q(\Lambda, \hat{\Lambda})$. This iterative procedure can be proved to

increase likelihood $p(\mathbf{O} \mid \Lambda)$ monotonically and converge to a certain critical point, since it can be proved that the \mathcal{Q} -function satisfies the following theorems:

- Theorem 1

$$\mathcal{Q}(\Lambda, \hat{\Lambda}) \geq \mathcal{Q}(\Lambda, \Lambda) \Rightarrow p(\mathbf{O} \mid \hat{\Lambda}) \geq p(\mathbf{O} \mid \Lambda) \quad (2.28)$$

- Theorem 2

The auxiliary function $\mathcal{Q}(\Lambda, \hat{\Lambda})$ has the unique global maximum as a function of Λ , and this maximum is the one and only critical point.

- Theorem 3

A parameter set Λ is a critical point of the likelihood $p(\mathbf{O} \mid \Lambda)$ if and only if it is a critical point of the \mathcal{Q} -function.

2.4.2 Maximization of \mathcal{Q} -function

According to Eqs. (2.2) and (2.7), $\log p(\mathbf{O}, \mathbf{q} \mid \Lambda)$ can be written as

$$\log p(\mathbf{O}, \mathbf{q} \mid \Lambda) = \log p(\mathbf{O} \mid \mathbf{q}, \Lambda) + \log P(\mathbf{q} \mid \Lambda), \quad (2.29)$$

$$\log p(\mathbf{O} \mid \mathbf{q}, \Lambda) = \sum_{t=1}^T \log \mathcal{N}(\mathbf{O}_t \mid \boldsymbol{\mu}_{q_t s_t}, \boldsymbol{\sigma}_{q_t s_t}), \quad (2.30)$$

$$\log P(\mathbf{q} \mid \Lambda) = \log \pi_{q_1} + \sum_{t=2}^T \log a_{q_{t-1} q_t} + \sum_{t=1}^T \log w_{q_t s_t}. \quad (2.31)$$

Hence, \mathcal{Q} -function (Eq. (2.26)) can be rewritten as

$$\begin{aligned} \mathcal{Q}(\Lambda, \hat{\Lambda}) = & \sum_{i=1}^N p(\mathbf{O}, q_1 = i \mid \Lambda) \cdot \log \pi_i \\ & + \sum_{i=1}^N \sum_{j=1}^N \sum_{t=1}^{T-1} p(\mathbf{O}, q_t = i, q_{t+1} = j) \cdot \log a_{ij} \\ & + \sum_{i=1}^N \sum_{m=1}^M \sum_{t=1}^T p(\mathbf{O}, q_t = i, s_t = m \mid \Lambda) \cdot \log w_{im} \\ & + \sum_{i=1}^N \sum_{m=1}^M \sum_{t=1}^T p(\mathbf{O}, q_t = i, s_t = m \mid \Lambda) \cdot \log \mathcal{N}(\mathbf{O}_t \mid \boldsymbol{\mu}_{im}, \boldsymbol{\sigma}_{im}). \end{aligned} \quad (2.32)$$

The parameter set Λ which maximizes the above equation subject to the stochastic constraints

$$\sum_{i=1}^N \pi_i = 1, \quad (2.33)$$

$$\sum_{j=1}^N a_{ij} = 1, \quad 1 \leq i \leq N \quad (2.34)$$

$$\sum_{m=1}^M w_{im} = 1, \quad 1 \leq i \leq N \quad (2.35)$$

can be derived by Lagrange multipliers or differential calculus as follows [20]:

$$\pi_i = \gamma_1(i), \quad 1 \leq i \leq N \quad (2.36)$$

$$a_{ij} = \frac{\sum_{t=2}^T \xi_{t-1}(i, j)}{\sum_{t=2}^T \gamma_{t-1}(i)}, \quad \begin{array}{l} 1 \leq i \leq N \\ 1 \leq j \leq N \end{array} \quad (2.37)$$

$$w_{im} = \frac{\sum_{t=1}^T \gamma_t(i, m)}{\sum_{t=1}^T \gamma_t(i)}, \quad \begin{array}{l} 1 \leq i \leq N \\ 1 \leq m \leq M \end{array} \quad (2.38)$$

$$\boldsymbol{\mu}_{im} = \frac{\sum_{t=1}^T \gamma_t(i, m) \cdot \mathbf{O}_t}{\sum_{t=1}^T \gamma_t(i, m)}, \quad \begin{array}{l} 1 \leq i \leq N \\ 1 \leq m \leq M \end{array} \quad (2.39)$$

$$\boldsymbol{\sigma}_{im} = \frac{\sum_{t=1}^T \gamma_t(i, m) \cdot (\mathbf{O}_t - \boldsymbol{\mu}_{im}) (\mathbf{O}_t - \boldsymbol{\mu}_{im})^\top}{\sum_{t=1}^T \gamma_t(i, m)}, \quad \begin{array}{l} 1 \leq i \leq N \\ 1 \leq m \leq M \end{array} \quad (2.40)$$

where $\gamma_t(i)$, $\gamma_t(i, m)$, and $\xi_t(i, j)$ are the probability of being in the j -th state at time t ,

the probability of being in the m -th substate of the i -th state at time t , and the probability of being in the i -th state at time t and j -th state at time $t + 1$, respectively, that is

$$\begin{aligned}\gamma_t(i) &= p(\mathbf{O}, q_t = i \mid \Lambda) \\ &= \frac{\alpha_t(i)\beta(i)}{\sum_{j=1}^N \alpha_t(j)\beta_t(j)}, \quad \begin{array}{l} 1 \leq i \leq N \\ t = 1, \dots, T \end{array} \end{aligned} \quad (2.41)$$

$$\begin{aligned}\gamma_t(i, m) &= p(\mathbf{O}, q_t = i, s_t = m \mid \Lambda) \\ &= \frac{\alpha_t(i)\beta(i)}{\sum_{j=1}^N \alpha_t(j)\beta_t(j)} \cdot \frac{w_{im}\mathcal{N}(\mathbf{O}_t \mid \boldsymbol{\mu}_{im}, \boldsymbol{\sigma}_{im})}{\sum_{k=1}^M w_{ik}\mathcal{N}(\mathbf{O}_t \mid \boldsymbol{\mu}_{ik}, \boldsymbol{\sigma}_{ik})}, \quad \begin{array}{l} 1 \leq i \leq N \\ 1 \leq m \leq M \\ t = 1, \dots, T \end{array} \end{aligned} \quad (2.42)$$

$$\begin{aligned}\xi_t(i, j) &= p(\mathbf{O}, q_t = i, q_{t+1} = j \mid \Lambda) \\ &= \frac{\alpha_t(i)a_{ij}b_j(\mathbf{O}_{t+1})\beta_{t+1}(j)}{\sum_{l=1}^N \sum_{n=1}^N \alpha_t(l)a_{ln}b_n(\mathbf{O}_{t+1})\beta_{t+1}(n)}, \quad \begin{array}{l} 1 \leq i \leq N \\ t = 1, \dots, T \end{array} \end{aligned} \quad (2.43)$$

2.5 Summary

In this chapter, the basic theories of the hidden Markov models (HMMs), its algorithm for calculating the output probability (forward-backward algorithm), searching the optimal state sequence (Viterbi algorithm), and estimating its parameters (EM algorithm) are described. Following chapters show the HMMs for acoustic modeling in speech recognition.

Chapter 3

HMM-based speech recognition

Most of the current speech recognition systems uses HMMs as its acoustic model. In this chapter, statistical speech recognition framework based on the HMM is described. General speech recognition systems may be divided into five basic blocks: the front-end, acoustic models, language models, lexicon and search algorithm. These blocks are introduced in more detail in the following sections.

3.1 Statistical speech recognition

The goal of large vocabulary continuous speech recognition (LVCSR) systems is to take an acoustic waveform as its input and generate a transcription of the words being uttered. First, the speech waveform is recorded and sampled by a digital device. Next, processor converts the sampled waveform into an observation vector sequence $\mathbf{O} = \{\mathbf{O}_1, \dots, \mathbf{O}_T\}$ by removing redundant or unimportant informations such as noises. There is a large amount of variability in observation vector sequences even if the same words were uttered by the same speaker. Therefore, a statistical approach is adopted to map the observation vector sequence into the most likely word sequence. The speech recognition system usually choose the word sequence, $\mathbf{w} = \{w_1, \dots, w_L\}$, with the maximum a posteriori (MAP) probability given the observation sequence as follows:

$$\hat{\mathbf{w}} = \arg \max_{\mathbf{w}} P(\mathbf{w} | \mathbf{O}) \quad (3.1)$$

Recently, discriminative models such as maximum entropy Markov models (MEMMs) [21] or conditional random fields (CRFs) [22] have been applied for modeling $P(\mathbf{w} | \mathbf{O})$ directly [23,24]. However, applying the discriminative models for LVCSR is still difficult

due to variabilities of the observation vector sequences and the vast number of possible word sequences. Therefore, most of the current speech recognition systems uses generative models rather than the discriminative ones. By using Bayes' rule, Eq. (3.1) can be written as

$$P(\mathbf{w} | \mathbf{O}) = \frac{p(\mathbf{O} | \mathbf{w}) P(\mathbf{w})}{p(\mathbf{O})}. \quad (3.2)$$

Since $p(\mathbf{O})$ is independent of the word sequence \mathbf{w} , the MAP decoding rule of Eq. (3.1) is

$$\hat{\mathbf{w}} = \arg \max_{\mathbf{w}} p(\mathbf{O} | \mathbf{w}) P(\mathbf{w}). \quad (3.3)$$

A general statistical speech recognition system may be described by the formulation in Eq. (3.3). The system consists of five main blocks: the front-end, acoustic models, language models, pronunciation lexicon and search algorithm.

The first term in Eq. (3.3), $p(\mathbf{O} | \mathbf{w})$, corresponds to the acoustic model, as it estimates the probability of an observation vector sequence \mathbf{O} , conditioned on the word sequence \mathbf{w} . For large vocabulary continuous speech recognition, the way of $p(\mathbf{O} | \mathbf{w})$ computation is to build statistical models for sub-word speech units, build up word models from these sub-word speech units using a pronunciation lexicon, and then postulate word sequences and evaluate the acoustic model probabilities of concatenated word models. It is possible to use any kind of models for $p(\mathbf{O} | \mathbf{w})$. Currently, context-dependent sub-word HMMs are used for most of speech recognition systems as its acoustic model.

The second term in Eq. (3.3), $P(\mathbf{w})$, corresponds to the language model, as it describes the probability associated with a postulated sequence of words. Generally language models are represented in a finite state network so as to be integrated into the acoustic model in a straightforward manner.

The final block, the search algorithm, implements the maximization in Eq. (3.3).

3.2 Front-ends

Comparing the sampled acoustic waveforms is difficult due to varying speaker and acoustic characteristics. However, the spectral shape of the speech signal have most of the important information [25]. Front-end of speech recognition systems generate observation vector sequences which represent the short-term spectrum of the speech signal. There

are many techniques for parameterizing speech spectra, i.e., linear prediction coefficients (LPC) [26,27], line spectral pair (LSP), cepstrum [28], mel-cepstrum [29], and so on. Mel filterbank cepstral coefficients (MFCC) [30] or perceptual linear prediction (PLP) [31] is generally used in most of the current speech recognition systems. In all cases the speech signal is assumed to be quasi-stationary so that it can be decided into short frames. In each frame period a new parameterized short-time spectra vector is produced by analyzing a speech segment. In a final step, delta and delta-delta coefficients are appended to the acoustic vector [32–35]. The delta and delta-delta coefficients are usually calculated as regression coefficients from their neighboring static features as follows:

$$\Delta \mathbf{c}_t = \sum_{\tau=-L_-^{(1)}}^{L_+^{(1)}} w^{(1)}(\tau) \mathbf{c}_{t+\tau}, \quad \Delta^2 \mathbf{c}_t = \sum_{\tau=-L_-^{(2)}}^{L_+^{(2)}} w^{(2)}(\tau) \mathbf{c}_{t+\tau}, \quad (3.4)$$

where \mathbf{c}_t , $\Delta \mathbf{c}_t$, and $\Delta^2 \mathbf{c}_t$ are static, delta, and delta-delta coefficients at time t , respectively, and $\{w^{(d)}(\tau)\}_{d=1,2} \tau=-L_-^{(d)}, \dots, L_+^{(d)}$ are regression window coefficients to calculate the d -th order dynamic feature. As a result, the observation vector at time t , \mathbf{O}_t , consists of static and dynamic features as

$$\mathbf{O}_t = [\mathbf{c}_t^\top, \Delta \mathbf{c}_t^\top, \Delta^2 \mathbf{c}_t^\top]^\top. \quad (3.5)$$

3.3 HMM-based acoustic modeling

The HMMs are used to provide the estimates of $p(\mathbf{O} | \mathbf{w})$ in the speech recognition systems. For isolated word recognition with sufficient training data, an HMM can be trained for each word. However, for LVCSR tasks, it is unlikely that there are enough training examples of each word in the dictionary. Therefore, sub-word units such as phone or syllable is used. An HMM is generally trained for each phone. The HMMs corresponding to the phone sequence may then be concatenated to form a composite model representing words and sentences.

When the HMMs are trained for the set of phones, it is referred to as a monophone or context-independent system. However, there is a large amount of variation between realizations of the same phone depending on the previous and next phones. Triphones which take the previous and next phones into account are commonly used as context-dependent phones. The number of states and model parameters of a triphone system is significantly higher than a monophone system. However, it is unlikely that sufficient training data is

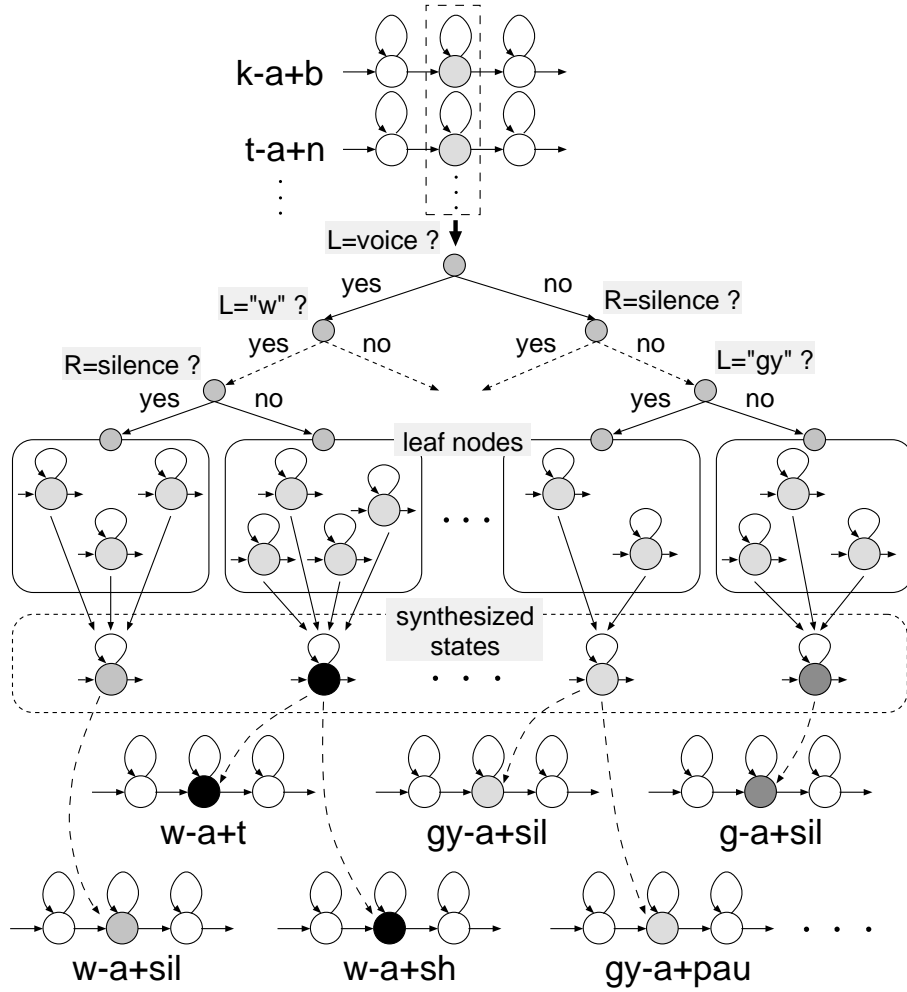


Figure 3.1: Example of a phonetic decision tree for triphone models.

available for parameter estimation. To avoid this problem, the state output probability distributions are generally shared.

A phonetic decision tree [36–38] is generally used to construct state tying structure in context-dependent systems (Figure 3.1). First, all phones are pooled in the root node. Next, the state clusters are split based on contextual questions. When the number of training data per state falls below a threshold, the splitting will terminate. A disadvantage of decision tree-based state clustering is that the splits maximize the likelihood of the training data locally [39, 40].

3.4 Word N -gram-based language modeling

The language model provides $P(\mathbf{w})$ in the speech recognition systems. Using chain rule, this can be expressed as

$$P(\mathbf{w}) = \prod_{l=1}^L P(w_l | w_{l-1}, \dots, w_1). \quad (3.6)$$

To reduce the number of parameters, different histories can be divided into equivalence class using a function $h(w_{l-1}, \dots, w_1)$. In general, equivalence classes are defined by truncating the history to $N - 1$ words. These word N -gram language models are defined as

$$P(\mathbf{w}) = \prod_{l=1}^L P(w_l | w_{l-1}, \dots, w_{l-N+1}). \quad (3.7)$$

Standard values are $N = 2, 3$ which are called bi-gram or tri-gram models, respectively. The N -grams are estimated by counting relative frequencies from text corpus. For a vocabulary of V words, there are still V^N N -gram models. Word sequences can be assigned a zero probability for given a finite training data. Many smoothing technique such as discounting, backing off, and deleted interpolation have been proposed [41].

In the speech recognition systems, there is often a mismatch between the acoustic and language model. Dynamic ranges is different between the discrete probability, $P(\mathbf{w})$, estimated from a text corpus and the acoustic likelihood, $p(\mathbf{O} | \mathbf{w})$, obtained from high dimensional observation densities. For this mismatch, the language model probability is generally increased by a constant called the grammar scale factor. The speech recognition system also tend to output short words result in many insertion errors. To compensate this problem, an insertion penalty which reduce the total score $p(\mathbf{O} | \mathbf{w}) P(\mathbf{w})$ depending on the number of hypothesized words in the sequence is generally used. By taking these modifications into account in Eq. (3.3), a practical speech recognition system uses

$$\hat{\mathbf{w}} = \arg \max_{\mathbf{w}} \left[\log \{p(\mathbf{O} | \mathbf{w})\} + \alpha \log \{P(\mathbf{w}) + \beta L\} \right] \quad (3.8)$$

where α , β , and L are the grammar scale factor, the insertion penalty, and the total number of words, respectively. The α and β are empirically set.

3.5 Pronunciation lexicon

Each word is defined by a pronunciation obtained from a dictionary. The word HMM is the concatenation of the relevant sequence of sub-word HMMs. The lexicon is stored as a tree for computational efficiency. A tree-based lexicon have been used in various speech recognition system. Tree-based lexicon allows pronunciations with similar heads to share memory when being evaluated. Therefore, different pronunciations of the same word are stored as separate lexical items. The disadvantage of using the tree-based lexicon is that it is not an efficient approach to represent multiple pronunciations of the same word.

3.6 Search algorithms

To determine the word sequene yielding maximum combined probability from the acoustic and language model, the following problems must be resolved.

1. The number of words in given utterance is unknown.
2. Word boundaries in given utterance are also unknown.
3. The word boundaries are often fuzzy.
4. For a set of V word-reference patterns and L words in the utterance, there are $V L$ possible combinations of composite matching patterns.

To solve these problems, efficient search algorithm have been proposed. Most of these algorithms can categorized into two basic classes: Viterbi decoding [42] and stack decoding [43].

3.7 Summary

In this chapter, the statistical speech recognition framework and its main modules, front-ends, acoustic modeling, language modeling, and search algorithm, are described. Following chapter show the HMMs for acoustic modeling in speech synthesis.

Chapter 4

Speech recognition based on statistical models including multiple phonetic decision trees

To optimize state sequences, the EM and DAEM algorithms require a parameter tying structure. However, the parameter tying structure is usually constructed from unreliable model parameters, because an appropriate model structure has not yet been constructed for estimating model parameters. This means that the estimation of state sequences and the construction of model structures depend on each other. Hence, they should be optimized simultaneously. However, the exact solution of this optimization is computationally intractable. Consequently, I reformulate this optimization problem as a maximization of a newly defined likelihood function that includes multiple model structures.

4.1 Deterministic annealing EM algorithm in parameter estimation

4.1.1 EM algorithm

The objective of the EM algorithm is to estimate a set of model parameters that maximizes the incomplete log-likelihood function:

$$\mathcal{L}(\Lambda) = \log \sum_{\mathbf{q}} P(\mathbf{o}, \mathbf{q} \mid \Lambda), \quad (4.1)$$

where $\mathbf{o} = (\mathbf{o}_1, \mathbf{o}_2, \dots, \mathbf{o}_T)$ and $\mathbf{q} = (q_1, q_2, \dots, q_T)$ are the observation and state sequences, respectively, and Λ denotes a set of model parameters. The EM algorithm itera-

tively maximizes the auxiliary function, the Q -function:

$$Q(\Lambda, \Lambda') = \sum_{\mathbf{q}} P(\mathbf{q} | \mathbf{o}, \Lambda) \log P(\mathbf{o}, \mathbf{q} | \Lambda'), \quad (4.2)$$

where $P(\mathbf{q} | \mathbf{o}, \Lambda)$ is the posterior probability of \mathbf{q} . It can be obtained by applying the Bayes rule as follows:

$$P(\mathbf{q} | \mathbf{o}, \Lambda) = \frac{P(\mathbf{o}, \mathbf{q} | \Lambda)}{\sum_{\mathbf{q}} P(\mathbf{o}, \mathbf{q} | \Lambda)}. \quad (4.3)$$

The EM algorithm starts with an initial model parameter $\Lambda^{(0)}$, and iterates between the following two steps.

E step: compute $Q(\Lambda, \Lambda^{(k)})$

M step: $\Lambda^{(k+1)} = \arg \max_{\Lambda} Q(\Lambda, \Lambda^{(k)})$

Here, k denotes the iteration number. This procedure is repeated until the convergence of the likelihood. However, since the EM algorithm is a hill-climbing approach, it sometimes suffers from the local maxima problem.

4.1.2 Deterministic annealing EM algorithm

In the DAEM algorithm [10], the problem of maximizing the log-likelihood function is reformulated as the problem of minimizing the following free energy function:

$$\begin{aligned} \mathcal{F}_{\beta}(\Lambda) &= -\frac{1}{\beta} \log \sum_{\mathbf{q}} P^{\beta}(\mathbf{o}, \mathbf{q} | \Lambda) \\ &= -\sum_{\mathbf{q}} f(\mathbf{q} | \mathbf{o}, \Lambda) \log P(\mathbf{o}, \mathbf{q} | \Lambda) \\ &\quad - \frac{1}{\beta} I[f(\mathbf{q} | \mathbf{o}, \Lambda)], \end{aligned} \quad (4.4)$$

where $I[x]$ denotes the entropy of x and $1/\beta$ is called “temperature.” If $\beta = 1$, the negative free energy $-\mathcal{F}_{\beta}(\Lambda)$ becomes equal to the log-likelihood function $\mathcal{L}(\Lambda)$. In the deterministic annealing approach, the new posterior distribution f is derived so as to minimize the free energy under the constraint of $\sum_{\mathbf{q}} f = 1$. To solve this problem, we can use the elementary calculus of variations to take functional derivatives of Eq. (4.4) with respect to f , and the optimal distribution can be derived as

$$f(\mathbf{q} | \mathbf{o}, \Lambda) = \frac{P^{\beta}(\mathbf{o}, \mathbf{q} | \Lambda)}{\sum_{\mathbf{q}} P^{\beta}(\mathbf{o}, \mathbf{q} | \Lambda)}. \quad (4.5)$$

In the DAEM algorithm, the temperature parameter β is gradually increased while iterating the EM-steps at each temperature. The procedure of the DAEM algorithm is as follows:

1. Give an initial model, and set $\beta = \beta^{(0)}$.
2. Iterate EM-steps with β fixed until $\mathcal{F}_\beta(\Lambda)$ converged:
 - (E step) compute
 - (M step) $\Lambda^{(k+1)} = \arg \max_{\Lambda} I[f(\mathbf{q} \mid \mathbf{o}, \Lambda)]$.
3. Increase β .
4. If $\beta > 1$, stop the procedure. Otherwise go to step 2.

When $1/\beta$ is set to the initial temperature $\beta^{(0)} \simeq 0$, the EM-steps may achieve a single global minimum of $\mathcal{F}_\beta(\Lambda)$. At the initial temperature, the posterior distribution f takes a form of nearly uniform distribution. While the temperature is decreasing, the form of f changes from uniform to the original posterior distribution. Finally, at the temperature $1/\beta = 1$, the DAEM algorithm is identical to the original EM algorithm. Similarly to the EM algorithm, the DAEM algorithm is also guaranteed to converge at a fixed temperature by decreasing $\mathcal{F}_\beta(\Lambda)$.

4.1.3 Optimization of state sequences

In the HMM case, the DAEM posterior distribution f can be calculated by the forward-backward algorithm. The numerator of the posterior distribution in Eq. (4.5) is written as

$$\begin{aligned}
 P^\beta(\mathbf{o}, \mathbf{q} \mid \Lambda) &= P^\beta(\mathbf{o} \mid \mathbf{q}, \Lambda) P^\beta(\mathbf{q} \mid \Lambda) \\
 &= \prod_{t=1}^T P^\beta(\mathbf{o}_t \mid q_t, \Lambda) \prod_{t=1}^T P^\beta(q_t \mid q_{t-1}, \Lambda),
 \end{aligned} \tag{4.6}$$

where $P(\mathbf{o}_t \mid q_t, \Lambda)$ and $P(q_t \mid q_{t-1}, \Lambda)$ indicate state output and transition probabilities, respectively. It can be observed that Eq. (4.6) has the same form as the likelihood function of HMMs. Therefore, the expectations with respect to the DAEM posterior distribution f can be calculated by replacing the state output and transition probabilities with $P^\beta(\mathbf{o}_t \mid q_t, \Lambda)$ and $P^\beta(q_t \mid q_{t-1}, \Lambda)$, respectively.

4.2 Speech recognition based on multiple phonetic decision trees

To optimize state sequences, the EM and DAEM algorithms require a parameter tying structure. However, the parameter tying structure is usually constructed from unreliable model parameters, because an appropriate model structure has not yet been constructed for estimating model parameters. This means that the estimation of state sequences and the construction of model structures depend on each other. Hence, they should be optimized simultaneously. However, the exact solution of this optimization is computationally intractable. Consequently, we reformulate this optimization problem as a maximization of a newly defined likelihood function that includes multiple model structures.

4.2.1 Acoustic modeling based on model structure annealing

To derive the algorithm of model structure annealing, we define a new likelihood function that includes parameter tying structures as a hidden variable as follows:

$$P(\mathbf{o} \mid \Lambda) = \sum_{\mathbf{q}} \sum_m P(\mathbf{o}, \mathbf{q}, m \mid \Lambda), \quad (4.7)$$

$$P(\mathbf{o}, \mathbf{q}, m \mid \Lambda) = P(m)P(\mathbf{q} \mid \Lambda)P(\mathbf{o} \mid \mathbf{q}, m, \Lambda), \quad (4.8)$$

where $m \in \{1, \dots, M\}$ are indexes of parameter tying structures and $\Lambda \in \{\Lambda_1, \dots, \Lambda_M\}$ denotes a set of model parameters. We assume each parameter tying structure is represented by a phonetic decision tree. In the EM algorithm, the ML estimation of the model parameters is obtained using the posterior distribution of hidden variables estimated in the E-step. Therefore, the ML solution for the newly defined model is regarded as the simultaneous optimization of state sequences and a parameter tying structure. The free energy function including the multiple decision trees for the DAEM algorithm also can be written as

$$\mathcal{F}_\beta(\Lambda) = -\frac{1}{\beta} \log \sum_{\mathbf{q}} \sum_m P^\beta(\mathbf{o}, \mathbf{q}, m \mid \Lambda). \quad (4.9)$$

However, estimating the DAEM posterior distribution $f(\mathbf{q}, m \mid \mathbf{o}, \Lambda)$ is intractable owing to the combination of hidden variables. To solve this problem, we apply the variational EM algorithm [44]. The objective of the algorithm is to minimize the upper bound of the

free energy function. The upper bound of the free energy function $\bar{\mathcal{F}}_\beta(\Lambda)$ is defined as

$$\begin{aligned}\mathcal{F}_\beta(\Lambda) &= -\frac{1}{\beta} \log \sum_{\mathbf{q}} \sum_m Q(\mathbf{q}, m) \frac{P^\beta(\mathbf{o}, \mathbf{q}, m \mid \Lambda)}{Q(\mathbf{q}, m)} \\ &\leq -\frac{1}{\beta} \sum_{\mathbf{q}} \sum_m Q(\mathbf{q}, m) \log \frac{P^\beta(\mathbf{o}, \mathbf{q}, m \mid \Lambda)}{Q(\mathbf{q}, m)} \\ &= \bar{\mathcal{F}}_\beta(\Lambda),\end{aligned}\tag{4.10}$$

where $Q(\mathbf{q}, m)$ is an arbitrary distribution. The upper bound $\bar{\mathcal{F}}_\beta(\Lambda)$ can be transformed as follows:

$$\bar{\mathcal{F}}_\beta(\Lambda) = \frac{1}{\beta} KL(Q \parallel f) - \log P(\mathbf{o} \mid \Lambda) + \text{const},\tag{4.11}$$

where $KL(\parallel)$ denotes the Kullback-Leibler (KL) divergence. The above equation shows that minimizing $\bar{\mathcal{F}}_\beta(\Lambda)$ with respect to $Q(\mathbf{q}, m)$ is equivalent to minimizing the KL-divergence between Q and f . If there is no constraint with distribution Q , minimizing $\bar{\mathcal{F}}_\beta(\Lambda)$ results in $f = Q$. Assuming a constraint to reduce the complexity, the distribution Q that minimizes $\bar{\mathcal{F}}_\beta(\Lambda)$ becomes an approximate distribution of f . Hence, we assume the following constraint:

$$Q(\mathbf{q}, m) = Q(\mathbf{q})Q(m),\tag{4.12}$$

where $\sum_{\mathbf{q}} Q(\mathbf{q}) = 1$ and $\sum_m Q(m) = 1$. Using these factorized distributions, the upper bound $\bar{\mathcal{F}}_\beta(\Lambda)$ can be rewritten as

$$\begin{aligned}\bar{\mathcal{F}}_\beta(\Lambda) &= -\sum_{\mathbf{q}} \sum_m Q(\mathbf{q})Q(m) \log P(\mathbf{o}, \mathbf{q}, m \mid \Lambda) \\ &\quad - \frac{1}{\beta} I[Q(\mathbf{q})] - \frac{1}{\beta} I[Q(m)].\end{aligned}\tag{4.13}$$

It can be seen that the temperature parameter β changes the ratio between the value of the Q -function and the entropy of hidden variables in $\bar{\mathcal{F}}_\beta(\Lambda)$. Extending this interpretation, we can control the annealing process of decision trees and state sequences individually. By introducing $\beta_{\mathbf{q}}$ and β_m , $\bar{\mathcal{F}}_\beta(\Lambda)$ is rewritten as

$$\begin{aligned}\bar{\mathcal{F}}_\beta(\Lambda) &= -\sum_{\mathbf{q}} \sum_m Q(\mathbf{q})Q(m) \log P(\mathbf{o}, \mathbf{q}, m \mid \Lambda) \\ &\quad - \frac{1}{\beta_{\mathbf{q}}} I[Q(\mathbf{q})] - \frac{1}{\beta_m} I[Q(m)].\end{aligned}\tag{4.14}$$

The optimal variational posterior distributions $Q(\mathbf{q})$ and $Q(m)$ are derived by minimizing $\bar{\mathcal{F}}_\beta(\Lambda)$. This functional optimization can be solved by the variational method, and the

following formulae are obtained:

$$Q(\mathbf{q}) \propto P^{\beta_q}(\mathbf{q} \mid \Lambda) \exp \left\langle \log P^{\beta_q}(\mathbf{o} \mid \mathbf{q}, m, \Lambda) \right\rangle_{Q(m)}, \quad (4.15)$$

$$Q(m) \propto P^{\beta_m}(m) \exp \left\langle \log P^{\beta_m}(\mathbf{o} \mid \mathbf{q}, m, \Lambda) \right\rangle_{Q(\mathbf{q})}, \quad (4.16)$$

where $\langle \cdot \rangle_{Q(\cdot)}$ denotes the expectation with respect to the distribution $Q(\cdot)$. Since Eqs. (4.15) and (6.7) are dependent on each other, these updates should be iterated in the E-step. Figure 4.1 illustrates the joint optimization process based on the DAEM algorithm. At the initial temperature ($\beta_q^{(0)}, \beta_m^{(0)} \simeq 0$), the variational posterior distributions $Q(\mathbf{q})$ and $Q(m)$ take forms with nearly uniform distribution. While the temperature is decreasing, the forms of $Q(\mathbf{q})$ and $Q(m)$ change from uniform to each original posterior distribution, and at the final temperature ($\beta_q, \beta_m = 1$), $Q(\mathbf{q})$ and $Q(m)$ have different original posterior distributions. Then, the posterior probability of each model structure is in proportion to the likelihood of each model structure. This process represents the approximation of the joint optimization of the state sequences and the model structures.

4.2.2 Speech decoding based on multiple model structures

In the proposed method, multiple decision trees are used in decoding process. However, the multiple decision trees are inapplicable to standard decoders. Therefore, we propose two types of decoding procedures. One is that a single model structure is chosen by setting the temperature β_m to ∞ (the DAEM algorithm with $\beta_q = \infty$ becomes the Viterbi training. However, the final temperature is fixed as $\beta_q = 1$ in this paper). Although the model structure with the largest decision tree is selected at $\beta_m = \infty$ in most cases, reliable state sequences can be obtained by using multiple model structures in the early stage of the training procedure. The other is to use multiple model structures not only in the training process but also in the decoding process. Although there are many approaches to using multiple model structures in decoding [13] [?], we use Eq. (4.7) to control the degree of use of multiple decision trees for training and decoding processes. In preliminary experiments, there was a tendency to select only the largest decision tree at the final stage of the training process. This is because the range of the likelihood was very different among the differently sized decision trees, i.e., the largest decision tree had a significantly higher likelihood than the other trees. Consequently, the posterior probability $Q(m)$ of the largest decision tree became almost 1, and the other trees were not used. Therefore, to use multiple decision trees in the decoding process, we adopt a method in which the annealing is stopped in the early stage of the training process. In this method, the decoding is performed so as to minimize the upper bound $\bar{\mathcal{F}}_\beta$. Using $Q(m)$, the criterion for decoding

can be written as

$$\max_{\mathbf{q}} P(\mathbf{q} \mid \Lambda) \prod_m P^{Q(m)}(\mathbf{o} \mid \mathbf{q}, m, \Lambda). \quad (4.17)$$

By inspection, this criterion can be calculated by the output probabilities of a multistream HMM where $Q(m)$ becomes the weight of each stream.

4.3 Experiments

4.3.1 Speaker dependent phoneme recognition

Experimental condition

In this experiment, I used 503 phonetically balanced sentences uttered by a single male speaker MHT from the ATR Japanese speech database b-set [45]. For training, 450 sentences were used and the remaining 53 sentences were used for testing. The speech data was down-sampled from 20 kHz to 16 kHz, windowed at a 25 ms Blackman window, and parameterized into 19 mel-cepstral coefficients by the mel-cepstral analysis technique [29]. Static coefficients including the zeroth coefficients and their first and second derivatives were used as feature parameters. Three-state left-to-right HMMs were used to model 37 Japanese phonemes, and 144 questions were prepared for decision tree clustering. Each state output probability distribution was modeled by a Gaussian distribution with a diagonal covariance matrix. As a decoder, HVite in HTK [16] was used.

In this experiment, the following five training methods were compared.

- “flat-start”: HMMs were initialized with equal mean and variance for all states using no phoneme boundary labels, and re-estimated using the EM algorithm.
- “ k -means”: HMMs were initialized by the segmental k -means algorithm using phoneme boundary labels and re-estimated using the EM algorithm.
- “DAEM-state”: The DAEM algorithm was applied only to the estimation of state sequences. A single decision tree was used.
- “DAEM-tree”: The DAEM algorithm was applied only to decision trees. The estimation process of state sequences is equivalent to “flat-start.”
- “DAEM-joint”: The DAEM algorithm was applied to both state sequences and decision trees.

In the methods using a single decision tree (“flat-start,” “ k -means,” and “DAEM-state”), a decision tree is obtained by context clustering based on the minimum description length (MDL) criterion [46]. In addition to this model structure, “DAEM-tree” and “DAEM-joint” use a decision tree representing monophone HMMs. It is desirable to use multiple decision trees. However, when several decision trees are used, I must determine many conditions (e.g., the size and structure of trees, and the number of trees). Although how to determine the number of decision trees and how to construct multiple decision trees are essential problems in the proposed method, in this experiment, I only focus on the evaluation of the integration part of multiple decision trees. Therefore, in this experiment, I simply use only two decision trees for model structure annealing ($m = 1$: monophone, $m = 2$: MDL). In the two-decision-tree case, determining the temperature parameter β_m is equivalent to setting the variational posterior probabilities $Q(m)$ directly, because the update equation of $Q(m)$ includes β_m in Eq. (16), and the ratio between $Q(1)$ and $Q(2)$ is determined by β_m . Therefore, $Q(m)$ can also be arbitrarily determined instead of β_m , and at the start and end of the temperature update, $Q(m)$ should be fixed as follows. When β_m is set to 0, all decision trees have the same posterior probabilities ($Q(1) = 0.5$ and $Q(2) = 0.5$). When β_m is set to 1, $Q(m)$ is in proportion to the likelihood of each decision tree. However, since MDL has a much higher likelihood than a monophone, the posterior probabilities should be $Q(1) = 0$ and $Q(2) = 1$. Therefore, it was assumed that $Q(m)$ was updated by the following linear functions:

$$Q(\text{monophone}) = 0.5 \left(1 - \frac{i}{I} \right), \quad (4.18)$$

$$Q(\text{MDL}) = 0.5 \left(1 + \frac{i}{I} \right). \quad (4.19)$$

The temperature parameter β_q was updated by

$$\beta_q(i) = \left(\frac{i}{I} \right)^\alpha, \quad (i = 0, \dots, I), \quad (4.20)$$

where i denotes the iteration number of temperature updates, and α was varied as $\alpha = 2^n$ ($n = -7, \dots, 7$). Figures 4.2 and 4.3 show plots of the schedules of the temperature parameters β_q and β_m , respectively. In the DAEM algorithm (“DAEM-state,” “DAEM-tree” and “DAEM-joint”), the number of temperature update steps was set to 20 ($I = 20$), and 10 EM-steps were conducted at each temperature. To evenly compare the proposed method with the conventional method, the number of EM-steps was set to 200 for the standard EM algorithm (“flat-start” and “ k -means”).

Experimental results

Figure 4.4 shows the log-likelihood of the training data. It can be seen that the likelihood of “flat-start” was lower than that of “ k -means.” This is because “flat-start” uses no phoneme boundary information for initializing HMMs and inappropriate initial model parameters cause the local maxima problem. Although the “DAEM-state” also uses no phoneme boundaries, the likelihood of the “DAEM-state” was close to that of “ k -means” when an appropriate temperature schedule was used. This result confirmed that the local maxima problem can be relaxed by using the DAEM algorithm. Comparisons of the proposed structure annealing with the conventional methods reveals that “DAEM-tree” yielded similar likelihoods of “ k -means” and the “DAEM-state.” Furthermore, “DAEM-joint” exhibited the highest likelihood at $\alpha = 2^2$. These results show that structure annealing can yield reliable estimates of state sequences with the use of multiple decision trees.

Figure 5.2 shows the phoneme accuracy of each method. It is noted that only one decision tree (MDL) is used for decoding. Similar to the likelihood, the phoneme accuracy of “flat-start” was worse than those of the other methods because of the local maxima problem. It can also be seen that the methods using the DAEM algorithm outperformed “ k -means,” even though phoneme boundary information was not used in the DAEM algorithm. Moreover, “DAEM-tree” and “DAEM-joint” had improved performance compared with the conventional “DAEM-state,” and an 11.1% relative error reduction was achieved for “DAEM-joint” over “ k -means” at $\alpha = 2^0$. This result indicates that the reliable HMM parameters estimated using structure annealing are effective for improving the speech recognition performance.

4.3.2 Speaker independent phoneme recognition

Experimental condition

To train speaker-independent HMM sets, I used 37,618 sentences uttered by 122 male and 122 female speakers from Japanese Newspaper Article Sentences (JNAS) [47] as the training data. Two hundred sentences uttered by 23 male and 23 female speakers from JNAS were used for testing. The speech data was windowed at a 25 ms Hamming window, and parameterized into 13 mel-cepstral coefficients by the mel-cepstral analysis technique. Static coefficients including the zeroth coefficients and their first and second derivatives were used as feature parameters. Three-state left-to-right HMMs were used to model 43 Japanese phonemes, and 261 questions were prepared for decision tree clustering. Each state output probability distribution was modeled by a Gaussian distribution

with a diagonal covariance matrix. In this experiment, Julius [?] was used as the decoder and a word forward 2-gram and a backward 3-gram were used as the language models.

The compared methods and the conditions for decision trees were the same as those in the speaker-dependent experiment. The number of EM-steps was set to 100 for the standard EM algorithm (“flat-start” and “ k -means”). In the DAEM algorithm (“DAEM-state,” “DAEM-tree” and “DAEM-joint”), the number of temperature update steps was set to 20 ($I = 20$), and 5 EM-steps were conducted at each temperature. The word insertion penalty and the language weight were adjusted to yield the best performance for each method.

Experimental results

Figure 4.6 shows the log-likelihood of the training data. In the speaker-independent experiment, the estimation of acoustic models is more difficult than that in the speaker-dependent experiment. Therefore, the local maxima problem becomes more serious. However, the likelihood of the DAEM methods (“DAEM-state,” “DAEM-tree” and “DAEM-joint”) were higher than that of “ k -means,” and “DAEM-joint” obtained the best value of the likelihood. This result indicates that not only using the DAEM algorithm but also using multiple decision trees is more effective for the local maxima problem in speaker-independent tasks.

Figure 4.7 shows the word accuracy of each method using a single decision tree (MDL) for decoding. It can be seen that the DAEM methods achieved higher accuracy than “flat-start” when the appropriate temperature schedule was used. Although “DAEM-joint” did not outperform “ k -means,” the proposed method is still effective because “DAEM-joint” uses no phoneme boundary information.

Figure 4.8 illustrates the word accuracy of each method in which multiple decision trees are used for decoding. From the figure, it can be seen that “DAEM-tree” and “DAEM-joint” achieved higher word accuracy than “ k -means.” This result suggests that even in the speaker-independent word recognition tasks, the proposed method can improve the performance of speech recognition.

4.4 Summary

In this chapter, we proposed a speech recognition technique using multiple decision trees. In the proposed method, speech recognition was performed by ML estimation of the

newly defined statistical model that includes multiple decision trees as a hidden variable. Applying the DAEM algorithm and using multiple decision trees in the early stage of the training process, reliable state sequences can be obtained. In continuous phoneme recognition experiments, the proposed technique improved the performance of speech recognition even when using only two decision trees. As future work, we will consider the optimization of the temperature schedules, investigate the effect of increasing the number of decision trees, and develop an approach for preparing multiple decision trees.

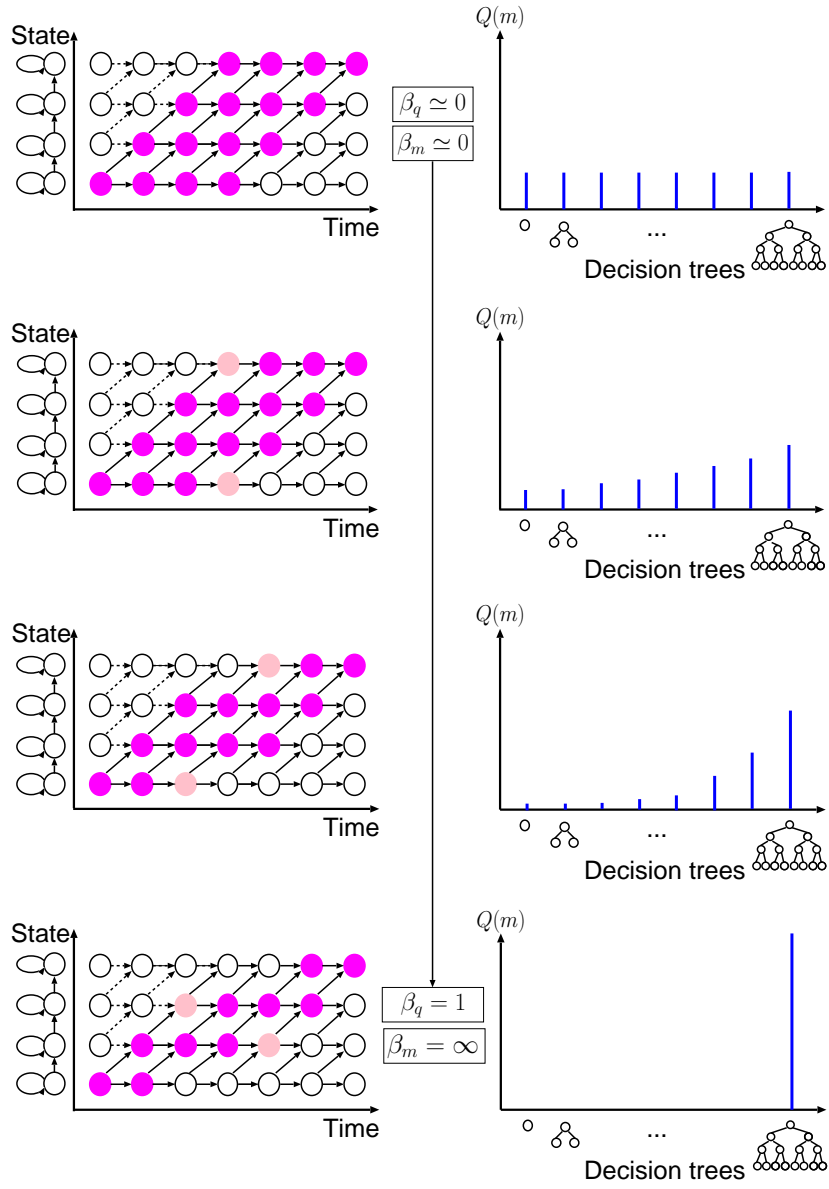


Figure 4.1: Joint optimization process

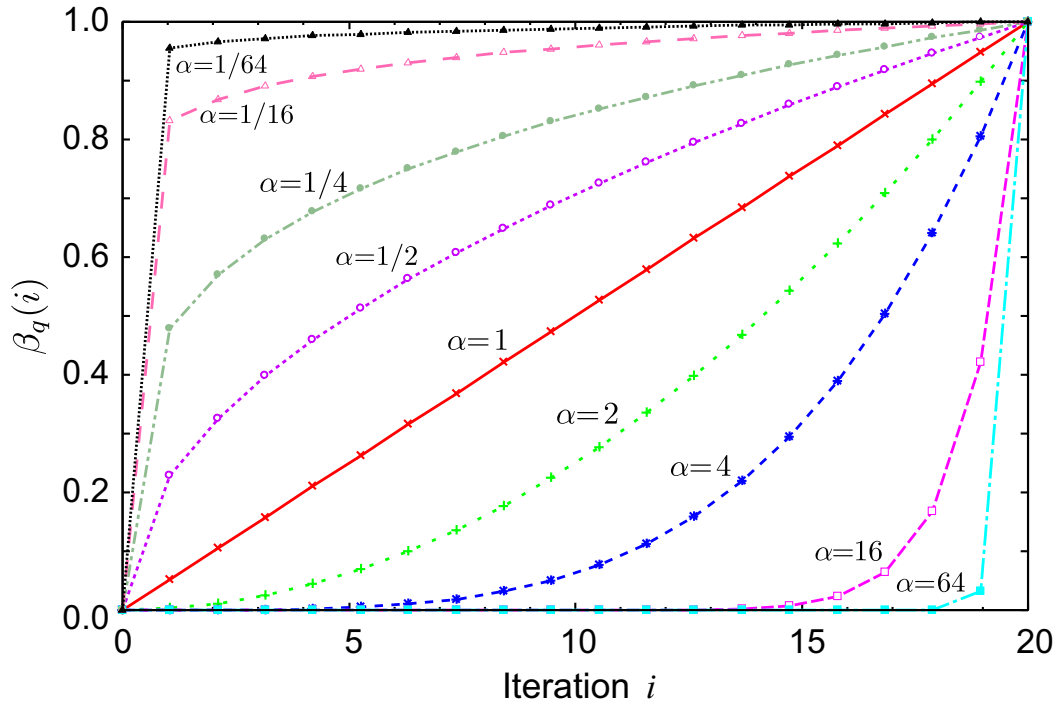


Figure 4.2: Schedule of temperature parameter β_q

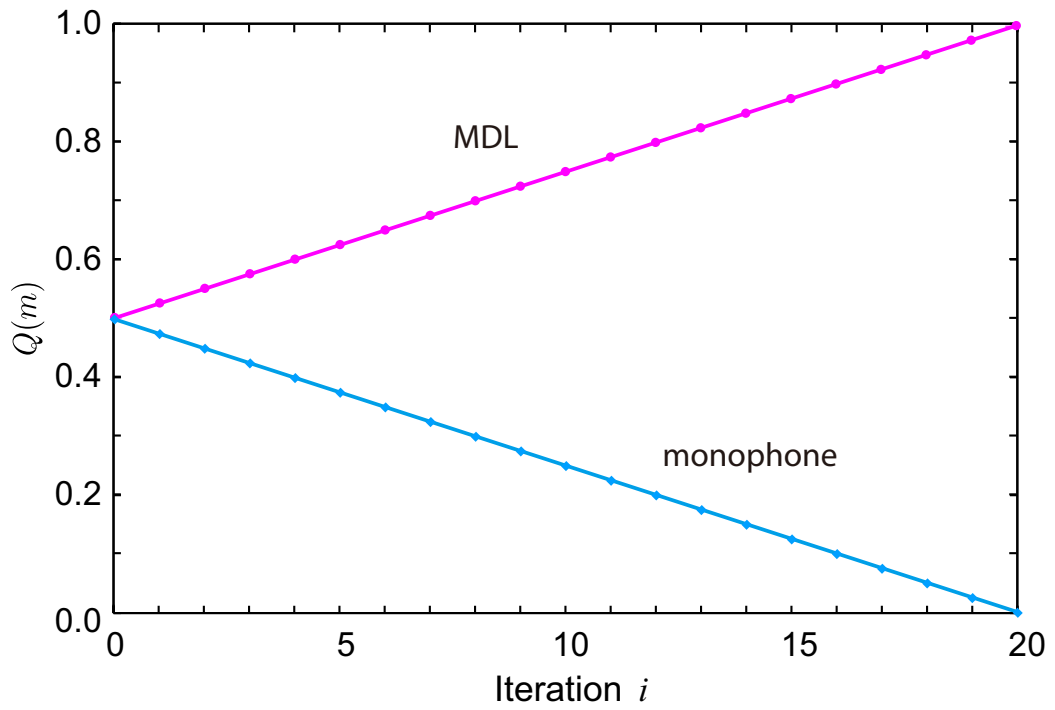


Figure 4.3: Schedule of update $Q(m)$

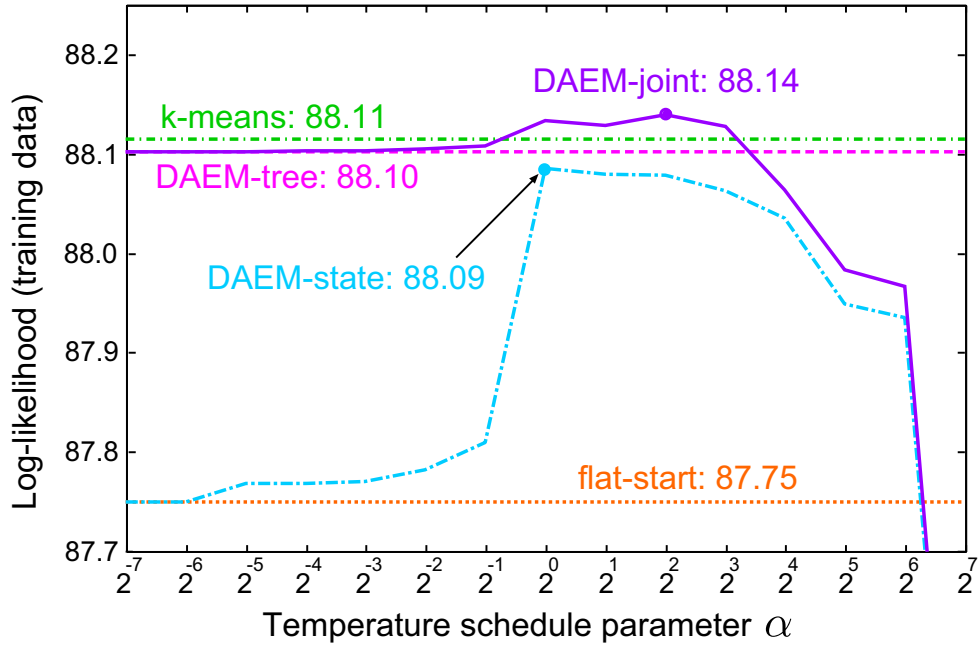


Figure 4.4: Log-likelihood of training data (Speaker dependent)

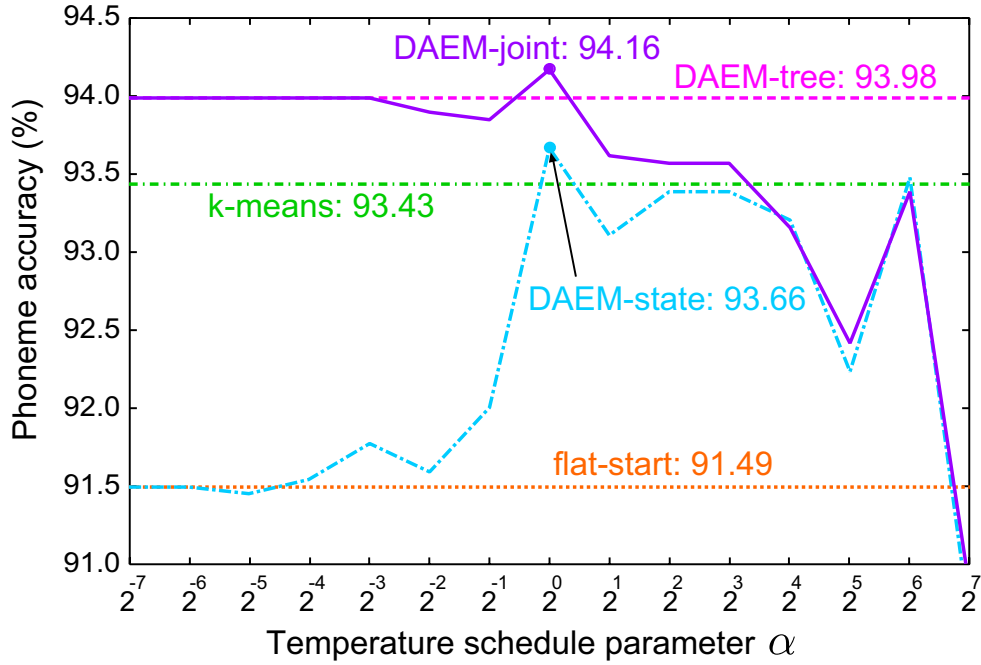


Figure 4.5: Phoneme accuracy for each temperature schedule

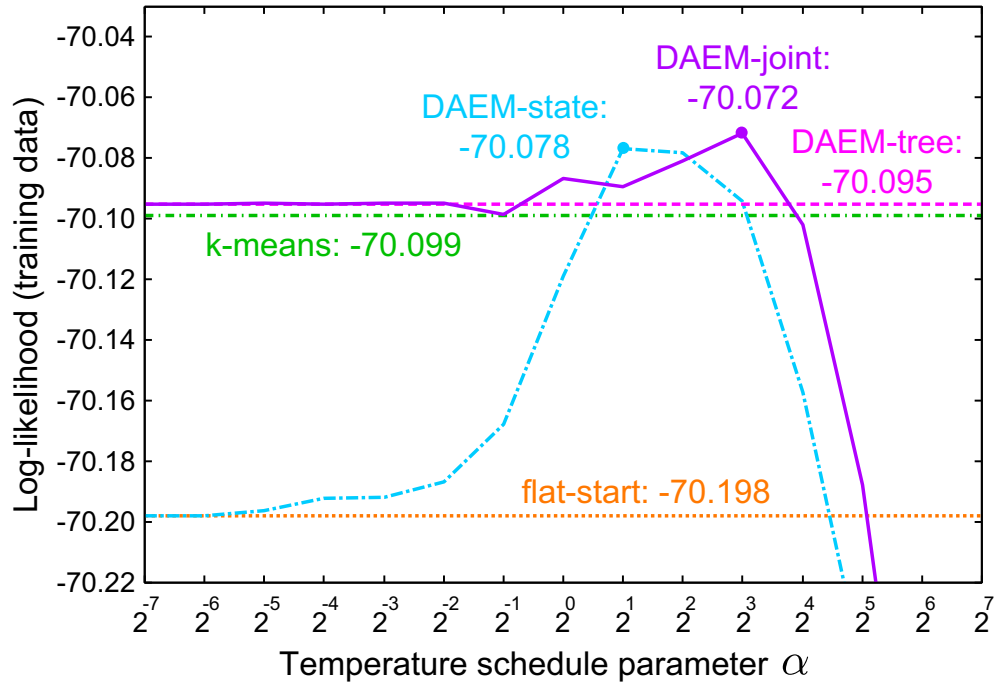


Figure 4.6: Log-likelihood of training data (Speaker independent)

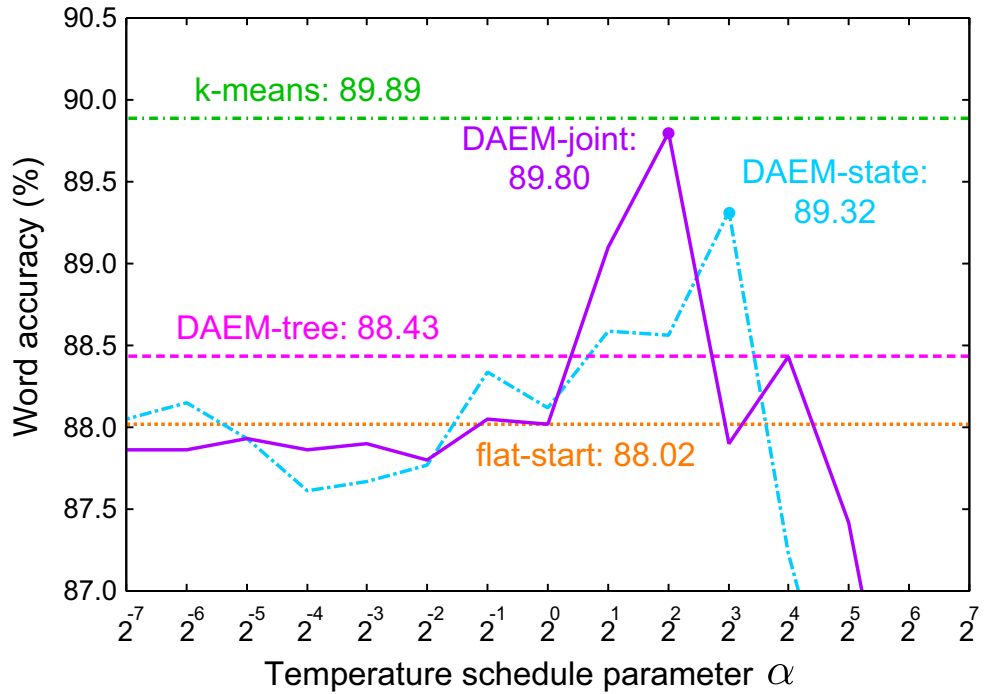


Figure 4.7: Word accuracy for each temperature schedule

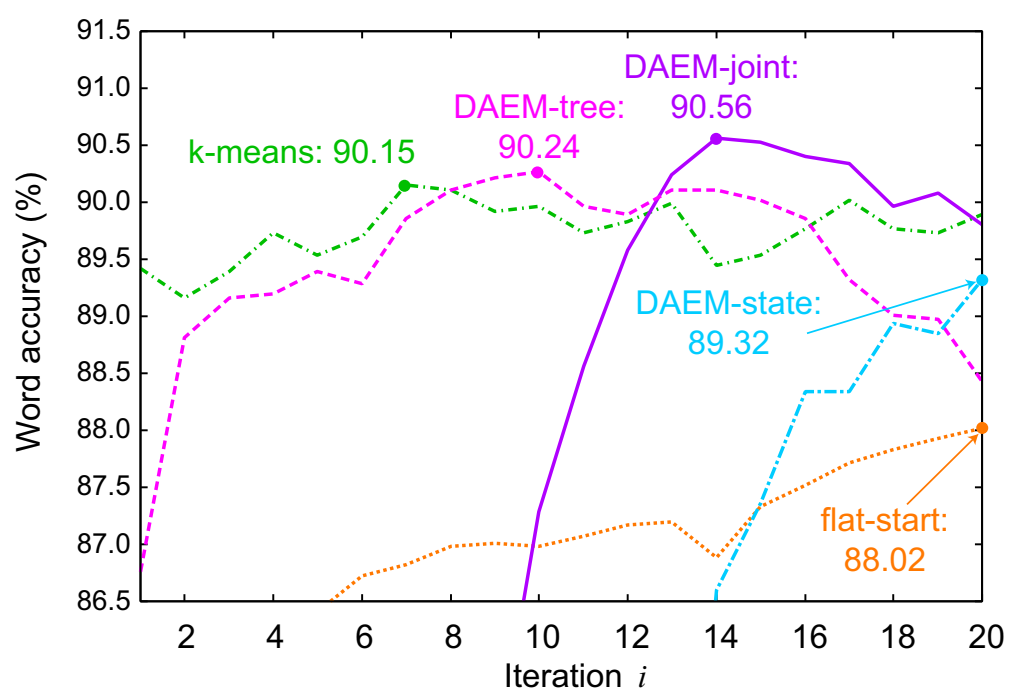


Figure 4.8: Word accuracy using multiple trees in decoding

Chapter 5

Speech recognition based on variational Bayesian method

The maximum likelihood (ML) criterion has usually been used for training statistical models for speech recognition systems. However, since the ML criterion produces a point estimate of model parameters, the estimation accuracy may degrade when little training data is available. The Bayesian approach is a statistical technique for estimating reliable predictive distributions by marginalizing model parameters, and it can accurately estimate observation distributions even if the amount of training data is small. However, the calculation becomes complicated due to the combination of latent variables, i.e., state sequences and model parameters. To solve this problem, the variational Bayesian (VB) method has been proposed as an effective approximation method of the Bayesian approach [2], and it shows a good performance in HMM-based speech recognition [4].

5.1 Speech recognition based on variational Bayesian method

5.1.1 Bayesian approach

Let $\mathbf{O} = (\mathbf{o}_1, \mathbf{o}_2, \dots, \mathbf{o}_T)$ be a set of training data of D dimensional feature vectors, and T is used to denote the frame number. The likelihood function of an HMM is represented by:

$$P(\mathbf{O}, \mathbf{Z} \mid \Lambda) = \prod_{t=1}^T a_{z_{t-1}z_t} \mathcal{N}(\mathbf{o}_t \mid \boldsymbol{\mu}_{z_t}, \mathbf{S}_{z_t}^{-1}), \quad (5.1)$$

where $\mathbf{Z} = (z_1, z_2, \dots, z_T)$ is a sequence of HMM states, $z_t \in \{1, \dots, N\}$ denotes a state at frame t and N is the number of states in an HMM. A set of model parameters

$\Lambda = \{a_{ij}, \boldsymbol{\mu}_i, \mathbf{S}_i\}_{i,j=1}^N$ consists of the state transition probability a_{ij} from state i to state j , the mean vector $\boldsymbol{\mu}_i$ and the covariance matrix \mathbf{S}_i^{-1} of a Gaussian distribution $\mathcal{N}(\cdot | \boldsymbol{\mu}_i, \mathbf{S}_i^{-1})$.

The Bayesian approach assumes that a set of model parameters Λ is random variables, while the ML approach estimates constant model parameters. The posterior distribution for a set of model parameters Λ is obtained with the famous Bayes theorem as follows:

$$P(\Lambda | \mathbf{O}) = \frac{P(\mathbf{O} | \Lambda)P(\Lambda)}{P(\mathbf{O})}, \quad (5.2)$$

where $P(\Lambda)$ is a prior distribution for Λ . Once the posterior distribution $P(\Lambda | \mathbf{O})$ is estimated, the predictive distribution for input data \mathbf{X} is represented by:

$$P(\mathbf{X} | \mathbf{O}) = \int P(\mathbf{X} | \Lambda)P(\Lambda | \mathbf{O})d\Lambda. \quad (5.3)$$

The model parameters are integrated out in Eq. (5.3), so that the effect of over-fitting is mitigated. However, it is difficult to solve the integral and expectation calculations. Especially, when a model includes latent variables, the calculation becomes more complicated. To overcome this problem, the variational Bayesian (VB) method has been proposed as a tractable approximation method of the Bayesian approach and it showed good performance in the HMM-based speech recognition [2], [4].

5.1.2 Variational Bayesian method

The variational Bayesian method maximizes a lower bound of log marginal likelihood \mathcal{F} instead of the true likelihood. A lower bound of log marginal likelihood is defined by using Jensen's inequality:

$$\begin{aligned} \mathcal{L}(\mathbf{O}) &= \log \sum_{\mathbf{Z}} \int P(\mathbf{O}, \mathbf{Z} | \Lambda)P(\Lambda) d\Lambda \\ &= \log \sum_{\mathbf{Z}} \int Q(\mathbf{Z})Q(\Lambda) \frac{P(\mathbf{O}, \mathbf{Z} | \Lambda)P(\Lambda)}{Q(\mathbf{Z})Q(\Lambda)} d\Lambda \\ &\geq \sum_{\mathbf{Z}} \int Q(\mathbf{Z})Q(\Lambda) \log \frac{P(\mathbf{O}, \mathbf{Z} | \Lambda)P(\Lambda)}{Q(\mathbf{Z})Q(\Lambda)} d\Lambda \\ &= \mathcal{F}. \end{aligned} \quad (5.4)$$

In the VB method, VB posterior distributions $Q(\Lambda)$ and $Q(\mathbf{Z})$ are introduced to approximate the true posterior distributions. The optimal VB posterior distributions can be ob-

tained by maximizing the objective function \mathcal{F} with the variational method as follows:

$$Q(\Lambda) = C_{\Lambda} P(\Lambda) \exp \left\{ \sum_{\mathbf{Z}} Q(\mathbf{Z}) \log P(\mathbf{O}, \mathbf{Z} | \Lambda) \right\}, \quad (5.5)$$

$$Q(\mathbf{Z}) = C_{\mathbf{Z}} \exp \left\{ \int Q(\Lambda) \log P(\mathbf{O}, \mathbf{Z} | \Lambda) d\Lambda \right\}, \quad (5.6)$$

where C_{Λ} and $C_{\mathbf{Z}}$ are the normalization terms of $Q(\Lambda)$ and $Q(\mathbf{Z})$, respectively. Since equations (5.5) and (5.6) are depend on each other, these updates should be iterated as the EM algorithm, which increases the value of objective function \mathcal{F} at each iteration until convergence.

5.1.3 Bayesian context clustering using cross validation

In the Bayesian approach, prior distributions are usually determined heuristically. However, hyper-parameters (parameters of prior distributions) affect the model selection as tuning parameters. Therefore, to automatically select an appropriate model structure, a determination technique of prior distribution is required. One possible approach is to optimize the hyper-parameters using training data so as to maximize the marginal likelihood. However, it still needs tuning parameters which control influences of prior distributions, and often leads to the over-fitting problem as the ML criterion. To overcome this problem, the prior distribution determination technique using cross validation has been proposed [48]. The cross validation is known as a straightforward and useful method for model structure optimization. By using cross valid prior distributions, an appropriate model structure can be selected in the Bayesian context clustering without tuning parameters. I apply the prior determination technique based on K -fold cross validation as a baseline system of the Bayesian approach.

5.2 DAEM algorithm for variational Bayes method

In the VB method, the free energy function for Bayesian approach can be rewritten as follows:

$$\mathcal{F}_{\beta}(\Lambda) = -\frac{1}{\beta} \log \sum_{\mathbf{Z}} \int P^{\beta}(\mathbf{O}, \mathbf{Z} | \Lambda) P^{\beta}(\Lambda) d\Lambda. \quad (5.7)$$

An upper bound of log marginal likelihood $\bar{\mathcal{F}}_\beta(\Lambda)$ is defined by using Jensen's inequality:

$$\begin{aligned}\mathcal{F}_\beta(\Lambda) &= -\frac{1}{\beta} \log \sum_{\mathbf{Z}} \int \hat{Q}(\mathbf{Z}) \hat{Q}(\Lambda) \frac{P^\beta(\mathbf{O}, \mathbf{Z} | \Lambda) P^\beta(\Lambda)}{\hat{Q}(\mathbf{Z}) \hat{Q}(\Lambda)} d\Lambda \\ &\leq -\frac{1}{\beta} \sum_{\mathbf{Z}} \int \hat{Q}(\mathbf{Z}) \hat{Q}(\Lambda) \log \frac{P^\beta(\mathbf{O}, \mathbf{Z} | \Lambda) P^\beta(\Lambda)}{\hat{Q}(\mathbf{Z}) \hat{Q}(\Lambda)} d\Lambda \\ &= \bar{\mathcal{F}}_\beta(\Lambda)\end{aligned}\tag{5.8}$$

The optimal VB posterior distributions can be obtained by minimizing the objective function $\bar{\mathcal{F}}_\beta(\Lambda)$ with the variational method as follows:

$$\hat{Q}(\Lambda) = C_\Lambda P^\beta(\Lambda) \exp \left\{ \sum_{\mathbf{Z}} \hat{Q}(\mathbf{Z}) \log P^\beta(\mathbf{O}, \mathbf{Z} | \Lambda) \right\}, \tag{5.9}$$

$$\hat{Q}(\mathbf{Z}) = C_{\mathbf{Z}} \exp \left\{ \int \hat{Q}(\Lambda) \log P^\beta(\mathbf{O}, \mathbf{Z} | \Lambda) d\Lambda \right\}. \tag{5.10}$$

Since equations (5.9) and (5.10) are dependent each other, these updates should be iterated in the E-step of the DAEM algorithm. At the initial temperature $\beta^{(0)} \simeq 0$, the VB posterior distributions $Q(\Lambda)$ and $Q(\mathbf{Z})$ take a form nearly uniform distribution. While the temperature is decreasing, the form of $Q(\Lambda)$ and $Q(\mathbf{Z})$ change from uniform to each original posterior distribution. Finally the temperature $\beta = 1$, $Q(\Lambda)$ and $Q(\mathbf{Z})$ take each original posterior distribution and the reliable posterior distributions can be estimated.

5.3 Experiments

To evaluate the effectiveness of the proposed method, speaker independent continuous phoneme recognition experiments were conducted.

5.3.1 Experimental conditions

The experimental conditions are summarized in Table 5.1. The training data of about 20,000 Japanese sentences and testing data of 100 sentences were prepared from Japanese Newspaper Article Sentences (JNAS). Three-state left-to-right HMMs were used to model 43 Japanese phonemes, and 144 questions were prepared for the decision tree context clustering. Each state output probability distribution was modeled by a single Gaussian distribution with a diagonal covariance matrix.

In these experiments, the following five algorithms were compared.

Table 5.1: Experimental conditions

Training data	JNAS 20,000 utterances
Test data	JNAS 100 utterances
Sampling rate	16 kHz
Feature vector	12-order MFCC + Δ MFCC + Δ Energy
Window	Hamming
Frame size	25ms
Frame shift	10ms
Number of HMM state	3 (left-to-right)
Number of phoneme categories	43

- “ML” : Acoustic models trained by ML criterion and model structures selected by MDL criterion [46] and 50 EM-steps was conducted in the EM algorithm. HMMs were initialized by the segmental k -means algorithm.
- “CV-Bayes(f-EM50)” : Acoustic models trained by the Bayesian criterion and model structures selected by the Bayesian criterion using cross validation and 50 EM-steps was conducted in the EM algorithm. The posterior distributions were initialized by the flat start training.
- “CV-Bayes(EM5)” : Acoustic models trained by the Bayesian criterion and model structures selected by the Bayesian criterion using cross validation and 5 EM-steps was conducted in the EM algorithm. The posterior distributions were initialized by the k -means algorithm.
- “CV-Bayes(EM50)” : Acoustic models trained by the Bayesian criterion and model structures selected by the Bayesian criterion using cross validation and 50 EM-steps was conducted in the EM algorithm. The posterior distributions were initialized by the k -means algorithm.
- “CV-Bayes(DAEM)” : Acoustic models trained by the Bayesian criterion and model structures selected by the Bayesian criterion using cross validation and the DAEM algorithm was used for training algorithm.

The flat start training (“CV-Bayes(f-EM50)”) assumes that initial posterior distributions

of state sequences are uniform distribution. Once the posterior distributions of state sequences are given, the posterior distributions of model parameters can be estimated by the statistics of state sequences. In the initialization by the k -means algorithm, the posterior distribution of state sequences were initialized by the segmental k -means algorithm using phoneme boundary labels. In the Bayesian approaches, the posterior distribution of model parameters are also updated in the segmental k -means algorithm. Although the DAEM algorithm includes the initialization process, the DAEM algorithm (“CV-Bayes(DAEM)”) with $\beta = 0$ is equivalent to the initial values of the flat start training. This means that the DAEM algorithm uses no phoneme boundary labels in the initialization of posterior distributions. However, even though the flat start training updates the posterior distributions immediately at the first iteration based on unreliable initial parameters (this corresponds to the DAEM with $\beta = 0$ at the 1st iteration and $\beta = 1$ at the n d iteration), the DAEM algorithm gradually increase the temperature parameter β , and updates the posterior distributions slowly based on the annealing process.

The model structure based on MDL criterion has 5400 states and based on the Bayesian approach using cross validation has 16205 states. In “CV-Bayes” methods, the cross validation uses 10 folds. The temperature parameter β for the DAEM algorithm was updated by

$$\beta(i) = \frac{i}{I}, (i = 0, \dots, I) \quad (5.11)$$

where i denotes the iteration number. The number of temperature update steps was set to 10 ($I = 10$), and 5 EM-steps were conducted at each temperature, in total 50 EM-steps were conducted.

5.3.2 Experimental results

Figure 5.1 compares the lower bound of the log marginal likelihood \mathcal{F} for the training data, though the value of “ML” shows the log likelihood of the ML parameters (not marginal). Since the marginal likelihood is defined as the weighted sum of the likelihood function (equation (5.4)), the marginal likelihoods of the Bayesian approaches were lower than the likelihood of “ML.” The marginal likelihood of “CV-Bayes(f-EM50)” was the lowest among Bayesian methods. This is because of the local maxima problem caused by the inappropriate initial posterior distributions obtained without using phoneme boundary information. Although “CV-Bayes(DAEM)” also uses no phoneme boundaries, the marginal likelihood of “CV-Bayes(DAEM)” was improved than that of “CV-Bayes(f-EM50).” This result confirmed that the local maxima problem can be relaxed by the DAEM algorithm. Comparing “CV-Bayes(EM5)” with “CV-Bayes(EM50),”

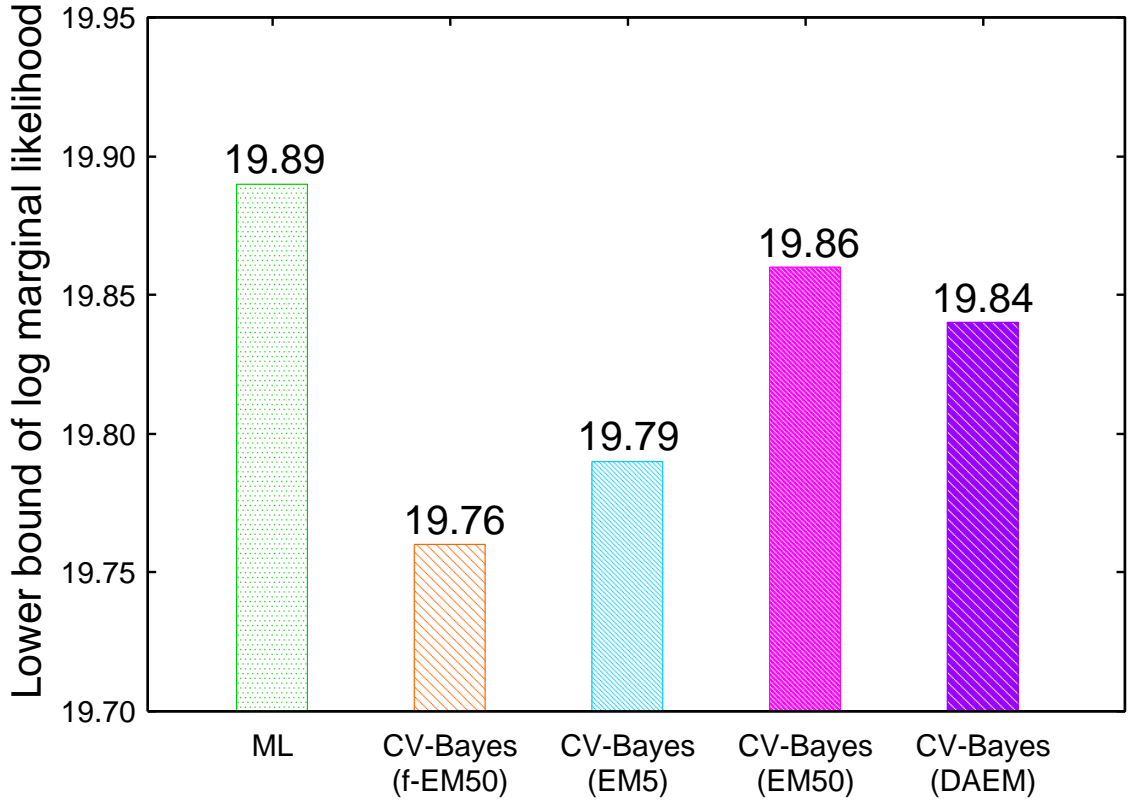


Figure 5.1: Log marginal likelihood

“CV-Bayes(EM50)” obtained the higher likelihood. This means that 5 EM-steps are not enough to converge the marginal likelihood. “CV-Bayes(DAEM)” also iterated the EM-steps 5 times at the last temperature ($\beta = 1$), and this may be the reason that the marginal likelihood of “CV-Bayes(DAEM)” was lower than that of “CV-Bayes(EM50).” However, the likelihood of “CV-Bayes(DAEM)” was higher than that of “CV-Bayes(EM5).” This means that the DAEM algorithm obtained reliable posterior distributions by using annealing process, even though no phoneme boundary information was used.

Figure 5.2 shows the phoneme accuracy of acoustic models. Contrary to the marginal likelihood, the Bayesian approaches outperformed “ML.” This result confirmed that the Bayesian approach is useful for HMM-based speech recognition. Comparing the Bayesian approaches, “CV-Bayes(f-EM50)” was the lowest recognition performance, because of the local maxima problem. Although “CV-Bayes(EM50)” achieved the highest likelihood, “CV-Bayes(EM50)” obtained no significant improvement as compared with “CV-Bayes(EM5)” in phoneme accuracy. Comparing the EM and DAEM algorithm, “CV-Bayes(DAEM)” achieved the higher phoneme accuracy than the EM algorithm using phoneme boundary information. This result indicated that the DAEM algorithm is effective to relax the serious local maxima problem in the VB speech recognition.

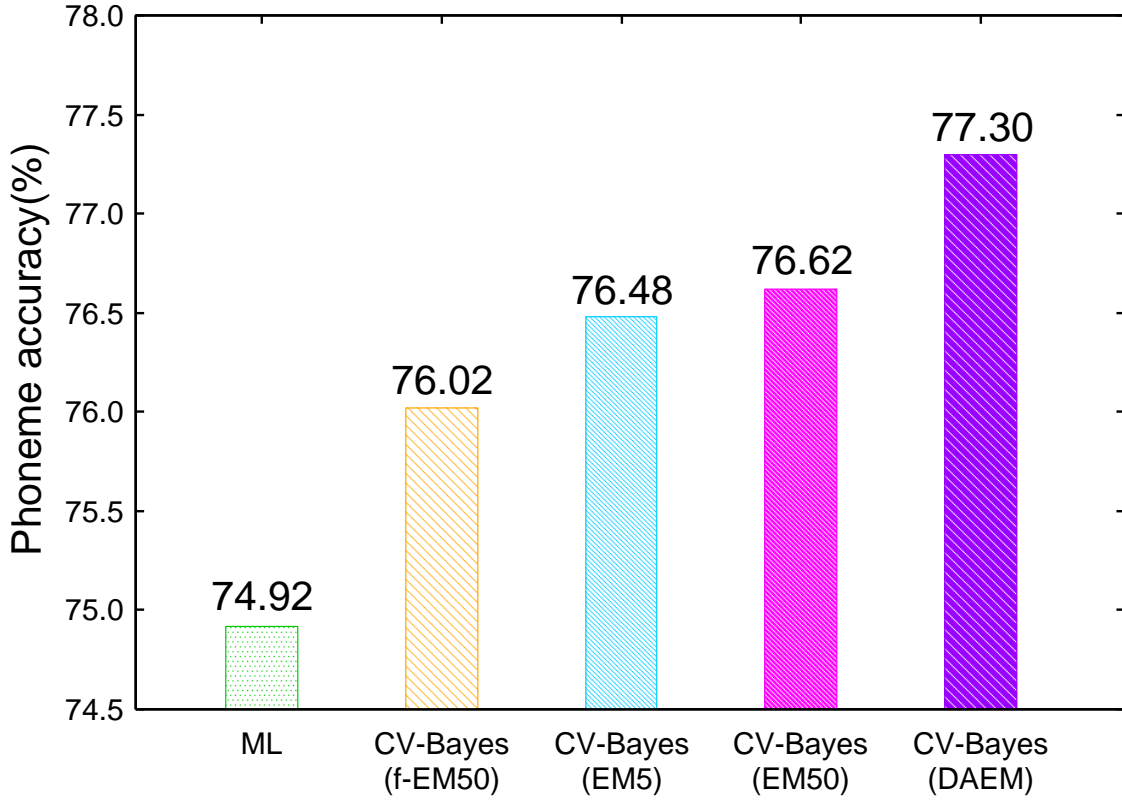


Figure 5.2: Phoneme accuracy

5.4 Summary

This chapter proposed a deterministic annealing based training algorithm for Bayesian speech recognition. The local maxima problem in the Bayesian method is more serious than in the ML-based approach, because the Bayesian method treats not only state sequences but also model parameters as latent variables. In this paper, the DAEM algorithm was applied to the Bayesian speech recognition to improve the recognition performance. The results of speech recognition experiments showed that the proposed method achieved higher performance than the conventional methods. As future work, I will apply this proposed framework to the simultaneous optimization of state sequences and model structures [49].

Chapter 6

Integration of multiple model structures based on Bayesian framework

Some approaches using multiple model structures have recently been proposed to increase model complexity (e.g., random forest [12], ROVER [13], and model structure annealing [49]). Although various integration techniques and criteria can be considered, this paper focuses on a model structure integration based on the Bayesian framework.

6.1 Bayesian speech recognition using multiple model structures

6.1.1 Marginalized likelihood function including multiple model structures

I define a marginal likelihood function treating model structures as latent variables to consider the framework using multiple model structures in Bayesian speech recognition.

$$\log P(\mathbf{O}) = \sum_m \sum_{\mathbf{Z}} \int P(\mathbf{O}, \mathbf{Z}, m, \Lambda_m) d\Lambda_m, \quad (6.1)$$

$$P(\mathbf{O}, \mathbf{Z}, m, \Lambda_m) = P(\mathbf{O}, \mathbf{Z} \mid m, \Lambda_m) P(\Lambda_m \mid m) P(m), \quad (6.2)$$

where $m \in \{1, \dots, M\}$ indexes the model structures, M is the number of the model structures, and $\Lambda_m \in \{\Lambda_1, \dots, \Lambda_M\}$ denotes a set of model parameters for the m -th model structure. Prior distribution $P(\Lambda_m \mid m)$ is prepared for each model structure m . Since state sequence \mathbf{Z} is not dependent on model structures in this framework, the state sequences are estimated from a combination of likelihoods calculated from multiple

model structures. Although the proposed model can be trained in the same manner as the variational Bayesian method, it has been confirmed [11] that even conventional Bayesian speech recognition using a single model structure suffers from the local maxima problem. Since the proposed method not only treats state sequences and model parameters but also model structures as latent variables, the local maxima problem is more serious than conventional Bayesian speech recognition. Deterministic annealing was adopted in the proposed framework to overcome this problem.

6.1.2 Training algorithm based on deterministic annealing

The problem of maximizing the log likelihood function is reformulated in the DAEM algorithm [10] as the problem of minimizing a free energy function. To adopt deterministic annealing for the proposed method, I redefine the free energy function based on the marginal likelihood function in Eq. (6.1) as:

$$\bar{\mathcal{F}}_\beta = -\frac{1}{\beta} \sum_m \sum_{\mathbf{Z}} \int \log P^\beta(\mathbf{O}, \mathbf{Z} \mid m, \Lambda_m) \times P^\beta(\Lambda_m \mid m) P^\beta(m) d\Lambda_m. \quad (6.3)$$

where β is called a temperature parameter. The upper bound of the free energy function is defined by using Jensen's inequality:

$$\bar{\mathcal{F}}_\beta \leq -\frac{1}{\beta} \sum_m \sum_{\mathbf{Z}} \int \tilde{Q}(\mathbf{Z}, m, \Lambda_m) \times \log \frac{P^\beta(\mathbf{O}, \mathbf{Z} \mid m, \Lambda_m) P^\beta(\Lambda_m \mid m) P^\beta(m)}{\tilde{Q}(\mathbf{Z}, m, \Lambda_m)} d\Lambda_m. \quad (6.4)$$

Since approximate distribution $\tilde{Q}(\mathbf{Z}, m, \Lambda_m)$ is a joint distribution of the three latent variables, calculating the upper bound becomes more complicated than that with the conventional VB method using only one model structure. To obtain the minimum upper bound, I assume the constraint:

$$\tilde{Q}(\mathbf{Z}, m, \Lambda_m) = \tilde{Q}(\mathbf{Z}) \tilde{Q}(m) \tilde{Q}(\Lambda_m \mid m). \quad (6.5)$$

Note that the dependence between model parameters and model structures remains as a prior distribution in Eq. (6.2). Under this constraint, optimal posterior distributions $\tilde{Q}(\mathbf{Z})$, $\tilde{Q}(m)$, and $\tilde{Q}(\Lambda_m \mid m)$ are obtained as:

$$\tilde{Q}(\mathbf{Z}) = C_{\mathbf{Z}} \exp \left\langle \left\langle \log P^\beta(\mathbf{O}, \mathbf{Z} \mid m, \Lambda_m) \right\rangle_{\tilde{Q}(\Lambda_m \mid m)} \right\rangle_{\tilde{Q}(m)}, \quad (6.6)$$

$$\begin{aligned}\tilde{Q}(m) = & C_m P^\beta(m) \exp \left\langle \left\langle \log P^\beta(\mathbf{O}, \mathbf{Z} \mid m, \Lambda_m) \right\rangle_{\tilde{Q}(\mathbf{Z})} \right. \\ & \left. + \log \frac{P^\beta(\Lambda_m \mid m)}{\tilde{Q}(\Lambda_m \mid m)} \right\rangle_{\tilde{Q}(\Lambda_m)},\end{aligned}\quad (6.7)$$

$$\begin{aligned}\tilde{Q}(\Lambda_m \mid m) = & C_{\Lambda_m} P^\beta(\Lambda_m \mid m) \times \\ & \exp \left\langle \log P^\beta(\mathbf{O}, \mathbf{Z} \mid m, \Lambda_m) \right\rangle_{\tilde{Q}(\mathbf{Z})}.\end{aligned}\quad (6.8)$$

where $C_{\mathbf{Z}}$, C_m and C_{Λ_m} correspond to the normalization terms of $\tilde{Q}(\mathbf{Z})$, $\tilde{Q}(m)$, and $\tilde{Q}(\Lambda_m \mid m)$ and $\langle \cdot \rangle_Q$ denotes the expectation with respect to Q . Since optimal variational posterior distributions $\tilde{Q}(\mathbf{Z})$, $\tilde{Q}(m)$, and $\tilde{Q}(\Lambda_m \mid m)$ depend on one another, from Eqs. (6.6), (6.7), and (6.8), these distributions should be iteratively updated. Temperature parameter β in the deterministic annealing process is gradually increased from 0 to 1, and the form of the VB posterior distributions changes being dependent on the temperature parameter. Variational posterior distributions $\tilde{Q}(\mathbf{Z})$, $\tilde{Q}(m)$, and $\tilde{Q}(\Lambda_m \mid m)$ take a form that has a nearly uniform distribution at the initial temperature ($\beta \simeq 0$). This means that all model structures are uniformly used for estimating the posterior distribution of the model parameters and the state sequences in the initial step. While the temperature parameter is increasing ($\beta \rightarrow 1$), the form of $\tilde{Q}(\mathbf{Z})$, $\tilde{Q}(m)$, and $\tilde{Q}(\Lambda_m \mid m)$ change to each original posterior distribution. The factorized posterior distributions at this stage gradually interact with one another while taking into account the reliability of their estimates, and this process leads to a good solution as a joint posterior distribution. The $\tilde{Q}(\mathbf{Z})$, $\tilde{Q}(m)$, and $\tilde{Q}(\Lambda_m \mid m)$ at the final temperature ($\beta = 1$) take each original posterior distribution. The posterior probability of model structures can be appropriately estimated through this process because the Bayesian criterion works as a model selection criterion, and reliable posterior distributions of model parameters can be estimated.

6.2 Related approach

6.2.1 Random Forest

The random forest (RF) [12] is one technique that uses multiple model structures. However, there are some differences between RF and the proposed method. One difference is how the model structures are constructed. The RF method changes the data set or question set used for constructing the model structures. Although the proposed approach can also use these methods, I used the Bayesian framework to construct adequate model structures. Another difference is how multiple model structures can be used. Several ways of

combining models have been tried in the RF method because there are no criteria for estimating combined weights. The proposed method can be used to automatically estimate the posterior probabilities of model structures based on the consistent Bayesian criterion.

6.2.2 Non-parametric Bayes

From another point of view, the proposed method has a similarity to the non-parametric Bayesian method [50] because both methods use multiple model structures with different complexities and are integrated based on the Bayesian framework. The main difference between them is that the non-parametric Bayesian method assumes processes to generate multiple model structures for each data sample. Although the proposed method simply prepared multiple model structures, it still has the effect of model structure marginalization and can be performed without increasing the complexity of the training process.

6.2.3 Discriminative approaches

In recent state-of-the-art speech recognition systems, discriminative approaches have been used [51] [52]. Contrary to this, the proposed method is based on a generative model of the observations as the conventional HMM based speech recognition. However, the most discriminative approaches use structures of generative statistical models, and finding the appropriate model structures is still essential problem of speech recognition. Therefore, the authors think that the idea of using multiple model structures and integration based on the consistent statistical criterion are useful and available for various approaches including discriminative approaches in future work.

6.3 Experiments

6.3.1 Speaker independent speech recognition (small training data)

Experimental Conditions

The experimental conditions are summarized in Tab. 6.1. The training data of 1,238 Japanese sentences (eight male speakers) and testing data of 50 sentences (thirteen male speakers) were prepared from Japanese Newspaper Article Sentences (JNAS) [47]. HMMs were used to 43 Japanese phoneme and 204 questions were prepared for the context clustering. Each state output probability distribution was modeled by a Gaussian distribution

Table 6.1: Experimental condition

Sampling rate	16 kHz
Feature vector	12-order MFCC + Δ + Δ^2
Frame size	25 ms
Frame shift	10 ms
Window	Hamming
Topology	3-state left-to-right

with a diagonal covariance matrix. In this experiment, the following five algorithms were compared.

- **Flat-start** : HMMs were initialized by flat-start training and trained by the EM algorithm (the EM-steps were iterated 50 times).
- **DAEM** : HMMs were initialized by flat-start training and trained by the DAEM algorithm.
- **Mtree** : HMMs were initialized by flat-start training and trained by the DAEM algorithm with multiple model structures.
- **Label5** : HMMs were initialized by the segmental k -means algorithm and trained by the EM algorithm (the EM-steps were iterated 5 times).
- **Label50** : HMMs were initialized by the segmental k -means algorithm and trained by the EM algorithm (the EM-steps were iterated 50 times).

ML and Bayes criteria can be applied to each of the above algorithms, and comparative methods were represented by the combination of algorithms and criteria. **Mtree(Bayes)** is the proposed method and **Mtree(ML)** is the previous proposed method using ML criterion reported in [5]. The DAEM methods using single decision tree **DAEM(ML)** and **DAEM(Bayes)** were also compared with the proposed method and their details were reported in [53] and [11], respectively. For using a single decision tree approaches (**Flat-start**, **DAEM**, **Label5** and **Label50**), the following tree structures were respectively used for ML and Bayes criteria:

- **ML** : a model structure is selected by the minimum description length (MDL) criterion [46]. This structure has 616 leaf nodes.

- Bayes : a model structure is selected by the Bayesian criterion using 10-folds cross validation [48]. This structure has 7,755 leaf nodes (CV-Bayes).

For **Mtree**, I additionally prepared a model structure which represents monophone. Monophone has 129 leaf nodes. In the DAEM algorithm, the number of updates of the temperature parameter was set to 10 ($I = 10$), and EM-steps were iterated 5 times at each temperature. The temperature parameter β was updated by

$$\beta(i) = \left(\frac{i}{I}\right)^n, \quad i = 0, \dots, I, \quad (6.9)$$

where i denotes the iteration number of temperature updates, and n was varied as $n = 2^\alpha$, ($\alpha = -3, \dots, 3$). Because the EM-steps in **DAEM** were totally iterated 50 times, the EM-steps in **Flat-start** and **Label50** were iterated 50 times. In **Mtree(ML)**, since it is difficult to estimate the accurate posterior probabilities of the model structures, I heuristically assumed that $Q_{ML}(m)$ was updated by the following linear functions:

$$Q_{ML}(\text{Monophone}) = 0.5\left(1 - \frac{i}{I}\right) \quad (6.10)$$

$$Q_{ML}(\text{MDL}) = 0.5\left(1 + \frac{i}{I}\right). \quad (6.11)$$

Figures 4.2 and 4.3 show plots of the schedules of the temperature parameter β and the update schedules of $Q_{ML}(m)$. Note that the proposed method does not require pre-determined posterior probabilities of the model structures such as Eqs. (6.10) and (6.11).

Experimental Results

Figure 6.1 compares the upper bound of the log marginal likelihood $\bar{\mathcal{F}}_\beta$ for the training data. The highest marginal likelihood of the methods using the DAEM algorithm (**DAEM** and **Mtree**) were obtained at $\alpha = 2$. This result shows that the marginal likelihood of **Flat-start** was lowest among the Bayesian methods. This is because in **Flat-start**, HMMs were initialized by inappropriate initial posterior distributions which were obtained using no phoneme boundaries. **DAEM** also used no phoneme boundaries, the marginal likelihood of **DAEM** was slightly improved from that of **Flat-start**. The EM-steps were iterated 50 times in **Flat-start**, thus it can be considered that the estimation of the model parameters are converged. On the other hand, in **DAEM**, the EM-steps were iterated five times after the temperature parameter β achieved the final temperature $\beta = 1$. Even though the iteration times might be not enough, **DAEM** can yield little improvement from **Flat-start**. **Mtree** obtain the highest marginal likelihood among the Bayesian methods. Furthermore, comparing **Mtree** with using phoneme boundaries method **Label**

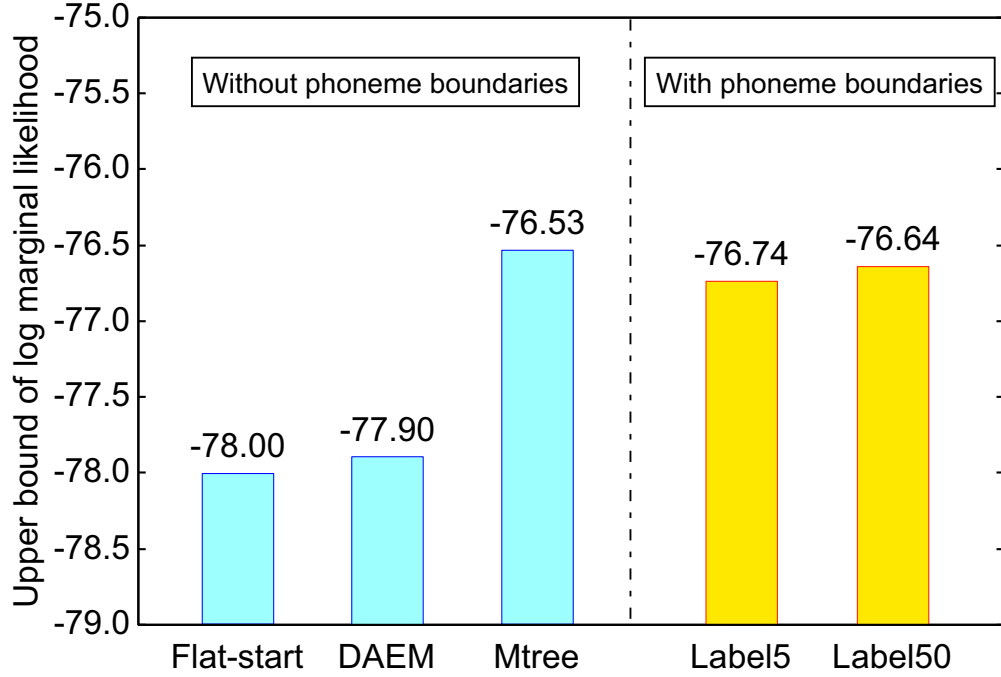


Figure 6.1: Upper bound of log marginal likelihood (the temperature parameter of **DAEM** and **Mtree** are set to $\alpha = 2$)

(**Label5** and **Label50**), even though HMMs were initialized using no phoneme boundaries in **Mtree**, the marginal likelihood of **Mtree** could achieve a slightly higher value than that of **Label** methods. This result shows that the method using multiple model structures can estimate more reliable model parameters than the conventional Bayesian methods.

Figure 6.2 shows the phoneme accuracy of each method. The accuracies of **DAEM** and **Mtree** were obtained when an appropriate temperature schedules were given. (**DAEM**(ML): $\alpha = 1$, **Mtree**(ML): $\alpha = 2$, **DAEM**(Bayes): $\alpha = 2$, **Mtree**(Bayes): $\alpha = 2$). Comparing the ML-based methods with the Bayesian methods, all Bayesian methods were obtained the higher accuracy than ML-based methods. This result indicates the effectiveness of the Bayesian approach for speech recognition. Similar to the marginal likelihood, **Flat-start** was yielded the lowest accuracy in the ML-based methods and the Bayesian methods, respectively. On the other hand, the accuracy of **DAEM** improved from that of **Flat-start**. Comparing the ML-based methods, **Mtree**(ML) was achieved the higher accuracy than **Flat-start**(ML) and **DAEM**(ML). This result confirms that using multiple model structures for the ML-based speech recognition can estimate reliable model parameters [49]. Moreover, **Mtree**(Bayes) yielded the highest accuracy among **Flat-start**, **DAEM**, and **Mtree**. This result indicates that using multiple model structures for the Bayesian speech recognition is effective for improving the speech recognition perfor-

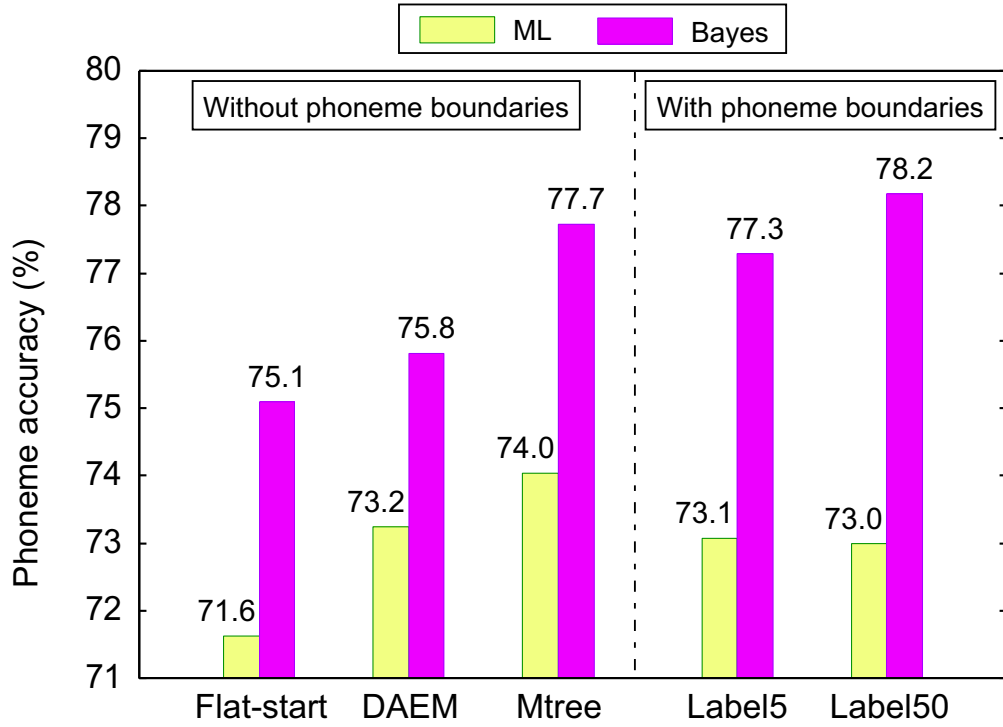


Figure 6.2: Phoneme accuracy

mance. Comparing **Mtree**, **Label5**, and **Label50**, although **Mtree(ML)** used no phoneme boundaries, **Mtree(ML)** obtained higher accuracy than **Label5(ML)** and **Label50(ML)** which used phoneme boundaries. While **Mtree(Bayes)** yielded the higher accuracy than **Label5(Bayes)**, **Mtree(Bayes)** could not achieve **Label50(Bayes)**. The accuracies of **Label5(Bayes)** and **Label50(Bayes)** show that Bayesian framework requires many iterations of the EM-steps than the ML-based framework. Actually, when the EM-steps were iterated 50 times at $\beta = 1$, the accuracy of **Mtree(Bayes)** is more closer to **Label50(Bayes)**. Thus, I expect that **Mtree(Bayes)** can obtain higher accuracy than **Label50(Bayes)** by adjusting the iteration number and the temperature update schedule.

In **Mtree(ML)**, the posterior probabilities of the model structures depend on the likelihood of each model structure. Due to this, even though the biggest model structure is not adequate, the biggest model structure is selected in the training process. Hence, in **Mtree(ML)**, the hueristics are required for selecting the adequate posterior probabilities of the model structures (Fig. 4.3). By contrast, **Mtree(Bayes)** can estimate the accurate posterior distributions of the model structures automatically. Figure 6.3 plots the posterior distributions of the model structures with the each temperature schedule in the training process. It can be seen that the forms of the plots was different between **Mtree(ML)** and **Mtree(Bayes)**, and the posterior probabilities of **Mtree(Bayes)** did not increase linearly.

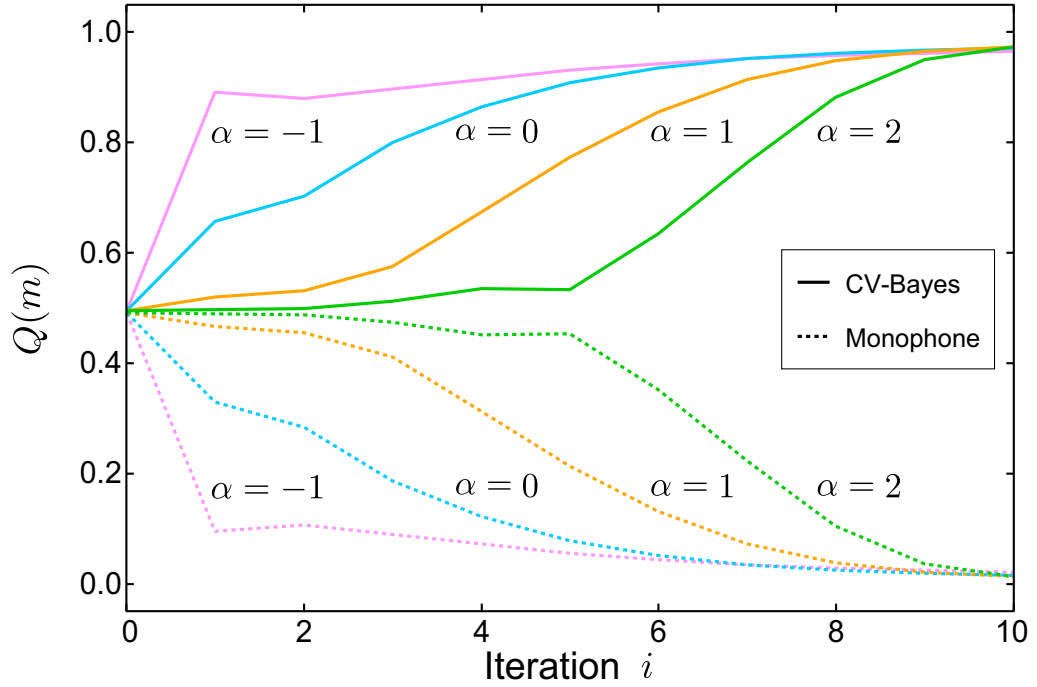


Figure 6.3: The posterior distributions of the model structures. Monophone has 129 leaf nodes and CV-Bayes has 7,755 leaf nodes.

Using these automatically estimated posterior distributions and the accurate temperature schedule, **Mtree(Bayes)** obtained a higher speech recognition performance. From these results, since the proposed method can treat multiple model structures without heuristics, I can investigate the effectiveness of more kinds of the model structures.

6.3.2 Speaker independent speech recognition (large training data)

Experimental condition

I conducted speaker independent experiments on continuous phoneme recognition to evaluate the effectiveness of the proposed method, where training data from 18,823 Japanese sentences and testing data from 100 sentences were prepared from Japanese Newspaper Article Sentences (JNAS). Speech signals were sampled at a frequency of 16 kHz and windowed at 10-ms frame rates using a 25-ms Hamming window. The spectrum parameter vectors consisted of 12-order MFCC and their delta and delta-delta coefficients. Three-state left-to-right HMMs were used to model triphones consisting of 43 Japanese phonemes and 204 questions were prepared for context clustering. All state output proba-

bility distributions were modeled by using a Gaussian distribution with a diagonal covariance matrix. The five algorithms below were compared in this experiment.

- **Flat-start** : HMMs were initialized by flat-start training and trained with the EM algorithm (the EM-steps were iterated 200 times).
- **DAEM** : HMMs were initialized by flat-start training and trained with the DAEM algorithm.
- **Mtree** : HMMs were initialized by flat-start training and trained with the DAEM algorithm with multiple model structures.
- **Label10** : HMMs were initialized with the segmental k -means algorithm using phoneme boundary labels and trained with the EM algorithm (the EM-steps were iterated 10 times).
- **Label200** : HMMs were initialized with the segmental k -means algorithm using phoneme boundary labels and trained with the EM algorithm (the EM-steps were iterated 200 times).

The ML and Bayes criteria could be applied to all five algorithms, and comparative methods were represented by combining the algorithms and criteria. **Mtree(Bayes)** is the new proposed method and **Mtree(ML)** is the previous method I proposed using the ML criterion reported in [49]. DAEM methods using a single model structure **DAEM(ML)** and **DAEM(Bayes)** were also compared with the proposed method and their details have been reported [53], [11]. Two tree structures were used for the approaches utilizing a single model structure (**Flat-start**, **DAEM**, **Label10**, and **Label200**).

- **ML** : a model structure was selected by using the minimum description length (MDL) criterion. This structure had 4,021 leaf nodes.
- **Bayes** : a model structure was selected by using the Bayesian criterion utilizing 200-folds cross validation [48]. This structure had 18,099 leaf nodes (CV-Bayes).

I also prepared a model structure representing monophone models for **Mtree(ML)** and **Mtree(Bayes)**. The monophone structure had 129 leaf nodes. The number of temperature parameter updates in the DAEM algorithm was set to 20 ($I = 20$), and EM-steps were iterated 10 times at each temperature. Temperature parameter β was updated by using $\beta(i) = (i/I)^n$, $i = 0, \dots, I$, where i denotes the number of iterations of temperature updates, and n was varied to $n = 2^\alpha$, ($\alpha = -3, \dots, 3$). Because the EM-steps in **DAEM** were iterated a total of 200 times, the EM-steps in **Flat-start** and **Label200** were iterated

200 times. Since it is difficult to estimate the accurate posterior probabilities of the model structures in **Mtree(ML)**, I heuristically assumed that $Q_{ML}(m)$ would be updated by the following linear functions: ($Q_{ML}(Monophone) = 0.5(1 - i/I)$, $Q_{ML}(MDL) = 0.5(1 + i/I)$). Note that **Mtree(Bayes)** does not require pre-determined posterior probabilities of the model structures.

Experimental results

Equation 6.4 summarized the upper bounds of the log marginal likelihood $\bar{\mathcal{F}}_\beta$ for the training data. The temperature update schedules were adjusted to obtain the highest marginal likelihood ($\alpha = 0$). The table indicates that the marginal likelihood of **Flat-start** was lowest for the Bayesian methods. This is because HMMs were initialized by inappropriate initial posterior distributions using no phoneme boundaries. Although **DAEM** also used no phoneme boundaries, the marginal likelihood of **DAEM** was improved from that of **Flat-start**. This indicates the DAEM algorithm effectively solved the local maxima problem. **Mtree** obtained the highest marginal likelihood of the Bayesian methods. Moreover, **Mtree** could achieve a higher marginal likelihood than the methods using label information (**Label10** and **Label200**). This demonstrates that the method using multiple model structures could estimate more reliable posterior distributions than the conventional Bayesian methods.

Figure 6.5 shows the phoneme accuracy for each method. The temperature schedules were adjusted to obtain the best phoneme accuracy (**DAEM(ML)**: $\alpha = 0$, **Mtree(ML)**: $\alpha = 1$, **DAEM(Bayes)**: $\alpha = 0$, **Mtree(Bayes)**: $\alpha = 0$). Comparing the ML-based methods with the Bayesian methods, all Bayesian methods were more accurate than those that were ML-based. This confirmed the effectiveness of the Bayesian approach for speech recognition. Similar to the comparison of marginal likelihoods, **Mtree** achieved the highest accuracy of methods using no phoneme boundaries (**Flat-start**, **DAEM** and **Mtree**) in both criteria. Moreover, the improvement for **Mtree** was higher than that for **DAEM** by comparing the improvements from the ML criterion to the Bayesian criterion between **DAEM** and **Mtree** methods. This means that consistently optimizing the model parameters and model structures based on the Bayesian criterion effectively improved recognition. While **Mtree(Bayes)** yielded higher accuracy than **Label10(Bayes)**, **Mtree(Bayes)** could not achieve the accuracy of **Label200(Bayes)**. Since **Label200** obtained higher accuracy than **Label10** in both criteria, **Mtree(Bayes)** might be able to obtain higher accuracy when I adjust the number of iterations or the schedule for temperature updates.

The posterior probabilities of the model structures in **Mtree(ML)** were in proportion to the likelihoods obtained by the ML estimates in all model structures. Since a larger model

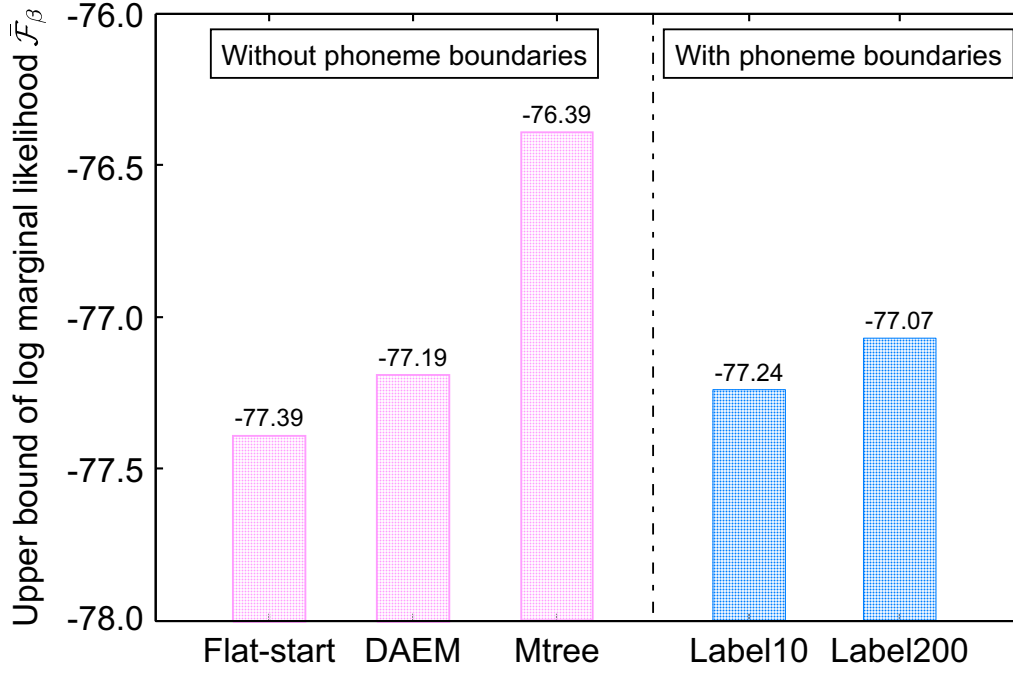


Figure 6.4: Upper bound of log marginal likelihood $\bar{\mathcal{F}}_\beta$

structure obtained a higher likelihood in the ML criterion, the largest model structure was always selected. However, this was inappropriate in most cases due to the over-fitting problem. A heuristic approach to control the posterior probabilities of model structures is required to avoid this problem. However, when the number of model structures increases, it is difficult to use such heuristics to obtain an appropriate posterior distribution. In contrast, **Mtree(Bayes)** could automatically estimate accurate posterior distributions of model structures. Figure 6.6 plots the posterior distribution of model structures with all temperature schedules during the training process. It can be seen that the posterior probability of the larger model structure (CV-Bayes) gradually increased begin dependent on the temperature parameter. to estimate the posterior distributions of the model parameters and state sequences in the early stages. Since the posterior distribution of the model structures was automatically estimated based on the Bayesian criterion, I could easily increase the number of model structures without heuristics, and I intend to investigate the effectiveness of using more than two model structures in future work.

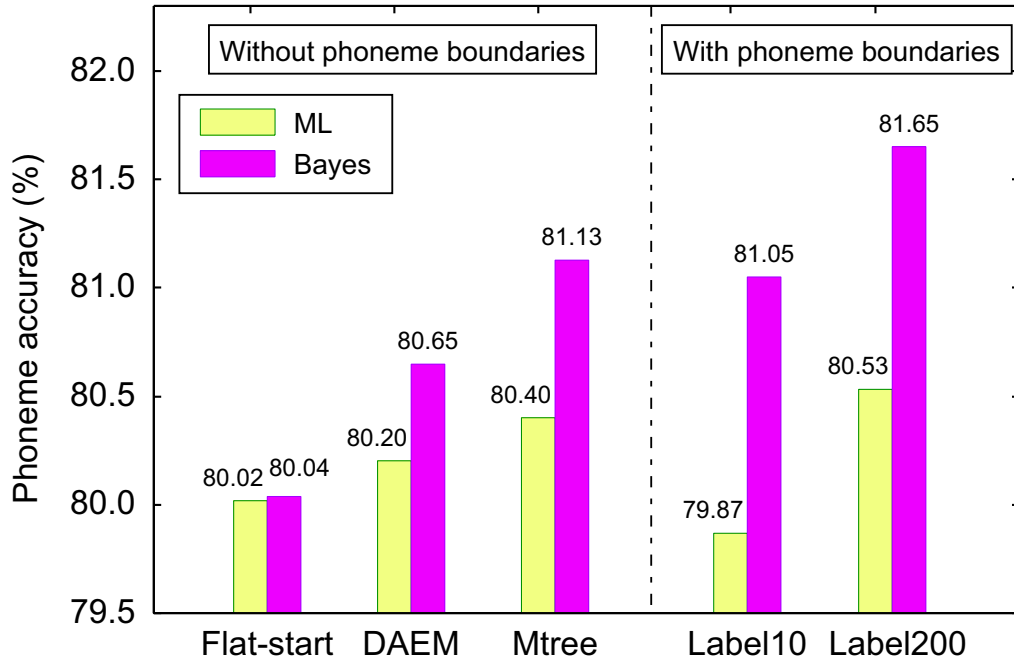


Figure 6.5: Phoneme accuracy

6.4 Summary

This chapter proposed integrating model structures based on the Bayesian framework for speech recognition. The proposed method not only treated state sequences and model parameters but also model structures as latent variables. Furthermore, deterministic annealing was applied to the proposed framework for relaxing the local maxima problem. The speech recognition experiment demonstrated the proposed method could automatically estimate reliable posterior distributions of model parameters and an adequate posterior distribution of model structures. I intend to investigate what effect increasing the number of model structures will have in future work and consider optimizing the training process.

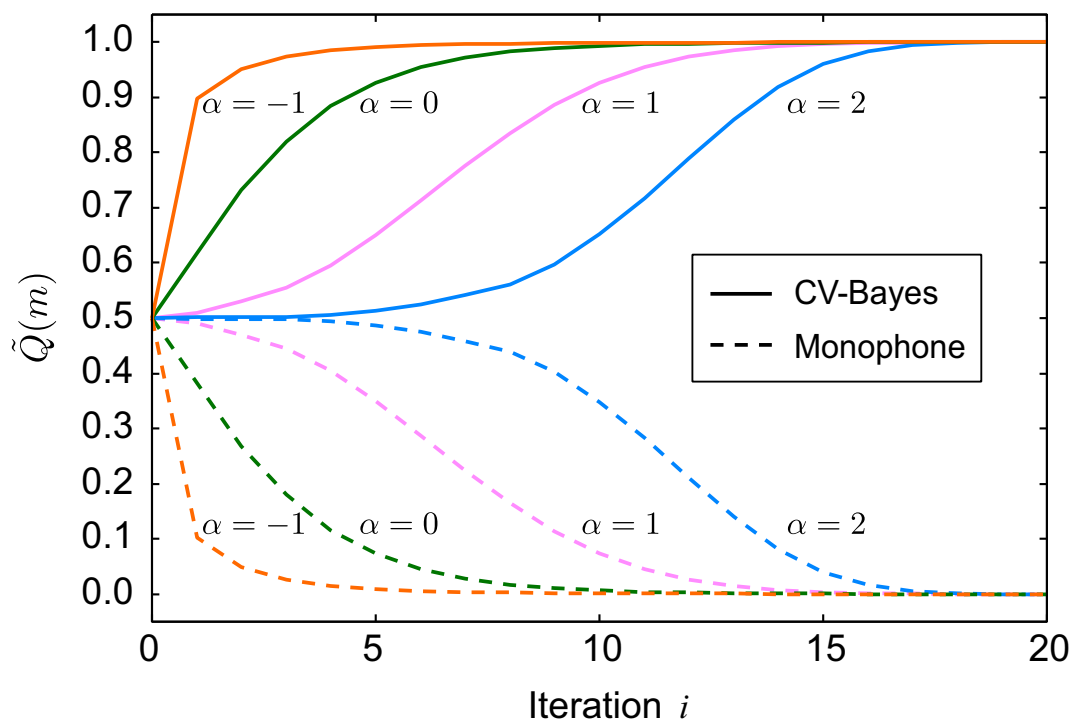


Figure 6.6: Posterior distributions of model structures.

Bibliography

- [1] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of IEEE*, Vol. 77, pp. 257–285, 1989.
- [2] H. Attias. Inferring parameters and structure of latent variable models by variational Bayes. *Proceedings of UAI 15*, 1999.
- [3] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, Vol. 37, No. 2, pp. 183–233, 2005.
- [4] S. Watanabe, Y. Minami, A. Nakamura, and N. Ueda. Variational Bayesian estimation and clustering for speech recognition. *IEEE Transactions on Speech and Audio Processing*, Vol. 12, No. 4, pp. 365–381, 2004.
- [5] T. Jitsuhiro and S. Nakamura. Automatic generation of non-uniform and context-dependent hmms based on the variational Bayesian approach. *IEICE Transactions on Information and Systems*, Vol. D88–D, pp. 391–400, 2005.
- [6] K. Yu and M.J.K. Gales. Bayesian adaptation and adaptively trained systems. *Proceedings of Automatic Speech Recognition and Understanding Workshop (ASRU) 2005*, pp. 209–214, 2005.
- [7] N. Ding and Z. Ou. Variational nonparametric bayesian hidden markov model. *Proceedings of ICASSP 2010*, pp. 2098–2101, 2005.
- [8] K. Katahira, K. Watanabe, and M. Okada. Deterministic annealing variant of variational bayes method. *Journal of Physics: Conference Series*, Vol. 95, No. 1, p. 012015, 2008.
- [9] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, Vol. 3, pp. 993–1022, 2003.
- [10] N. Ueda and R. Nakano. Deterministic annealing EM algorithm. *Neural Networks*, Vol. 11, pp. 271–282, 1998.

- [11] S. Shiota, K. Hashimoto, Y. Nankaku, A. Lee, and K. Tokuda. Deterministic annealing based training algorithm for bayesian speech recognition. *Proceedings of Interspeech 2009*, pp. 680–683, 2009.
- [12] J. Xue and Y. Zhao. Random forest of phonetic decision trees for acoustic modeling in conversational speech recognition. *IEEE Trans. on Audio, Speech, and Language Processing*, Vol. 16, No. 3, pp. 519–528, 2008.
- [13] J.G. Fiscus. A post-processing system to yield reduced word error rates: Recognizer output error reduction (rover). *Proceedings of IEEE workshop on Automatic Speech Recognition and Understanding*, pp. 347–352, 1997.
- [14] X.D. Huang, Y. Ariki, and M.A. Jack. *Hidden Markov models mor speech recognition*. Edinburgh University Press, 1990.
- [15] Rabiner L. and B.H. Juang. *Fundamentals of speech recognition*. Prentice-Hall, 1993.
- [16] S. Young, G. Evermann, M.J.F. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland. *The HTK Book (for HTK Version 3.3)*. Cambridge University, 2005.
- [17] L. Rabiner, B.H. Juang, S.E. Levinson, and M.M. Sondhi. Recognition of isolated digits using hidden Markov models with continuous mixture densities. *AT&T Technical Journal*, Vol. 64, No. 6, pp. 1211–1234, 1985.
- [18] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, Vol. 13, pp. 260–269, 1967.
- [19] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistics Society*, Vol. 39, pp. 1–38, 1977.
- [20] B.H. Juang. Maximum likelihood estimation for mixture of multivariate stochastic observations of Markov c hains. *AT&T Technical Journal*, Vol. 64, No. 6, pp. 1235–1249, 1985.
- [21] A. McCallum, D. Freitag, and F. Pereira. Maximum entropy Markov models for information extraction and segmentation. *Proceedings of International Conference Machine Learning*, pp. 591–598, 2000.
- [22] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proceedings of International Conference on Machine Learning*, pp. 282–289, 2001.

- [23] H.K.J. Kuo and Y. Gao. Maximum entropy direct models as a unified model for acoustic models in speech recognition. *Proceedings of International Conference on Spoken Language Processing*, pp. 681–684, 2004.
- [24] A. Gunawardana, L. Mahajan, A. Acero, and J.C. Platt. Hidden conditional random fields for phone classification. *Proceedings of European Conference on Speech Communication and Technology*, pp. 1117–1120, 2005.
- [25] J.R. Deller, J.H.L. Hansen, and J.G. Proakis. *Discrete-time processing of speech signals*. Macmillan, 1993.
- [26] F. Itakura and S. Saito. A statistical method for estimation of speech spectral density and formant frequencies. *Transactions of Institute of Electronics, Information and Communication Engineering*, Vol. J53-A, pp. 35–42, 1970.
- [27] J.D. Markel and A.H. Gray Jr. *Linear prediction of speech*. Springer-Verlag, 1976.
- [28] A.V. Oppenheim and R.W. Schaffer. *Digital signal processing*. Englewood Cliffs, 1975.
- [29] T. Fukada, K. Tokuda, Kobayashi T., and S. Imai. An adaptive algorithm for mel-cepstral analysis of speech. *Proceedings of International Conference on Acoustics, Speech, and Signal Processing'92*, Vol. 1, pp. 137–140, 1992.
- [30] S.B. Gavis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, & Signal Processing*, Vol. 28, pp. 357–366, 1980.
- [31] H. Hermansky. Perceptual linear prediction (PLP) of speech. *Journal of the Acoustic Society of America*, Vol. 87, No. 4, pp. 1738–1752, 1990.
- [32] S. Sagayama and F. Itakura. On individuality in a dynamic measure of speech. *Proceedings of Spring Conference of Acoustic Society of Japan*, pp. 589–590, 1979. (in Japanese).
- [33] S. Furui. Speaker independent isolated word recognition using dynamic features of speech spectrum. *IEEE Transactions Acoustics, Speech, & Signal Processing*, Vol. 34, pp. 52–59, 1986.
- [34] C.H. Lee and E. Giachin. Improved acoustic modeling for speaker independent large vocabulary continuous speech recognition. *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 161–164, 1991.

- [35] J.G. Wilpon, C.H. Lee, and L. Rabiner. Improvements in connected digit recognition using higher order spectral and energy features. *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 349–352, 1991.
- [36] S. Sagayama. Phoneme environment clustering for speech recognition. *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, pp. 397–400, 1989.
- [37] K.F. Lee, S. Hayamizu, H.W. Hon, C. Huang, J. Swartz, and R. Weide. Allophone clustering for continuous speech recognition. *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, pp. 749–752, 1990.
- [38] J.J. Odell. *The use of context in large vocabulary speech recognition*. PhD thesis, Cambridge University, 1995.
- [39] H.J. Nock, M.J.F. Gales, and S.J. Young. A comparative study of methods for phonetic decision-tree state clustering. *Proceedings of European Conference on Speech Communication and Technology*, Vol. 1, pp. 111–114, 1997.
- [40] S. Gao, J.S. Zhang, S. Nakamura, C.H. Lee, and T.S. Chu. Weighted graph based decision tree optimization for high accuracy acoustic modeling. *Proceedings of International Conference on Spoken Language Processing*, pp. 1233–1236, 2002.
- [41] S.M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech, & Signal Processing*, Vol. 35, No. 3, pp. 400–411, 1987.
- [42] H. Ney, D. Mergel, A. Noll, and A. Paelser. Data-driven search organisation for continuous speech recognition. *IEEE Transactions on Signal Processing*, Vol. 40, pp. 272–281, 1992.
- [43] F. Jelinek. Fast sequential decoding algorithm using a stack. *IBM Journal of Research and Development*, Vol. 13, pp. 675–685, 1969.
- [44] An introduction to variational methods for graphical models. *Machine Learning*.
- [45] Y. Sagisaka S. Katagiri H. Kawabara A. Kuramatsu, K. Takeda and K. Shikano. Atr japanese speech database as a tool of speech recognition and synthesis, 1990.
- [46] K. Shinoda and T. Watanabe. Acoustic modeling based on the mdl principle for speech recognition. *Proceedings of Eurospeech 1997*, Vol. 1, pp. 99–102, 1997.

- [47] K. Ito, M. Yamamoto, K. Takeda, T. Takezawa, T. Matsuoka, T. Kobayashi, K. Shikano, and S. Itahashi. Jnas: Japanese speech corpus for large vocabulary continuous speech recognition research. *Journal of the Acoustical Society of Japan (E)*, Vol. 20, No. 3, pp. 199–206, 1999.
- [48] K. Hashimoto, H. Zen, Y. Nankaku, A. Lee, and K. Tokuda. Bayesian context clustering using cross valid prior distribution for HMM-based speech recognition. *Proceedings of Interspeech 2008*, pp. 936–939, 2008.
- [49] S. Shiota, K. Hashimoto, H. Zen, Y. Nankaku, A. Lee, and K. Tokuda. Acoustic modeling based on model structure annealing for speech recognition. *Proceedings of Interspeech 2008*, pp. 932–935, 2008.
- [50] N. Ueda and T. Yamada. Nonparametric bayes. *Journal of Japanese Applied Mathematics*, Vol. 17, No. 3, pp. 196–214, 2007.
- [51] D. Povey and P. C. Woodland. Minimum phone error and i-smoothing for improved discriminative training. *Proceedings of ICASSP 2002*, Vol. 1, pp. 13–17, 2002.
- [52] E. McDermott, T.J. Hazen, J. Le Roux, A. Nakamura, and S. Katagiri. Discriminative training for large-vocabulary speech recognition using minimum classification error. *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 15, No. 1, pp. 203–223, 2007.
- [53] Y. Itaya, H. Zen, Y. Nankaku, C. Miyajima, K. Tokuda, and T. Kitamura. Deterministic annealing EM algorithm in parameter estimation for acoustic model. *IEICE Trans. Inf. & Syst.*, Vol. E88–D, No. 3, pp. 425–431, 2005.

List of Publications

Journal papers

- [1] **Sayaka Shiota**, Kei Hashimoto, Heiga Zen, Yoshihiko Nankaku, Akinobu Lee, and Keiichi Tokuda, “Speech recognition based on statistical models including multiple model structures,” *Acoustical Science and Technology*, vol. 32, no. 6, Nov. 2011.
- [2] **Sayaka Shiota**, Kei Hashimoto, Yoshihiko Nankaku, and Keiichi Tokuda, “A Bayesian framework using multiple model structures for speech recognition,” *IEICE Transactions on Information & Systems*, (Conditional acceptance).

International conference proceedings

- [3] **Sayaka Shiota**, Kei Hashimoto, Heiga Zen, Yoshihiko Nankaku, Akinobu Lee, and Keiichi Tokuda, “Acoustic modeling based on model structure annealing for speech recognition,” *Proceedings of Interspeech 2008*, pp. 932–935, Sep. 2008.
- [4] **Sayaka Shiota**, Kei Hashimoto, Yoshihiko Nankaku, and Keiichi Tokuda, “Deterministic annealing based training algorithm for Bayesian speech recognition,” *Proceedings of Interspeech 2009*, pp. 680–683, Sep. 2009.
- [5] Mikko Kurimo, William Byrne, John Dines, Philip N. Garner, Matthew Gibson, Yong Guan, Teemu Hirsimäki, Reima Karhila, Simon King, Hui Liang, Keiichiro Oura, Lakshmi Saheer, Matt Shannon, **Sayaka Shiota**, Jilei Tian, Keiichi Tokuda, Mirjam Wester, Yi-Jian Wu and Junichi Yamagishi, “Personalising speech-to-speech

translation in the EMIME project,” *Proceedings of ACL 2010*, pp. 48–53, Jul. 2010.

- [6] Mirjam Wester, John Dines, Matthew Gibson, Hui Liang, Yi-Jian Wu, Lakshmi Saheer, Simon King, Keiichiro Oura, Philip N. Garner, William Byrne, Yong Guan, Teemu Hirsimäki, Reima Karhila, Mikko Kurimo, Matt Shannon, **Sayaka Shiota**, Jilei Tian, Keiichi Tokuda, Junichi Yamagishi, “Speaker adaptation and the evaluation of speaker similarity in the EMIME speech-to-speech translation project,” *Proceedings of SSW7*, pp. 192–197, Sep. 2010.
- [7] Keiichiro Oura, Kei Hashimoto, **Sayaka Shiota**, and Keiichi Tokuda, “Overview of NIT HMM-based speech synthesis system for Blizzard Challenge 2010,” *Proceedings of Blizzard Challenge 2010*, Sep. 2010.

Technical reports

- [8] **Sayaka Shiota**, Kei Hashimoto, Heiga Zen, Yoshihiko Nankaku, Akinobu Lee, and Keiichi Tokuda, “Acoustic Modeling Based on Model Structure Annealing for Speech Recognition,” *Technical Report of IEICE*, vol. 107, no. 165, pp. 67–72, Jun. 2007.
- [9] **Sayaka Shiota**, Kei Hashimoto, Heiga Zen, Yoshihiko Nankaku, Akinobu Lee, and Keiichi Tokuda, “Speech recognition based on statistical models including multiple decision trees,” *Technical Report of IEICE*, vol. 108, no. 338, pp. 221–226, Dec. 2008.
- [10] **Sayaka Shiota**, Kei Hashimoto, Yoshihiko Nankaku, and Keiichi Tokuda, “Bayesian speech recognition based on model structure integration,” *Technical Report of IEICE*, Vol. 111, No. 97, pp.11–16, Jun. 2011.

Domestic conference proceedings

- [11] **Sayaka Shiota**, Kei Hashimoto, Heiga Zen, Yoshihiko Nankaku, Akinobu Lee, and Keiichi Tokuda, “Acoustic modeling based on model structure annealing for speech recognition,” *Proceedings of Autumn Meeting of the ASJ*, pp. 143–146, Sep. 2007.
- [12] **Sayaka Shiota**, Kei Hashimoto, Heiga Zen, Yoshihiko Nankaku, Akinobu Lee, and Keiichi Tokuda, “Speech recognition based on multiple phonetic decision tree structures,” *Proceedings of Autumn Meeting of the ASJ*, pp. 125–126, Sep. 2008.
- [13] **Sayaka Shiota**, Kei Hashimoto, Yoshihiko Nankaku, and Keiichi Tokuda, “Training algorithm based on deterministic annealing for Bayesian speech recognition,” *Proceedings of Autumn Meeting of the ASJ*, pp. 3–6, Sep. 2009.
- [14] **Sayaka Shiota**, Kei Hashimoto, Yoshihiko Nankaku, and Keiichi Tokuda, “Acoustic modeling based model structure annealing for Bayesian speech recognition,” *Proceedings of Autumn Meeting of the ASJ*, pp. 21–24, Mar. 2011.

Appendix A

Derivation of Bayesian framework using multiple model structures

A.1 Parameter estimation based Bayesian framework using multiple model structures (diagonal matrix)

The marginal likelihood is defined as:

$$P(\mathbf{O}) = \sum_m \sum_{\mathbf{Z}} \int \int P(\mathbf{O}, \mathbf{Z}, m, \Lambda^{(a)}, \Lambda_m^{(b)}) d\Lambda^{(a)} d\Lambda_m^{(b)} \quad (\text{A.1})$$

$$P(\mathbf{O}, \mathbf{Z}, m, \Lambda^{(a)}, \Lambda_m^{(b)}) = P(m)P(\Lambda_m^{(b)} | m)P(\Lambda^{(a)})P(\mathbf{Z} | \Lambda^{(a)})P(\mathbf{O} | m, \mathbf{Z}, \Lambda_m^{(b)}). \quad (\text{A.2})$$

The lower bound of the free energy function \mathcal{F} is derived by using the Jensen's inequality.

$$\begin{aligned} \mathcal{L}(\mathbf{O}) &= \log P(\mathbf{O}) \\ &= \log \sum_m \sum_{\mathbf{Z}} \int \int P(\mathbf{O}, \mathbf{Z}, m, \Lambda^{(a)}, \Lambda_m^{(b)}) d\Lambda^{(a)} d\Lambda_m^{(b)} \\ &\geq \sum_m \sum_{\mathbf{Z}} \int \int Q(m, \mathbf{Z}, \Lambda^{(a)}, \Lambda_m^{(b)}) \log \frac{P(\mathbf{O}, \mathbf{Z}, m, \Lambda^{(a)}, \Lambda_m^{(b)})}{Q(m, \mathbf{Z}, \Lambda^{(a)}, \Lambda_m^{(b)})} d\Lambda^{(a)} d\Lambda_m^{(b)} \\ &= \mathcal{F} \end{aligned} \quad (\text{A.3})$$

Because of the lower bound contains the joint probability, maximizing the lower bound is intractable. Thus, the following constraint is given:

$$Q(m, \mathbf{Z}, \Lambda^{(a)}, \Lambda_m^{(b)}) = \tilde{Q}(\mathbf{Z})\tilde{Q}(m)\tilde{Q}(\Lambda_m | m), \quad (\text{A.4})$$

where model parameter $\Lambda_m^{(b)}$ depends on model structure m .

$$\begin{aligned}\mathcal{F} = & \sum_m \sum_{\mathbf{Z}} \int \int \tilde{Q}(\mathbf{Z}) \tilde{Q}(m) \tilde{Q}(\Lambda_m | m) \log P(\mathbf{O}, \mathbf{Z}, m, \Lambda^{(a)}, \Lambda_m^{(b)}) d\Lambda^{(a)} d\Lambda_m^{(b)} \\ & - \sum_m Q(m) \log Q(m) - \sum_{\mathbf{Z}} Q(\mathbf{Z}) \log Q(\mathbf{Z}) - \int Q(\Lambda^{(a)}) \log Q(\Lambda^{(a)}) d\Lambda^{(a)} \\ & - \sum_m Q(m) \int Q(\Lambda_m^{(b)} | m) \log Q(\Lambda_m^{(b)} | m) d\Lambda_m^{(b)}.\end{aligned}\quad (\text{A.5})$$

By maximizing the lower bound, $Q(m)$, $Q(\mathbf{Z})$, $Q(\Lambda^{(a)})$, $Q(\Lambda_m^{(b)} | m)$ are obtained:

$$\begin{aligned}Q(m) = & C_m P(m) \exp \left\{ \sum_{\mathbf{Z}} \int Q(\mathbf{Z}) Q(\Lambda_m^{(b)} | m) \log P(\mathbf{O} | \mathbf{Z}, m, \Lambda_m^{(b)}) d\Lambda_m^{(b)} \right. \\ & \left. + \int Q(\Lambda_m^{(b)} | m) \log P(\Lambda_m^{(b)} | m) d\Lambda_m^{(b)} - \int Q(\Lambda_m^{(b)} | m) \log Q(\Lambda_m^{(b)} | m) d\Lambda_m^{(b)} \right\}\end{aligned}\quad (\text{A.6})$$

$$Q(\mathbf{Z}) = C_{\mathbf{Z}} \exp \left\{ \sum_m \int Q(m) Q(\Lambda_m^{(b)} | m) \log P(\mathbf{O} | \mathbf{Z}, m, \Lambda_m^{(b)}) d\Lambda_m^{(b)} \right. \quad (\text{A.7})$$

$$\left. + \int Q(\Lambda^{(a)}) \log P(\mathbf{Z} | \Lambda^{(a)}) d\Lambda^{(a)} \right\} \quad (\text{A.8})$$

$$Q(\Lambda^{(a)}) = C_{\Lambda^{(a)}} P(\Lambda^{(a)}) \exp \left\{ \sum_{\mathbf{Z}} Q(\mathbf{Z}) \log P(\mathbf{Z} | \Lambda^{(a)}) \right\}. \quad (\text{A.9})$$

$$Q(\Lambda_m^{(b)} | m) = C_{\Lambda_m^{(b)}} P(\Lambda_m^{(b)} | m) \exp \left\{ \sum_{\mathbf{Z}} Q(\mathbf{Z}) \log P(\mathbf{O} | \mathbf{Z}, m, \Lambda_m^{(b)}) \right\}, \quad (\text{A.10})$$

Note only $Q(\Lambda_m^{(b)} | m)$ depends on the model structure m . Then, the VB posterior distributions $Q(m)$, $Q(\mathbf{Z})$, $Q(\Lambda^{(a)})$, $Q(\Lambda_m^{(b)} | m)$ are adopted to the upper bound \mathcal{F} .

$$\begin{aligned}\mathcal{F} = & \sum_m \sum_{\mathbf{Z}} \int Q(m) Q(\mathbf{Z}) Q(\Lambda_m^{(b)} | m) \log P(\mathbf{O} | \mathbf{Z}, m, \Lambda_m^{(b)}) d\Lambda_m^{(b)} \\ & + \sum_m Q(m) \log P(m) + \sum_{\mathbf{Z}} \int Q(\mathbf{Z}) Q(\Lambda^{(a)}) \log P(\mathbf{Z} | \Lambda^{(a)}) d\Lambda^{(a)} \\ & + \int Q(\Lambda^{(a)}) \log P(\Lambda^{(a)}) d\Lambda^{(a)} + \sum_m \int Q(m) Q(\Lambda_m^{(b)} | m) \log P(\Lambda_m^{(b)} | m) d\Lambda_m^{(b)} \\ & - \sum_m Q(m) \log Q(m) - \sum_{\mathbf{Z}} Q(\mathbf{Z}) \log Q(\mathbf{Z}) \\ & - \int Q(\Lambda^{(a)}) \log Q(\Lambda^{(a)}) d\Lambda^{(a)} - \sum_m \int Q(m) Q(\Lambda_m^{(b)} | m) \log Q(\Lambda_m^{(b)} | m) d\Lambda_m^{(b)}\end{aligned}\quad (\text{A.11})$$

Moreover, each entropy of model structures, state sequences, and model parameters are obtained:

$$\begin{aligned}
\sum_m Q(m) \log Q(m) = & \\
& \log C_m + \sum_m Q(m) \log P(m) \\
& + \sum_m \sum_{\mathbf{Z}} \int Q(m) Q(\mathbf{Z}) Q(\Lambda_m^{(b)} | m) \log P(\mathbf{O} | \mathbf{Z}, m, \Lambda_m^{(b)}) d\Lambda_m^{(b)} \\
& + \sum_m \int Q(m) Q(\Lambda_m^{(b)} | m) \log P(\Lambda_m^{(b)} | m) d\Lambda_m^{(b)} \\
& - \sum_m \int Q(m) Q(\Lambda_m^{(b)} | m) \log Q(\Lambda_m^{(b)} | m) d\Lambda_m^{(b)} \tag{A.12}
\end{aligned}$$

$$\begin{aligned}
\sum_{\mathbf{Z}} Q(\mathbf{Z}) \log Q(\mathbf{Z}) = & \\
& \log C_{\mathbf{Z}} + \sum_m \sum_{\mathbf{Z}} \int Q'(m) Q(\mathbf{Z}) Q'(\Lambda_m^{(b)} | m) \log P(\mathbf{O} | \mathbf{Z}, m, \Lambda_m^{(b)}) d\Lambda_m^{(b)} \\
& + \sum_{\mathbf{Z}} \int Q(\mathbf{Z}) Q'(\Lambda^{(a)}) \log P(\mathbf{Z} | \Lambda^{(a)}) d\Lambda^{(a)} \tag{A.13}
\end{aligned}$$

$$\begin{aligned}
\int Q(\Lambda^{(a)}) \log Q(\Lambda^{(a)}) d\Lambda^{(a)} = & \log C_{\Lambda^{(a)}} + \int Q(\Lambda^{(a)}) \log P(\Lambda^{(a)}) d\Lambda^{(a)} \\
& + \sum_{\mathbf{Z}} \int Q(\mathbf{Z}) Q(\Lambda^{(a)}) \log P(\mathbf{Z} | \Lambda^{(a)}) d\Lambda^{(a)} \tag{A.14}
\end{aligned}$$

$$\begin{aligned}
\sum_m \int Q(m) Q(\Lambda_m^{(b)} | m) \log Q(\Lambda_m^{(b)} | m) d\Lambda_m^{(b)} = & \\
& \sum_m Q'(m) \log C_{\Lambda_m^{(b)}} + \sum_m \int Q'(m) Q(\Lambda_m^{(b)} | m) \log P(\Lambda_m^{(b)} | m) d\Lambda_m^{(b)} \\
& + \sum_m \sum_{\mathbf{Z}} \int Q'(m) Q(\mathbf{Z}) Q(\Lambda_m^{(b)} | m) \log P(\mathbf{O} | \mathbf{Z}, m, \Lambda_m^{(b)}) d\Lambda_m^{(b)} \tag{A.15}
\end{aligned}$$

Thus, the lower bound \mathcal{F} is rewritten as follows:

$$\begin{aligned}
\mathcal{F} = & \sum_m \sum_{\mathbf{Z}} \int Q(m) Q(\mathbf{Z}) Q(\Lambda_m^{(b)} | m) \log P(\mathbf{O} | \mathbf{Z}, m, \Lambda_m^{(b)}) d\Lambda_m^{(b)} \\
& + \sum_m Q(m) \log P(m) \\
& + \int Q(\Lambda^{(a)}) \log P(\Lambda^{(a)}) d\Lambda^{(a)} \\
& + \sum_{\mathbf{Z}} \int Q(\mathbf{Z}) Q(\Lambda^{(a)}) \log P(\mathbf{Z} | \Lambda^{(a)}) d\Lambda^{(a)} \\
& + \sum_m \int Q(m) Q(\Lambda_m^{(b)} | m) \log P(\Lambda_m^{(b)} | m) d\Lambda_m^{(b)} \\
& - \sum_m Q(m) \log P(m) \\
& - \sum_m \sum_{\mathbf{Z}} \int Q(m) Q(\mathbf{Z}) Q(\Lambda_m^{(b)} | m) \log P(\mathbf{O} | \mathbf{Z}, m, \Lambda_m^{(b)}) d\Lambda_m^{(b)} \\
& - \sum_m \int Q(m) Q(\Lambda_m^{(b)} | m) \log P(\Lambda_m^{(b)} | m) d\Lambda_m^{(b)} \\
& + \sum_m \int Q(m) Q(\Lambda_m^{(b)} | m) \log Q(\Lambda_m^{(b)} | m) d\Lambda_m^{(b)} \\
& - \sum_m \sum_{\mathbf{Z}} \int Q'(m) Q(\mathbf{Z}) Q'(\Lambda_m^{(b)} | m) \log P(\mathbf{O} | \mathbf{Z}, m, \Lambda_m^{(b)}) d\Lambda_m^{(b)} \\
& - \sum_{\mathbf{Z}} \int Q(\mathbf{Z}) Q'(\Lambda^{(a)}) \log P(\mathbf{Z} | \Lambda^{(a)}) d\Lambda^{(a)} \\
& - \int Q(\Lambda^{(a)}) \log P(\Lambda^{(a)}) d\Lambda^{(a)} \\
& - \sum_{\mathbf{Z}} \int Q(\mathbf{Z}) Q(\Lambda^{(a)}) \log P(\mathbf{Z} | \Lambda^{(a)}) d\Lambda^{(a)} \\
& - \sum_m \int Q'(m) Q(\Lambda_m^{(b)} | m) \log P(\Lambda_m^{(b)} | m) d\Lambda_m^{(b)} \\
& - \sum_m \sum_{\mathbf{Z}} \int Q'(m) Q(\mathbf{Z}) Q(\Lambda_m^{(b)} | m) \log P(\mathbf{O} | \mathbf{Z}, m, \Lambda_m^{(b)}) d\Lambda_m^{(b)} \\
& - \log C_m - \log C_{\mathbf{Z}} - \log C_{\Lambda^{(a)}} \\
& - \sum_m Q'(m) \log C_{\Lambda_m^{(b)}}
\end{aligned} \tag{A.16}$$

$$\begin{aligned}
\mathcal{F} = & -\log C_m - \log C_{\mathbf{Z}} - \log C_{\Lambda^{(a)}} - \sum_m Q'(m) \log C_{\Lambda_m^{(b)}} \\
& - \sum_m \sum_{\mathbf{Z}} \int Q'(m) Q(\mathbf{Z}) Q(\Lambda_m^{(b)} | m) \log P(\mathbf{O} | \mathbf{Z}, m, \Lambda_m^{(b)}) d\Lambda_m^{(b)} \\
& - \sum_m \sum_{\mathbf{Z}} \int Q'(m) Q(\mathbf{Z}) Q'(\Lambda_m^{(b)} | m) \log P(\mathbf{O} | \mathbf{Z}, m, \Lambda_m^{(b)}) d\Lambda_m^{(b)} \\
& - \sum_m \int Q'(m) Q(\Lambda_m^{(b)} | m) \log P(\Lambda_m^{(b)} | m) d\Lambda_m^{(b)} \\
& + \sum_m \int Q(m) Q(\Lambda_m^{(b)} | m) \log Q(\Lambda_m^{(b)} | m) d\Lambda_m^{(b)} \\
& - \sum_{\mathbf{Z}} \int Q(\mathbf{Z}) Q'(\Lambda^{(a)}) \log P(\mathbf{Z} | \Lambda^{(a)}) d\Lambda^{(a)}. \tag{A.17}
\end{aligned}$$

From Eq. (A.17), the normalization term C_m^{-1} is derived

$$\begin{aligned}
C_m^{-1} = & \sum_m \left[P(m) \exp \left\{ \sum_{\mathbf{Z}} \int Q(\mathbf{Z}) Q(\Lambda_m^{(b)} | m) \log P(\mathbf{O} | \mathbf{Z}, m, \Lambda_m^{(b)}) d\Lambda_m^{(b)} \right. \right. \\
& + \int Q(\Lambda_m^{(b)} | m) \log P(\Lambda_m^{(b)} | m) d\Lambda_m^{(b)} \\
& \left. \left. - \int Q(\Lambda_m^{(b)} | m) \log Q(\Lambda_m^{(b)} | m) d\Lambda_m^{(b)} \right\} \right]. \tag{A.18}
\end{aligned}$$

Note the prior of model strucrues $P(m)$ takes uniform distribution. The seventh term of Eq. (A.17) could calculate as below.

$$\begin{aligned}
& \sum_m \int Q'(m) Q(\Lambda_m^{(b)} | m) \log P(\Lambda_m^{(b)} | m) d\Lambda_m^{(b)} \\
& = \left\langle \log P(\Lambda_m^{(b)} | m) \right\rangle_{Q(\Lambda_m^{(b)} | m)} = \left\langle \log P(\boldsymbol{\mu}_{im}, \mathbf{S}_{im}) \right\rangle_{Q(\boldsymbol{\mu}_{im}, \mathbf{S}_{im})} \tag{A.19}
\end{aligned}$$

$$= \left\langle \log \mathcal{N}(\boldsymbol{\mu}_{im} | \boldsymbol{\nu}_{im}, (\xi_{im} \mathbf{S}_{im})^{-1}) \mathcal{G}(\mathbf{S}_{im} | \eta_{im}, \mathbf{B}_{im}) \right\rangle_{Q(\boldsymbol{\mu}_{im}, \mathbf{S}_{im})} \tag{A.20}$$

$$= \left\langle \log \mathcal{N}(\boldsymbol{\mu}_{im} | \boldsymbol{\nu}_{im}, (\xi_{im} \mathbf{S}_{im})^{-1}) \right\rangle_{Q(\boldsymbol{\mu}_{im}, \mathbf{S}_{im})} + \left\langle \log \mathcal{G}(\mathbf{S}_{im} | \eta_{im}, \mathbf{B}_{im}) \right\rangle_{Q(\boldsymbol{\mu}_{im}, \mathbf{S}_{im})} \tag{A.21}$$

$$\begin{aligned}
& \left\langle \log \mathcal{N}(\boldsymbol{\mu}_{im} \mid \boldsymbol{\nu}_{im}, (\xi_{im} \mathbf{S}_{im})^{-1}) \right\rangle_{Q(\boldsymbol{\mu}_{im}, \mathbf{S}_{im})} \\
&= -\frac{d}{2} \log(2\pi) + \frac{1}{2} \left\langle \log |\xi_{im} \mathbf{S}_{im}| \right\rangle_{Q(\boldsymbol{\mu}_{im}, \mathbf{S}_{im})} \\
&\quad - \frac{1}{2} \text{Tr} \left\{ \left\langle \xi_{im} \mathbf{S}_{im} (\boldsymbol{\mu}_{im} - \boldsymbol{\nu}_{im}) (\boldsymbol{\mu}_{im} - \boldsymbol{\nu}_{im})^T \right\rangle_{Q(\boldsymbol{\mu}_{im}, \mathbf{S}_{im})} \right\} \quad (\text{A.22})
\end{aligned}$$

$$\begin{aligned}
&= -\frac{d}{2} \log(2\pi) + \frac{1}{2} \int \mathcal{G}(\mathbf{S}_{im} \mid \bar{\boldsymbol{\nu}}_{im}, \bar{\mathbf{B}}_{im}) \log |\xi_{im} \mathbf{S}_{im}| d\mathbf{S}_{im} \\
&\quad - \frac{1}{2} \text{Tr} \left[\int \mathcal{G}(\mathbf{S}_{im} \mid \bar{\eta}_{im}, \bar{\mathbf{B}}_{im}) \xi_{im} \mathbf{S}_{im} \left\{ \int \mathcal{N}(\boldsymbol{\mu}_{im} \mid \bar{\boldsymbol{\nu}}_{im}, (\bar{\xi}_{im} \mathbf{S}_{im})^{-1}) \right. \right. \\
&\quad \left. \left. \times (\boldsymbol{\mu}_{im} - \boldsymbol{\nu}_{im}) (\boldsymbol{\mu}_{im} - \boldsymbol{\nu}_{im})^T d\boldsymbol{\mu}_{im} \right\} d\mathbf{S}_{im} \right] \quad (\text{A.23})
\end{aligned}$$

$$\begin{aligned}
& \int \mathcal{N}(\boldsymbol{\mu}_{im} \mid \bar{\boldsymbol{\nu}}_{im}, (\bar{\xi}_{im} \mathbf{S}_{im})^{-1}) (\boldsymbol{\mu}_{im} - \boldsymbol{\nu}_{im}) (\boldsymbol{\mu}_{im} - \boldsymbol{\nu}_{im})^T d\boldsymbol{\mu}_{im} \\
&= \bar{\boldsymbol{\nu}}_{im} \bar{\boldsymbol{\nu}}_{im}^T - \bar{\boldsymbol{\nu}}_{im} \boldsymbol{\nu}_{im}^T - \boldsymbol{\nu}_{im} \bar{\boldsymbol{\nu}}_{im}^T + \boldsymbol{\nu}_{im} \boldsymbol{\nu}_{im}^T + (\bar{\xi}_{im} \mathbf{S}_{im})^{-1} \quad (\text{A.24})
\end{aligned}$$

$$= (\bar{\boldsymbol{\nu}}_{im} - \boldsymbol{\nu}_{im}) (\bar{\boldsymbol{\nu}}_{im} - \boldsymbol{\nu}_{im})^T + (\bar{\xi}_{im} \mathbf{S}_{im})^{-1} \quad (\text{A.25})$$

$$\begin{aligned}
& \int \mathcal{G}(\mathbf{S}_{im} \mid \bar{\eta}_{im}, \bar{\mathbf{B}}_{im}) \xi_{im} \mathbf{S}_{im} \left\{ \int \mathcal{N}(\boldsymbol{\mu}_{im} \mid \bar{\boldsymbol{\nu}}_{im}, (\bar{\xi}_{im} \mathbf{S}_{im})^{-1}) (\boldsymbol{\mu}_{im} - \boldsymbol{\nu}_{im}) (\boldsymbol{\mu}_{im} - \boldsymbol{\nu}_{im})^T d\boldsymbol{\mu}_{im} \right\} d\mathbf{S}_{im} \\
&= \int \mathcal{G}(\mathbf{S}_{im} \mid \bar{\eta}_{im}, \bar{\mathbf{B}}_{im}) \xi_{im} \mathbf{S}_{im} \left\{ (\bar{\boldsymbol{\nu}}_{im} - \boldsymbol{\nu}_{im}) (\bar{\boldsymbol{\nu}}_{im} - \boldsymbol{\nu}_{im})^T + (\bar{\xi}_{im} \mathbf{S}_{im})^{-1} \right\} d\mathbf{S}_{im} \quad (\text{A.26})
\end{aligned}$$

$$= \int \mathcal{G}(\mathbf{S}_{im} \mid \bar{\eta}_{im}, \bar{\mathbf{B}}_{im}) \xi_{im} \mathbf{S}_{im} (\bar{\boldsymbol{\nu}}_{im} - \boldsymbol{\nu}_{im}) (\bar{\boldsymbol{\nu}}_{im} - \boldsymbol{\nu}_{im})^T d\mathbf{S}_{im} + \xi_{im} \bar{\xi}_{im}^{-1} I \quad (\text{A.27})$$

$$= \xi_{im} (\bar{\boldsymbol{\nu}}_{im} - \boldsymbol{\nu}_{im}) (\bar{\boldsymbol{\nu}}_{im} - \boldsymbol{\nu}_{im})^T \int \mathcal{G}(\mathbf{S}_{im} \mid \bar{\eta}_{im}, \bar{\mathbf{B}}_{im}) \mathbf{S}_{im} d\mathbf{S}_{im} + \xi_{im} \bar{\xi}_{im}^{-1} I \quad (\text{A.28})$$

$$= \bar{\eta}_{im} \bar{\mathbf{B}}_{im}^{-1} \xi_{im} (\bar{\boldsymbol{\nu}}_{im} - \boldsymbol{\nu}_{im}) (\bar{\boldsymbol{\nu}}_{im} - \boldsymbol{\nu}_{im})^T + \xi_{im} \bar{\xi}_{im}^{-1} I \quad (\text{A.29})$$

$$\left\langle \log |\xi_{im} \mathbf{S}_{im}| \right\rangle_{Q(\boldsymbol{\mu}_{im}, \mathbf{S}_{im})} = \int \mathcal{G}(\mathbf{S}_{im} \mid \bar{\eta}_{im}, \bar{\mathbf{B}}_{im}) \log |\xi_{im} \mathbf{S}_{im}| d\mathbf{S}_{im} \quad (\text{A.30})$$

$$= \int \mathcal{G}(\mathbf{S}_{im} \mid \bar{\eta}_{im}, \bar{\mathbf{B}}_{im}) \log |\mathbf{S}_{im}| d\mathbf{S}_{im} + \log |\xi_{im}^d| \quad (\text{A.31})$$

$$= \log |\xi_{im}^d| + \log 2^d - \log |\bar{\mathbf{B}}_{im}| + d\Psi\left(\frac{\bar{\eta}_{im}}{2}\right) \quad (\text{A.32})$$

By adopting Eqs. (A.29) and (A.32) to Eq. (A.22), each terms could represent as follows:

$$\left\langle \log \mathcal{N}(\boldsymbol{\mu}_{im} \mid \boldsymbol{\nu}_{im}, (\xi_{im} \mathbf{S}_{im})^{-1}) \right\rangle_{Q(\boldsymbol{\mu}_{im}, \mathbf{S}_{im})} \quad (\text{A.33})$$

$$\begin{aligned} &= -\frac{d}{2} \log(2\pi) + \frac{1}{2} \left\{ \log |\xi_{im}^d| + \log 2^d - \log |\bar{\mathbf{B}}_{im}| + d\Psi\left(\frac{\bar{\eta}_{im}}{2}\right) \right\} \\ &\quad - \frac{1}{2} \text{Tr} \left(\bar{\eta}_{im} \bar{\mathbf{B}}_{im}^{-1} \xi_{im} (\bar{\boldsymbol{\nu}}_{im} - \boldsymbol{\nu}_{im})(\bar{\boldsymbol{\nu}}_{im} - \boldsymbol{\nu}_{im})^T + \xi_{im} \bar{\xi}_{im}^{-1} I \right) \quad (\text{A.34}) \end{aligned}$$

$$\begin{aligned} &\left\langle \log \mathcal{G}(\mathbf{S}_{im} \mid \eta_{im}, \mathbf{B}_{im}) \right\rangle_{Q(\boldsymbol{\mu}_{im}, \mathbf{S}_{im})} \\ &= \int \mathcal{G}(\mathbf{S}_{im} \mid \bar{\eta}_{im}, \bar{\mathbf{B}}_{im}) \int \mathcal{N}(\boldsymbol{\mu}_{im} \mid \bar{\boldsymbol{\nu}}_{im}, (\xi_{im} \mathbf{S}_{im})^{-1}) \log \mathcal{G}(\mathbf{S}_{im} \mid \eta_{im}, \mathbf{B}_{im}) d\boldsymbol{\mu}_{im} d\mathbf{S}_{im} \quad (\text{A.35}) \end{aligned}$$

$$= \int \mathcal{G}(\mathbf{S}_{im} \mid \bar{\eta}_{im}, \bar{\mathbf{B}}_{im}) \log \mathcal{G}(\mathbf{S}_{im} \mid \eta_{im}, \mathbf{B}_{im}) d\mathbf{S}_{im} \quad (\text{A.36})$$

$$= \int \mathcal{G}(\mathbf{S}_{im} \mid \bar{\eta}_{im}, \bar{\mathbf{B}}_{im}) \log C_{\mathcal{G}} |\mathbf{S}_{im}|^{\frac{\eta_{im}}{2}-1} \exp \left\{ -\frac{1}{2} \text{Tr}(\mathbf{S}_{im} \mathbf{B}_{im}) \right\} d\mathbf{S}_{im} \quad (\text{A.37})$$

$$\begin{aligned} &= \log C_{\mathcal{G}} + \left(\frac{\eta_{im}}{2} - 1 \right) \left\{ \log 2^d - \log |\bar{\mathbf{B}}_{im}| + d\Psi\left(\frac{\bar{\eta}_{im}}{2}\right) \right\} \\ &\quad - \frac{1}{2} \text{Tr} \left(\mathbf{B}_{im} \int \mathcal{G}(\mathbf{S}_{im} \mid \bar{\eta}_{im}, \bar{\mathbf{B}}_{im}) \mathbf{S}_{im} d\mathbf{S}_{im} \right) \quad (\text{A.38}) \end{aligned}$$

$$= \log C_{\mathcal{G}} + \left(\frac{\eta_{im}}{2} - 1 \right) \left\{ \log 2^d - \log |\bar{\mathbf{B}}_{im}| + d\Psi\left(\frac{\bar{\eta}_{im}}{2}\right) \right\} - \frac{1}{2} \text{Tr} \left(\mathbf{B}_{im} \bar{\eta}_{im} \bar{\mathbf{B}}_{im}^{-1} \right) \quad (\text{A.39})$$

$$\begin{aligned}
& \left\langle \log \mathcal{N}(\boldsymbol{\mu}_{im} \mid \boldsymbol{\nu}_{im}, (\xi_{im} \mathbf{S}_{im})^{-1}) \right\rangle_{Q(\boldsymbol{\mu}_{im}, \mathbf{S}_{im})} + \left\langle \log \mathcal{G}(\mathbf{S}_{im} \mid \eta_{im}, \mathbf{B}_{im}) \right\rangle_{Q(\boldsymbol{\mu}_{im}, \mathbf{S}_{im})} \\
&= -\frac{d}{2} \log(2\pi) + \frac{1}{2} \left\{ \log |\xi_{im}^d| + \log 2^d - \log |\bar{\mathbf{B}}_{im}| + d\Psi\left(\frac{\bar{\eta}_{im}}{2}\right) \right\} \\
&\quad - \frac{1}{2} \text{Tr} \left(\bar{\eta}_{im} \bar{\mathbf{B}}_{im}^{-1} \xi_{im} (\bar{\boldsymbol{\nu}}_{im} - \boldsymbol{\nu}_{im})(\bar{\boldsymbol{\nu}}_{im} - \boldsymbol{\nu}_{im})^T + \xi_{im} \bar{\xi}_{im}^{-1} I \right) + \log C_{\mathcal{G}} \\
&\quad + \left(\frac{\eta_{im}}{2} - 1 \right) \left\{ \log 2^d - \log |\bar{\mathbf{B}}_{im}| + d\Psi\left(\frac{\bar{\eta}_{im}}{2}\right) \right\} - \frac{1}{2} \text{Tr} \left(\mathbf{B}_{im} \bar{\eta}_{im} \bar{\mathbf{B}}_{im}^{-1} \right)
\end{aligned} \tag{A.40}$$

$$\begin{aligned}
&= \frac{d}{2} \log \frac{|\xi_{im}|}{2\pi} + \log C_{\mathcal{G}} + \frac{\eta_{im} - 1}{2} \left\{ \log 2^d - \log |\bar{\mathbf{B}}_{im}| + d\Psi\left(\frac{\bar{\eta}_{im}}{2}\right) \right\} \\
&\quad - \frac{1}{2} \text{Tr} \left(\bar{\eta}_{im} \bar{\mathbf{B}}_{im}^{-1} \xi_{im} (\bar{\boldsymbol{\nu}}_{im} - \boldsymbol{\nu}_{im})(\bar{\boldsymbol{\nu}}_{im} - \boldsymbol{\nu}_{im})^T + \xi_{im} \bar{\xi}_{im}^{-1} I \right) - \frac{1}{2} \text{Tr} \left(\mathbf{B}_{im} \bar{\eta}_{im} \bar{\mathbf{B}}_{im}^{-1} \right)
\end{aligned} \tag{A.41}$$

$$\begin{aligned}
&= \frac{d}{2} \log |\xi_{im}| - \frac{d}{2} \log 2\pi + \frac{\eta_{im}}{2} \log |\mathbf{B}_{im}| - \frac{d\eta_{im}}{2} \log 2 - d \log \Gamma\left(\frac{\eta_{im}}{2}\right) \\
&\quad + \frac{d\eta_{im} - d}{2} \log 2 - \frac{\eta_{im} - 1}{2} \log |\bar{\mathbf{B}}_{im}| + \frac{d\eta_{im} - d}{2} \Psi\left(\frac{\bar{\eta}_{im}}{2}\right) \\
&\quad - \frac{1}{2} \text{Tr} \left(\bar{\eta}_{im} \bar{\mathbf{B}}_{im}^{-1} \xi_{im} (\bar{\boldsymbol{\nu}}_{im} - \boldsymbol{\nu}_{im})(\bar{\boldsymbol{\nu}}_{im} - \boldsymbol{\nu}_{im})^T + \xi_{im} \bar{\xi}_{im}^{-1} I \right) - \frac{1}{2} \text{Tr} \left(\mathbf{B}_{im} \bar{\eta}_{im} \bar{\mathbf{B}}_{im}^{-1} \right)
\end{aligned} \tag{A.42}$$

$$\begin{aligned}
&= \frac{d}{2} \log |\xi_{im}| - \frac{d}{2} \log 2\pi - \frac{d}{2} \log 2 + \frac{\eta_{im}}{2} \log |\mathbf{B}_{im}| \\
&\quad - \frac{\eta_{im} - 1}{2} \log |\bar{\mathbf{B}}_{im}| - d \log \Gamma\left(\frac{\eta_{im}}{2}\right) + \frac{d\eta_{im} - d}{2} \Psi\left(\frac{\bar{\eta}_{im}}{2}\right) \\
&\quad - \frac{1}{2} \text{Tr} \left(\bar{\eta}_{im} \bar{\mathbf{B}}_{im}^{-1} \xi_{im} (\bar{\boldsymbol{\nu}}_{im} - \boldsymbol{\nu}_{im})(\bar{\boldsymbol{\nu}}_{im} - \boldsymbol{\nu}_{im})^T + \xi_{im} \bar{\xi}_{im}^{-1} I \right) - \frac{1}{2} \text{Tr} \left(\mathbf{B}_{im} \bar{\eta}_{im} \bar{\mathbf{B}}_{im}^{-1} \right)
\end{aligned} \tag{A.43}$$

The eighth term of Eq. (A.17) could be represented:

$$\begin{aligned}
& \sum_m \int Q'(m) Q(\Lambda_m^{(b)} \mid m) \log Q(\Lambda_m^{(b)} \mid m) d\Lambda_m^{(b)} \\
&= \left\langle \log Q(\Lambda_m^{(b)} \mid m) \right\rangle_{Q(\Lambda_m^{(b)} \mid m)} = \left\langle \log Q(\boldsymbol{\mu}_{im}, \mathbf{S}_{im}) \right\rangle_{Q(\boldsymbol{\mu}_{im}, \mathbf{S}_{im})}
\end{aligned} \tag{A.44}$$

$$= \left\langle \log \mathcal{N}(\boldsymbol{\mu}_{im} \mid \bar{\boldsymbol{\nu}}_{im}, (\bar{\xi}_{im} \mathbf{S}_{im})^{-1}) \mathcal{G}(\mathbf{S}_{im} \mid \bar{\eta}_{im}, \bar{\mathbf{B}}_{im}) \right\rangle_{Q(\boldsymbol{\mu}_{im}, \mathbf{S}_{im})} \tag{A.45}$$

$$= \left\langle \log \mathcal{N}(\boldsymbol{\mu}_{im} \mid \bar{\boldsymbol{\nu}}_{im}, (\bar{\xi}_{im} \mathbf{S}_{im})^{-1}) \right\rangle_{Q(\boldsymbol{\mu}_{im}, \mathbf{S}_{im})} + \left\langle \log \mathcal{G}(\mathbf{S}_{im} \mid \bar{\eta}_{im}, \bar{\mathbf{B}}_{im}) \right\rangle_{Q(\boldsymbol{\mu}_{im}, \mathbf{S}_{im})} \tag{A.46}$$

$$\begin{aligned}
& \left\langle \log \mathcal{N}(\boldsymbol{\mu}_{im} \mid \bar{\boldsymbol{\nu}}_{im}, (\bar{\xi}_{im} \mathbf{S}_{im})^{-1}) \right\rangle_{Q(\boldsymbol{\mu}_{im}, \mathbf{S}_{im})} \\
&= -\frac{d}{2} \log(2\pi) + \frac{1}{2} \left\langle \log |\bar{\xi}_{im} \mathbf{S}_{im}| \right\rangle_{Q(\boldsymbol{\mu}_{im}, \mathbf{S}_{im})} \\
&\quad - \frac{1}{2} \text{Tr} \left\{ \left\langle \bar{\xi}_{im} \mathbf{S}_{im} (\boldsymbol{\mu}_{im} - \bar{\boldsymbol{\nu}}_{im}) (\boldsymbol{\mu}_{im} - \bar{\boldsymbol{\nu}}_{im})^T \right\rangle_{Q(\boldsymbol{\mu}_{im}, \mathbf{S}_{im})} \right\} \quad (\text{A.47})
\end{aligned}$$

$$\begin{aligned}
&= -\frac{d}{2} \log(2\pi) + \frac{1}{2} \int \mathcal{G}(\mathbf{S}_{im} \mid \bar{\boldsymbol{\nu}}_{im}, \bar{\mathbf{B}}_{im}) \log |\bar{\xi}_{im} \mathbf{S}_{im}| d\mathbf{S}_{im} \\
&\quad - \frac{1}{2} \text{Tr} \left[\int \mathcal{G}(\mathbf{S}_{im} \mid \bar{\boldsymbol{\eta}}_{im}, \bar{\mathbf{B}}_{im}) \bar{\xi}_{im} \mathbf{S}_{im} \left\{ \int \mathcal{N}(\boldsymbol{\mu}_{im} \mid \bar{\boldsymbol{\nu}}_{im}, (\bar{\xi}_{im} \mathbf{S}_{im})^{-1}) \right. \right. \\
&\quad \left. \left. \times (\boldsymbol{\mu}_{im} - \bar{\boldsymbol{\nu}}_{im}) (\boldsymbol{\mu}_{im} - \bar{\boldsymbol{\nu}}_{im})^T d\boldsymbol{\mu}_{im} \right\} d\mathbf{S}_{im} \right] \quad (\text{A.48})
\end{aligned}$$

$$\begin{aligned}
& \int \mathcal{N}(\boldsymbol{\mu}_{im} \mid \bar{\boldsymbol{\nu}}_{im}, (\bar{\xi}_{im} \mathbf{S}_{im})^{-1}) (\boldsymbol{\mu}_{im} - \bar{\boldsymbol{\nu}}_{im}) (\boldsymbol{\mu}_{im} - \bar{\boldsymbol{\nu}}_{im})^T d\boldsymbol{\mu}_{im} \\
&= \bar{\boldsymbol{\nu}}_{im} \bar{\boldsymbol{\nu}}_{im}^T - \bar{\boldsymbol{\nu}}_{im} \bar{\boldsymbol{\nu}}_{im}^T - \bar{\boldsymbol{\nu}}_{im} \bar{\boldsymbol{\nu}}_{im}^T + \bar{\boldsymbol{\nu}}_{im} \bar{\boldsymbol{\nu}}_{im}^T + (\bar{\xi}_{im} \mathbf{S}_{im})^{-1} \quad (\text{A.49})
\end{aligned}$$

$$= (\bar{\xi}_{im} \mathbf{S}_{im})^{-1} \quad (\text{A.50})$$

$$\begin{aligned}
& \int \mathcal{G}(\mathbf{S}_{im} \mid \bar{\boldsymbol{\eta}}_{im}, \bar{\mathbf{B}}_{im}) \bar{\xi}_{im} \mathbf{S}_{im} \left\{ \int \mathcal{N}(\boldsymbol{\mu}_{im} \mid \bar{\boldsymbol{\nu}}_{im}, (\bar{\xi}_{im} \mathbf{S}_{im})^{-1}) (\boldsymbol{\mu}_{im} - \bar{\boldsymbol{\nu}}_{im}) (\boldsymbol{\mu}_{im} - \bar{\boldsymbol{\nu}}_{im})^T d\boldsymbol{\mu}_{im} \right\} d\mathbf{S}_{im} \\
&= \int \mathcal{G}(\mathbf{S}_{im} \mid \bar{\boldsymbol{\eta}}_{im}, \bar{\mathbf{B}}_{im}) \bar{\xi}_{im} \mathbf{S}_{im} \left\{ (\bar{\boldsymbol{\nu}}_{im} - \bar{\boldsymbol{\nu}}_{im}) (\bar{\boldsymbol{\nu}}_{im} - \bar{\boldsymbol{\nu}}_{im})^T + (\bar{\xi}_{im} \mathbf{S}_{im})^{-1} \right\} d\mathbf{S}_{im} \quad (\text{A.51})
\end{aligned}$$

$$= \bar{\xi}_{im} \bar{\xi}_{im}^{-1} I = I \quad (\text{A.52})$$

$$\left\langle \log |\bar{\xi}_{im} \mathbf{S}_{im}| \right\rangle_{Q(\boldsymbol{\mu}_{im}, \mathbf{S}_{im})} = \int \mathcal{G}(\mathbf{S}_{im} \mid \bar{\boldsymbol{\eta}}_{im}, \bar{\mathbf{B}}_{im}) \log |\bar{\xi}_{im} \mathbf{S}_{im}| d\mathbf{S}_{im} \quad (\text{A.53})$$

$$= \int \mathcal{G}(\mathbf{S}_{im} \mid \bar{\boldsymbol{\eta}}_{im}, \bar{\mathbf{B}}_{im}) \log |\mathbf{S}_{im}| d\mathbf{S}_{im} + \log |\bar{\xi}_{im}^d| \quad (\text{A.54})$$

$$= \log |\bar{\xi}_{im}^d| + \log 2^d - \log |\bar{\mathbf{B}}_{im}| + d\Psi\left(\frac{\bar{\eta}_{im}}{2}\right) \quad (\text{A.55})$$

By adopting Eqs. (A.52) and (A.55) to Eq. (A.47), each terms could represent as follows:

$$\left\langle \log \mathcal{N}(\boldsymbol{\mu}_{im} \mid \bar{\boldsymbol{\nu}}_{im}, (\bar{\xi}_{im} \mathbf{S}_{im})^{-1}) \right\rangle_{Q(\boldsymbol{\mu}_{im}, \mathbf{S}_{im})} \quad (\text{A.56})$$

$$\begin{aligned}
&= -\frac{d}{2} \log(2\pi) + \frac{1}{2} \left\{ \log |\bar{\xi}_{im}^d| + \log 2^d - \log |\bar{\mathbf{B}}_{im}| + d\Psi\left(\frac{\bar{\eta}_{im} + 1 - j}{2}\right) \right\} - \frac{1}{2} \text{Tr} \left(\bar{\xi}_{im} \bar{\xi}_{im}^{-1} I \right) \quad (\text{A.57})
\end{aligned}$$

$$\begin{aligned}
& \left\langle \log \mathcal{G}(\mathbf{S}_{im} \mid \bar{\eta}_{im}, \bar{\mathbf{B}}_{im}) \right\rangle_{Q(\boldsymbol{\mu}_{im}, \mathbf{S}_{im})} \\
&= \int \mathcal{G}(\mathbf{S}_{im} \mid \bar{\eta}_{im}, \bar{\mathbf{B}}_{im}) \int \mathcal{N}(\boldsymbol{\mu}_{im} \mid \bar{\boldsymbol{\nu}}_{im}, (\bar{\xi}_{im} \mathbf{S}_{im})^{-1}) \log \mathcal{G}(\mathbf{S}_{im} \mid \bar{\eta}_{im}, \bar{\mathbf{B}}_{im}) d\boldsymbol{\mu}_{im} d\mathbf{S}_{im}
\end{aligned} \tag{A.58}$$

$$= \int \mathcal{G}(\mathbf{S}_{im} \mid \bar{\eta}_{im}, \bar{\mathbf{B}}_{im}) \log \mathcal{G}(\mathbf{S}_{im} \mid \bar{\eta}_{im}, \bar{\mathbf{B}}_{im}) d\mathbf{S}_{im} \tag{A.59}$$

$$= \int \mathcal{G}(\mathbf{S}_{im} \mid \bar{\eta}_{im}, \bar{\mathbf{B}}_{im}) \log \bar{C}_{\mathcal{G}} |\mathbf{S}_{im}|^{\frac{\bar{\eta}_{im}}{2}-1} \exp\left\{-\frac{1}{2} \text{Tr}(\mathbf{S}_{im} \bar{\mathbf{B}}_{im})\right\} d\mathbf{S}_{im} \tag{A.60}$$

$$\begin{aligned}
&= \log \bar{C}_{\mathcal{G}} + \left(\frac{\bar{\eta}_{im}}{2} - 1\right) \left\{ \log 2^d - \log |\bar{\mathbf{B}}_{im}| + d\Psi\left(\frac{\bar{\eta}_{im}}{2}\right) \right\} \\
&\quad - \frac{1}{2} \text{Tr}\left(\bar{\mathbf{B}}_{im} \int \mathcal{G}(\mathbf{S}_{im} \mid \bar{\eta}_{im}, \bar{\mathbf{B}}_{im}) \mathbf{S}_{im} d\mathbf{S}_{im}\right)
\end{aligned} \tag{A.61}$$

$$= \log \bar{C}_{\mathcal{G}} + \left(\frac{\bar{\eta}_{im}}{2} - 1\right) \left\{ \log 2^d - \log |\bar{\mathbf{B}}_{im}| + d\Psi\left(\frac{\bar{\eta}_{im}}{2}\right) \right\} - \frac{1}{2} \text{Tr}\left(\bar{\mathbf{B}}_{im} \bar{\eta}_{im} \bar{\mathbf{B}}_{im}^{-1}\right) \tag{A.62}$$

$$\begin{aligned}
& \left\langle \log \mathcal{N}(\boldsymbol{\mu}_{im} \mid \bar{\boldsymbol{\nu}}_{im}, (\bar{\xi}_{im} \mathbf{S}_{im})^{-1}) \right\rangle_{Q(\boldsymbol{\mu}_{im}, \mathbf{S}_{im})} + \left\langle \log \mathcal{G}(\mathbf{S}_{im} \mid \bar{\eta}_{im}, \bar{\mathbf{B}}_{im}) \right\rangle_{Q(\boldsymbol{\mu}_{im}, \mathbf{S}_{im})} \\
&= -\frac{d}{2} \log(2\pi) + \frac{1}{2} \left\{ \log |\bar{\xi}_{im}^d| + \log 2^d - \log |\bar{\mathbf{B}}_{im}| + d\Psi\left(\frac{\bar{\eta}_{im}}{2}\right) \right\} - \frac{1}{2} \text{Tr}(I) \\
&\quad + \log \bar{C}_{\mathcal{G}} + \left(\frac{\bar{\eta}_{im}}{2} - 1\right) \left\{ \log 2^d - \log |\bar{\mathbf{B}}_{im}| + d\Psi\left(\frac{\bar{\eta}_{im}}{2}\right) \right\} - \frac{1}{2} \text{Tr}\left(\bar{\mathbf{B}}_{im} \bar{\eta}_{im} \bar{\mathbf{B}}_{im}^{-1}\right)
\end{aligned} \tag{A.63}$$

$$\begin{aligned}
&= \frac{d}{2} \log \frac{|\bar{\xi}_{im}|}{2\pi} + \log \bar{C}_{\mathcal{G}} + \frac{\bar{\eta}_{im} - 1}{2} \left\{ \log 2^d - \log |\bar{\mathbf{B}}_{im}| + d\Psi\left(\frac{\bar{\eta}_{im}}{2}\right) \right\} \\
&\quad - \frac{1}{2} \text{Tr}(I) - \frac{1}{2} \text{Tr}\left(\bar{\mathbf{B}}_{im} \bar{\eta}_{im} \bar{\mathbf{B}}_{im}^{-1}\right)
\end{aligned} \tag{A.64}$$


$$\begin{aligned}
&= \frac{d}{2} \log |\bar{\xi}_{im}| - \frac{d}{2} \log 2\pi + \frac{\bar{\eta}_{im}}{2} \log |\bar{\mathbf{B}}_{im}| - \frac{d\bar{\eta}_{im}}{2} \log 2 - d \log \Gamma\left(\frac{\bar{\eta}_{im}}{2}\right) \\
&\quad + \frac{d\bar{\eta}_{im} - d}{2} \log 2 - \frac{\bar{\eta}_{im} - 1}{2} \log |\bar{\mathbf{B}}_{im}| + \frac{d\bar{\eta}_{im} - d}{2} \Psi\left(\frac{\bar{\eta}_{im}}{2}\right) \\
&\quad - \frac{1}{2} \text{Tr}(I) - \frac{1}{2} \text{Tr}\left(\bar{\mathbf{B}}_{im} \bar{\eta}_{im} \bar{\mathbf{B}}_{im}^{-1}\right)
\end{aligned} \tag{A.65}$$

$$\begin{aligned}
&= \frac{d}{2} \log |\bar{\xi}_{im}| - \frac{d}{2} \log 2\pi - \frac{d}{2} \log 2 + \frac{1}{2} \log |\bar{\mathbf{B}}_{im}| \\
&\quad - d \log \Gamma\left(\frac{\bar{\eta}_{im}}{2}\right) + \frac{d\bar{\eta}_{im} - d}{2} \Psi\left(\frac{\bar{\eta}_{im}}{2}\right) - \frac{1}{2} \text{Tr}(I) - \frac{1}{2} \text{Tr}\left(\bar{\eta}_{im} I\right)
\end{aligned} \tag{A.66}$$

$$\begin{aligned}
&= \frac{d}{2} \log |\bar{\xi}_{im}| - \frac{d}{2} \log 2\pi - \frac{d}{2} \log 2 + \frac{1}{2} \log |\bar{\mathbf{B}}_{im}| \\
&\quad - d \log \Gamma\left(\frac{\bar{\eta}_{im}}{2}\right) + \frac{d\bar{\eta}_{im} - d}{2} \Psi\left(\frac{\bar{\eta}_{im}}{2}\right) - \frac{d\bar{\eta}_{im} + d}{2}
\end{aligned} \tag{A.67}$$

Appendix B

Software



HMM-based Speech Synthesis System (HTS) - Home

[Front page] [Edit | Freeze | Diff | Backup | Upload | Reload] [New | List of pages | Search | Recent changes | Help]

Contents

- Home
- History
- Download
- License
- Acknowledgments
- Who we are
- Voice demos
- Publications
- Mailing list
- Bug reports
- Extensions
- Contact

Links

- HTK
- SPTK
- hts_engine API
- Festival
- Festvox
- DFKI MARY
- STRAIGHT
- Galatea
- Julius
- Blizzard Challenge
- ISCA SynSIG

recent(10)

2010-01-12

- Download

2010-01-06

- Extensions
- Acknowledgments
- History
- Home

2009-10-01

- Who we are

2009-09-15

- The first HTS meeting

2009-09-14

- Tutorial

2009-03-14

- Publications

2009-01-01

- Mailing List

Total: 31151

Welcome! ↑

The HMM-based Speech Synthesis System (HTS) has been being developed by the HTS working group and others (see [Who we are](#) and [Acknowledgments](#)). The training part of HTS has been implemented as a modified version of HTK and released as a form of patch code to HTK. The patch code is released under a free software license. However, it should be noted that **once you apply the patch to HTK, you must obey the license of HTK**. Related publications about the techniques and algorithms used in HTS can be found [here](#).

HTS version 2.1 includes hidden semi-Markov model (HSMM) training/adaptation/synthesis, speech parameter generation algorithm considering global variance (GV), SMAPLR/CSMAPLR adaptation, and other minor new features. Many bugs in HTS version 2.0.1 were also fixed. The API for runtime synthesis module, hts_engine API, version 1.0 was also released. Because hts_engine can run without the HTK library, users can develop their own open or proprietary softwares based on hts_engine. HTS and hts_engine API does not include any text analyzers but the [Festival Speech Synthesis System](#), [DFKI MARY Text-to-Speech System](#), or other text analyzers can be used with HTS. This distribution includes demo scripts for training speaker-dependent and speaker-adaptive systems using [CMU ARCTIC database](#) (English). Six HTS voices for Festival 1.96 are also released. They use the hts_engine module included in Festival. Each of HTS voices can be used without any other HTS tools.

For training Japanese voices, a demo script using the Nitech database is also prepared. Japanese voices trained by the demo script can be used on [GalateaTalk](#), which is a speech synthesis module of an open-source toolkit for anthropomorphic spoken dialogue agents developed in [Galatea project](#). An HTS voice for Galatea trained by the demo script is also released.

News! ↑

- **December 25, 2009**
HTS version 2.1.1 beta was released to the hts-users ML members.
- **August 27, 2009**
[The first HTS meeting in Interspeech 2009.](#)
- **May 22, 2009**
HTS-Demo for Brazilian Portuguese is released.
- **March 16, 2009**
Prof. Keiichi Tokuda & Dr. Heiga Zen have a [tutorial about HMM-based speech synthesis at Interspeech 2009](#)

Figure B.7: HTS: <http://hts.sp.nitech.ac.jp/>