DOCTORAL DISSERTATION

REFORMULATING HMM AS A TRAJECTORY MODEL BY IMPOSING EXPLICIT RELATIONSHIPS BETWEEN STATIC AND DYNAMIC FEATURES

DOCTOR OF ENGINEERING

JANUARY 2006

全 炳河

Heiga ZEN (in Japanese)

Byung Ha CHUN (in Korean)

Supervisor : Dr. Keiichi TOKUDA

Department of Computer Science and Engineering
Graduate School of Engineering
Nagoya Institute of Technology

Abstract

In recent years, the most popular acoustic model in automatic speech recognition (ASR) and text-to-speech synthesis (TTS) is a hidden Markov model (HMM), due to its ease of implementation and modeling flexibility. However, a number of limitations for modeling sequences of speech spectra using the HMM have been pointed out, such as i) piece-wise constant statistics within a state and ii) conditional independence assumption of state output probabilities. To overcome these shortcomings, a variety of alternative acoustic models have been proposed. Although these models can improve model accuracy and speech recognition performance, they generally require an increase in the number of model parameters. In contrast, dynamic features can also enhance performances of HMM-based speech recognizers and has been widely adopted. It can be viewed as a simple mechanism to capture time dependencies in the HMM. However, this approach is mathematically improper in the sense of statistical modeling. Generally, the dynamic features are calculated as regression coefficients from their neighboring static features. Therefore, relationships between the static and dynamic features are *deterministic*. However, these relationships are ignored and the static and dynamic features are modeled as independent statistical variables in the HMM framework. Ignoring these interdependencies allows inconsistency between the static and dynamic features when the HMM is used as a generative model in the obvious way.

In the present dissertation, a novel acoustic model, named a *trajectory HMM*, is described. This model is derived from the HMM whose state output vector includes both static and dynamic features. By imposing explicit relationships between the static and dynamic features, the HMM is naturally translated into a trajectory model. The above inconsistency and limitations of the HMM can be alleviated by the trajectory HMM. Furthermore, parameterization of the trajectory HMM is completely the same as that of the HMM with the same model topology. Therefore, any additional parameters are not required. In the present dissertation, model training algorithms based on a Viterbi approximation and a Markov chain Monte Carlo (MCMC) method and a search algorithm based on a delayed decision strategy are also derived. Results of continuous speech recognition and speech synthesis experiments show that the trajectory HMM can improve the performance both of speech recognizers and synthesizers.

Keywords: Speech recognition, speech synthesis, acoustic modeling, hidden Markov models, dynamic features, trajectory model, Viterbi approximation, Markov chain Monte Carlo

Abstract in Japanese

計算機の高速化，大規模音声コーパスの整備などに伴い，大語彙連続音声認識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)，特徴ベクトルに MFCC とその動的特徴量 (ΔMFCC，Δ^2MFCC) を用いる枠組みがある．HMM は，学習データに基づきパラメータを推定する実現容易なアルゴリズムが存在する，トポロジーを認識対象に応じて設計できる，現実的な計算量で学習・認識を行えるなどの特徴があり，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．

しかしながら，HMM を用いた音声スペクトル系列のモデル化に関する問題点が，これまで幾つか指摘されている．本論文で取り上げる問題点は以下の二つである．

1. HMM は，定常とみなせる短区間を表現する状態を連結することで，音声スペクトルの動的特性をモデル化する．このため，各状態内では出力確率分布が一定であり，状態内での時間的変化をモデル化できない．

2. 各時刻における観測ベクトルの出力確率は，その時刻に滞在する状態にのみ依存し，前後の時刻に滞在する状態には依存しない．これは，独立性の仮定 (Conditional Independence Assumption) と呼ばれる．

これらの問題を克服するため，これまでに様々な音響モデルが提案されており，性能の向上が報告されているが，HMM と比較してパラメータ数や計算量が増加する場合が多い．

一方，各時刻における音声特徴量だけでなく，その時間微分に対応する動的特徴量を利用することにより，認識性能が改善することがよく知られている．動的特徴量の利用は，HMM の枠組みの中で時間方向の依存性をモデル化するための一手法とみなすことができる．一般に動的特徴量は，前後数フレームの静的特徴量から回帰係数として計算される場合が多い．このため，静的特徴量・動的特徴量間には，線形変換で表現できる確定的な関係が存在するが，これまでの枠組みではこの関係を無視してモデル化を行っていた．

本論文では，静的及び動的特徴を含む特徴ベクトルを状態出力ベクトルとする HMM に対して，静的・動的特徴間の関係を明示的に導入し，HMM を再定式化する．その結果，HMM はトラジェクトリモデル (トラジェクトリ HMM) として定義できることを示す．トラジェクトリ HMM は，先にあげた通常の HMM の問題点を避けることができる．また，トラジェクトリ HMM のパラメータ構造は通常の HMM と同一であり，同じトポロジーであればそのパラメータ数は HMM と完全に等しい．

また，トラジェクトリ HMM の学習アルゴリズムとして，Viterbi 近似及びマルコフ連鎖モンテカルロ (MCMC) に基づく手法を導出した．特定話者連続音声認識実験においてトラジェクトリ HMM の性能を評価したところ，HMM と比較して最大約 50%の誤り削減率が得られた．また，トラジェクトリ HMM を HMM に基づく音声合成システムの音響モデルに適用したところ，合成音の自然性が大きく改善することが確認された．

以上のように本論文では，静的・動的特徴間の関係を HMM に明示的に導入することにより新たな統計モデルを導出し，音声認識・音声合成における音響モデルとして有用であることを示す．

Acknowledgement

First of all, I would like to express my sincere gratitude to Dr. Keiichi Tokuda, my advisor, for his support, encouragement, and guidance. I would also like to thank Drs. Tadashi Kitamura and Ryohei Nakano for reviewing this dissertation and their insightful comments.

I would like to thank Drs. Akinobu Lee, Yoshihiko Nankaku, Chiyomi Miyajima (currently with Nagoya University), Toru Takahashi (currently with Wakayama University), Takayoshi Yoshimura (currently with Toyota Central Research and Development Laboratories), and Shinji Sako (currently with Tokyo University) for their technical supports and helpful discussions. Special thanks go to all the members of Kitamura and Tokuda 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 Rie Matae, Junko Mori and Natsuki Kuromiya, secretaries of the laboratory, for their kind assistance.

I am grateful to Drs. Michael Picheny and Wael Hamza for giving me the opportunity to work in the IBM Thomas J. Watson Research Center and for their valuable advice. I would also like to thank Drs. Ellen Eide, Raimo Bakis, John Pitrelli, and Raul Fernandez for their helpful discussions.

Finally, I would sincerely like to thank my family for their encouragement.

Abbreviations

HMM	Hidden Markov model
ASR	Automatic speech recognition
TTS	Text-to-speech synthesis
CD	Continuous-density
ML	Maximum likelihood
EM	Expectation-maximization
MAP	Maximum a posteriori
MEMM	Maximum entropy Markov model
CRF	Conditional random fields
LPC	Linear prediction coefficients
LSP	Line spectral pair
PLP	Perceptual linear prediction
MFCC	Mel-filterbank cepstral coefficients
STC	Semi-tied covariance matrix
EMLLT	Extended maximum likelihood linear transform
SPAM	Subspaces for precision and mean
PoE	Product of experts
MoE	Mixture of experts
MMI	Maximum mutual information
MCE	Minimum classification error
MCMC	Markov chain Monte Carlo
MCEM	Monte Carlo EM
ATR	Advanced telecomunication research
KL divergence	Kullback-Leibler divergence
HTK	Hidden Markov model toolkit
CMU	Carnegie Mellon university
ESPS	Entropic signal processing system
MDL	Minimum discription length
CART	Classification and regression trees

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Since speech is obviously one of the most important ways for human to communicate, there have been a great number of efforts to incorporate speech into human-machine communication environments. As computers become more functional and prevalent, demands for technologies in speech processing area is increasing to establish high-quality human-machine communication with voice. These technologies will also be applicable to human-to-human communication.

Automatic speech recognition (ASR) and text-to-speech synthesis (TTS) are fundamental technologies for realizing speech-oriented interfaces. In recent years, these technologies are used in real-world applications, such as car navigation, information retrieval over the telephone, voice mail, or speech-to-speech translation system.

In the early years of speech technologies, rule-based systems were mainly used. However, in these days statistical approaches based on hidden Markov models (HMMs) have been dominant both in ASR [1] and TTS [2–5], due to their ease of implementation and modeling flexibility.

In this approach, the HMMs are used for modeling sequences of speech spectra. However, a number of limitations of the HMM have been pointed out [6–8]. In the present dissertation, following two limitations are considered:

- One of the underlying assumptions in the HMM is that observation vector sequences are quasi-stationary and each stationary part can be represented by a single state of the HMM. The statistics of each state does not vary dynamically. Therefore, intra-state time-dependency cannot be represented. Although this assumption might be satisfied in reading speech, it is not valid in spontaneous speech because they are dynamically affected by various factors such as speaking rate.

- The output probability of an observation vector depends only on the current state, neither on any other states nor observations. This is generally called as *state conditional independence assumption*.

To overcome these shortcomings of the HMM, a variety of alternative acoustic models, such as HMMs with polynomial regression functions [9–11], hidden dynamic models and its variants [12–19], frame-correlated HMMs [20–25], partly hidden Markov models [26], stochastic segment model [27], segmental HMMs [28–30], dynamical system models [31], switching linear dynamical systems [32, 33], buried Markov models [34], dynamic Bayesian networks [35], temporally varying means and precisions [36] etc., have been proposed. Most of them have attempted to capture explicit dynamics of speech parameter trajectories. Although the above models can improve model accuracy and speech recognition performance, generally an increase in the number of model parameters and computational complexity is required.

Alternatively, the use of dynamic features (e.g., delta and delta-delta cepstral coefficients) [37, 38] can also enhance the performance of HMM-based speech recognizers [39, 40]. It can be considered as a simple mechanism to capture time dependencies. However, it has been thought of as an ad hoc rather than an essential solution. Generally, the dynamic features are calculated as regression coefficients from their neighboring static features. Therefore, relationships between the static and dynamic features are *deterministic*. However, these relationships are ignored and the static and dynamic features are modeled as independent statistical variables. Ignoring these interdependencies allows inconsistency between the static and dynamic features when the HMM is used as a generative model in the obvious way.

In the present dissertation, a novel acoustic model is derived by reformulating the HMM whose state output vector includes both static and dynamic feature vectors. By imposing explicit relationships between static and dynamic features, the HMM is naturally translated into a trajectory model, referred to as *trajectory HMM* in the present dissertation. The trajectory HMM can overcome the above limitations of the HMM without any additional parameters. In the present dissertation training algorithms based on a Viterbi approximation and a Markov chain Monte Carlo (MCMC) method are derived. A new searching algorithm used both in training and recognition are also presented.

The formulation of the trajectory HMM is closely related to a technique for speech parameter generation from the HMM [41–43], in which the speech parameter sequence is determined so as to maximize its output probability for the HMM under the constraints between static and dynamic features. While the speech parameter generation algorithm was derived to construct HMM-based speech synthesizers [5], which can synthesize speech with various voice characteristics [44–46], the generation algorithm was also applied to speech recognition [47, 48].

The rest of the present dissertation is organized as follows. The next chapter introduces basic theories of the HMM. Chapters 3 and 4 describe statistical speech recognition and synthesis framework based on the HMM, respectively. Chapter 5 reformulates the HMM by imposing explicit relationship between static and dynamic features and defines the trajectory HMM. Relationships between the trajectory HMM and other techniques are also discussed in this chapter. Chapter 6 shows the training algorithms for the trajectory HMM. Chapter 7 describes the new search algorithm based on a delayed decision strategy. Results of speech recognition and synthesis experiments are shown in Chapter 8. Concluding remarks and future plans are presented in the final chapter.

Chapter 2

Hidden Markov Models

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 both in speech recognition and synthesis. This chapter describes its basic theories, how to calculate output probabilities of an observation vector sequence, and how to estimate its parameters.

2.1 Definition of HMM

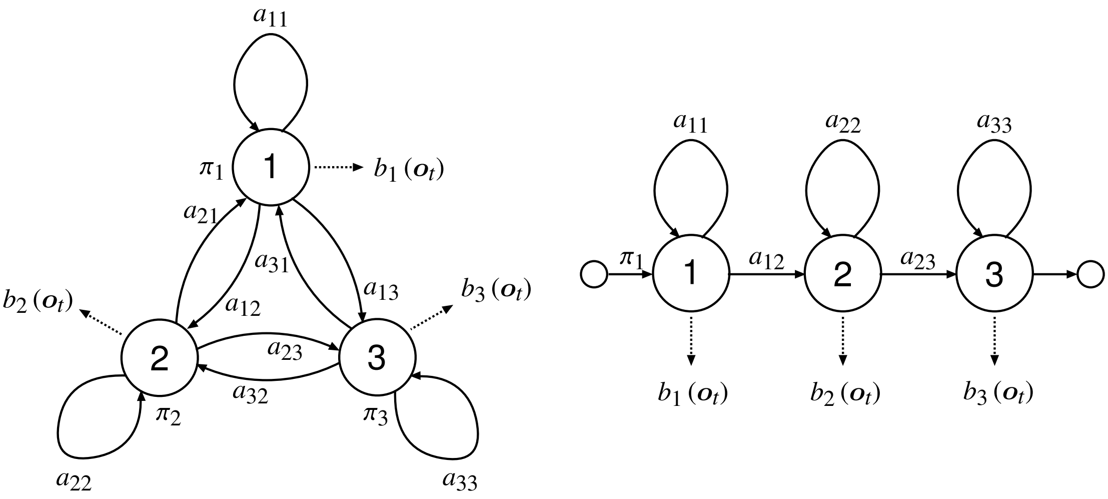
An HMM [49–51] is a finite state machine which generates a sequence of discrete time observations. At each time unit (i.e., frame), it changes states according to its state transition probability distributions, and then generates an observation at time t, $\boldsymbol{o}_t$, according to its output probability distribution of the current state. Hence, the HMM is a doubly stochastic random process model.

An N-state HMM is specified by state transition probability distributions $\left\{a_{ij}\right\}_{i,j=1}^{N}$, output probability distributions $\left\{b_j\left(\boldsymbol{o}_t\right)\right\}_{j=1}^{N}$, and initial state probability distributions $\left\{\pi_i\right\}_{i=1}^{N}$. For convenience, the compact notation

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

is used to indicate the parameter set of the model.

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 other state of the model in a single step, and Fig. 2.1(b) shows a 3-state left-to-right model, in which



(a) A 3-state ergodic model

(b) A 3-state left-to-right model

Figure 2.1: Examples of HMM structure.

the state index increases or stays the same as time increases. Generally, the left-to-right HMMs are used to model speech parameter sequences since they can appropriately model signals whose properties change in a successive manner.

The output probability distributions $\left\{b_j\left(\cdot\right)\right\}_{j=1}^{N}$ can be discrete or continuous depending on the observations. Usually in continuous distribution HMM (CD-HMM), each output probability distribution is modeled by a mixture of multivariate Gaussian components [52] as follows:

$$b_j\left(\boldsymbol{o}_t\right) = \sum_{m=1}^{M} w_{jm} \cdot \mathcal{N}\left(\boldsymbol{o}_t \;\middle|\; \boldsymbol{\mu}_{jm}, \boldsymbol{\Sigma}_{jm}\right), \tag{2.2}$$

where M is the number of Gaussian components, w_{jm}, $\boldsymbol{\mu}_{jm}$, $\boldsymbol{\Sigma}_{jm}$ are 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

$$\mathcal{N}\left(\boldsymbol{o}_t \;\middle|\; \boldsymbol{\mu}_{jm}, \boldsymbol{\Sigma}_{jm}\right) = \frac{1}{\sqrt{(2\pi)^K \left|\boldsymbol{\Sigma}_{jm}\right|}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{o}_t - \boldsymbol{\mu}_{jm}\right)^\top \boldsymbol{\Sigma}_{jm}^{-1}\left(\boldsymbol{o}_t - \boldsymbol{\mu}_{jm}\right)\right\}, \tag{2.3}$$

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

$$\sum_{m=1}^{M} w_{jm} = 1, \quad 1 \le j \le N \tag{2.4}$$

$$w_{jm} \ge 0, \quad \begin{array}{l} 1 \le j \le N \\ 1 \le m \le M \end{array} \tag{2.5}$$

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

$$\int_{\mathbb{R}^K} b_j\left(\boldsymbol{o}_t\right) d\boldsymbol{o}_t = 1. \quad 1 \le j \le N \tag{2.6}$$

2.2 Calculating output probability

2.2.1 Total output probability of an observation vector sequence from an HMM

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

$$p\left(\boldsymbol{o}, \boldsymbol{q} \mid \Lambda\right) = \prod_{t=1}^{T} a_{q_{t-1}q_t} b_{q_t}\left(\boldsymbol{o}_t\right), \tag{2.7}$$

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,

$$p\left(\boldsymbol{o} \mid \Lambda\right) = \sum_{\text{all } \boldsymbol{q}} \prod_{t=1}^{T} a_{q_{t-1}q_t} b_{q_t}\left(\boldsymbol{o}_t\right). \tag{2.8}$$

This definition involves on the order of $2T \cdot N^T$ calculation, 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), and for each term in the sum of Eq. (2.8). 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

Forward variables $\alpha_t(i)$ are defined as

$$\alpha_t(i) = p\left(\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_t, q_t = i \mid \Lambda\right) \tag{2.9}$$

that is, 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. We can calculate $\alpha_t(i)$ recursively as follows:

1. Initialization
$$\alpha_1(i) = \pi_i b_i\left(\boldsymbol{o}_1\right), \quad 1 \le i \le N \tag{2.10}$$

2. Recursion
$$\alpha_t(j) = \left[\sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij}\right] b_j\left(\boldsymbol{o}_t\right), \qquad \begin{array}{l} 1 \le j \le N \\ t = 2, \ldots, T \end{array} \tag{2.11}$$

3. Termination
$$p\left(\boldsymbol{o} \mid \Lambda\right) = \sum_{i=1}^{N} \alpha_T(i). \tag{2.12}$$

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

$$\beta_t(i) = p\left(\boldsymbol{o}_{t+1}, \boldsymbol{o}_{t+2}, \ldots, \boldsymbol{o}_T \mid s_t = i, \Lambda\right), \tag{2.13}$$

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
$$\beta_T(i) = 1, \quad 1 \le i \le N \tag{2.14}$$

2. Recursion
$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j\left(\boldsymbol{o}_{t+1}\right) \beta_{t+1}(j), \qquad \begin{array}{l} 1 \le i \le N \\ t = T-1, \ldots, 1. \end{array} \tag{2.15}$$

3. Termination
$$p\left(\boldsymbol{o} \mid \Lambda\right) = \sum_{i=1}^{N} \beta_1(i). \tag{2.16}$$

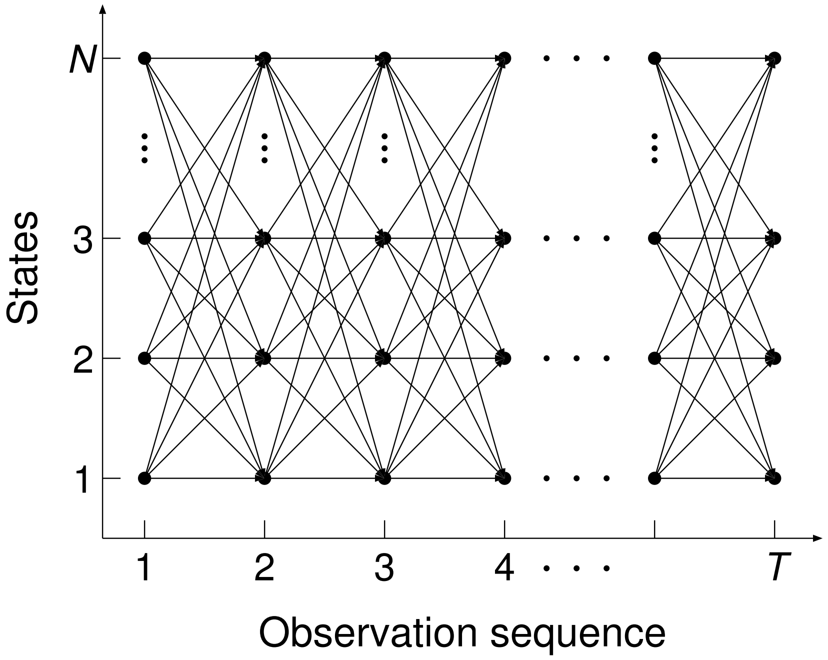


Figure 2.2: Implementation of the computation using forward-backward algorithm in terms of a trellis of observations and states.

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

$$p\,(\boldsymbol{o} \mid \Lambda) = \sum_{j=1}^{N} \alpha_t(j)\beta_t(j). \quad 1 \leq t \leq T \tag{2.17}$$

The forward-backward algorithm is based on the trellis structure shown in Fig. 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 remerge into these N nodes no matter how long the observation sequence. In the case of the forward algorithm, at time $t = 1$, we need to calculate values of $\alpha_1(i)$, $1 \leq i \leq N$. At times $t = 2, 3, \ldots, T$, we 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 optimum state sequence

The single optimum state sequence $\hat{\boldsymbol{q}} = \{\hat{q_1}, \hat{q_2}, \ldots, \hat{q_T}\}$ of length T for a given observation vector sequence $\boldsymbol{o} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_T\}$ is also useful for various applications (e.g., decoding, initializing HMM parameters).

The optimum state sequence $\hat{\boldsymbol{q}}$ can be obtained by a manner similar to the forward algorithm, which is often referred to as the Viterbi algorithm [53]. Let $\delta_t(i)$ be the likelihood of the most likely state sequence ending in the i-th state at time t

$$\delta_t(i) = \max_{q_1,\ldots,q_{t-1}} p\left(q_1, \ldots, q_{t-1}, q_t = i, \boldsymbol{o}_1, \ldots, \boldsymbol{o}_t \mid \Lambda\right), \tag{2.18}$$

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

1. Initialization

$$\delta_1(i) = \pi_i b_i(\boldsymbol{o}_1), \quad 1 \le i \le N \tag{2.19}$$

$$\psi_1(i) = 0, \quad 1 \le i \le N \tag{2.20}$$

2. Recursion

$$\delta_t(j) = \max_i \left[\delta_{t-1}(i)\, a_{ij}\right] b_j(\boldsymbol{o}_t), \quad \begin{array}{l} 1 \le i \le N \\ t = 2, 3, \ldots, T \end{array} \tag{2.21}$$

$$\psi_t(j) = \arg\max_i \left[\delta_{t-1}(i)\, a_{ij}\right], \quad \begin{array}{l} 1 \le i \le N \\ t = 2, 3, \ldots, T \end{array} \tag{2.22}$$

3. Termination

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

$$\hat{q_T} = \arg\max_i \left[\delta_T(i)\right]. \tag{2.24}$$

4. Back tracking

$$\hat{q}_t = \psi_{t+1}(\widehat{q_{t+1}}), \quad t = T-1, \ldots, 1. \tag{2.25}$$

It should be noted that the Viterbi algorithm is similar in the implementation 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, that is, to obtain Λ which maximizes its likelihood $p\left(\boldsymbol{o} \mid \Lambda\right)$ for a given observation sequence $\boldsymbol{o}$, 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 $p\left(\boldsymbol{o} \mid \Lambda\right)$. However, the model parameter set Λ locally maximizes $p\left(\boldsymbol{o} \mid \Lambda\right)$ can be obtained using an iterative procedure such as the expectation-maximization (EM) algorithm [54] (which is often referred to as the Baum-Welch algorithm [55]), and the obtained parameter set will be a good estimate 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 $\mathcal{Q}$-function

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

$$\mathcal{Q}(\Lambda, \hat{\Lambda}) = \sum_{\text{all } \boldsymbol{q}} p(\boldsymbol{q} \mid \boldsymbol{o}, \Lambda) \log p(\boldsymbol{o}, \boldsymbol{q} \mid \hat{\Lambda}). \tag{2.26}$$

Here, each mixture of Gaussian components is decomposed into a substate, and $\boldsymbol{q}$ is redefined as a substate sequence, i.e.,

$$\boldsymbol{q} = \left\{(q_1, s_1), (q_2, s_2), \ldots, (q_T, s_T)\right\}, \tag{2.27}$$

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 $\mathcal{Q}(\Lambda, \hat{\Lambda})$. This iterative procedure can be proved to increase likelihood $p\left(\boldsymbol{o} \mid \Lambda\right)$ monotonically and converge to a certain critical point, since it can be proved that the $\mathcal{Q}$-function satisfies the following theorems:

- Theorem 1

$$\mathcal{Q}(\Lambda, \hat{\Lambda}) \geq \mathcal{Q}(\Lambda, \Lambda) \;\Rightarrow\; p(\boldsymbol{o} \mid \hat{\Lambda}) \geq p(\boldsymbol{o} \mid \Lambda) \tag{2.28}$$

- 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 $p(\boldsymbol{o} \mid \Lambda)$ 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), $\log p\left(\boldsymbol{o}, \boldsymbol{q} \mid \Lambda\right)$ can be written as

$$\log p\left(\boldsymbol{o}, \boldsymbol{q} \mid \Lambda\right) = \log p\left(\boldsymbol{o} \mid \boldsymbol{q}, \Lambda\right) + \log P\left(\boldsymbol{q} \mid \Lambda\right), \tag{2.29}$$

$$\log p\left(\boldsymbol{o} \mid \boldsymbol{q}, \Lambda\right) = \sum_{t=1}^{T} \log \mathcal{N}\left(\boldsymbol{o}_t \;\middle|\; \boldsymbol{\mu}_{q_t s_t}, \boldsymbol{\Sigma}_{q_t s_t}\right), \tag{2.30}$$

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

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

$$\mathcal{Q}(\Lambda, \hat{\Lambda}) = \sum_{i=1}^{N} p\left(\boldsymbol{o}, q_1 = i \mid \Lambda\right) \cdot \log \pi_i \tag{2.32}$$

$$+ \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{t=1}^{T-1} p\left(\boldsymbol{o}, q_t = i, q_{t+1} = j\right) \cdot \log a_{ij} \tag{2.33}$$

$$+ \sum_{i=1}^{N} \sum_{m=1}^{M} \sum_{t=1}^{T} p\left(\boldsymbol{o}, q_t = i, s_t = m \mid \Lambda\right) \cdot \log w_{im} \tag{2.34}$$

$$+ \sum_{i=1}^{N} \sum_{m=1}^{M} \sum_{t=1}^{T} p\left(\boldsymbol{o}, q_t = i, s_t = m \mid \Lambda\right) \cdot \log \mathcal{N}\left(\boldsymbol{o}_t \mid \boldsymbol{\mu}_{im}, \boldsymbol{\Sigma}_{im}\right). \tag{2.35}$$

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

$$\sum_{i=1}^{N} \pi_i = 1, \tag{2.36}$$

$$\sum_{j=1}^{N} a_{ij} = 1, \quad 1 \le i \le N \tag{2.37}$$

$$\sum_{m=1}^{M} w_{im} = 1, \quad 1 \le i \le N \tag{2.38}$$

can be derived by Lagrange multipliers (Eqs. (2.32)–(2.33)) or differential calculus (Eq. (2.35)) as follows [56]:

$$\pi_i = \gamma_1(i), \quad 1 \le i \le N \tag{2.39}$$

$$a_{ij} = \frac{\sum_{t=2}^{T} \xi_{t-1}(i, j)}{\sum_{t=2}^{T} \gamma_{t-1}(i)}, \quad \begin{array}{l} 1 \le i \le N \\ 1 \le j \le N \end{array} \tag{2.40}$$

$$w_{im} = \frac{\sum_{t=1}^{T} \gamma_t(i, m)}{\sum_{t=1}^{T} \gamma_t(i)}, \quad \begin{array}{l} 1 \le i \le N \\ 1 \le m \le M \end{array} \tag{2.41}$$

$$\boldsymbol{\mu}_{im} = \frac{\sum_{t=1}^{T} \gamma_t(i, m) \cdot \boldsymbol{o}_t}{\sum_{t=1}^{T} \gamma_t(i, m)}, \quad \begin{array}{l} 1 \le i \le N \\ 1 \le m \le M \end{array} \tag{2.42}$$

$$\boldsymbol{\Sigma}_{im} = \frac{\sum_{t=1}^{T} \gamma_t(i, m) \cdot (\boldsymbol{o}_t - \boldsymbol{\mu}_{im}) (\boldsymbol{o}_t - \boldsymbol{\mu}_{im})^\top}{\sum_{t=1}^{T} \gamma_t(i, m)}, \quad \begin{array}{l} 1 \le i \le N \\ 1 \le m \le M \end{array} \tag{2.43}$$

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

$$\begin{aligned}\gamma_t\left(i\right) &= p\left(\boldsymbol{o}, q_t = i \mid \Lambda\right) \\ &= \frac{\alpha_t(i)\beta(i)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)}, \qquad \begin{array}{l} 1 \leq i \leq N \\ t = 1, \ldots, T \end{array}\end{aligned} \tag{2.44}$$

$$\begin{aligned}\gamma_t\left(i, m\right) &= p\left(\boldsymbol{o}, q_t = i, s_t = m \mid \Lambda\right) \\ &= \frac{\alpha_t(i)\beta(i)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)} \cdot \frac{w_{im}\mathcal{N}\left(\boldsymbol{o}_t \mid \boldsymbol{\mu}_{im}, \boldsymbol{\Sigma}_{im}\right)}{\sum_{k=1}^{M} w_{ik}\mathcal{N}\left(\boldsymbol{o}_t \mid \boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik}\right)}, \qquad \begin{array}{l} 1 \leq i \leq N \\ 1 \leq m \leq M \\ t = 1, \ldots, T \end{array}\end{aligned} \tag{2.45}$$

$$\begin{aligned}\xi_t(i, j) &= p\left(\boldsymbol{o}, q_t = i, q_{t+1} = j \mid \Lambda\right) \\ &= \frac{\alpha_t(i)a_{ij}b_j\left(\boldsymbol{o}_{t+1}\right)\beta_{t+1}(j)}{\sum_{l=1}^{N}\sum_{n=1}^{N} \alpha_t(l)a_{ln}b_n\left(\boldsymbol{o}_{t+1}\right)\beta_{t+1}(n)}. \qquad \begin{array}{l} 1 \leq i \leq N \\ t = 1, \ldots, T \end{array}\end{aligned} \tag{2.46}$$

2.5 Summary

This chapter has outlined the basic theories of the hidden Markov models (HMMs) and described its algorithms for calculating the output probability (forward-backward algorithm), searching the optimum state sequence (Viterbi algorithm), and estimating its parameters (EM algorithm). Further chapters will apply the HMMs for acoustic modeling both in speech recognition (Chapter 3) and synthesis (Chapter 4).

Chapter 3

HMM-based speech recognition

Most of the current speech recognition systems uses HMMs as its acoustic model. This chapter describes statistical speech recognition framework based on the HMM. 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 acoustic waveform is recorded and sampled to allow processing by a digital device. Then front-end processor converts the sampled waveform into an observation vector sequence $\boldsymbol{o} = \{\boldsymbol{o}_1, \ldots, \boldsymbol{o}_T\}$ by removing redundant or unimportant informations such as fundamental frequencies or noises. The brief overview of the front-end processor for speech recognition will be discussed later in Section 3.2. There is a considerable amount of variability in observation vector sequences even if the same words were uttered by the same speaker. Hence a statistical approach is adopted to map the observation vector sequence into the most likely word sequence. The speech recognizer usually choose the word sequence, $\boldsymbol{w} = \{w_1, \ldots, w_L\}$, with the maximum a posteriori (MAP) probability given the observation sequence as follows:

$$\hat{\boldsymbol{w}} = \arg\max_{\boldsymbol{w}} P(\boldsymbol{w} \mid \boldsymbol{o}) \tag{3.1}$$

Recently, discriminative models such as maximum entropy Markov models (MEMMs) [57] or conditional random fields (CRFs) [58] have been applied for modeling $P(\boldsymbol{w} \mid \boldsymbol{o})$

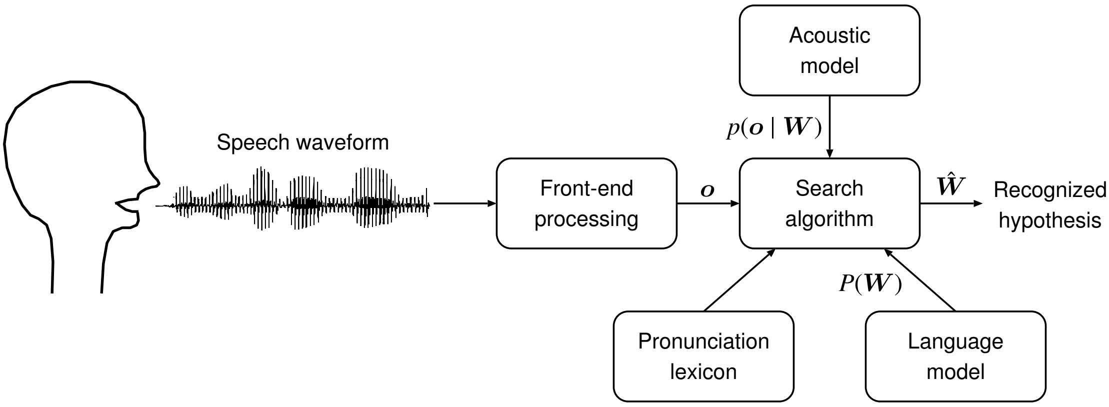


Figure 3.1: General statistical speech recognition framework.

directly [59,60]. 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

$$P(\boldsymbol{w} \mid \boldsymbol{o}) = \frac{p(\boldsymbol{o} \mid \boldsymbol{w}) P(\boldsymbol{w})}{p(\boldsymbol{o})}. \tag{3.2}$$

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

$$\hat{\boldsymbol{w}} = \arg\max_{\boldsymbol{w}} \; p(\boldsymbol{o} \mid \boldsymbol{w}) P(\boldsymbol{w}). \tag{3.3}$$

A general statistical speech recognition system illustrated in Fig. 3.1 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), $p(\boldsymbol{o} \mid \boldsymbol{w})$, corresponds to the acoustic model (second block), as it estimates the probability of an observation vector sequence $\boldsymbol{o}$, conditioned on the word sequence $\boldsymbol{w}$. The way in which we compute $p(\boldsymbol{o} \mid \boldsymbol{w})$, for large vocabulary continuous speech recognition, is to build statistical models for sub-word speech units, build up word models from these sub-word speech units using a pronunciation lexicon (fourth block), and then postulate word sequences and evaluate the acoustic model probabilities via concatenated word models. It is possible to use any kind of generative models for modeling $p(\boldsymbol{o} \mid \boldsymbol{w})$. Currently, most of speech recognition systems uses context-dependent sub-word HMMs as its acoustic model. The HMM-based acoustic modeling will be discussed later in Section 3.3.

The second term in Eq. (3.3), $P(\boldsymbol{w})$, corresponds to the language model (third block), as it describes the probability associated with a postulated sequence of words. Such language models can incorporate both syntactic and semantic constraints of the language and the recognition task. Often, when only syntactic constraints are used, the language model is called a grammar and may be of the form of a formal parser and syntax analyzer, and word N-gram model, or a word pair grammar of some type. Generally such language models are represented in a finite state network so as to be integrated into the acoustic model in a straightforward manner. The word N-gram model based language modeling is also briefly reviewed later in Section 3.4.

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

3.2 Front-ends

Comparing the sampled acoustic waveforms is not easy due to varying speaker and acoustic characteristics. Instead, the spectral shape of the speech signal conveys most of the significant information [61]. Acoustic front-ends in speech recognizers produce sequences of observation vectors which represent the short-term spectrum of the speech signal. There are a lot of techniques for parameterizing speech spectra, i.e., linear prediction coefficients (LPC) [62, 63], line spectral pair (LSP), cepstrum [64], mel-cepstrum [65], and so on. In most of the current speech recognition systems, mel filterbank cepstral coefficients (MFCC) [66] or perceptual linear prediction (PLP) [67] is commonly used. In all cases the speech signal is assumed to be quasi-stationary so that it can be decided into short frames, often 20–30ms. In each frame period a new parameterized short-time spectra vector is produced by analyzing a speech segment. In a typical final step, first and second-order dynamic features (delta and delta-delta coefficients) are appended to the acoustic vector [37–40]. Usually, the delta and delta-delta coefficients are calculated as regression coefficients from their neighbouring static features as follows:

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

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

$$\boldsymbol{o}_t = \left[\boldsymbol{c}_t^\top, \Delta\boldsymbol{c}_t^\top, \Delta^2\boldsymbol{c}_t^\top\right]^\top. \tag{3.5}$$

3.3 HMM-based acoustic modeling

The HMMs may be used to provide the estimates of $p(\boldsymbol{o} \mid \boldsymbol{w})$ in the speech recognizers. For isolated word recognition with sufficient training data, it is possible to build an HMM 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 have to be used. Usually an HMM is trained for each phone. The phone model set does not have to represent every phone in the language and it often includes silence and short pause models. The chosen phone set depends on the availability of sufficient training data. The pronunciation lexicon is used to map word sequences to phone sequences. The brief overview of pronunciation lexicon will be shown later in Section 3.5. 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 considerable amount of variation between realizations of the same phone depending on the succeeding and preceding phones. This effect is called co-articulation and is due to the inertia restricting any abrupt movement of the articulators. Context dependent phone models acknowledge the influence of the surrounding phones on the realization. Commonly used context-dependent phones are triphones which take the preceding and succeeding phones into account. The number of states, and model parameters, is significantly higher in a triphone system compared to a monophone system. It is therefore unlikely that sufficient training data will be available for reliable parameter estimation. The most common solution is to share some of the model parameters by tying the state output probability distributions among different models. An important question is how to determine when states should share the same parameters.

A phonetic decision tree [68–70] is often used to construct state tying structure in context-dependent systems. Figure 3.2 shows an example of a decision tree where binary ‘yes/no’ questions are asked. All instances of a phone are first pooled in the root node and the state clusters are split based on contextual questions. The splitting will terminate if the number of training data examples per state falls below a threshold. Expert knowledge may be incorporated into the decision tree and every state is guaranteed to have a minimum amount of training data. A disadvantage of decision tree-based state clustering is that the splits maximize the likelihood of the training data locally [71, 72].

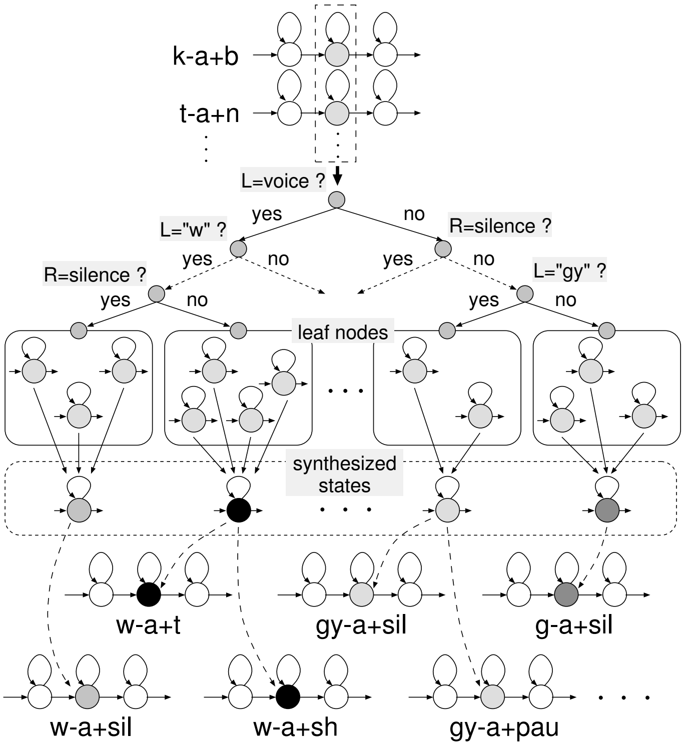


Figure 3.2: Example of a phonetic decision tree for triphone models.

3.4 Word *N*-gram-based language modeling

The language model provides the estimates of $P(\boldsymbol{w})$ in the speech recognizers. Using chain rule, this can be expressed as

$$P(\boldsymbol{w}) = \prod_{l=1}^{L} P(w_l \mid w_{l-1}, \ldots, w_1). \tag{3.6}$$

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

$$P(\boldsymbol{w}) = \prod_{l=1}^{L} P(w_l \mid w_{l-1}, \ldots, w_{l-N+1}). \tag{3.7}$$

Typical values are $N = 2, 3$ which are called bi-gram or trigram models, respectively. The ML estimation of the N-grams are obtained simply by counting relative frequencies from real, often domain specific, text documents. For a vocabulary of V words there are still V^N N-gram models. Some of the word sequences may be so absurd that zero probabilities may be assigned. However, for given a finite training data, valid word sequences may also be assigned a zero probability. A number of smoothing schemes such as discounting, backing off and deleted interpolation have been proposed [73].

There is often a mismatch between the contribution of the acoustic model and language model in the speech recognizers. This is due to different dynamic ranges of the discrete probability mass function, $P(\boldsymbol{w})$, estimated from a finite set of text documents and the acoustic likelihood score, $p(\boldsymbol{o} \mid \boldsymbol{w})$, obtained from high dimensional observation densities. To compensate this mismatch many systems raise the language model probability of the power of a constant called the grammar scale factor. The speech recognizers also tend to favor short words resulting in many insertion errors. This is often compensated for by introducing an insertion penalty which scales down the total score $p(\boldsymbol{o} \mid \boldsymbol{w}) P(\boldsymbol{w})$ depending on the number of hypothesized words in the sequence. By taking these modifications into account in Eq. (3.3), a practical speech recognizer uses

$$\hat{\boldsymbol{w}} = \arg\max_{\boldsymbol{w}} \Big[\log\{p(\boldsymbol{o} \mid \boldsymbol{w})\} + \alpha \log\{P(\boldsymbol{w}) + \beta L\} \Big] \tag{3.8}$$

where α is the grammar scale factor, β is the insertion penalty and L is the total number of words in the hypothesis. The parameters α and β are empirically set. The terms inside the maximization are often called the acoustic and language model scores. Logarithms are also taken to deal with the high dynamic range and prevent underflow due to repeated multiplications of values between zero and one.

3.5 Pronunciation lexicon

The pronunciation lexicon contains a set of the HMMs for each word in the vocabulary. Each word is defined by a pronunciation (or a set of pronunciations) obtained from a dictionary. The word HMM for a pronunciation of a word is the concatenation of the relevant sequence of basic sub-word HMMs. For computational efficiency, the lexicon is stored as a tree, where each node in the tree corresponds to an instantiation of a basic sub-word HMM. Tree-structured lexicon have been used by various researchers. This allows pronunciations with similar heads to share memory and computation when being evaluated. Owing to the tree structure, different pronunciations of a word are stored as separate lexical items. The primary disadvantage of using a tree-based lexicon is that it may not be an efficient approach to representing multiple pronunciations where there are only minor differences between pronunciations.

3.6 Search algorithms

The maximization problem in Eq. (3.3) can be compactly stated as follows:

> *Given a observation vector sequence, how can we determine the word sequence yielding the maximum combined probability from the acoustic and language model?*

To solve this problem, the following problems must be resolved:

1. Usually the number of words in given utterance is unknown.
2. Usually 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 for the given number of words in the utterance, L, there are V^L possible combinations of composite matching patterns; for anything but extremely small values of V and L the exponential number of composite matching patterns implies that the continuous speech recognition problem cannot be solved by exhaustive means.

Hence, an efficient search algorithms is required to solve continuous speech recognition problem. Fortunately, several efficient algorithms have been proposed that solve maximization problem in Eq. (3.3) without the exponential growth in computation. Most of these algorithms can categorized into two basic classes: time-synchronous Viterbi decoding [74] and stack decoding [75]. In its basic form, the Viterbi decoding may be regarded as an efficient recursive algorithm that performs an exhaustive search. On the other hand, stack decoding is an optimal, best-first search which is guided by a heuristic. However, for either of these algorithms to be computationally feasible for continuous speech recognition, the effective size of the search space must be reduced and the search algorithm must be efficient as possible. This may be achieved through pruning, fast lookahead or elimination of any repeated computations.

3.7 Summary

This chapter has outlined the statistical speech recognition framework and reviewd its main blocks: front-ends, HMM-based acoustic modeling, word N-gram-based language modeling, and search algorithm. The next chapter will describe statistical speech synthesis framework using the HMM.

Chapter 4

HMM-based speech synthesis

In the previous chapter, statistical framework for speech recognition system using the HMM was described. This chapter describes general statistical speech synthesis framework and speech parameter generation algorithm which is used in the HMM-based speech synthesis system.

4.1 Statistical speech synthesis

Text-to-speech synthesis can be viewed as an inverse procedure of speech recognition. The goal of a text-to-speech synthesizer is to take a word sequence as its input and produce an acoustic speech waveform. In a typical system, first given word sequence $\boldsymbol{w}$ is processed in a text analysis part. In this part, contextual informations (e.g., accent, lexical stress, part-of-speech, phrase boundary, etc.) are assigned to $\boldsymbol{w}$ by a natural language processor. Secondly, $\boldsymbol{w}$ is mapped into corresponding sub-word sequence with contextual informations $\boldsymbol{u}$. Finally, speech synthesizer generates speech waveform for given $\boldsymbol{u}$.

Most of state-of-the-art speech synthesis systems is based on a large amount of speech data. This type of system is generally called as a *corpus-based speech synthesis system* [76]. This approach makes it possible to dramatically improve the naturalness of synthetic speech compared with the early speech synthesis systems such as rule-based one.

One of the major approaches in the corpus-based speech synthesis area is sample-based system, such as *unit-selection and concatenation* [77–79]. In this system, speech corpus are segmented into small units, such as HMM state [4], half-phone [80], phone [81], diphone, syllable, etc. and stored. Then a unit sequence corresponding to given sub-word sequence is selected by minimizing total cost consisted of target and concatenation

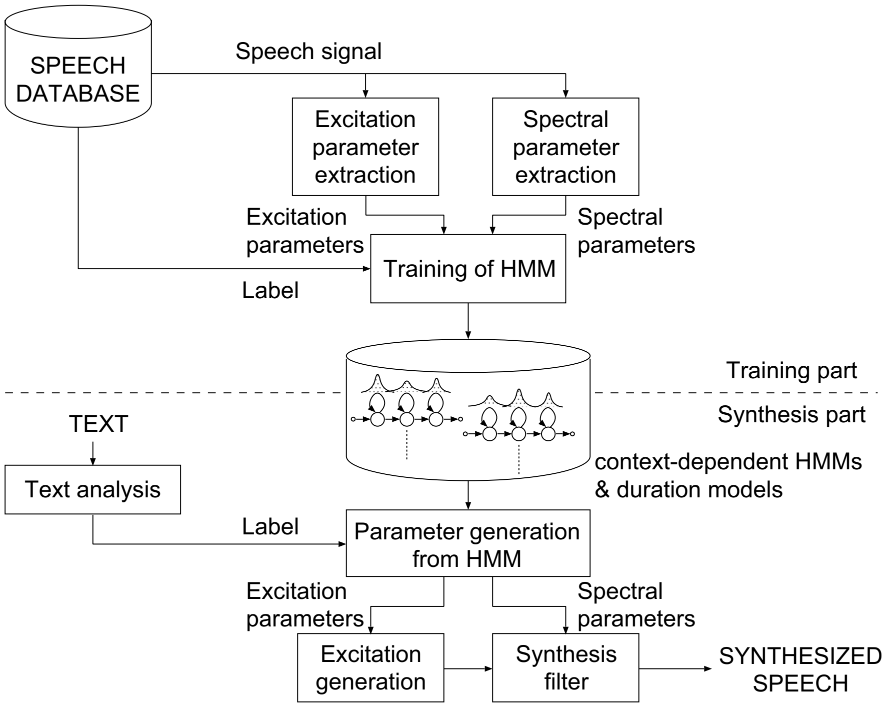


Figure 4.1: An overview of a typical HMM-based speech synthesis system.

costs [78]. These cost functions have been formed from a variety of heuristic or ad hoc quality measures based on features of the acoustic signal and given text. Recently, target and concatenation cost functions based on a statistical model have been proposed and investigated [82–84].

Another major approach is statistics-based system, such as *HMM-based speech synthesis* [5]. This system generates speech parameter sequence $\boldsymbol{o} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_T\}$ with the maximum a posteriori (MAP) probability given the sub-word sequence $\boldsymbol{u}$ as follows:

$$\hat{\boldsymbol{o}} = \arg\max_{\boldsymbol{o}} P\left(\boldsymbol{o} \mid \boldsymbol{u}\right). \tag{4.1}$$

The term in Eq. (4.1) has the same form to the first term (acoustic model) in Eq. (3.3). For speech recognition problem, Bayes'rule is required to use generative models (see Section 3.1). On the other hand, generative models can directly be applied for speech synthesis problem. Although any kind of generative models can be applied for modeling $p\left(\boldsymbol{o} \mid \boldsymbol{u}\right)$, currently the HMM is the most popular one. The detail of the HMM-based approach will be described later in the next section.

4.2 HMM-based speech synthesis

4.2.1 Overview

Figure 4.1 shows a block diagram of the HMM-based speech synthesis system [5]. It consists of the training and synthesis parts. In the training part, spectrum (e.g., Mel-cepstrum) and excitation (e.g., F_0) parameters are extracted from a speech database and modeled by context-dependent sub-word HMMs as the same manner in the speech recognition. State duration models are also estimated. In the synthesis part, a sentence HMM is constructed by concatenating the context-dependent sub-word HMMs according to a context-dependent label sequence which is mapped from a given text to be synthesized. Then, sequences of spectrum and excitation parameters are generated from the sentence HMM using speech parameter generation algorithm [41–43]. This algorithm will be briefly described later in Section 4.2.2. Finally, a synthesis filter module synthesizes speech waveform using the generated the speech parameters. The attraction of this approach is in that voice qualities of synthesized speech can easily be modified by transforming HMM parameters. In fact, it has been shown that its voice characteristics can be modified by speaker adaptation [44], speaker interpolation [45], or eigenvoice technique [46].

4.2.2 Speech parameter generation algorithm

Problem

For a sentence HMM $\Lambda_{\boldsymbol{u}}$ corresponding to a given sub-word sequence $\boldsymbol{u}$ (context-dependent label sequence), the speech synthesis problem is to obtain an output vector sequence consisted of spectral and excitation parameters

$$\boldsymbol{o} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_T\} \tag{4.2}$$

which maximizes its posterior probability with respect to $\boldsymbol{o}$, that is

$$\begin{aligned}
\hat{\boldsymbol{o}} &= \arg\max_{\boldsymbol{o}} p\,(\boldsymbol{o} \mid \Lambda_{\boldsymbol{u}}) && (4.3)\\
&= \arg\max_{\boldsymbol{o}} \sum_{\text{all } \boldsymbol{q}} p\,(\boldsymbol{o}, \boldsymbol{q} \mid \Lambda_{\boldsymbol{u}}) && (4.4)\\
&= \arg\max_{\boldsymbol{o}} \sum_{\text{all } \boldsymbol{q}} p\,(\boldsymbol{o} \mid \boldsymbol{q}, \Lambda_{\boldsymbol{u}})\, P\,(\boldsymbol{q} \mid \Lambda_{\boldsymbol{u}}) && (4.5)
\end{aligned}$$

where $\boldsymbol{q} = \{(q_1, s_1), (q_2, s_2), \ldots, (q_T, s_T)\}$ is a substate (Gaussian component) sequence, (q_t, s_t) represents being the s_t-th substate of the q_t-th state, respectively. Since there is no known method to analytically obtain $\boldsymbol{o}$ which maximizes $p(\boldsymbol{o} \mid \Lambda_{\boldsymbol{u}})$ in a closed form[1], this problem is approximated by a Viterbi approximation. As a result, this maximization problem can be decomposed into two stages: finding the best substate sequence $\hat{\boldsymbol{q}}$ for given $\Lambda_{\boldsymbol{u}}$ and obtaining $\boldsymbol{o}$ which maximizes $p(\boldsymbol{o} \mid \boldsymbol{q}, \Lambda_{\boldsymbol{u}})$ with respect to $\boldsymbol{o}$, i.e.,

$$\hat{\boldsymbol{q}} = \arg\max_{\boldsymbol{q}} P(\boldsymbol{q} \mid \Lambda_{\boldsymbol{u}}), \tag{4.6}$$

$$\hat{\boldsymbol{o}} = \arg\max_{\boldsymbol{o}} p(\boldsymbol{o} \mid \hat{\boldsymbol{q}}, \Lambda_{\boldsymbol{u}}). \tag{4.7}$$

In the HMM-based speech synthesis system proposed in [5], the optimization of Eq (4.6) is performed using explicit state duration models [85]. If the output vector at time t, $\boldsymbol{o}_t$, is determined independently of preceding and succeeding frames, the output vector sequence $\boldsymbol{o}$ which maximizes $p(\boldsymbol{o} \mid \boldsymbol{q}, \Lambda_{\boldsymbol{u}})$ is obtained as a sequence of mean vectors of substates. This causes discontinuity in the output vector sequence at transitions of substates, resulting in clicks in synthesized speech which degrade the naturalness [86]. To avoid this problem, dynamic features (delta and delta-delta coefficients) have been introduced. We assume that the output vector $\boldsymbol{o}_t$ consists of a static feature vector

$$\boldsymbol{c}_t = [c_t(1), \ldots, c_t(K)]^\top \tag{4.8}$$

and its dynamic features, that is

$$\boldsymbol{o}_t = \left[\boldsymbol{c}_t^\top, \Delta\boldsymbol{c}_t^\top, \Delta^2\boldsymbol{c}_t^\top\right]^\top, \tag{4.9}$$

where $\Delta\boldsymbol{c}_t$ and $\Delta^2\boldsymbol{c}_t$ are the first and second-order dynamic feature vectors (delta and delta-delta coefficients), respectively. They are calculated as the same manner used in the speech recognition (see Section 3.2):

$$\Delta\boldsymbol{c}_t = \sum_{\tau=-L_-^{(1)}}^{L_+^{(1)}} w^{(1)}(\tau)\boldsymbol{c}_{t+\tau}, \quad \Delta^2\boldsymbol{c}_t = \sum_{\tau=-L_-^{(2)}}^{L_+^{(2)}} w^{(2)}(\tau)\boldsymbol{c}_{t+\tau}. \tag{4.10}$$

Solution for the Problem

First, the output vector sequence $\boldsymbol{o}$ is rewritten in a vector form as

$$\boldsymbol{o} = [\boldsymbol{o}_1^\top, \boldsymbol{o}_2^\top, \ldots, \boldsymbol{o}_T^\top]^\top, \tag{4.11}$$

that is, $\boldsymbol{o}$ is a supervector composed of entire output vectors. In the same way, a static feature vector sequence $\boldsymbol{c}$ is defined as

$$\boldsymbol{c} = [\boldsymbol{c}_1^\top, \boldsymbol{c}_2^\top, \ldots, \boldsymbol{c}_T^\top]^\top. \tag{4.12}$$

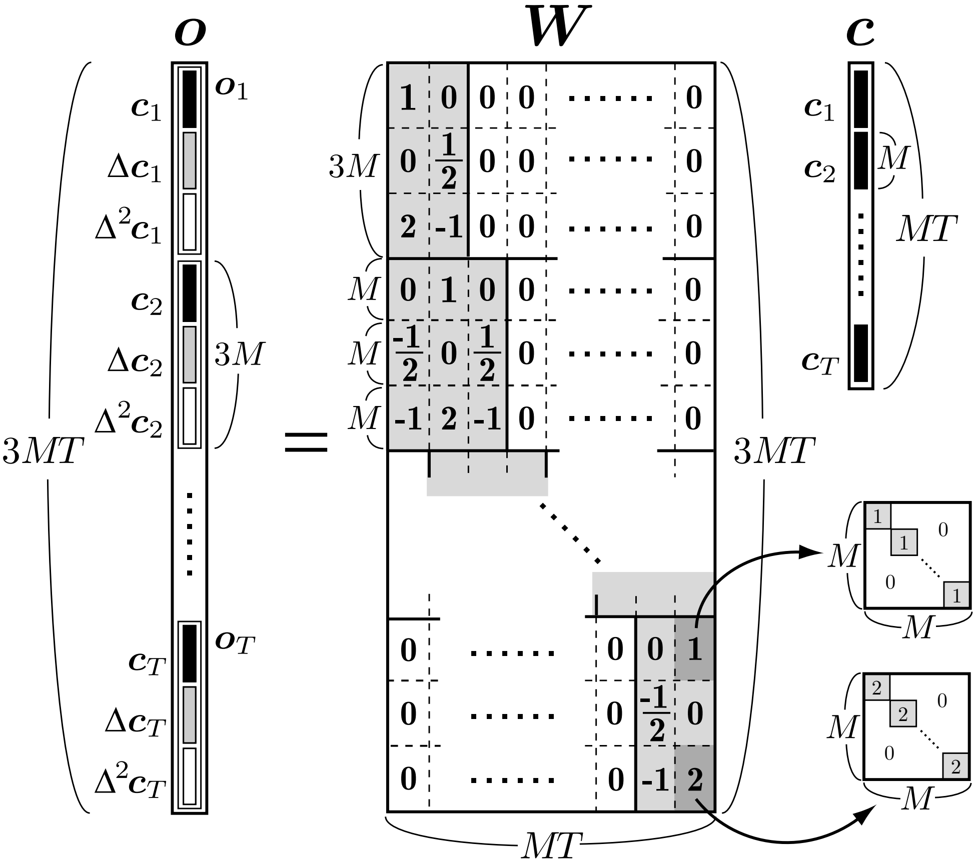


Figure 4.2: An example of the relationship between the static feature vector sequence $\boldsymbol{c}$ and the speech parameter vector sequence $\boldsymbol{o}$ in a matrix form (the dynamic features are calculated using $L_{-}^{(1)} = L_{+}^{(1)} = L_{-}^{(2)} = L_{+}^{(2)} = 1$, $w^{(1)}(-1) = -0.5$, $w^{(1)}(0) = 0.0$, $w^{(1)}(1) = 0.5$, $w^{(2)}(-1) = 1.0$, $w^{(2)}(0) = -2.0$, $w^{(2)}(1) = 1.0$).

Then, the relationship between $\boldsymbol{c}$ and $\boldsymbol{o}$ can be expressed in a matrix form as (see also Fig. 4.2):

$$\boldsymbol{o} = \boldsymbol{W}\boldsymbol{c}, \tag{4.13}$$

where $\boldsymbol{W}$ is a regression window matrix given by

$$\begin{aligned}
\boldsymbol{W} &= [\boldsymbol{W}_1, \boldsymbol{W}_2, \ldots, \boldsymbol{W}_T]^\top \otimes \boldsymbol{I}_{M\times M}, && (4.14)\\
\boldsymbol{W}_t &= \left[\boldsymbol{w}_t^{(0)}, \boldsymbol{w}_t^{(1)}, \boldsymbol{w}_t^{(2)}\right], && (4.15)\\
\boldsymbol{w}_t^{(0)} &= \Big[\underbrace{0, \ldots, 0}_{t-1}, 1, \underbrace{0, \ldots, 0}_{T-t}\Big]^\top, && (4.16)\\
\boldsymbol{w}_t^{(1)} &= \Big[\underbrace{0, \ldots, 0}_{t-L_-^{(1)}-1}, w^{(1)}(-L_-^{(1)}), \ldots, w^{(1)}(0), \ldots, w^{(1)}(L_+^{(1)}), \underbrace{0, \ldots, 0}_{T-\left(t+L_+^{(1)}\right)}\Big]^\top, && (4.17)\\
\boldsymbol{w}_t^{(2)} &= \Big[\underbrace{0, \ldots, 0}_{t-L_-^{(2)}-1}, w^{(2)}(-L_-^{(2)}), \ldots, w^{(2)}(0), \ldots, w^{(2)}(L_+^{(2)}), \underbrace{0, \ldots, 0}_{T-\left(t+L_+^{(2)}\right)}\Big]^\top, && (4.18)
\end{aligned}$$

The output probability of $\boldsymbol{o}$ conditioned on $\boldsymbol{q}$ is calculated by multiplying the output probabilities of entire observation vectors,

$$p\left(\boldsymbol{o} \mid \boldsymbol{q}, \Lambda_{\boldsymbol{u}}\right) = \prod_{t=1}^{T} \mathcal{N}\left(\boldsymbol{o}_t \;\middle|\; \boldsymbol{\mu}_{q_t s_t}, \boldsymbol{\Sigma}_{q_t s_t}\right), \tag{4.19}$$

where $\boldsymbol{\mu}_{q_t s_t}$ and $\boldsymbol{\Sigma}_{q_t s_t}$ are the $3K \times 1$ mean vector and $3K \times 3K$ covariance matrix, respectively, associated with the s_t-th substate of the q_t-th state. Equation (4.19) can be rewritten as an output probability of $\boldsymbol{o}$ from a single Gaussian component, that is

$$p\left(\boldsymbol{o} \mid \boldsymbol{q}, \Lambda_{\boldsymbol{u}}\right) = \mathcal{N}\left(\boldsymbol{o} \;\middle|\; \boldsymbol{\mu}_{\boldsymbol{q}}, \boldsymbol{\Sigma}_{\boldsymbol{q}}\right), \tag{4.20}$$

where $\boldsymbol{\mu}_{\boldsymbol{q}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{q}}$ are supervector and supermatrix corresponding to entire substate sequence $\boldsymbol{q}$, that is

$$\begin{aligned}
\boldsymbol{\Sigma}_{\boldsymbol{q}} &= \operatorname{diag}\left[\boldsymbol{\Sigma}_{q_1 s_1}, \boldsymbol{\Sigma}_{q_2 s_2}, \ldots, \boldsymbol{\Sigma}_{q_t s_t}\right], && (4.21)\\
\boldsymbol{\mu}_{\boldsymbol{q}} &= \left[\boldsymbol{\mu}_{q_1 s_1}^\top, \boldsymbol{\mu}_{q_2 s_2}^\top, \ldots, \boldsymbol{\mu}_{q_t s_t}^\top\right]^\top. && (4.22)
\end{aligned}$$

Thus, the logarithm of Eq. (4.19) can be written as

$$\log \mathcal{N}\left(\boldsymbol{o} \;\middle|\; \boldsymbol{\mu}_{\boldsymbol{q}}, \boldsymbol{\Sigma}_{\boldsymbol{q}}\right) = -\frac{1}{2}\left\{3KT \log 2\pi + \log\left|\boldsymbol{\Sigma}_{\boldsymbol{q}}\right| + \left(\boldsymbol{o} - \boldsymbol{\mu}_{\boldsymbol{q}}\right)^\top \boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1}\left(\boldsymbol{o} - \boldsymbol{\mu}_{\boldsymbol{q}}\right)\right\}. \tag{4.23}$$

Under the condition (4.13), maximizing $\mathcal{N}\left(\boldsymbol{o} \;\middle|\; \boldsymbol{\mu}_{\boldsymbol{q}}, \boldsymbol{\Sigma}_{\boldsymbol{q}}\right)$ with respect to $\boldsymbol{o}$ is equivalent to that with respect to $\boldsymbol{c}$. By setting

$$\frac{\partial \log \mathcal{N}\left(\boldsymbol{o} \;\middle|\; \boldsymbol{\mu}_{\boldsymbol{q}}, \boldsymbol{\Sigma}_{\boldsymbol{q}}\right)}{\partial \boldsymbol{c}} = \boldsymbol{0}_{KT}, \tag{4.24}$$

[1] An algorithm to obtain $\boldsymbol{o}$ which maximizes $p\left(\boldsymbol{o} \mid \Lambda_{\boldsymbol{u}}\right)$ using the EM algorithm is shown in [41].

we obtain a set of linear equations

$$\boldsymbol{R_q c} = \boldsymbol{r_q}, \tag{4.25}$$

where $\mathbf{0}_{KT}$ is a KT-dimensional zero vector, $\boldsymbol{R_q}$ and $\boldsymbol{r_q}$ are given as

$$\boldsymbol{R_q} = \boldsymbol{W\Sigma_q^{-1}W}, \tag{4.26}$$

$$\boldsymbol{r_q} = \boldsymbol{W\Sigma_q^{-1}\mu_q}. \tag{4.27}$$

For direct solution of Eq. (4.25), $O(K^3T^3)$ operations is required because $\boldsymbol{R_q}$ is a $KT \times KT$ matrix. By utilizing the special structure of $\boldsymbol{R_q}$, Eq. (4.25) can be solved by the Cholesky or QR decomposition with $O(K^3L^2T)$ operations[2]. Equation (4.25) can also be solved by an algorithm derived in [41–43], which can operate in a time-recursive manner [88].

4.3 Summary

In this chapter, a statistical speech synthesis framework and the speech parameter generation algorithm has been reviewed. The next chapter will derive a new statistical model, named trajectory HMM by imposing explicit relationship between static and dynamic features into the HMM.

[2] This is reduced to $O\left(KL^2T\right)$ when $\boldsymbol{\Sigma_q}$ is diagonal because each of the K-dimensions can be calculated independently. Furthermore, when $L_-^{(1)} = -1$, $L_+^{(1)} = 0$, and $\forall_i \; w^{(2)}(i) \equiv 0$, it is reduced to $O(KT)$ as described in [87].

Chapter 5

Reformulating the HMM as a trajectory model

In the previous chapters, theoretical overview of the HMMs and its applications for acoustic modeling both in speech recognition and synthesis were reviewed. However, a number of limitations for modeling sequences of speech spectra by the HMMs have been pointed out [6–8]. In the present dissertation, the following two limitations are considered:

1. One of the underlying assumptions in the HMMs is that observation vector sequences are quasi-stationary and each stationary part can be represented by a single state of the HMM. The statistics of each state does not vary dynamically. Therefore, intra-state time-dependency cannot be represented. Although this assumption might be satisfied in reading speech, it isn't valid in spontaneous speech because they are dynamically affected by various factors such as speaking rate.

2. The output probability of an observation vector depends only on the current state, neither on any other states nor observations. It is generally called as *conditional independence assumption*.

To overcome these shortcomings of the HMM, a number of alternative acoustic models have been proposed. Most of them have attempted to capture explicit dynamics of speech parameter trajectories. Although the above models can improve model accuracy and speech recognition performance, they generally require an increase in the number of model parameters and computational complexity.

Alternatively, dynamic features (delta and delta-delta coefficients) [37,38] have been used to capture time dependencies in the HMM-based acoustic modeling framework. This greatly enhances the performance of the HMM-based recognizers [39, 40] and it is essential in the HMM-based synthesizers. However, it has been thought as an ad hoc rather than an essential solution. Generally, dynamic features are calculated as regression coefficients from their neighboring static features (see Eq. (3.4)). Therefore, relationships between static and dynamic features are *deterministic* (see Eq. (4.13)). However, these relationships are ignored and static and dynamic features are modeled as independent statistical variables. Ignoring these interdependencies allows inconsistency between the static and dynamic feature vector sequences when the HMM is used as a generative model in the obvious way.

In this chapter, a novel statistical model is derived by reformulating the HMM whose state output vector includes both static and dynamic features. By imposing the explicit relationships between the static and dynamic features, the HMM is naturally translated into a trajectory model, referred to as *trajectory HMM* in the present dissertation. The trajectory HMM can overcome the above two limitations of the HMM without any additional parameters. Furthermore, it provides a computational model for co-articulation of human speech. Relationships between the trajectory HMM and other techniques is also discussed.

5.1 Reformulating HMM as a trajectory model

5.1.1 Imposing explicit relationships between static and dynamic features into HMM

The total output probability of an observation vector sequence $\boldsymbol{o} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_T\}$ of length T for an HMM Λ is given by

$$p(\boldsymbol{o} \mid \Lambda) = \sum_{\text{all } \boldsymbol{q}} p(\boldsymbol{o} \mid \boldsymbol{q}, \Lambda) P(\boldsymbol{q} \mid \Lambda), \tag{5.1}$$

where $\boldsymbol{o}_t$ is an observation vector at time t. Here, we assume that each state output probability is modeled by a mixture of M-Gaussian components (Eq. (2.2)) and each mixture of Gaussian components is decomposed into a substate (Gaussian component). Then, $\boldsymbol{q}$ is defined as a substate sequence, i.e.,

$$\boldsymbol{q} = \{(q_1, s_1), (q_2, s_2), \ldots, (q_T, s_T)\}, \tag{5.2}$$

where (q_t, s_t) denotes being the s_t-th substate of the q_t-th state at time t. Hence, the first and second terms in Eq. (5.1) can be written as

$$p\,(\boldsymbol{o} \mid \boldsymbol{q}, \Lambda) = \prod_{t=1}^{T} p\,(\boldsymbol{o}_t \mid q_t, s_t, \lambda) = \prod_{t=1}^{T} \mathcal{N}\left(\boldsymbol{o}_t \;\middle|\; \boldsymbol{\mu}_{q_t s_t}, \boldsymbol{\Sigma}_{q_t s_t}\right), \tag{5.3}$$

$$P\,(\boldsymbol{q} \mid \Lambda) = \prod_{t=1}^{T} P\,(q_t, s_t \mid q_{t-1}, s_{t-1}, \Lambda) = \prod_{t=1}^{T} a_{q_{t-1}q_t} w_{q_t s_t}, \tag{5.4}$$

where $w_{q_t s_t}$, $\boldsymbol{\mu}_{q_t s_t}$ and $\boldsymbol{\Sigma}_{q_t s_t}$ are the mixture weight, mean vector and covariance matrix of the s_t-th substate of the q_t-th state, respectively, and $a_{q_{t-1}q_t}$ is the state transition probability from the q_{t-1}-th state to the q_t-th state, and $a_{q_0 q_1}$ denotes a initial state probability of the q_1-th state.

Typically, the observation vector $\boldsymbol{o}_t$ is a $3K \times 1$ vector consists of a K-dimensional static feature vector (e.g., cepstral coefficients, etc.)

$$\boldsymbol{c}_t = [c_t(1), c_t(2), \ldots, c_t(K)]^\top, \tag{5.5}$$

and its dynamic feature vectors (e.g., delta and delta-delta cepstral coefficients), that is

$$\boldsymbol{o}_t = \left[\boldsymbol{c}_t^\top, \Delta\boldsymbol{c}_t^\top, \Delta^2\boldsymbol{c}_t^\top\right]^\top, \tag{5.6}$$

where $\Delta\boldsymbol{c}_t$ and $\Delta^2\boldsymbol{c}_t$ are first and second order dynamic feature vectors, respectively. Usually, the dynamic features are calculated as regression coefficients from their neighboring static features, that is

$$\Delta\boldsymbol{c}_t = \sum_{\tau=-L_-^{(1)}}^{L_+^{(1)}} w^{(1)}(\tau)\boldsymbol{c}_{t+\tau}, \quad \Delta^2\boldsymbol{c}_t = \sum_{\tau=-L_-^{(2)}}^{L_+^{(2)}} w^{(2)}(\tau)\boldsymbol{c}_{t+\tau}, \tag{5.7}$$

where $\left\{w^{(d)}(\tau)\right\}_{d=1,2\;\tau=-L_-^{(d)},\ldots,L_+^{(d)}}$ are window coefficients to calculate dynamic features.

As the same manner used in the speech parameter generation algorithm described in Section 4.2.2, the observation vector sequence $\boldsymbol{o}$ and the static feature vector sequence $\boldsymbol{c}$ can be written in a vector form as

$$\boldsymbol{o} = [\boldsymbol{o}_1^\top, \boldsymbol{o}_2^\top, \ldots, \boldsymbol{o}_T^\top]^\top, \tag{5.8}$$

$$\boldsymbol{c} = [\boldsymbol{c}_1^\top, \boldsymbol{c}_2^\top, \ldots, \boldsymbol{c}_T^\top]^\top, \tag{5.9}$$

that is, $\boldsymbol{o}$ and $\boldsymbol{c}$ are supervectors made from entire observation and static feature vectors, respectively. Then, the relationship between $\boldsymbol{o}$ and $\boldsymbol{c}$ can be arranged in a matrix form as

$$\boldsymbol{o} = \boldsymbol{W}\boldsymbol{c}, \tag{5.10}$$

where $\boldsymbol{W}$ is a window matrix to calculate $\boldsymbol{o}$ from $\boldsymbol{c}$, that is

$$
\begin{aligned}
\boldsymbol{W} &= [\boldsymbol{W}_1, \boldsymbol{W}_2, \ldots, \boldsymbol{W}_T]^\top \otimes \boldsymbol{I}_{K\times K}, && (5.11)\\
\boldsymbol{W}_t &= \left[\boldsymbol{w}_t^{(0)}, \boldsymbol{w}_t^{(1)}, \boldsymbol{w}_t^{(2)}\right], && (5.12)\\
\boldsymbol{w}_t^{(0)} &= \Big[\underbrace{0, \ldots, 0}_{t-1}, 1, \underbrace{0, \ldots, 0}_{T-t}\Big]^\top, && (5.13)\\
\boldsymbol{w}_t^{(1)} &= \Big[\underbrace{0, \ldots, 0}_{t-L_-^{(1)}-1}, w^{(1)}(-L_-^{(1)}), \ldots, w^{(1)}(0), \ldots, w^{(1)}(L_+^{(1)}), \underbrace{0, \ldots, 0}_{T-\left(t+L_+^{(1)}\right)}\Big]^\top, && (5.14)\\
\boldsymbol{w}_t^{(2)} &= \Big[\underbrace{0, \ldots, 0}_{t-L_-^{(2)}-1}, w^{(2)}(-L_-^{(2)}), \ldots, w^{(2)}(0), \ldots, w^{(2)}(L_+^{(2)}), \underbrace{0, \ldots, 0}_{T-\left(t+L_+^{(2)}\right)}\Big]^\top. && (5.15)
\end{aligned}
$$

According to Eq. 5.3 and assuming independence between the static and dynamic features, the output probability of $\boldsymbol{o}$ conditioned on $\boldsymbol{q}$ can be calculated as

$$
p\left(\boldsymbol{o} \mid \boldsymbol{q}, \Lambda\right) = \mathcal{N}\left(\boldsymbol{o} \;\middle|\; \boldsymbol{\mu}_{\boldsymbol{q}}, \boldsymbol{\Sigma}_{\boldsymbol{q}}\right), \tag{5.16}
$$

where $\boldsymbol{\mu}_{\boldsymbol{q}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{q}}$ are $3KT\times 1$ and $3KT\times 3KT$ supervector and supermatrix corresponding to entire substate sequence $\boldsymbol{q}$, respectively, that is

$$
\begin{aligned}
\boldsymbol{\mu}_{\boldsymbol{q}} &= \left[\boldsymbol{\mu}_{q_1 s_1}^\top, \boldsymbol{\mu}_{q_2 s_2}^\top, \ldots, \boldsymbol{\mu}_{q_T s_T}^\top\right]^\top, && (5.17)\\
\boldsymbol{\mu}_{q_t s_t} &= \left[\Delta^{(0)}\boldsymbol{\mu}_{q_t s_t}^\top, \Delta^{(1)}\boldsymbol{\mu}_{q_t s_t}^\top, \Delta^{(2)}\boldsymbol{\mu}_{q_t s_t}^\top\right]^\top, && (5.18)\\
\Delta^{(d)}\boldsymbol{\mu}_{q_t s_t} &= \left[\Delta^{(d)}\mu_{q_t s_t}(1), \ldots, \Delta^{(d)}\mu_{q_t s_t}(K)\right]^\top, \quad d = 0, 1, 2 && (5.19)\\
\boldsymbol{\Sigma}_{\boldsymbol{q}} &= \operatorname{diag}\left[\boldsymbol{\Sigma}_{q_1 s_1}, \boldsymbol{\Sigma}_{q_2 s_2}, \ldots, \boldsymbol{\Sigma}_{q_T s_T}\right], && (5.20)\\
\boldsymbol{\Sigma}_{q_t s_t} &= \operatorname{diag}\left[\Delta^{(0)}\boldsymbol{\Sigma}_{q_t s_t}, \Delta^{(1)}\boldsymbol{\Sigma}_{q_t s_t}, \ldots, \Delta^{(2)}\boldsymbol{\Sigma}_{q_t s_t}\right], && (5.21)\\
\Delta^{(d)}\boldsymbol{\Sigma}_{q_t s_t} &= \operatorname{diag}\left[\Delta^{(d)}\sigma_{q_t s_t}(1), \ldots, \Delta^{(d)}\sigma_{q_t s_t}(K)\right]. \quad d = 0, 1, 2 && (5.22)
\end{aligned}
$$

However, the above model is mathematically improper in the sense of statistical modeling. In this model, *the static and dynamic features are modeled as independent statistical variables*. When it is used as a generative model, it allows inconsistent static and dynamic features even though they are constrained by Eq. (5.10) [89]. To avoid this problem, the statistical model should be defined as a function of $\boldsymbol{c}$ because the original observation is $\boldsymbol{c}$ rather than $\boldsymbol{o}$. By introducing the explicit relationships between $\boldsymbol{o}$ and $\boldsymbol{c}$, Eq. (5.16) becomes invalid probability distribution because

$$
\int_{\mathbb{R}^{KT}} \mathcal{N}\left(\boldsymbol{W}\boldsymbol{c} \;\middle|\; \boldsymbol{\mu}_{\boldsymbol{q}}, \boldsymbol{\Sigma}_{\boldsymbol{q}}\right) d\boldsymbol{c} \neq 1, \tag{5.23}
$$

where the integral is over the KT-dimensional feature-space. Therefore, it should be normalized to yield a valid probability distribution. In the present paper, the output probability of $\boldsymbol{c}$ conditioned on $\boldsymbol{q}$ is defined using normalized $\mathcal{N}\left(\boldsymbol{W}\boldsymbol{c} \;\middle|\; \boldsymbol{\mu}_{\boldsymbol{q}}, \boldsymbol{\Sigma}_{\boldsymbol{q}}\right)$ as follows (see Appendix A for detail):

$$p\left(\boldsymbol{c} \mid \boldsymbol{q}, \Lambda\right) = \frac{1}{Z_{\boldsymbol{q}}} \mathcal{N}\left(\boldsymbol{W}\boldsymbol{c} \mid \boldsymbol{\mu}_{\boldsymbol{q}}, \boldsymbol{\Sigma}_{\boldsymbol{q}}\right) = \mathcal{N}\left(\boldsymbol{c} \mid \bar{\boldsymbol{c}}_{\boldsymbol{q}}, \boldsymbol{P}_{\boldsymbol{q}}\right), \tag{5.24}$$

$$Z_{\boldsymbol{q}} = \int_{\mathbb{R}^{MT}} \mathcal{N}\left(\boldsymbol{W}\boldsymbol{c} \mid \boldsymbol{\mu}_{\boldsymbol{q}}, \boldsymbol{\Sigma}_{\boldsymbol{q}}\right) d\boldsymbol{c} \tag{5.25}$$

$$= \frac{\sqrt{(2\pi)^{MT} \left|\boldsymbol{P}_{\boldsymbol{q}}\right|}}{\sqrt{(2\pi)^{3MT} \left|\boldsymbol{\Sigma}_{\boldsymbol{q}}\right|}} \cdot \exp\left\{-\frac{1}{2}\left(\boldsymbol{\mu}_{\boldsymbol{q}}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1}\boldsymbol{\mu}_{\boldsymbol{q}} - \boldsymbol{r}_{\boldsymbol{q}}^{\top}\boldsymbol{P}_{\boldsymbol{q}}\boldsymbol{r}_{\boldsymbol{q}}\right)\right\}, \tag{5.26}$$

where $Z_{\boldsymbol{q}}$ is a normalization term, and $\bar{\boldsymbol{c}}_{\boldsymbol{q}}$, $\boldsymbol{P}_{\boldsymbol{q}}$, $\boldsymbol{R}_{\boldsymbol{q}}$ and $\boldsymbol{r}_{\boldsymbol{q}}$ are calculated from $\boldsymbol{W}$, $\boldsymbol{\mu}_{\boldsymbol{q}}$, and $\boldsymbol{\Sigma}_{\boldsymbol{q}}$ as

$$\boldsymbol{R}_{\boldsymbol{q}} = \boldsymbol{W}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1}\boldsymbol{W} = \boldsymbol{P}_{\boldsymbol{q}}^{-1}, \tag{5.27}$$

$$\boldsymbol{r}_{\boldsymbol{q}} = \boldsymbol{W}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1}\boldsymbol{\mu}_{\boldsymbol{q}}, \tag{5.28}$$

$$\boldsymbol{R}_{\boldsymbol{q}}\bar{\boldsymbol{c}}_{\boldsymbol{q}} = \boldsymbol{r}_{\boldsymbol{q}}. \tag{5.29}$$

5.1.2 Definition of trajectory HMM

By marginalizing $p\left(\boldsymbol{c} \mid \boldsymbol{q}, \Lambda\right) P\left(\boldsymbol{q} \mid \Lambda\right)$ over all possible substate sequences, a new statistical model can be defined as follows:

$$p\left(\boldsymbol{c} \mid \Lambda\right) = \sum_{\text{all } \boldsymbol{q}} p\left(\boldsymbol{c} \mid \boldsymbol{q}, \Lambda\right) P\left(\boldsymbol{q} \mid \Lambda\right) \tag{5.30}$$

where

$$p\left(\boldsymbol{c} \mid \boldsymbol{q}, \Lambda\right) = \frac{1}{Z_{\boldsymbol{q}}} \mathcal{N}\left(\boldsymbol{W}\boldsymbol{c} \;\middle|\; \boldsymbol{\mu}_{\boldsymbol{q}}, \boldsymbol{\Sigma}_{\boldsymbol{q}}\right) \tag{5.31}$$

$$= \mathcal{N}\left(\boldsymbol{c} \;\middle|\; \bar{\boldsymbol{c}}_{\boldsymbol{q}}, \boldsymbol{P}_{\boldsymbol{q}}\right), \tag{5.32}$$

$$P\left(\boldsymbol{q} \mid \Lambda\right) = \prod_{t=1}^{T} a_{q_{t-1}q_t} w_{q_t s_t}, \tag{5.33}$$

In the present dissertation, the above model is referred to as *trajectory HMM*.

It is interesting to note that its mean vector $\bar{\boldsymbol{c}}_{\boldsymbol{q}}$ is exactly the same as the output vector sequence $\boldsymbol{c}$ obtained by the speech parameter generation algorithm which has been described in Section 4.2.2. By assuming $\bar{\boldsymbol{c}}_{\boldsymbol{q}}$ is the mean for the static feature vector sequence

$\boldsymbol{c}$, corresponding to an utterance, the HMM whose state output vector includes both static and dynamic features can naturally be translated into a trajectory model: statistics of the state output probability distributions varies within a state, and they are affected by statistics at neighboring times. Note that $\boldsymbol{c}$ is modeled by a mixture of $(NM)^T$-Gaussian components whose dimensionality is KT, and their covariances $\boldsymbol{P_q}$ are generally full. As a result, the trajectory HMM can alleviate the deficiency of the HMM. It is also noted that parameterization of the trajectory HMM is completely the same as the HMM with the same model topology. Therefore, any additional parameters are not required.

5.2 Example

Figure 5.1 shows an example of a trajectory HMM. Model training conditions are the same as those in Section 8.2. To obtain a substate sequence $\boldsymbol{q}$, a concatenated model composed of monophone models /sil/, /a/, /i/, /d/, /a/, /sil/ was aligned to a natural speech not included in the training data using a delayed decision Viterbi algorithm which will be described in later Chapter 7. Note that only elements corresponding to the first coefficient of mel-cepstrum are shown in the figure. It can be seen that not only the mean vector $\bar{\boldsymbol{c}}_{\boldsymbol{q}}$ varies in each state but also the inter-frame correlation can be modeled by the covariance matrix $\boldsymbol{P_q}$. It is also interesting to note that the mean vector and the inter-frame covariance corresponding to each monophone model vary according to its durations and neighboring models (see phoneme /a/ in Fig. 5.1). This shows that the trajectory HMM has the capability to capture the co-articulation effects naturally.

5.3 Relation to other techniques

5.3.1 Relation to HMM-trajectory method

The idea of the trajectory HMM was originally inspired from the HMM-trajectory method [47, 48]. In this method, speech parameter vector sequence obtained by the speech parameter generation algorithm with sliding window [88][1] was used as its mean vector and variance between mean trajectory and training data was modeled using additional parameters. The essential difference between this method and trajectory HMM is in that this method requires additional parameters and the inter-frame correlations are not modeled explicitly.

[1] This algorithm can be viewed as a Kalman filtering for mean vector sequence of the standard HMM [90].

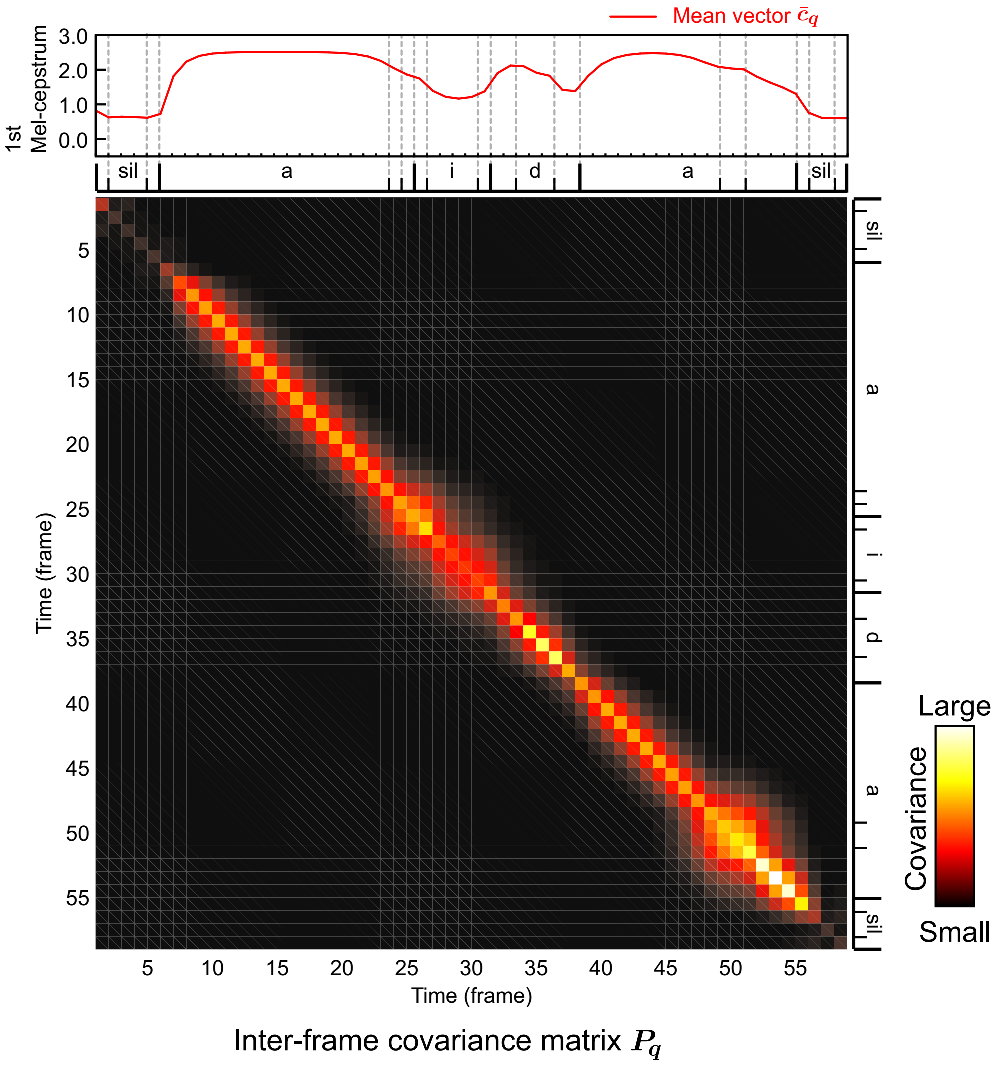


Figure 5.1: An example of the static feature sequence $\boldsymbol{c}$, the mean trajectory $\bar{\boldsymbol{c}}_q$ and covariance matrix $\boldsymbol{P}_q$ for a word “aida”.

5.3.2 Relations to structured precision matrix modeling techniques

Recently, a number of techniques for modeling inverse covariance (precision) matrices to capture intra-frame correlation efficiently in large vocabulary continuous speech recognition system have been proposed [91, 92]. Models that have been successfully applied include Semi-Tied Covariance matrices (STC) [93], Extended Maximum Likelihood Linear Transform (EMLLT) [94], and Subspace for Precision And Mean (SPAM) [94]. The precision matrix models mentioned above can be described within a generic framework of basis superposition [95]. The trajectory HMM can be viewed as a basis superposition for inter-frame precision matrices.

For notation simplicity, the dimensionality of static feature vector K is assumed to be 1. From Eqs. (5.11)–(5.15), (5.20)–(5.22) and (5.27), the precision matrix of the trajectory HMM, $\boldsymbol{R_q}$, can be rewritten as follows:

$$\begin{aligned}
\boldsymbol{R_q} &= \boldsymbol{P_q}^{-1} && (5.34)\\
&= \boldsymbol{W}^\top \boldsymbol{\Sigma_q}^{-1} \boldsymbol{W} && (5.35)\\
&= \sum_{t=1}^{T} \sum_{d=0}^{2} \frac{1}{\Delta^{(d)} \sigma_{q_t s_t}(1)} \cdot \left[\boldsymbol{w}_t^{(d)}\right]^\top \boldsymbol{w}_t^{(d)}. && (5.36)
\end{aligned}$$

In the above equation, $T \times T$ precision matrix $\boldsymbol{R_q}$ is represented as a weighted sum of $3T$ rank-1 symmetric matrices. This form can be viewed as the EMLLT for inter-frame precision matrix. In the general EMLLT framework, both the basis and diagonal matrices, $\boldsymbol{W}$ and $\boldsymbol{\Sigma_q}$, are estimated. However, in the trajectory HMM the basis matrix is given as a window matrix. In addition, the diagonal matrix $\boldsymbol{\Sigma_q}$ has parameter sharing structure according to the substate sequence $\boldsymbol{q}$. Thus, inter-frame correlation can be captured efficiently without increasing the number of model parameters compared with the HMM. Furthermore, the trajectory HMM can be viewed as the SPAM model because both mean vector and covariance matrix is also constrained within a linear subspace.

5.3.3 Relation to product of Gaussian experts

Recently, there has been interest in the use of classifiers based on the product of experts (PoE) framework [96]. It offers an alternative to the mixture of experts (MoE) framework for combining multiple probabilistic models. This is an efficient way to model high-dimensional data which simultaneously satisfies many different low-dimensional constraints because each individual experts can focus on giving high probability to data vectors that satisfy just one of the constraints.

One interesting relationship between the basis superposition framework and product of Gaussians (PoG) framework [97, 98] has been shown [95]. They have indicated that basis superposition could be viewed as an example of a PoG system. Because the trajectory HMM can be viewed as an example of the basis superposition over time, it can also be viewed as the PoG system. According to Eq. (5.24), the output probability of $\boldsymbol{c}$ conditioned on $\boldsymbol{q}$ can be rewritten as

$$p\left(\boldsymbol{c} \mid \boldsymbol{q}, \Lambda\right) = \mathcal{N}\left(\boldsymbol{c} \;\middle|\; \bar{\boldsymbol{c}}_{\boldsymbol{q}}, \boldsymbol{P}_{\boldsymbol{q}}\right) = \frac{1}{Z_{\boldsymbol{q}}} \prod_{t=1}^{T} \mathcal{N}\left(\boldsymbol{o}_t \;\middle|\; \boldsymbol{\mu}_{q_t s_t}, \boldsymbol{\Sigma}_{q_t s_t}\right). \tag{5.37}$$

The above equation shows that each static feature vector is augmented by their dynamic features and modeled by a Gaussian expert. Then, these experts are producted over time and normalized to yield a valid probability distribution. Similar discussions can be found in [89, 99].

5.4 Summary

This chapter has derived a new statistical model by reformulating the HMM whose state output vector includes both static and dynamic features. By imposing the explicit relationships between static and dynamic features, the HMM has been translated into a trajectory model, referred to as trajectory HMM in the present dissertation. Relationships between the trajectory HMM and other techniques were also discussed. The next chapter will derive training algorithms for the trajectory HMM.

Chapter 6

Training algorithm for trajectory HMM

In the previous chapter, the trajectory HMM was derived from the HMM whose state output vector includes both static and dynamic features. In this chapter, training algorithms for the trajectory HMM are derived. It should be noted that although the model has the same parameterization as the HMM, the output probability is defined by Eq. (5.30) rather than Eq. (5.1). Accordingly, the model parameters should be trained based on Eq. (5.30).

6.1 $\mathcal{Q}$-function

The maximum likelihood criterion is used to estimate trajectory HMM parameters. Although it is also possible to find a discriminative training scheme such as the minimum classification error (MCE) or the maximum mutual information (MMI), for this initial work only the ML training is considered. In common with the HMM training, the EM algorithm may be used for estimating parameters of the trajectory HMM based on the ML criterion. An auxiliary function of the current parameter set Λ and the new parameter set $\hat{\Lambda}$ is defined as follows:

$$\mathcal{Q}(\Lambda, \hat{\Lambda}) = \sum_{\text{all } \boldsymbol{q}} p\left(\boldsymbol{q} \mid \boldsymbol{c}, \Lambda\right) \; \log p(\boldsymbol{c}, \boldsymbol{q} \mid \hat{\Lambda}), \tag{6.1}$$

The logarithm of $p\left(\boldsymbol{c} \mid \boldsymbol{q}, \Lambda\right)$ can be written as

$$\log p\left(\boldsymbol{c}, \boldsymbol{q} \mid \Lambda\right) = \log p\left(\boldsymbol{c} \mid \boldsymbol{q}, \Lambda\right) + \log P\left(\boldsymbol{q} \mid \Lambda\right), \tag{6.2}$$

$$\log p\left(\boldsymbol{c} \mid \boldsymbol{q}, \Lambda\right) = \log \mathcal{N}\left(\boldsymbol{c} \;\middle|\; \bar{\boldsymbol{c}}_{\boldsymbol{q}}, \boldsymbol{P}_{\boldsymbol{q}}\right) \tag{6.3}$$

$$= -\frac{1}{2}\left\{MT \log(2\pi) + \log\left|\boldsymbol{P}_{\boldsymbol{q}}\right| + \left(\boldsymbol{c} - \bar{\boldsymbol{c}}_{\boldsymbol{q}}\right)^\top \boldsymbol{P}_{\boldsymbol{q}}^{-1}\left(\boldsymbol{c} - \bar{\boldsymbol{c}}_{\boldsymbol{q}}\right)\right\} \tag{6.4}$$

$$= -\frac{1}{2}\left\{MT \log(2\pi) - \log\left|\boldsymbol{R}_{\boldsymbol{q}}\right| + \boldsymbol{c}^\top \boldsymbol{R}_{\boldsymbol{q}} \boldsymbol{c} + \boldsymbol{r}_{\boldsymbol{q}}^\top \boldsymbol{P}_{\boldsymbol{q}} \boldsymbol{r}_{\boldsymbol{q}} - 2\boldsymbol{r}_{\boldsymbol{q}}^\top \boldsymbol{c}\right\}, \tag{6.5}$$

$$\log P\left(\boldsymbol{q} \mid \Lambda\right) = \log \pi_{q_1} + \sum_{t=2}^{T} \log a_{q_{t-1}q_t} + \sum_{t=1}^{T} \log w_{q_t s_t}. \tag{6.6}$$

Hence, the auxiliary function can be written as

$$\begin{aligned} \mathcal{Q}(\Lambda, \hat{\Lambda}) = \sum_{\text{all } \boldsymbol{q}} p\left(\boldsymbol{q} \mid \boldsymbol{c}, \Lambda\right) \cdot \Bigg[& \log \pi_{q_1} + \sum_{t=2}^{T} \log a_{q_{t-1}q_t} + \sum_{t=1}^{T} \log w_{q_t s_t} \\ & - \frac{1}{2}\left\{KT \log(2\pi) - \log\left|\boldsymbol{R}_{\boldsymbol{q}}\right| + \boldsymbol{c}^\top \boldsymbol{R}_{\boldsymbol{q}} \boldsymbol{c} + \boldsymbol{r}_{\boldsymbol{q}}^\top \boldsymbol{P}_{\boldsymbol{q}} \boldsymbol{r}_{\boldsymbol{q}} - 2\boldsymbol{r}_{\boldsymbol{q}}^\top \boldsymbol{c}\right\} \Bigg], \end{aligned} \tag{6.7}$$

$$p\left(\boldsymbol{q} \mid \boldsymbol{c}, \Lambda\right) = \frac{p\left(\boldsymbol{c}, \boldsymbol{q} \mid \Lambda\right)}{\sum_{\text{all } \boldsymbol{q}} p\left(\boldsymbol{c}, \boldsymbol{q} \mid \Lambda\right)}. \tag{6.8}$$

6.2 Maximization of $\mathcal{Q}$-function

First, N is redefined as the number of independent substates (Gaussian components) in the whole model set and unique indexes are assigned for these independent substates.[1] Then, a supervector $\boldsymbol{m}$ and supermatrix $\boldsymbol{\phi}$ are defined by concatenating the mean vectors and covariance matrices of all independent substates, that is

$$\boldsymbol{m} = \left[\boldsymbol{\mu}_1^\top, \boldsymbol{\mu}_2^\top, \dots, \boldsymbol{\mu}_N^\top\right]^\top, \tag{6.9}$$

$$\boldsymbol{\phi} = \left[\boldsymbol{\Sigma}_1^{-1}, \boldsymbol{\Sigma}_2^{-1}, \dots, \boldsymbol{\Sigma}_N^{-1}\right]^\top, \tag{6.10}$$

where $\boldsymbol{\mu}_n$ and $\boldsymbol{\Sigma}_n$ are the mean vector and the covariance matrix of the n-th independent Gaussian component, respectively.

[1] We assume that both mean vectors and covariance matrices have the same tying structure.

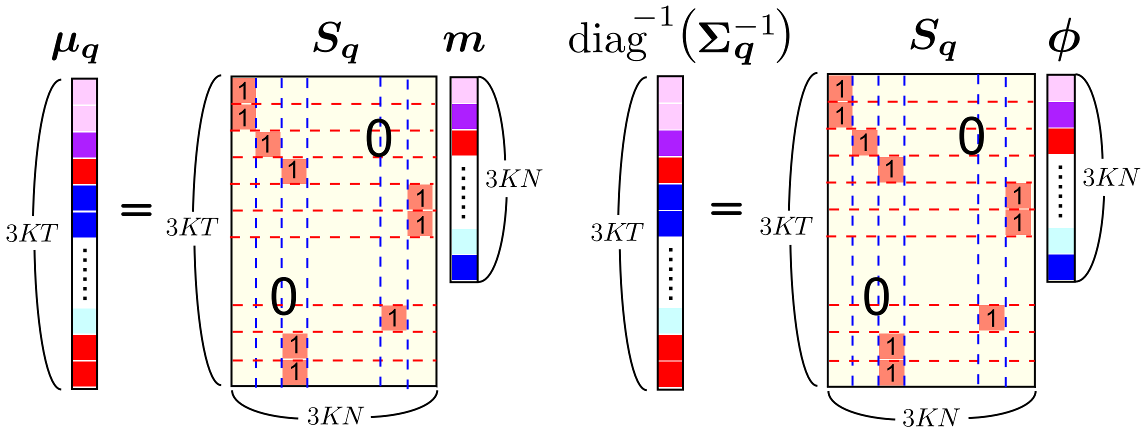


Figure 6.1: Relationships between $\boldsymbol{\mu}_{\boldsymbol{q}}$ and $\boldsymbol{m}$, and $\boldsymbol{\Sigma}_{\boldsymbol{q}}$ and $\boldsymbol{\phi}$ in matrix form.

Secondly, we define a $3KT \times 3KN$ matrix $\boldsymbol{S}_{\boldsymbol{q}}$ which represents substate sequence $\boldsymbol{q}$, that is

$$\boldsymbol{S}_{\boldsymbol{q}} = \left[\boldsymbol{s}_{(q_1,s_1)}, \boldsymbol{s}_{(q_2,s_2)}, \dots, \boldsymbol{s}_{q_t s_t}\right]^\top \otimes \boldsymbol{I}_{3K\times 3K}, \tag{6.11}$$

$$\boldsymbol{s}_{q_t s_t} = \left[f(q_t, s_t, 1), f(q_t, s_t), \dots, f(q_t, s_t, N)\right]^\top, \tag{6.12}$$

$$f(q_t, s_t, i) = \begin{cases} 1 & \text{index of the } s_t\text{-th substate of the } q_t\text{-th state is equal to } i \\ 0 & \text{otherwise} \end{cases}, \tag{6.13}$$

where $\boldsymbol{I}_{3K\times 3K}$ is the $3K \times 3K$ identity matrix. By using the substate sequence matrix $\boldsymbol{S}_{\boldsymbol{q}}$, relationships between $\boldsymbol{\mu}_{\boldsymbol{q}}$ and $\boldsymbol{m}$, and $\boldsymbol{\Sigma}_{\boldsymbol{q}}$ and $\boldsymbol{\phi}$ illustrated in Fig. 6.1 can be represented as

$$\boldsymbol{\mu}_{\boldsymbol{q}} = \boldsymbol{S}_{\boldsymbol{q}}\boldsymbol{m}, \tag{6.14}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1} = \mathrm{diag}\left(\boldsymbol{S}_{\boldsymbol{q}}\boldsymbol{\phi}\right). \tag{6.15}$$

Thus, Eqs. (5.27) and (5.28) can be rewritten using $\boldsymbol{m}$ and $\boldsymbol{\phi}$ as

$$\boldsymbol{R}_{\boldsymbol{q}} = \boldsymbol{W}^\top \cdot \mathrm{diag}(\boldsymbol{S}_{\boldsymbol{q}}\boldsymbol{\phi}) \cdot \boldsymbol{W}, \tag{6.16}$$

$$\boldsymbol{r}_{\boldsymbol{q}} = \boldsymbol{W}^\top \cdot \mathrm{diag}(\boldsymbol{S}_{\boldsymbol{q}}\boldsymbol{\phi}) \cdot \boldsymbol{S}_{\boldsymbol{q}}\boldsymbol{m}. \tag{6.17}$$

After substituting Eqs. (6.16)–(6.17) for Eq. (6.7), a partial derivative of Eq. (6.7) with respect to $\boldsymbol{m}$ is given by

$$\frac{\partial Q(\Lambda, \hat{\Lambda})}{\partial \boldsymbol{m}} = \sum_{\text{all } \boldsymbol{q}} p\left(\boldsymbol{q} \mid \boldsymbol{c}, \Lambda\right) \cdot \left\{ \boldsymbol{S}_{\boldsymbol{q}}^\top \boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1} \boldsymbol{W} \left(\boldsymbol{c} - \bar{\boldsymbol{c}}_{\boldsymbol{q}}\right) \right\}. \tag{6.18}$$

By setting Eq. (6.18) equal to $\boldsymbol{0}_{3KN}$, a set of linear equations for determination of $\boldsymbol{m}$ maximizing Eq. (6.7) are obtained as

$$\sum_{\text{all } \boldsymbol{q}} p\left(\boldsymbol{q} \mid \boldsymbol{c}, \Lambda\right) \cdot \left\{ \boldsymbol{S}_{\boldsymbol{q}}^\top \boldsymbol{W} \boldsymbol{P}_{\boldsymbol{q}} \boldsymbol{W}^\top \boldsymbol{S}_{\boldsymbol{q}} \right\} \boldsymbol{\Phi}^{-1}\boldsymbol{m} = \sum_{\text{all } \boldsymbol{q}} p\left(\boldsymbol{q} \mid \boldsymbol{c}, \Lambda\right) \cdot \left\{ \boldsymbol{S}_{\boldsymbol{q}}^\top \boldsymbol{W}\boldsymbol{c} \right\}, \tag{6.19}$$

where

$$\boldsymbol{\Phi}^{-1} = \operatorname{diag}(\boldsymbol{\phi})\,. \tag{6.20}$$

The dimensionality of Eq. (6.19) is $3KN \times 3KN$: it could be millions when context-dependent acoustic models with mixtures of Gaussian components are used. It is difficult to store and solve such a huge set of linear equations using currently available computational resources. For such case, $\boldsymbol{m}$ can be optimized by a gradient method using Eq. (6.18). Please refer Appendix B for detail.

For maximizing Eq. (6.7) with respect to $\boldsymbol{\phi}$, a gradient method is applied using its first derivative

$$\begin{aligned}\frac{\partial Q(\Lambda, \hat{\Lambda})}{\partial \boldsymbol{\phi}} = \sum_{\text{all } \boldsymbol{q}} p(\boldsymbol{q} \mid \boldsymbol{c}, \Lambda) \cdot \Bigg\{ \frac{1}{2}\, \boldsymbol{S}_{\boldsymbol{q}}^{\top} \operatorname{diag}^{-1} \Big(& \boldsymbol{W}\boldsymbol{P}_{\boldsymbol{q}}\boldsymbol{W}^{\top} - \boldsymbol{W}\boldsymbol{c}\boldsymbol{c}^{\top}\boldsymbol{W}^{\top} + 2\boldsymbol{\mu}_{\boldsymbol{q}}\boldsymbol{c}^{\top}\boldsymbol{W}^{\top} \\ & + \boldsymbol{W}\overline{\boldsymbol{c}}_{\boldsymbol{q}}\overline{\boldsymbol{c}}_{\boldsymbol{q}}^{\top}\boldsymbol{W}^{\top} - 2\boldsymbol{\mu}_{\boldsymbol{q}}\overline{\boldsymbol{c}}_{\boldsymbol{q}}^{\top}\boldsymbol{W}^{\top} \Big) \Bigg\},\end{aligned} \tag{6.21}$$

because Eq. (6.21) is not a quadratic function of $\boldsymbol{\phi}$.

By using Eqs. (6.19) and (6.21), the model parameters which maximizes the likelihood can be estimated. However, the exact EM algorithm for the trajectory HMM is intractable. Since joint probability of the static feature vector sequence $\boldsymbol{c}$ and the substate sequence $\boldsymbol{q}$ depends on the entire $\boldsymbol{q}$, marginalization becomes prohibitively expensive. Exact computation of the total output probability (Eq. (5.24)) or the posterior probability of the substate sequence given the static feature vector sequence (Eq. (6.8)) has to be carried out over $O\left(N^{T}\right)$ paths.

6.3 Approximate training algorithms

To estimate such an intractable model, a number of approximate training techniques have been proposed. In the present dissertation, a Viterbi approximation and a Markov Chain Monte Carlo method are employed for estimating parameters of trajectory HMM.

6.3.1 Viterbi approximation

The Viterbi approximation keeps only the path with the highest log-likelihood active. Unfortunately, since the output probability of static feature vector sequence depends on the entire substate sequence, the Viterbi algorithm [53] is not admissible. To avoid this problem, a delayed decision Viterbi algorithm for the trajectory HMM will be used. It can find a better sub-optimal substate sequence for given static feature vector sequence. The detail of the delayed decision Viterbi algorithm will be described later in Chapter 7.

Under the Viterbi approximation, the auxiliary function can be expressed as follows:

$$\mathcal{Q}(\Lambda, \hat{\Lambda}) \approx \log p(\boldsymbol{c}, \hat{\boldsymbol{q}} \mid \hat{\Lambda}). \tag{6.22}$$

where

$$\hat{\boldsymbol{q}} = \arg\max_{\boldsymbol{q}} p\left(\boldsymbol{c}, \boldsymbol{q} \mid \Lambda\right). \tag{6.23}$$

Hence summations in Eqs. (6.18)–(6.21) can be ignored and they can be rewritten as

$$\frac{\partial Q(\Lambda, \hat{\Lambda})}{\partial \boldsymbol{m}} \approx \boldsymbol{S}_{\boldsymbol{q}}^{\top} \boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1} \boldsymbol{W} \left(\boldsymbol{c} - \bar{\boldsymbol{c}}_{\boldsymbol{q}} \right), \tag{6.24}$$

$$\frac{\partial Q(\Lambda, \hat{\Lambda})}{\partial \boldsymbol{\phi}} \approx \frac{1}{2} \boldsymbol{S}_{\boldsymbol{q}}^{\top} \operatorname{diag}^{-1} \left(\boldsymbol{W} \boldsymbol{P}_{\boldsymbol{q}} \boldsymbol{W}^{\top} - \boldsymbol{W} \boldsymbol{c} \boldsymbol{c}^{\top} \boldsymbol{W}^{\top} + 2\boldsymbol{\mu}_{\boldsymbol{q}} \boldsymbol{c}^{\top} \boldsymbol{W}^{\top} + \boldsymbol{W} \bar{\boldsymbol{c}}_{\boldsymbol{q}} \bar{\boldsymbol{c}}_{\boldsymbol{q}}^{\top} \boldsymbol{W}^{\top} - 2\boldsymbol{\mu}_{\boldsymbol{q}} \bar{\boldsymbol{c}}_{\boldsymbol{q}}^{\top} \boldsymbol{W}^{\top} \right). \tag{6.25}$$

As a result, a set of linear equations for determination of $\boldsymbol{m}$ under the Viterbi approximation are obtained as

$$\boldsymbol{S}_{\boldsymbol{q}}^{\top} \boldsymbol{W} \boldsymbol{P}_{\boldsymbol{q}} \boldsymbol{W}^{\top} \boldsymbol{S}_{\boldsymbol{q}} \boldsymbol{\Phi}^{-1} \boldsymbol{m} = \boldsymbol{S}_{\boldsymbol{q}}^{\top} \boldsymbol{W} \boldsymbol{c}. \tag{6.26}$$

6.3.2 Markov chain Monte Carlo

It is generally considered that the Viterbi approximation might be too strict for modeling spontaneous speech because phone boundaries are ambiguous in spontaneous speech. Furthermore, this approach is unable to marginalize model parameters over hidden variables (substate sequences). This may degrade model robustness and recognition performance. To relax this approximation, the auxiliary function is approximated over small number of substate sequences sampled by the Markov chain Monte Carlo (MCMC) algorithm. This case is known to as the Monte Carlo EM (MCEM) [100].

The auxiliary function of the trajectory HMM for the EM algorithm can be approximated by summation over V substate sequences as follows:

$$\boldsymbol{q}^{(v)} \sim P\left(\boldsymbol{q} \mid \boldsymbol{c}, \Lambda\right) \quad 1 \leq v \leq V, \tag{6.27}$$

$$\mathcal{Q}(\Lambda, \hat{\Lambda}) \approx \sum_{v=1}^{V} \frac{1}{V} \log p(\boldsymbol{c}, \boldsymbol{q}^{(v)} \mid \hat{\Lambda}), \tag{6.28}$$

where $\boldsymbol{q}^{(v)}$ is a v-th substate sequence sampled by the MCMC[2].

[2] In the case of Markov chain Monte Carlo, the ~ symbol is used to indicate that the sample on the left hand side was drawn from the density function on the right hand side.

There are several instances of the MCMC algorithms, in the present dissertation the Gibbs sampler is employed. The Gibbs sampling algorithm for the trajectory HMM can be summarized as follows:

1. Initialization
$$\boldsymbol{q}^{(1)} = \left\{(q_1^{(1)}, s_1^{(1)}), \dots, (q_T^{(1)}, s_T^{(1)})\right\} \tag{6.29}$$
2. For iteration $v = 2, \dots, V$ draw a substate
$$(q_t^{(v)}, s_t^{(v)}) \sim p\left((q_t, s_t) \mid \boldsymbol{c}, \boldsymbol{q}_{-t}^{(v)}, \Lambda\right), \quad t = 1, \dots, T \tag{6.30}$$
where
$$\boldsymbol{q}_{-t}^{(v)} = \left\{(q_1^{(v)}, s_1^{(v)}), \dots, (q_{t-1}^{(v)}, s_{t-1}^{(v)}), (q_{t+1}^{(v-1)}, s_{t+1}^{(v-1)}), \dots, (q_T^{(v-1)}, s_T^{(v-1)})\right\}. \tag{6.31}$$

However, exact computation of $P\left((q_t, s_t) \mid \boldsymbol{c}, \boldsymbol{q}_{-t}^{(v)}, \Lambda\right)$ is still expensive. Thus, we introduce additional approximation for dependency between static feature vector sequence and substate sequence. Instead of using exact posterior probability distribution, (q_t, s_t) is sampled from posterior probability distribution depending only on past, current and further J observations and substates. The sampling algorithm is rewritten as follows:

1. Initialization
$$\boldsymbol{q}^{(1)} = \left\{(q_1^{(1)}, s_1^{(1)}), \dots, (q_T^{(1)}, s_T^{(1)})\right\} \tag{6.32}$$
2. For iteration $v = 2, \dots, V$ draw a substate
$$(q_t^{(v)}, s_t^{(v)}) \sim p\left((q_t, s_t) \mid \boldsymbol{c}, \boldsymbol{q}_{-t,J}^{(v)}, \Lambda\right), \quad t = 1, \dots, T \tag{6.33}$$
where
$$\boldsymbol{q}_{-t,J}^{(v)} = \left\{(q_1^{(v)}, s_1^{(v)}), \dots, (q_{t-1}^{(v)}, s_{t-1}^{(v)}), (q_{t+1}^{(v-1)}, s_{t+1}^{(v-1)}), \dots, (q_{t+J}^{(v-1)}, s_{t+J}^{(v-1)})\right\}. \tag{6.34}$$

The partial derivative of Eq. (6.28) with respect to $\boldsymbol{m}$ and $\boldsymbol{\phi}$ is given as

$$\frac{\partial Q(\Lambda, \hat{\Lambda})}{\partial \boldsymbol{m}} \approx \sum_{v=1}^{V} \frac{1}{V} \cdot \left\{ \boldsymbol{S}_{\boldsymbol{q}^{(v)}}^{\top} \boldsymbol{\Sigma}_{\boldsymbol{q}^{(v)}}^{-1} \boldsymbol{W} \left(\boldsymbol{c} - \bar{\boldsymbol{c}}_{\boldsymbol{q}^{(v)}}\right) \right\}, \tag{6.35}$$

$$\begin{aligned}\frac{\partial Q(\Lambda, \hat{\Lambda})}{\partial \boldsymbol{\phi}} \approx \sum_{v=1}^{V} \frac{1}{V} \cdot \Bigg\{ \frac{1}{2} \boldsymbol{S}_{\boldsymbol{q}^{(v)}}^{\top} \operatorname{diag}^{-1} \Big(&\boldsymbol{W}\boldsymbol{P}_{\boldsymbol{q}^{(v)}}\boldsymbol{W}^{\top} - \boldsymbol{W}\boldsymbol{c}\boldsymbol{c}^{\top}\boldsymbol{W}^{\top} + 2\boldsymbol{\mu}_{\boldsymbol{q}^{(v)}}\boldsymbol{c}^{\top}\boldsymbol{W}^{\top} \\ &+ \boldsymbol{W}\bar{\boldsymbol{c}}_{\boldsymbol{q}^{(v)}}\bar{\boldsymbol{c}}_{\boldsymbol{q}^{(v)}}^{\top}\boldsymbol{W}^{\top} - 2\boldsymbol{\mu}_{\boldsymbol{q}^{(v)}}\bar{\boldsymbol{c}}_{\boldsymbol{q}^{(v)}}^{\top}\boldsymbol{W}^{\top}\Big) \Bigg\}.\end{aligned} \tag{6.36}$$

As a result, a set of linear equations for determination of $\boldsymbol{m}$ using the MCMC algorithm are obtained as

$$\sum_{v=1}^{V} \boldsymbol{S}_{\boldsymbol{q}^{(v)}}^{\top} \boldsymbol{W}\boldsymbol{P}_{\boldsymbol{q}^{(v)}}\boldsymbol{W}^{\top}\boldsymbol{S}_{\boldsymbol{q}^{(v)}}\boldsymbol{\Phi}^{-1}\boldsymbol{m} = \sum_{v=1}^{V} \boldsymbol{S}_{\boldsymbol{q}^{(v)}}^{\top}\boldsymbol{W}\boldsymbol{c}. \tag{6.37}$$

6.4 Summary

In this chapter, the training algorithms for the trajectory HMM based on the Viterbi approximation and Markov chain Monte Carlo method have been derived. Although the trajectory HMM has the same parameterization as the HMM, the definition of the output probability is different. Accordingly, the training algorithms should be re-derived for the trajectory HMM. In the next chapter, the delayed decision Viterbi algorithm based on a time-recursive likelihood calculation will be described. It can find a better sub-optimum substate sequence for given static feature vector sequence.

Chapter 7

Search algorithm for trajectory HMM

In the previous chapter, training algorithms for the trajectory HMM based on the Viterbi and the Markov chain Monte Carlo approximations were described. This chapter discusses the maximization problem of Eq. (6.23). Solution of this problem is essential because it is used not only in the training part but also the decoding part of the system.

In the trajectory HMM, inter-frame correlations can be captured because the inter-frame covariance matrix $\boldsymbol{P_q}$ is generally full. However, this property makes it intractable to find the most likely substate sequence using the Viterbi algorithm [53]. Based on the approximation

$$\begin{aligned}
\boldsymbol{q}_{\max} &= \arg\max_{\boldsymbol{q}} p\left(\boldsymbol{c}, \boldsymbol{q} \mid \Lambda\right) && (7.1)\\
&= \arg\max_{\boldsymbol{q}} p\left(\boldsymbol{c} \mid \boldsymbol{q}, \Lambda\right) \cdot P\left(\boldsymbol{q} \mid \Lambda\right) && (7.2)\\
&= \arg\max_{\boldsymbol{q}} \frac{1}{Z_{\boldsymbol{q}}} \mathcal{N}\left(\boldsymbol{o} \;\middle|\; \boldsymbol{\mu_q}, \boldsymbol{\Sigma_q}\right) \cdot P\left(\boldsymbol{q} \mid \Lambda\right) && (7.3)\\
&\approx \arg\max_{\boldsymbol{q}} \mathcal{N}\left(\boldsymbol{o} \mid \boldsymbol{\mu_q}, \boldsymbol{\Sigma_q}\right) \cdot P\left(\boldsymbol{q} \mid \Lambda\right), && (7.4)
\end{aligned}$$

the Viterbi algorithm can be used. However, this approximation reduces the accuracy of the alignment.

To avoid this problem, a delayed decision Viterbi algorithm based on a time-recursive likelihood calculation is derived. This algorithm can find a better sub-optimum Gaussian component sequence.

7.1 Time recursive output probability calculation

Before describing the delayed decision Viterbi algorithm, a time-recursive algorithm to calculate $p\left(\boldsymbol{c} \mid \boldsymbol{q}, \Lambda\right)$, which is used in the delayed decision Viterbi algorithm, is derived.

According to Eq. (5.24), the joint probability of the static feature vector sequence and substate sequence is given as

$$p\left(\boldsymbol{c}, \boldsymbol{q} \mid \Lambda\right) = p\left(\boldsymbol{c} \mid \boldsymbol{q}, \Lambda\right) P\left(\boldsymbol{q} \mid \Lambda\right) \tag{7.5}$$

$$p\left(\boldsymbol{c} \mid \boldsymbol{q}, \Lambda\right) = \frac{1}{Z_{\boldsymbol{q}}} \mathcal{N}\left(\boldsymbol{W}\boldsymbol{c} \mid \boldsymbol{q}, \Lambda\right), \tag{7.6}$$

$$Z_{\boldsymbol{q}} = \frac{\sqrt{(2\pi)^{KT} \left|\boldsymbol{P}_{\boldsymbol{q}}\right|}}{\sqrt{(2\pi)^{3KT} \left|\boldsymbol{\Sigma}_{\boldsymbol{q}}\right|}} \cdot \exp\left\{-\frac{1}{2}\left(\boldsymbol{\mu}_{\boldsymbol{q}}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1}\boldsymbol{\mu}_{\boldsymbol{q}} - \boldsymbol{r}_{\boldsymbol{q}}^{\top}\boldsymbol{P}_{\boldsymbol{q}}\boldsymbol{r}_{\boldsymbol{q}}\right)\right\}. \tag{7.7}$$

Equations (7.6)–(7.7) show that calculating $\mathcal{N}\left(\boldsymbol{W}\boldsymbol{c} \mid \boldsymbol{\mu}_{\boldsymbol{q}}, \boldsymbol{\Sigma}_{\boldsymbol{q}}\right)$, $\left|\boldsymbol{\Sigma}_{\boldsymbol{q}}\right|$, $\boldsymbol{\mu}_{\boldsymbol{q}}^{\top}\boldsymbol{\Sigma}_{q}\boldsymbol{\mu}_{\boldsymbol{q}}$, $\left|\boldsymbol{P}_{\boldsymbol{q}}\right|$, and $\boldsymbol{r}_{\boldsymbol{q}}^{\top}\boldsymbol{P}_{\boldsymbol{q}}\boldsymbol{r}_{\boldsymbol{q}}$ are required to obtain the output probability of $\boldsymbol{c}$ conditioned on $\boldsymbol{q}$.

Since $\boldsymbol{\Sigma}_{\boldsymbol{q}}$ is diagonal, $\mathcal{N}\left(\boldsymbol{W}\boldsymbol{c} \mid \boldsymbol{\mu}_{\boldsymbol{q}}, \boldsymbol{\Sigma}_{\boldsymbol{q}}\right)$, $\left|\boldsymbol{\Sigma}_{\boldsymbol{q}}\right|$ and $\boldsymbol{\mu}_{\boldsymbol{q}}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{q}}\boldsymbol{\mu}_{\boldsymbol{q}}$ can be rewritten as follows:

$$\mathcal{N}\left(\boldsymbol{W}\boldsymbol{c} \mid \boldsymbol{\mu}_{\boldsymbol{q}}, \boldsymbol{\Sigma}_{\boldsymbol{q}}\right) = \prod_{t=1}^{T} \mathcal{N}\left(\boldsymbol{o}_t \mid \boldsymbol{\mu}_{q_t s_t}, \boldsymbol{\Sigma}_{q_t s_t}\right), \tag{7.8}$$

$$\left|\boldsymbol{\Sigma}_{\boldsymbol{q}}\right| = \prod_{t=1}^{T} \left|\boldsymbol{\Sigma}_{q_t s_t}\right|, \tag{7.9}$$

$$\boldsymbol{\mu}_{\boldsymbol{q}}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1}\boldsymbol{\mu}_{\boldsymbol{q}} = \prod_{t=1}^{T} \boldsymbol{\mu}_{q_t s_t}^{\top}\boldsymbol{\Sigma}_{q_t s_t}^{-1}\boldsymbol{\mu}_{q_t s_t}. \tag{7.10}$$

However, decomposing $\left|\boldsymbol{P}_{\boldsymbol{q}}\right|$ and $\boldsymbol{r}_{\boldsymbol{q}}^{\top}\boldsymbol{P}_{\boldsymbol{q}}\boldsymbol{r}_{\boldsymbol{q}}$ in the same manner is unable because $\boldsymbol{P}_{\boldsymbol{q}}$ is generally full. Fortunately, by utilizing the special structure of $\boldsymbol{P}_{\boldsymbol{q}}$, both $\boldsymbol{r}_{\boldsymbol{q}}^{\top}\boldsymbol{P}_{\boldsymbol{q}}\boldsymbol{r}_{\boldsymbol{q}}$ and $\left|\boldsymbol{P}_{\boldsymbol{q}}\right|$ can be computed in a recursive manner.

For notation simplicity, the dimensionality of the static feature vector, K, is assumed to be 1. The precision matrix $\boldsymbol{R}_{\boldsymbol{q}}$ becomes a positive definite $(4KL + 1)$-diagonal band symmetric matrix and it can be decomposed into its Cholesky factorization:

$$\boldsymbol{R}_{\boldsymbol{q}} = \boldsymbol{U}_{\boldsymbol{q}}^{\top}\boldsymbol{U}_{\boldsymbol{q}}, \tag{7.11}$$

where $\boldsymbol{U}_{\boldsymbol{q}}$ is a upper-triangular $(2KL + 1)$-diagonal matrix. Elements of $\boldsymbol{U}_{\boldsymbol{q}}$ is calculated in a recursive manner as

$$U_{\boldsymbol{q}}^{(t,t)} = \sqrt{R_{\boldsymbol{q}}^{(t,t)} - \sum_{i=t-2L}^{t-1} U_{\boldsymbol{q}}^{(i,t)\,2}}, \qquad t = 1, \dots, T \tag{7.12}$$

$$U_{\boldsymbol{q}}^{(t,i)} = \frac{\sqrt{R_{\boldsymbol{q}}^{(t,i)} - \sum_{j=t-2L}^{t-1} U_{\boldsymbol{q}}^{(j,t)} \cdot U_{\boldsymbol{q}}^{(j,i)}}}{U_{\boldsymbol{q}}^{(t,t)}}, \qquad \begin{array}{l} t = 1, \dots, T \\ t+1 \leq i \leq t+2L \end{array} \tag{7.13}$$

$$R_{\boldsymbol{q}}^{(t,i)} = R_{\boldsymbol{q}}^{(i,t)} = \sum_{d=0}^{2} \sum_{\tau=-L_-^{(d)}}^{L_+^{(d)}} \frac{1}{\Delta^{(d)} \sigma_{q_{t+\tau} s_{t+\tau}}(1)} \, w^{(d)}(-\tau) \cdot w^{(d)}(i-t-\tau), \qquad \begin{array}{l} t = 1, \dots, T \\ t \leq i \leq t+2L \end{array} \tag{7.14}$$

where $U_{\boldsymbol{q}}^{(i,j)}$ and $R_{\boldsymbol{q}}^{(i,j)}$ are the (i, j)-th elements of $\boldsymbol{U_q}$ and $\boldsymbol{R_q}$, respectively.

Thus, $\left|\boldsymbol{P_q}\right|$ can be written as

$$\begin{aligned} \left|\boldsymbol{P_q}\right| &= \left|\boldsymbol{R_q}\right|^{-1} && (7.15) \\ &= \left|\boldsymbol{U_q}^\top \boldsymbol{U_q}\right|^{-1} && (7.16) \\ &= \left|\boldsymbol{U_q}\right|^{-2}. && (7.17) \end{aligned}$$

The determinant of a triangular matrix is simply the product of its diagonal elements. Hence, $\left|\boldsymbol{P_q}\right|$ is given by

$$\left|\boldsymbol{P_q}\right| = \prod_{t=1}^{T} \left|U_{\boldsymbol{q}}^{(t,t)}\right|^{-2}. \tag{7.18}$$

Since $U_{\boldsymbol{q}}^{(t,t)}$ are computed in a recursive manner and depends only on the substates from time 1 to $t + L$, $\left|\boldsymbol{P_q}\right|$ can be computed recursively.

Next we definite a vector $\boldsymbol{g_q}$ which is given by solving following a set of linear equations:

$$\begin{aligned} \boldsymbol{U_q}^\top \boldsymbol{g_q} &= \boldsymbol{r_q}, && (7.19) \\ \boldsymbol{U_q} \overline{\boldsymbol{c}}_{\boldsymbol{q}} &= \boldsymbol{g_q}, && (7.20) \end{aligned}$$

From Eqs. (7.19)–(7.20), $\boldsymbol{r_q}^\top \boldsymbol{P_q} \boldsymbol{r_q}$ can be rewritten as

$$\begin{aligned} \boldsymbol{r_q}^\top \boldsymbol{P_q} \boldsymbol{r_q} &= \boldsymbol{r_q}^\top \boldsymbol{P_q}^\top \boldsymbol{R_q} \boldsymbol{P_q} \boldsymbol{r_q} && (7.21) \\ &= \overline{\boldsymbol{c}}_{\boldsymbol{q}}^\top \boldsymbol{U_q}^\top \boldsymbol{U_q} \overline{\boldsymbol{c}}_{\boldsymbol{q}} && (7.22) \\ &= \boldsymbol{g_q}^\top \boldsymbol{g_q} && (7.23) \\ &= \sum_{t=1}^{T} g_{\boldsymbol{q}}^{(t)\,2}, && (7.24) \end{aligned}$$

where $g_{\boldsymbol{q}}^{(t)}$ and $r_{\boldsymbol{q}}^{(t)}$ are the t-th elements of $\boldsymbol{g}_{\boldsymbol{q}}$ and $\boldsymbol{r}_{\boldsymbol{q}}$, respectively, given as

$$g_{\boldsymbol{q}}^{(t)} = \frac{r_{\boldsymbol{q}}^{(t)} - \sum_{i=t-2L}^{t-1} U_{\boldsymbol{q}}^{(i,t)} \cdot g_{\boldsymbol{q}}^{(i)}}{U_{\boldsymbol{q}}^{(t,t)}}, \quad t = 1, \ldots, T \tag{7.25}$$

$$r_{\boldsymbol{q}}^{(t)} = \sum_{d=0}^{2} \sum_{\tau=-L_{-}^{(d)}}^{L_{+}^{(d)}} \frac{1}{\Delta^{(d)}\sigma_{q_{t+\tau}}(1)} w^{(d)}(-\tau) \cdot \Delta^{(d)}\mu_{q_{t+\tau}}(1), \quad t = 1, \ldots, T \tag{7.26}$$

Since $g_{\boldsymbol{q}}^{(t)}$ is calculated recursively and depends only on the substates from time 1 to $t + L$, $\boldsymbol{r}_{\boldsymbol{q}}^{\top} \boldsymbol{P}_{\boldsymbol{q}} \boldsymbol{r}_{\boldsymbol{q}}$ can also be calculated in a recursive manner.

As a result, the output probability of $\boldsymbol{c}$ conditioned on $\boldsymbol{q}$ is given by

$$p(\boldsymbol{c} \mid \boldsymbol{q}, \Lambda) = \prod_{t=1}^{T} \frac{1}{Z_{\boldsymbol{q}_{t+L}}^{(t)}} \mathcal{N}\left(\boldsymbol{o}_t \mid \boldsymbol{\mu}_{q_t s_t}, \boldsymbol{\Sigma}_{q_t s_t}\right), \tag{7.27}$$

where $\boldsymbol{q}_{t+L} = \{(q_1, s_1), \ldots, (q_{t+L}, s_{t+L})\}$ is a partial substate sequence from time 1 to $t + L$, and $Z_{\boldsymbol{q}_{t+L}}^{(t)}$ is a decomposed normalization term at time t given by

$$Z_{\boldsymbol{q}_{t+L}}^{(t)} = \frac{\sqrt{(2\pi) \left|U_{\boldsymbol{q}_{t+L}}^{(t,t)}\right|^{-2}}}{\sqrt{(2\pi)^3 \left|\boldsymbol{\Sigma}_{q_t s_t}\right|}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{\mu}_{q_t s_t}^{\top} \boldsymbol{\Sigma}_{q_t s_t}^{-1} \boldsymbol{\mu}_{q_t s_t} - g_{\boldsymbol{q}_{t+L}}^{(t)\,2}\right)\right\}. \quad t = 1, \ldots, T \tag{7.28}$$

7.2 Delayed decision Viterbi algorithm

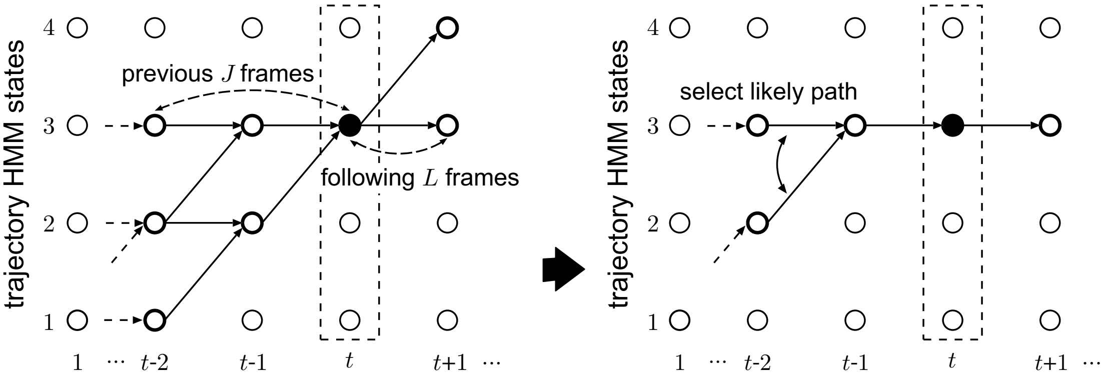
In this section, the delayed decision Viterbi algorithm for the trajectory HMM to find a better sub-optimum substate sequence is described. This algorithm is based on the time-recursive algorithm described in the previous section. In the delayed decision Viterbi algorithm, the substate at time $t - J$ is determined at time t, according to its partial joint probability.

The delayed decision Viterbi algorithm illustrated in Fig. 7.1 involves the following recursion:

1. Initialization: $t = 1$; $\forall \boldsymbol{q}_{1-J}^{1+L}$

$$\delta_1\left(\boldsymbol{q}_{1-J}^{1+L}\right) = \prod_{t=1}^{t+L} a_{q_{t-1}q_t} w_{q_t s_t} \cdot b\left(\boldsymbol{q}_{1+L}\right)$$

$$\psi_1\left(\boldsymbol{q}_{1-J}^{1+L}\right) = 0$$



For each node, compute all $\delta_t(\boldsymbol{q}_{t-J}^{t+L})$

For each $\boldsymbol{q}_{t-J}^{t+L}$, select likely path at time t-2

Figure 7.1: An overview of the proposed delayed decision Viterbi algorithm ($J = 2, L = 1$).

2. Recursion: $t = 2, \dots, T$; $\forall \boldsymbol{q}_{t-J}^{t+L}$

$$\delta_t\left(\boldsymbol{q}_{t-J}^{t+L}\right) = \max_{(q_{t-J-1}, s_{t-J-1})} \left[\delta_{t-1}\left(\boldsymbol{q}_{t-J-1}^{t+L-1}\right)\ a_{q_{t+L-1}q_{t+L}} w_{q_{t+L}s_{t+L}}\right] b(\boldsymbol{q}_{t+L})$$

$$\psi_t\left(\boldsymbol{q}_{t-J}^{t+L}\right) = \arg\max_{(q_{t-J-1}, s_{t-J-1})} \left[\delta_{t-1}\left(\boldsymbol{q}_{t-J-1}^{t+L-1}\right)\ a_{q_{t-+L1}q_{t+L}} w_{q_{t+L}s_{t+L}}\right]$$

3. Termination:

$$\hat{P} = \max_{\boldsymbol{q}_{T-J}^{T+L}} \left[\delta_T\left(\boldsymbol{q}_{T-J}^{T+L}\right)\right]$$

$$\hat{\boldsymbol{q}}_{T-J}^{T+L} = \arg\max_{\boldsymbol{q}_{T-J}^{T+L}} \left[\delta_T\left(\boldsymbol{q}_{T-J}^{T+L}\right)\right]$$

$$= \{(\hat{q}_{T-J}, \hat{s}_{T-J}) \dots, (\hat{q}_{T+L}, \hat{s}_{T+L})\}$$

4. Back tracking: $t = T, \dots, 1$

$$(\hat{q}_{t-J-1}, \hat{s}_{t-J-1}) = \psi_t\left(\hat{\boldsymbol{q}}_{t-J}^{t+L}\right)$$

$$\hat{\boldsymbol{q}}_{t-J-1}^{t+L-1} = \{(\hat{q}_{t-J-1}, \hat{s}_{t-J-1}), \dots, (\hat{q}_{t+L-1}, \hat{q}_{t+L-1})\}$$

$$\hat{\boldsymbol{q}} = \{(\hat{q}_1, \hat{s}_1), (\hat{q}_2, \hat{s}_2), \dots, (\hat{q}_T, \hat{s}_T)\}$$

where $\boldsymbol{q}_{t-J}^{t+L}$ is a partial substate sequence from time $t - J$ to $t + L$, and $b\left(\boldsymbol{q}_{t+L}\right)$ is given by

$$b(\boldsymbol{q}_{t+L}) = \frac{1}{Z_{\boldsymbol{q}_{t+L}}^{(t)}}\ \mathcal{N}\left(\boldsymbol{o}_t \ \middle|\ \boldsymbol{\mu}_{q_t s_t}, \boldsymbol{\Sigma}_{q_t s_t}\right). \tag{7.29}$$

The optimum substate sequence can be found if J is equal to T, because this is equivalent to evaluating all possible substate sequences. While this algorithm with longer delay J could obtain more likely substate sequence, it requires a huge amount of computations. Hence, the delay J should be set to a proper value balancing its performance and computational complexity. It is generally considered that the co-articulation affects neighboring frames within 100–200 ms. This indicates that the optimum substate sequence might be obtained when the number of delay J is set to around 10–20 for a 10-ms frame shift.

7.3 Decoding

For decoding, we can use i) rescoring scheme, or ii) decoder based on the delayed decision Viterbi algorithm. In approach i), each hypothesis generated by the HMMs is rescored by the Eq. (7.6). The substate sequence for each hypothesis can also be adjusted using the delayed decision Viterbi algorithm. On the other hand, in approach ii), the normalization term $Z_{\boldsymbol{q}}$ is calculated in the decoding procedure. However, implementing a decoder based on approach ii) is difficult. In the speech recognition experiment described in Chapter 8, approach i) was adopted.

7.4 Summary

In this chapter, the delayed decision Viterbi algorithm based on the time-recursive output probability calculation has been described. It can find a better sub-optimum substate sequence for given static feature vector sequence. The next chapter will evaluate performances of the trajectory HMM both in speech recognition and synthesis experiments.

Chapter 8

Experiments

In the previous two chapters, the training and search algorithms for the trajectory HMM have been derived. In this chapter, the performance of the trajectory HMM was evaluated both in speech recognition and synthesis experiments.

8.1 Simulation experiment

A simple simulation experiment was conducted to show that trajectory HMM was a proper generative model.

Phonetically balanced 503 sentences uttered by a male speaker MHT from the ATR Japanese speech database b-set [101] were used. The first 450 sentences were used for training. Speech analysis conditions are shown in Tab. 8.1. To simplify the experiment,

Table 8.1: Speech analysis conditions.

Sampling frequency	16kHz
Frame shift	10ms
Frame length	25.6ms
Window	Blackman window
Parameterization	18-th order Mel-cepstral analysis [65]
Feature vector	19 Mel-cesptral coefficients including zeroth coefficient, its delta and delta-delta $\Delta \boldsymbol{c}_t = -0.5\boldsymbol{c}_{t-1} + 0.5\boldsymbol{c}_{t+1}$ $\Delta^2 \boldsymbol{c}_t = \boldsymbol{c}_{t-1} - 2\boldsymbol{c}_t + \boldsymbol{c}_{t+1}$

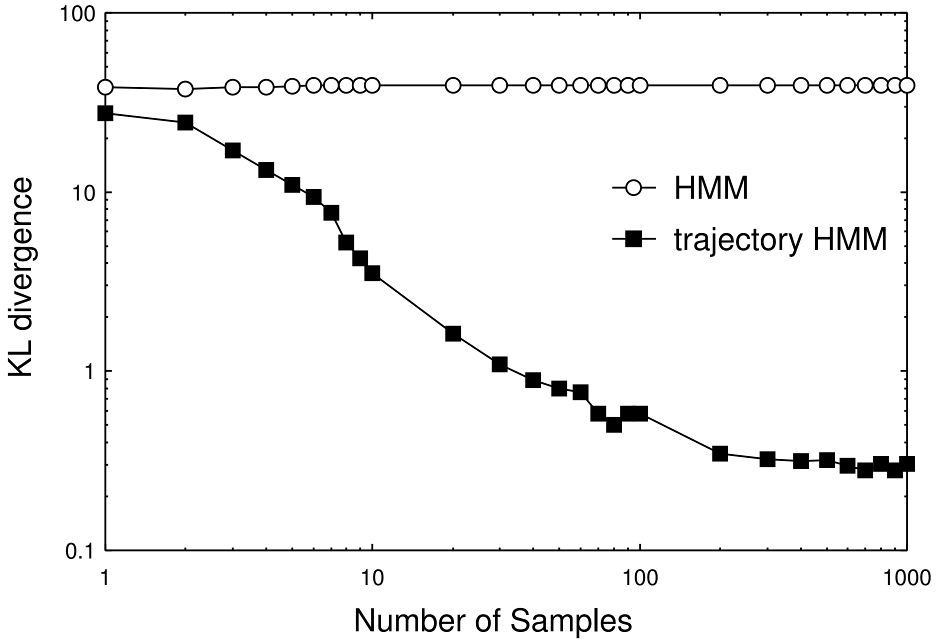


Figure 8.1: Convergence of the Gaussian components estimated from drawn samples from original distributions.

we used the single state with a Gaussian component structure for modeling HMM and trajectory HMM.

Static feature vector sequences of 1000-frame length were drawn from the HMM and trajectory HMM under the constraints between the static and dynamic features. Then we reconstructed a HMM and trajectory HMM using drawn samples, and confirmed that the estimated model converged to the original model, by measuring the KL divergence between the original and reconstructed models.

Figure 8.1 shows the result of the experiment. It can be seen from the figure that as the number of drawn sample increased, the KL divergence between the original and reconstructed trajectory HMMs decreased. On the other hand, the reconstructed HMM did not converge to the original HMM. It indicates that trajectory HMM is a proper generative model and HMM is not.

8.2 Speech recognition experiments

8.2.1 Experimental conditions

The performance of the trajectory HMM was evaluated in a speech recognition experiment. The phonetically balanced 503 sentences uttered by the speaker MHT from the ATR Japanese speech database b-set [101] was used. The first 450 sentences were used for training both the context-independent HMMs and trajectory HMMs. The remaining 53 sentences were used for evaluation. Speech analysis conditions were the same used in the previous section.

Three-state left-to-right structure was used for modeling 36 Japanese phonemes including silence and short pause. Each state had a single Gaussian component with a diagonal covariance matrix.

First, the HMMs were initialized by the segmental k-means algorithm and reestimated using the EM (Baum-Welch) algorithm. Then, the trajectory HMMs were iteratively reestimated by the Viterbi training (Viterbi), the MCEM training with 10 samples (MCEM10) and 50 samples (MCEM50). For training the trajectory HMMs, the HMMs were used as their initial models. We constructed models by changing the number of delay J from 2 to 5 in the delayed decision Viterbi algorithm and Gibbs sampling. In the MCEM training, the initial Gaussian component sequences $\boldsymbol{q}^{(1)}$ were determined by the delayed decision Viterbi algorithm. The same number of delay J was used both in the delayed decision Viterbi algorithm and the Gibbs sampling.

8.2.2 Experimental results

Evaluation of the parameter reestimation formulas

First we evaluated the parameter update formulas derived in Chapter 6. For this experiment, the substate sequence maximizing $p(\boldsymbol{q} \mid \boldsymbol{o}, \Lambda)$ was obtained by the conventional Viterbi algorithm and Eqs. (6.25)–(6.26) were used for updating model parameters.

Table 8.2 shows average log likelihoods of the trajectory HMMs. It is shown that by using the derived reestimation formulas the model likelihoods were improved considerably for both training and test data sets.

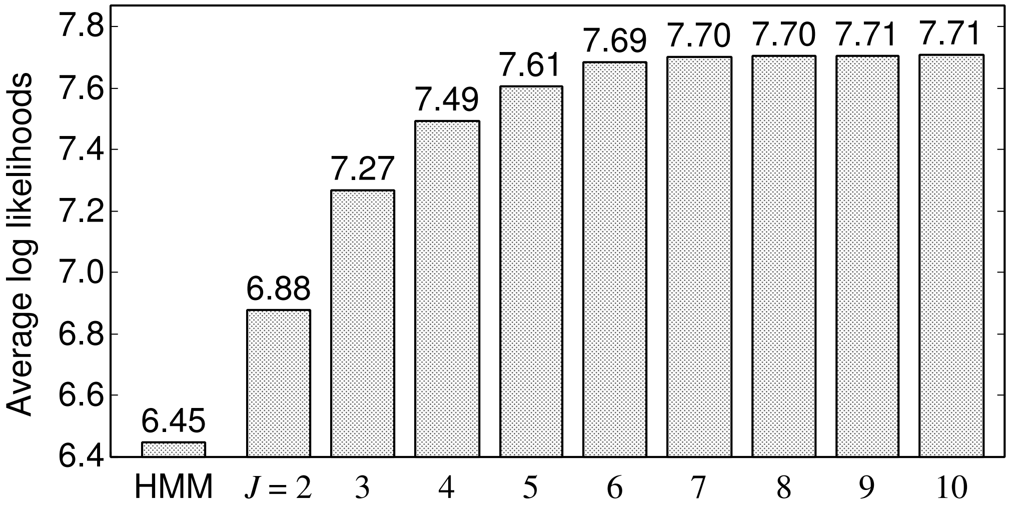


Figure 8.2: Average log-likelihoods for the substate sequences obtained by the delayed decision Viterbi algorithm.

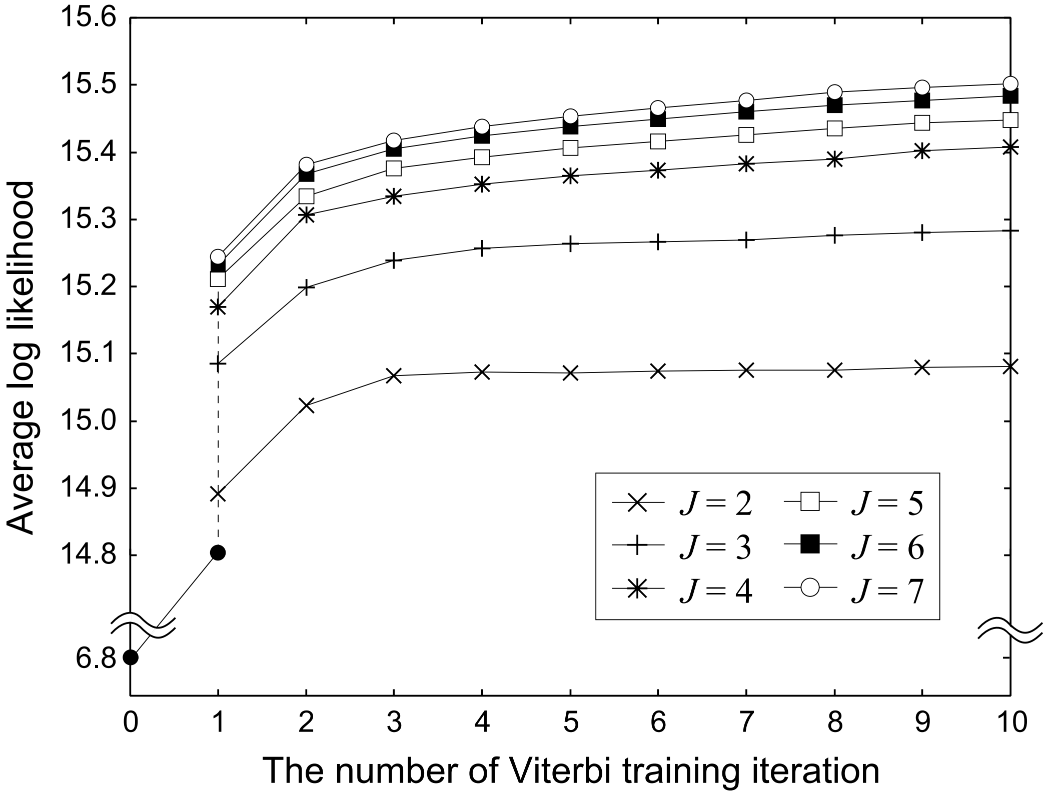


Figure 8.3: Average log likelihoods of the trajectory HMMs for the training data.

Table 8.2: Average log $p(\boldsymbol{c} \mid \boldsymbol{q}, \Lambda)$ of the trajectory HMMs ("w/o update": model parameters were not updated, i.e., those of the were used, "$\boldsymbol{m}$": only means were updated, "$\boldsymbol{\phi}$": only variances were updated, "$\boldsymbol{m}$ & $\boldsymbol{\phi}$": both means and variances were updated).

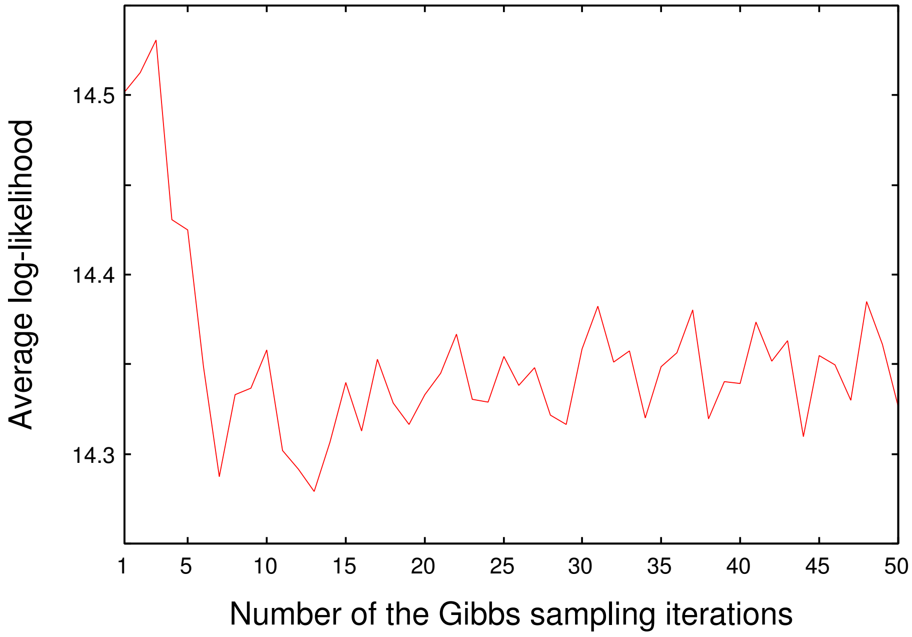
	trajectory HMM			
	w/o update	$\boldsymbol{m}$	$\boldsymbol{\phi}$	$\boldsymbol{m}$ & $\boldsymbol{\phi}$
training data	6.86	7.91	12.0	14.8
test data	6.93	7.83	12.1	14.8

Evaluation of the delayed decision Viterbi algorithm

Secondly, we evaluated the performance of the delayed decision Viterbi algorithm in the likelihood of the obtained substate sequence. To compare with the conventional Viterbi algorithm and the delayed decision Viterbi algorithm in the same condition, the HMMs which was used as the initial models for training the trajectory HMMs were used. Figure 8.2 shows the average $p(\boldsymbol{c} \mid \boldsymbol{q}, \Lambda)$ per frame of substate sequences for an utterance included in the training data. In this figure, "HMM" corresponds to the substate sequence obtained by the conventional Viterbi algorithm, and "$J = 2, J = 3, \dots, J = 10$" means that the substate sequences were obtained by the delayed decision Viterbi algorithm with J-frame delay. It shows that the delayed decision Viterbi algorithm could obtain more likely substate sequences than the HMM. Furthermore, as J increased, the likelihood was gradually increased and converged.

Evaluation of the iterative training

Thirdly, effectiveness of the iterative training was evaluated. The average log likelihoods of the trajectory HMMs for the training data against the number of Viterbi training iteration are illustrated in Fig. 8.3. The average log likelihood of the initial models (HMMs) according to substate sequences found by the Viterbi algorithms was 6.87. After updating $\boldsymbol{m}$ and $\boldsymbol{\phi}$, the average log likelihood improved to 14.8. Then Viterbi training was iterated. Therefore, the same models were used at the first iteration in Fig 8.3. It can be seen from the figure that the delayed decision Viterbi algorithm with longer J could find better sub-optimum Gaussian component sequences. In addition, iterative training improved the model likelihood for the training data.



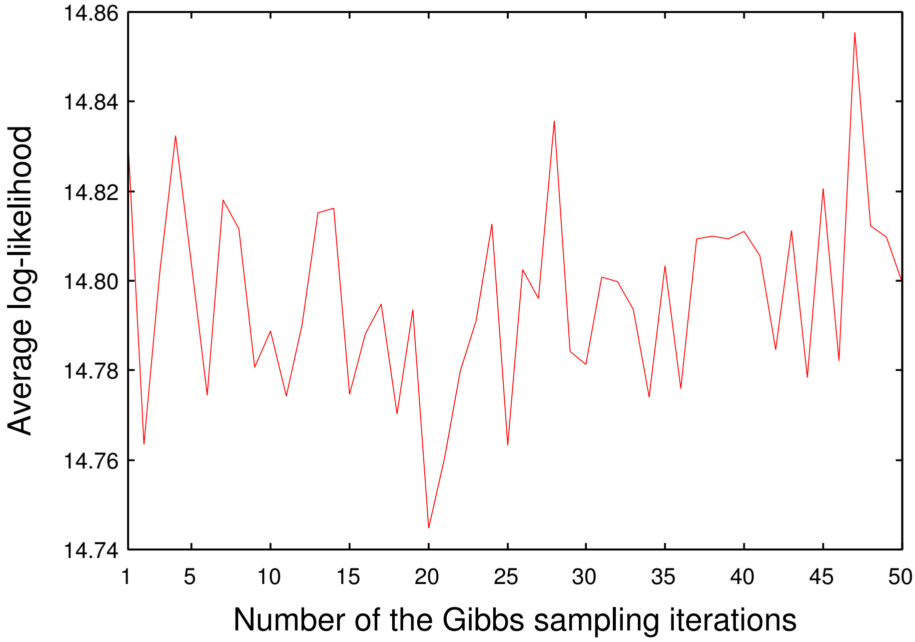


Figure 8.4: Average log-likelihoods of utterances “mhtsda01” (above) and “mhtsda10” (below) for the substate sequences generated by the Gibbs sampler.

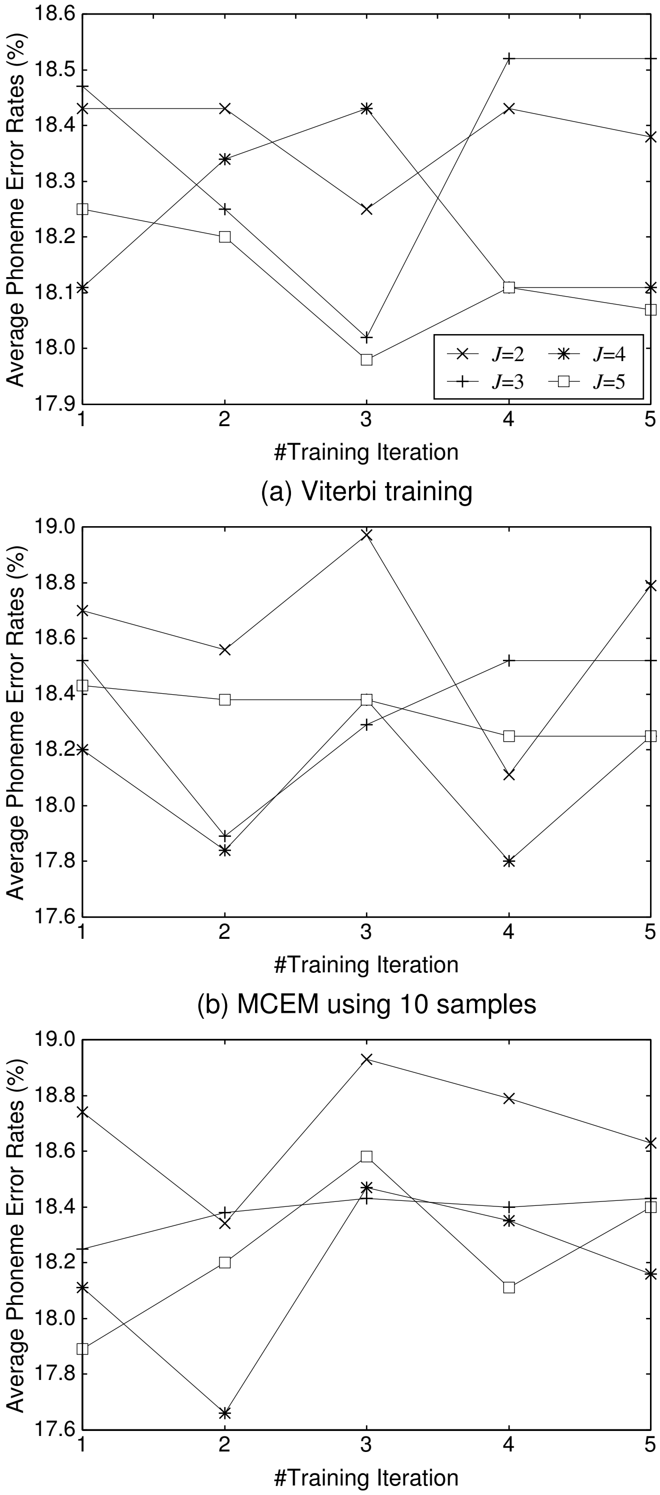
Evaluation of the Gibbs sampling

Figure 8.4 shows average log-likelihoods for utterances included in the training data against the number of Gibbs sampling iterations. We used trajectory HMM estimated by the Viterbi training with 1 iteration and 2-frame lookahead for sampling $\boldsymbol{q}$. The initial substate sequence was given by the delayed decision Viterbi algorithm with 2-frame delay. It shows that the more likely substate sequences compared with the initial one found by the delayed decision Viterbi algorithm were sampled during the Gibbs sampling iteration.

In all recognition experiments reported in this section, a 100-best rescoring paradigm was used. The 100-best lists were generated for each test utterance using the HTK Viterbi decoder [51] with the HMMs used as the initial models for training the trajectory HMMs. These initial models (HMMs) were described as the baseline system in the following recognition experiments. Then, each candidate was re-segmented by the delayed decision Viterbi algorithm and re-scored with its Eq. (7.5). In this experiment, the delayed decision Viterbi algorithm with the same number of delay J was used both in training and evaluation. To give an idea of the range of these 100-best lists, the error rates of the baseline (1-best hypotheses by the HMMs), best, worst, and average of randomly selected hypotheses (100 times) were 19.7%, 13.9%, 27.4%, and 21.2%, respectively. These are the bounds on subsequent rescoring results.

Figures 8.5 and 8.6 show that recognition experimental results of the trajectory HMMs with and without reference included in the hypotheses, respectively. When reference hypotheses were not included, trajectory HMMs estimated by the MCEM using 50 samples (4 iterations and 4 frames delay) achieved the best result. Compared with the Viterbi-trained trajectory HMMs and HMMs, MCEM-trained trajectory HMMs achieved about 3% and 11% error reductions, respectively. When reference hypotheses were included, trajectory HMMs estimated by the MCEM using 10 samples (3 or 5 iteration and 4 frames delay) achieved the best result. When reference hypotheses were included, about 55% error reduction over the HMM was achieved by the MCEM. Although it is difficult to know how to interrupt such results, they might suggest that the decoder based on the approach ii) in the Section 7.3 could further improve the recognition accuracy.

Compared with the Viterbi-trained trajectory HMMs, the MCEM-trained ones were relatively unstable. In Fig. 8.5, phoneme error rates of the Viterbi-trained trajectory HMMs ranged from 18.0% to 18.5%. On the other hand, that of the MCEM-trained ones ranged from 17.6% to 19.0%. It may be caused by that the MCEM is based on a stochastic method. Inconsistency between decoding and training (sampling method was used only on training) could be another reason of this phenomenon.

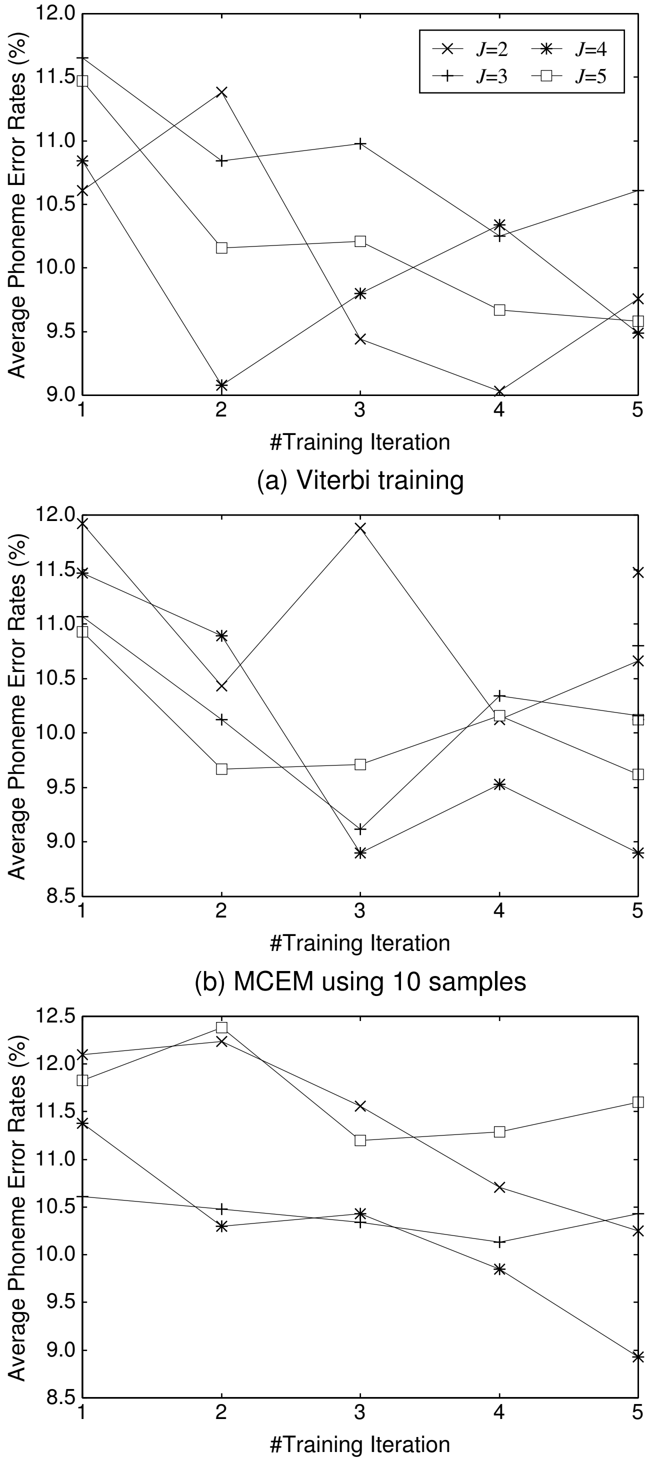


(a) Viterbi training

(b) MCEM using 10 samples

(c) MCEM using 50 samples

Figure 8.5: Phoneme error rates (%) for the test data against the number of training iteration for the trajectory HMM with various number of delay (J). In this experiment, references were not included in the hypotheses).



(a) Viterbi training

(b) MCEM using 10 samples

(c) MCEM using 50 samples

Figure 8.6: Phoneme error rates (%) for the test data against the number of training iteration for the trajectory HMM with various number of delay (*J*). In this experiment, references were included in the hypotheses).

8.3 Speech synthesis experiments

To evaluate the performance of the trajectory HMM in speech synthesis, a subjective listening test was conducted. The first 1096 sentences from CMU ARCTIC database [102] uttered by a male speaker AWB were used for training the HMM-based speech synthesis system. Speech signals were sampled at a rate of 16 kHz and windowed by a 25.6-ms Blackman window with a 5-ms shift, and mel-cepstral coefficients were obtained by a mel-cepstral analysis technique [65]. Fundamental frequency (F_0) values were extracted by the ESPS get_f0 [103] for 5-ms intervals. Static feature vector consisted of spectrum vector and $\log F_0$: the spectrum vector consisted of 25 mel-cepstral coefficients including the zeroth coefficient. They were augmented by adding 1st and 2nd dynamic features which were computed in the same manner described in Section 8.2. We used 5-state left-to-right with noskip structure. Each state output probability distribution consisted of spectrum and F_0 parts. The spectrum part was modeled by a single Gaussian component with diagonal covariance matrix. The F_0 part was modeled by a multi-space probability distribution [104] consisted of a single Gaussian component with a diagonal covariance matrix (voiced space) and a single discrete distribution which outputted only one symbol (unvoiced space).

In this work, the following contextual factors were taken into account:

- phoneme:
    - {before preceding, preceding, current, succeeding, after succeeding} phoneme
    - position of current phoneme in current syllable
- syllable:
    - number of phonemes at {preceding, current, succeeding} syllable
    - {stress[1], accent[2]} of {preceding, current, succeeding} syllable
    - position of current syllable in current {word, phrase}
    - number of {preceding, succeeding} {stressed, accented} syllables in current phrase
    - number of syllables {from previous, to next} {stressed, accented} syllable
    - vowel within current syllable
- word:
    - guess at part of speech of {preceding, current, succeeding} word
    - number of syllables in {preceding, current, succeeding} word

[1] The lexical stress of the syllable as specified from the lexicon entry corresponding to the word related to this syllable.

[2] An intonational accent of the syllable predicted by a CART tree (0 or 1).

    - position of current word in current phrase
    - number of {preceding, succeeding} content words in current phrase
    - number of words {from previous, to next} content word
- phrase:
    - number of syllables in {preceding, current, succeeding} phrase
    - position in major phrase
    - ToBI endtone of current phrase
- utterance:
    - number of {syllables, words, phrases} in current utterance

These contextual factors were extracted using feature extraction functions implemented on the Festival speech synthesis system [105] from utterance informations included in the database. We applied a decision-tree based context clustering technique [70] to distributions for spectrum, F_0, and duration independently. In this dissertation, the minimum description length criterion [106] was used for stopping tree growth [5,107]. For spectrum and F_0, the decision trees were constructed for each state position. The resultant trees for spectrum, F_0, and duration had 978, 1180, and 449 leaf nodes in total, respectively.

To compare the influences of the parameter estimation criteria for the quality of synthetic speech, we trained 3 acoustic models based on different criteria. First, we estimated the HMMs using the Baum-Welch algorithm (model parameters maximizing $p(\boldsymbol{o} \mid \Lambda)$ were reestimated). Then, the trajectory HMMs were estimated using the HMMs as its initial model (model parameters maximizing $p(\boldsymbol{c} \mid \boldsymbol{q}, \Lambda)$ were reestimated). To investigate the influence of the Viterbi (single path) approximation, the Viterbi-trained HMMs were also prepared (model parameters maximizing $p(\boldsymbol{o} \mid \boldsymbol{q}, \Lambda)$ were reestimated). Both the trajectory HMMs and the Viterbi-trained HMMs were not iteratively reestimated. A single pass of reestimation was used both for the trajectory HMMs and the Viterbi-trained HMMs. In this experiment, model parameters of the F_0 part and duration models were not updated. Therefore, prosodic informations (F_0 and duration) of synthetic speech generated from these 3 models were completely the same.

Figure 8.7 shows an example of 2nd mel-cepstral coefficient of a natural speech included in the training data, generated trajectories $\boldsymbol{c}_{\max}$ from the HMMs and trajectory HMMs. It can be seen that the generated trajectory from the trajectory HMMs was almost always closer to the training data than that from the HMMs.

Figure 8.8 shows sequences of speech spectra generated from the HMMs and trajectory HMMs for a sentence fragment “tropic land” taken from a sentence not included in the training data. It can be seen from Fig. 8.8 that formant structures generated from the

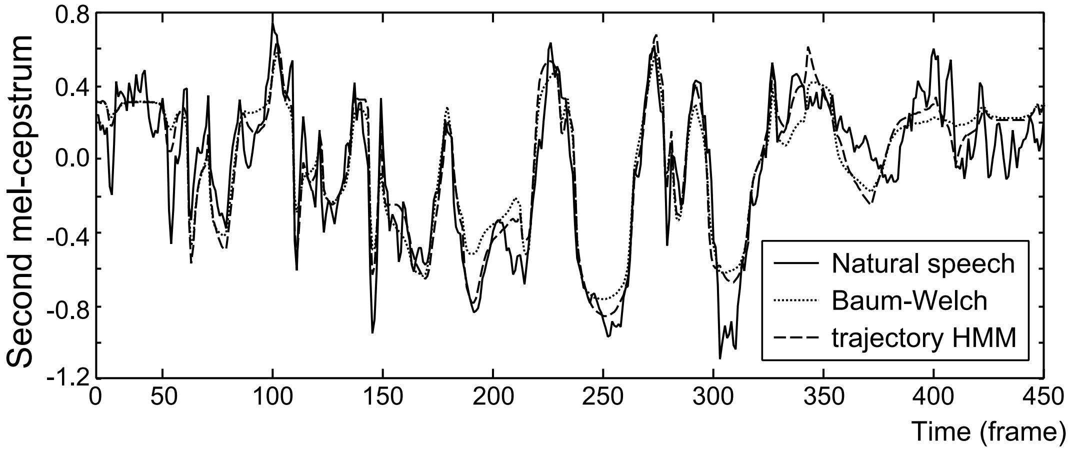
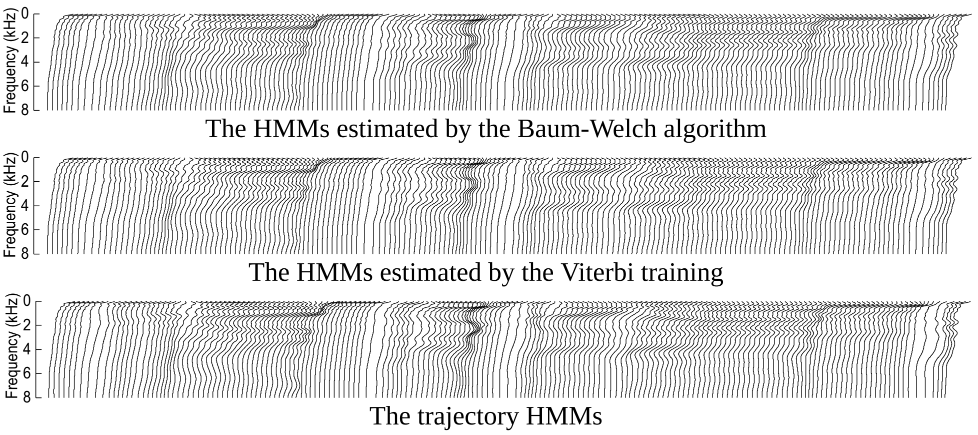


Figure 8.7: Examples of sequence of mel-cesptral coefficients of the natural speech, generated trajectories from the HMMs and trajectory HMMs.



The HMMs estimated by the Baum-Welch algorithm

The HMMs estimated by the Viterbi training

The trajectory HMMs

Figure 8.8: Generated speech spectra for a sentence fragment “tropic land”.

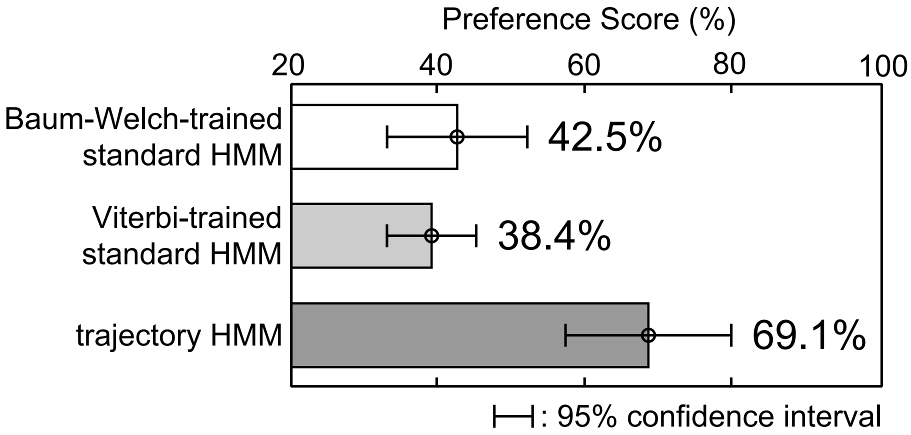


Figure 8.9: Preference scores of the standard HMMs and trajectory HMMs.

trajectory HMMs were slightly clearer than that from the HMMs.

To evaluate the effectiveness of the trajectory HMMs for the HMM-based speech synthesis system, a subjective listening test was conducted. We compared the naturalness of the synthesized speech by paired comparison tests. Subjects were 8 graduate students in our research group, and presented a pair of synthesized speech from different models in random order and then asked which speech sounded more natural. For each subject, 20 test sentences were chosen at random from 42 test sentences not included in the training sentences.

Figure 8.9 shows the preference scores. It can be seen from the figure that the introduction of trajectory HMMs improved the reported naturalness of synthetic speech. Although the Viterbi approximation was used both in the Viterbi-trained standard HMMs and trajectory HMMs, average preference score of the trajectory HMMs was much better than that of the standard HMMs. It indicates that this improvement was achieved by the introduction of the trajectory HMMs, not by the Viterbi approximation.

8.4 Summary

In this chapter, the performances of the trajectory HMM has been evaluated both in speech recognition and synthesis experiments. Both in the speaker-dependent continuous phoneme recognition and the subjective listening test experiments, the trajectory HMMs have achieved significant progress over the corresponding HMMs.

Chapter 9

Conclusions

9.1 Summary

The present dissertation described a novel statistical model named trajectory HMM. This model was derived from the HMM whose state output vector includes both static and dynamic features. In particular, the present dissertation examined the application of the trajectory HMM for acoustic modeling both in speech recognition and synthesis. The trajectory HMM can alleviate two known limitations of the HMM, which are i) constant statistics within a state and ii) conditional independence assumption.

Basic theories and fundamental algorithms of the HMM were reviewed in Chapter 2. Statistical speech recognition and synthesis frameworks based on the HMM were presented in Chapters 3 and 4, respectively. In Chapter 5, the trajectory HMM was derived by reformulating the HMM whose state output vector includes both static and dynamic features. By imposing the explicit relationships between static and dynamic features, the HMM was naturally translated into a trajectory model. This model can overcome two known limitations of the HMM without any additional parameters and it provides a computational model for co-articulation and the dynamics of the human speech. Relationships between the trajectory HMM and other techniques, such as the HMM-trajectory method, the structured precision matrix modeling technique, and product of experts were also discussed in this chapter. In Chapter 6,the training algorithms for the trajectory HMM were described. Unfortunately, exact EM algorithm for the trajectory HMM is intractable. Thus, approximate training algorithms based on the Viterbi and Markov chain Monte Carlo approximations were derived. Chapter 7 showed the search algorithm for the trajectory HMM. The Viterbi algorithm is not applicable for the trajectory HMM. To alleviate this problem, the delayed decision Viterbi algorithm was derived. This algorithm can find a better sub-optimal substate sequence for given static feature vector sequence. The speech recognition and synthesis experiments using the trajectory HMM in acoustic modeling were presented in Chapter 8. In the speaker-dependent continuous speech recognition experiment using the ATR speech database b-set, significant improvements over the corresponding HMM was achieved. In the subjective listening test experiment, reported naturalness of the trajectory HMMs was much better than that of the HMMs.

9.2 Future work

The trajectory HMM was evaluated only in simple speaker-dependent, continous phoneme recognition task using context-independent models with single Gaussian components. To investigate more practical performance of the trajectory HMM, evaluations in speaker-independent, large vocaburary continuous speech recognition task using context-dependent models with mixtures of Gaussian components should be conducted. Due to the huge amount of computational cost both in training and decoding, it is difficult to apply the trajectory HMMs for such a large task. However, improvements of computational power may resolve this problem.

Currently, the ML criterion was used to optimize the model paramters in the present dissertation. However, the ML training is optimal only if the model is close the the true system. One may argue whether there exists a true model for speech, but it is clearly the case that the trajectory HMM does not provide one. For the HMM, systems trained using discriminative training typically outperform ones using the ML training. Different discriminative training schemes for the trajectory HMM could be investigated.

In the present dissertation, rescoring scheme was used for speech recognition experiments. As mentioned in Section 8.2, the decoder based on the delayed decision Viterbi algorithm could further improve the recognition accuracy. It may be hard task, however, it may be worthy of exploring.

Bibliography

[1] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. In *Proceedings of IEEE*, Vol. 77, pp. 257–285, 1989.

[2] A. Ljolje, J. Hirschberg, and J.P.H. van Santen. Automatic speech segmentation for concatenative inventory selection. In J.P.H. van Santen, R.W. Sproat, J.P. Olive, and J. Hirshberg, editors, *Progress in speech synthesis*, pp. 305–311. Springer-Verlag, 1997.

[3] R.E. Donovan and P.C. Woodland. Automatic speech synthesizer parameter estimation using HMMs. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing'95*, pp. 640–643, 1995.

[4] R.E. Donovan and E.M. Eide. The IBM trainable speech synthesis system. In *Proceedings of International Conference on Spoken Language Processing'98*, Vol. 5, pp. 1703–1706, 1998.

[5] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura. Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis. In *Proceedings of European Conference on Speech Communication and Technology'99*, Vol. 5, pp. 2347–2350, 1999.

[6] M. Ostendorf, V. Digalakis, and O.A. Kimball. From HMMs to segment models. *IEEE Transactions on Speech & Audio Processing*, Vol. 4, No. 5, pp. 360–378, 1996.

[7] S. Nakagawa. A survey on automatic speech recognition. *IEICE Transactions Information & System*, Vol. E85-D, No. 3, pp. 465–486, 2002.

[8] X.-D. Huang, A. Acero, and H.-W. Hon. *Spoken Language Processing: a guide to theory, algorithm and system development*. Prentice Hall, 2001.

[9] L. Deng, M. Aksmanovic, X. Sun, and J. Wu. Speech recognition using hidden Markov models with polynomial regression functions as nonstationary states. *IEEE Transactions on Speech & Audio Processing*, Vol. 2, No. 4, pp. 507–520, 1994.

[10] L. Deng and M. Aksmanovic. Speaker-independent phonetic classification using hidden Markov models with mixture of trend functions. *IEEE Transactions Speech & Audio Processing*, Vol. 5, No. 4, pp. 319–324, 1997.

[11] H. Gish and K. Ng. Parametric trajectory models for speech recognition. In *Proceedings of International Conference on Spoken Language Processing'96*, Vol. 1, pp. 466–469, 1996.

[12] L. Deng. A dynamic, feature-based approach to the interface between phonology and phonetics for speech modeling and recognition. *Speech Communication*, Vol. 24, No. 4, pp. 299–323, 1998.

[13] H.B. Richards and J.S. Bridle. The HDM: a segmental hidden dynamic model of coarticulation. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing'99*, Vol. 1, pp. 357–360, 1999.

[14] J. Picone, S. Pike, R. Regan, T. Kamm, J. Bridle, L. Deng, Z. Ma, H. Richards, and M. Schuster. Initial evaluation of hidden dynamic models on conversational speech. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 109–112, 1999.

[15] J. Ma. *Spontaneous speech recognition using statistical dynamic models for the vocal tract resonance dynamics*. PhD thesis, University of Waterloo, 2000.

[16] J. Ma and L. Deng. Target-directed mixture linear dynamic models for spontaneous speech recognition. *IEEE Transactions on Speech and Audio Processing*, Vol. 12, No. 1, pp. 47–58, 2004.

[17] J.-L. Zhou, F. Seide, and L. Deng. Coarticulation modeling by embedding a target-directed hidden trajectory model into HMM – model and training. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing 2003*, Vol. 1, pp. 744–747, 2003.

[18] L. Deng, I. Bazzi, and A. Acero. Tracking vocal tract resonances using an analytical nonlinear predictor and a target-guided temporal constraint. In *Proceedings of European Conference on Speech Communication and Technology 2003*, pp. 73–76, 2003.

[19] L. Deng and J. Ma. Spontaneous speech recognition using a statistical coarticulatory model for the hidden vocal-tract-resonance dynamics. *Journal of Acoustic Society of America*, Vol. 108, No. 6, pp. 3036–3048, 2000.

[20] C.J. Wellekens. Explicit correlation in hidden Markov model for speech recognition. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing'87,* pp. 383–386, 1987.

[21] P. Brown. *The acoustic modeling problem in automatic speech recognition*. PhD thesis, Carnegie Mellon University, 1987.

[22] S. Takahashi. Phoneme HMM's constrained by frame correlations. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing'93,* pp. 219–222, 1993.

[23] K.K. Paliwal. Use of temporal correlation between successive frames in hidden Markov model based speech recognizer. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing'93,* pp. 215–218, 1993.

[24] N.-S. Kim and C.-K. Un. Frame-correlated hidden Markov model based on extended logarithmic pool. *IEEE Transactions on Acoustics, Speech, and Signal Processing,* Vol. 5, No. 2, pp. 149–160, 1997.

[25] G. Qing, Z. Fang, W. Jian, and W. Wenhu. A new method used in HMM for modeling frame correlation. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing'99*, Vol. 1, pp. 169–172, 1999.

[26] T. Kobayashi, K. Masumitsu, and J. Furuyama. Partly hidden Markov model and its application to speech recognition. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing'99*, Vol. 1, pp. 121–124, 1999.

[27] M. Ostendorf and S. Roukos. A stochastic segment model for phoneme-based continuous speech recognition. *IEEE Transactions Acoustics, Speech & Signal Processing,* Vol. 37, No. 12, pp. 1857–1869, 1989.

[28] M. Russel. A segmental HMM for speech pattern modeling. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing'93,* pp. 499–502, 1993.

[29] M.J.F. Gales and S.J. Young. The theory of segmental hidden Markov models. Technical Report CUED/F-INFENG/TR.133, Cambridge University, 1993.

[30] W.J. Holmes and M. Russel. Probabilistic-trajectory segmental HMMs. *Computer, Speech and Language,* Vol. 13, No. 1, pp. 3–37, 1999.

[31] V. Digalakis, J.R. Rohlicek, and M. Ostendorf. A dynamical system approach to continuous speech recognition. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing'91*, pp. 282–292, 1991.

[32] A.V.I. Rosti and M.J.F Gales. Switching linear dynamical systems for speech recognition. Technical Report CUED/F-INFENG/TR.461, Cambridge University, 2003.

[33] L.J. Lee, H. Attias, L. Deng, and P. Fieguth. A multimodal variational approach to learning and inference in switching state space models. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing 2004*, pp. 505–508, 2004.

[34] J. Bilmes. Burried Markov models: A graphical modeling approach for automatic speech recognition. *Computer, Speech and Language*, Vol. 17, No. 2–3, pp. 213–231, 2003.

[35] G. Zweig. *Speech recognition using dynamic Bayesian networks*. PhD thesis, University of California, Berkeley, 1998.

[36] K.-C. Sim and M.J.F. Gales. Temporally varying model parameters for large vocaburaly continuous speech recognition. In *Proceedings of Interspeech'05*, pp. 2137–2140, 2005.

[37] S. Sagayama and F. Itakura. On individuality in a dynamic measure of speech. In *Proceedings of Spring Conference of Acoustic Society of Japan*, pp. 589–590, 1979. (in Japanese).

[38] 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.

[39] C.-H. Lee and E. Giachin. Improved acoustic modeling for speaker independent large vocabulary continuous speech recognition. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 161–164, 1991.

[40] J.G. Wilpon, C.-H. Lee, and L.R. Rabiner. Improvements in connected digit recognition using higher order spectral and energy features. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 349–352, 1991.

[41] K. Tokuda, T. Kobayashi, and S. Imai. Speech parameter generation from HMM using dynamic features. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing'95*, pp. 660–663, 1995.

[42] K. Tokuda, T. Masuko, Y. Yamada, T. Kobayashi, and S. Imai. An algorithm for speech parameter generation from continuous mixture HMMs with dynamic features. In *Proceedings of European Conference on Speech Communication and Technology'95*, pp. 757–760, 1995.

[43] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura. Speech parameter generation algorithms for HMM-based speech synthesis. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing 2000*, Vol. 3, pp. 1315–1318, 2000.

[44] M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi. Adaptation of pitch and spectrum for HMM-based speech synthesis using mllr. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing 2001*, Vol. 2, pp. 805–808, 2001.

[45] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura. Speaker interpolation in HMM-based speech synthesis system. In *Proceedings of European Conference on Speech Communication and Technology'97*, Vol. 5, pp. 2523–2526, 1997.

[46] K. Shichiri, A. Sawabe, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura. Eigenvoices for HMM-based speech synthesis. In *Proceedings of International Conference on Spoken Language Processing 2002*, pp. 1269–1272, 2002.

[47] Y. Minami, E. McDermott, A. Nakamura, and S. Katagiri. A recognition method with parametric trajectory synthesized using direct relations between static and dynamic feature vector time series. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing 2002*, Vol. 1, pp. 957–960, 2002.

[48] Y. Minami, E. McDermott, A. Nakamura, and S. Katagiri. Recognition method with parametric trajectory generated from mixture distribution HMMs. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing 2003*, Vol. 1, pp. 124–127, 2003.

[49] X.-D. Huang, Y. Ariki, and M.A. Jack. *Hidden Markov models mor speech recognition*. Edinburgh University Press, 1990.

[50] Rabiner L. and B.-H. Juang. *Fundamentals of speech recognition*. Prentice-Hall, 1993.

[51] 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.

[52] L.R. 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.

[53] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, Vol. 13, pp. 260–269, 1967.

[54] 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.

[55] L.E. Baum. An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. *Inequalities*, Vol. 3, pp. 1–8, 1972.

[56] B.-H. Juang. Maximum likelihood estimation for mixture of multivariate stochastic observations of Markov chains. *AT&T Technical Journal*, Vol. 64, No. 6, pp. 1235–1249, 1985.

[57] A. McCallum, D. Freitag, and F. Pereira. Maximum entropy markov models for information extraction and segmentation. In *Proceedings of International Conference Machine Learning*, pp. 591–598, 2000.

[58] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of International Conference on Machine Learning*, pp. 282–289, 2001.

[59] H.-K.J. Kuo and Y. Gao. Maximum entropy dicrect models as a unified model for acoustic models in speech recognition. In *Proceedings of International Conference on Spoken Language Processing*, pp. 681–684, 2004.

[60] A. Gunawardana, L. Mahajan, A. Acero, and J.C. Platt. Hidden conditional random fields for phone classification. In *Proceedings of European Conference on Speech Communication and Technology*, pp. 1117–1120, 2005.

[61] J.R. Deller, J.H.L. Hansen, and J.G. Proakis. *Discrete-time processing of speech signals*. Macmillan, 1993.

[62] F. Itakura and S. Saito. A statistical method for estimation of speech spectral density and formant frequencies. *Tranactions of IEICE*, Vol. J53-A, pp. 35–42, 1970.

[63] J.D. Markel and A.H. Gray Jr. *Linear prediction of speech*. Springer-Verlag, 1976.

[64] A.V. Oppenheim and R.W. Schafer. *Digital signal processing*. Englewood Cliffs, 1975.

[65] T. Fukada, K. Tokuda, Kobayashi T., and S. Imai. An adaptive algorithm for mel-cepstral analysis of speech. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing'92,* Vol. 1, pp. 137–140, 1992.

[66] S.B. Gavis and P. Mermelstein. Comparison of parameteric representations for monosylabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics Speech and Signal Processing*, Vol. 28, pp. 357–366, 1980.

[67] H. Hermansky. Perceptual linear prediction (plp) of speech. *Journal of the Acoustic Society of America*, Vol. 87, No. 4, pp. 1738–1752, 1990.

[68] S. Sagayama. Phoneme environment clustering for speech recognition. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, pp. 397–400, 1989.

[69] K.-F. Lee, S. Hayamizu, H.-W. Hon, C. Huang, J. Swartz, and R. Weide. Allophone clustering for continuous speech recognition. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing,* pp. 749–752, 1990.

[70] J.J. Odell. *The use of context in large vocabulary speech recognition*. PhD thesis, Cambridge University, 1995.

[71] H.J. Nock, M.J.F. Gales, and S.J. Young. A comparative study of methods for phonetic decision-tree state clustering. In *Proceedings of European Conference on Speech Communication and Technology,* Vol. 1, pp. 111–114, 1997.

[72] 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. In *Proceedings of International Conference on Spoken Language Processing*, pp. 1233–1236, 2002.

[73] S.M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing,* Vol. 35, No. 3, pp. 400–411, 1987.

[74] 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.

[75] F. Jelinek. Fast sequential decoding algorithm using a stack. *IBM Journal of Research and Development*, Vol. 13, pp. 675–685, 1969.

[76] R. Sproat, J. Hirschberg, and D. Yarowsky. A corpus-based synthesizer. In *Proceedings of International Conference on Spoken Language Processing*, pp. 563–566, 1992.

[77] A.W. Black and P. Taylor. CHATR: a generic speech synthesis system. In *Proceedings of COLING94*, 1994.

[78] A. Hunt and A.W. Black. Unit selection in a concatenative speech synthesis system using a large speech database. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 373–376, 1996.

[79] A.W. Black and P. Taylor. The Festival speech synthesis system: system documentation. Technical Report HCRC/TR-83, University of Edinburgh, 1997.

[80] A. Conkie. A robust unit selection system for speech synthesis. In *Proceedings of Acoustic Society of America*, 1999.

[81] X.-D. Huang, A. Acero, J. Adcock, H.-W. Hon, J. Goldsmith, J. Liu, and M. Plumpe. Whistler: A trainable text-to-speech system. In *Proceedings of International Conference on Spoken Language Processing*, 1996.

[82] N. Mizutani, K. Tokuda, and T. Kitamura. Concatenative speech synthesis based on hmm. In *Proceedings of Autumn Conference of Acoustic Society of Japan*, pp. 241–242, 2002. (in Japanese).

[83] C. Allauzen, M. Mohri, and M. Riley. Statistical modeling for unit selection in speech synthesis. In *Proceedings of the 42nd meeting of the Association for Computational Linguistics*, 2004.

[84] S. Sakai and H. Shu. A probabilistic approach to unit selection for corpus-based speech synthesis. In *Proceedings of the European Conference on Speech Communication and Technology*, pp. 81–84, 2005.

[85] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura. Duration modeling for HMM-based speech synthesis. In *Proceedings of International Conference on Spoken Language Processing'98*, Vol. 2, pp. 29–32, 1998.

[86] T. Masuko, K. Tokuda, T. Kobayashi, and S. Imai. Speech synthesis from HMMs using dynamic features. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing'96*, pp. 389–392, 1996.

[87] A. Acero. Formant analysis and synthesis using hidden Markov models. In *Proceedings of European Conference on Speech Communication and Technology'99*, pp. 1047–1050, 1999.

[88] K. Koishida, K. Tokuda, T. Masuko, and T. Kobayashi. Vector quantization of speech spectral parameters using statistics of dynamic features. In *Proceedings of International Conference on Signal Processing'97*, pp. 247–252, 1997.

[89] J.S. Bridle. Towards better understanding of the model implied by the use of dynamic features in HMMs. In *Proceedings of International Conference on Spoken Language Processing 2004*, Vol. 1, pp. 725–728, 2004.

[90] Y. Minami, E. McDermott, A. Nakamura, and S. Katagiri. A theoretical analysis of speech recognition based on feature trajectory models. In *Proceedings of International Conference on Spoken Language Processing 2004*, Vol. 1, pp. 549–552, 2004.

[91] J. Bilmes. Factored sparce inverse covariance matrices. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing'97*, Vol. 2, pp. 1009–1012, 2000.

[92] V. Vanhoucke. *Mixtures of inverse covariances: covariance modeling for Gaussian mixtures with applications to automatic speech recognition*. PhD thesis, Stanford University, 2003.

[93] M.J.F. Gales. Semi-tied covariance matrices for hidden Markov models. *IEEE Transactions on Speech and Audio Processing*, Vol. 7, No. 3, pp. 272–281, 1999.

[94] P.A. Olsen and R.A. Gopinath. Modeling inverse covariance matrices by basis expansion. *IEEE Transactions on Acoustic, Speech, and Signal Processing*, Vol. 12, pp. 37–46, 2004.

[95] K.-C. Sim and M.J.F. Gales. Precision matrix modeling for large vocabulary continuous speech recognition. Technical Report CUED/F-INFENG/TR.485, Cambridge University, 2004.

[96] G. Hinton. Product of experts. In *Proceedings of ICANN*, Vol. 1, pp. 1–6, 1999.

[97] C.K.I. Williams, F.V. Agakov, and S.N. Felderhof. Products of Gaussians. In T.G. Diettrich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pp. 1014–1017. MIT Press, 2002.

[98] M.J.F. Gales and S.S. Airey. Product of Gaussians for speech recognition. Technical Report CUED/F-INFENG/TR.458, Cambridge University, 2003.

[99] C.K.I. Williams. How to pretend that correlated variables are independent by using difference observations. *Neural Computation*, Vol. 17, No. 1, pp. 1–7, 2005.

[100] G.C.G. Wei and M.A. Tanner. A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, Vol. 85, pp. 699–704, 1990.

[101] A. Kurematsu, K. Takeda, Y. Sagisaka, S. Katagiri, H. Kuwabara, and K. Shikano. ATR Japanese speech database as a tool of speech recognition and synthesis. *Speech Communication*, Vol. 9, pp. 357–363, 1990.

[102] J. Kominek and A.W. Black. CMU ARCTIC databases for speech synthesis. Technical Report CMU-LTI-03-177, Carnegie Mellon University, 2003.

[103] D. Talkin. A robust algorithm for pitch tracking (RAPT). In W.B. Kleijn and K.K. Paliwal, editors, *Speech coding and synthesis*, pp. 497–518. Elsevier, 1995.

[104] K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi. Hidden Markov models based on multi-space probability distribution for pitch pattern modeling. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing'99*, pp. 229–232, 1999.

[105] A.W. Black and R. Caley. The festival speech synthesis system ver. 2.0, 2004. `http://www.festvox.org/festival/`.

[106] J. Rissanen. *Stochastic complexity in stochastic inequiry*. World Scrientific Publishing Company, 1980.

[107] K. Shinoda and T. Watanabe. Acoustic modeling based on the MDL criterion for speech recognition. In *Proceedings of European Conference on Speech Communication and Technology'97*, pp. 99–102, 1997.

Appendix A

Derivation of trajectory HMM

By imposing the explicit relationships between static and dynamic features represented by Eq. (5.10), Eq. 5.16 can be rewritten as follows:

$$\mathcal{N}\left(\boldsymbol{W}\boldsymbol{c} \mid \boldsymbol{\mu}_{\boldsymbol{q}}, \boldsymbol{\Sigma}_{\boldsymbol{q}}\right) = \frac{1}{\sqrt{(2\pi)^{3KT}\left|\boldsymbol{\Sigma}_{\boldsymbol{q}}\right|}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{W}\boldsymbol{c} - \boldsymbol{\mu}_{\boldsymbol{q}}\right)^{\top} \boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1}\left(\boldsymbol{W}\boldsymbol{c} - \boldsymbol{\mu}_{\boldsymbol{q}}\right)\right\} \tag{A.1}$$

$$= \frac{1}{\sqrt{(2\pi)^{3KT}\left|\boldsymbol{\Sigma}_{\boldsymbol{q}}\right|}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{\mu}_{\boldsymbol{q}}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1}\boldsymbol{\mu}_{\boldsymbol{q}} + \boldsymbol{c}^{\top}\mathbf{W}\boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1}\boldsymbol{W}\boldsymbol{c} - 2\boldsymbol{\mu}_{\boldsymbol{q}}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1}\boldsymbol{W}\boldsymbol{c}\right)\right\} \tag{A.2}$$

$$= \frac{1}{\sqrt{(2\pi)^{3KT}\left|\boldsymbol{\Sigma}_{\boldsymbol{q}}\right|}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{\mu}_{\boldsymbol{q}}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1}\boldsymbol{\mu}_{\boldsymbol{q}} + \boldsymbol{c}^{\top}\boldsymbol{R}_{q}\boldsymbol{c} - 2\boldsymbol{r}_{q}\boldsymbol{c}\right)\right\} \tag{A.3}$$

$$= \frac{1}{\sqrt{(2\pi)^{3KT}\left|\boldsymbol{\Sigma}_{\boldsymbol{q}}\right|}} \exp\left[-\frac{1}{2}\left\{\left(\boldsymbol{c} - \bar{\boldsymbol{c}}_{\boldsymbol{q}}\right)^{\top} \boldsymbol{R}_{\boldsymbol{q}}\left(\boldsymbol{c} - \bar{\boldsymbol{c}}_{\boldsymbol{q}}\right) - \boldsymbol{r}_{\boldsymbol{q}}^{\top}\boldsymbol{P}_{q}\boldsymbol{r}_{q} + \boldsymbol{\mu}_{\boldsymbol{q}}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1}\boldsymbol{\mu}_{\boldsymbol{q}}\right\}\right] \tag{A.4}$$

$$= \frac{\sqrt{(2\pi)^{KT}\left|\boldsymbol{P}_{\boldsymbol{q}}\right|}}{\sqrt{(2\pi)^{3KT}\left|\boldsymbol{\Sigma}_{\boldsymbol{q}}\right|}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{\mu}_{\boldsymbol{q}}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1}\boldsymbol{\mu}_{\boldsymbol{q}} - \boldsymbol{r}_{\boldsymbol{q}}^{\top}\boldsymbol{P}_{q}\boldsymbol{r}_{q}\right)\right\} \\ \frac{1}{\sqrt{(2\pi)^{KT}\left|\boldsymbol{P}_{\boldsymbol{q}}\right|}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{c} - \bar{\boldsymbol{c}}_{\boldsymbol{q}}\right)^{\top} \boldsymbol{R}_{\boldsymbol{q}}\left(\boldsymbol{c} - \bar{\boldsymbol{c}}_{\boldsymbol{q}}\right)\right\} \tag{A.5}$$

$$= Z_{\boldsymbol{q}} \cdot \mathcal{N}\left(\boldsymbol{c} \mid \bar{\boldsymbol{c}}_{q}, \boldsymbol{P}_{q}\right). \tag{A.6}$$

Appendix B

Derivation of reestimation formula for the concatenated mean vector *m*

From Eq. (6.7), the auxiliary function of the trajectory HMM is given as follows:

$$
\begin{aligned}
\mathcal{Q}(\Lambda, \hat{\Lambda}) = \sum_{\text{all } \boldsymbol{q}} p\,(\boldsymbol{q} \mid \boldsymbol{c}, \Lambda) \cdot \Bigg[& \log \pi_{q_1} + \sum_{t=2}^{T} \log a_{q_{t-1}q_t} + \sum_{t=1}^{T} \log w_{q_t s_t} \\
& - \frac{1}{2} \left\{ KT \log(2\pi) - \log \left| \boldsymbol{R}_{\boldsymbol{q}} \right| + \boldsymbol{c}^\top \boldsymbol{R}_{\boldsymbol{q}} \boldsymbol{c} + \boldsymbol{r}_{\boldsymbol{q}}^\top \boldsymbol{P}_{\boldsymbol{q}} \boldsymbol{r}_{\boldsymbol{q}} - 2 \boldsymbol{r}_{\boldsymbol{q}}^\top \boldsymbol{c} \right\} \Bigg]. \qquad \text{(B.7)}
\end{aligned}
$$

According to Eq. (6.17), the auxiliary function can be rewritten as

$$
\begin{aligned}
\mathcal{Q}(\Lambda, \hat{\Lambda}) = \sum_{\text{all } \boldsymbol{q}} p\,(\boldsymbol{q} \mid \boldsymbol{c}, \Lambda) \cdot \Bigg[& \log \pi_{q_1} + \sum_{t=2}^{T} \log a_{q_{t-1}q_t} + \sum_{t=1}^{T} \log w_{q_t s_t} \\
& - \frac{1}{2} \Bigg\{ KT \log(2\pi) - \log \left| \boldsymbol{R}_{\boldsymbol{q}} \right| + \boldsymbol{c}^\top \boldsymbol{R}_{\boldsymbol{q}} \boldsymbol{c} \\
& + \boldsymbol{m}^\top \boldsymbol{S}_{\boldsymbol{q}}^\top \boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1} \boldsymbol{W}^\top \boldsymbol{P}_{\boldsymbol{q}} \boldsymbol{W} \boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1} \boldsymbol{S}_{\boldsymbol{q}} \boldsymbol{m} - 2 \boldsymbol{m}^\top \boldsymbol{S}_{\boldsymbol{q}}^\top \boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1} \boldsymbol{W}^\top \boldsymbol{c} \Bigg\} \Bigg]. \qquad \text{(B.8)}
\end{aligned}
$$

By taking a partial derivative with respect to ***m***, we obtain

$$
\frac{\partial \mathcal{Q}(\Lambda, \hat{\Lambda})}{\partial \boldsymbol{m}} = \sum_{\text{all } \boldsymbol{q}} p\,(\boldsymbol{q} \mid \boldsymbol{c}, \Lambda) \cdot \boldsymbol{S}_{\boldsymbol{q}}^\top \Sigma_{\boldsymbol{q}}^{-1} \boldsymbol{W} \left(\boldsymbol{c} - \boldsymbol{P}_{\boldsymbol{q}} \boldsymbol{W}^\top \boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1} \boldsymbol{S}_{\boldsymbol{q}} \boldsymbol{m} \right) \qquad \text{(B.9)}
$$

By setting the above equation equal to **0**, a set of linear equations for determination of ***m*** maximizing the auxiliary function are obtained as follows:

$$\sum_{\text{all }\boldsymbol{q}} p\left(\boldsymbol{q} \mid \boldsymbol{c}, \Lambda\right) \cdot \boldsymbol{S}_{\boldsymbol{q}}^{\top} \Sigma_{\boldsymbol{q}}^{-1} \boldsymbol{W} \left(\boldsymbol{c} - \boldsymbol{P}_{\boldsymbol{q}} \boldsymbol{W}^{\top} \boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1} \boldsymbol{S}_{\boldsymbol{q}} \boldsymbol{m}\right) = \mathbf{0} \tag{B.10}$$

$$\sum_{\text{all }\boldsymbol{q}} p\left(\boldsymbol{q} \mid \boldsymbol{c}, \Lambda\right) \cdot \boldsymbol{S}_{\boldsymbol{q}}^{\top} \Sigma_{\boldsymbol{q}}^{-1} \boldsymbol{W} \boldsymbol{P}_{\boldsymbol{q}} \boldsymbol{W}^{\top} \boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1} \boldsymbol{S}_{\boldsymbol{q}} \boldsymbol{m} = \sum_{\text{all }\boldsymbol{q}} p\left(\boldsymbol{q} \mid \boldsymbol{c}, \Lambda\right) \cdot \boldsymbol{S}_{\boldsymbol{q}}^{\top} \Sigma_{\boldsymbol{q}}^{-1} \boldsymbol{W} \boldsymbol{c} \tag{B.11}$$

$$\sum_{\text{all }\boldsymbol{q}} p\left(\boldsymbol{q} \mid \boldsymbol{c}, \Lambda\right) \cdot \boldsymbol{\Phi}^{-1} \boldsymbol{S}_{\boldsymbol{q}}^{\top} \boldsymbol{W} \boldsymbol{P}_{\boldsymbol{q}} \boldsymbol{W}^{\top} \boldsymbol{S}_{\boldsymbol{q}} \boldsymbol{\Phi}^{-1} \boldsymbol{m} = \sum_{\text{all }\boldsymbol{q}} p\left(\boldsymbol{q} \mid \boldsymbol{c}, \Lambda\right) \cdot \boldsymbol{\Phi}^{-1} \boldsymbol{S}_{\boldsymbol{q}} \boldsymbol{W} \boldsymbol{c} \tag{B.12}$$

$$\sum_{\text{all }\boldsymbol{q}} p\left(\boldsymbol{q} \mid \boldsymbol{c}, \Lambda\right) \cdot \boldsymbol{S}_{\boldsymbol{q}}^{\top} \boldsymbol{W} \boldsymbol{P}_{\boldsymbol{q}} \boldsymbol{W}^{\top} \boldsymbol{S}_{\boldsymbol{q}} \boldsymbol{\Phi}^{-1} \boldsymbol{m} = \sum_{\text{all }\boldsymbol{q}} p\left(\boldsymbol{q} \mid \boldsymbol{c}, \Lambda\right) \cdot \boldsymbol{S}_{\boldsymbol{q}}^{\top} \boldsymbol{W} \boldsymbol{c} \tag{B.13}$$

List of Publications

Publications

Journal papers

[1] **Heiga Zen**, Keiichi Tokuda, Tadashi Kitamura, "Decision tree based simultaneous clustering of phonetic contexts, dimensions, and state positions for acoustic modeling," *IEICE Trans. Inf. & Syst.*, vol.J87-D-II, no.8, pp.1593–1602, Aug. 2004 (in Japanese).

[2] **Heiga Zen**, Keiichi Tokuda and Tadashi Kitamura, "Reformulating HMM as a trajectory model by imposing explicit relationships between static and dynamic features," *Computer, Speech & Language* (Accepted).

International conference proceedings

[3] **Heiga Zen**, Keiichi Tokuda and Tadashi Kitamura, "Decision tree distribution tying based on a dimensional split technique," *Proc. of ICSLP 2002*, pp.1257–1260, Sept. 2002.

[4] **Heiga Zen**, Keiichi Tokuda and Tadashi Kitamura, "Decision tree based simultaneous clustering of phonetic contexts, dimensions, and state positions for acoustic modeling," *Proc. of Eurospeech 2003*, pp.3189–3192, Sept. 2003.

[5] Keiichi Tokuda, **Heiga Zen** and Tadashi Kitamura, "Trajectory modeling based on HMMs with the explicit relationship between static and dynamic features," *Proc. of Eurospeech 2003*, pp.865–868, Sept. 2003.

[6] **Heiga Zen**, Keiichi Tokuda and Tadashi Kitamura, "A Viterbi algorithm for a trajectory model derived from HMM with explicit relationship between static and dynamic features," *Proc. of ICASSP 2004*, pp.837–840, Montreal, May 2004.

[7] **Heiga Zen**, Keiichi Tokuda and Tadashi Kitamura, “An introduction of trajectory model into HMM-based speech synthesis,” *Proc. of* 5th ISCA Speech Synthesis Workshop, Pittsburgh, June 2004.

[8] Keiichi Tokuda, **Heiga Zen** and Tadashi Kitamura, “Reformulating the HMM as a trajectory model,” *Proc. of Beyond HMM – Workshop on statistical modeling approach for speech recognition*, Kyoto, Dec. 2004.

[9] **Heiga Zen**, Yoshihiko Nankaku, Keiichi Tokuda and Tadashi Kitamura, “Estimating trajectory HMM parameters using Monte Carlo EM with Gibbs sampler,” *Proc. of ICASSP2006*, May 2006 (Accepted).

Technical reports

[10] **Heiga Zen**, Keiichi Tokuda, Tadashi Kitamura, “Decision tree based simultaneous clustering of phonetic contexts, dimensions, and state positions for acoustic modeling,” *Technical Report of IEICE*, SP2003-6, Apr. 2003 (in Japanese).

[11] **Heiga Zen**, Keiichi Tokuda, Tadashi Kitamura, “A trajectory model derived from HMM with explicit relationship between static and dynamic features,” *Technical Report of IEICE*, SP2003-122, Dec. 2003 (in Japanese).

[12] **Heiga Zen**, Tomoki Toda, Masaru Nakamura, Keiichi Tokuda, “Details of Nitech HMM-based speech synthesis system for Blizzard Challenge 2005,” *Technical Report of IEICE*, Jan. 2006.

Domestic conference proceedings

[13] **Heiga Zen**, Keiichi Tokuda, Tadashi Kitamura, “HMM state partial sharing method based on tree-based clustering,” *Proc. of Spring Meeting of the ASJ*, 2-5-2, March 2002 (In Japanese).

[14] **Heiga Zen**, Keiichi Tokuda, Tadashi Kitamura, “Decision tree based simultaneous clustering of phonetic context, dimension, and state position,” *Proc. of Autumn Meeting of the ASJ*, 1-9-20, Sept. 2002 (In Japanese).

[15] **Heiga Zen**, Keiichi Tokuda, Tadashi Kitamura, “A trajectory model derived from continuous density HMM with static and dynamic features,” *Proc. of Spring Meeting of the ASJ*, 1-4-7, March 2003 (In Japanese).

[**16**] **Heiga Zen**, Keiichi Tokuda, Tadashi Kitamura, “A Viterbi algorithm for trajectory HMM,” *Proc. of Autumn Meeting of the ASJ*, vol. I, pp. 65-66, 2-6-3, Sept. 2003 (In Japanese).

[**17**] **Heiga Zen**, Yoshihiko Nankaku, Keiichi Tokuda, Tadashi Kitamura, “An EM-type training algorithm for trajectory-HMM using Gibbs sampling,” *Proc. of Autumn Meeting of the ASJ*, 2-7-16, Sept. 2005 (in Japanese).

[**18**] **Heiga Zen**, Yoshihiko Nankaku, Keiichi Tokuda, Tadashi Kitamura, “Speaker adaptation of trajectory-HMM using constrained MLLR,” *Proc. of Autumn Meeting of the ASJ*, 3-7-6, Sept. 2005 (in Japanese).

Related Publications

Journal papers

[**1**] Amaro Lima, **Heiga Zen**, Yoshihiko Nankaku, Chiyomi Miyajima, Keiichi Tokuda and Tadashi Kitamura, “On the use of kernel PCA for feature extraction of speech recognition,” *IEICE Trans. Inf. & Syst.*, vol.E87-D, no.12, pp.2802–2811, Dec. 2004.

[**2**] Amaro Lima, **Heiga Zen**, Yoshihiko Nankaku, Keiichi Tokuda, Tadashi Kitamura and Fernand. G. Resende, “Applying sparse KPCA for feature extraction in speech recognition,” *IEICE Trans. Inf. & Syst.*, vol.E88-D, no.3, pp.401–409, March 2005.

[**3**] Hiroyuki Suzuki, **Heiga Zen**, Yoshihiko Nankaku, Chiyomi Miyzjima, Keiichi Tokuda and Tadashi Kitamura, “Continuous speech recognition based on general factor dependent acoustic models,” *IEICE Trans. Inf. & Syst.*, vol.E88-D, no.3, pp.410–417, March 2005.

[**4**] Yohei Itaya, **Heiga Zen**, Yoshihiko Nankaku, Chiyomi Miyajima, Keiichi Tokuda and Tadashi Kitamura, “Deterministic annealing EM algorithm in acoustic modeling for speaker and speech recognition,” *IEICE Trans. Inf. & Syst.*, vol.E88-D, no.3, pp.425–431, March 2005.

International conference an workshop proceedings

[**5**] Keiichi Tokuda, **Heiga Zen** and Alan W. Black, “An HMM-based speech synthesis system applied to English,” *Proc. of IEEE Speech Synthesis Workshop*, Sept. 2002.

[**6**] Hiroyuki Suzuki, **Heiga Zen**, Yoshihiko Nankaku, Chiyomi Miyajima, Keiichi Tokuda and Tadashi Kitamura, "Speech recognition using voice-characteristic dependent acoustic model," *Proc. of ICASSP 2003*, vol.1, pp.740–743, Apr. 2003.

[**7**] Takahiro Hoshiya, Shinji Sako, **Heiga Zen**, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi and Tadashi Kitamura, "Improving the performance of HMM-based very low bitrate speech coding," *Proc. of ICASSP 2003*, vol.1, pp.800–803, Apr. 2003.

[**8**] Ranniery S. Maia, **Heiga Zen**, Keiichi Tokuda and Tadashi Kitamura, "Towards the development of a Brazilian Portuguese text-to-speech system based on HMM," *Proc. of Eurospeech 2003*, pp.2465–2468, Sept. 2003.

[**9**] Amaro Lima, **Heiga Zen**, Yoshihiko Nankaku, Chiyomi Miyajima, Keiichi Tokuda and Tadashi Kitamura, "On the use of kernel PCA for feature extraction in speech recognition," *Proc. of Eurospeech 2003*, pp.2625–2628, Sept. 2003.

[**10**] **Heiga Zen**, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi and Tadashi Kitamura, "Hidden semi-Markov model based speech synthesis," *Proc. of ICSLP 2004*, vol.II, pp.1397–1400, Jeju, Oct. 2004.

[**11**] Yohei Itaya, **Heiga Zen**, Yoshihiko Nankaku, Chiyomi Miyajima, Keiichi Tokuda and Tadashi Kitamura, "Deterministic annealing EM algorithm in parameter estimation for acoustic model," *Proc. of ICSLP 2004*, vol.I, pp.433–436, Jeju, Oct. 2004.

[**12**] Ryosuke Tsuduki, **Heiga Zen**, Keiichi Tokuda, Tadashi Kitamura, Murtaza Bulut and Shrikanth S. Narayanan, "Constructing emotional speech synthesizers with limited speech database," *Proc. of ICSLP 2004*, vol.II, pp.1185–1188, Jeju, Oct. 2004.

[**13**] Amaro Lima, **Heiga Zen**, Yoshihiko Nankaku, Keiichi Tokuda and Tadashi Kitamura, "Sparse KPCA for feature extraction in speech recognition," *Proc. of ICASSP2005*, vol.I, pp.353–356, Philadelphia, PA, Mar. 2005.

[**14**] **Heiga Zen** and Tomoki Toda, "An overview of Nitech HMM-based speech synthesis system for Blizzard Challenge 2005," *Proc. of Interspeech2005 (Eurospeech)*, pp.93–96, Lisbon, Sept. 2005.

[**15**] Wael Hamza, Raimo Bakis, Zhang Wei Shuang and **Heiga Zen**, "On building a concatenative speech synthesis system from the Blizzard Challenge speech databases," *Proc. of Interspeech2005 (Eurospeech)*, pp.97–100, Lisbon, Sept. 2005.

Technical reports

[**16**] **Heiga Zen**, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, Tadashi Kitamura, "A pitch pattern modeling technique using dynamic features on the border of voiced and unvoiced segments," *Technical Report of IEICE,* SP-69, Sept. 2001 (in Japanese).

[**17**] **Heiga Zen**, Jinlin Lu, Jinfu Ni, Keiichi Tokuda, Hisashi Kawai, "HMM-based prosody modeling and synthesis for Japanese and Chinese speech synthesis," *Technical Report of ATR-SLT*, TR-SLT-0032, March 2003 (in Japanese).

[**18**] Y. Nankaku, **Heiga Zen**, Keiichi Tokuda, Tadashi Kitamura, Takashi Masuko, "A Bayesian approach to HMM-based speech synthesis," *Technical Report of IEICE,* SP2003-77, Aug. 2003 (in Japanese).

[**19**] R. Tsuduki, **Heiga Zen**, Keiichi Tokuda, Tadashi Kitamura, Murtaza Bulut, Shrikanth Narayanan, "Emotional speech modeling in HMM-based speech synthesis," *Technical Report of IEICE*, SP2003-78, Aug. 2003 (in Japanese).

[**20**] Y. Itaya, **Heiga Zen**, Y. Nankaku, C. Miyajima, Keiichi Tokuda, Tadashi Kitamura, "Deterministic annealing EM algorithm for speaker and speech recognition," *Technical Report of IEICE*, SP2003-113, Dec. 2003 (in Japanese).

[**21**] Hiroyuki Suzuki, **Heiga Zen**, Yoshihiko Nankaku, Chiyomi Miyajima, Keiichi Tokuda, Tadashi Kitamura, "Acoustic modeling in consideration of unknown variation factors at the time of recognition," *Technical Report of IEICE*, SP2003-139, Dec. 2003.

Domestic conference proceedings

[**22**] **Heiga Zen**, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, Tadashi Kitamura, "An accurate modeling method of pitch pattern considering dynamic features," *Proc. of Autumn Meeting of the ASJ*, 1-2-7, Oct. 2001 (In Japanese).

[**23**] Amaro Lima, **Heiga Zen**, Yoshihiko Nankaku, Chiyomi Miyajima, Keiichi Tokuda, Tadashi Kitamura, "On the use of KPCA for feature extraction in speech recognition," *Proc. of Autumn Meeting of the ASJ*, 1-9-4, Sept. 2002.

[**24**] Hiroyuki Suzuki, **Heiga Zen**, Yoshihiko Nankaku, Chiyomi Miyajima, Keiichi Tokuda, Tadashi Kitamura, "Speech recognition using voice characteristic dependent acoustic models," *Proc. of Autumn Meeting of the ASJ*, 1-9-19, Sept. 2002 (In Japanese).

[**25**] Takahiro Hoshiya, Shinji Sako, **Heiga Zen**, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, Tadashi Kitamura, "Improving the performance of HMM-based very low bitrate speech coding," *Proc. of Autumn Meeting of the ASJ*, 1-10-3, Sept. 2002 (In Japanese).

[**26**] Yuka Kishimoto, **Heiga Zen**, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, Tadashi Kitamura, "A postfiltering technique for HMM-based speech synthesis," *Proc. of Autumn Meeting of the ASJ*, 2-1-1, Sept. 2002 (In Japanese).

[**27**] **Heiga Zen**, Takayoshi Yoshimura, Masatsune Tamura, Takashi Masuko, Keiichi Tokuda, "A toolkit for HMM-based speech synthesis," *Proc. of Autumn Meeting of the ASJ*, 3-10-14, Sept. 2002 (In Japanese).

[**28**] Keiichi Tokuda, **Heiga Zen**, Alan W. Black, "An HMM-based approach to English speech synthesis," *Proc. of Autumn Meeting of the ASJ*, 3-10-15, Sept. 2002.

[**29**] Tsuyoshi Ishikawa, Yuko Sawada, **Heiga Zen**, Yoshihiko Nankaku, Chiyomi Miyajima, Keiichi Tokuda, Tadashi Kitamura, "Audio-visual large vocabulary continuous speech recognition based on early integration," *Proc. of Meeting of the Forum of Information Technology*, Sept., 2002 (In Japanese).

[**30**] Yuka Kishimoto, **Heiga Zen**, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, Tadashi Kitamura, "Automatic estimation of postfilter coefficients for HMM-based speech synthesis," *Proc. of Spring Meeting of the ASJ*, 1-6-11, March 2003 (In Japanese).

[**31**] **Heiga Zen**, Keiichi Tokuda, Hisashi Kawai, "Designing texts for speech synthesis corpus by sentence generation from language model," *Proc. of Spring Meeting of the ASJ*, 1-6-18, March 2003 (In Japanese).

[**32**] Ryosuke Tsuduki, **Heiga Zen**, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, Tadashi Kitamura, "A study on HMM-based emotional speech synthesis," *Proc. of Spring Meeting of the ASJ*, 1-6-24, March 2003 (In Japanese).

[**33**] Tsuyoshi Ishikawa, **Heiga Zen**, Yoshihiko Nankaku, Chiyomi Miyajima, Keiichi Tokuda, Tadashi Kitamura, "Bimodal continuous speech recognition using late integration method based on acoustic likelihood rescoring," *Proc. of Spring Meeting of the ASJ*, 3-Q-21, March 2003 (In Japanese).

[**34**] Hiroyuki Suzuki, **Heiga Zen**, Yoshihiko Nankaku, Chiyomi Miyajima, Keiichi Tokuda, Tadashi Kitamura, "Speech recognition using acoustic models depend on noise environments," *Proc. of Autumn Meeting of the ASJ*, vol. I, pp. 25-26, 1-6-13, Sept. 2003 (In Japanese).

[**35**] Yoshihiko Nankaku, **Heiga Zen**, Keiichi Tokuda, Tadashi Kitamura, Takashi Masuko, “Speech synthesis based on variational Bayesian HMM ,” *Proc. of Autumn Meeting of the ASJ*, vol. I, pp. 207-208, 1-8-13, Sept. 2003 (In Japanese).

[**36**] Ranniery da S. Maia, **Heiga Zen**, Keiichi Tokuda, Tadashi, Kitamura, Fernand G. V. Resende Jr. “On the application of HMM-based speech synthesis to Brazilian Portuguese ,” *Proc. of Autumn Meeting of the ASJ*, vol. I, pp. 215-216, 1-8-17, Sept. 2003 (In Japanese).

[**37**] Kikuo Emoto, **Heiga Zen**, Keiichi Tokuda, Tadashi Kitamura, “Accent type recognition for automatic prosodic labeling ,” *Proc. of Autumn Meeting of the ASJ*, vol. I, pp. 225-226, 1-8-22, Sept. 2003 (In Japanese).

[**38**] Ryosuke Tsuduki, **Heiga Zen**, Keiichi Tokuda, Tadashi Kitamura, Murtaza Bulut, Shrikanth S. Narayanan, “A study on emotional speech synthesis based on HMM,” *Proc. of Autumn Meeting of the ASJ*, vol. I, pp. 241-242, 2-6-4, Sept. 2003 (In Japanese).

[**39**] Yohei Itaya, **Heiga Zen**, Yoshihiko Nankaku, Chiyomi Miyajima, Keiichi Tokuda, Tadashi Kitamura, “Continuous speech recognition using HMM based on DAEM algorithm ,” *Proc. of Autumn Meeting of the ASJ*, vol. I, pp. 67-68, 2-6-4, Sept. 2003 (In Japanese).

[**40**] Masaru Sugiura, **Heiga Zen**, Yoshihiko Nankaku, Keiichi Tokuda, Tadashi Kitamura, “Context clustering based on maximum mutual information criterion,” *Proc. of Autumn Meeting of the ASJ*, vol. I, pp. 69-70, 2-6-5, Sept. 2003 (In Japanese).

[**41**] **Heiga Zen**, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, Tadashi Kitamura, “HMM parameter reestimation with explicit duration model for HMM-based speech synthesis,” *Proc. of Spring Meeting of the ASJ*, vol. I, pp. 223-224, 1-7-6, March 2004 (In Japanese).

[**42**] Ryosuke Tsuduki, **Heiga Zen**, Keiichi Tokuda, Tadashi Kitamura, Murtaza Bulut, Shrikanth S. Narayanan, “Emotional speech synthesis based on HMM using subjective evaluation,” *Proc. of Spring Meeting of the ASJ*, vol. I, pp. 251-252, 1-7-20, March 2004 (In Japanese).

[**43**] Amaro Lima, **Heiga Zen**, Yoshihiko Nankaku, Keiichi Tokuda, Tadashi Kitamura, Fernand G. Resende, “Feature extraction for speech recognition using sparse KPCA,” *Proc. of Autumn Meeting of the ASJ*, vol. I, pp. 69-70, 2-1-17, Sept. 2004.

[**44**] Hiroaki Kuwabara, Ryosuke Tsuduki, **Heiga Zen**, Shinji Sako, Yoshihiko Nankaku, Keiichi Tokuda, Tadashi Kitamura, “A study on designing context-dependent labels

based on musical score for HMM-based singing voie synthesis," *Proc. of Autumn Meeting of the ASJ*, vol. I, pp. 323-324, 3-2-4, Sept. 2004 (in Japanese).

[**45**] Yusuke Morioka, Shunsuke Kataoka, **Heiga Zen**, Yoshihiko Nankaku, Keiichi Tokuda, Tadashi Kitamura, "Minutuarization of HMM-based speech synthesis," *Proc. of Autumn Meeting of the ASJ*, vol. I, pp. 325-326, 3-2-5, Sept. 2004 (in Japanese).

[**46**] Shunsuke Kataoka, **Heiga Zen**, Yoshihiko Nankaku, Keiichi Tokuda, Tadashi Kitamura, "Simultaneous backing-off of spectrum, f0 and duration decision trees in HSMM-based speech synthesis," *Proc. of Spring Meeting of the ASJ*, vol. I, pp. 193-194, 1-1-20, March 2005 (in Japanese).

[**47**] Masaru Nakamura, **Heiga Zen**, Tomoki Toda, Keiichi Tokuda, "An evaluation of improvements of HMM-based speech synthesis system for Blizzard Challenge 2005," *Proc. of Autumn Meeting of the ASJ*, 3-6-18, Sept. 2005 (in Japanese).

Research talks

[**48**] **Heiga Zen**, "A trajectory model derived from the HMM by imposing explicit relationship between static and dynamic features for statistical speech recognition and synthesis," Research Seminar, IBM T.J.Watson Research Center, Yorktown Heights, NY, Dec. 2004.

[**49**] **Heiga Zen**, "Reformulating the HMM as a trajectory model by imposing explicit relationship between static and dynamic features," Research Seminar, Microsoft Research, Redmond, WA, June 2005.

[**50**] **Heiga Zen**, "Reformulating the HMM as a trajectory model by imposing explicit relationship between static and dynamic features," SSLI Seminar, University of Washington, Seattle, WA, June 2005.

[**51**] **Heiga Zen**, "Reformulating the HMM as a trajectory model by imposing explicit relationship between static and dynamic features," MIL Speech Seminar, Cambridge University, Cambridge, England, July 2005.

[**52**] **Heiga Zen**, "Reformulating the HMM as a trajectory model by imposing explicit relationship between static and dynamic features," CSTR Seminar, Edinburgh University, Edinburgh, Scotland, July 2005.