

DOCTORAL DISSERTATION

**CONTEXTUAL ADDITIVE STRUCTURES IN
HMM-BASED SPEECH SYNTHESIS**

DOCTOR OF ENGINEERING

FEBRUARY 2014

Shinji TAKAKI

Supervisor : Dr. Keiichi TOKUDA

**Department of Scientific and Engineering Simulation
Nagoya Institute of Technology**

Abstract

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

Most state-of-art speech synthesis systems are based on large amounts of speech data. This type of approach is generally called corpus-based systems. This approach makes it possible to dramatically improve the performance compared with early systems such as rule-based one. In these days statistical approaches based on hidden Markov models (HMMs) have been dominant in TTS, due to their ease of implementation and modeling flexibility. In this approach, the HMMs are used for modeling sequences of speech spectra. In this paper, improved techniques for acoustic modeling are proposed for HMM-based speech synthesis.

It is well known that spectral features are affected by contextual factors, e.g., phoneme identities, accent, parts-of-speech, etc., and extracting the context dependencies is a critical problem for acoustic modeling. One of the major difficulties in the context dependent modeling is finding a good balance between model complexity and availability of training data. In this paper, a novel acoustic modeling is proposed for representing complicated context dependencies.

First, an acoustic modeling with contextual additive structures in HMM-based speech synthesis is proposed. To represent more moderate dependencies between contextual factors and acoustic features, an additive structure of acoustic feature components that have different context dependencies has been proposed for HMM-based speech recognition. Contextual additive structure models can represent complicated dependencies between acoustic features and context labels using multiple decision trees. However, the computational complexity of the context clustering is too high for the full context labels of

speech synthesis. To overcome this problem, this paper proposes two approaches; covariance parameter tying and a likelihood calculation algorithm using the matrix inversion lemma. Additive structure models can be applied to HMM-based speech synthesis using these techniques and speech quality would significantly be improved. Experimental results show that the proposed method outperforms the conventional one in subjective listening tests.

Next, a technique for constructing independent parameter tying structures of mean and variance using additive structure models for HMM-based speech synthesis is proposed. Conventionally, an HMM stream-level tying structure is constructed in HMM-based speech synthesis, i.e., mean vectors and variance matrices have exactly the same parameter tying structure. However, it has been reported that a clustering technique of mean vectors while tying all variance matrices improves the quality of synthesized speech. This indicates that mean and variance parameters should have different optimal tying structures. In the proposed technique, the decision trees for mean and variance parameters are simultaneously grown by taking into account the dependency on mean and variance parameters. Experimental results show that the proposed technique outperforms the conventional one.

Finally, I proposed a spectral modeling technique based on a contextual partial additive structure which provides an efficient representation of context dependencies to acoustic features for HMM-based speech synthesis. The contextual additive structure models assume that the observation vectors are generated from the sum of additive components with tree regression structures and they can be regarded as an intermediate structure between linear regression and tree regression. However, the additive structure models still have a limitation that the number of additive components is fixed for all output probability distributions. The proposed technique is a generalization of the additive structure models which have variable number of additive components dependently on contextual sub-spaces, and the clustering algorithm for extracting partial additive structure is provided. Experimental results show that the proposed technique outperformed the technique of extracting only standard additive structures in a subjective test.

For HMM-based speech synthesis system, above improved techniques were proposed and systems using these techniques improved their performance.

Keywords: Speech synthesis, Hidden Markov Model, Context dependent models, Context clustering, Decision trees, Additive structure, Distribution convolution

Abstract in Japanese

昨今、コンピュータは広く普及し、多くの人が日常的に触れるものとなり、生活に深く影響を及ぼすものとなっている。同時に、ハードウェアの性能の飛躍的な向上により、高性能、高機能なソフトウェアが実現してきている。このような流れの中で、ユーザへの情報提供、ユーザとのコミュニケーションを取ることを目的とした音声対話システムの実用化に向けた試みが数多くされている。その中で音声合成は、音声認識とともに音声対話を実現するための重要な技術となっている。

音声合成における代表的な枠組みとして、音響モデルに統計モデルの一種である隠れマルコフモデル (Hidden Markov Model; HMM) を用いる枠組みがある。HMM は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し、トポロジーを適切に設計可能である。HMM 音声合成では尤度最大化基準に基づく音声パラメータ生成アルゴリズムを用いて直接音声パラメータを出力し音声合成するため、単位選択型の音声合成手法と比べて素片接続歪みが生じない、パラメータを変換することで様々な声質に変換できるなどの特徴がある。

この HMM での音声のモデル化においては、同一の音素でも文脈的な要因 (コンテキスト) を考慮することで、より精度の高いモデルを構築することができることが知られている。しかし、コンテキストの全ての組み合わせを考慮すると組み合わせは膨大となり、学習データとして全てのコンテキストを用意することは不可能である。この問題を解決するため、決定木によるコンテキストクラスタリング手法や線形回帰モデルによるコンテキスト依存性の表現が提案されてきたが、学習データから適切にコンテキスト依存性を抽出することは音声合成の品質に直接影響を与え、依然重要な課題である。本論文では、より高性能な HMM 音声システムの構築のために、より適切なコンテキスト依存性の表現の可能な音響モデル化手法を提案する。

まず、HMM 音声合成のためのコンテキストの加算的構造に基づく音響モデル化を提案する。従来の HMM に基づく音声合成システムでは、各コンテキストラベルに対して、1つの音響特徴量の分布を対応付けている。しかし、音響特徴量は複雑なコンテキスト依存性を持っており、このような直接的なモデル化が必ずしも適切とは言えない。この問題に対して、HMM 音声認識においてコンテキストの加算的構造に基づくモデル化が提案されている。加算構造モデルでは音響特徴量が複数の加

算因子の和として表現され、各加算因子は異なるコンテキスト依存性を持つと仮定している。これにより、音響特徴量に対する変動要因の独立性の表現が可能となり、少量のパラメータで複雑なコンテキスト依存性を表現することができる。しかし、HMM 音声合成では HMM 音声認識と比べ非常に多くのコンテキストを考慮するため、加算構造モデルで用いるコンテキストクラスタリングの計算量が実現不可能なほど膨大になる。この問題を解決するため、本論文では共分散共有と逆行列の補題に基づく尤度計算の2つの計算量削減手法を提案する。これらの手法を用いることで、HMM 音声合成への加算構造モデルの適用が可能となり、合成音声の品質の向上が期待できる。また、客観評価実験、主観評価実験により提案法の有効性を確認した。

また、加算構造モデルの枠組みを用いた平均、分散パラメータの共有構造の同時最適化クラスタリングアルゴリズムを提案する。従来の HMM に基づく音声合成システムでは平均、分散パラメータは同じ共有構造を持つことを仮定している。しかし、一方で全クラスタで分散パラメータを共有することを仮定し、平均パラメータについてのみクラスタリングを行うことで合成音声の品質が改善することが確認されている。このことから、平均、分散パラメータには、それぞれ個別の最適な共有構造が存在すると考えられる。提案法は平均、分散パラメータの依存性を考慮しつつ、同時にそれらの共有構造を構築する。提案法では平均、分散パラメータそれぞれに異なる構造、大きさを持つ決定木を構築することができるため、合成音声の品質の向上が期待できる。また、主観評価実験により提案法の有効性を確認した。

最後に、HMM 音声合成においてより適切で効率の良いコンテキスト依存性の表現のため、コンテキストの部分的な加算構造に基づく音響モデル化を提案する。加算構造モデルでは、音響特徴量は複数の決定木を用いることで複数の加算因子から生成される。そのため、加算構造モデルは決定木に基づくコンテキストクラスタリングによる状態共有と線形回帰モデルの中間に位置すると考えられる。しかし、加算構造モデルにはすべての出力確率分布において固定数の加算因子が用いられるという問題が依然として存在する。本手法は、コンテキストの部分空間に依存して異なる個数の因子を持つことを考慮した、加算構造の一般化手法とみなすことができる。また、そのようなコンテキストの部分的な加算構造を抽出する手法を提案する。

以上のように、本論文ではより高性能な HMM 音声合成システムの構築のために、より適切なコンテキスト依存性の表現の可能なモデルを提案し、その有効性を示す。

Acknowledgment

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

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

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

Contents

List of Tables	ix
List of Figures	x
1 Introduction	1
2 Speech Synthesis based on Hidden Markov Models	5
2.1 Hidden Markov Model	5
2.1.1 Definition of HMM	5
2.1.2 Total output probability of an observation vector sequence	7
2.1.3 Forward-Backward algorithm	8
2.1.4 Searching optimal state sequence	10
2.1.5 Maximum likelihood estimation of HMM parameters	11
2.2 HMM-based speech synthesis	15
2.2.1 Statistical speech synthesis framework	15
2.2.2 Overview of HMM-based speech synthesis	16
2.2.3 Speech parameter generation algorithm	17
2.3 Context Dependent Acoustic Models	20
2.3.1 Context dependency	20
2.3.2 Context Dependent Acoustic Models	22

2.4	Summary	25
3	Acoustic modeling with contextual additive structures	28
3.1	Additive structure models	28
3.1.1	EM algorithm for additive structure models	30
3.1.2	Context clustering for multiple decision trees	33
3.2	Computational complexity reduction in the training algorithm	34
3.2.1	Computational complexity reduction by covariance parameters tying	35
3.2.2	Computational complexity reduction with matrix inversion lemma	36
3.3	Experiments	39
3.3.1	Experimental conditions	39
3.3.2	Objective results	40
3.3.3	Subjective results	41
3.4	Summary	42
4	An optimization algorithm for mean and variance tying structures	48
4.1	Independent Tying Structures for Mean and Variance Parameters	48
4.1.1	Proposed Model Structure	49
4.1.2	Parameter Estimation for the proposed technique	50
4.1.3	Simultaneous Context Clustering for Mean and Variance Parameters	51
4.2	Experiments	52
4.2.1	Experimental conditions	52
4.2.2	Experimental results	52
4.3	Summary	53
5	Acoustic modeling with contextual partial additive structures	55

5.1	Contextual partial additive structure	55
5.1.1	Related model structures	57
5.2	Experiment	58
5.2.1	Experimental conditions	58
5.2.2	Experimental results	59
5.2.3	Objective results	59
5.2.4	Subjective results	61
5.3	Summary	62
6	Conclusions	66
	List of Publications	75
	Journal papers	75
	International conference proceedings	75
	Technical reports	76
	Domestic conference proceedings	76
	Appendix A Software	78

List of Tables

2.1	An example of contexts used in HMM-based speech synthesis.	26
2.1	An example of contexts used in HMM-based speech synthesis (cont.). . .	27
3.1	The total number of parameters (200 sentences).	43
3.2	Avg. likelihood per frame (200 sentences).	43
3.3	The total number of parameters (450 sentences).	44
3.4	Avg. likelihood per frame (450 sentences).	44
3.5	The total number of parameters (1,267 sentences).	45
3.6	Avg. likelihood per frame (1,267 sentences).	45
4.1	Number of leaf nodes and total number of parameters.	53
5.1	Number of decision trees in each state. The number of decision trees In <i>PADD</i> consists of that attached to the root node and internal nodes (200 sentences).	59
5.2	Number of leaf clusters, total number of parameters and average likelihood per frame of training and test data (200 sentences).	59
5.3	Number of decision trees in each state. The number of decision trees In <i>PADD</i> consists of that attached to the root node and internal nodes (450 sentences).	60
5.4	Number of leaf clusters, total number of parameters and average likelihood per frame of training and test data (450 sentences).	60

List of Figures

2.1	Examples of HMM structure.	6
2.2	Implementation of the computation using forward-backward algorithm in terms of a trellis of observations and states.	9
2.3	An overview of a typical HMM-based speech synthesis system.	16
2.4	An example of the relationship between the static feature vector sequence \mathbf{c} and the speech parameter vector sequence \mathbf{o} in a matrix form (the dynamic features are calculated using $L_-^{(1)} = L_+^{(1)} = L_-^{(2)} = L_+^{(2)} = 1$, $w^{(1)}(-1) = -0.5$, $w^{(1)}(0) = 0.0$, $w^{(1)}(1) = 0.5$, $w^{(2)}(-1) = 1.0$, $w^{(2)}(0) = -2.0$, $w^{(2)}(1) = 1.0$).	19
2.5	Fujisaki model.	21
2.6	An example of the decision tree based context clustering.	22
3.1	An example of a contextual additive structure. This outlines the generative process for the triphone feature.	29
3.2	<i>Examples of parameter tying structures constructed by the conventional and the proposed techniques.</i>	35
3.3	<i>An example of splitting a leaf node of a tree.</i>	37
3.4	<i>The relation between \mathbf{G}' and \mathbf{G}''.</i>	38
3.5	<i>Number of leaf nodes for each state (200 sentences). The proposed method only has half the number of parameters in each node because of covariance parameter tying.</i>	43
3.6	<i>Number of leaf nodes for each state (450 sentences). The proposed method only has half the number of parameters in each node because of covariance parameter tying.</i>	44

3.7	<i>Number of leaf nodes for each state (1,267 sentences). The proposed method only has half the number of parameters in each node because of covariance parameter tying.</i>	45
3.8	<i>Spectrograms of test speech and synthesized speech in Conv and Comp3 (450 sentences). Spectrograms corresponding to each component of Comp3 are also shown.</i>	46
3.9	<i>Mean opinion scores for synthesized speech with 95% confidence intervals obtained by conventional and proposed methods (200 sentences).</i>	47
3.10	<i>Mean opinion scores for synthesized speech with 95% confidence intervals obtained by conventional and proposed methods (450 sentences).</i>	47
3.11	<i>Mean opinion scores for synthesized speech with 95% confidence intervals obtained by conventional and proposed methods (1,267 sentences).</i>	47
4.1	<i>Example of parameter tying structures constructed with the conventional and proposed techniques.</i>	49
4.2	<i>Mean opinion scores for synthesized speech obtained by the conventional and proposed techniques.</i>	54
5.1	<i>Examples of standard and partial additive structures.</i>	56
5.2	<i>The effect of an partial additive structure in distribution modeling of acoustic features.</i>	56
5.3	<i>Histograms for the numbers of components for each context dependent model about each state (200 sentences).</i>	63
5.4	<i>Histograms for the numbers of components for each context dependent model about each state (450 sentences).</i>	64
5.5	<i>Mean opinion scores for synthesized speech obtained by conventional, standard and proposed techniques (200 sentences).</i>	65
5.6	<i>Mean opinion scores for synthesized speech obtained by conventional, standard and proposed techniques (450 sentences).</i>	65
A.1	<i>HTS: http://hts.sp.nitech.ac.jp/</i>	78

Chapter 1

Introduction

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

The majority of state-of-the-art speech synthesis systems is trained by using a large amount of speech data. In general, this type of system is called as a corpus-based speech synthesis system [1]. Compared with the previous speech synthesis systems, corpus-based one especially improve the naturalness of synthesized speech. An HMM-based speech synthesis system is major approach to enable machines to speak naturally like humans [2, 3]. In HMM-based speech synthesis, the spectrum, excitation and duration of speech are modeled simultaneously with HMMs, and speech parameter sequences are generated from the HMMs themselves [3]. In HMM-based speech synthesis, the ML criterion has been typically used for training HMMs and generating speech parameters. The ML criterion guarantee that the ML estimates approach the true values of the parameters. In synthesis part, the sequences of spectrum and excitation parameters are generated from the sentence HMM using speech parameter generation algorithm [4–6].

It is well known that spectral features are affected by contextual factors, e.g., phoneme identities, accent, parts-of-speech, etc., and extracting the context dependencies is a critical problem for acoustic modeling. One of the major difficulties in the context dependent modeling is finding a good balance between model complexity and availability of training data. Although increasing the model complexity makes it possible to accurately capture variations in spectral features, the reliability of parameter estimation is degraded

due to decreasing the amount of training data for each model. Furthermore, since it is difficult to prepare training data covering all context dependent models, there are numerous unseen models that are not observed in the training data but that are required in the synthesis phase. To avoid this problem, the decision tree based context clustering has been proposed [7]. In the clustering, HMM states of the context dependent models are grouped into “clusters,” and all states belonging to the same cluster are assumed to have the same distribution. A binary tree is constructed based on the maximum likelihood criterion by applying a phonetic question to each node and iteratively splitting the cluster into two child clusters. By limiting the number of possible splits using prior knowledge, linguistic and articulatory information can be reflected in the clustering results. Instead of the maximum likelihood criterion, the minimum description length (MDL) criterion can be adopted to automatically determine the optimal number of clusters without setting a threshold [8].

Although many researches about structures and training of context dependent models have been carried out, context dependent models is a very important and critical research topic for HMM-based speech synthesis. The context space in the decision tree based context clustering is divided into clusters by contextual factors and the distributions of acoustic features are individually estimated for each cluster. This means that the distributions of each cluster are specified immediately from only training data assigned to the cluster and trained context models have direct dependencies of contexts. On the other hand, the linear regression model [9] is another approach to modeling spectral variations in which all the contextual factors independently affect the acoustic features. Since the combination of contextual factors determines the distribution of spectral features, it can efficiently represent the variety of distributions. However, the dependence among contextual factors is ignored and it is difficult to determine those factors that should additively affect acoustic features. To represent more moderate dependencies between contextual factors and acoustic features, an additive structure of acoustic feature components that have different context dependencies has been proposed for HMM-based speech recognition [10]. This approach includes intermediate structures of decision tree based context clustering and linear regression models as special cases. Since the output probability distribution is composed of the sum of the mean vectors and covariance matrices of additive components, a number of different distributions can be efficiently represented by a combination of fewer distributions. However, it is unknown what kinds of contexts have additive dependencies on acoustic features. To solve this problem, a context clustering algorithm for the additive structure that automatically extracts additive components by simultaneously constructing multiple decision trees has been proposed [10]. Moreover, it can automatically determine an appropriate number of additive components. It has been reported that contextual additive structures are very effective for HMM-based speech recognition [10].

In this paper, a technique for applying additive structure models to HMM-based speech synthesis is proposed. Labels using multiple decision trees. Although additive structure models would significantly be effective for HMM-based speech synthesis as well as for recognition, it is difficult to apply the additive structure models to HMM-based speech synthesis due to the high computational cost caused by context labels, which are richer than the triphone context used in speech recognition. This problem is critical for extracting the additive structure in context clustering with multiple decision trees. To reduce the computational complexity, I propose the techniques to deal with the following three major problems: 1) As mean parameters depend on covariance parameters, those parameters should be simultaneously or iteratively updated until a convergence, 2) A gradient method is required to estimate covariance parameters because no closed form analytical solution has been found, and 3) A matrix whose dimension depends on the number of leaves in the decision trees should be treated when estimating mean parameters. The first and second problems are solved by covariance parameter tying [11]. Tying all covariance matrices of all additive components, the mean parameters can be estimated independently of the covariance matrix and the tied covariance can be analytically estimated. Although covariance parameter tying is a strong approximation, it has been reported that the context clustering of mean parameters assuming the tied covariance can improve the speech quality [11], and tying covariance parameters would be effective for additive structure models as well as for the conventional HMMs. For the third problem, an efficient likelihood calculation technique based on the matrix inversion lemma is proposed. This technique eliminates the redundancy of the context clustering; the likelihood calculation after node splitting includes very similar matrix inversions when different questions are applied at the same leaf node. Additive structure models for HMM-based speech synthesis can be achieved using these two proposed approaches.

Moreover, this paper proposes a technique for constructing independent parameter tying structures of mean and variance using additive structure models in HMM-based speech synthesis. Conventionally, an HMM stream-level tying structure is constructed in HMM-based speech synthesis, i.e., mean vectors and variance matrices have exactly the same parameter tying structure. However, it may not be always appropriate that mean and variance parameters have the same tying structure. As an example, the effectiveness of a technique for context clustering mean vectors while tying all variance matrices was confirmed [11]. In this technique, the synthesized speech can be expected to improve by constructing different tying structures for both mean and variance parameters. However, some degree of freedom for variance parameters may be necessary for improving the quality of synthesized speech. In this paper, it is assumed that both mean and variance parameters have their own tying structure and the construction of appropriate parameter tying structures is examined. In the clustering algorithm, it is necessary to simultaneously

construct each parameter tying structure due to the dependency on mean and variance parameters. Although such a context clustering algorithm can be derived by expanding the conventional context clustering algorithm, a context clustering algorithm is derived using the fact that simultaneous context clustering of mean and variance parameters can be regarded as a special case of context clustering in additive structure models.

Finally, a spectral modeling technique based on a contextual partial additive structure for HMM-based speech synthesis is proposed. The additive structure models still have a limitation that the number of additive components is fixed for all output probability distributions, though the number of components can be automatically determined through the context clustering for the additive structure models. However, it is natural to assume that an appropriate number of additive components depends on contexts. That is, it is expected that some context dependent models require many additive components to represent variations in acoustic features and others do not. To represent such context dependencies appropriately, a technique which enable us to extract additive components affecting arbitrary contextual sub-spaces as well as the entire contextual space is proposed. In the proposed clustering algorithm, the partial additive components are created on demand at an arbitrary node in the context clustering to increase the likelihood. Therefore, the number of additive components corresponding to each context dependent model is automatically determined from the resultant structure of decision trees. The model structure with various number of additive components yields larger combination of components than the standard additive structure with the same number of parameters. This means that it can effectively represent the context dependencies with a limited amount of the training data.

Chapter 2

Speech Synthesis based on Hidden Markov Models

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

2.1 Hidden Markov Model

2.1.1 Definition of HMM

An HMM [12–14] is a finite state machine which generates a sequence of discrete time observations. At each frame it changes states according to its state transition probability distributions, and then generates an observation at time t , \mathbf{o}_t , according to its output probability distribution of the current state. Therefore, the HMM is a doubly stochastic random process model.

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

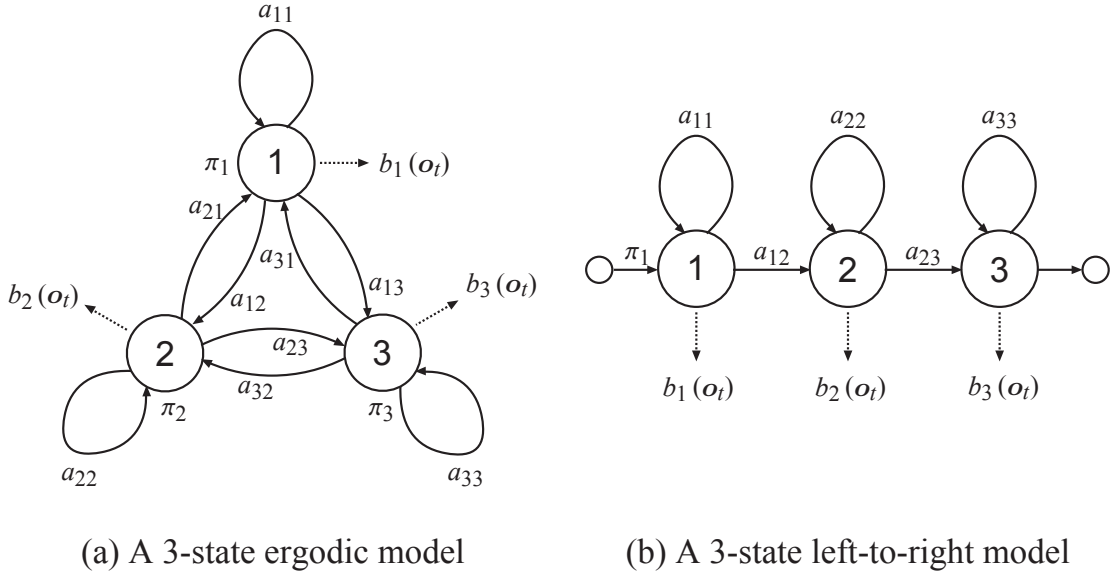


Figure 2.1: Examples of HMM structure.

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

Figure 2.1 shows examples of the HMM structure. Figure 2.1(a) shows a 3-state ergodic model, in which every state of the model could be reached from every state of the model in a single step, and Figure 2.1(b) shows a 3-state left-to-right model, in which the state index increases or stays the same state as time increases. The left-to-right HMMs are generally used to model speech parameter sequences, since they can appropriately model signals.

The output probability distributions $\{b_j(\cdot)\}_{j=1}^N$ can be discrete or continuous depending on the observations. In continuous distribution HMM (CD-HMM), each output probability distribution is usually modeled by a mixture of multivariate Gaussian components [15] as follows:

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

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

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

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

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

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

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

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

2.1.2 Total output probability of an observation vector sequence

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

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

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

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

The order of $2T \cdot N^T$ calculation is required, since at every $t = 1, 2, \dots, T$ there are N possible states that can be reached (i.e., there are N^T possible state sequences). This

calculation is computationally infeasible, even for small values of N and T ; e.g., for $N = 5$ (states), $T = 100$ (observations), there are on the order of $2 \cdot 100 \cdot 5^{100} \approx 10^{72}$ computations. Fortunately, there is an efficient algorithm to calculate Eq. (2.8) using forward and backward procedures.

2.1.3 Forward-Backward algorithm

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

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

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

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

1. Initialization

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

2. Recursion

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

3. Termination

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

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

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

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

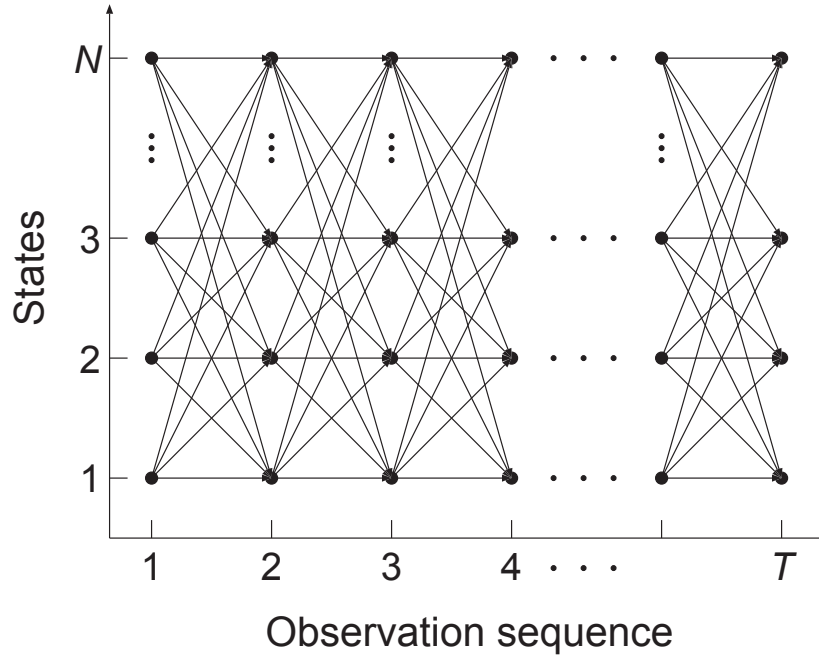


Figure 2.2: Implementation of the computation using forward-backward algorithm in terms of a trellis of observations and states.

1. Initialization

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

2. Recursion

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

3. Termination

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

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

$$p(\mathbf{o} | \Lambda) = \sum_{j=1}^N \alpha_t(j) \beta_t(j). \quad 1 \leq t \leq T \quad (2.17)$$

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

2.1.4 Searching optimal state sequence

The single optimal state sequence $\hat{\mathbf{q}} = \{\hat{q}_1, \hat{q}_2, \dots, \hat{q}_T\}$ for a given observation vector sequence $\mathbf{o} = \{\mathbf{o}_1, \mathbf{o}_2, \dots, \mathbf{o}_T\}$ is useful for various applications (e.g., decoding, initializing HMM parameters). By using a manner similar to the forward algorithm, which is often referred to as the Viterbi algorithm [16], the optimal state sequence $\hat{\mathbf{q}}$ can be obtained. Let $\delta_t(i)$ be the likelihood of the most likely state sequence ending in the i -th state at time t

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

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

1. Initialization

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

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

2. Recursion

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

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

3. Termination

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

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

4. Back tracking

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

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

2.1.5 Maximum likelihood estimation of HMM parameters

There is no known method to analytically obtain the model parameter set based on the maximum likelihood (ML) criterion to obtain Λ which maximizes its likelihood $p(\mathbf{o} | \Lambda)$ for a given observation sequence \mathbf{o} , in a closed form. Since this problem is a high dimensional nonlinear optimization problem, and there will be a number of local maxima, it is difficult to obtain Λ which globally maximizes $p(\mathbf{o} | \Lambda)$. However, the model parameter set Λ locally maximizes $p(\mathbf{o} | \Lambda)$ can be obtained using an iterative procedure such as the expectation-maximization (EM) algorithm [17], and the obtained parameter set will be appropriately estimated if a good initial estimate is provided.

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

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:

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

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

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

where (q_t, s_t) represents being in the s_t -th sub-state (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 $p(\mathbf{o} | \Lambda)$ monotonically and converge to a certain critical point, since it can be proved that the \mathcal{Q} -function satisfies the following theorems:

- Theorem 1

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

- Theorem 2

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

- Theorem 3

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

Maximization of \mathcal{Q} -function

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

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

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

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

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

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

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

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

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

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

can be derived by Lagrange multipliers or differential calculus as follows [18]

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

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

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

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

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

where $\gamma_t(i)$, $\gamma_t(i, m)$, and $\xi_t(i, j)$ are the probability of being in the j -th state at time t , the probability of being in the m -th sub-state of the i -th state at time t , and the probability of being in the i -th state at time t and j -th state at time $t + 1$, respectively, that is

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

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

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

2.2 HMM-based speech synthesis

2.2.1 Statistical speech synthesis framework

The goal of a text-to-speech system is acoustic speech waveform generation from a word sequence. In general, given word sequence \mathbf{w} is processed by a text analysis module. In this part, contextual factors (e.g., accent, lexical stress, part-of-speech, phrase boundary, etc.) are estimated. Next, a speech waveform is generated by a speech synthesis module.

The majority of state-of-the-art speech synthesis systems is trained by using a large amount of speech data. In general, this type of system is called as a corpus-based speech synthesis system [1]. Compared with the previous speech synthesis systems, corpus-based one especially improve the naturalness of synthesized speech.

One of the major approaches in the corpus-based speech synthesis is unit selection based one [19–21]. In this system, the speech waveform is segmented into the small units, phone, di-phone, syllable, etc.. Next, a unit sequence with minimum target and concatenation costs is selected [20] and connected.

Another major approach is statistical speech synthesis, such as HMM-based one [3] This system generates speech parameter sequence $\mathbf{o} = \{\mathbf{o}_1, \mathbf{o}_2, \dots, \mathbf{o}_T\}$ with the maximum a posteriori (MAP) probability given the sub-word sequence \mathbf{u} as follows:

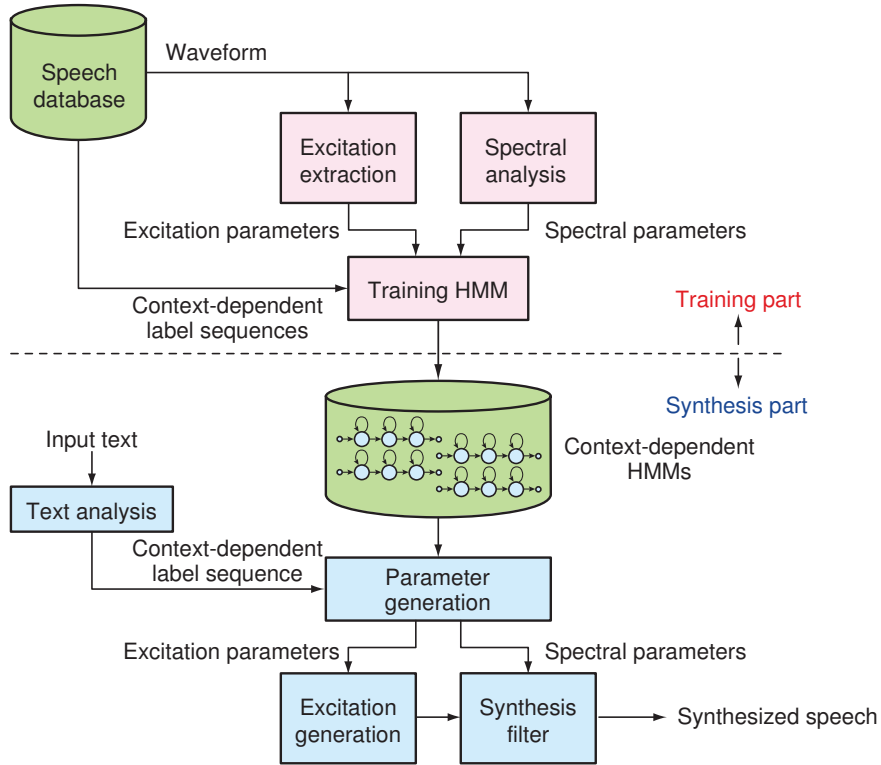


Figure 2.3: An overview of a typical HMM-based speech synthesis system.

$$\hat{o} = \arg \max_{\mathbf{o}} P(\mathbf{o} | \mathbf{u}). \quad (2.44)$$

Eq. (2.44) means that generative models can directly be applied in speech synthesis system. The HMM is the most popular generative models.

2.2.2 Overview of HMM-based speech synthesis

Figure 2.3 shows the HMM-based speech synthesis system [3]. It consists of the training and synthesis part. In the training part, spectrum and excitation parameters are extracted from a speech database. These parameters are modeled by context-dependent HMMs. State duration models are also estimated. In the synthesis part, a sentence HMM is constructed by concatenating the context-dependent HMMs from a given text to be synthesized. In synthesis part, the sequences of spectrum and excitation parameters are generated from the sentence HMM using speech parameter generation algorithm [4–6]. Finally, speech waveform is synthesized from a synthesis filter module. One of the advantage is that voice qualities of synthesized speech can be modified by transforming HMM

parameters. It has been shown that its voice characteristics can be modified by speaker adaptation [22], speaker interpolation [23], or eigenvoice technique [24].

2.2.3 Speech parameter generation algorithm

Problem

For a sentence HMM Λ_u corresponding to a given sub-word sequence u , the speech synthesis problem is to obtain an output vector sequence consisted of spectral and excitation parameters.

$$\mathbf{o} = \{\mathbf{o}_1, \mathbf{o}_2, \dots, \mathbf{o}_T\} \quad (2.45)$$

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

$$\begin{aligned} \hat{\mathbf{o}} &= \arg \max_{\mathbf{o}} p(\mathbf{o} | \Lambda_u) \\ &= \arg \max_{\mathbf{o}} \sum_{\text{all } \mathbf{q}} p(\mathbf{o}, \mathbf{q} | \Lambda_u) \\ &= \arg \max_{\mathbf{o}} \sum_{\text{all } \mathbf{q}} p(\mathbf{o} | \mathbf{q}, \Lambda_u) P(\mathbf{q} | \Lambda_u) \end{aligned} \quad (2.46)$$

$$\mathbf{q} = \{(q_1, s_1), (q_2, s_2), \dots, (q_T, s_T)\} \quad (2.47)$$

where, \mathbf{q} and (q_t, s_t) represent a sub-state sequence and the s_t -th sub-state of the q_t -th state, respectively. This problem is approximated by a Viterbi approximation, because there is not method to analytically obtain \mathbf{o} which maximizes $p(\mathbf{o} | \Lambda_u)$ in a closed form. As a result, this maximization problem can be separated into two stages: finding the best sub-state sequence $\hat{\mathbf{q}}$ for given Λ_u and obtaining \mathbf{o} which maximizes $p(\mathbf{o} | \mathbf{q}, \Lambda_u)$ with respect to \mathbf{o} , i.e.,

$$\hat{\mathbf{q}} = \arg \max_{\mathbf{q}} P(\mathbf{q} | \Lambda_u), \quad (2.48)$$

$$\hat{\mathbf{o}} = \arg \max_{\mathbf{o}} p(\mathbf{o} | \hat{\mathbf{q}}, \Lambda_u). \quad (2.49)$$

The optimization of Eq. (2.48) is performed using explicit state duration models [25] in the HMM-based speech synthesis system. If the output vector \mathbf{o}_t is independent from previous and next frames, the output vector sequence \mathbf{o} which maximize $p(\mathbf{o} | \mathbf{q}, \Lambda_u)$ is obtained as a sequence of mean vectors of sub-states. This causes discontinuity in the output vector sequence at transitions of sub-states. To avoid this problem, dynamic features have been introduced. It is assumed that the output vector \mathbf{o}_t consists of a static feature vector

$$\mathbf{c}_t = [c_t(1), \dots, c_t(K)]^\top \quad (2.50)$$

and its dynamic features, that is

$$\mathbf{o}_t = [\mathbf{c}_t^\top, \Delta \mathbf{c}_t^\top, \Delta^2 \mathbf{c}_t^\top]^\top, \quad (2.51)$$

where $\Delta \mathbf{c}_t$ and $\Delta^2 \mathbf{c}_t$ are delta and delta-delta coefficients, respectively. They are calculated as follows:

$$\Delta \mathbf{c}_t = \sum_{\tau=-L_-^{(1)}}^{L_+^{(1)}} w^{(1)}(\tau) \mathbf{c}_{t+\tau}, \quad (2.52)$$

$$\Delta^2 \mathbf{c}_t = \sum_{\tau=-L_-^{(2)}}^{L_+^{(2)}} w^{(2)}(\tau) \mathbf{c}_{t+\tau}. \quad (2.53)$$

Solution for the Problem

First, the output vector sequence \mathbf{o} and the static feature vector sequence \mathbf{c} can be rewritten as follows:

$$\mathbf{o} = [\mathbf{o}_1^\top, \mathbf{o}_2^\top, \dots, \mathbf{o}_T^\top]^\top, \quad (2.54)$$

$$\mathbf{c} = [\mathbf{c}_1^\top, \mathbf{c}_2^\top, \dots, \mathbf{c}_T^\top]^\top. \quad (2.55)$$

Then, the relationship between \mathbf{c} and \mathbf{o} can be expressed in a matrix form (Figure 2.4) as follows:

$$\mathbf{o} = \mathbf{W} \mathbf{c}, \quad (2.56)$$

where, \mathbf{W} is a regression window matrix given by

$$\mathbf{W} = [\mathbf{W}_1, \mathbf{W}_2, \dots, \mathbf{W}_T]^\top \otimes \mathbf{I}_{M \times M}, \quad (2.57)$$

$$\mathbf{W}_t = [\mathbf{w}_t^{(0)}, \mathbf{w}_t^{(1)}, \mathbf{w}_t^{(2)}], \quad (2.58)$$

$$\mathbf{w}_t^{(0)} = \left[\underbrace{0, \dots, 0}_{t-1}, 1, \underbrace{0, \dots, 0}_{T-t} \right]^\top, \quad (2.59)$$

$$\mathbf{w}_t^{(1)} = \left[\underbrace{0, \dots, 0}_{t-L_-^{(1)}-1}, w^{(1)}(-L_-^{(1)}), \dots, w^{(1)}(0), \dots, w^{(1)}(L_+^{(1)}), \underbrace{0, \dots, 0}_{T-(t+L_+^{(1)})} \right]^\top, \quad (2.60)$$

$$\mathbf{w}_t^{(2)} = \left[\underbrace{0, \dots, 0}_{t-L_-^{(2)}-1}, w^{(2)}(-L_-^{(2)}), \dots, w^{(2)}(0), \dots, w^{(2)}(L_+^{(2)}), \underbrace{0, \dots, 0}_{T-(t+L_+^{(2)})} \right]^\top, \quad (2.61)$$

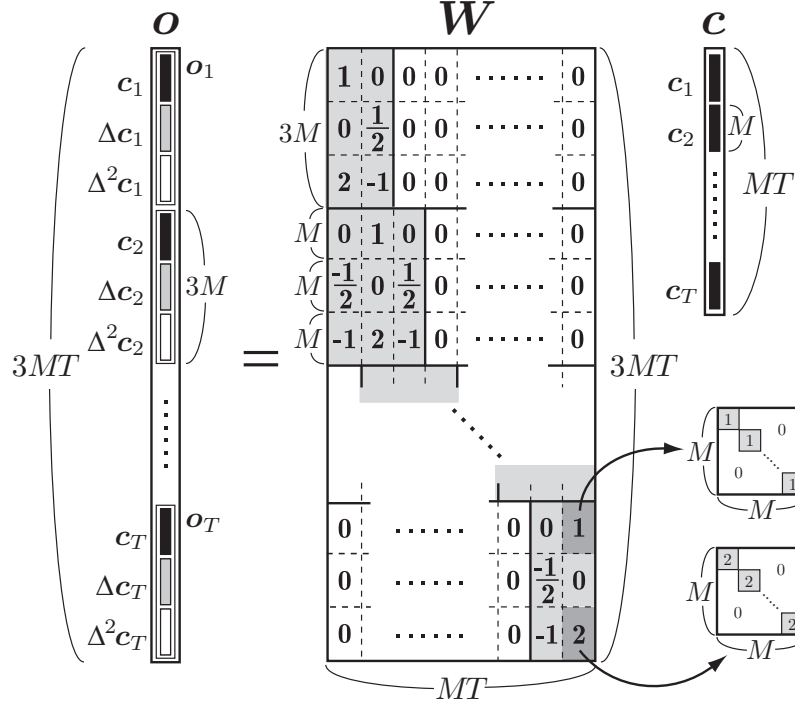


Figure 2.4: An example of the relationship between the static feature vector sequence \mathbf{c} and the speech parameter vector sequence \mathbf{o} in a matrix form (the dynamic features are calculated using $L_-^{(1)} = L_+^{(1)} = L_-^{(2)} = L_+^{(2)} = 1$, $w^{(1)}(-1) = -0.5$, $w^{(1)}(0) = 0.0$, $w^{(1)}(1) = 0.5$, $w^{(2)}(-1) = 1.0$, $w^{(2)}(0) = -2.0$, $w^{(2)}(1) = 1.0$).

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

$$p(\mathbf{o} | \mathbf{q}, \Lambda_u) = \prod_{t=1}^T \mathcal{N}(\mathbf{o}_t | \boldsymbol{\mu}_{q_t s_t}, \boldsymbol{\Sigma}_{q_t s_t}), \quad (2.62)$$

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. Eq. (2.62) can be rewritten as an output probability of \mathbf{o} from a single Gaussian component, that is

$$p(\mathbf{o} | \mathbf{q}, \Lambda_u) = \mathcal{N}(\mathbf{o} | \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q), \quad (2.63)$$

where, $\boldsymbol{\mu}_q$ and $\boldsymbol{\Sigma}_q$ are super-vector and super-matrix corresponding to entire sub-state sequence \mathbf{q} , that is

$$\boldsymbol{\Sigma}_q = \text{diag}[\boldsymbol{\Sigma}_{q_1 s_1}, \boldsymbol{\Sigma}_{q_2 s_2}, \dots, \boldsymbol{\Sigma}_{q_t s_t}], \quad (2.64)$$

$$\boldsymbol{\mu}_q = [\boldsymbol{\mu}_{q_1 s_1}^\top, \boldsymbol{\mu}_{q_2 s_2}^\top, \dots, \boldsymbol{\mu}_{q_t s_t}^\top]^\top. \quad (2.65)$$

Therefore, the logarithm of Eq. (2.62) can be written as

$$\log \mathcal{N}(\mathbf{o} \mid \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q) = -\frac{1}{2} \left\{ 3KT \log 2\pi + \log |\boldsymbol{\Sigma}_q| + (\mathbf{o} - \boldsymbol{\mu}_q)^\top \boldsymbol{\Sigma}_q^{-1} (\mathbf{o} - \boldsymbol{\mu}_q) \right\}. \quad (2.66)$$

Under the condition in Eq. (2.56), maximizing $\mathcal{N}(\mathbf{o} \mid \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)$ with respect to \mathbf{o} is equivalent to that with respect to \mathbf{c} . By setting

$$\frac{\partial \log \mathcal{N}(\mathbf{o} \mid \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)}{\partial \mathbf{c}} = \mathbf{0}_{KT}, \quad (2.67)$$

a set of linear equations can be obtained

$$\mathbf{R}_q \mathbf{c} = \mathbf{r}_q, \quad (2.68)$$

where, $\mathbf{0}_{KT}$ is a KT -dimensional zero vector, \mathbf{R}_q and \mathbf{r}_q are given as

$$\mathbf{R}_q = \mathbf{W} \boldsymbol{\Sigma}_q^{-1} \mathbf{W}, \quad (2.69)$$

$$\mathbf{r}_q = \mathbf{W} \boldsymbol{\Sigma}_q^{-1} \boldsymbol{\mu}_q. \quad (2.70)$$

Since \mathbf{R}_q is a $KT \times KT$ matrix, $O(K^3 T^3)$ operations are required for solution of Eq. (2.68). Eq. (2.68) can be solved by the Cholesky with $O(K^3 L^2 T)$ operations by utilizing the special structure of \mathbf{R}_q . Eq. (2.68) can also be solved by an algorithm derived in [4–6], which can operate in a time-recursive manner [26].

2.3 Context Dependent Acoustic Models

I introduce context dependent acoustic models in this Section. Firstly, context dependency is described.

2.3.1 Context dependency

It is well known that contextual factors, e.g., phoneme identities, accent, parts-of-speech, etc., affect acoustic features. In normal fluent speech every instance of a particular sound can be different. For example, it is well known that prosodic information such as F_0 is affected by multiple contextual factors [27]. One of the most famous models for the generative process of a F_0 contour is the Fujisaki model [28]. Figure 2.5 is a conceptual diagram of this model. It is assumed that the superposition of three components, i.e. a

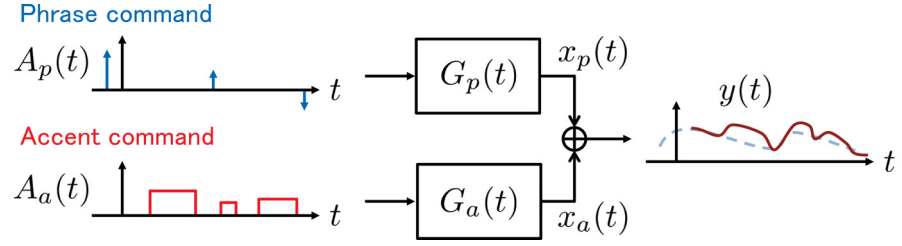


Figure 2.5: Fujisaki model.

phrase component $x_p(t)$, an accent component $x_a(t)$, and a base component x_b , represents an F_0 contour on a logarithmic scale $y(t)$ as follows:

$$y(t) = x_p(t) + x_a(t) + x_b. \quad (2.71)$$

In this model, the phrase commands with a function $A_p(t)$ are assumed to be impulses applied to the phrase control mechanism to generate the phrase components, while the accent commands with a function $A_a(t)$ are assumed to be positive stepwise functions applied to the accent control mechanism to generate the accent components. These two components are modeled as the outputs of second-order critically damped filters.

$$x_p(t) = G_p(t) * A_p(t), \quad (2.72)$$

$$G_p(t) = \begin{cases} \alpha^2 t e^{-\alpha t} & (t \geq 0) \\ 0 & (t < 0) \end{cases}, \quad (2.73)$$

$$x_a(t) = G_a(t) * A_a(t), \quad (2.74)$$

$$G_a(t) = \begin{cases} \beta^2 t e^{-\beta t} & (t \geq 0) \\ 0 & (t < 0) \end{cases}, \quad (2.75)$$

where $*$ denotes convolution over time. Parameters α and β are natural angular frequency of the phrase and accent control mechanism and assumed to be constant at least within an utterance. The phrase component $x_p(t)$ consists of the major-scale pitch variation, the accent component $x_a(t)$ consists of the smaller-scale pitch variations in accented syllables, and the baseline component x_b is a constant value related to the lower bound of the speaker's F_0 . This means that the phrase and the accent, i.e. contexts, affect acoustic features F_0 .

Table 2.1 shows an example of contexts for English used in HMM-based speech synthesis. In HMM-based speech recognition contextual factors about next and previous phonemes are typically used. However, in HMM-based speech synthesis enormous contextual factors, which are richer than the triphone context, are used. This is because that richer models are required for speech synthesis than recognition. To improve modeling

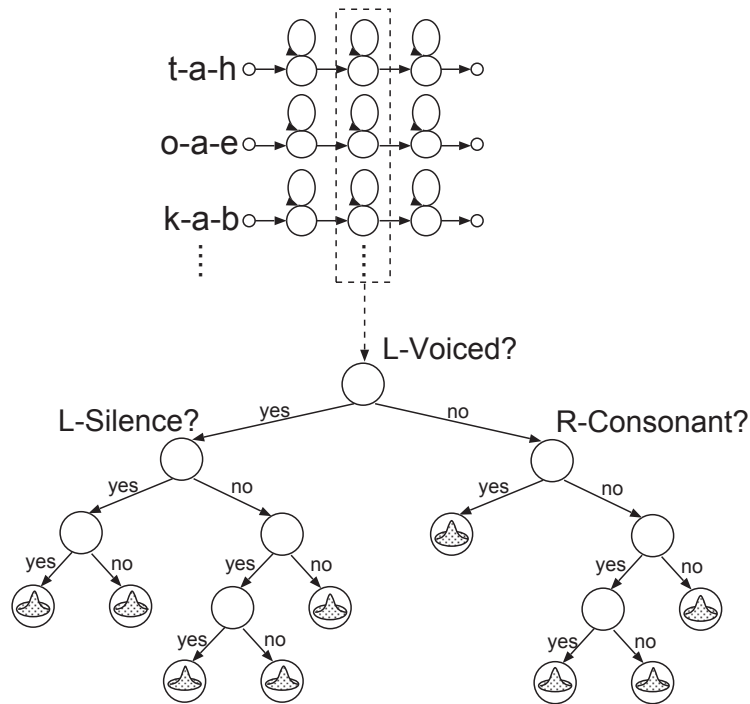


Figure 2.6: An example of the decision tree based context clustering.

accuracy variations of acoustic features caused by contextual effects should be taken account. In order to accurately capture the variations of acoustic features, context dependent acoustic models [7, 29] are widely used in HMM-based speech synthesis. However, this models produce a data insufficiency problem because context dependent acoustic models have a large number of model parameters. Furthermore, the data is usually unevenly spread. Sharing models across different contexts is a traditional method of dealing with these problem. Although a large number of context dependent acoustic models can capture variations in speech data, too many model parameters lead to the over-fitting problem. Consequently, maintaining a good balance between model complexity and the amount of training data is very important for obtaining a high generalization performance. I introduce typical context dependent models in next subsection.

2.3.2 Context Dependent Acoustic Models

Decision Tree based Context Clustering

The decision tree based context clustering [30] is an efficient method for estimating robust model parameters of context dependent models. Figure 2.6 shows an example of the

decision tree based context clustering. In this clustering technique, top-down clustering is performed to locally maximize the likelihood of parameters with respect to the training data using pre-defined questions about contexts. Then, mean vectors and covariance matrices of HMM states clustered on the same leaf node are tied. The four steps in the procedure for the decision tree based context clustering algorithm are as follows:

Step 1. Create the root node and compute its likelihood.

Step 2. Evaluate questions at the root node or two nodes created by previous splitting. The likelihood after the node is split is calculated by estimating the ML parameters of new nodes created by splitting.

Step 3. Select the pair of a node and a question that gives the maximum likelihood and then split the node into two by applying the question. The model parameters of new created nodes are updated by the ML parameters calculated in Step 1.

Step 4. If the change of likelihood after the node is split is below a predefined threshold, stop the procedure. Otherwise, go to Step 2.

In decision tree based context clustering, the total log likelihood is as follows as \mathbf{o}_t is an acoustic feature vector at time t :

$$\mathcal{L} = \sum_{t=1}^T \sum_{c \in C} \gamma_t(c) \log \mathcal{N}(\mathbf{o}_t | \boldsymbol{\mu}_{f(c)}, \boldsymbol{\Sigma}_{f(c)}) \quad (2.76)$$

where C denotes all contexts observed in the training data, $\gamma_t(c)$ is the state occupancy probability with respect to context c , and $\boldsymbol{\mu}_{f(c)}$ and $\boldsymbol{\Sigma}_{f(c)}$ represent the mean vector and the covariance matrix associated with the leaf node, respectively. In the decision based context clustering, single Gaussian distributions are typically used for output probability distributions. The function $f(c)$ gives the index of the leaf nodes in the decision tree. In Eq (2.76) the state index is ignored. The mean vector $\boldsymbol{\mu}_{f(c)}$ and the covariance matrix $\boldsymbol{\Sigma}_{f(c)}$ can be estimated using the ML criterion. In HMM-based speech synthesis, the minimum description length (MDL) criterion [8] is widely used to automatically control the size of decision trees. The context space in the decision tree based context clustering is divided into clusters by contextual factors and the distributions of acoustic features are individually estimated for each cluster.

Linear Regression

The linear regression model [9] is another approach to modeling acoustic variations in which all the contextual factors independently affect the acoustic features. In the linear

regression model, the context dependencies are represented by a linear model in which an acoustic feature is decomposed into a context independent vector, a context-dependent component and a residual vector. The context-dependent component is given by the product of a weight matrix and an context vector. It is well known that the acoustic feature is varied by contexts. To represent the effect, in the linear regression model an acoustic feature vector \mathbf{o}_t at time t is represented as follows:

$$\mathbf{o}_t = \mathbf{a}_p + \mathbf{B}_p \mathbf{z}_t + \boldsymbol{\epsilon}_t \quad (2.77)$$

where \mathbf{a} , \mathbf{B} , \mathbf{z}_t and $\boldsymbol{\epsilon}_t$ represent a context independent vector, a weight matrix to the context vector, a context vector, and a residual vector respectively. In Eq. (2.77), templates, e.g. context independent phonemes, are used [31, 32] and p represents the number of templates. If the residual vector $\boldsymbol{\epsilon}_t$ is distributed according to the multivariate normal distribution with a $\mathbf{0}$ mean vector and a covariance matrix $\boldsymbol{\Sigma}_p$ and \mathbf{z}_t is given, the parameters, \mathbf{a}_p , \mathbf{B}_p and $\boldsymbol{\Sigma}_p$, can be estimated using the ML criterion. The total log likelihood is as follows:

$$\mathcal{L} = \sum_{t=1}^T \sum_{c \in C} \gamma_t(c) \log \mathcal{N}(\mathbf{o}_t | \mathbf{a}_p + \mathbf{B}_p \mathbf{z}_t, \boldsymbol{\Sigma}_p). \quad (2.78)$$

By letting the partial derivation of Eq. (2.78) with respect to \mathbf{a}_p , \mathbf{B}_p or $\boldsymbol{\Sigma}_p$ equal zeros, the solutions can be obtained using the generalized inverse.

Since the combination of contextual factors determines the acoustic feature, it can efficiently represent the variety of distributions. However, the dependence among contextual factors is ignored and it is difficult to determine those factors that should additively affect acoustic features. Although in [9] three kinds of the context vectors are described as below, it is difficult to heuristically find the best structure because there are numerous contexts.

Bottom-up

In a bottom-up model the context vector at t -th frame is obtained from the feature vectors as

$$\mathbf{z}_t^\top = \mathbf{z}_t^{(x)\top} = [o_{t+\Delta_x(1)}^\top, \dots, o_{t+\Delta_x(M_x)}^\top] \quad (2.79)$$

where M_x is the number of feature vectors extracted as acoustic contexts and $\Delta_x(m)$ represents a relative position of the m -th acoustic feature. In this case \mathbf{z} is an $M_x \times N$ dimensional vector,

Top-down model

In the top-down model the context vector is obtained from a phoneme sequence as

$$\mathbf{z}_t^\top = \mathbf{z}_t^{(y)\top} = [e(p_{t+\Delta_y(1)})^\top, \dots, e(p_{t+\Delta_y(M_y)})^\top] \quad (2.80)$$

where q_t is the phoneme at time t , $e(q)$ represents a unit P -dimensional vector whose row corresponded to q has a one, $\Delta_y(m)$ represents the relative position of the m -th phoneme and M_y is the number of phonemes extracted as contexts.

Combined model

In the combined model the context vector is defined by a concatenation of the context vectors in above two models.

$$\mathbf{z}_t^\top = [\mathbf{z}_t^{(x)\top}, \mathbf{z}_t^{(y)\top}] \quad (2.81)$$

2.4 Summary

In this chapter, the basic theories of the hidden Markov models (HMMs), its algorithm for calculating the output probability (forward-backward algorithm), searching the optimal state sequence (Viterbi algorithm), and estimating its parameters (EM algorithm) are described. And then, the HMM-based speech synthesis system and context dependent models are described. Following chapter will derive a acoustic modeling with contextual additive structures for HMM-based speech synthesis.

Table 2.1: An example of contexts used in HMM-based speech synthesis.

the phoneme identity before the previous phoneme
the previous phoneme identity
the current phoneme identity
the next phoneme identity
the phoneme after the next phoneme identity
position of the current phoneme identity in the current syllable (forward)
position of the current phoneme identity in the current syllable (backward)
whether the previous syllable stressed or not (0: not stressed, 1: stressed)
whether the previous syllable accented or not (0: not accented, 1: accented)
the number of phonemes in the previous syllable
whether the current syllable stressed or not (0: not stressed, 1: stressed)
whether the current syllable accented or not (0: not accented, 1: accented)
the number of phonemes in the current syllable
position of the current syllable in the current word (forward)
position of the current syllable in the current word (backward)
position of the current syllable in the current phrase (forward)
position of the current syllable in the current phrase (backward)
the number of stressed syllables before the current syllable in the current phrase
the number of stressed syllables after the current syllable in the current phrase
the number of accented syllables before the current syllable in the current phrase
the number of accented syllables after the current syllable in the current phrase
the number of syllables from the previous stressed syllable to the current syllable
the number of syllables from the current syllable to the next stressed syllable
the number of syllables from the previous accented syllable to the current syllable
the number of syllables from the current syllable to the next accented syllable
name of the vowel of the current syllable
whether the next syllable stressed or not (0: not stressed, 1: stressed)
whether the next syllable accented or not (0: not accented, 1: accented)
the number of phonemes in the next syllable
gpos (guess part-of-speech) of the previous word
the number of syllables in the previous word
gpos (guess part-of-speech) of the current word
the number of syllables in the current word
position of the current word in the current phrase (forward)
position of the current word in the current phrase (backward)
the number of content words before the current word in the current phrase
the number of content words after the current word in the current phrase
the number of words from the previous content word to the current word
the number of words from the current word to the next content word

Table 2.1: An example of contexts used in HMM-based speech synthesis (cont.).

gpos (guess part-of-speech) of the next word
the number of syllables in the next word
the number of syllables in the previous phrase
the number of words in the previous phrase
the number of syllables in the current phrase
the number of words in the current phrase
position of the current phrase in this utterance (forward)
position of the current phrase in this utterance (backward)
TOBI endtone of the current phrase
the number of syllables in the next phrase
the number of words in the next phrase
the number of syllables in this utterance
the number of words in this utterance
the number of phrases in this utterance

Chapter 3

Acoustic modeling with contextual additive structures

In this chapter, an acoustic modeling with contextual additive structures for HMM-based speech synthesis is described. In additive structure models, a more complex structure, i.e., the additive structure of acoustic feature components is considered. Contextual additive structure models can represent complicated dependencies between acoustic features and context labels using multiple decision trees. However, the computational complexity of the context clustering is too high for the full context labels of speech synthesis. To overcome this problem, this paper proposes two approaches; covariance parameter tying and a likelihood calculation algorithm using the matrix inversion lemma.

3.1 Additive structure models

In additive structure models, an acoustic feature vector \mathbf{o}_t at time t is generated by the sum of additive components:

$$\mathbf{o}_t = \sum_{n=1}^N \mathbf{o}_t^{(n)} \quad (3.1)$$

where $\mathbf{o}_t^{(n)}$ denotes the n -th additive component. If each component is independent and generated according to a Gaussian distribution, the probabilistic density function of acoustic features is represented by the convolution of the additive components [33] so that

$$\begin{aligned} P(\mathbf{o}_t | c_t, \lambda) &= \int \prod_{n=1}^N \mathcal{N}(\mathbf{o}_t^{(n)} | \boldsymbol{\mu}_{c_t}^{(n)}, \boldsymbol{\Sigma}_{c_t}^{(n)}) d\mathbf{o}_t^{(1)} \cdots \mathbf{o}_t^{(N-1)} \\ &= \mathcal{N}(\mathbf{o}_t | \boldsymbol{\mu}_{c_t}, \boldsymbol{\Sigma}_{c_t}) \end{aligned} \quad (3.2)$$

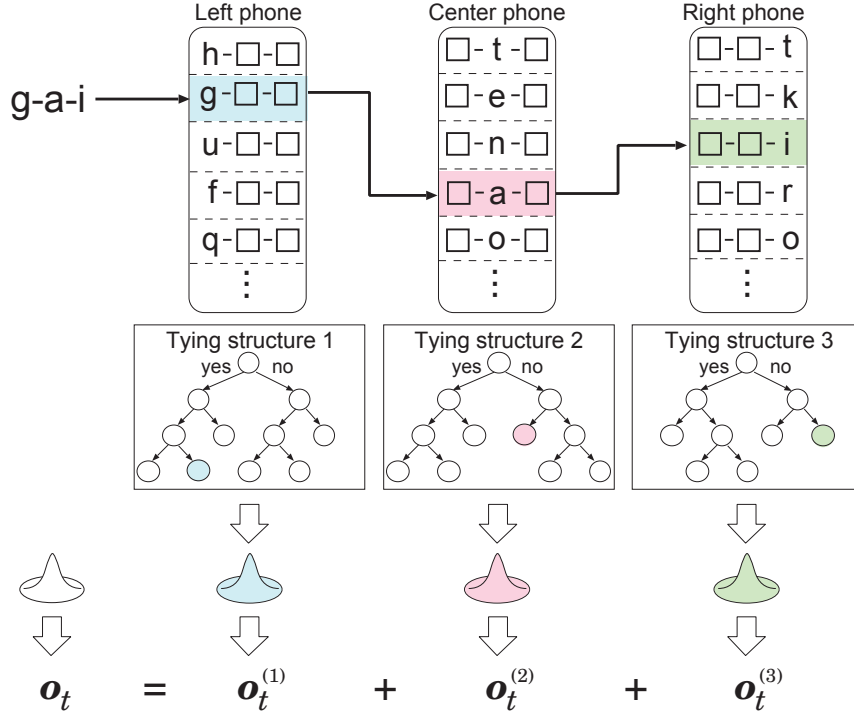


Figure 3.1: An example of a contextual additive structure. This outlines the generative process for the triphone feature.

where $\boldsymbol{\mu}_{c_t}^{(n)}$ and $\boldsymbol{\Sigma}_{c_t}^{(n)}$ are respectively the mean vector and covariance matrix of the n -th component $\mathbf{o}_t^{(n)}$ given a context c_t . The output probability distribution is a Gaussian distribution whose mean vector and covariance matrix are respectively given as

$$\boldsymbol{\mu}_{c_t} = \sum_{n=1}^N \boldsymbol{\mu}_{c_t}^{(n)}, \quad \boldsymbol{\Sigma}_{c_t} = \sum_{n=1}^N \boldsymbol{\Sigma}_{c_t}^{(n)} \quad (3.3)$$

Since each additive component $\mathbf{o}_t^{(n)}$ has different context dependencies, it is assumed that each component has a different decision tree that represents tying structures of model parameters $\boldsymbol{\mu}_{c_t}$ and $\boldsymbol{\Sigma}_{c_t}$.

Although it is unknown which kinds of contexts have additive dependencies on acoustic features in practice, an example of a contextual additive structure of triphone HMMs to explain the effective of the additive structure is present. Here, it is assumed that the left, center, and right phones are the contexts of additive components. Figure 3.1 outlines the generative process for the triphone feature. The generative process for acoustic features is as follows: first, the component of a given monophone (center phone) context is generated from a corresponding distribution obtained by descending the tree. Then, the additive components of the left and right contexts are also generated independently from each

distribution and added to the monophone feature.

The effective of the additive structure depends on whether acoustic features really have an additive structure for contexts. When acoustic features have an additive structure, a number of different distributions can be efficiently represented by a combination of fewer distributions. Furthermore, it is also effective to predict the acoustic features of unseen contexts. Although in the conventional method unseen models are assigned to one of the clusters in the decision tree, the proposed method can construct the distribution for unseen contexts, which are different from any distributions of observed contexts.

3.1.1 EM algorithm for additive structure models

The Maximum Likelihood (ML) parameters of additive component distributions can be estimated with the EM algorithm. In the E-step, since the convolved output probability distribution becomes a Gaussian distribution, the standard forward-backward algorithm can simply be applied as in the standard HMMs. However, there is difficulty in the M-step due to the dependencies among additive component distributions.

Using the statistics obtained by the E-step, the Q -function with respect to the output probability distribution can be written as

$$\begin{aligned}
\mathcal{L} &= \sum_{t=1}^T \sum_{c \in C} \gamma_t(c) \log P(\mathbf{o}_t | c_t = c, \lambda) \\
&= \sum_{t=1}^T \sum_{c \in C} \gamma_t(c) \log \mathcal{N}(\mathbf{o}_t | \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) \\
&= -\frac{1}{2} \sum_{c \in C} \left[\sum_{t=1}^T \gamma_t(c) (K \log 2\pi + \log |\boldsymbol{\Sigma}_c|) \right. \\
&\quad \left. + Tr \left\{ \boldsymbol{\Sigma}_c^{-1} \sum_{t=1}^T \gamma_t(c) (\mathbf{o}_t - \boldsymbol{\mu}_c) (\mathbf{o}_t - \boldsymbol{\mu}_c)^\top \right\} \right] \\
&= -\frac{1}{2} \sum_{c \in C} \tilde{T}_c \left[K \log 2\pi + \log |\boldsymbol{\Sigma}_c| \right. \\
&\quad \left. + Tr \left\{ \boldsymbol{\Sigma}_c^{-1} \left(\tilde{\boldsymbol{\Sigma}}_c + (\boldsymbol{\mu}_c - \tilde{\boldsymbol{\mu}}_c)(\boldsymbol{\mu}_c - \tilde{\boldsymbol{\mu}}_c)^\top \right) \right\} \right] \tag{3.4}
\end{aligned}$$

where K is the dimensionality of feature vectors and C denotes all contexts observed in the training data. The state index is ignored. The statistics with respect to context c are

represented by $(\tilde{\cdot})_c$ and each of the statistics is calculated as follows:

$$\tilde{T}_c = \sum_{t=1}^T \gamma_t(c), \quad \tilde{\boldsymbol{\mu}}_c = \frac{1}{\tilde{T}_c} \sum_{t=1}^T \gamma_t(c) \boldsymbol{o}_t \quad (3.5)$$

$$\tilde{\boldsymbol{\Sigma}}_c = \frac{1}{\tilde{T}_c} \sum_{t=1}^T \gamma_t(c) (\boldsymbol{o}_t - \tilde{\boldsymbol{\mu}}_c) (\boldsymbol{o}_t - \tilde{\boldsymbol{\mu}}_c)^\top \quad (3.6)$$

In [10], the updating the parameters of a particular additive component has been proposed. To represent the tree structure, a function $f^{(n)}(c)$ is introduced that gives the index of the Gaussian distribution (number of leaves in the decision tree) of the n -th additive components for c . Using this function, the mean parameter and the covariance parameter of the convolved distribution are given by

$$\boldsymbol{\mu}_c = \sum_{n=1}^N \boldsymbol{\mu}_{f^{(n)}(c)}, \quad \boldsymbol{\Sigma}_c = \sum_{n=1}^N \boldsymbol{\Sigma}_{f^{(n)}(c)} \quad (3.7)$$

The derivative of the \mathcal{Q} -function with respect to the mean and covariance of the particular additive component can be written as

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}_m} = \sum_{c \in \phi_m^{(g(m))}} \tilde{T}_c \boldsymbol{\Sigma}_c^{-1} (\tilde{\boldsymbol{\mu}}_c - \boldsymbol{\mu}_c) \quad (3.8)$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Sigma}_m} = -\frac{1}{2} \sum_{c \in \phi_m^{(g(m))}} \tilde{T}_c \left\{ \boldsymbol{\Sigma}_c^{-1} + \boldsymbol{\Sigma}_c^{-1} \tilde{\boldsymbol{\Sigma}}_c \boldsymbol{\Sigma}_c^{-1} + \boldsymbol{\Sigma}_c^{-1} (\boldsymbol{\mu}_c - \tilde{\boldsymbol{\mu}}_c) (\boldsymbol{\mu}_c - \tilde{\boldsymbol{\mu}}_c)^\top \boldsymbol{\Sigma}_c^{-1} \right\} \quad (3.9)$$

where m is the index of the leaf node and $\phi_m^{(n)}$ denotes the contexts which are included in the m -th cluster, i.e., $\phi_m^{(n)} = \{c \mid f^{(n)}(c) = m\}$. The function $g(m)$ gives the index of the component of the m -th cluster. It can be seen from the above equations that updating of $\boldsymbol{\mu}_m$ and $\boldsymbol{\Sigma}_m$ requires the parameters of the other additive component (decision trees). Hence, all parameters of all trees are dependent on each other to compose the output probabilities. This means that all parameters of all trees should be estimated simultaneously or iteratively until a convergence.

Although there are the several update procedures for this optimization problem, in [10] the iterative update of each parameter is used. This technique estimates each parameter while keeping the other parameters fixed. If the other parameters are fixed, the mean can be easily estimated by setting the derivative to zero. However, the update of covariance matrices is difficult to solve analytically. Therefore, one of the gradient methods is needed for the covariance update and the Newton method is applied in [10]. In [10], iteratively update of all mean vectors of all trees until a convergence and the same update of covariance matrices are selected for the update process. These update processes for mean and covariance parameters are also iterated until a convergence.

In this paper, the update process of mean parameters is different from the technique used in [10]. Mean parameters of all leaf nodes of all decision trees are updated by solving a set of linear equations. For simplicity of notation, Σ_c is the diagonal covariance matrix and one of dimensions of feature vectors is focused. Under this assumption, the covariance parameter is σ_c and the mean parameters of all leaf nodes are $\boldsymbol{\mu} = [\mu_1, \dots, \mu_M]^\top$, where M is the sum of all leaf nodes of all decision trees. Then, Eq. (4.5) can be rewritten as

$$\mathcal{L} = -\frac{1}{2} \sum_{c \in \mathcal{C}} \tilde{T}_c \left\{ \log 2\pi + \log |\sigma_c| + \frac{\tilde{\sigma}_c + (\mu_c - \tilde{\mu}_c)^2}{\sigma_c} \right\} \quad (3.10)$$

The terms with respect to $\boldsymbol{\mu}$ of \mathcal{L} can be rewritten as

$$\begin{aligned} \mathcal{L} &\propto -\frac{1}{2} \sum_{c \in \mathcal{C}} \tilde{T}_c \left(\frac{\mu_c^2 + -2\mu_c \tilde{\mu}_c}{\sigma_c} \right) \\ &= -\frac{1}{2} \sum_{c \in \mathcal{C}} \tilde{T}_c \frac{1}{\sigma_c} \left\{ \left(\sum_{n=1}^N \mu_{f^{(n)}(c)} \right)^2 - 2 \left(\sum_{n=1}^N \mu_{f^{(n)}(c)} \right) \tilde{\mu}_c \right\} \\ &= -\frac{1}{2} (\boldsymbol{\mu}^\top \mathbf{G} \boldsymbol{\mu} - 2\boldsymbol{\mu}^\top \mathbf{k}) \end{aligned} \quad (3.11)$$

where

$$\mathbf{G} = \begin{bmatrix} g_{1,1} & \cdots & g_{1,M} \\ \vdots & \ddots & \vdots \\ g_{M,1} & \cdots & g_{M,M} \end{bmatrix}, \quad \mathbf{k} = \begin{bmatrix} k_1 \\ \vdots \\ k_M \end{bmatrix} \quad (3.12)$$

$$g_{m_1, m_2} = g_{m_2, m_1} = \sum_{c \in \phi_{m_1}^{(g(m_1))} \cap \phi_{m_2}^{(g(m_2))}} \tilde{T}_c \frac{1}{\sigma_c} \quad (3.13)$$

$$k_{m_1} = \sum_{c \in \phi_{m_1}^{(g(m_1))}} \tilde{T}_c \frac{1}{\sigma_c} \tilde{\mu}_c \quad (3.14)$$

Since \mathbf{G} is a symmetric matrix, the first partial derivative of Eq. (3.11) with respect to $\boldsymbol{\mu}$ can be written as

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}} &= -\frac{1}{2} \{ (\mathbf{G} + \mathbf{G}^\top) \boldsymbol{\mu} - 2\mathbf{k} \} \\ &= -\mathbf{G} \boldsymbol{\mu} + \mathbf{k} \end{aligned} \quad (3.15)$$

By setting Eq. (3.15) to 0, the solution of $\boldsymbol{\mu}$ is given as follows:

$$\mathbf{G} \boldsymbol{\mu} = \mathbf{k} \quad (3.16)$$

However, \mathbf{G} is typically a singular matrix. Therefore, to solve Eq. (3.16), a Moore-Penrose generalized inverse is used. Covariance parameters are updated with the same technique as in [10]. Hence, the iterative update of covariance parameters is necessary, though the iterative update of mean parameters is not necessary. The update process for mean and covariance parameters are also iterated until a convergence.

3.1.2 Context clustering for multiple decision trees

A context clustering algorithm for multiple decision trees has been proposed to automatically extract the additive structure from training data [10]. It is easy to construct a decision tree if the other decision trees and their parameters are fixed. However, as the tree structures of the additive components interact with each other to compose the output probabilities, the multiple decision trees for additive components should be constructed simultaneously. The four steps in the procedure for the proposed clustering algorithm are as follows:

- Step 1.** Set the number of trees N to one, create the root node of the first tree and compute its likelihood.
- Step 2.** Evaluate questions at all leaf nodes of all trees and a root node of a new tree. The likelihood after the node is split is calculated by estimating the ML parameters of all leaf nodes of all trees.
- Step 3.** Select the pair of a node and a question that gives the maximum likelihood and then split the node into two by applying the question. The model parameters of all leaf nodes are updated by the ML parameters.
- Step 4.** If the change of likelihood after the node is split is below a predefined threshold, stop the procedure. Otherwise, go to Step 2.

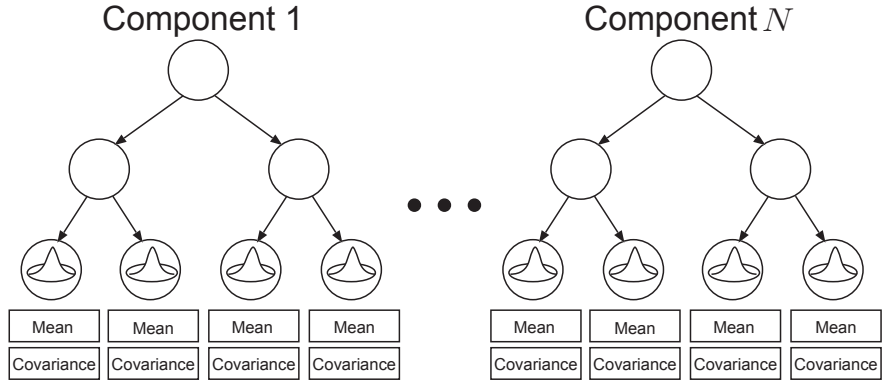
There are some differences from the conventional clustering algorithm in the procedure. First, in Step 2, the ML estimates of all parameters of all trees are required to evaluate questions at a candidate node. In the conventional clustering, the ML parameters of the two nodes that are split can be obtained independently of the other nodes. However, in additive structure models, the change of likelihood before and after the node splitting is calculated not only with the parameters of the nodes created by splitting but also the parameters of the other trees. From the same reason, the likelihood of a candidate node is affected by other nodes in additive structure models. Therefore, all questions should be re-evaluated at all leaf nodes after a node is split. The computational complexity of

selecting the pair of a node and a question for a splitting in the conventional clustering and the clustering for multiple trees are $O(Q \cdot D)$ and $O(Q \cdot D \cdot M^4)$ with a diagonal covariance matrix, respectively, where Q is the number of questions and D is the dimension of the feature. The computational complexity of the clustering for multiple trees is derived from the computational complexity of calculating the solution of Eq. (3.16) for all dimensions, i.e., $O(D \cdot M^3)$, and the number of the evaluations for all questions at all leaf nodes. Furthermore, the computational complexity is also dependent on the number of iteration of the update for mean and covariance parameters in the clustering for multiple trees. The computational complexity between the two techniques are completely different. Second, in the context clustering for multiple decision trees, an appropriate splitting of a leaf node or a root node representing a new tree is selected based on the MDL criterion in STEP 2. A splitting of a root node is equivalent to creating a new component. Therefore, the number of components can be automatically determined based on the MDL criterion.

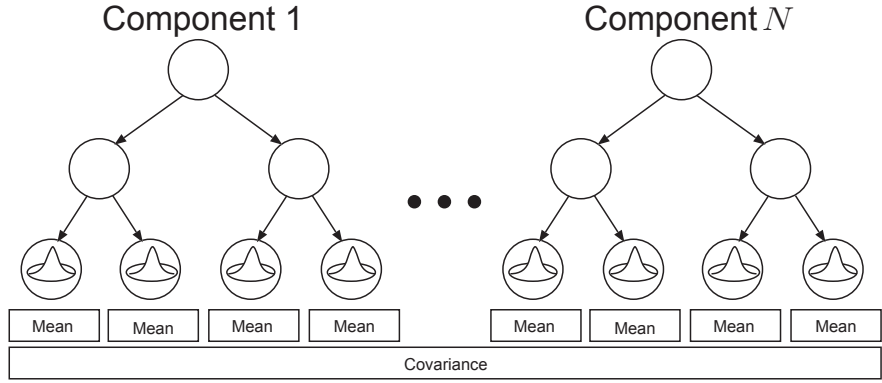
Additive structure models can be regarded as an intermediate model between a decision tree based context dependent model and a linear regression model, and it includes these two models as special cases. If the number of additive components is restricted to one, the model becomes a decision tree based model, and if all trees have only two nodes (only one question is applied), the model is equivalent to a linear regression model. Furthermore, the above clustering algorithm can automatically select the appropriate model structures, i.e., the number of trees and the tree structures, from training data.

3.2 Computational complexity reduction in the training algorithm

In the context clustering for multiple decision trees, the ML parameters of all leaf nodes need to be simultaneously estimated. Moreover, all questions need to be re-evaluated at all leaf nodes after node splitting, because the likelihood gain of all candidate questions are dependently changed by the split node. Since speech synthesis uses richer context labels than speech recognition, the computational complexity becomes enormous to conduct the exact context clustering algorithm for additive structure models. A computational time of more than several years is required to extract an additive structure using the general training data for speech synthesis. Therefore, some approximation techniques are required.



(a) Parameter tying structures constructed by the conventional technique



(b) Parameter tying structures constructed by the proposed technique

Figure 3.2: Examples of parameter tying structures constructed by the conventional and the proposed techniques.

3.2.1 Computational complexity reduction by covariance parameters tying

In additive structure models, mean parameters can be analytically estimated. However, as it is difficult to analytically solve the update of covariance parameters, a gradient method is applied to each covariance parameter. Furthermore, as Eqs. (3.13) and (3.14) indicate that mean parameters depend on covariance parameters, the mean and covariance parameters should be re-estimated until convergence. Therefore, huge computational cost is involved when extracting additive structures.

In this paper, covariance parameter tying is applied to additive structure models. It has been reported that mean parameters are relatively more important than covariance parameters for the quality of HMM-based speech synthesis [11]. The impact on speech

quality in additive structure models caused by the covariance parameter tying would also be small. Figure 3.2 shows examples of parameter tying structures constructed with the conventional technique (Figure 3.2(a)) and the proposed technique (Figure 3.2(b)). By tying covariance parameters, the mean parameters can be updated independently of the covariance parameters and iterative updates are not required. Using the tied covariance parameter Σ_g , the \mathcal{Q} -function with respect to the output probability distribution (Eq. (4.5)) can be rewritten as

$$\begin{aligned} \mathcal{L} = & -\frac{1}{2} \sum_{c \in \mathcal{C}} \tilde{T}_c \left[K \log 2\pi + \log |N\Sigma_g| \right. \\ & \left. + \text{Tr} \left\{ (N\Sigma_g)^{-1} \left(\tilde{\Sigma}_c + (\boldsymbol{\mu}_c - \tilde{\boldsymbol{\mu}}_c)(\boldsymbol{\mu}_c - \tilde{\boldsymbol{\mu}}_c)^\top \right) \right\} \right] \end{aligned} \quad (3.17)$$

The first partial derivative of Eq. (3.17) with respect to Σ_g can be written as

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \Sigma_g} = & -\frac{1}{2} \sum_{c \in \mathcal{C}} \tilde{T}_c \left[\Sigma_g^{-1} - N^{-1} \Sigma_g^{-1} \right. \\ & \left. \cdot \left\{ \tilde{\Sigma}_c + (\boldsymbol{\mu}_c - \tilde{\boldsymbol{\mu}}_c)(\boldsymbol{\mu}_c - \tilde{\boldsymbol{\mu}}_c)^\top \right\} \Sigma_g^{-1} \right] \end{aligned} \quad (3.18)$$

By setting Eq. (3.18) to 0, Σ_g is analytically calculated as follows:

$$\begin{aligned} \Sigma_g = & N^{-1} \left(\sum_{c \in \mathcal{C}} \tilde{T}_c \right)^{-1} \\ & \cdot \sum_{c \in \mathcal{C}} \tilde{T}_c \left\{ \tilde{\Sigma}_c + (\boldsymbol{\mu}_c - \tilde{\boldsymbol{\mu}}_c)(\boldsymbol{\mu}_c - \tilde{\boldsymbol{\mu}}_c)^\top \right\} \end{aligned} \quad (3.19)$$

the log likelihood \mathcal{L} after the parameters are estimated can be written as

$$\mathcal{L} = -\frac{1}{2} \sum_{c \in \mathcal{C}} \tilde{T}_c \left\{ K \log 2\pi + \log |N\Sigma_g| + K \right\} \quad (3.20)$$

3.2.2 Computational complexity reduction with matrix inversion lemma

Since the size of \mathbf{G} depends on the sum of all leaf nodes of all trees in Eq. (3.16), the computational complexity to solve the linear equations becomes enormous. However, when a leaf node is split by different questions, the statistics corresponding to the leaf nodes locally change in newly created nodes and the statistics corresponding to the other nodes are fixed. Figure 3.3 shows an example of such local change of statistics. Since \mathbf{G}

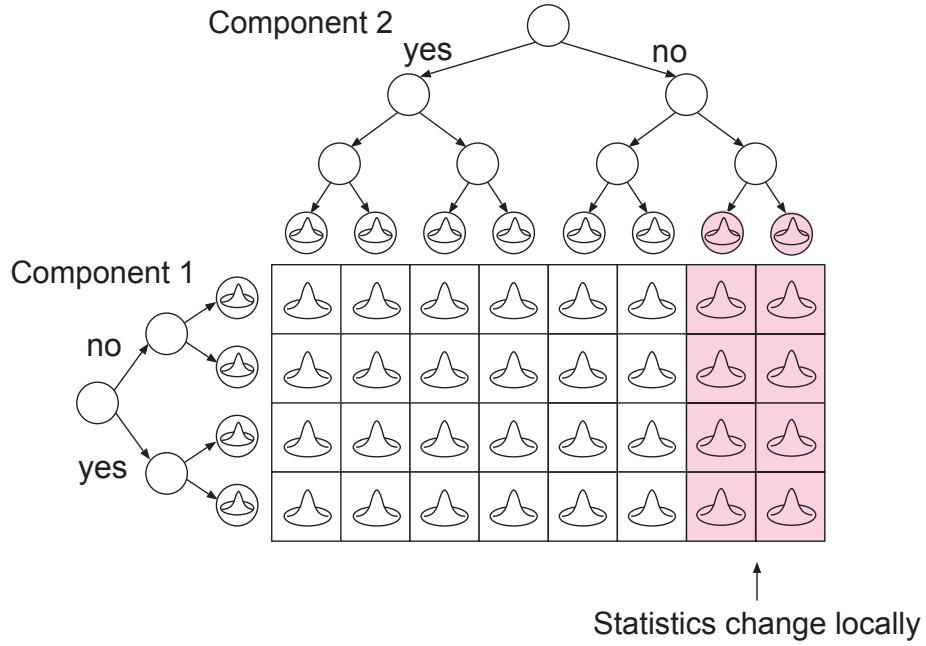


Figure 3.3: An example of splitting a leaf node of a tree.

only becomes dependent on \tilde{T}_c due to covariance parameter tying, many elements of \mathbf{G} do not change among the questions at the same node. The computational complexity can significantly be reduced by using this property.

Assuming that \mathbf{G}' is obtained with one question, and \mathbf{G}'' is obtained with another question at the same node, \mathbf{G}'' can be represented by using \mathbf{G}' as follows:

$$\mathbf{G}'' = \mathbf{G}' + \mathbf{G}^{(d)} \quad (3.21)$$

where $\mathbf{G}^{(d)}$ is a symmetric matrix and can be written as

$$\mathbf{G}^{(d)} = \begin{bmatrix} \mathbf{0} & g_{1,m}^{(d)} & g_{1,m+1}^{(d)} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots \\ g_{m,1}^{(d)} & \cdots & g_{m,m}^{(d)} & g_{m,m+1}^{(d)} & \cdots & g_{m,M}^{(d)} \\ g_{m+1,1}^{(d)} & \cdots & g_{m+1,m}^{(d)} & g_{m+1,m+1}^{(d)} & \cdots & g_{m+1,M}^{(d)} \\ \mathbf{0} & \vdots & \vdots & \mathbf{0} \\ & g_{M,m}^{(d)} & g_{M,m+1}^{(d)} & & & \end{bmatrix} \quad (3.22)$$

where m and $m + 1$ are indexes of leaf nodes created by splitting. A matrix $\mathbf{G}^{(d)}$ is represented by $M \times 4$ and $4 \times M$ matrices i.e., \mathbf{D} and \mathbf{E} , as follows:

$$\mathbf{G}^{(d)} = \mathbf{D}\mathbf{E} \quad (3.23)$$

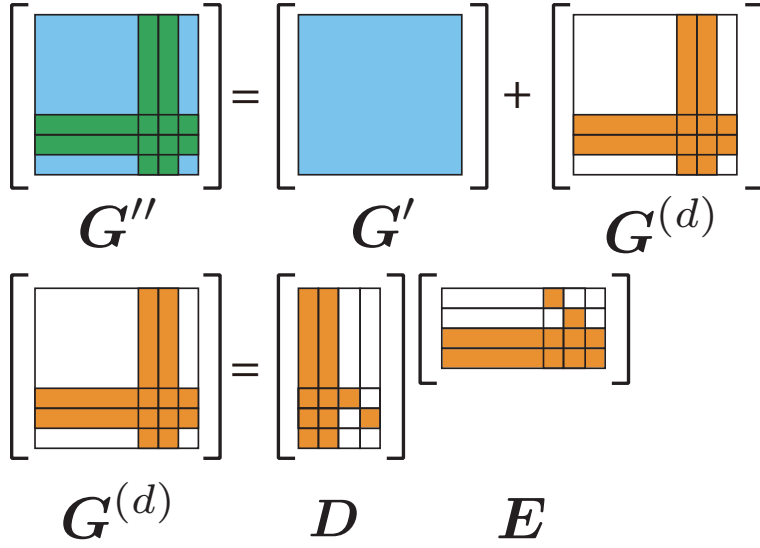


Figure 3.4: The relation between G' and G'' .

$$\begin{aligned}
 D &= [D_1 \quad D_2 \quad D_3 \quad D_4] \\
 &= \begin{bmatrix} g_{1,m}^{(d)} & g_{1,m+1}^{(d)} & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ g_{m-1,m}^{(d)} & g_{m-1,m+1}^{(d)} & 0 & 0 \\ \frac{1}{2}g_{m,m}^{(d)} & \frac{1}{2}g_{m,m+1}^{(d)} & 1 & 0 \\ \frac{1}{2}g_{m+1,m}^{(d)} & \frac{1}{2}g_{m+1,m+1}^{(d)} & 0 & 1 \\ g_{m+2,m}^{(d)} & g_{m+2,m+1}^{(d)} & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ g_{M,m}^{(d)} & g_{M,m+1}^{(d)} & 0 & 0 \end{bmatrix} \quad (3.24)
 \end{aligned}$$

$$E = [D_3 \quad D_4 \quad D_1 \quad D_2]^\top \quad (3.25)$$

Figure 3.4 shows the relation between G' and G'' . Assuming that G'^{-1} is given, G''^{-1} can be calculated as follows:

$$\begin{aligned}
 G''^{-1} &= (G' + DE)^{-1} \\
 &= G'^{-1} - G'^{-1} D \Psi E G'^{-1} \quad (3.26)
 \end{aligned}$$

where $\Psi = (CG'^{-1}B + I)^{-1}$ and I is the identity matrix. Eq. (3.26) is derived using the

following matrix inversion lemma.

$$\begin{aligned}
& (\mathbf{G}'^{-1} - \mathbf{G}'^{-1} \mathbf{D} \Psi \mathbf{E} \mathbf{G}'^{-1}) (\mathbf{G}' + \mathbf{D} \mathbf{E}) \\
&= \mathbf{I} + \mathbf{G}'^{-1} \mathbf{D} \mathbf{E} - \mathbf{G}'^{-1} \mathbf{D} \Psi \mathbf{E} - \mathbf{G}'^{-1} \mathbf{D} \Psi \mathbf{E} \mathbf{G}'^{-1} \mathbf{D} \mathbf{E} \\
&= \mathbf{I} + \mathbf{G}'^{-1} \mathbf{D} \{ \mathbf{E} - \Psi (\mathbf{I} + \mathbf{E} \mathbf{G}'^{-1} \mathbf{D}) \mathbf{E} \} \\
&= \mathbf{I} + \mathbf{G}'^{-1} \mathbf{D} (\mathbf{E} - \Psi \Psi^{-1} \mathbf{E}) \\
&= \mathbf{I}
\end{aligned} \tag{3.27}$$

The size of matrix Ψ is 4×4 in Eq. (3.26). Therefore, it can significantly reduce the computational complexity in comparison with directly calculating the inverse of \mathbf{G}'' .

In the context clustering, this algorithm can be applied to the likelihood calculation of questions at the same leaf node. The matrix \mathbf{G}'^{-1} is calculated from the first question using the Moore-Penrose inverse, and the likelihood of other questions can then be calculated by using Eq. (3.26) with lower computational complexity.

3.3 Experiments

3.3.1 Experimental conditions

Objective and subjective experiments were conducted to evaluate the effectiveness of the proposed method. The 200 and 450 sentences of the phonetically balanced 503 sentences from the ATR Japanese speech database B-set, uttered by male speaker MHT, were used for training. The 1,267 sentences including 450 sentences of the phonetically balanced sentences, uttered by female speaker, was also used for training. The remaining 53 sentences were used for evaluation. The speech data was down-sampled from 20 to 16 kHz and windowed at a frame rate of 5-ms using a 25-ms Blackman window.

The feature vectors consisted of spectral and F_0 feature vectors. The mel-cepstral coefficients were obtained from STRAIGHT spectra [34]. The spectrum parameter vectors consisted of 39 STRAIGHT mel-cepstral coefficients including the zero coefficient and their delta and delta-delta coefficients. The excitation parameter vectors consisted of log F_0 and its delta and delta-delta.

A five-state, left-to-right, no-skip structure with a diagonal covariance matrix was used for the hidden semi-Markov model. Additive structure modeling is applied to only the spectrum parameters, and the excitation parameters were modeled with conventional multi-space probability distributions HMMs [35]. The proposed and the conventional methods

had the same tying structures for the excitation parameters. The MDL criterion was used to determine the size of the decision trees [8].

Five techniques are compared; *Conv* is the conventional decision tree based method and *Comp1* to *Comp3* and *Variable* are the additive structure models trained by the proposed method, where the number after *Comp* represents the number of decision trees and *Variable* automatically determines the number of decision trees in the clustering algorithm. Note that *Comp1* is equivalent to the conventional decision tree based method but covariance parameter tying was applied.

For the subjective experiments, mean opinion score tests were conducted. Ten subjects participated in these listening tests. Twenty sentences were randomly selected from the 53 sentences for each subject. The subjects were asked to rate the naturalness of the synthesized speech on a scale from one (completely unnatural) to five (natural). All experiments were carried out using headphones in a soundproof room.

3.3.2 Objective results

Figures 3.5, 3.6 and 3.7 show bar charts of the number of leaf nodes for each state. In additive structure models, the bars were divided and the length of each division represents the number of leaf nodes of each decision tree. When the conventional and proposed methods have the same number of leaf nodes, the proposed method only has half the number of parameters because of covariance parameter tying. Figure 3.5, 3.6 and 3.7 show that *Comp1* has more leaf nodes than *Conv*. This means that decision trees with respect to the mean parameters are constructed with taking account of tying covariance parameters. Similar to *Comp1*, the number of leaf nodes increases in the additive structure models with multiple decision trees. This is because the MDL criterion was used to determine the size of decision trees and decision trees were constructed to represent variations in acoustic features by only using mean parameters in the additive structure models. Although the size of decision trees differs among additive components, multiple decision trees were split. This suggests that additive structures are inherent in the training data. In *Variable*, the number of decision trees was automatically determined, and it can actually be seen from Figures 3.5, 3.6 and 3.7 that a different number of decision trees was constructed in each state. In the 200 sentence case, a larger number of decision trees was obtained for State1 and State5 than for the middle state of HMMs. This might be because the triphone or quinphone contexts strongly affect the spectral features around phone boundaries. With increasing the amount of training data, the spectral variations caused by other contextual factors were modeled by increasing the number of decision trees in the middle states.

Tables 3.1, 3.3 and 3.5 show the total number of parameters and Tables 3.2, 3.4 and 3.6

show the average likelihoods per frame of training and test data obtained from 200, 450 and 1,267 sentences, respectively. *Conv* obtained the highest likelihood among the five techniques in both training and test data. This is because covariance parameter tying was not applied to *Conv* and the number of parameters is larger than the other techniques. It can also be seen that the likelihoods of additive structure models (*Comp2*, *Comp3*, and *Variable*) were higher than *Comp1*. In addition, the likelihoods tended to increase with increasing additive components. This is because the number of parameters was slightly increased with increasing additive components and multiple additive components were appropriate for representing the spectral variations. However, *Comp3* and *Variable* obtained almost the same values in the 200 sentences. This is because three additive components were enough for 200 training sentences and *Variable* estimates automatically appropriate the number of additive components dependently on the amount of training data. With increasing the amount of training data, a larger number of additive components are needed for capturing the spectral variations and the likelihood is increased in *Variable* from *Comp3*.

Figure 3.8 shows spectrograms of test speech and synthesized speech in *Conv* and *Comp3* with 450 sentences. Spectrograms corresponding to each component of *Comp3* are also shown. From this figure, it can be observed that three components additively affect to the resultant spectrogram of *Comp3*. It can be also seen that component affects different frequencies, e.g., it seems that components represent different formants. For examples, component 1 represents formants at about 0.7, 1.2, etc seconds and component 2 represents at about 0.6, 2.6, etc seconds. However, the relation to the contextual factors is unknown and further analysis will be required in future work.

3.3.3 Subjective results

Figures 5.5, 5.6 and 3.11 show the results of MOS tests using 200, 450 and 1,267 training sentences, respectively. It can be seen from the figures that *Conv* and *Comp1* obtained almost the same score. This confirmed that the impact of speech quality by tying covariance parameters is small. Although *Conv* obtained the highest likelihood, *Conv* and *Comp1* obtained almost the same subjective score. This is because that mean parameters are relatively more important than covariance parameters for the speech quality, though covariance parameter contributed greatly to the likelihood. It can also be observed that additive structure models (*Comp2*, *Comp3*, and *Variable*) achieved better subjective scores than the conventional methods (*Conv* and *Comp1*). This means that the additive structure models appropriately extracted context dependencies from training data and they were effectively used to predict spectral features of unseen contexts. Similar to the objective evaluation, the scores tended to increase with increasing additive components. “*Variable*”

obtained the highest scores in the case of 200, 450 and 1,267 sentences. This is because the proposed method appropriately selected the model structures including the number of decision trees dependently on the amount of training data.

3.4 Summary

In this chapter, I proposed an efficient training algorithm for additive structure models and applied it to HMM-based speech synthesis. Additive structure models are significantly effective for extracting the context dependencies and accurately capturing variations in spectral features. However, it is difficult to apply this model to HMM-based speech synthesis due to its computational complexity caused by richer context labels. Covariance parameter tying in each state and using the matrix inversion lemma can significantly reduce the amount of computational complexity and allow us to apply additive structure models to HMM-based speech synthesis. In experiments, the proposed method outperformed the conventional method. Additive structure modeling for prosodic information such as F0 will be a future work, because F0 has an additive structure with multiple contextual factors [27]. The proposed method for F0 would significantly improve synthesized speech quality.

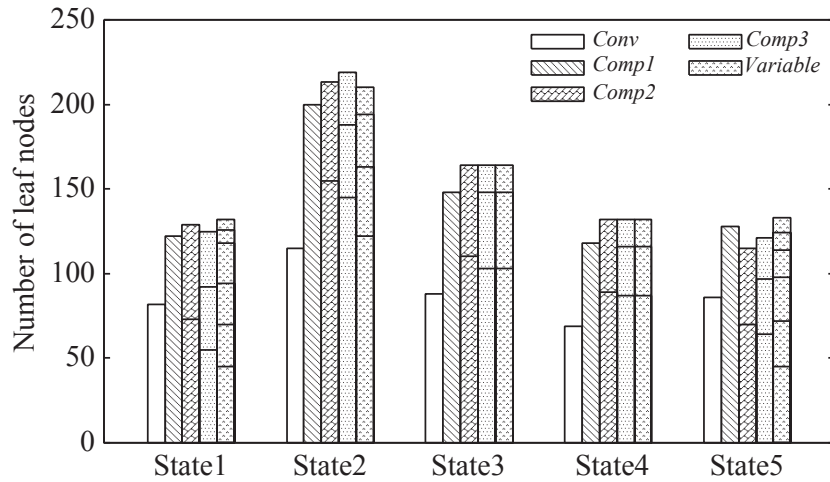


Figure 3.5: Number of leaf nodes for each state (200 sentences). The proposed method only has half the number of parameters in each node because of covariance parameter tying.

Table 3.1: The total number of parameters (200 sentences).

<i>Conv</i>	<i>Comp1</i>	<i>Comp2</i>	<i>Comp3</i>	<i>Variable</i>
105,600	86,520	90,960	91,920	93,120

Table 3.2: Avg. likelihood per frame (200 sentences).

	Avg. likelihood (training)	Avg. likelihood (test)
<i>Conv</i>	139.66	131.62
<i>Comp1</i>	131.87	124.30
<i>Comp2</i>	132.36	124.86
<i>Comp3</i>	132.52	124.86
<i>Variable</i>	132.60	124.97

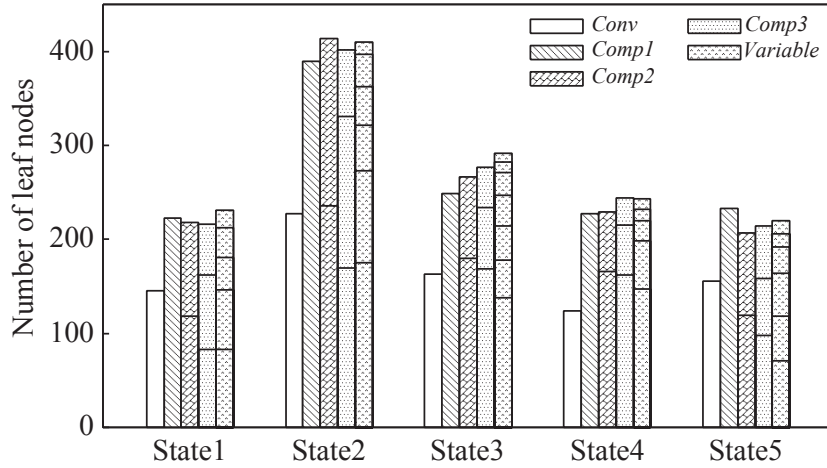


Figure 3.6: Number of leaf nodes for each state (450 sentences). The proposed method only has half the number of parameters in each node because of covariance parameter tying.

Table 3.3: The total number of parameters (450 sentences).

<i>Conv</i>	<i>Comp1</i>	<i>Comp2</i>	<i>Comp3</i>	<i>Variable</i>
195,600	159,120	160,680	162,960	168,120

Table 3.4: Avg. likelihood per frame (450 sentences).

	Avg. likelihood (training)	Avg. likelihood (test)
<i>Conv</i>	138.70	136.12
<i>Comp1</i>	131.15	129.13
<i>Comp2</i>	131.51	129.53
<i>Comp3</i>	131.78	129.80
<i>Variable</i>	132.12	130.05

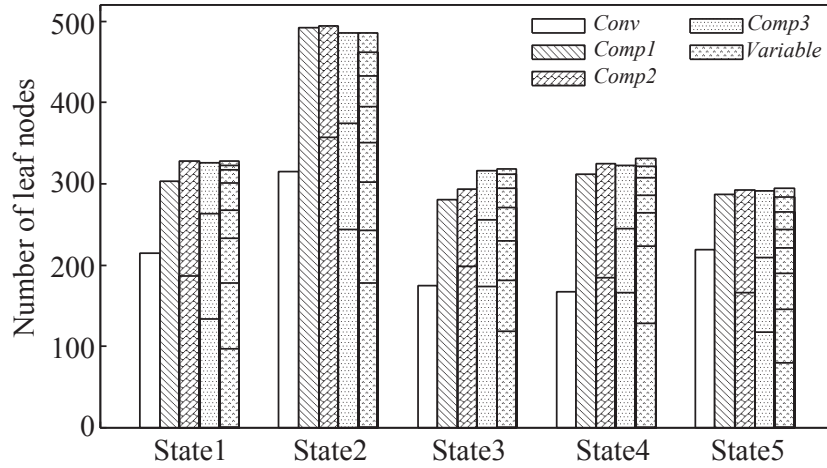


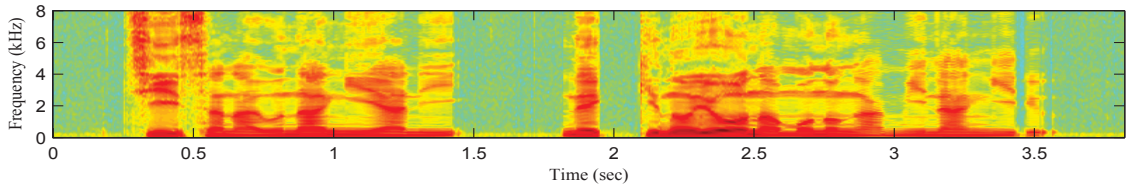
Figure 3.7: Number of leaf nodes for each state (1,267 sentences). The proposed method only has half the number of parameters in each node because of covariance parameter tying.

Table 3.5: The total number of parameters (1,267 sentences).

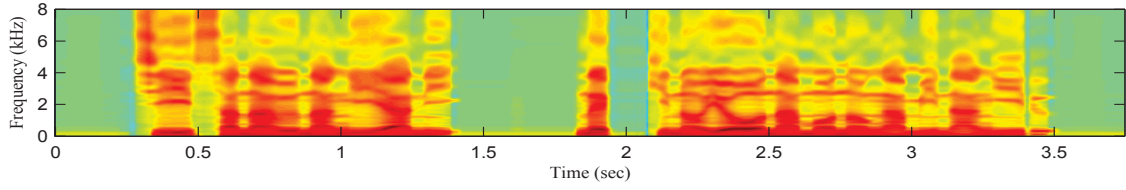
<i>Conv</i>	<i>Comp1</i>	<i>Comp2</i>	<i>Comp3</i>	<i>Variable</i>
261,840	201,480	208,440	209,640	211,320

Table 3.6: Avg. likelihood per frame (1,267 sentences).

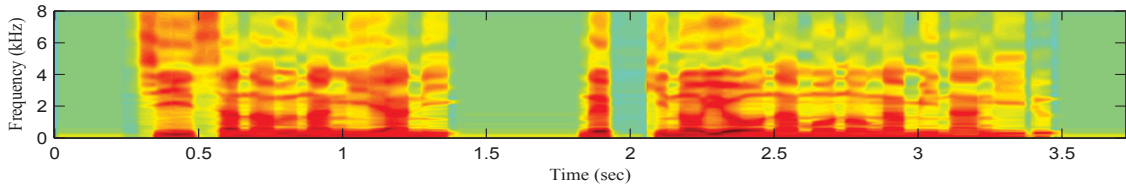
	Avg. likelihood (training)	Avg. likelihood (test)
<i>Conv</i>	120.84	116.90
<i>Comp1</i>	116.42	113.04
<i>Comp2</i>	116.59	113.21
<i>Comp3</i>	116.67	113.21
<i>Variable</i>	116.81	113.34



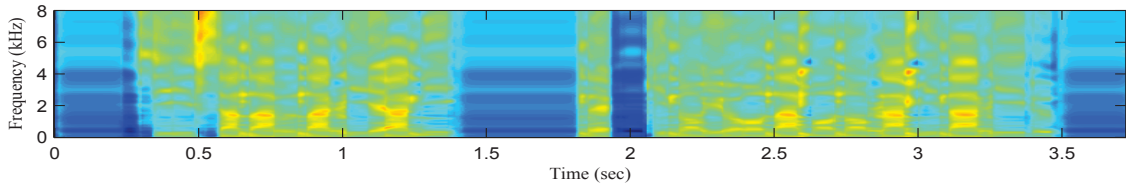
(a) Test speech



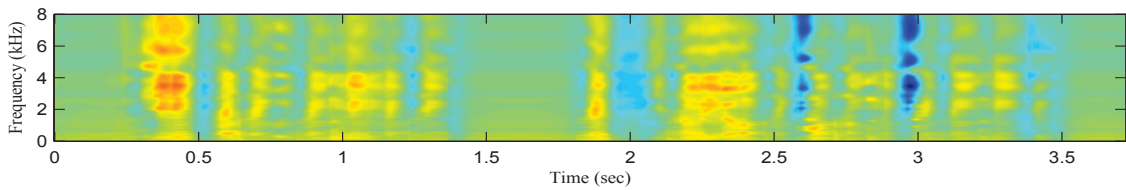
(b) *Conv*



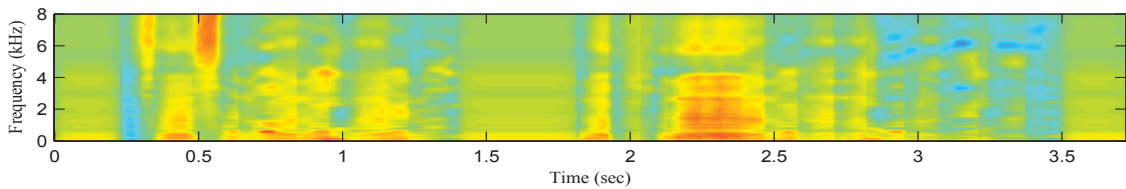
(c) *Comp3*



(d) Component 1 of *Comp3*



(e) Component 2 of *Comp3*



(f) Component 3 of *Comp3*

Figure 3.8: Spectrograms of test speech and synthesized speech in *Conv* and *Comp3* (450 sentences). Spectrograms corresponding to each component of *Comp3* are also shown.

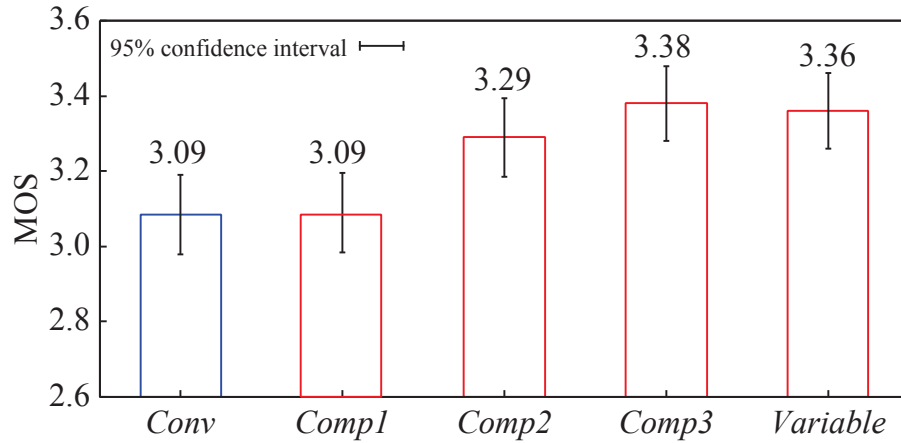


Figure 3.9: Mean opinion scores for synthesized speech with 95% confidence intervals obtained by conventional and proposed methods (200 sentences).

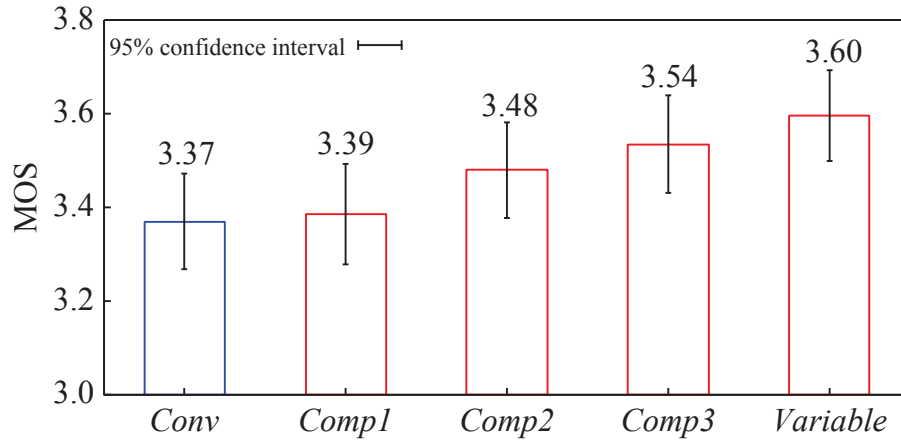


Figure 3.10: Mean opinion scores for synthesized speech with 95% confidence intervals obtained by conventional and proposed methods (450 sentences).

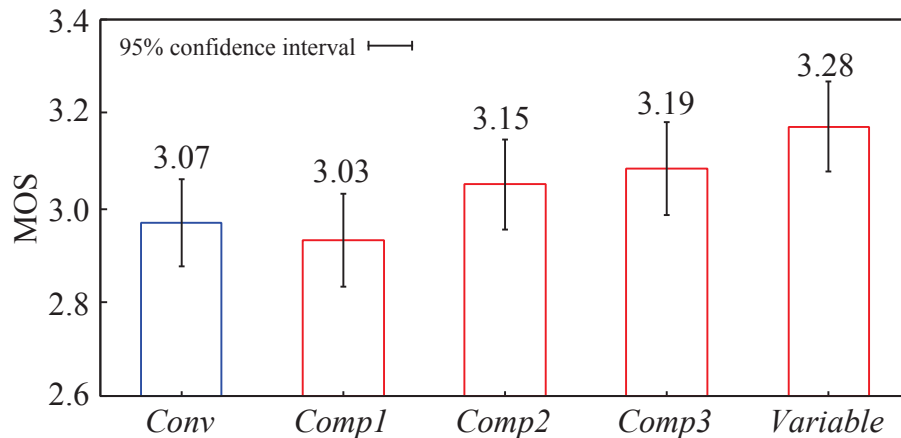


Figure 3.11: Mean opinion scores for synthesized speech with 95% confidence intervals obtained by conventional and proposed methods (1,267 sentences).

Chapter 4

An optimization algorithm for mean and variance tying structures

In this chapter, a technique for constructing independent parameter tying structures of mean and variance using additive structure models for HMM-based speech synthesis is described. This model structure is one of the structures of additive structure models and equivalent to an constrained ones. However, this model structure is very interesting and experimenta show useful results for modeling of acoustic features. Conventionally, mean and variance parameters are assumed to have the same tying structure. However, it has been reported that a clustering technique of mean vectors while tying all variance matrices improves the quality of synthesized speech. This indicates that mean and variance parameters should have different optimal tying structures. In the proposed technique, the decision trees for mean and variance parameters are simultaneously grown by taking into account the dependency on mean and variance parameters.

4.1 Independent Tying Structures for Mean and Variance Parameters

In this section, a context clustering technique for both mean and variance parameters is described. Conventionally, an HMM stream-level tying structure is constructed in HMM-based speech synthesis, i.e., mean vectors and variance matrices have exactly the same parameter tying structure. In this paper, it is assumed that both mean and variance parameters have their own tying structure and examine the construction of appropriate parameter tying structures. Figure 4.1 shows an example of parameter tying structures constructed with the conventional and proposed techniques. In the clustering algorithm, it is neces-

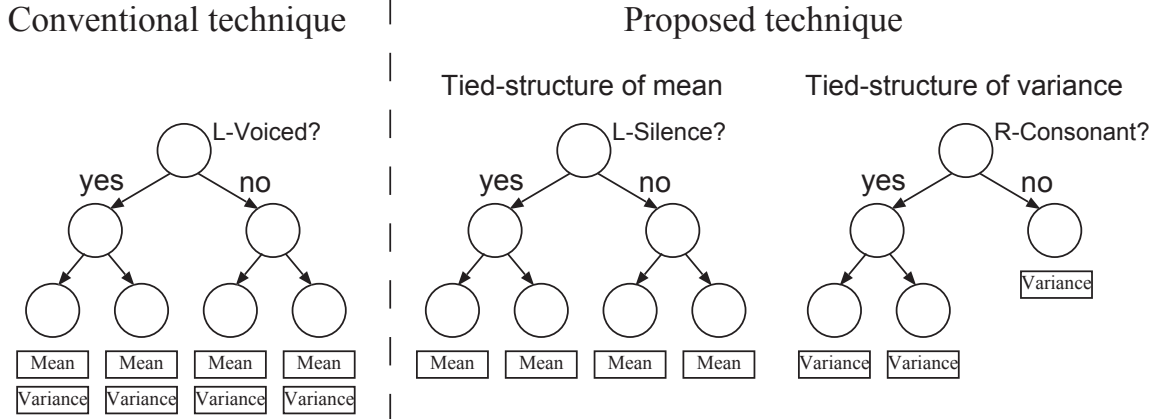


Figure 4.1: Example of parameter tying structures constructed with the conventional and proposed techniques.

sary to simultaneously construct each parameter tying structure due to the dependency on mean and variance parameters. Although such a context clustering algorithm can be derived by expanding the conventional context clustering algorithm, the algorithm is derived using the fact that simultaneous context clustering of mean and variance parameters can be regarded as a special case of context clustering in additive structure models.

4.1.1 Proposed Model Structure

In additive structure models, an acoustic feature vector is generated by the sum of additive components.

In this paper, an acoustic feature vector \mathbf{o}_t is generated by the sum of two components, i.e., $\mathbf{o}_t^{(m)}$ and $\mathbf{o}_t^{(v)}$:

$$\mathbf{o}_t = \mathbf{o}_t^{(m)} + \mathbf{o}_t^{(v)}. \quad (4.1)$$

If each component is independent and generated according to a Gaussian distribution, each component usually has mean and variance parameters. In this paper, it is assumed that $\mathbf{o}_t^{(m)}$ is generated from a Gaussian distribution that has only a mean parameter and zero variance and $\mathbf{o}_t^{(v)}$ is generated from one that has only a variance parameter and zero mean. In this case, the probabilistic density function of the acoustic feature is represented by the convolution of these two components so that

$$\mathbf{o}_t^{(m)} \sim \mathcal{N}(\boldsymbol{\mu}_{c_t}, \mathbf{0}), \quad (4.2)$$

$$\mathbf{o}_t^{(v)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{c_t}), \quad (4.3)$$

$$P(\mathbf{o}_t | c_t, \lambda) = \mathcal{N}(\mathbf{o}_t | \boldsymbol{\mu}_{c_t}, \boldsymbol{\Sigma}_{c_t}). \quad (4.4)$$

Assuming that each component has a different decision tree, independent parameter tying structures of mean and variance can be represented.

4.1.2 Parameter Estimation for the proposed technique

In this model structure, the Maximum Likelihood (ML) parameters can be estimated with the Expectation Maximization (EM) algorithm. In the E-step, since the convolved output probability distribution becomes a Gaussian distribution, the standard forward-backward algorithm and the Viterbi algorithm can simply be applied as in standard HMMs.

Using the statistics obtained by the E-step, the Q -function with respect to the output probability distribution can be written as

$$\begin{aligned} Q &= \sum_{t=1}^T \sum_{c \in C} \gamma_t(c) \log P(\mathbf{o}_t | c_t = c, \lambda) \\ &= -\frac{1}{2} \sum_{c \in C} \tilde{T}_c \left[K \log 2\pi + \log |\tilde{\Sigma}_c| \right. \\ &\quad \left. + \text{Tr} \left\{ \tilde{\Sigma}_c^{-1} \left(\tilde{\Sigma}_c + (\boldsymbol{\mu}_c - \tilde{\boldsymbol{\mu}}_c)(\boldsymbol{\mu}_c - \tilde{\boldsymbol{\mu}}_c)^\top \right) \right\} \right], \end{aligned} \quad (4.5)$$

where K is the dimensionality of feature vectors and C denotes all contexts observed in the training data. The statistics with respect to context c are represented by $(\tilde{\cdot})_c$ and each of the statistics is calculated as follows:

$$\tilde{T}_c = \sum_{t=1}^T \gamma_t(c), \quad \tilde{\boldsymbol{\mu}}_c = \frac{1}{\tilde{T}_c} \sum_{t=1}^T \gamma_t(c) \mathbf{o}_t, \quad (4.6)$$

$$\tilde{\Sigma}_c = \frac{1}{\tilde{T}_c} \sum_{t=1}^T \gamma_t(c) (\mathbf{o}_t - \tilde{\boldsymbol{\mu}}_c)(\mathbf{o}_t - \tilde{\boldsymbol{\mu}}_c)^\top, \quad (4.7)$$

where $\gamma_t(c)$ is the state occupancy probability and the state index is ignored for simplicity of notation.

By setting the first partial derivative of Q function with respect to an arbitrary mean vector or variance matrix, the ML parameters are given as follows:

$$\begin{aligned} \boldsymbol{\mu}_{n(m)} &= \left(\sum_{c \in \phi_{n(m)}} \tilde{T}_c \tilde{\Sigma}_c^{-1} \right)^{-1} \sum_{c \in \phi_{n(m)}} \tilde{T}_c \tilde{\Sigma}_c^{-1} \tilde{\boldsymbol{\mu}}_c, \\ \Sigma_{n(v)} &= \left(\sum_{c \in \phi_{n(v)}} \tilde{T}_c \right)^{-1} \cdot \sum_{c \in \phi_{n(v)}} \tilde{T}_c \left\{ \tilde{\Sigma}_c + (\boldsymbol{\mu}_c - \tilde{\boldsymbol{\mu}}_c)(\boldsymbol{\mu}_c - \tilde{\boldsymbol{\mu}}_c)^\top \right\}, \end{aligned} \quad (4.8)$$

where $n^{(m)}, n^{(v)}$ are respectively the number of clusters in mean and variance parameter trees, and $\phi_{n^{(\cdot)}}$ denotes the contexts included in the $n^{(\cdot)}$ -th cluster.

It can be seen from the Eqs. (4.8) and (4.9) that the update of $\mu_{n^{(m)}}$ and $\Sigma_{n^{(v)}}$ requires the parameters of the other clusters. Hence, all parameters of all trees have dependencies on each other to compose the output probabilities; therefore, all parameters of all trees should be estimated simultaneously. Thus, iterative updates are needed for estimating mean and variance parameters until a convergence.

4.1.3 Simultaneous Context Clustering for Mean and Variance Parameters

In the context clustering, the optimal parameter tying structures are given by maximizing Eq. (4.5). However, it is necessary to simultaneously construct each parameter tying structure due to the dependency on mean and variance parameters. Since this problem corresponds to a problem of estimating parameter tying structures of additive components $\mathbf{o}_t^{(m)}$ and $\mathbf{o}_t^{(v)}$, appropriate parameter tying structures of mean and variance parameters are constructed with simultaneous context clustering in additive structure models. The procedure for the proposed context clustering algorithm is as follows.

- Step 1.** The root nodes of the two trees of mean and variance parameters are created.
- Step 2.** Questions at all leaf nodes of two trees are evaluated. The likelihood after the node is split is calculated by estimating the ML parameters of all leaf nodes of all trees.
- Step 3.** The pair of a node and question that gives the maximum likelihood is selected, and the node is split into two by applying the question. The model parameters of all leaf nodes are updated by the ML parameters.
- Step 4.** If the change of likelihood after the node is split is below a predefined threshold, stop the procedure. Otherwise, go to Step 2.

The decision trees of mean and variance parameters can be simultaneously constructed with this technique. Furthermore, the size of mean and variance decision trees can be independently controlled with the the proposed technique by adjusting the weights in the MDL criterion. Thus, the proposed context clustering would construct more appropriate parameter tying structures than the conventional one.

4.2 Experiments

4.2.1 Experimental conditions

The first 450 sentences of the phonetically balanced 503 sentences the ATR Japanese speech database B-set, uttered by male speaker MHT, were used for training. The remaining 53 sentences were used for evaluation. The speech data was down-sampled from 20 to 16 kHz and windowed at a frame rate of 5-ms using a 25-ms Blackman window.

The feature vectors consisted of spectral and F_0 feature vectors. The spectrum parameter vectors consisted of 39 STRAIGHT mel-cepstral coefficients [34] including the zero coefficient, their delta and delta-delta coefficients. The excitation parameter vectors consisted of $\log F_0$, its delta and delta-delta.

A five-state, left-to-right, no-skip structure with diagonal covariance matrices was used for the hidden semi-Markov model. The proposed context clustering technique for mean and variance parameters is applied to only the spectrum parameters. The conventional and proposed techniques have the same tying structures for the excitation parameters. The MDL criterion was used to control the size of the tree of the conventional technique and the mean parameter tree of the proposed technique. The heuristic weight for the penalty term (Eq. (18) in [8]) is changed to construct the variance parameter tree of the proposed technique. The weights used here were 4.0, 2.0, and 1.0. In addition, the proposed technique is compared with a technique for tying variance parameters in each state of HMMs as conventional one. In [11], variance parameters are tied to one in all states of HMMs.

4.2.2 Experimental results

Table 4.1 lists the number of leaf nodes and the total number of parameters for each technique. In this table, *Baseline* is the conventional technique, *TieVar* is the technique for tying variance parameters in each state of HMMs, and *MDL4.0*, *MDL2.0*, and *MDL1.0* respectively represent the proposed technique with 4.0, 2.0, and 1.0 weights of the MDL criterion. Although leaf nodes have mean and variance parameters in *Baseline*, in the other techniques leaf nodes have only parameters of either. First, it can be seen from the table that *MDL1.0* has more mean parameters and less variance parameters than *Baseline*. This indicates that the proposed technique constructs decision trees that are appropriately sized for both mean and variance parameters. Next, *MDL2.0* and *MDL4.0* have less variance parameters and slightly more mean parameters in the proposed technique. This means that the mean parameter decision tree was constructed to compensate for less

Table 4.1: Number of leaf nodes and total number of parameters.

	Number of leaf nodes		The total number of parameters
	Mean	Variance	
<i>Baseline</i>	809	809	194160
<i>TieVar</i>	1316	5	158520
<i>MDL4.0</i>	1255	147	168240
<i>MDL2.0</i>	1249	247	179520
<i>MDL1.0</i>	1235	403	196560

variance parameters.

A subjective listening test was conducted to evaluate quality of synthesized speech. The subjects were asked to rate the naturalness of the synthesized speech on a scale from one (completely unnatural) to five (natural). The subjects were 10 Japanese. Twenty sentences were randomly chosen from the evaluation sentences. Figure 4.2 plots the experimental results. In this figure, although *TieVar* and *MDL4.0* obtained almost the same score, the proposed technique with the small weight of MDL criterion achieved better subjective scores than the conventional one. This indicates that the proposed technique constructed the optimal tying structures for each of mean and variance parameters. It can be seen from the table 4.1 that although the total number of parameters is almost the same in *Baseline* and *MDL1.0*, their balance between the number of mean and variance parameters are different. Even though this indicates that mean parameters are relatively more important than variance parameters, some degree of freedom for variance parameters is necessary for improving the quality of synthesized speech.

4.3 Summary

In this chapter, an optimization algorithm of independent mean and variance parameter tying structures for HMM-based speech synthesis was proposed. The proposed technique constructed simultaneously tying structures for both mean and variance parameters using context clustering algorithm in additive structure models. In the experiments, the proposed technique outperformed the conventional one. Investigation of the appropriate size of the trees will be future work.

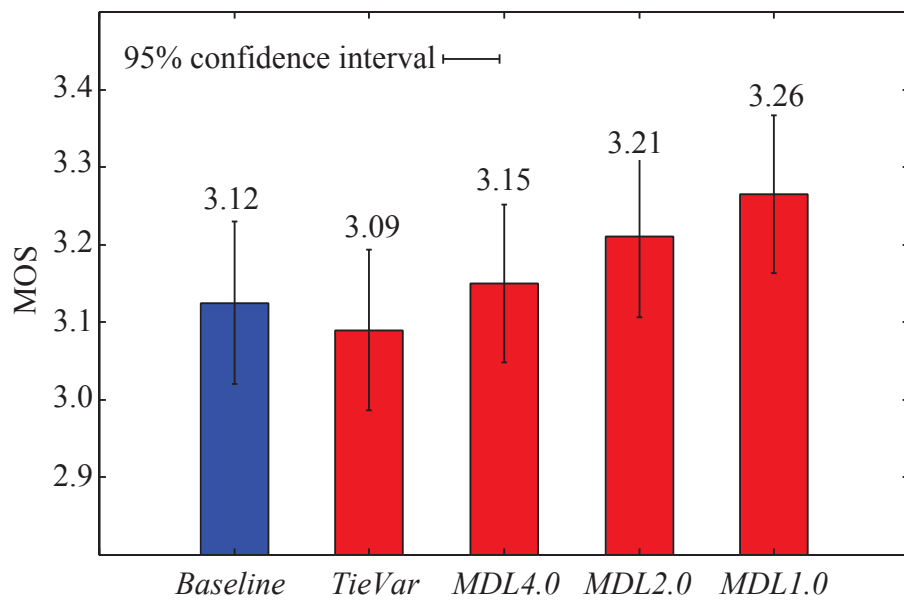


Figure 4.2: Mean opinion scores for synthesized speech obtained by the conventional and proposed techniques.

Chapter 5

Acoustic modeling with contextual partial additive structures

The contextual additive structure models assume that the observation vectors are generated from the sum of additive components with tree regression structures and they can be regarded as an intermediate structure between linear regression and tree regression. However, the additive structure models still have a limitation that the number of additive components is fixed for all output probability distributions. The proposed technique is a generalization of the additive structure models which have variable number of additive components dependently on contextual sub-spaces, and the clustering algorithm for extracting partial additive structure is provided.

5.1 Contextual partial additive structure

Although additive structure models can automatically determine the number of components, there is a constraint that a fixed number of additive components are used for generating acoustic features. However, it is natural to assume that an appropriate number of additive components depends on contexts. That is, it is expected that some context dependent models require many additive components to represent variations in acoustic features and others not. To represent such context dependencies, partial additive components affecting arbitrary contextual sub-spaces is introduced.

In the proposed technique, a partial additive component is represented by a decision tree attached to an internal node of another decision tree. Figure 5.1 shows examples of the standard and partial additive structure. The standard technique extracts additive components for the only entire contextual space corresponding to a root node. The proposed

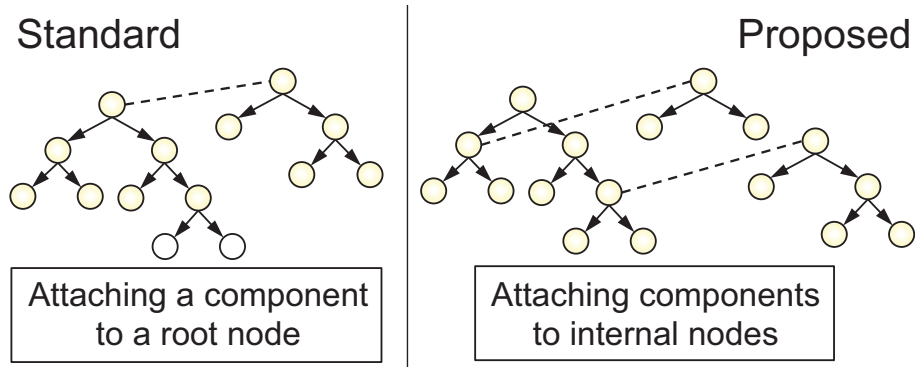


Figure 5.1: Examples of standard and partial additive structures.

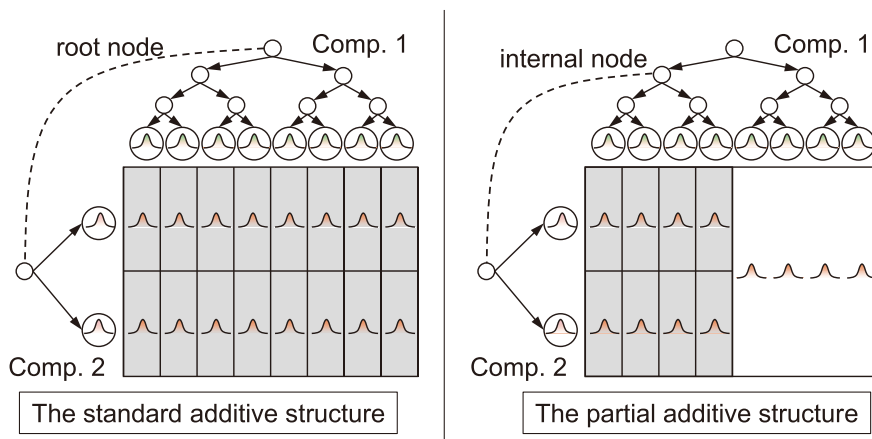


Figure 5.2: The effect of an partial additive structure in distribution modeling of acoustic features.

technique can attach the additive component to an arbitrary node including internal nodes as well as root nodes. Figure 5.2 shows the effect of a partial additive structure in distribution modeling of acoustic features. The gray regions represent the contextual spaces affected by the second additive component “Comp. 2”. The second component of the partial additive structure affects the contextual sub-space corresponding to the internal node of the first component, even though the second component of the standard structure always divides the entire contextual space. The proposed model structure yields larger combination of components than the standard additive structure with the same number of parameters.

Considering the relation between the standard and partial additive structure models, an arbitrary partial additive structure can be converted to a global additive component, because a partial decision tree can be expanded to a global decision tree by copying the upper

structure of the parent decision tree. In this case, the partial decision tree is represented as a sub-tree at the internal node of the copied tree and the other nodes are assumed to have zero mean and variance. Therefore, the proposed structure can be regarded as special case of the standard additive structure. This means there is no advantage of the proposed technique in the representation of decision trees. However, the proposed technique provides an efficient representation for partial context dependencies with a smaller number of model parameters. Furthermore, if there exists an optimal structure representing partial context dependencies, it is difficult to extract an equivalent global additive structure by using the context clustering algorithm described in Section 2.2, due to the greedy strategy. Therefore, an explicit representation of partial context dependencies and a context clustering algorithm for extracting partial additive structures are required.

The context clustering algorithm for the partial additive structure can be derived by modifying STEP. 2 in the standard context clustering algorithm for multiple decision trees as follows:

STEP 2. Evaluate questions at all leaf nodes of all trees and a root node representing a new tree. In addition, all candidate root nodes representing partial additive components are also evaluated at all internal nodes. The likelihood after the node splitting is calculated by estimating the ML parameters of all leaf nodes of all trees.

The difference with the standard context clustering algorithm for multiple decision trees is to explicitly evaluate all questions at all internal nodes for constructing a new tree representing a partial additive component. The number and position of additive components corresponding to each context dependent model are automatically determined on demand to increase the likelihood based on the ML criterion. Thus, the proposed technique can effectively represent the context dependencies with a limited amount of the training data. For an unseen context, the corresponding distribution can be found by answering the question from the top-node as the standard decision tree. However, if there is an attached decision tree at the current node, the number of components for the current context is increased and the corresponding distributions must be searched for in both the parent and attached decision trees.

5.1.1 Related model structures

The additive structure models include different model structures as special cases. If the additive structure is restricted to having a single decision tree, it becomes the conventional decision tree (tree regression). Linear regression models [9] can also be represented by additive structure models, which consist of additive components each of which has only

one contextual question. Therefore, additive structure models can be regarded as intermediate models between tree regression and linear regression. Partial decision trees in the proposed technique inherit this property. Constrained Tree Regression (CTR) [36] also has a strong relation to the proposed model structure. CTR has an additive component corresponding to a contextual question at each intermediate node, and feature vectors are predicted by adding all additive components from the top-node to leaf-node. Although CTR can also represent a variable number of additive components, similar to the proposed structure, only a sub-set of standard additive structure models can be represented by CTR because it integrates the structures of tree regression and linear regression into a single tree structure. As mentioned above, partial additive structure models have the same ability in the representing model structures as standard additive structure models.

5.2 Experiment

5.2.1 Experimental conditions

Objective and subjective experiments were conducted to evaluate the effectiveness of the proposed method. The 200 and 450 sentences of the phonetically balanced 503 sentences from the ATR Japanese speech database B-set, uttered by male speaker MHT, were used for training. The remaining 53 sentences were used for evaluation. The speech data was down-sampled from 20 to 16 kHz and windowed at a frame rate of 5-ms using a 25-ms Blackman window.

The feature vectors consisted of spectral and F_0 feature vectors. The spectrum parameter vectors consisted of 39 STRAIGHT mel-cepstral coefficients including the zero coefficient and their delta and delta-delta coefficients. The excitation parameter vectors consisted of $\log F_0$ and its delta and delta-delta. A five-state, left-to-right, no-skip structure with a diagonal covariance matrix was used for the hidden semi-Markov model. Additive structure modeling was applied to only the spectrum parameters, and the excitation parameters were modeled with conventional multi-space probability distribution HMMs [37]. The tying structures for excitation parameters were constructed with the conventional decision tree based context clustering.

Four techniques were compared; *CONV*: the conventional decision tree, *LR*: the linear regression, *ADD*: the standard additive structure models, and *PADD*: the proposed partial additive structure models. Covariance parameter tying was applied to *LR*, *ADD* and *PADD* for reducing computational cost.

The minimum description length (MDL) criterion [8] was used to select splitting a node in

Table 5.1: Number of decision trees in each state. The number of decision trees In *PADD* consists of that attached to the root node and internal nodes (200 sentences).

	<i>CONV</i>	<i>LR</i>	<i>ADD</i>	<i>PADD</i>
State 1	1	59	6	7 (root 3 + internal 4)
State 2	1	73	4	7 (root 2 + internal 5)
State 3	1	81	3	9 (root 2 + internal 7)
State 4	1	59	3	7 (root 2 + internal 5)
State 5	1	63	6	8 (root 3 + internal 5)
Total	5	335	22	38 (root 12 + internal 26)

Table 5.2: Number of leaf clusters, total number of parameters and average likelihood per frame of training and test data (200 sentences).

	<i>CONV</i>	<i>LR</i>	<i>ADD</i>	<i>PADD</i>
# of leaf nodes	440	670	771	844
Total # of parameters	105,600	81,000	93,120	101,880
Ave. likelihood (train)	139.66	130.92	132.60	132.98
Ave. likelihood (test)	131.62	124.87	124.97	125.18

all techniques. In the proposed technique, the increase in the the number of parameters of splitting a leaf node and extracting a new component differs. The increase in the number of parameters by extracting a new component doubles compared with that by splitting a leaf node. Penalty terms of the description length then grows large in extracting a new component. The MDL criterion was used to determine the size of the decision trees.

In subjective experiments, mean opinion score tests were conducted. Ten subjects participated in these listening tests. Twenty sentences were randomly selected from the 53 sentences for each subject. The subjects were asked to rate the naturalness of the synthesized speech on a scale from one (completely unnatural) to five (natural). The experiment was carried out using headphones in a soundproof room .

5.2.2 Experimental results

5.2.3 Objective results

Table 5.1 and 5.3 lists the number of decision trees in each HMM state and total number of decision trees in each technique. The number of decision trees in *PADD* consists of

Table 5.3: Number of decision trees in each state. The number of decision trees in *PADD* consists of that attached to the root node and internal nodes (450 sentences).

	<i>CONV</i>	<i>LR</i>	<i>ADD</i>	<i>PADD</i>
State 1	1	80	5	10 (root 4 + internal 6)
State 2	1	140	6	18 (root 2 + internal 16)
State 3	1	103	7	14 (root 3 + internal 11)
State 4	1	83	5	13 (root 3 + internal 10)
State 5	1	80	6	9 (root 3 + internal 6)
Total	5	486	30	59 (root 15 + internal 44)

Table 5.4: Number of leaf clusters, total number of parameters and average likelihood per frame of training and test data (450 sentences).

	<i>CONV</i>	<i>LR</i>	<i>ADD</i>	<i>PADD</i>
# of leaf nodes	814	972	1391	1446
Total # of parameters	195,360	117,240	167,520	174,120
Ave. likelihood (train)	138.65	129.30	132.15	132.36
Ave. likelihood (test)	136.10	127.83	130.07	130.27

that attached to the root node and internal nodes corresponding to the global and partial additive components respectively. It can be seen from Table 5.1 and 5.3 that the additive structure models constructed multiple trees for each state in the context clustering, even though they can select single tree structures. This results suggest that there is an additive structure in the training data. The additive structure models also constructed less decision trees compared to *LR*. In the additive structure models, intermediate structures between tree regression and linear regression were constructed to represent appropriate context dependencies. Furthermore, *PADD* created decision trees at internal nodes as well as the root node. This means that the proposed clustering algorithm extracted partial additive components to efficiently represent context dependencies in the training data. In 200 sentence case, a larger number of decision trees were obtained for State 1 and State 5 than the middle state of HMMs in *ADD*. This might be because the triphone or quinphone contexts strongly affect the spectral features around phone boundaries. However, almost the same number of components were extracted in the all states in *PADD*. This is because *PADD* extracted an appropriate number of components depending on contexts, while *ADD* extracted only components for entire contextual space. With increasing the amount of training data, a larger number of partial additive components as well as the global components were extracted for representing the spectral variations caused by var-

ious contextual factors. These means that *PADD* can effectively represent the context dependencies with a limited amount of the training data.

Figures 5.3 and 5.4 are histograms of the number of components for each context dependent model about each state in *PADD*. It can be seen from figure 5.3 and 5.4 that the different numbers of components were used for each context model. For examples, the larger number of components were used for representing acoustic features affected by various contextual factors, i.e., vowels and some context features in the larger amount of training data. Acoustic features that have less spectral variations, i.e., features about silence, pause, and some context in the smaller amount of training data, were represented by the smaller number of components. This means that *PADD* extracted the different numbers of components depending on contexts. However, there were unused and less used numbers of components in especially a larger amount of training data case. This might be because the greedy strategy of context clustering algorithm still affected the resultant structure, though proposed context clustering algorithm effectively extracted the partial additive structure.

Table 5.2 and 5.4 list the number of leaf nodes, the total number of parameters and the average likelihoods per frame of training (200 and 450 sentences) and test data (53 sentences). Note that *CONV* has double number of parameters in each leaf node compared with *LR*, *ADD* and *PADD*, because the covariance parameter tying was applied to *LR*, *ADD* and *PADD*. In table 5.2 and 5.4, the likelihood of *CONV* in the training and test data was the highest of the four techniques. This is because covariance parameter tying was not applied to *CONV* and the total number of parameters was larger than other three techniques. It can also be seen that the likelihood of additive structure models is higher than *LR*. This means that additive structure models represented complicated spectral variations, while the linear regression was too simple structure to represent that. It can also be seen from Table 5.2 and 5.4 that *ADD* and *PADD* have almost the same number of parameters and there is not the significant difference in the likelihood of *ADD* and *PADD*.

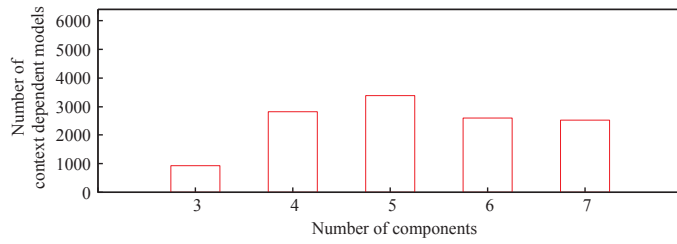
5.2.4 Subjective results

Figure 5.5 and 5.6 show the subjective listening results. In the subjective test, *LR* that synthesizes low quality speech is not included from the result of the preliminary experiment. In figure 5.5 and 5.6, *ADD* and *PADD* achieved better subjective scores than *CONV* that has larger number of parameters. This means that additive structure models could represent complicated context dependencies. It can be seen from Figure 5.5 and 5.6 that *PADD* achieved better subjective scores than *ADD*. These results mean that the proposed technique can represent appropriate context dependencies with the contextual partial addi-

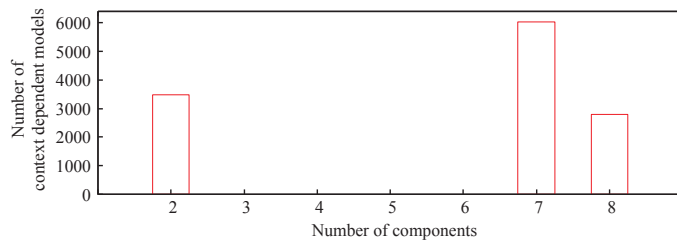
tive structure, even though *ADD* and *PADD* have almost the same number of parameters. In 450 sentences case, the difference of subjective scores between *ADD* and *PADD* was more clear than 200 sentences case. This is because a larger number of partial components were extracted and the more effective structure was constructed from the larger amount of training data. Moreover, the proposed technique could automatically determine the number of components affecting contextual sub-spaces as well as the entire contextual space and effectively represent the context dependencies with the training data.

5.3 Summary

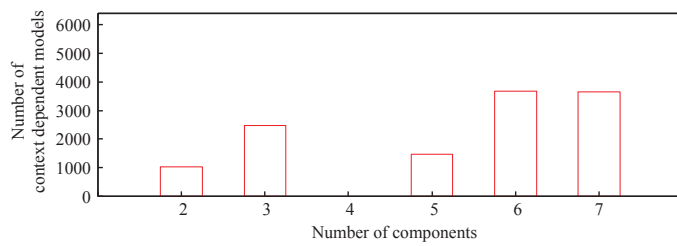
In this chapter, a spectral modeling technique based on the contextual partial additive structure was proposed. In the standard additive structure models, it is difficult to extract partial additive components which affects arbitrary contextual sub-spaces. The proposed technique can extract the contextual partial additive structure. Furthermore, the number of partial additive components as well as standard global additive components can be automatically determined with the training data. In the experiment, the proposed technique outperformed the conventional technique and the standard additive structure models. Additive structure modeling for prosodic information such as F0 and experiments on other dataset including style, emotions, etc, will be a future work.



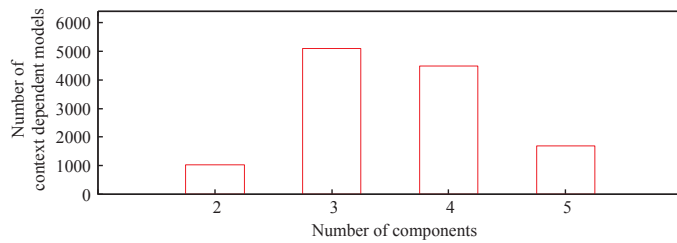
(a) State 1



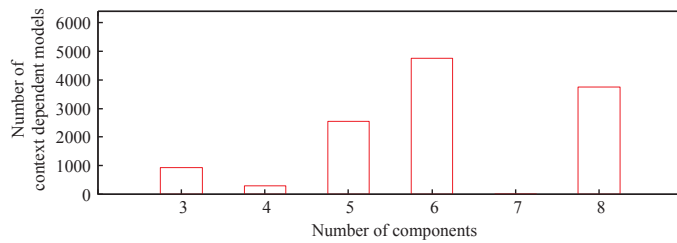
(b) State 2



(c) State 3

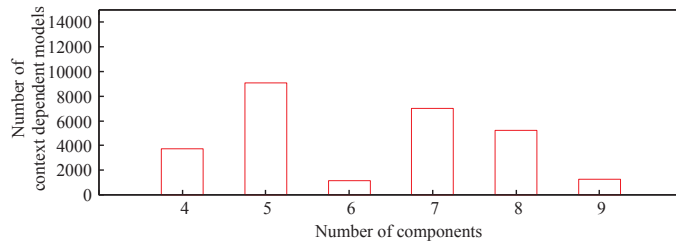


(d) State 4

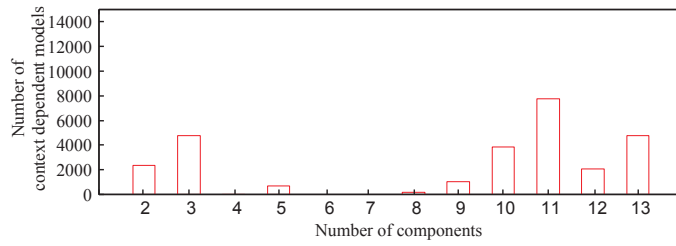


(e) State 5

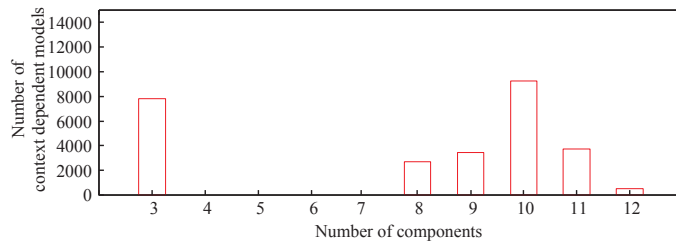
Figure 5.3: Histograms for the numbers of components for each context dependent model about each state (200 sentences).



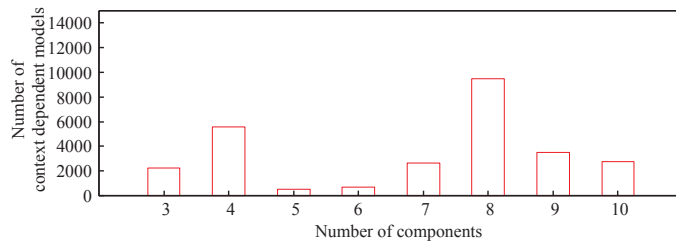
(a) State 1



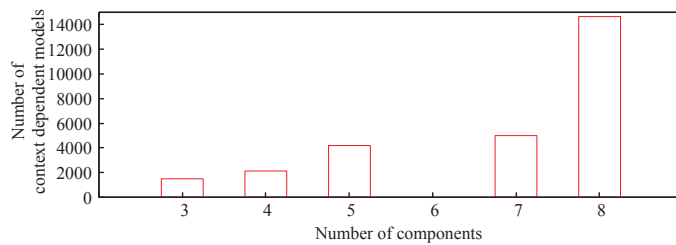
(b) State 2



(c) State 3



(d) State 4



(e) State 5

Figure 5.4: Histograms for the numbers of components for each context dependent model about each state (450 sentences).

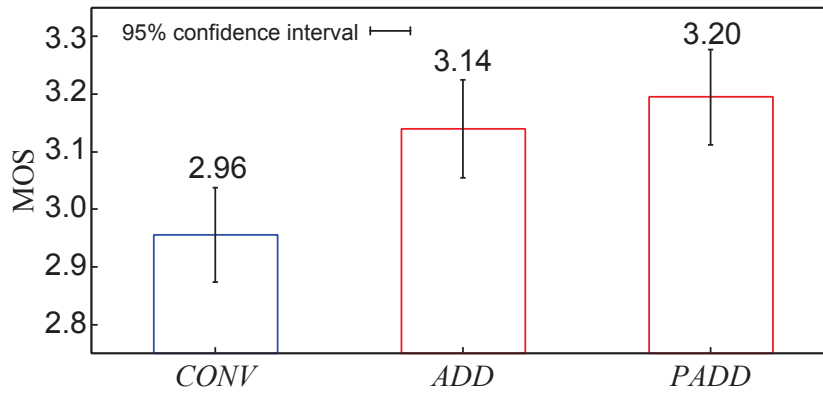


Figure 5.5: Mean opinion scores for synthesized speech obtained by conventional, standard and proposed techniques (200 sentences).

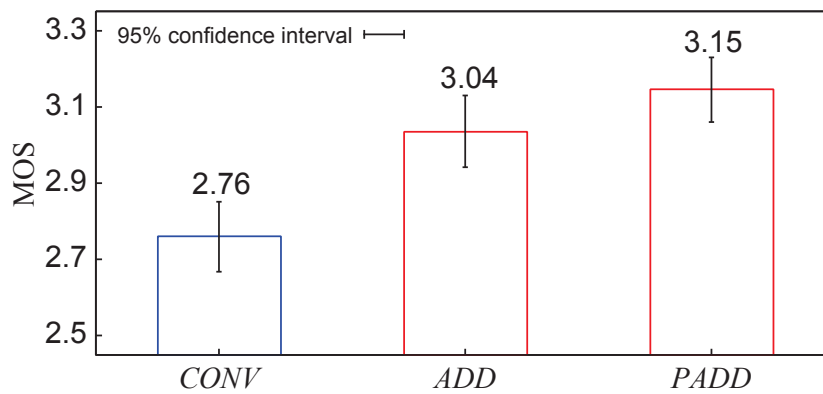


Figure 5.6: Mean opinion scores for synthesized speech obtained by conventional, standard and proposed techniques (450 sentences).

Chapter 6

Conclusions

The present paper described improved acoustic modeling for HMM-based speech synthesis. The basic theories of the hidden Markov models (HMMs), its algorithm for calculating the output probability (forward-backward algorithm), searching the optimal state sequence (Viterbi algorithm), and estimating its parameters (EM algorithm) are described in Chapter 2. In Chapter 2, statistical speech synthesis frameworks based on the HMM were also presented. In Chapter 3, an acoustic modeling with contextual additive structures for HMM-based speech synthesis was described. Contextual additive structure models can represent complicated dependencies between acoustic features and context labels using multiple decision trees. However, the computational complexity of the context clustering is too high for the full context labels of speech synthesis. Covariance parameter tying in each state and using the matrix inversion lemma can significantly reduce the amount of computational complexity and allow us to apply additive structure models to HMM-based speech synthesis. In objective results, although the size of decision trees differs among additive components, multiple decision trees were split. This suggests that additive structures are inherent in the training data. This suggests that additive structures are inherent in the training data. In subjective results, additive structure models achieved better subjective scores than the conventional methods. Additive structure modeling for prosodic information such as F0 will be a future work, because F0 has an additive structure with multiple contextual factors. In Chapter 4, a technique for constructing independent parameter tying structures of mean and variance using additive structure models for HMM-based speech synthesis was described. In the proposed technique, the decision trees for mean and variance parameters are simultaneously grown by taking into account the dependency on mean and variance parameters. The proposed technique with the small weight of MDL criterion achieved better subjective scores than the conventional one. This indicates that the proposed technique constructed the optimal tying structures for each of mean and variance parameters. Even though experimental results indicate that

mean parameters are relatively more important than variance parameters, some degree of freedom for variance parameters is necessary for improving the quality of synthesized speech. Investigation of the appropriate size of the trees will be future work. In Chapter 5, an acoustic modeling with contextual partial additive structures for HMM-based speech synthesis was described. The additive structure models still have a limitation that the number of additive components is fixed for all output probability distributions. The proposed technique is a generalization of the additive structure models which have variable number of additive components dependently on contextual sub-spaces, and provided the clustering algorithm for extracting partial additive structure. In the subjective test, partial additive structure models achieved better subjective scores. These results mean that the proposed technique can represent appropriate context dependencies with the contextual partial additive structure. Additive structure modeling for prosodic information such as F0 and experiments on other dataset including style, emotions, etc, will be a future work.

Bibliography

- [1] R. Sproat, J. Hirschberg, and D. Yarowsky. A corpus-based synthesizer. *Proceedings of International Conference on Spoken Language Processing*, pp. 563–566, 1992.
- [2] T. Masuko, K. Tokuda, T. Kobayashi, and S. Imai. Speech synthesis from HMMs using dynamic features. *Proceedings of ICASSP*, pp. 389–392, 1996.
- [3] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura. Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis. *Proceedings of Eurospeech 1999*, pp. 2347–2350, 1999.
- [4] K. Tokuda, T. Kobayashi, and S. Imai. Speech parameter generation from HMM using dynamic features. *Proceedings of International Conference on Acoustics, Speech, and Signal Processing '95*, pp. 660–663, 1995.
- [5] K. Tokuda, T. Masuko, Y. Yamada, T. Kobayashi, and S. Imai. An algorithm for speech parameter generation from continuous mixture HMMs with dynamic features. *Proceedings of European Conference on Speech Communication and Technology '95*, pp. 757–760, 1995.
- [6] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura. Speech parameter generation algorithms for HMM-based speech synthesis. *Proceedings of ICASSP 2000*, pp. 936–939, 2000.
- [7] J.J. Odell. The use of context in large vocabulary speech recognition. *PhD dissertation, Cambridge University*, 1995.
- [8] K. Shinoda and T. Watanabe. MDL-based context-dependent subword modeling for speech recognition. *J. Acoust. Soc. Jpn. (E)*, Vol. 21, No. 2, pp. 76–86, 2000.
- [9] Y. Abe and K. Nakajima. Speech recognition using dynamic transformation of phoneme templates depending of acoustic/phonetic environments. *Proceedings of ICASSP*, pp. 326–329, 1989.

- [10] Y. Nankaku, K. Nakamura, H. Zen, and T. Tokuda. Acoustic modeling with contextual additive structure for HMM-based speech recognition. *Proceedings of ICASSP*, pp. 4469–4472, 2008.
- [11] K. Oura, H. Zen, A. Lee, and K. Tokuda. A covariance-tying technique for HMM-based speech synthesis. *Proceedings of IEICE*, Vol. E93–D, No. 3, pp. 595–601, 2010.
- [12] X.D. Huang, Y. Ariki, and M.A. Jack. *Hidden Markov models for speech recognition*. Edinburgh University Press, 1990.
- [13] Rabiner L. and B.H. Juang. *Fundamentals of speech recognition*. 1993.
- [14] 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)*. 2005.
- [15] L. Rabiner, B.H. Juang, S.E. Levinson, and M.M. Sondhi. Recognition of isolated digits using hidden Markov models with continuous mixture densities. *AT&T Technical Journal*, Vol. 64, No. 6, pp. 1211–1234, 1985.
- [16] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, Vol. 13, pp. 260–269, 1967.
- [17] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum-likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B (Methodological)*, Vol. 39, No. 1, pp. 1–38, 1977.
- [18] 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.
- [19] A.W. Black and P. Taylor. CHATR: a generic speech synthesis system. *Proceedings of COLING94*, pp. 983–986, 1994.
- [20] A. Hunt and A.W. Black. Unit selection in a concatenative speech synthesis system using a large speech database. *Proceedings of International Conference on Acoustics, Speech, and Signal Processing '96*, Vol. 1, pp. 373–376, 1996.
- [21] A.W. Black and P. Taylor. *The Festival speech synthesis system: system documentation*. Technical Report HCRC/TR-83, University of Edinburgh, 1997.

- [22] M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi. Adaptation of pitch and spectrum for HMM-based speech synthesis using mllr. *Proceedings of ICASSP 2001*, pp. 805–808, 2001.
- [23] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura. Speaker interpolation in HMM-based speech synthesis system. *Proceedings of Eurospeech 1997*, pp. 2523–2526, 1997.
- [24] K. Shichiri, A. Sawabe, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura. Eigenvoices for HMM-based speech synthesis. *Proceedings of ICSLP*, pp. 1269–1272, 2002.
- [25] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura. Duration modeling for HMM-based speech synthesis. *Proceedings of International Conference on Spoken Language Processing '98*, Vol. 2, pp. 29–32, 1998.
- [26] K. Koishida, K. Tokuda, T. Masuko, and T. Kobayashi. Vector quantization of speech spectral parameters using statistics of dynamic features. *Proceedings of International Conference on Signal Processing '97*, pp. 247–252, 1997.
- [27] H. Fujisaki. In search of models in speech communication research. *Proceedings of Interspeech*, pp. 1–10, 2008.
- [28] H. Fujisaki and K. Hirose. Analysis of voice fundamental frequency contours for declarative sentences of japanese. *J. Acoust. Soc. Jpn. (E)*, Vol. 5, No. 4, 1984.
- [29] K. F. Lee. Context-dependent phonetic hidden Markov models for speaker-independent continuous speech recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 38, No. 4, pp. 599–609, 1990.
- [30] S. Young, J.J. Odell, and P. Woodland. Tree-based state tying for high accuracy acoustic modelling. *Proceedings of ARPA Workshop on Human Language Technology*, pp. 307–312, 1994.
- [31] S. Furui. A training procedure for isolated word recognition systems. *IEEE Transactions on Acoustics, Speech and Signal Processing*, pp. 129–136, 1980.
- [32] Y. Abe and K. Nakajima. Training of lexical models based on dtw-based parameter reestimation algorithm. *Proceedings of ICASSP*, pp. 623–626, 1988.
- [33] S. Matsoukas and G. Zavaliagkos. Convolutional density estimation in hidden Markov models for speech recognition. *Proceedings of ICASSP*, pp. 113–116, 1999.

- [34] H. Kawahara, I. Masuda-Katsuse, and A. Cheveigne. Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds. *Speech Communication*, Vol. 27, pp. 187–207, 1999.
- [35] K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi. Hidden Markov models based on multi-space probability distribution for pitch pattern modeling. *Proceedings of ICASSP*, pp. 229–232, 1999.
- [36] N. Iwahashi and Y. Sagisaka. Statistical modeling of speech segment duration by constrained tree regression. *Proceedings of IEICE trans*, Vol. E83–D, No. 7, pp. 1550–1559, 2000.
- [37] K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi. Multi-space probability distribution HMM. *IEICE Transactions on Information & Systems*, Vol. E85–D, No. 3, pp. 455–464, 2002.
- [38] T. Fukada, K. Tokuda, T. Kobayashi, and S. Imai. An adaptive algorithm for mel-cestral analysis of speech. *Proceedings of ICASSP*, pp. 137–140, 1992.
- [39] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of IEEE*, Vol. 77, pp. 257–285, 1989.
- [40] 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.
- [41] A. Ljolje, J. Hirschberg, and J.P.H. van Santen. Automatic speech segmentation for concatenative inventory selection. pp. 305–311. Springer-Verlag, 1997.
- [42] R.E. Donovan and P.C. Woodland. Automatic speech synthesizer parameter estimation using HMMs. *Proceedings of International Conference on Acoustics, Speech, and Signal Processing '95*, pp. 640–643, 1995.
- [43] A. Conkie. A robust unit selection system for speech synthesis. *Proceedings of Acoustic Society of America*, 1999.
- [44] S. Sakai and H. Shu. A probabilistic approach to unit selection for corpus-based speech synthesis. *Proceedings of the European Conference on Speech Communication and Technology*, pp. 81–84, 2005.
- [45] S. Imai. Cepstral analysis synthesis on the mel frequency scale. *Proceedings of International Conference on Acoustics, Speech, and Signal Processing '83*, pp. 93–96, 1983.

- [46] R.E. Donovan and E.M. Eide. The IBM trainable speech synthesis system. *Proceedings of International Conference on Spoken Language Processing'98*, Vol. 5, pp. 1703–1706, 1998.
- [47] H.J. Nock, M.J.F. Gales, and S.J. Young. A comparative study of methods for phonetic decision-tree state clustering. *Proceedings of European Conference on Speech Communication and Technology*, Vol. 1, pp. 111–114, 1997.
- [48] J. L. Gauvain and C. H. Lee. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Transactions on Speech and Audio Processing*, Vol. 2, pp. 291–298, 1994.
- [49] R. Kuhn, J.C. Janqua, P. Nguyen, and N. Niedzielski. Rapid speaker adaptation in eigenvoice space. *IEEE Transactions on Speech & Audio Processing*, Vol. 8, No. 6, pp. 695–707, 2000.
- [50] M.J.F. Gales. Semi-tied covariance matrices for hidden Markov models. *IEEE Transactions on Speech & Audio Processing*, Vol. 7, No. 3, pp. 272–281, 1999.
- [51] V. Vanhoucke. *Mixtures of inverse covariances: covariance modeling for Gaussian mixtures with applications to automatic speech recognition*. PhD thesis, Stanford University, 2003.
- [52] E. Moulines and F. Charpentier. Pitch synchronous waveform processing techniques for text-to-speech synthesis using diphones. *Speech Communication*, Vol. 9, pp. 453–467, 1990.
- [53] D. Talkin. A robust algorithm for pitch tracking (RAPT). pp. 497–518. Elsevier, 1995.
- [54] T. Toda and K. Tokuda. Speech parameter generation algorithm considering global variance for HMM-based speech synthesis. *Proceedings of Interspeech 2005*, pp. 2801–2804, 2005.
- [55] A. W. Black and A. J. Hunt. Generating f0 contours from ToBI labels using linear regression. *Proceedings of ICSLP*, Vol. 3, pp. 1385–1388, 1996.
- [56] Y. Qian, H. Liang, and F. K. Soong. Generating natural f0 trajectory with additive trees. *Proceedings of Interspeech 2008*, pp. 2126–2129, 2008.
- [57] H. Zen and N. Braunschweiler. Context-dependent additive log f0 model for HMM-based speech synthesis. *Proceedings of Interspeech*, pp. 2091–2094, 2009.

- [58] Y. Wu and F. K. Soong. Modeling pitch trajectory by hierarchical hmm with minimum generation error training. *Proceedings of ICASSP*, pp. 4017–4020, 2012.
- [59] M. Gales. Cluster adaptive training of hidden markov models. *IEEE Transactions on Speech and Audio Processing*, Vol. 8, pp. 417–428, 2000.
- [60] T. Anastasakos, J. McDonough, and J Makhoul. Speaker adaptive training: a maximum likelihood approach to speaker normalization. *Proceedings of ICASSP 1997*, pp. 813–816, 1997.
- [61] S. Sakai. F0 modeling with multi-layer additive modeling based on a statistical learning technique. *Proceedings of SSW5*, pp. 151–154, 2004.
- [62] K. Hashimoto, Y. Nankaku, and K. Tokuda. Bayesian speech synthesis framework integrating training and synthesis processes. *Proceedings of SSW7*, pp. 106–111, 2010.
- [63] K. Ito, M. Yamamoto, K. Takeda, T. Takezawa, T. Matsuoka, T. Kobayashi, K. Shikano, and S. Itahashi. JNAS : Japanese speech corpus for large vocabulary continuous speech recognition research. *Journal of the Acoustical Society of Japan (E)*, Vol. 20, No. 3, pp. 199–206, 1999.
- [64] M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi. Speaker adaptation for HMM-based speech synthesis system using MLLR. *Proceedings of ESCA/COCOSDA Third International Workshop on Speech Synthesis*, pp. 273–276, 1998.
- [65] R. Tsuzuki, H. Zen, K. Tokuda, T. Kitamura, M. Bulut, and S. Narayanan. Constructing emotional speech synthesizers with limited speech database. *Proceedings of ICSLP*, Vol. 2, pp. 1185–1188, 2004.
- [66] J. Yamagishi, M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi. A context clustering technique for average voice models. *IEICE Transactions on Information & Systems*, Vol. E86-D, No. 3, pp. 534–542, 2003.
- [67] J. Yamagishi, M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi. A training method of average voice model for HMM-based speech synthesis. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E86-A, No. 8, pp. 1956–1963, 2003.
- [68] J. Yamagishi, K. Onishi, T. Masuko, and T. Kobayashi. Acoustic modeling of speaking styles and emotional expressions in HMM-based speech synthesis. *IEICE Transactions on Information & Systems*, Vol. E88-D, No. 3, pp. 502–509, 2005.

- [69] J. Yamagishi and T. Kobayashi. Average-voice-based speech synthesis using HSMM-based speaker adaptation and adaptive training. *IEICE Transactions on Information & Systems*, Vol. E90-D, No. 2, pp. 533–543, 2007.
- [70] H. Zen, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura. Hidden semi-Markov model based speech synthesis. *Proceedings of ICSLP*, pp. 1185–1180, 2004.
- [71] H. Zen, N. Braunschweiler, S. Buchholz, M. Gales, K. Knill, S. Krstulovic, and J. Latorre. Statistical parametric speech synthesis based on speaker and language factorization. *IEEE Transactions on Speech and Audio Processing*, Vol. 20, pp. 1713–1724, 2012.

List of Publications

Journal papers

- [1] **Shinji Takaki**, Yoshihiko Nankaku, and Keiichi Tokuda, “Contextual additive structure for HMM-based speech synthesis,” *the IEEE Journal of Selected Topics in Signal Processing*, (Accept).
- [2] **Shinji Takaki**, Yoshihiko Nankaku, and Keiichi Tokuda, “Spectral modeling with contextual partial additive structures for HMM-based speech synthesis,” *IEICE Transactions*, (Conditional acceptance).

International conference proceedings

- [3] **Shinji Takaki**, Yoshihiko Nankaku, and Keiichi Tokuda, “Spectral modeling with contextual additive structure for HMM-based speech synthesis,” Proc. of 7th ISCA Speech Synthesis Workshop pp. 100–105, 2010.9.
- [4] **Shinji Takaki**, Keiichiro Oura, Yoshihiko Nankaku, and Keiichi Tokuda, “An Optimization Algorithm of Independent Mean and Variance Parameter Tying Structures for HMM-based speech synthesis,” Proc. of ICASSP2011 pp. 4100–4103 2011.5.
- [5] Kei Hashimoto, **Shinji Takaki**, Keiichiro Oura, and Keiichi Tokuda, “Overview of NIT HMM-based speech synthesis system for Blizzard Challenge 2011,” in Proc. Blizzard Challenge 2011, 2011.9.
- [6] **Shinji Takaki**, Kei Sawada, Kei Hashimoto, Keiichiro Oura, and Keiichi Tokuda,

“Overview of NIT HMM-based speech synthesis system for Blizzard Challenge 2012,” in Proc. Blizzard Challenge 2012, 2012.9.

- [7] Takaya Makino, **Shinji Takaki**, Kei Hashimoto, Yoshihiko Nankaku, and Keiichi Tokuda, “Separable Lattice 2-D HMMs Introducing State Duration of Images with Various Variations,” Proc. of ICASSP2013 pp. 3203–3207 2013.5.
- [8] **Shinji Takaki**, Yoshihiko Nankaku, and Keiichi Tokuda, “Contextual Partial Additive Structure for HMM-based Speech Synthesis,” Proc. of ICASSP2013 pp. 7878–7882 2013.5.
- [9] **Shinji Takaki**, Kei Sawada, Kei Hashimoto, Keiichiro Oura, and Keiichi Tokuda, “Overview of NITECH HMM-based speech synthesis system for Blizzard Challenge 2013,” in Proc. Blizzard Challenge 2013, 2013.9.

Technical reports

- [9] Takaya Makino, **Shinji Takaki**, Kei Hashimoto, Yoshihiko Nankaku, and Keiichi Tokuda, “Extended separable lattice HMMs with state duration control for recognition of images with variations,” Technical Report of IEICE, vol.112, no.441, PRMU2012-164, pp. 149–154, 2013.2.

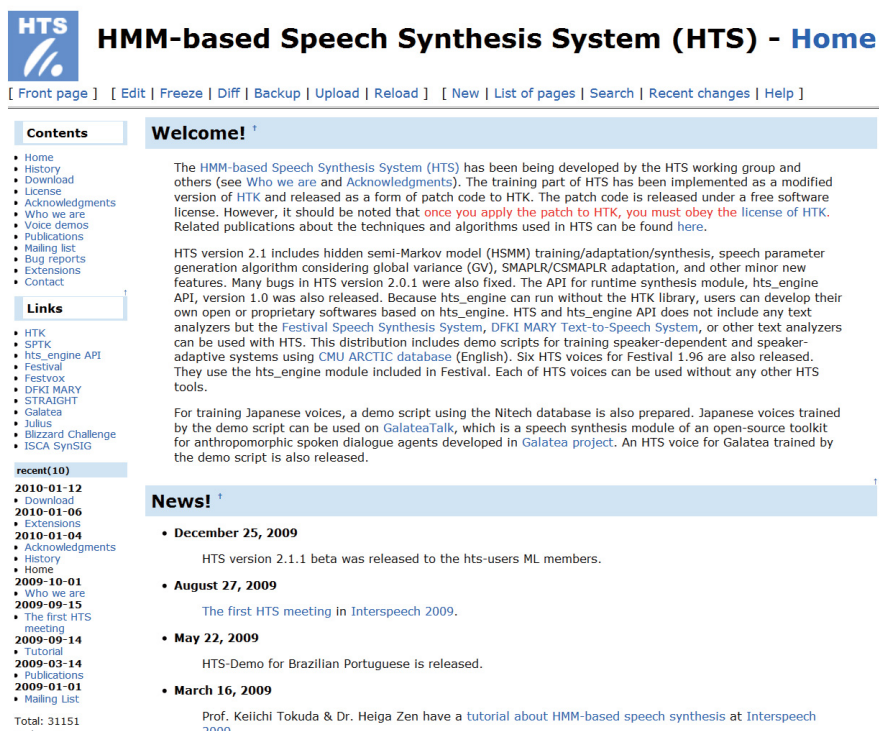
Domestic conference proceedings

- [10] **Shinji Takaki**, Yoshihiko Nankaku, and Keiichi Tokuda, “Spectral modeling with contextual additive structure for HMM-based synthesis,” Proceedings of Spring Meeting of the ASJ, 2-7-3, pp. 335–338, 2010.3.
- [11] **Shinji Takaki**, Keiichiro Oura, Yoshihiko Nankaku, and Keiichi Tokuda, “An optimization algorithm of independent mean and variance parameter tying structures for HMM-based speech synthesis,” Proceedings of Autumn Meeting of the ASJ, 2-1-8, pp. 241–242, 2010.9.

- [12] Toshihiko Sawada **Shinji Takaki**, Yoshihiko Nankaku, and Keiichi Tokuda, “An optimization algorithm of independent mean and variance parameter tying structures for HMM-based speech recognition,” Proceedings of Spring Meeting of the ASJ, 1-5-9, pp. 25–26, 2011.3.
- [13] **Shinji Takaki**, Yoshihiko Nankaku, and Keiichi Tokuda, “HMM-based speech synthesis with contextual partial additive structure,” Proceedings of Spring Meeting of the ASJ, 1-11-5, pp. 303–304, 2012.3.
- [14] **Shinji Takaki**, Yoshihiko Nankaku, and Keiichi Tokuda, “Multiple speaker modeling based on the additive structure for HMM-based speech synthesis,” Proceedings of Autumn Meeting of the ASJ, 2-2-5, pp. 281–282, 2012.9.
- [15] Shuichi Kuwako **Shinji Takaki**, Kei Hashimoto, Yoshihiko Nankaku, and Keiichi Tokuda, “Utterance adaptive training in factor analyzed acoustic models for HMM-based speech synthesis,” Proceedings of Spring Meeting of the ASJ, 1-7-12, pp. 291–292, 2013.3.
- [16] Tomohiro Okada, **Shinji Takaki**, Yoshihiko Nankaku, and Keiichi Tokuda, “Evaluation of individual clustering of F_0 and voice/unvoiced weights for HMM-based speech synthesis,” Proceedings of Spring Meeting of the ASJ, 1-7-13, pp. 293–294, 2013.3.

Appendix A

Software



The screenshot shows the homepage of the HMM-based Speech Synthesis System (HTS). At the top left is the HTS logo, a blue square with a white stylized 'H' and 'S'. To its right is the title 'HMM-based Speech Synthesis System (HTS) - Home'. Below the title is a navigation bar with links: [Front page], [Edit], [Freeze], [Diff], [Backup], [Upload], [Reload], [New], [List of pages], [Search], [Recent changes], [Help].

The main content area is divided into three columns. The left column contains a 'Contents' section with a list of links: Home, History, Download, License, Acknowledgments, Who we are, Voice demos, Publications, Mailing list, Bug reports, Extensions, and Contact. Below this is a 'Links' section with links to HTK, SPTK, hts_engine API, Festival, Festvox, DFKI MARY, STRAIGHT, Galatea, Julius, Blizzard Challenge, and ISCA SynSIG. At the bottom of the left column is a 'recent(10)' section listing dates from 2010-01-12 down to 2009-01-01, with sub-links for each date.

The middle column features a 'Welcome!' section with a paragraph explaining that HTS has been developed by the HTS working group and others, and that the training part has been implemented as a modified version of HTK. It also mentions HTS version 2.1 and its features, including hidden semi-Markov model (HSMM) training/adaptation/synthesis, speech parameter generation algorithm, and global variance (GV). It notes that HTS version 2.0.1 bugs were fixed and that the API for runtime synthesis module, hts_engine API, version 1.0 was released. It states that hts_engine can run without the HTK library, allowing users to develop their own open or proprietary softwares. It also mentions that HTS and hts_engine API do not include any text analyzers but the Festival Speech Synthesis System, DFKI MARY Text-to-Speech System, or other text analyzers can be used with HTS. It notes that the distribution includes demo scripts for training speaker-dependent and speaker-adaptive systems using CMU ARCTIC database (English). Six HTS voices for Festival 1.96 are also released. They use the hts_engine module included in Festival. Each of HTS voices can be used without any other HTS tools.

Below the welcome section is a 'News!' section with a list of dates and corresponding news items: December 25, 2009 (HTS version 2.1.1 beta was released to the hts-users ML members.), August 27, 2009 (The first HTS meeting in Interspeech 2009.), May 22, 2009 (HTS-Demo for Brazilian Portuguese is released.), and March 16, 2009 (Prof. Keiichi Tokuda & Dr. Heiga Zen have a tutorial about HMM-based speech synthesis at Interspeech 2009).

Figure A.1: HTS: <http://hts.sp.nitech.ac.jp/>