STATISTICAL OPTIMIZATION OF ACOUSTIC MODELS FOR LARGE VOCABULARY SPEECH RECOGNITION

A dissertation submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

University of Missouri-Columbia

by

Rusheng Hu

Dr. Yunxin Zhao, Supervisor

December 2006

The undersigned, appointed by the Dean of the Graduate School, have examined the dissertation entitled

STATISTICAL OPTIMIZATION OF ACOUSTIC MODELS FOR LARGE VOCABULARY SPEECH RECOGNITION

Presented by Rusheng Hu

A candidate for the degree of Doctor of Philosophy

And hereby certify that in their opinion it is worthy of acceptance.

Professor Yunxin Zhao

Professor Xinhua Zhuang

Professor Dominic K.C. Ho

Professor Hongchi Shi

Professor Allanus Tsoi

ACKNOWLEDGMENTS

The author wishes to thank my supervisor, Dr. Yunxin Zhao. I could not have imagined having a better advisor and mentor for my PhD, and without her knowledge, perceptiveness and guidance, I would never have finished. Thank-you to my graduate committee members, Dr. Xinhua Zhuang, Dr. Hongchi Shi, Dr. Dominic K.C. Ho and Dr. Allanus Tsoi, for managing to read the whole writing and always giving inspirational comments. I would also like to thank all the rest of the academic and support stuff of the Department of Computer Science at the University of Missouri-Columbia, for their kindness and help during my graduate study.

Much respect to my officemates, working with them has been a great experience. In particular, many thanks to Xiaolong Li, Rong Hu, Jian Xue, Lili Che and Xiaojia Zhang, for their friendship and collegial support.

I can not end without thanking my best friend, Wan Yan, on whose endless encouragement and love I have relied throughout my study, especially during those hard times. Wherever she is, I hope to continue, in my own small way, the noble mission she inspired me. It is to her that I dedicated this work.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

This dissertation investigates statistical optimization of acoustic models in speech recognition. Two new optimization methods are proposed for phonetic decision tree (PDT) search and Hidden Markov modeling (HMM)-- the knowledge-based adaptive PDT algorithm and the HMM gradient boosting algorithm.

Investigations are conducted to applying both methods to improve word error rate of the state-of-the-art speech recognition system. However, these two methods are developed in a general machine learning background and their applications are not limited to speech recognition.

The HMM gradient boosting method is based on a function approximation scheme from the perspective of optimization in function space rather than the parameter space, based on the fact that the Gaussian mixture model in each HMM state is an additive model of homogeneous functions (Gaussians). It provides a new scheme which can jointly optimize model structure and parameters. Experiments are conducted on the World Street Journal (WSJ) task and good improvements on word error rate are observed.

The knowledge-based adaptive PDT algorithm is developed under a trend toward knowledge-based systems and aims at optimizing the mapping from contextual phones to articulatory states by maximizing implicit usage of the phonological and phonetic information, which is presumed to be contained in large data corpus. A computational efficient algorithm is developed to incorporate this prior knowledge in PDT construction. This algorithm is evaluated on the Telehealth conversational speech recognition and significant improvement on system performance is achieved.

Chapter 1

INTRODUCTION

Telemedicine or telehealth is becoming an important means of providing quality health care to rural areas and needed patients in the United States [1], among which an important application of spoken language processing is to provide voice-driven automatic captioning system to hearing impaired users. Developing such a system in telemedicine domain is challenging in several ways. First, telehealth conversations are spontaneous and contain various amounts of filled-pauses, repetitions, repairs and noises. Second, relatively sparse training data make it difficult to train a large number of parameters in both acoustic and language modeling. Third, variations in speaking style and fluency call for effective methods to describe the pronunciation patterns of different speakers [2]. This dissertation is an effort to answer these challenges from a perspective of acoustic modeling, which includes of several innovative works carried out at the Spoken Language and Information Processing Lab in the University of Missouri-Columbia.

Major contributions of this work can be divided into two parts: firstly, a new acoustic modeling architecture is developed under a unified view of hidden Markov process (HMP) to improve the mapping between two traditional structures in speech-one physical or phonetic, the other cognitive or linguistic. In this new framework, a distinct “distinctive feature parsing” module is introduced to project logical models (triphones) into a high dimensional space, as defined by phonological rules. These logically generated features are then used to interpret correlations of the phonologically defined states. This standalone module can be as simple as generating “yes” or “no” answers to the traditional question sets as used by HTK, or as complex as those layered structures in [3], and most

importantly, it can also be made adaptive to different task domains. Therefore, the traditional HMM state-tying procedure can be divided into two sequential processes: linguistic knowledge based "distinctive feature parsing" and decision theoretical class mapping. Each of them requires different expertise and can be developed independently. Although standard class mapping algorithms can be developed for different problems, such as classification and regression tree (CART), the "distinctive feature parsing" mechanism of reading speech should be different from that of conversational speech.

The second part of the contributions relates to investigation of two core algorithms used in traditional acoustic modeling-phonetic decision tree search and HMM training, which optimal solutions are still unknown. Two new statistical learning algorithms are proposed for overcoming known weaknesses in the existing algorithms, i.e., the difficulties in selecting optimal phonological rule sets in PDT construction and in determining optimal HMM structures. For the first problem, a knowledge-based adaptive PDT algorithm is developed on the basis of the concept of articulatary state mapping, which aims at optimizing the implicit usage of rich linguistic-phonetic information contained in large speech recognition corpuses. For the second problem, a gradient boosting algorithm is developed to jointly optimize model structure and parameters, given the fact that the Gaussian mixture densities in each HMM state is an additive model of homogeneous functions (Gaussians). Work presented here shows that each of the newly proposed algorithms can consistently improve speech recognition performance over existing algorithms.

The rest of this dissertation is organized as following: Chapter 2 introduces the speech recognition problem and gives a brief overview of automatic speech recognition, Chapter 3 discusses acoustic modeling in a statistical dynamic system framework and presents the new architecture of the knowledge-based adaptive

PDT modeling, Chapter 4 concerns the theory of HMM gradient boosting and issues of its implementation with experiments presented in Chapter 5, Chapters 6 and 7 give detailed description and experiments of the knowledge-based adaptive PDT algorithm, and conclusion as well as discussions are made in Chapter 8.

Chapter 2

STATISTICAL SPEECH RECOGNITION

2.1 The Speech Recognition Problem

The speech recognition problem, as traditionally defined, is the task of taking an utterance of speech signal and converting it into a text sequence as close as possible to what was represented by the acoustic data. The task can be viewed as a decoding problem in a source-channel representation, shown in Figure 2.1 [4].

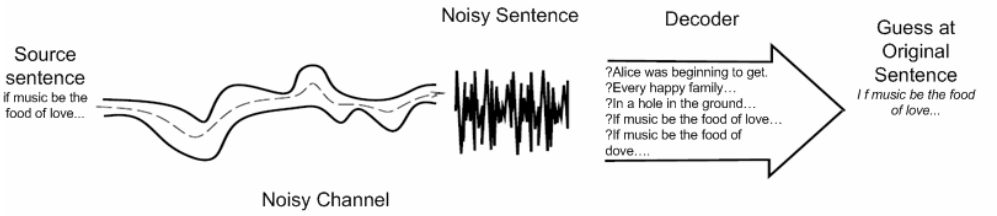

Figure 2.1 The source-channel model of speech recognition [4]

This representation begins with a speaker creating an utterance which consists of sound waves. The sound waves are then captured by a microphone and converted to electrical signals, which are then transmitted through some channels (such as telephone line or network). As a result, the original signal undergoes some known or unknown filtering and may also be contaminated by additive noise before it reaches the receiver for processing by a speech recognition system. Modern speech recognition system works by searching over a large space of sentence representations to determine the hypothesis which has the highest probability of generating the speech utterance. To do this, acoustic signals first

need to be pre-processed to generate features which retain only information necessary for speech recognition task. Second, statistical models of word-level sentence realization (N-grams), phone-level word realization (HMMs) and acoustic realization of phones (GMMs) need to be estimated from certain amount of training data. Finally, fast and memory-efficient searching algorithms are needed for picking the best match of the utterance out of a huge number of hypotheses. Figure 2.2 shows a general diagram for speech recognition.

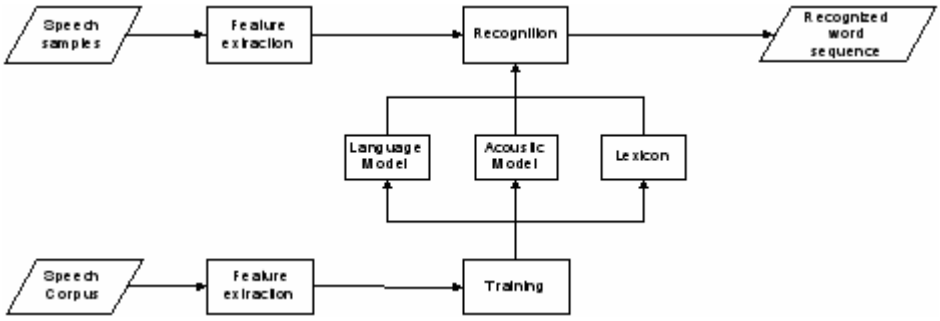

Figure 2.2 Diagram of Speech Recognition System

2.2 Statistical Speech Recognition

Given the acoustic observation sequence $O = o_1, o_2, ..., o_T$, the recognition system needs to find a word string $\hat{W}$ which is the closest guess to the sequence of words spoken by the speaker, $W^* = w_1, w_2, ..., w_N$. According to Bayesian decision theory, $\hat{W}$ is determined by a MAP decision rule, i.e.

$$\begin{aligned} \hat{W} &= \arg\max_W p(W \mid O) \\ &= \arg\max_W \frac{p(O \mid W)P(W)}{P(O)} \\ &= \arg\max_W p(O \mid W)P(W) \end{aligned} \tag{2.1}$$

where the observation likelihood $p(O|W)$ is evaluated based on an acoustic model and the prior probability $p(W)$ is determined by a language model. Note that the denominator $p(O) = \sum_{W'} p(O|W')p(W')$, the probability of acoustic observation, can be neglected because it is the same for all hypotheses *W* and will not affect the decision.

Statistical modeling for estimating the prior probability $p(W)$ of a given utterance *W* is called language modeling. The most commonly used language model is the n-gram, which uses the previous *n-1* words to predict the n^{th} word, i.e., the probability of the n^{th} word is conditional on the previous *n-1* words. These conditional probabilities are estimated from counting the relative frequencies in a large speech corpus. Given the conditional probabilities, the joint probability of sequence of words $W = w_1, w_2, ..., w_N$ can be computed by the chain rule:

$$p(w_1, w_2, ..., w_{n-1}, w_n) = \prod_{k=1}^{N} p(w_k \mid w_{k-1}, ..., w_{k-n+1}) \tag{2.2}$$

Most commonly used n-grams are bigram (n = 2) and trigram (n = 3) language models.

The acoustic model $p(O|W)$ typically consists of two parts. The first is to describe how a word sequence can be represented by sub-word units, often known as pronunciation modeling. The second is the mapping from each sub-word units to acoustic observations [3]. Algorithms used in acoustic modeling involves phonetic decision tree (PDT) and hidden Markov model (HMM). HMM will be explained in section 2.4, and introduction of PDT will be given in chapter 3.

2.3 Pre-processing of Speech

Speech recognition requires effective representation of speech signals. The raw data as input to ASR system is the speech waveform sampled at a rate perhaps 8 kHz (for telephone speech) or between 16-20 kHz. This data is pre-processed to generate feature vectors which retain only necessary information for the speech recognition task, referred to as feature extraction. Each feature vector is often computed from a 10ms frame, with an overlapped sliding window of 20 to 25 ms. Well-known feature extraction algorithms include [5]:

1. Mel Frequency Cepstral Coefficients (MFCC)- the cepstrum resulted from first warping the log energy spectrum according to the Mel frequency scale and then taking the cosine transform.

2. Perceptual Linear Prediction (PLP)- a variation of linear prediction coefficients taking into account human auditory perceptions [6].

MFCC and PLP are considered to be short-term locally stationary features and can not cover the temporal dynamics in speech. It is a common practice to use first-order and second-order time-derivatives of static features to capture such information.

Extracted features can be further transformed to improve ASR system performance. Such transformation algorithms include principal components analysis (PCA), linear discriminant analysis (LDA or HLDA [7]), vocal tract length normalization (VTLN) and independent component analysis (ICA) [8]. The ultimate goal of speech pre-processing is to produce discriminative and robust features to close the gap between the performance of human listeners and that of ASR systems. However, there is still much work remains to be done to fulfill this task.

2.4 Use of HMMs in Speech Recognition

The core statistical modeling technique in ASR system is Hidden Markov Model (HMM), which is used to model the production of speech signals and to compute the acoustic score $p(O \mid W)$ [9]. When the emphasis is on the dynamic process itself rather than on its function as a statistical model, HMM is also referred to as hidden Markov process (HMP). In an information theoretical view, HMP is defined to be a discrete-time finite state homogeneous Markov chain observed through a discrete-time memoryless invariant channel, shown in Figure 2.3 [10].

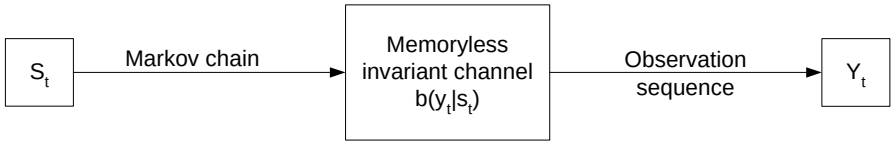

Fig. 2.3 A hidden Markov process (HMP).

where the state variable S_t, observation variable Y_t, and the emission probability $b(y_t \mid s_t)$ are as defined in the following section.

2.4.1. Statistical Definition of HMM

There is a substantial literature on the theory and application of HMM, see for instance [10], [11], [12], [13] and [14]. The definitions and notations of a basic HMM presented here follow those given in [10]. Let $\{Y_1, Y_2 ..., Y_T\}$ denote an observation sequence taking the values in an observation space Y, which is regarded as a realization of a discrete-time Markov process $\{S_1, S_2 ..., S_T\}$ that takes the values in a finite state space S. Without loss of generality denote $S = \{1,2,...,M\}$, let $\pi_j = p(S_1 = j)$ be the probability that the initial state is j and $\pi = \{\pi_1,...,\pi_M\}$ represent the prior distribution. The Markov process is

assumed (time-) homogeneous and the Markov property is specified by the transition probabilities $a_{ij} = p(S_t = j \mid S_{t-1} = i)$, where $A = \{a_{ij}\}$ is called the transition matrix. The memoryless invariant channel is described by the observation probability $b(Y_t = y_t \mid S_t = s_t)$, also known as emission probability, which denote the probability of emitting the observation from the current state. The joint density of $(Y_1,...,Y_n, S_1,...,S_n)$ can be written as

$$p(y_1,...,y_n, s_1,...,s_n) = p(y_1, s_1)\prod_{t=2}^{n} p(y_t, s_t \mid s_{t-1}) \tag{2.3}$$

Where

$$\begin{aligned} &p(y_1, s_1) = \pi_{s_1} b(y_1 \mid s_1) \\ &p(y_t, s_t \mid s_{t-1}) = a_{s_{t-1}s_t} b(y_t \mid s_t), \quad t = 2,3,... \end{aligned} \tag{2.4,2.5}$$

Using the convention $a_{s_0 s_1} = \pi_{s_1}$ for all $s_1 \in S$ and marginalizing over all state sequence, we have the likelihood function

$$p(y_1,...,y_n) = \sum_{s_1,...,s_n} \prod_{t=1}^{n} a_{s_{t-1}s_t} b(y_t \mid s_t) \tag{2.6}$$

There are usually three problems concerning learning and inference of HMM, namely, likelihood computation, parameter estimation, and state sequence decoding. These problems can be solved by the forward-backward recursion, ML parameter estimation (Baum algorithm), and dynamic programming (Viterbi algorithm) respectively, which will be introduced in the following.

2.4.2. Forward-backward Recursion

The forward-backward recursions were first introduced by Chang and Hancock [13] and later rediscovered by Baum, Petrie, Soules, and Weiss [15][16]. Define the forward probability by $\alpha_t(s_t) = p(s_t, y_1,..., y_t)$ and the backward probability by $\beta_t(s_t) = p(y_{t+1},..., y_n \mid s_t)$ with $\beta_n(s_t) = 1$. Then, we have

$$\begin{aligned} p(s_t, y_1,..., y_n) &= p(s_t, y_1,..., y_t, y_{t+1},..., y_n) \\ &= p(s_t, y_1,..., y_t)\, p(y_{t+1},..., y_n \mid s_t) \\ &= \alpha_t(s_t)\beta_t(s_t) \end{aligned} \tag{2.7}$$

for *t = 1, …, n.* Note that equation (2.7) is from the conditional independence property of sequences $\{y_1,..., y_t\}$ and $\{y_{t+1},..., y_n\}$ given state s_t. The forward and backward recursions are given in the following induction equations.

$$\alpha_t(s_t) = \begin{cases} \pi_{s_1} b(y_1 \mid s_1), & t = 1 \\ b(y_t \mid s_t) \sum_{s_{t-1}=1}^{M} \alpha_{t-1}(s_{t-1}) a_{s_{t-1}s_t}, & t = 2,..., n \end{cases}$$

$$\beta_t(s_t) = \begin{cases} 1, & t = n \\ \sum_{s_{t+1}=1}^{M} \beta_{t+1}(s_{t+1}) a_{s_t s_{t+1}} b(y_{t+1} \mid s_{t+1}), & t = n-1,...,1 \end{cases} \tag{2.8,2.9}$$

The conditional probabilities $p(s_t \mid y_1,..., y_n)$ and $p(s_{t-1}, s_t \mid y_1,..., y_n)$ can be computed as

$$p(s_t \mid y_1,..., y_n) = \frac{\alpha_t(s_t)\beta_t(s_t)}{\sum_{s_t=1}^{M} \alpha_t(s_t)\beta_t(s_t)}, \quad \mathrm{t} = 1,..., \mathrm{n}$$

$$p(s_{t-1}, s_t \mid y_1,..., y_n) = \frac{\alpha_{t-1}(s_{t-1})\beta_t(s_t) a_{s_{t-1}s_t} b(y_t \mid s_t)}{\sum_{s_{t-1}, s_t=1}^{M} \alpha_{t-1}(s_{t-1})\beta_t(s_t) a_{s_{t-1}s_t} b(y_t \mid s_t)} \tag{2.10,2.11}$$

The likelihood function can be obtained by using the forward recursion as

$$p(y_1,...,y_n)=\sum_{s_n=1}^{M}\alpha_n(s_n) \quad , \tag{2.12}$$

or by using the backward recursion as:

$$p(y_1,...,y_n)=\sum_{s_1=1}^{M}\pi_{s_1}b(y_1|s_1)\beta_1(s_1) \tag{2.13}$$

2.4.3. ML Parameter Estimation

Let $\phi=\{\pi,A,\theta\}$ denote the parameter set of a HMP, where $\theta=\{\theta_j, j=1,..,M\}$ is the parameter set of the conditional observation distributions. Assume an observation sequence $\{y_1,...,y_n\}$ was generated by an identifiable HMP with parameter $\phi^0\in\Phi$, the maximum likelihood (ML) estimator of ϕ^0 is defined by

$$\hat{\phi}=\arg\max_{\phi\in\Phi} L(\phi|y_1,...,y_n) \tag{2.14}$$

where $L(\phi|y_1,...,y_n)=\log p(y_1,...,y_n|\phi)$ is the log-likelihood function.

There is no known closed form solution for the optimization in equation (2.14) and numerical algorithms are often used. The Baum algorithm is such a computationally efficient algorithm which belongs to the family of expectation-maximization (EM) algorithms proposed by Dempster, Laird and Rubin [17]. As other EM algorithms, the Baum algorithm uses an iterative hill-climbing technique based on an auxiliary function *Q*. Let $\phi_m\in\Phi$ be the estimator at iteration *m*, and $\hat{\phi}\in\Phi$ denote a new estimator, *Q* is defined as following:

$$Q\left(\hat{\phi},\phi_m\right)=E_{\phi_m}\left[\log p\left(S_1,...,S_n,y_1,...,y_n\mid\hat{\phi}\right)\right] \tag{2.15}$$

where the observation sequence $\{y_1,...,y_n\}$ is given and the expectation is taken over the state space $\{S_1,...,S_n\}\in S^n$.

The rational of using this auxiliary function is that the increase in *Q* will result in increase in *L*, because [15],

$$\begin{aligned} L\left(\hat{\phi}\mid y_1,...,y_n\right)-L\left(\phi_m\mid y_1,...,y_n\right) &= \log\frac{p\left(y_1,...,y_n\mid\hat{\phi}\right)}{p\left(y_1,...,y_n\mid\phi_m\right)} \\ &= \log E_{\phi_m}\left[\frac{p\left(S_1,...,S_n,y_1,...,y_n\mid\hat{\phi}\right)}{p\left(S_1,...,S_n,y_1,...,y_n\mid\phi_m\right)}\right] \\ &\geq E_{\phi_m}\left[\log\frac{p\left(S_1,...,S_n,y_1,...,y_n\mid\hat{\phi}\right)}{p\left(S_1,...,S_n,y_1,...,y_n\mid\phi_m\right)}\right] \\ &= Q\left(\hat{\phi},\phi_m\right)-Q\left(\phi_m,\phi_m\right) \end{aligned} \tag{2.16}$$

by Jensen's inequality, where equality holds if and only if
$p\left(S_1,...,S_n,y_1,...,y_n\mid\hat{\phi}\right)=p\left(S_1,...,S_n,y_1,...,y_n\mid\phi_m\right)$ *a.e.*

Given *Q*, the new estimator at $(m+1)^{th}$ iteration is obtained from

$$\phi_{m+1}=\arg\max_{\phi\in\Phi}Q\left(\phi,\phi_m\right) \tag{2.17}$$

Substitute (2.6) into (2.17), the auxiliary function can be written as [16]

$$
\begin{aligned}
Q(\phi,\phi_m) &= \sum_{j=1}^{M} p(S_1 = j \mid y_1,...,y_n;\phi_m)\log \pi_j \\
&+ \sum_{i,j=1}^{M}\sum_{t=2}^{n} p(S_{t-1} = i, S_t = j \mid y_1,...,y_n;\phi_m)\log a_{ij} \\
&+ \sum_{j=1}^{M}\sum_{t=1}^{n} p(S_t = j \mid y_1,...,y_n;\phi_m)\log b(y_t \mid \theta_j)
\end{aligned}
\tag{2.18}
$$

Maximizing (2.18) gives re-estimation formulas for $\pi = \{\pi_j\}$ and $A = \{a_{ij}\}$ as following

$$
\begin{aligned}
\pi_j(m+1) &= p(S_1 = j \mid y_1,...,y_n;\phi_m) \\
a_{ij}(m+1) &= \frac{\sum_{t=2}^{n} p(S_{t-1} = i, S_t = j \mid y_1,...,y_n;\phi_m)}{\sum_{t=2}^{n} p(S_{t-1} = i \mid y_1,...,y_n;\phi_m)}
\end{aligned}
\tag{2.19,2.20}
$$

where $p(s_t \mid y_1,...,y_n;\phi_m)$ and $p(s_{t-1}, s_t \mid y_1,...,y_n;\phi_m)$ can be computed by equations (2.10) and (2.11). For Gaussian conditional observation densities, the re-estimation formulas for the Gaussian mean $\mu = \{\mu_j\}$ and covariance $\Sigma = \{\Sigma_j\}$ are given by

$$
\begin{aligned}
\mu_j(m+1) &= \frac{\sum_{t=1}^{n} p(S_t = j \mid y_1,...,y_n;\phi_m) y_t}{\sum_{t=1}^{n} p(S_t = j \mid y_1,...,y_n;\phi_m)} \\
\Sigma_j(m+1) &= \frac{\sum_{t=1}^{n} p(S_t = j \mid y_1,...,y_n;\phi_m)(y_t - \mu_j(m+1))(y_t - \mu_j(m+1))^T}{\sum_{t=1}^{n} p(S_t = j \mid y_1,...,y_n;\phi_m)}
\end{aligned}
\tag{2.21,2.22}
$$

2.4.4. Viterbi Decoding

In speech recognition, the state sequence of HMP corresponds to a sequence of classification labels (words or phoneme units). The recognition task is performed by finding the most likely state sequence and matching it with the corresponding word sequence. The state sequence is found by the following maximization:

$$\{s_1,...,s_n\}^* = \arg\max_{s_1,...,s_n} p(s_1,...,s_n \mid y_1,...,y_n) = \arg\max_{s_1,...,s_n} p(s_1,...,s_n, y_1,...,y_n) \quad (2.23)$$

This is achieved by the Viterbi algorithm, which is in fact an application of Bellman's dynamic programming algorithm [18]. Define new variable

$$V(j,t) = \max_{s_1,...,s_{t-1}} p(y_1,...,y_t, s_1,...,s_{t-1}, S_t = j) \quad (2.24)$$

which can be computed using the following recursive formulas:

$$\begin{aligned} &V(j,t) = p(y_t \mid S_t = j)\max_k \{p(S_t = j \mid S_{t-1} = k)V(k,t-1)\} \\ &k^*(j,t) = \arg\max_k \{p(S_t = j \mid S_{t-1} = k)V(k,t-1)\} \\ &\text{with initialization} \\ &V(j,1) = \max_{s_1} p(y_1 \mid s_1)\pi_{s_1} \end{aligned} \quad (2.25,2.26)$$

where $k^*(j,t)$ tracks the best previous state of state *j* at time *t*. At the end of the recursion, the optimal state at time *n* is identified by $S_n^* = \arg\max_j V(j,n)$ and the best state sequence is obtained by back-tracking using $S_{t-1}^* = k^*(S_t^*, t)$.

2.4.5. The Baum-Viterbi Algorithm

The Baum-Viterbi algorithm is also known as Viterbi extraction or segmental *k*-means in literature. Viterbi extraction was first introduced by Jelinek at IBM in 1976 [19]. This algorithm was further studied by Rabiner, Wilpon, and Juang,

where it was referred to as segmental *k*-means [20]. The name Baum-Viterbi used here follows [10] and it is more descriptive since the iteration involves Viterbi decoding and Baum's re-estimation. The Baum-Viterbi algorithm jointly optimizes the parameter and state sequence as following

$$\max_{\phi\in\Phi} \max_{s_1,...,s_n\in S^n} p(s_1,...,s_n, y_1,...,y_n \mid \phi) \tag{2.27}$$

Given a parameter estimate at the m^{th} iteration $\phi_m \in \Phi$, the best state sequence $\{s_1,..,s_n\}^*(\phi_m)$ is first estimated by the Viterbi algorithm. Then a new parameter estimate $\phi_{m+1} \in \Phi$ is obtained by maximizing (2.27) given the optimal state sequence

$$\phi_{m+1} = \arg\max_{\phi\in\Phi} p\left(\{s_1,..,s_n\}^*(\phi_m), y_1,...,y_n \mid \phi\right) \tag{2.28}$$

The auxiliary function for maximizing (2.28) is given as following

$$Q_1(\phi,\phi_m) = \sum_{s_1,...,s_n} \delta\left(\{s_1,...,s_n\} - \{s_1,...,s_n\}^*(\phi_m)\right) \log p(s_1,...,s_n, y_1,...,y_n \mid \phi) \tag{2.29}$$

where $\delta(\cdot)$ is the Kronecker delta function. Recall that the auxiliary function (2.15) for the Baum algorithm can be written as

$$Q(\phi,\phi_m) = \sum_{s_1,...,s_n} p(s_1,...,s_n \mid y_1,...,y_n;\phi_m) \log p(s_1,...,s_n, y_1,...,y_n \mid \phi) \tag{2.30}$$

Comparing equation (2.29) with (2.30), the re-estimation formulas for Baum-Viterbi algorithm can be obtained by substituting $p(s_1,...,s_n \mid y_1,...,y_n;\phi_m)$ by $\delta\left(\{s_1,...,s_n\} - \{s_1,...,s_n\}^*(\phi_m)\right)$ in corresponding formulas for Baum algorithm,

i.e., in equations (2.19), (2.20), (2.21) and (2.22). In speech recognition, this substitution is also called Viterbi approximation.

The Baum-Viterbi algorithm generates inconsistent estimators of the state sequence and parameters when the dimension of observation variables, say k, is fixed and the number of observations $n \to \infty$ [21]. In speech recognition, the number of observations in each sentence is always limited and the observations are extracted from a feature space with relatively large dimension. When $k \to \infty$, it can be shown that $p(s_1,...,s_n \mid y_1,...,y_n;\phi_m)$ converges to $\delta\left(\{s_1,...,s_n\}-\{s_1,...,s_n\}^*(\phi_m)\right)$ almost surely [22], which justifies the use of Baum-Viterbi algorithm in speech recognition applications.

2.4.6. Dynamical System Approach

The hidden Markov process in Figure 2.3 has a representation of a linear dynamical system, which can be written in a form of state-space model. The state variable S is assumed to evolve according to first-order Markov dynamics; the observed variable Y is presumed to be generated from the current state by a linear observation process. Assume that the state variable S takes continuous values, the state-space model can be described by the following equations

$$\begin{aligned} S_{t+1} &= AS_t + V_{t+1} \\ Y_{t+1} &= CS_{t+1} + W_{t+1} \end{aligned} \qquad (2.31,2.32)$$

where A is the state transition matrix, C is the observation matrix, V and W are taken to be Gaussian random variables with $V \overset{iid}{\sim} N(0,Q)$ and $W \overset{iid}{\sim} N(0,R)$. Two diagrams of state-space model are shown in Figure 2.4 [23]. In this figure, the top graph is a representation in control theory, while the bottom diagram is illustrated by a probabilistic graphical model. Probabilistic graphical models are graphs in

which nodes represent random variables and arcs represent conditional dependencies, and they provide a compact representation of joint probability distributions. Although discussions of graphical models are beyond the scope of this dissertation, it is important to point out that they provide a very useful view of the speech recognition problem, since the speech production and perception processes of human-being are far more complex than a simple hidden Markov process. We will come back to this topic in Chapter 3.

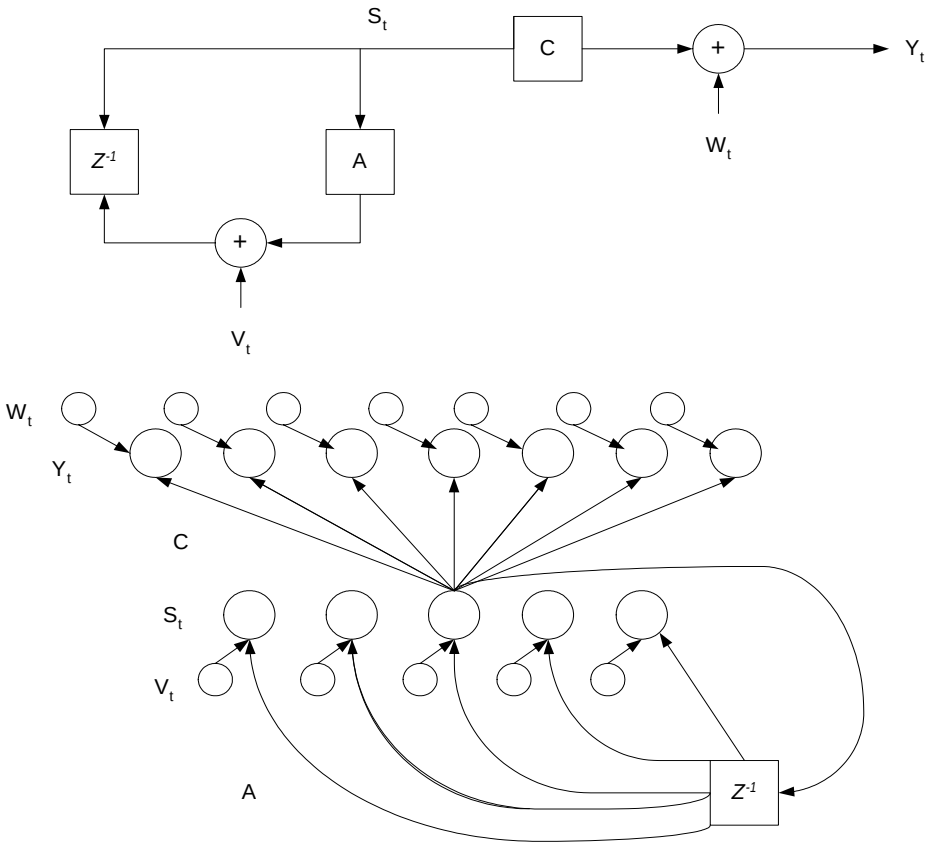

Figure 2.4 Diagrams of State-space Model

An example of continuous-state dynamical system is Kalman filter, which parameter estimation and inference algorithms are well developed [24][25][26].

When the state variable S only takes on integer values $\{1,2,...,N\}$ corresponding to N different regimes, the state equation (2.31) is equivalent to the traditional Markov chain with transition matrix A, with $\{a_{ij}\} = \{p(S_{t+1} = j \mid S_t = i)\}$. Therefore, this discrete-state dynamical system is equivalent to a standard HMP if the conditional observation densities at each state share the same covariance matrix. Note that when the covariance matrices are diagonal and not shared among different states, the observation equation (2.32) can be written as

$$Y_{t+1} = CS_{t+1} + (\sum S_{t+1})W_{t+1}$$

where W_{t+1} is i.i.d. standard (multivariate) Gaussian random variables, and each column in $\sum$ represents standard deviations in corresponding state. (2.33)

For detailed discussion of state-space model and HMP, please refer to [23][26].

A particular problem with discrete-state models is that the state sequence obtained from concatenating the most likely state estimated at each time step may have zero posterior probability, which is not an issue for continuous-state models. This is because in the discrete case, transition between two consecutive maximum a posteriori (MAP) states might not be allowed, but in the continuous case, the conditional distribution of state variable is Gaussian and the joint distribution of MAP states is the maximum among all possible state sequences. Therefore, a separate algorithm (Viterbi algorithm, see Section 2.4.4) is needed for inference of the most likely state sequence in discrete-state models, while for continuous-state models, the forward-backward recursions suffice.

2.5 Discriminative Training

In speech recognition, ML estimation of HMM parameters are traditionally used. The standard objective function for ML training is as follows

$$L_{ML}(\lambda) = \sum_{r=1}^{R} \log p_{\lambda}(Y_r \mid w_r) \tag{2.34}$$

where λ is the set of model parameters, w_r is the transcription of r^{th} utterance Y_r

which is the log-likelihood of the complete data $\{w_r, Y_r\}$:

$$L_{Model}(\lambda) = \sum_{r=1}^{R} \log p_{\lambda}(Y_r \mid w_r) p(w_r) \tag{2.35}.$$

with the prior $p(w_r)$ being dropped.

As stated in section 2.1, speech recognition is a process of classifying speech utterance into word sequence. One concern about ML training is that it is a density estimation method and might not be optimal for classification tasks such as speech recognition. Recent developments in non-ML training methods address this problem by using a decision theoretic framework. In literature, this family of algorithms is often called discriminative training method. Define sentence string w_r as class label for utterance Y_r, the training problem can be conducted by minimizing the following empirical risk:

$$L(\lambda) = \sum_{r=1}^{R} Q_{\lambda}(w_r, Y_r) \tag{2.36}$$

where Q_{λ} is the loss function

The best-known discriminative algorithms are the maximum mutual information (MMI) [2, 27] and minimum classification error (MCE) [28]. In MMI, the loss function is based on the empirical Bayesian risk with $0/1$ loss [29, 30], i.e., MMI maximizes the posterior probabilities of observed word sequences, written as following

$$L_{MMI}(\lambda)=\sum_{r=1}^{R}\log p_{\lambda}(w_r\,|\,Y_r)=\sum_{r=1}^{R}\log\frac{p_{\lambda}(Y_r\,|w_r)p(w_r)}{\sum_{w}p_{\lambda}(Y_r\,|\,w)p(w)} \tag{2.37}$$

The MCE approach simultaneously minimizes the empirical error rate of the recognizer over all training utterances. The recognition error is represented by an indicator variable and is approximated by the sigmoid function [29]. Let

$$d_r(Y_r)=-\log\frac{p_{\lambda}(Y_r\,|\,w_r)p(w_r)}{\left[\sum_{w\neq w_r}\left(p_{\lambda}(Y_r\,|\,w)p(w)\right)^{\eta}\right]^{1/\eta}},\quad \eta\geq 1 \tag{2.38}$$

Note that (2.38) is not exactly the same as equation (13) in [28], where the priors $p(w_r)$ and $p(w)$ are not present. For large η

$$d_r(Y_r)\approx-\log\frac{p_{\lambda}(Y_r\,|\,w_r)p(w_r)}{\max_{w\neq w_r}\left(p_{\lambda}(Y_r\,|\,w)p(w)\right)},\quad \eta>>0 \tag{2.39}$$

where the numerator represents the posterior of true transcription and the denominator is the best competing alternative hypothesis. From the MAP decision rule in recognition, the indicator variable of classification error is

$$I(d_r(Y_r))=\begin{cases}1, & d_r(Y_r)>0\\ 0, & \text{otherwise}\end{cases} \tag{2.40}$$

Approximate this indicator variable with sigmoid function and sum over all training utterances, the MCE objective function can be written as

$$L_{MCE}(\lambda)=\sum_{r=1}^{R}\frac{1}{1+\exp(-\gamma d_r(Y_r))}\quad \gamma>0 \tag{2.41}$$

In [26], MMI and MCE are generalized to a family of unified discriminative training criteria, defined by the following expression:

$$L_{Disc}(\lambda) = \frac{1}{R}\sum_{r=1}^{R} f\left(\log \frac{p_\lambda(Y_r \mid w_r)p(w_r)}{\sum_w p_\lambda(Y_r \mid w)p(w)}\right) \tag{2.42}$$

where f is called the smoothing function, which is actually a special form of the loss function Q in (2.36). It is also shown that MCE training can be made equivalent to the extended Baum algorithm (EBW) for MMI training through scaling the accumulated statistics by $\frac{\partial f(x)}{\partial x}$ in EBW iteration (f is the sigmoid function), which can be easily shown by taking the gradient in (2.42) [27].

2.6 Difficulties in Speech Recognition

Speech recognition is a difficult problem due to the ambiguity in language generation, complexity in speech production, and variation in acoustic signals. Natural language has inherent ambiguities, for example, homophones and boundary ambiguity. Homophones refers to two different words which sound the same; boundary ambiguity appears when there are multiple ways of grouping phones into words. Normal speech is usually filled with hesitations, repetitions, filled-pauses, and there is often reduction of morphemes and words in pronunciation.

In acoustic realization of phonemes, the composition of basic contrastive sound units is strongly dependent on the context, speaking style, speaking rate, or voice quality. Variations in speaker physical and emotional characteristics and regional or social dialects also account for large variations in pronunciation of speech. Finally, change in the acoustic environment (noise) or the communication

channel can create additional variability in speech. All these instabilities in speech are big challenges for designing a robust ASR system [3].

Chapter 3

STATISTICAL ACOUSTIC MODELING

3.1 Information's Role in Speech Recognition

Traditional automatic speech recognition systems use the phone as basic symbolic representation of speech. Frames of acoustic features (MFCCs) are associated with specific phonetic units and form a sequence of phones which are obtained by lexical expansion of words. In speech recognition literature, this is often referred to as the "beads-on-a-string" procedure. Despite the long use of phone as fundamental units in speech recognition, it is increasingly apparent that more sophisticated models are required to incorporate linguistic and phonetic knowledge into the ASR system. For example, a study on the Switchboard corpus showed significant amount of non-canonical (standard) phonetic phenomena in spontaneous speech, including spurious friction, devoicing and acoustic cue trading, which suggests that articulatory patterns can not be purely inferred from biomechanical factors [31]. In [32], an acoustic model clustering approach was developed based on syllable structure in addition to contextual phonemes, and significant improvement in performance over traditional systems was observed.

Recently, attempts were made to design a next-generation knowledge-based large vocabulary speech recognition system, as described in [33]. This system replaces traditional acoustic features (MFCC) with some distinctive features obtained from a bank of phonetic event detectors (neural nets), which are later used in phonological inference. The distinctive feature space consists of variables induced by articulatory phonetics and hence such defined features are knowledge-based. For example, 60 phonetic attributes were used in [34] and 5 multiple-categorical

valued articulatory features were defined in [35]. An articulatory state detection based speech recognition system is shown in Figure 3.1.

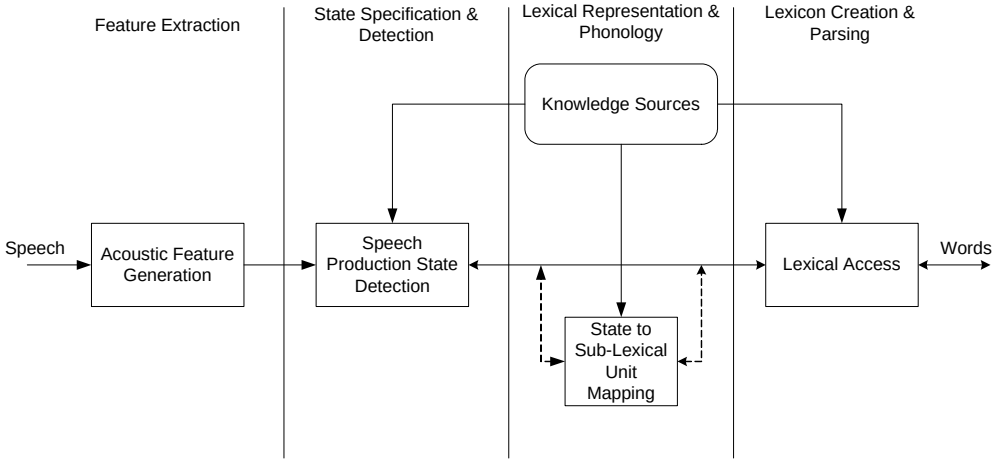

Figure 3.1 An articulatory state detection based speech recognition framework [35]

The performance of this system depends on the goodness of the knowledge and its efficient handling. Conceptually, by defining the articulatory state variable in a phonological-phonetic space, it resolves to some extent the ambiguity caused by PDT tying in traditional ASR systems. Although it is a promising direction toward a next-generation system, a lot of problems are waiting to be solved. In the following sections, we will focus on how to maximize the usage of knowledge in existing ASR systems.

3.2 Dynamical System Revisited

In the next generation ASR system design, the knowledge sources encompass a vast field of disciplines, including speech science, acoustics, linguistics and cognitive science [34]. Combining all the knowledge, what will be the observables in speech processing? An incomplete review in literature shows that most observations in speech processing fall into three major categories: distinctive

variables associated with the word sequence, biomechanical factors related to articulatory movements, and acoustic features generated by the vocal tract. In a speech production view [36], these observables are summarized in Figure 3.2.

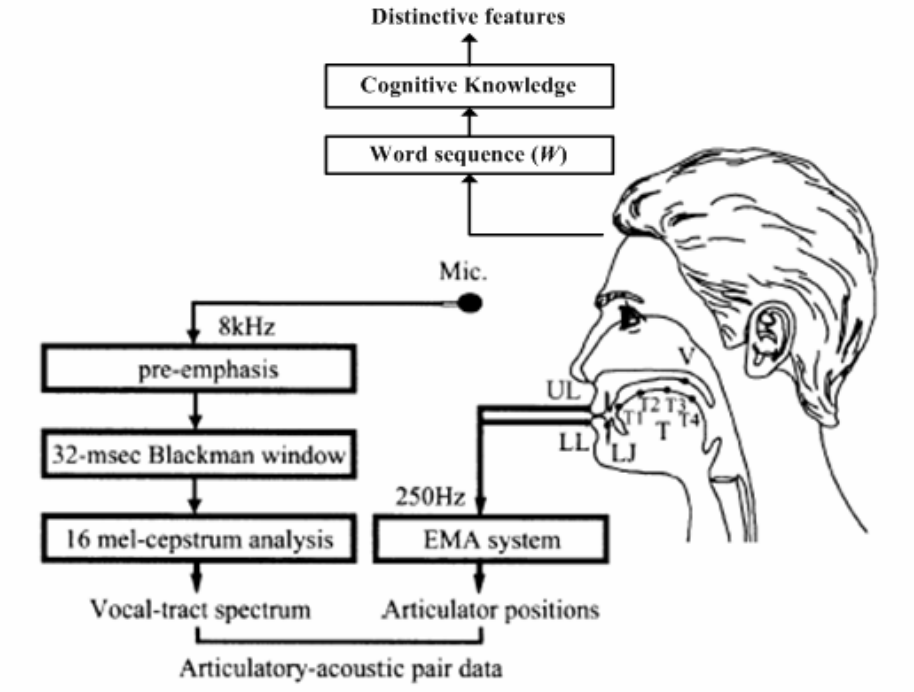

Figure 3.2 Observable Features in Speech Processing-in a Speech Production View [36]

where vocal-tract spectrum is derived from recorded waveform, articulator positions are directly measured with an electro-magnetic articulographic (EMA) system [37], and distinctive features are extracted from word sequence by applying cognitive knowledge of the speaker. Denote acoustic features as $Y = \{Y_1, Y_2, ..., Y_T\}$, articulatory features as $X = \{X_1, X_2, ..., X_T\}$, distinctive features as $V = \{V_1, V_2, ..., V_T\}$, and hidden articulatory state variables as $S = \{S_1, S_2, ..., S_T\}$ for a word sequence W which is formulated according to

some knowledge *M*, the speech production process can be represented by a multi-layer dynamical system, shown in Figure 3.3.

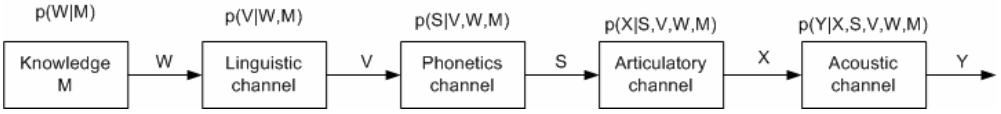

Figure 3.3. Speech production dynamics

Assuming the complete dynamics can be described by some model Λ, estimation and inference of this system can be performed as usual by mixmum likelihood estimation ($\hat{\Lambda} = \arg\max_{\Lambda} p(Y \mid W, \Lambda)$) and MAP rule ($\hat{W} = \arg\max_{W} p(W \mid Y, \Lambda)$). However, considering complexity of the problem and number of variables involved, modeling such a multi-layered system is hard even with the power of graphic models. In practice, the whole system is often simplified and reduced to manageable tasks by "divide and conquer". For instance, in traditional ASR systems, language modeling, phonetic decision trees (PDTs) and hidden Markov models (HMMs) are used to determine the conditional probabilities $p(W \mid M)$, $p(S \mid V, W, M)$ and $p(Y, X \mid S, V, W, M)$ correspondingly, where $p(S \mid V, W, M)$ is actually made deterministic (taking $0/1$ values) by PDTs.

One major benefit of deterministic mapping from linguistic space to articulatory space by PDTs is to reduce the computation needed in decoding. To show this, consider an alternative approach for modeling variables V, S and X by input-output hidden Markov models (IO-HMMs). IO-HMMs are HMMs whose emission probabilities of the output sequence, X, and transition probabilities are conditional on an input sequence, V [38]. It represents the statistical relationship between the input and output observations modulated by a hidden state variable. Estimation and inference of IO-HMM are based on conditional probabilities

$p(X \mid V)$ and $p(V \mid X)$, which can be computed by a dynamic Bayesian network (DBN), shown in Figure 3.4

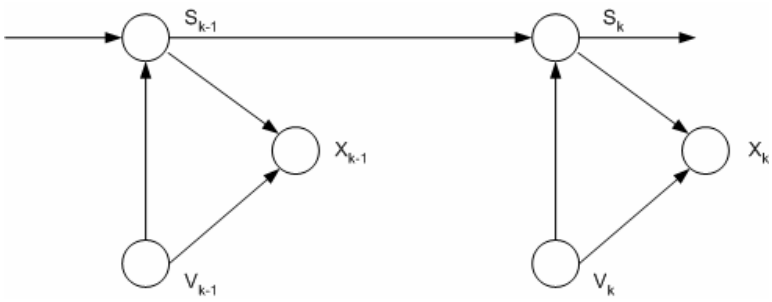

Figure 3.4 DBN representation of IO-HMM

It can be seen that computing $p(V \mid X)$ involves integration over the state space and decoding will incur much more computation than traditional PDT based methods. One may also notice that IO-HMM provides another way to compute $p(V \mid X)$ which is one of the objectives of the next generation ASR system discussed in Section 3.1.

To summarize, in this section we gave a complete view of speech production dynamics, investigated possible observable information sources and discussed estimation and inference problems under a general statistical formulation. In next section, we will examine the traditional acoustic modeling approach under a simplified framework for modeling speech production dynamics and explore opportunities of improving current ASR systems with the "knowledge-based" concept. The focus will be on distinctive features V and their use in PDT construction on which few studies were found in speech recognition literature.

3.3 Acoustic Modeling System

Two questions are often asked before designing an acoustic modeling system: first, "what entities of the phonological process can be associated with the articulatory state?", and second, "how speech sounds are pronounced at the articulatory state subject to vocal-tract constraints?". These two questions corresponds to different layers of speech production dynamics in Figure 3.3, reside in different space, and are traditionally resolved by different modeling techniques. A Bayesian network (BN) representation of the simplified dynamics is shown in Figure 3.5

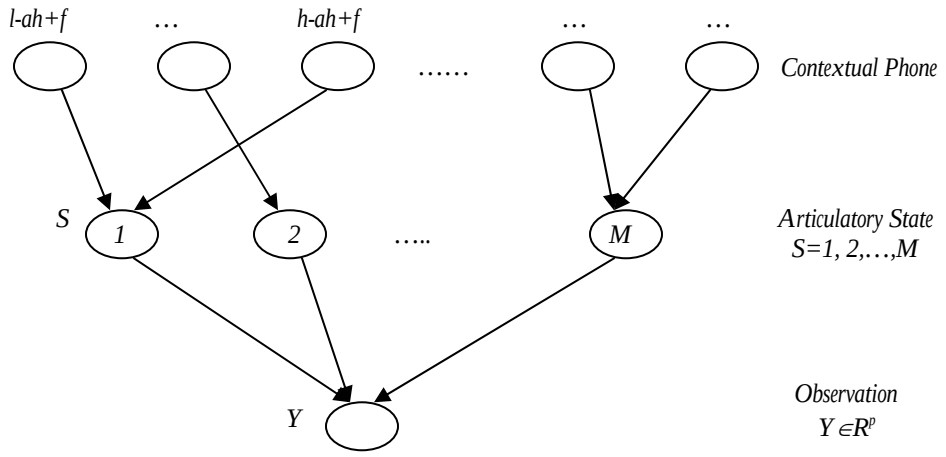

Figure 3.5 BN representation for traditional modeling of speech dynamics

In most existing ASR systems, the mapping from contextual phones to articulatory states is deterministic by phonetic decision trees. Therefore, this layer of dynamics in Figure 3.5 is eliminated by associating (multiple) contextual phone labels with a particular articulatory state, and the BN representation is compacted to a simple hidden Markov process. One known weakness of this approach is that certain degree of ambiguity is introduced by deterministically associating multiple contextual phones with a single state, because in inference, these tied

phones can not be distinguished by their acoustic scores (which are also shared), and this kind of ambiguity can only be handled by lexicon and language models in decoding. However, given the intrinsic ambiguity in language itself, leaving the ambiguity in acoustic models untreated and passing it to language models is risky, and sometimes disastrous if the PDTs are ill-structured.

The second level of dynamics, and most often, the core dynamics in current ASR systems, is the process of observations modulated by articulatory states, which is modeled by a hidden Markov process with mixture of multivariate Gaussian conditional observation distributions (GMM-HMMs). Such models have two regime variables, S_t for state at time *t* and H_t for the mixture component in state S_t, and is a simple extension of HMM with single Gaussian emission densities. Without loss of generality, assume H_t take values from $\{1,...,J\}$, and let $w_{l|j} = p(H_t = l \mid S_t = j)$. Using the conditional independence properties, equation (2.6) can be re-written as

$$p(y_1,...,y_n) = \sum_{s_1,...,s_n} \sum_{h_1,...,h_n} \prod_{t=1}^{n} a_{s_{t-1}s_t} w_{h_t|s_t} b(y_t \mid s_t, h_t) \qquad (3.1)$$

The principles presented in section 2.4 can be easily applied to GMM-HMMs.

3.4 PDT and Distinctive Features

In traditional ASR systems, PDT is the only bridge linking the dynamics between linguistic and articulatory-phonetic channels, playing an important role in message passing. The architecture of PDTs subject to the variations in language and speech production and should not be deemed as the same over different speech domains. For instance, studies conducted in [32] and [39] demonstrated strong evidence that PDTs generated from reading speech are different from those of

conversational speech. However, decision tree algorithms have been traditionally viewed as a nonparametric approach [40] and adaptation of tree structures to new domain of speech remains an open problem in speech recognition. In this section, we will first give a parametric representation of PDT structures under a statistical approach to decision tree modeling as proposed in [41], continue with discussions on applying prior knowledge to guide structural changes in adapting PDTs, and finally come with a PDT-adaptive system design.

3.4.1. Statistical Decision Tree Modeling

The probability models for decision trees will be introduced here follows what is given in [41]. Assume in a decision tree, a given input *x* follows a sequence of probabilistic decisions and hit a state which generates a corresponding output *y*. A probabilistic model of a decision tree involves a sequence of probabilistic decisions, each conditional on the input *x* and on previous sequence of decisions.

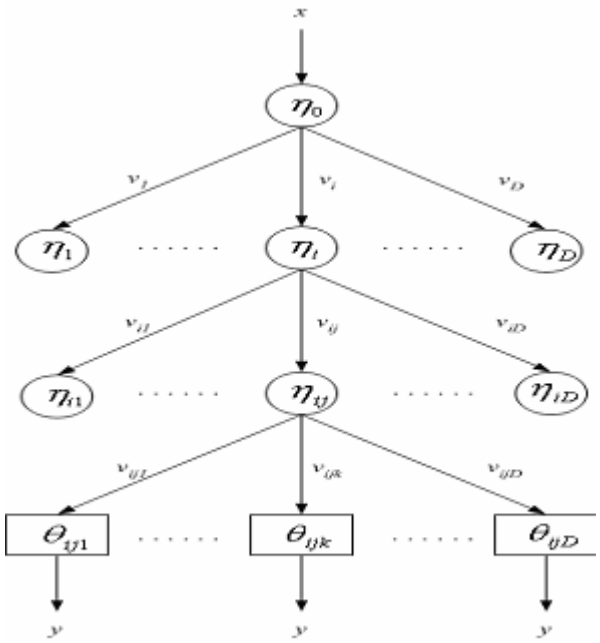

Figure 3.6 Probabilistic decision tree

Introduce a set of random decision variables $V = \{v_1,...,v_D\}$, a diagram of a probabilistic decision is shown in Figure 3.6, with additional parameters list on the nodes. Staring from the root node, the first decision is modeled by a probability distribution of random decision variables V, conditional input x and node-specific parameters η_0, defined as

$$p(v_i \mid x, \eta_0) \tag{3.2}$$

The second decision depends on the first with conditional probabilities

$$p(v_{ij} \mid x, v_i, \eta_i) \tag{3.3}$$

Propagating toward the leaf, the conditional observation probability is of the form

$$p(y \mid x, v_i, v_{ij}, ..., \theta_{ij...k}) \tag{3.4}$$

where $\theta = \{\theta_{ij...k}\}$ is the set of parameters of emission probabilities. Assuming Markov property, the likelihood of y given x can be obtained by

$$\begin{aligned} p(y \mid x) = & \sum_i p(v_i \mid x, \eta_0) \sum_j p(v_{ij} \mid x, v_i, \eta_i) \\ & ... \sum_k p(v_{ij...k} \mid x, v_i, v_{ij}, ..., \eta_{ij...k^-}) p(y \mid x, v_i, v_{ij}, ..., \theta_{ij...k}) \end{aligned} \tag{3.5}$$

where k^- is the index preceding k as in the sequence $i,j,\ldots, k^-,k$. Using Bayes rule, the posterior probability of the decision sequence at depth k can be written as

$$p(v_i, v_{ij}, ..., v_{ij...k} \mid x, y) = \frac{p(v_i \mid x, \eta_0) p(v_{ij} \mid x, v_i, \eta_i) ...}{\sum_i p(v_i \mid x, \eta_0) \sum_j p(v_{ij} \mid x, v_i, \eta_i) ...} \cdot \frac{... p(v_{ij...k} \mid x, v_i, v_{ij}, ..., \eta_{ij...k^-}) p(y \mid x, v_i, v_{ij}, ..., \theta_{ij...k})}{... \sum_k p(v_{ij...k} \mid x, v_i, v_{ij}, ..., \eta_{ij...k^-}) p(y \mid x, v_i, v_{ij}, ..., \theta_{ij...k})} \tag{3.6}$$

which can be computed using the upward-downward recursions [41], which is analogous to the forward-backward algorithm in HMMs.

Each subtree in a decision tree by itself is a decision tree and is a structural unit of its parent tree. Since each decision sequence is associated with a tree node, the posterior probability obtained from (3.6) assigns a credit score to the corresponding node and the subtree beneath it. Therefore, (3.6) provides a goodness measure of subtrees, which can potentially leads to adaptive learning of decision tree structures within a Bayesian framework.

3.4.2. Distinctive Feature as Decision Variable

In speech recognition, it is common practice to resort to higher level dynamics in speech production for decision variables. In traditional PDT modeling, decision variables are often called phonetic questions. For example, a total of 202 questions were used in the HTK system [44]. As previously discussed, these questions belong to a bigger class of knowledge-based features, called distinctive features. One important benefit of knowledge-based decision variable is to provide easy handling of unseen triphones.

Refer to the multi-layered speech production dynamics shown in Figure 3.3, distinctive feature V is an obvious choice for the role of random decision variable in PDT modeling. Defining a suitable set of distinctive features for speech recognition requires expertise knowledge in linguistic related fields and is an

immerging research topic. Note that the distinctive features used in [34][35] are closely related to traditional phonetic questions and far from complete.

3.4.3. Knowledge-based Phonetic Decision Tree Modeling

So far we have completed review of the theoretical backgrounds for designing a knowledge-based adaptive phonetic decision tree modeling framework, in the trend toward next generation ASR systems, which is part of the mission of this research. The proposed system is shown in Figure 3.7

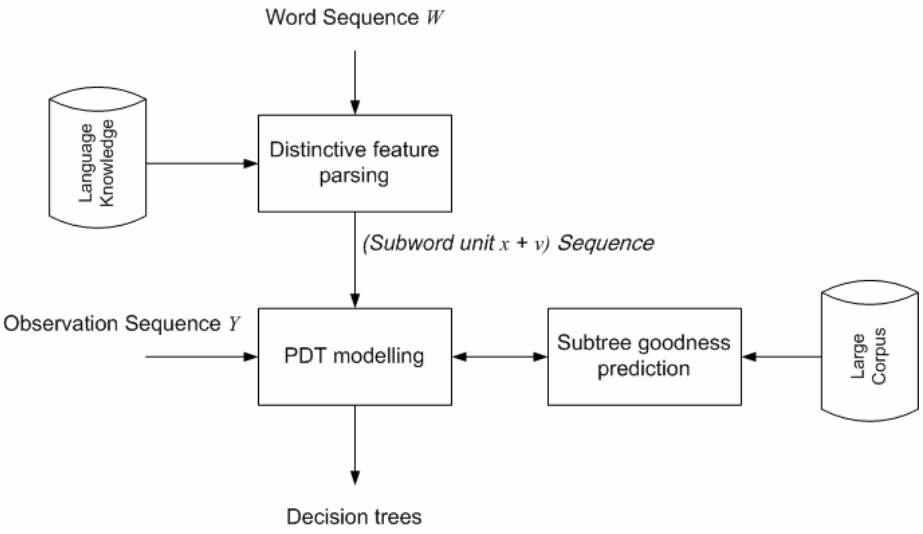

Figure 3.7 Proposed knowledge-based adaptive PDT modeling

Sub-word units *x* are often taken as lexical phonemes, or sub-phonemes when left-right HMM is used to represent a single phoneme, and distinctive features *v* consist of various factors which influence pronunciation variation (such as surrounding phones, stress and syllable structure, part-of-speech, etc.). Instead of specifying a parametric expression for the tree prior, we approximate it implicitly by a tree-generating process based on a large corpus, which is used to select good candidate decision variables for PDT splitting. These candidate decision variables

represent credible realizations of sub-trees and therefore called sub-tree goodness prediction. Note that the PDT modeling is still a greedy search with implicitly specified priors on decision variables.

Existence of a larger corpus for inference of prior knowledge is a valid assumption in most ASR applications, because each application confines to a specific domain which shares the same general pronunciation rules with a larger domain. Well-known existing large corpora include WSJ (Word Street Journal) for read speech and Switchboard for conversational speech. For tasks where large corpora are not available, a Monte Carlo strategy [42][43] can be used for empirical analysis of sub-tree instead of the proposed greedy algorithm.

3.5 PDT Algorithm

A phonetic decision tree is a binary tree which recursively partition the pronunciations of a lexical phoneme (or sub-phoneme unit) specified by distinctive features into subsets in which the acoustic features distributed more homogeneously, referred to as "surface forms" which are realizations of associated articulatory states. Each tree is built using the greedy algorithm by sequentially choosing splitting rules for nodes so as to maximize the increase in log likelihood. This process generates a sequence of trees with increasing sizes, and stops when the increase in log likelihood falls below a threshold.

Observation sequence $Y = \{y_t\}$ of acoustic features and sub-word sequence (associated with distinctive features) $O = \{(x+v)_i\}$ need to be first paired in probability by the forward-backward recursions. This requires direct access to speech data, which is computationally expensive for large data sets. Therefore, efficient PDT algorithms have been developed, although with limitations, by [44]

- structured organization of data: let $C = \{c_j\}$ denote the set of unique classes specified by the distinctive features (traditionally defined as triphones), sufficient statistics of training data are accumulated according to their class label $c \in C$, and

- fast computation of log-likelihood based on sufficient statistics.

Assuming single Gaussian conditional observation density, fast computation is achieved by approximating the total log likelihood by a simple average of the log likelihoods weighted by state occupancy:

$$L = \sum_{r=1}^{R} \sum_{t=1}^{T_r} \sum_{c \in C} \ln\left(p\left(y_t^r \mid \mu_c, \Sigma_c\right)\right) p\left(s_t^r = c \mid y^r\right) \approx \ln p(Y \mid C) \tag{3.7}$$

where r represent the r^{th} utterance in training corpora, $p\left(y_t^r \mid \mu_c, \Sigma_c\right)$ is single Gaussian, i.e.

$$\begin{aligned} \ln\left(p\left(y_t^r \mid \mu_c, \Sigma_c\right)\right) &= \ln\left((2\pi)^{-\frac{d}{2}} |\Sigma_c|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} \left(y_t^r - \mu_c\right)^T \Sigma_c^{-1} \left(y_t^r - \mu_c\right) \right) \right) \\ &= -\frac{1}{2} \left(d \ln(2\pi) + \ln|\Sigma_c| + \left(y_t^r - \mu_c\right)^T \Sigma_c^{-1} \left(y_t^r - \mu_c\right) \right) \end{aligned} \tag{3.8}$$

therefore

$$L = \sum_{r=1}^{R} \sum_{t=1}^{T_r} \sum_{c \in C} -\frac{1}{2} \left(d \ln(2\pi) + \ln|\Sigma_c| + \left(y_t^r - \mu_c\right)^T \Sigma_c^{-1} \left(y_t^r - \mu_c\right) \right) p\left(s_t^r = c \mid y^r\right) \tag{3.9}$$

From (2.22), we have

$$\Sigma_c = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T_r} p\left(s_t^r = c \mid y^r\right)\left(y_t^r - \mu_c\right)\left(y_t^r - \mu_c\right)^T}{\sum_{r=1}^{R}\sum_{t=1}^{T_r} p\left(s_t^r = c \mid y^r\right)}$$

hence

$$\sum_{r=1}^{R}\sum_{t=1}^{T_r}\left(y_t^r - \mu_c\right)^T \Sigma_c^{-1}\left(y_t^r - \mu_c\right) p\left(s_t^r = c \mid y^r\right) = d\sum_{r=1}^{R}\sum_{t=1}^{T_r} p\left(s_t^r = c \mid y^r\right) \qquad (3.10)$$

substitute into (3.9) gives

$$L = \sum_{c\in C} -\frac{1}{2}\left(d\left(1+\ln\left(2\pi\right)\right) + \ln\left|\Sigma_c\right|\right)\sum_{r=1}^{R}\sum_{t=1}^{T_r} p\left(s_t^r = c \mid y^r\right) \qquad (3.11)$$

This is the logarithm of the joint likelihood of all the sub-word units. Any decision tree will be a partition on *C*. Let *B* be a binary decision tree with leaf node $b \in B$, the log likelihood of *B* can be obtained by rewriting (3.11) as

$$L = \sum_{b\in B} -\frac{1}{2}\left(d\left(1+\ln\left(2\pi\right)\right) + \ln\left|\Sigma_b\right|\right)\sum_{r=1}^{R}\sum_{t=1}^{T_r} p\left(s_t^r = b \mid y^r\right) \qquad (3.12)$$

the covariance matrix of node *b* can be computed by

$$\Sigma_b = E\left(y^2\right) - \left(E\left(y\right)\right)^2 = \frac{\sum_{c\in B(b)}\gamma_c\left(\Sigma_c + \mu_c\mu_c^T\right)}{\sum_{c\in B(b)}\gamma_c} - \left(\frac{\sum_{c\in B(b)}\gamma_c\mu_c}{\sum_{c\in B(b)}\gamma_c}\right)\left(\frac{\sum_{c\in B(b)}\gamma_c\mu_c}{\sum_{c\in B(b)}\gamma_c}\right)^T \qquad (3.13)$$

where $B(b)$ represent the partition by tree *B*, $\gamma_c, \mu_c, \Sigma_c$ are accumulated sufficient statistics

$$\gamma_c = \sum_{r=1}^{R} \sum_{t=1}^{T_r} p\left(s_t^r = c \mid y^r\right) \tag{3.14}$$

$$\mu_c = \frac{\sum_{r=1}^{R} \sum_{t=1}^{T_r} p\left(s_t^r = c \mid y^r\right) y_t^r}{\gamma_c} \tag{3.15}$$

$$\sum_c = \frac{\sum_{r=1}^{R} \sum_{t=1}^{T_r} p\left(s_t^r = c \mid y^r\right)\left(y_t^r - \mu_c\right)\left(y_t^r - \mu_c\right)^T}{\gamma_c} \tag{3.16}$$

During each split of a leaf node, in order to use (3.12), only the covariance matrices and accumulated state occupancies of newly generated child nodes need to be computed, assuming that the values of the parent node have already been calculated. The computation complexity is only dependent on the number of sufficient statistics stored in the parent node since the split is a local operation.

3.6 GMM-HMM

In this section, we give the Baum re-estimation formulas for GMM-HMM which was defined in (3.1). The formulas for prior π and transition A are the same as (2.19) and (2.20). For the parameters of the Gaussian mixture emission densities, the estimates are given by

$$w_{jl} = \frac{\sum_{t=1}^{T} p\left(S_t = j, H_t = l \mid Y\right)}{\sum_{t=1}^{T} p\left(S_t = j \mid Y\right)} \tag{3.17}$$

$$\mu_{jl} = \frac{\sum_{t=1}^{T} p(S_t = j, H_t = l \mid Y) y_t}{\sum_{t=1}^{T} p(S_t = j \mid Y)}$$

$$\sum_{jl} = \frac{\sum_{t=1}^{T} p(S_t = j, H_t = l \mid Y)(y_t - \mu_{jl})(y_t - \mu_{jl})^T}{\sum_{t=1}^{T} p(S_t = j \mid Y)} \qquad (3.18,3.19)$$

where $p(S_t = j, H_t = l \mid Y) = p(S_t = j \mid Y) \frac{w_{jl} b(y_t \mid S_t = j, H_t = l)}{b(y_t \mid S_t = j)}$.

Chapter 4

GRADIENT BOOSTING OF HMMS

4.1 Introduction

Two important issues in EM algorithm are local optima and model complexity, which depend on the initialization of mixture components. In Gaussian mixture modeling of phone units, local optima often involve overlapped mixture components in over-populated center regions and too few components near class boundary. Furthermore, model complexity can not be determined within the ML framework since more complex models usually result in higher likelihood, which may ultimately lead to overtraining.

The key problem associated with local optima and model complexity is the lack of schemes which can jointly optimize model structure and parameters. In this chapter, we present the general framework of gradient boosting learning to address above problems. The theory of gradient boosting learning was first introduced in statistics literature. Friedman developed a general gradient-descent boosting paradigm for additive expansions of functions based on any fitting criterion [45]. This paradigm is extended to estimation of GMD based HMMs in our algorithm where GMDs are additive in nature. In addition, a partial EM algorithm for optimal component search is developed based on the ML criterion. In this new framework, GMDs are recursively constructed in a greedy manner—an optimal new component is located and inserted to the mixture model. In comparison with conventional algorithms, it offers a mechanism of dynamically allocating new components outside the local optimum regions. Conceptually, this algorithm differs from optimal splitting algorithm in that it uses an optimal insertion step instead of splitting, where the new component is found by a global

search to avoid local optima. To illustrate the difference of gradient boosting and conventional EM, in Figure 4.1, we show the 16-Gaussian component densities for a sub-phonetic unit /dh/ generated by the two methods. The densities are plotted against the first 2 principle components derived from 39 speech feature components. It can be seen that gradient boosting has more focus on the class boundaries.

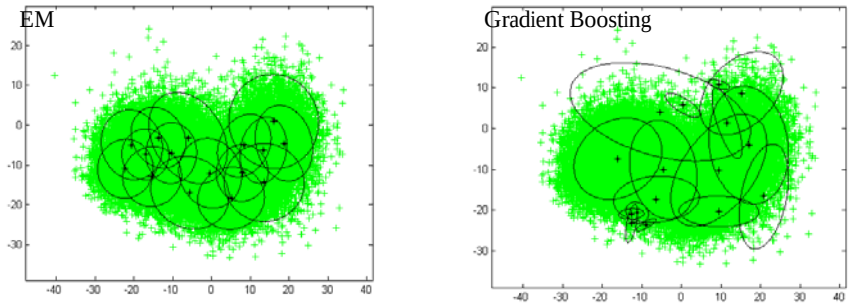

Figure 4.1.Two mixtures of 16 Gaussians obtained by EM & GB

4.2 Gradient Boosting Learning

In conventional parametric methods for estimation of function $F(x;\Lambda)$, model parameters Λ are estimated by optimizing some specified objective function $L(F(x;\Lambda))$. For most $F(x;\Lambda)$ and $L(F(x;\Lambda))$, closed form solution is difficult to find and numerical optimization methods are used. When steepest-decent method is used, the solution can be expressed as a sum of subsequent steps starting from an initial guess λ_0, i.e., $\Lambda_m = \sum_{i=0}^{m} \lambda_i$, where $\lambda_i = \rho_i g_i, i = 1,..,m$ is the incremental step of size ρ_i taken at the direction g_i. In contrast to conventional methods, gradient boosting learning targets the function

approximation problem from the perspective of numerical optimization in function space, rather than parameter space. The solutions seeking are "additive" expansions of the form

$$F(x;\Lambda)=\sum_{m=0}^{M}\alpha_m h(x;\theta_m) \tag{4.1}$$

where $h(x;\theta_m)$ is a basis function characterized by parameters θ_m, which is usually chosen as the best fit of the gradient in the function space at stage *m*, and α_m is the step size. Given N training observations $X=(x_1,\dots x_N)$, the general paradigm of gradient boosting contains the following steps [45]:

Algorithm 1: Gradient Boost

1. Initialize $F_0(x;\Lambda)$.
2. For $m = 1$ to M do:
3. $g_i=\left[\frac{\partial L(F(x_i;\Lambda))}{\partial F(x_i;\Lambda)}\right]_{F(x)=F_{m-1}(x)}, i=1,\dots,N.$
4. Fit basis function $h(x_i;\theta_m)$ to $\{g_i\}$.
5. $\alpha_m=\arg\max_{\alpha}\sum_{i=1}^{N}L(F_{m-1}(x_i;\Lambda)+\alpha h(x_i;\theta_m)).$
6. $F_m(x;\Lambda)=F_{m-1}(x;\Lambda)+\alpha_m h(x;\theta_m).$
7. End For

The analogy of gradient boosting to steepest-descent gives insight to estimation of GMDs in the model space instead of parameter space. Our goal is to estimate a probability density function $f(x;\Lambda)$ which optimizes some specified objective function $L(f(x;\Lambda))$ with the solution in the form of mixture of Gaussians $f(x)=\sum_{i=1}^{k}\alpha_k N_k(x)$ to obtain largest gain of $L(f(x;\Lambda))$ in a steepest-descent manner.

Special properties associated with GMD estimation present difficulties in direct application of gradient boosting. First, the sequential learning equation in line 6 needs to be constrained by being a proper GMD density function. This can be assured by defining the new GMD to be

$$f_m(x;\Lambda)=(1-\alpha_m)f_{m-1}(x;\Lambda)+\alpha_m N(x;\theta_m) \quad 0<\alpha_m<1 \qquad (4.2)$$

Second, fitting the steepest-descent direction in line 4 is sensitive to low valued probabilities. For example, in the case of MLE, the gradient of log-likelihood function is $g_i=\frac{\partial \log f(x_i;\Lambda)}{\partial f(x_i;\Lambda)}=\frac{1}{f(x_i;\Lambda)}$. This implies fitting a bell-shaped Gaussian kernel to the reciprocals of current probabilities, which could approach infinity when $f(x;\Lambda)$ is small. Third, steepest-descent methods have known problems of local optima. To overcome these problems, we developed an alternative searching procedure to obtain the basis function in line 4. This scheme consists of candidate generation, re-estimation and selection. In our candidate generation design, all candidates are obtained by randomly splitting the existing Gaussian components, which will maintain appropriate coverage of the model space. Each candidate is re-estimated using local data and its contribution to the improvement in the objective function is measured. The one which contributes the most to the objective function is chosen as the new component. Within this

scheme, the entire model space is covered by the globally generated candidates, and hence local optima can be alleviated. More details on new component allocation will be discussed in section 4.4.

Model complexity is one important issue in GMD estimation. The best value for number of components M can be determined by model selection methods, such as BIC, cross-validation, etc. By considering the GMD-related issues and incorporating model selection criterion, the gradient boosting algorithm for S-class GMDs $\{f_1,..., f_S\}$ is formulated as following:

Algorithm 2: GMD Gradient Boost

1. Initialize $f_{s,0}(x;\Lambda_s) = N(x;\theta_{s,0})$, $s = 1,\ldots,S$, set $m = 1$.
2. For $s = 1$ to S do:
3. Find a basis Gaussian $N(x;\tilde{\theta}_{s,m})$.
4. $\{\alpha_{s,m},\theta_{s,m}\} = \arg\max_{\alpha,\theta} \sum_{i=1}^{N} L((1-\alpha) f_{s,m-1}(x_i;\Lambda) + \alpha N(x_i;\theta))$, use $N(x;\tilde{\theta}_{s,m})$ found in line 3 for initialization.
5. $f_{s,m}(x;\Lambda) = (1-\alpha_{s,m}) f_{s,m-1}(x;\Lambda) + \alpha_{s,m} N(x;\theta_{s,m})$.
6. Update $f_{s,m}$ using EM [optional].
7. End For
8. Set $m = m+1$.
9. If a stopping criterion is met then exit, else go to line 2.

In line 4, as a modification of line 5 in Algorithm 1, the parameters $\alpha_{s,m}$ and $\theta_{s,m}$ are jointly optimized, which is an inherent property of EM algorithms. In this case the new component found in line 3 is used for initialization. Also note that the re-estimation step in line 6 is not in Algorithm 1. The step is added because in GMD estimation, it is often desirable to tune the model parameters after a structural change caused by insertion of a new component.

There is no closed-form solution for the optimization in line 4. However, it can be viewed as a sequential learning of two component models, with the component $f_{m-1}(x;\Lambda)$ fixed. A partial EM algorithm was proposed in [47] for ML estimation of GMDs, which can be easily extended to the ML estimation of HMMs. The update equations for the m^{th} component of GMD at state s are given as following:

$$p(s,m \mid x_i) = p(s \mid x_i)\frac{\alpha_{s,m} N(x_i \mid \mu_{s,m}, \Sigma_{s,m})}{(1-\alpha_{s,m}) f_{s,m-1}(x_i) + \alpha_{s,m} N(x_i \mid \mu_{s,m}, \Sigma_{s,m})} \tag{4.3}$$

$$\hat{\alpha}_{s,m} = \frac{\sum_{i=1}^{N} p(s,m \mid x_i)}{\sum_{i=1}^{N} p(s \mid x_i)} \tag{4.4}$$

$$\hat{\mu}_{s,m} = \frac{\sum_{i=1}^{N} p(s,m \mid x_i) x_i}{\sum_{i=1}^{N} p(s,m \mid x_i)} \tag{4.5}$$

$$\hat{\Sigma}_{s,m} = \frac{\sum_{i=1}^{N} p(s,m \mid x_i)(x_i - \mu_{s,m})(x_i - \mu_{s,m})^T}{\sum_{i=1}^{N} p(s,m \mid x_i)} \tag{4.6}$$

Normally, a global search as required in line 3 is computationally prohibitive. Since only one component needs to be re-estimated at each iteration, partial EM requires much less computation than full EM. The computational efficiency demonstrated by partial EM is critical in developing a global searching heuristic [46][47].

4.3 Model Complexity Selection

Both ML and gradient boosting have over-fitting problems and it is of high interest to automatically select a model and the number of mixture components. In statistics literature, most often used model selection criteria are cross validation (CV) and Bayesian Information Criterion (BIC). BIC is derived within the Bayesian statistics framework, and is preferred in a density estimation perspective. Denote the model *M* and parameter set θ, let $\pi(\theta \mid M)$ be the prior of θ given *M*, a classical way is to choose the model which maximizes the integrated likelihood,

$$\hat{M} = \arg\max_{M} f(X \mid M) = \arg\max_{M} \int f(X \mid \theta, M)\pi(\theta \mid M)d\theta \qquad (4.7)$$

with $f(X \mid \theta, M) = \prod_{i=1}^{n} f(x_i \mid \theta, M)$. Under regularity conditions, an asymptotic approximation of the integrated likelihood can be shown to be [48]

$$\log f(X \mid M) \approx \log f\left(X \mid \hat{\theta}, M\right) - \frac{\nu_M}{2}\log(n) \qquad (4.8)$$

where $\hat{\theta}$ is the ML estimator of θ, ν_M is the number of free parameters in model *M*. It leads to minimize the so-called BIC criterion

$$BIC_M = -2L_M + v_M \log(n) \tag{4.9}$$

where $L_M = \log f\left(X \mid \hat{\theta}, M\right)$ is the maximum log likelihood.

4.4 Approximate Gradient Boosting for HMM

In gradient boosting, searching for the new component requires evaluating the candidates using entire set of observation data, resulting in very high computation cost and memory requirement. Approximation is made based on following observations:

- Gradient boosting starts from the coarsest model, i.e., the single Gaussian model, and introduces finer models sequentially. Therefore, it is reasonable to reduce the range of training data for evaluating a finer candidate, as in the case of sparse EM [49].

- As shown in Figure 4.1, placing Gaussian components along class boundary may result in better coverage of data for classification tasks. To enhance this behavior, the influence on the class boundary from data in the center regions needs to be reduced.

Above observations indicate that it is desirable to evaluate the candidates in a localized neighborhood in order to save computation cost and improve model quality. To reduce computation complexity, sufficient statistics are accumulated in each HMM state by Viterbi approximation. We further assume that state transition probabilities remain unchanged during gradient boosting, then the approximated procedure can be summarized as the following three steps:

1) Train single Gaussian HMMs and segment training data to states of phone units by Viterbi segmentation.

2) Use gradient boosting to train Gaussian mixture model for individual state.

3) Re-estimate HMMs by conventional EM.

For each phone state, in order to generate candidates for the *(m+1)th* component, the training data set is quantized into *m* disjoint sets: $Q_i = \left\{ x \in X : i = \arg\max_{j=1,\ldots,m} P(j \mid x) \right\}$.

Then for each set Q_i, a pair of candidates is generated by randomly splitting Q_i into two disjoint subsets. The data sample means and variances in these two sets are chosen as candidate parameters, and the initial weight for each candidate component is set to be the half weight of $N(\bullet \mid \theta_i)$. If more candidates are needed from this component, then the random splitting process is carried out repeatedly to obtain the required number of candidates. Assuming *k* candidates are generated from each existing component, then *km* candidates are generated for the new component. Each candidate is re-estimated by using the partial EM. The candidates are first validated by their shapes (eigenvalues) and volumes (determinants) with pre-defined thresholds. Among surviving candidates, the one that gives the greatest likelihood increment when mixed into the existing mixture becomes the new member of the model.

Candidates are evaluated locally by a sparse partial EM. If a candidate is generated from the component Q_i, then it is evaluated only by data of Q_i. Specifically, the sparse algorithm approximates the likelihood of data *x* as $p'(x) = Cp(x)$, $x \in Q_i$; $p(x) = 0$, otherwise, where *C* is a normalizing constant taken as 1. Based on this approximation, the updating formulas for partial EM are put in the forms of (4.10-13). This approximation greatly reduces computation cost, and enables local measurement of each candidate on its capacity of modeling local pattern.

$$p(m \mid x_t) = \frac{\alpha N(x_t \mid \mu, \Sigma)}{(1-\alpha) f_{m-1}(x_t) + \alpha N(x_t \mid \mu, \Sigma)} \tag{4.10}$$

$$\hat{\alpha} = \frac{1}{T_j} \sum_{x_t \in Q_i} p(m \mid x_t) \tag{4.11}$$

$$\hat{\mu} = \frac{\sum_{x_t \in Q_i} p(m \mid x_t) x_t}{\sum_{x_t \in Q_i} p(m \mid x_t)} \tag{4.12}$$

$$\hat{\Sigma} = \frac{\sum_{x_t \in Q_i} p(m \mid x_t)(x_t - \mu)(x_t - \mu)^T}{\sum_{x_t \in Q_i} p(m \mid x_t)} \tag{4.13}$$

GEM requires more computation than EM. Denote T_j as the size of the data set segmented to phone sate *j*. Assuming *k* candidates are generated from each existing mixture component, then cost for component search is $O(kT_j)$. Adding the cost for EM update of f_i, which is $O(iT_j)$, the sum is $O((k+i)T_j)$. The run time of training a sequence of *1* to *m* mixture models in the phone state is then $O\left(\sum_{i=1}^{m} T_j(k+i)\right) \propto O(m^2 T_j)$ if $k << m$. In total, the running time of training will be $O\left(\sum_j T_j m^2\right) \propto O(m^2 T)$, where the complete training data size $T = \sum_j T_j$, which is a factor of *m* times slower than conventional EM.

4.5 Toward Large Margin HMM

A closer look at the misclassification measure in MCE reveals that it can be considered as a generalized log likelihood ratio (GLLR), which represents a

distance between the target model and its competing models [50]. Following this observation, we define the margin as

$$\zeta_r(Y_r) = \log \frac{p_\lambda(Y_r \mid w_r)p(w_r)}{\left[\frac{1}{|\{w \neq w_r\}|}\sum_{w \neq w_r}(p_\lambda(Y_r \mid w)p(w))^\eta\right]^{1/\eta}} = -d_r(Y_r), \quad \eta \geq 1 \qquad (4.14)$$

where $|\{w \neq w_r\}|$ is the number of competing hypothesis, $d_r(Y_r)$ is the misclassification measure.

To obtain a large margin speech recognizer, we define some convex loss function $L: R \to R$, and seek a set of HMMs that minimize the L-risk, $R_L = E(L(\zeta(Y)))$ [51]. The resulting HMMs are called large margin HMMs. Boosting has been shown to be a successful method for solving large margin problems by constructing ensembles of classifiers. However, boosting large margin HMMs appears to be a new topic in speech recognition. Therefore, the materials presented here are without supporting experiments and only for discussion purposes.

Several cost functions have been studied in literature. For instance, AdaBoost uses the exponential loss $L(\alpha) = \exp(-\alpha)$ [52], and LogitBoost takes the logistic loss $L(\alpha) = \ln(1+\exp(-2\alpha))$ as the cost function [53]. Considering both forms of loss, boosting for large margin HMMs attempts to minimize:

$$L_{\exp}(\zeta) = \sum_{r=1}^{R} \exp(-\gamma\zeta_r(Y_r)) \quad \gamma > 0 \quad \text{for exponential loss, and} \qquad (4.15)$$

$$L_{\log it}(\zeta) = \sum_{r=1}^{R} \ln(1+\exp(-\gamma\zeta_r(Y_r))) \quad \gamma > 0 \quad \text{for logistic loss.} \qquad (4.16)$$

Comparing (4.15) and (4.16) with the MCE objective function (2.41)

$$L_{MCE}(\lambda)=\sum_{r=1}^{R}\frac{1}{1+\exp(-\gamma d_r(Y_r))} \quad \gamma>0 \ ,$$

it can be seen that they all attempt to maximize the separation between models but in different functional forms. In a boosting view, this will resulting in different weighting schemes on the samples in search for the additive base models. Note that there are other related ensemble learning approaches such as in [54].

Chapter 5

EVALUATION OF GRADIENT BOOSTING

5.1 Experimental Setup

The approximate gradient boosting algorithm was evaluated on the WSJ 20K Nov 92 task. The standard training data set (WSJ0+WSJ1) including speech of 384 speakers were used. Speech feature analysis was made at a 10msec frame rate with a 25msec window-size. Speech feature components included 13 MFCCs and their first and second derivatives. Cepstral means were removed for every utterance. The baseline acoustic model was trained using HTK with a fixed number of Gaussians in each mixture.

The acoustic models were trained as the following. First, single Gaussian models were trained using conventional EM and were tied by phonetic decision tree with HTK. Second, a Viterbi forced alignment using the trained single Gaussian models was performed to segment training data into phone states. Third, Gaussian mixture models were trained for each tied state using segmented data by gradient boosting, where the maximum allowed number of Gaussians for each phone state was 32. As the last step, an ordinary embedded EM was applied to all the boosted models by using the entire set of training data.

For the WSJ task, standard trigram language model provided by LDC was used, including 19,982 unigrams, 3,518,595 bigrams, and 3,153,527 trigrams. Only within-word triphone acoustic model was tested. One-pass time-synchronous beam search was used for decoding speech with conservative pruning thresholds optimized for testing. The test set used was the WSJ si_et_20 evaluation set which consists of 333 sentences.

5.2 Comparing Gradient Boosting with EM

Word accuracy achieved under the same number of mixture components per mixture density was compared for baseline and gradient boosted models. A fixed number of Gaussian components per state were obtained by the standard splitting procedure during baseline training. For gradient boosting, the recursive procedure is controlled by a stopping criterion. In this experiment, the insertion of new component is terminated if no candidate can be found to improve the log likelihood by a pre-defined threshold. Therefore, the number of Gaussians in each mixture density varies based on available observations and true distribution of data. As a consequence, the model complexity can only be described by an average number over all states. Experimental results are listed in Table 6.1. The last row of the table gives the relative rate of error reduction (RER).

Mix. size	8	10	12	15	16	17
Baseline	88.37	88.66	88.59	88.84	89.31	89.33
GB	88.96	88.92	89.14	89.54	89.86	89.77
RER	5.1%	2.3%	4.8%	6.3%	5.1%	4.1%

Table 5.1. Word accuracy of conventional EM and GB

Over the range of studied model complexity, gradient boosting performs consistently better than EM, resulting in an average error reduction rate of 4.6%. For many states, gradient boosting produced models with smaller number of Gaussian components than EM. For example, when average model complexity is 16 Gaussians per state, nearly 50% GMDs have less than 16 Gaussians. This result confirms the fact that different phonetic units need models with different complexities under typical speech model training conditions. Figure 5.1 shows a histogram of number of Gaussians in each state.

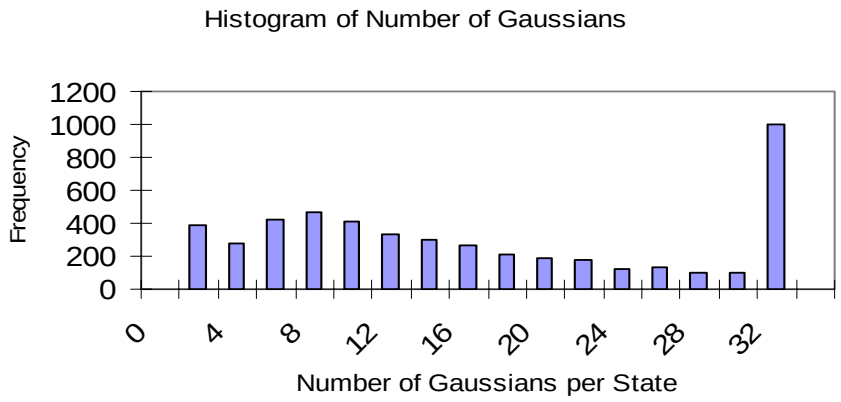

Figure 5.1. Histogram of number of Gaussians per state

In Figure 5.1, we can see that there were about 20% GMDs containing 32 components. An analysis shows that these states have large training data samples and the ML criterion is likely to result in overfitting. Therefore, BIC model selection is used for further investigation.

5.3 Comparing BIC selected Models

To perform BIC model selection, baseline models are grown to have maximum 32 Gaussian components in each state, which is comparable to the number used in gradient boosting. After BIC model complexity selection, the average numbers of Gaussians per state for GB and baseline models are reduced to 15 and 13, respectively. The word accuracy results are given in Table 5.2

	Mix. Size	without BIC	with BIC
Baseline	13	88.84	89.10
GB	15	89.54	89.74

Table 5.2. Comparison of BIC results of GB and EM models

Comparing tables 5.1 and 5.2, we can see that although BIC model selection can greatly reduce the complexity of the baseline models, reducing the number of Gaussians per state from 32 to 13 on average, it can not produce a model comparative to any one of the GB models. For instance, the GB model of same complexity level (average mixture size 13) still performed better than the BIC selected baseline model, although not by a significant margin. The reason is because of the structural difference as depicted in Figure 4.1. To show the effect of this structural difference on model selection, we plot histograms of number of Gaussians per state for the BIC selected models in Figures 5.2 and 5.3.

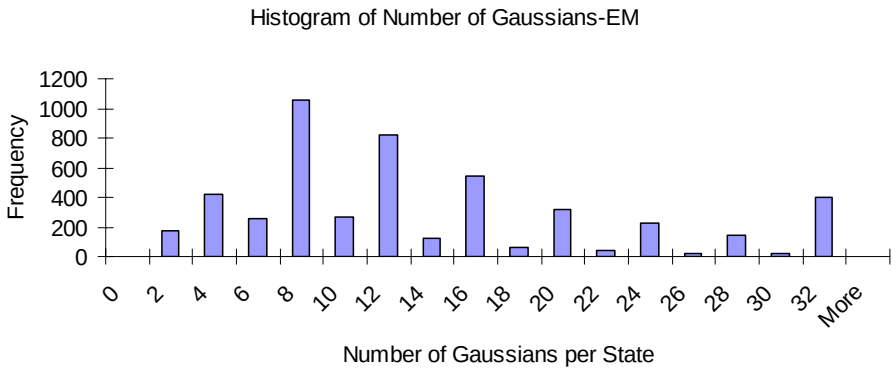

Figure 5.2 Histogram for BIC selected Baseline Models

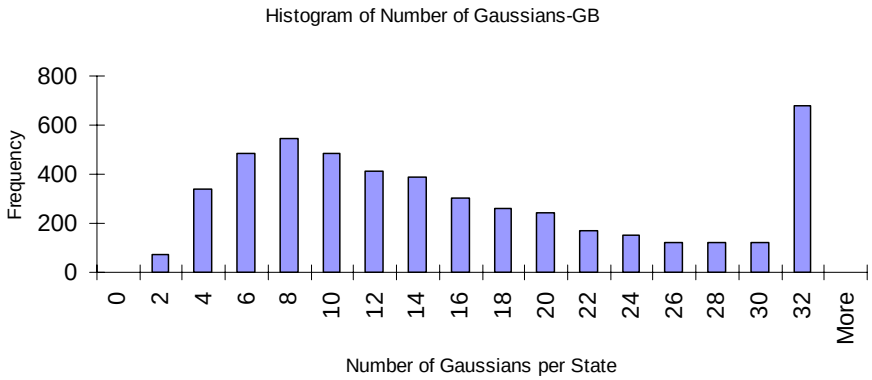

Figure 5.3 Histogram for BIC selected Gradient Boosted Models

Figure 5.3 shows a form of smooth distribution but Figure 5.2 does not. The pattern in Figure 5.2 reveals that a lot of models have been trapped at local optima and the standard splitting method by a small perturbation on existing components can not effectively move the model toward global optimum. This implies that by putting more components on the margin, gradient boosting embraces the power to avoid local optima and produce a fine-tuned model structure.

Chapter 6

KNOWLEDGE BASED ADAPTIVE PDT MODELING

6.1 Introduction

Pronunciation variation in conversational speech has been shown to pose a great challenge to current-generation automatic speech recognition (ASR) systems. Pronunciation variations can be influenced by various levels of factors, including linguistic features of a word (such as morphology, part of speech, tense, etc.), the syllabic and lexical structure, the presence of disfluencies or geographical dialects, and these variations are typically modeled by a combination of a lexical pronunciation dictionary and context dependent acoustic models [55, 39, 56]. In general, pronunciation modeling methods can be categorized into explicit and implicit approaches [57]. Explicit methods model pronunciation variations at a symbolic level by using multiple pronunciations per word or by tree-shaped pronunciation network in word pronunciation dictionaries. However, introducing multiple pronunciations per word may add in confusability in Viterbi decoding of speech, and in practice only small performance gains were observed [58]. On the other hand, pronunciation variation can be implicitly captured in HMM-based acoustic modeling process by utilizing the power of Gaussian mixture densities. In implicit modeling, a soft or hierarchical parameter tying scheme is used to represent the mapping between a phoneme sequence and its acoustic realization as HMMs. Works presented in [57] and [59] demonstrate that implicit methods can perform equally or better than explicit methods when evaluated on the Switchboard corpus. Both works suggest that acoustic modeling in spontaneous, conversational speech can be improved by robust mappings between context-dependent phonemes and HMM states, which is traditionally performed by phonetic decision tree (PDT) state tying.

Recently, many efforts have been made to improve PDT state tying based acoustic modeling for continuous speech recognition [60, 61, 62]. Tree-structured adaptation methods were also reported, which attempt to apply hierarchically organized priors in building more accurate acoustic models from speaker adaptation [32, 63]. Researchers tackled the tree construction problem from different perspectives, which can be roughly grouped into two categories, namely the knowledge-based and data-driven approaches. The knowledge-based method refers to phonetic decision tree state tying which uses phonetic decision rules for clustering of HMMs, and the data-driven method employs an agglomerative clustering procedure based on a distance measure between Gaussian densities. An earlier work in [44] has shown that the two approaches have similar performances while the knowledge-based method has the advantage of allowing model construction for unseen triphones. Another limitation of the data-driven method is its lack of robustness in dealing with mismatches between acoustic feature spaces caused by pronunciation variation when being applied to speaker adaptation in conversational speech.

It is our belief that knowledge-based modeling can be generalized better in large pronunciation variation situations. However, knowledge-based approach could possibly suffer from mismatches between the information source and the specific domain for which it has to be applied, if without adaptive learning [64]. Our hypothesis is that there are systematic relationship between phonological variation and acoustic realizations which can be extracted by a dynamic PDT process growing on the relatively larger data source. Such information can be in turn used selectively for generating domain-specific or speaker-dependent acoustic models.

The common framework of tree growing methods is recursive partitioning of the input space by using a one-step lookahead strategy. Research efforts on

improving phonetic decision tree modeling have been focused on tree growing strategy [60], model structure selection with information criterion [61], and enriching the set of splitting questions [60, 62]. However, without using appropriate prior knowledge on the favored decision tree structure, uncertainty remains in the resulting phonetic decision trees. This problem is acute when speaker adaptation is carried out based on an unreliable tree structure. To the best knowledge of the authors, adaptive learning of phonetic decision tree structures has not yet been shown in previous literatures.

In this work, we present a novel acoustic modeling approach using knowledge-based adaptive decision tree clustering. The prior knowledge on phonological rules is implicitly represented by a tree-generating process on a large corpus, which is used to select good candidate splitting variables for construction of target PDTs in a specific domain. In contrast to traditional methods which find an optimal tree cut in a single large tree (a single realization of prior tree), the proposed method employs prior knowledge on decision rules in greedy search for domain-specific PDTs, and thus results in transformed tree topology. The contributions of this paper are

--A general Bayesian learning framework for PDTs which incorporates prior knowledge on tree structure. The probability distribution of a decision tree is decomposed into probabilities on *tree structure*, which contains the *tree topology* and the *tests* carried out at internal nodes, and the *observation distributions* at leaf nodes. By making appropriate simplification, our tree priors mainly compose of prior probabilities of splitting variables at internal nodes.

--A Bayesian tree information criterion (BTIC) which is used as splitting rule. Assuming informative priors on favored tree structure, BTIC is derived as an extension to the well-known Bayesian information criterion (BIC).

--A hierarchical prior of splitting questions implicitly represented by a decision tree growing process based on a large corpus. In general, considering

the number of possible realizations of a decision tree, the computation of priors on tree structure would be intractable without specifying a computational efficient algorithm. We propose a novel solution to this problem by introducing an identical process of the targeting PDT on the large corpus, providing recursive estimation of prior probability of splitting variables.

6.2 Background on Bayesian Decision Tree

6.2.1. Statistical Decision Tree Modeling

The theory and algorithms on Bayesian learning of decision trees were first studied in [65], where probability distribution of a decision tree was decomposed into probabilities of a tree structure, which contains the tree topology and the tests at each splitting node, and the observation distribution densities at each leaf node. Subsequently, effective Bayesian stochastic search algorithms using Markov Chain Monte Carlo (MCMC) simulation were developed for Bayesian inference of trees [66, 67]. In introducing the framework of Bayesian decision tree, we will follow the notations as used in [67].

A binary decision tree is uniquely identified by a set of variables $T = \left(s_i^{pos}, s_i^{\text{var}}, s_i^{rule}\right), i = 1,...,k-1$, where s_i^{pos}, s_i^{var} and s_i^{rule} denote the position, variable and the point where the variable is split for each splitting node, *i*, and thus, *k* represents the number of terminal nodes. A parameter set associated with *k* terminal nodes is further defined as $\Theta = \left(\theta_1,...,\theta_k\right)$, where θ_j is the parameter of the observation distribution density at the j^{th} terminal node. A training data set is defined as $(Y, X) = \{y_t, x_t\}$, $t = 1,...,n$, where $y = \left(y_1,..., y_d\right)^T$ is the *d-dimensional* observation variable and $x = \left(x_1,..., x_p\right)^T$ is the *p-dimensional* splitting variable. Assuming conditioned on (Θ, T), the observations across

terminal nodes are independent, and those within terminal nodes are i.i.d.. The joint distribution is of the form

$$p(Y \mid X, \Theta, T) = \prod_{i=1}^{k} \prod_{j=1}^{n_i} p\left(y_{ij} \mid \theta_i\right) \tag{6.1}$$

where $Y_i = \left\{ y_{ij} \right\}$, $j = 1, ..., n_i$ denote the data points in terminal node *i*. The posterior distribution of *T* is given by

$$\begin{aligned} p(T \mid X, Y) &\propto p(Y \mid X, T) p(T) \\ &= p(T) \int p(Y \mid X, \Theta, T) p(\Theta \mid T) d\Theta \end{aligned} \tag{6.2}$$

up to a normalizing constant. Analytical forms of the integral $p\left(Y \mid X, T\right) = \int p\left(Y \mid X, \Theta, T\right) p\left(\Theta \mid T\right) d\Theta$ can be obtained by using conjugate priors or *Laplace* approximation [66,67,48].

6.2.2. Non-Informative Tree Prior

The prior on tree $T = \left(s_i^{pos}, s_i^{\text{var}}, s_i^{rule}\right), i = 1, ..., k-1$ can be specifies as follows. First, a discrete distribution $p\left(s_i^{\text{var}}\right)$ is defined over the domain $s_i^{\text{var}} \in \left\{1, ..., p\right\}$ which corresponds to index of the p splitting variables in $x = \left(x_1, ..., x_p\right)^T$. Second, a conditional distribution $p\left(s_i^{rule} \mid s_i^{\text{var}}\right)$ is specified with s_i^{rule} taking a total number of $n\left(s_i^{\text{var}}\right)$ possible values for the splitting variable s_i^{var}. Finally, an upper bound of splits allowed in one path down the tree, $S_{\max}$, is set to ensure a finite number of possible trees, i.e., $s_i^{pos} \in \left\{1, ..., 2^{S_{\max}+1} - 1\right\}$.

Usually the distributions $p\left(s_i^{\text{var}}\right)$ and $p\left(s_i^{rule} \mid s_i^{\text{var}}\right)$ are chosen as uniform

distributions. In such a case, the prior distribution for a complete tree structure becomes

$$\begin{aligned} p(T) &= \left\{ \prod_{i=1}^{k-1} p\left(s_i^{rule} \mid s_i^{\text{var}}\right) p\left(s_i^{\text{var}}\right) \right\} p\left(\left\{s_i^{pos}\right\}_1^{k-1}\right) \\ &= \left\{ \prod_{i=1}^{k-1} \frac{1}{n\left(s_i^{\text{var}}\right)} \frac{1}{p} \right\} \frac{k!}{S_k} \frac{1}{K} \end{aligned} \tag{6.3}$$

where S_k is the total number of possible ways of choosing $\left\{s_i^{pos}\right\}_1^{k-1}$ to produce a *k*-terminal node tree, and *K* is the maximum number of terminal nodes. For binary decision trees, S_k is given in *graphics* theory as the *Catalan number*

$$S_k = \frac{1}{k+1}\binom{2k}{k} \tag{6.4}$$

The prior on tree topology $p\left(\left\{s_i^{pos}\right\}_1^{k-1}\right) = \frac{k!}{S_k}\frac{1}{K}$ is a function of the number of terminal nodes *k* and independent of the rule assignment in splitting nodes.

In [66], a recursively defined prior on tree topology which favors "bushy" trees is specified by the node splitting probability

$$p_{SPLIT}(s,T) = \alpha(1+d_s)^{-\beta} \tag{6.5}$$

where d_s is the depth of node *s*, $0 < \alpha < 1, \beta \geq 0$. Although a "bushy" tree is often preferred in practice, we opt to the prior specified in (6.3) for the reason of simplicity. Since our interest is on informative priors of splitting rules and the prior on tree topology will be considered nuisance in our later discussions. This kind of simplification would not affect the justification of our findings to be presented in the following sections.

6.3 Bayesian PDT Learning based on Informative Prior

6.3.1. Informative Prior on Tree Structure

Note that the prior $p(T)$ defined in (6.3) is non-informative. When prior knowledge of favored tree structures is available, it is beneficial to consider more informative priors on tree structures. In phonetic decision tree based state tying, this knowledge is carried by the splitting variables, i.e., linguistic questions being asked at each splitting node. Since the answers to the linguistic questions only take Boolean values (true/false), we have $p\left(s_i^{rule} \mid s_i^{\text{var}}\right)=1$ conditioned on a given splitting variable. Assuming an implicitly defined belief on $p\left(s_i^{\text{var}}\right)$, we use the following form of prior in PDT modeling

$$p(T)=\left\{\prod_{i=1}^{k-1} p\left(s_i^{\text{var}}\right)\right\} p\left(\left\{s_i^{pos}\right\}_1^{k-1}\right) \propto \prod_{i=1}^{k-1} p\left(s_i^{\text{var}}\right) \tag{6.6}$$

where $p\left(\left\{s_i^{pos}\right\}_1^{k-1}\right)$ only depends on tree topology and will be treated as nuisance factor when focus is on splitting rules. The strategy of implicit modeling for $p\left(s_i^{\text{var}}\right)$ will be given in Section IV.

6.3.2. Bayesian Tree Information Criterion

The Bayesian model selection criterion chooses the tree structure which has the highest posterior probability. Substitute (6.11) and (6.6) into (6.2) yields

$$\begin{aligned} p(T \mid X, Y) &\propto p(T) \int p(Y \mid X, \Theta, T)\, p(\Theta \mid T)\, d\Theta \\ &\propto \left\{\prod_{i=1}^{k-1} p\left(s_i^{\text{var}}\right)\right\} \times \int \prod_{i=1}^{k}\left\{\prod_{j=1}^{n_i} p\left(\underset{\sim}{y}_{ij} \mid \underset{\sim}{\theta}_i\right) p\left(\underset{\sim}{\theta}_i \mid T\right)\right\} d\Theta \end{aligned} \tag{6.7}$$

The Bayesian tree information criterion (BTIC) is defined to be the logarithm of the tree posterior probability

$$BTIC(T) = \log p(T \mid X, Y) \tag{6.8}$$

A key problem in evaluating *BTIC* is to compute the evidence of observations, $p(Y \mid X, T)$, given as follows,

$$p(Y \mid X, T) = \int \prod_{i=1}^{k} \left\{ \prod_{j=1}^{n_i} p(y_{ij} \mid \theta_i) \, p(\theta_i \mid T) \right\} d\Theta \tag{6.9}$$

The integral over parameter space Θ is often intractable when considering complex model architectures. In PDT literature, two kinds of approaches were commonly employed to tackle this problem, referred to as exact methods and approximate method, respectively. Exact methods make assumption on the parametric forms of observation distributions and prior of parameters at leaf nodes. For multivariate normal observation distributions at leaf nodes, i.e.,

$$p(y_{ij} \mid \theta_i) = N_p(y_{ij} \mid m_i, R_i) \tag{6.10}$$

where $N_p(y_{ij} \mid m_i, R_i)$ is *p-dimensional multivariate normal distribution* with mean m_i and precision matrix R_i, exact methods use the normal-Wishart conjugate prior as follows [61][65],

$$\begin{aligned} &p(m_i, R_i \mid \tau_i, \mu_i, \alpha_i, \Psi_i) \propto |R_i|^{(\alpha_i - p)/2} \\ &\times \exp\left\{ -\frac{\tau_i}{2} (m_i - \mu_i)^T R_i (m_i - \mu_i) \right\} \exp\left\{ -\frac{1}{2} tr(\Psi_i R_i) \right\} \end{aligned} \tag{6.11}$$

where $\tau_i, \alpha_i, \mu_i, \Psi_i$ are hyper-parameters. Analytical results show that the

evidence $p(Y \mid X,T)$ is in the form of *p-dimensional multivariate student t distribution*

$$p(Y_i \mid X,T) \propto \left(\frac{\tau_i}{\tau_i + n_i} \right)^{1/2} \left| \Psi_i \right|^{-(\alpha_i + n_i)/2} \times \left[1 + s_i + t_i \right]^{-(\alpha_i + n_i)/2} \tag{6.12}$$

where $s_i = \sum_{t=1}^{n_i} (y_{it} - \bar{y}_i)^T \Psi_i^{-1} (y_{it} - \bar{y}_i)$, and $t_i = \frac{\tau_i n_i}{\tau_i + n_i} (\bar{y}_i - \mu_i)^T \Psi_i^{-1} (\bar{y}_i - \mu_i)$.

The *Laplace* approximation method for exponential family as described in [48] has been extensively used in literature to evaluate the integral in (6.9). Assuming that the function $p(Y_i \mid \theta_i)\, p(\theta_i \mid T)$ is strongly peaked at the ML estimate $\hat{\theta}_i$, i.e., $p(Y_i \mid \theta_i)\, p(\theta_i \mid T)$ is dominated by the term $p(Y_i \mid \theta_i)$, a second order Taylor expansion of the logarithm of this function around $\hat{\theta}_i$ leads to a tractable form

$$\begin{aligned} &\log \int p(Y_i \mid \theta_i)\, p(\theta_i \mid T)\, d\theta_i \approx \log p\left(Y_i \mid \hat{\theta}_i\right) \\ &+ \log p\left(\hat{\theta}_i \mid T\right) + \frac{p}{2} \log(2\pi) - \frac{p}{2} \log n_i - \frac{1}{2} \log \left| I_y(\theta_i) \right| \\ &\overset{n>>0}{\approx} \log p\left(Y_i \mid \hat{\theta}_i\right) - \frac{p}{2} \log n_i = BIC \end{aligned} \tag{6.13}$$

where *p* is the number of free parameters in the model and $I_y(\theta_i)$ is the Fisher information matrix. The resulting value is equivalent to the well known Bayesian information criterion (BIC), also known as Schwarz information criterion (SIC) [48].

Choosing between the exact and approximate methods can be considered as application dependent. One advantage of exact method is that it performs both roles of model selection and adaptation, since the MAP estimates of model parameters can be easily obtained by the expectation-maximization (EM)

algorithm [61]. But it also has limitations by requiring appropriate assumptions on probability distributions of data and specification of hyper-parameters. In PDT modeling for speech recognition, a choice between the two methods also implies a choice between access to data or sufficient statistics. Since BIC only needs to evaluate observations in the log-likelihood function, sufficient statistics from the *Baum-Welch* algorithm can be directly used to compute the BIC score [44]. In current study, we adopt the approximate method provided with its computation convenience. After standard analytical simplification, the Bayesian tree information criterion as defined in (6.8) can be found to be

$$BTIC(T) = BIC(T) + \gamma \sum_{i=1}^{k-1} \log p\left(s_i^{\text{var}}\right) \tag{6.14}$$

where γ is a regularizing parameter.

6.3.3. Relationship to Other Model Selection Criterion

From previous discussion we can see that the proposed Bayesian tree information criterion is an extension to the traditional Bayesian information criterion by introducing informative prior on tree structure. In general, considering prior model probability $p(M_m)$ with $M = \{M_m\}$, BIC is defined to be

$$\begin{aligned} BIC &= \log p(Y, M_m) \\ &= \log\left(p(M_m) \int p(Y \mid \theta)\, p(\theta \mid M_m)\, d\theta \right) \end{aligned} \tag{6.15}$$

However, the priors $p(M_m)$ and $p(\theta \mid M_m)$ are often taken as non-informative because of the lack of prior knowledge on distributions of parameters and model structure. From this point of view, equation (6.14) is a misuse of the

term BIC, since conceptually, BIC already considers the model prior which was later ignored as a small term during numerical approximation. Therefore, equation (6.14) is only considered numerically correct.

The predictive information criterion (PIC) differs from BIC in two ways [61]. First, PIC leaves out the term $p(M_m)$ in definition, which has been shown important in selecting appropriate decision trees in our study, and second, PIC uses exact calculation of the integral, as illustrated in equations (6.10), (6.11) and (6.12). By definition, PIC is given as

$$\begin{aligned} PIC &= \log p(Y \mid M_m) = \log \int p(Y \mid \theta)\, p(\theta \mid M_m)\, d\theta \\ &= BIC(M_m) - \log p(M_m) \end{aligned} \tag{6.16}$$

Therefore, BTIC can be specified in terms of PIC as

$$BTIC(T) = PIC(T) + \gamma \sum_{i=1}^{k+1} \log p\left(s_i^{\text{var}}\right) \tag{6.17}$$

following the definition in (6.8), when exact integration is used assuming proper conjugate priors.

6.4 Knowledge-Based Adaptive Decision Tree Clustering

Recently, much attention has been drawn to employing knowledge-based features for speech recognition, ranging from directly using linguistically derived features, so called “distinctive features” [34], in the recognition process, to implicitly integrating high-level contexts, such as syllable and stress, into decision tree based parameter sharing [58, 32]. Rational behind these methods is that incorporating human understanding of acoustics-phonetics about speech variation will provide

more accurate and consistent modeling of speech. The performance of these systems depends on the goodness of the knowledge used and the effectiveness of information handling in the system. In PDT modeling, this implies that two additional functions are needed, one is for coding the phoneme sequence in an utterance with distinctive features using an external knowledge source of language, and the other one is for interpreting the knowledge based on a relatively large corpus and providing a goodness measure of distinctive features when to be used in construction of PDTs. Following the discussions in Section 3.4, a PDT modeling system scheme which incorporates these two parts is named knowledge-based adaptive PDT clustering, and is depicted in Figure 6.1, where sub-word unit *v* is often taken as lexical phoneme, or sub-phoneme when left-right HMM is used to represent a single phone, and distinctive feature *x* consists of various factors which influence pronunciation variation (such as surrounding phones, stress and syllable structure, part-of-speech, etc.).

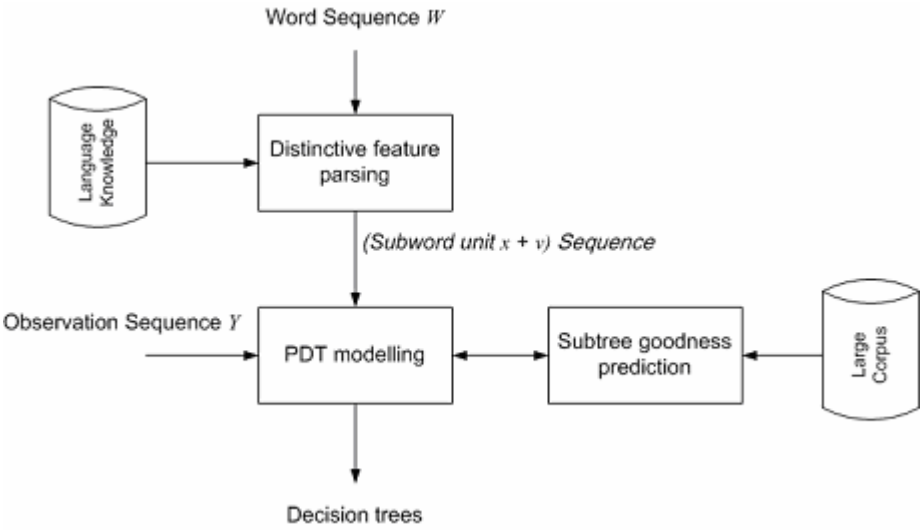

Figure 6.1 Diagram of knowledge-based adaptive PDT

The key part of this system is the "*subtree goodness prediction*," which is based on

the *BTIC* model selection where the subtree goodness is measured by the prior probability $p\left(s_i^{\text{var}}\right)$ of splitting variables estimated from a large corpus. However, considering the huge number of possible realizations of a decision tree, the estimation for $p\left(s_i^{\text{var}}\right)$ would be intractable if no computational efficient algorithm is specified [66, 67]. In an adaptive learning setting, we propose a novel solution to this problem by recursively defining $p\left(s_i^{\text{var}}\right)$ based on the belief represented by a dynamic decision tree growing process on a large data set, as follows

$$p\left(s_i^{\text{var}}\right) \propto \begin{cases} \Delta BTIC, & \text{if } s_i^{\text{var}} \in \text{ top } h \text{ variables} \\ 0, & \text{otherwise} \end{cases} \tag{6.18}$$

where

$$\Delta BTIC = \left(BTIC(s_{i\_L}) + BTIC(s_{i\_R}) \right) - BTIC(s_i) \tag{6.19}$$

is the information gain by splitting the node s_i to its left and right children nodes s_{i_L} and s_{i_R}. This probability is defined positive only for the top h number of hypothesis on splitting variables (distinctive features) which give the best improvement in *BTIC*, and its value is proportional to the corresponding information gain with the stochastic constraint that sum of the probabilities equal to one. Forcing the probabilities of ineffective splitting variables to zero is for reducing noise and uncertainty in the tree learning process.

As discussed above, *BTIC* model selection is performed by two interleaved tree growing processes, as shown in Figure 6.2. The primary tree process is the domain-specific PDT which we are searching for, and hence called a target tree. The secondary tree process provides beliefs on splitting variables to the primary

tree, and is therefore called an oracle tree. The splitting of oracle tree is governed by the targeting tree and is in fact an identical tree copy of the targeting tree but growing in a different observation space. Note that each tree can use its own representation of acoustic features, for instance, MFCCs can be used in one tree while PLPs being used in the other tree. When both trees use matched form of acoustic features, the oracle tree also generates prior information on observation distribution parameters. Equation (6.17) provides an option to integrate this information into the targeting tree process using the exact form in (6.12).

Both trees are constructed based on the splitting method as in [12], except that the splitting rule used is *BTIC*. Recall that we use the approximated *BTIC* given by

$$BTIC(T) \approx \log L(T) - \rho \frac{p}{2} \sum_{i=1}^{k} \log n_i + \gamma \sum_{i=1}^{k-1} \log p\left(s_i^{\text{var}}\right) \tag{6.20}$$

where $L(T)$ is the likelihood of the tree, ρ and γ are adjustable regularizing factors, and the sample count at leaf node *i*, n_i, is approximated by accumulated state occupancies which were estimated from the Baum-Welch algorithm. In splitting the oracle tree, $p\left(s_i^{\text{var}}\right)$ is assumed non-informative, i.e., uniformly distributed.

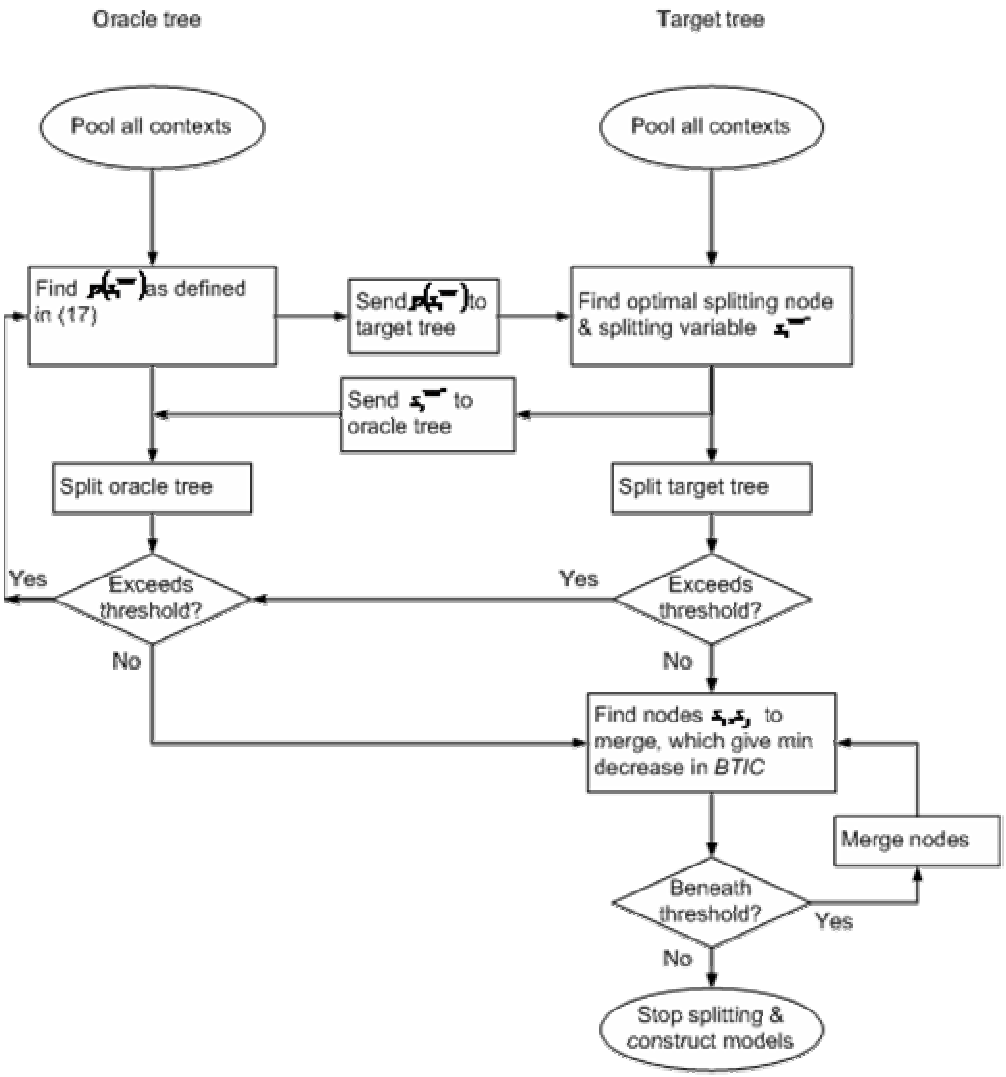

Figure 6.2. The BTIC based decision tree construction scheme

Chapter 7

EVALUATION OF KNOWLEDGE BASED ADAPTIVE DECISION TREE

7.1 Experimental Setup

The knowledge-based adaptive PDT algorithm was evaluated on the telemedicine automatic captioning task developed in the Spoken Language and Information Processing Laboratory at the University of Missouri-Columbia. The objective of this project is to develop an online captioning system to help hearing impaired users in telemedicine interviews. Developing such a system in telemedicine domain is challenging in several ways. First, telemedicine conversations are spontaneous and contain various amounts of filled-pauses, repetitions, repairs and noises. Second, relatively sparse training data make it difficult to train a large number of parameters in both acoustic and language modeling. Third, variations in speaking style and fluency call for effective methods to describe the pronunciation patterns of different speakers. The knowledge-based adaptive decision tree clustering algorithm is a powerful tool developed to combine acoustic-phonetic knowledge for resolving these difficulties from a perspective of acoustic modeling. Besides development and evaluation of new algorithms, acoustic modeling efforts also include tedious work on data collection and preprocessing, feature analysis as well as noise and filled pause modeling. For a comprehensive description of this project, interested readers are advised to refer to [68]. A brief layout of the telemedicine automatic captioning system is shown in Figure 7.1.

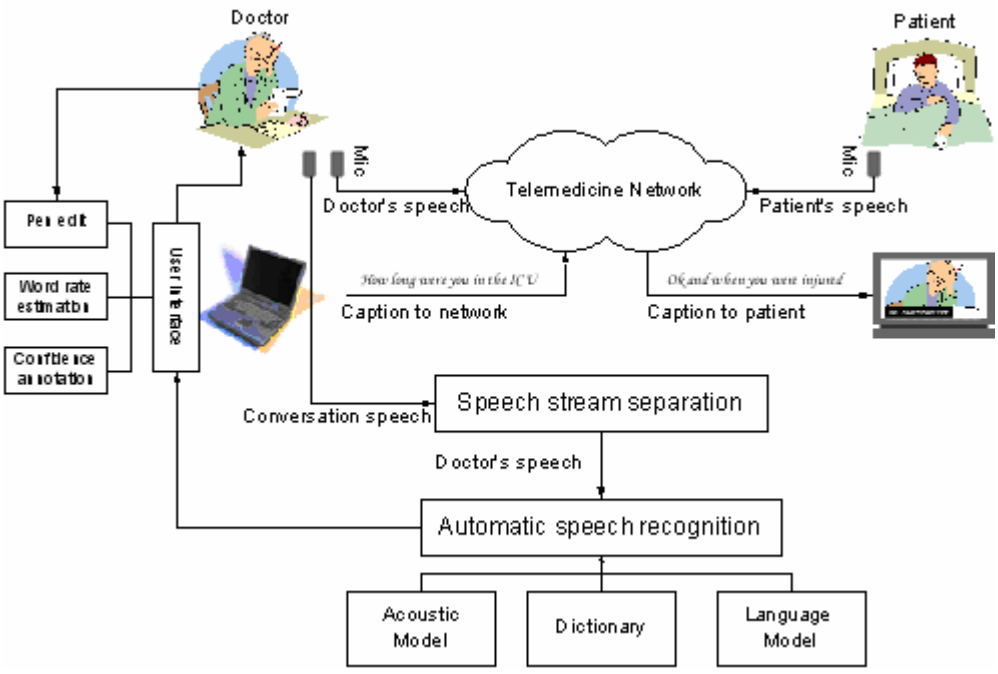

Figure 7.1. Automatic captioning system for telemedicine [68].

7.1.1. Data Collection

Speech data of telemedicine conversations were collected on the sites of the University of Missouri Telemedicine Network. A total of seven medical professionals contributed their voices in mock telemedicine interviews. Speech recordings were taken in sessions, each with one volunteered client and lasts 20~30 minutes long. Conversation topic of each session was determined by the health care provider and ranged from neuropsychology to dermatology, which choice mainly depended on the health care provider's professional background. About 51 hours of conversational data were collected from the health providers' sites, within which about 24 hours of speech were from the seven health care providers. The speech data of health care providers were then manually transcribed into word sentences by experienced personal. The resulting transcriptions consist totally 305,818 words of which 8.02% are medical terms.

A lexicon dictionary was built with special concern on covering medical terms. At lexicon level, pronunciation variations were roughly modeled by multiple pronunciations in dictionary, resulting in a vocabulary size of 46,489, with 3.07% of vocabulary words being medical terms [68].

7.1.2. Preprocessing of Speech Data

The original speech recordings and word transcriptions were too raw to be directly used by the speech recognition system. They need to be cleaned, aligned, and organized into a telemedicine corpus. Currently, speech recording data of five speakers, two female (D1 and D5) and three males (D2, D3 and D4) have been processed. One or two sessions of a speaker's speech were randomly chosen to be set aside for model evaluation purposes. A summary of the telemedicine corpus is given in Table 7.1. The conversation set consist patients' speech, and the training and test data sets which altogether constitute doctor's speech. Word counts from transcription texts are also listed [68].

	Conversation	*Training set*	*Test set*
D1	630	210/35,348	29.8/5,105
D2	480	200/39,398	14.3/2,760
D3	300	145/28,700	19.3/3,238
D4	420	180/39,148	27.8/6,492
D5	380	250/44,967	12.1/3,998

Table 7.1. Data sets of 5 doctors: speech(min)/text(no. of words)

7.1.3. Noise and Filled Pause Modeling

Symbolic forms of filled pauses and noises were first extracted from text transcriptions. Acoustically similar forms were then identified and merged into

one class. Finally, seven filled pause units and one dummy “fp” unit representing noise-like sounds including lip smack and microphone ruffling were defined. Details are given in Table 7.2.

Model unit	Filled pause / noise pattern				
aah	AH				
om	OM	OMM	AHM		
umm	UM	UMM	UHM	UMHM, UMHUM	
hum	HUM				
oh	OH			UHOH	
uhh	UH				UHUH
huh	HUH				
fp	(smack)	(mic. Sound)	MMM	HMM	

Table 7.2. Summary of filled pause and noise model units

7.1.4. Acoustic Modeling

Speaker dependent models were trained for each speaker. Speech features consisted of 39 components including 13 MFCC parameters and their first and second order time derivatives. Feature analysis was made at a 10msec frame rate with a 20msec window size. Adding in the filled pause and noise units, a total of 52 sound units were defined, including 42 speech monophone units as well as silence and short pause units. Within word context dependent triphone modeling was used for speech monophones, while context independent modeling was used for the rest units. Left-right hidden Markov model (HMM) with three states was used for acoustic modeling, where state emission probability was modeled by 16-Gaussian mixture density with diagonal covariance matrix. Baum-Welch estimation of CDGMM-HMM parameters were carried out by using the HTK toolkit [69].

7.2 Evaluation Issues

The proposed knowledge-based adaptive decision tree learning is a general statistical modeling approach, and therefore can be used to fulfill different goals in data modeling tasks, including but not limited to

- Construction of a decision tree with optimal prediction power.
- Inference on the splitting variables that may be used in a rule-based decision system.
- Adaptation of tree-structured models, where traditional adaptation methods such as MAP and MLLR may be involved.
- Model complexity selection based on BTIC.
- Combination with other tree construction techniques, for instance, lookahead algorithms.

In this work, we show the effectiveness of the proposed algorithm for construction of PDTs (KBA-PDTs) using the telemedicine automatic captioning corpus, and the effectiveness is judged based on the speech recognition word accuracy as well as the resulting acoustic model complexity. Possible extensions of this algorithm will be discussed in section 7.4.

7.3 Experimental Results

Speaker dependent triphone acoustic models were trained using the BTIC based decision tree state tying, where the large corpus for oracle tree construction contains pooled speech from all the speakers. Phonetic question set used was the 202-questin set as in [44]. Prior to building the trees, single Gaussian acoustic

models were first estimated for untied triphones and sufficient statistics were accumulated for corresponding oracle and target PDTs. The resulting speaker dependent PDTs were then used to cluster HMM states and to construct unseen triphones. At last, tied single Gaussian models were augmented to 16-Gaussian mixture models by the HTK splitting procedure. Baseline models were also trained using the maximum likelihood criterion (ML-PDTs). The model complexity and word accuracy results are summarized in Table 7.3

		KBA-PDT	ML-PDT
D1	Number of states	1611	2238
	Word accuracy	***81.75***	*81.17*
D2	Number of states	1119	1569
	Word accuracy	***73.73***	*73.15*
D3	Number of states	799	1156
	Word accuracy	***74.98***	*73.95*
D4	Number of states	1027	1521
	Word accuracy	***78.35***	*77.96*
D5	Number of states	1552	1838
	Word accuracy	***83.55***	*82.80*

Table 7.3. Effectiveness of knowledge-based adaptive PDT

7.3.1. Effects of the Number of Active Questions *h*

The value *h* for the prior probability of splitting variables as defined in (6.18) is called a tuning constant; smaller value of *h* means strong belief on the knowledge extracted from the large corpus and more resistant to noise and uncertainty in the domain-specific training data, but at the expense of lower robustness when the large data set is less representative. For example, in our pilot experiments, using

read speech (WSJ) as prior knowledge source for telemedicine conversational speech resulted in degraded recognition performance. The performance of KBA-PDTs versus different values of *h* for five speakers is give in Table 7.4

	h	1	5	10	200
D1	Number of states	1250	1463	1611	1804
	Word accuracy	*80.83*	*80.83*	***81.75***	*81.28*
D2	Number of states	871	1021	1119	1278
	Word accuracy	*73.13*	*73.01*	***73.73***	*73.20*
D3	Number of states	581	727	799	944
	Word accuracy	*74.67*	*74.67*	***74.98***	*74.73*
D4	Number of states	852	1027	1098	1204
	Word accuracy	*77.57*	***78.35***	*78.29*	*78.35*
D5	Number of states	1002	1216	1397	1552
	Word accuracy	*82.95*	*83.40*	*83.02*	***83.55***

Table 7.4. Performance of knowledge-based adaptive PDT (effects of *h*-factor)

From this table we can see that best (D1, D2 and D3) or almost best (D4 and D5) recognition results in word accuracy can be achieved with reduced model complexity by setting a small value of *h*. This implies that there was strong agreement between the oracle tree and target tree, which is not a surprise because the large corpus used was just the pooled speech from all five speakers. The effects of *h* on model complexity and word accuracy are plotted in Figures 7.2 and 7.3. From these plots we can also see that the word accuracy curves for speakers D1, D2 and D3 have similar patterns and all peaked at *h = 10*, but such patterns for speakers D4 and D5 are less obvious. This is because that the five speakers were from two different groups: D1, D2 and D3 were

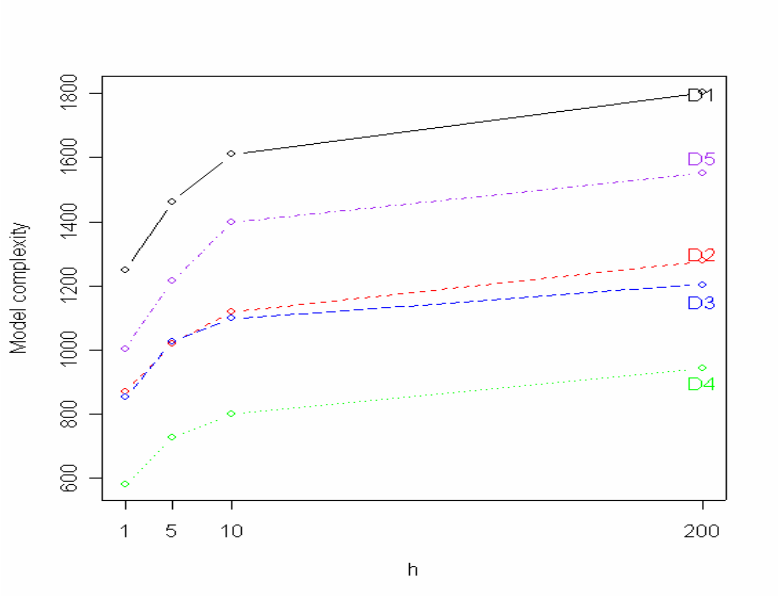

Figure 7.2. Model Complexity (number of states) vs. *h*

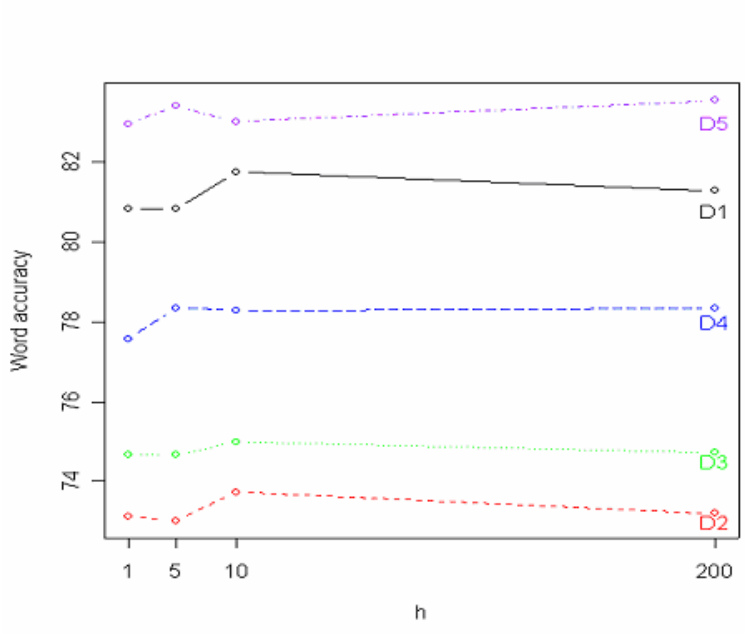

Figure 7.3. Word accuracy (%) vs. *h*

neuropsychologists, while speakers D4 and D5 were dermatologists. This kind of variation is caused by mismatch in speech topics, and is thus called knowledge variation. It is also noticed that the gender difference (which is often a major source of acoustic variation) within the Neuropsychologists' group (D1 is female, D2 and D3 are male) did not cause much disagreement across speakers in the results. The evidences from this experiment support our assumption that the knowledge-based adaptive PDTs can capture the knowledge variation in conversations and remains robust to the acoustic variations.

7.4 Discussions

From the discussions in Chapter 6, it is natural to extend the knowledge-based adaptive decision tree algorithm to decision tree model adaptation. Since the proposed approach is a Bayesian learning framework, MAP estimates of model parameters are easily computable by the expectation-maximization (EM) algorithm. When training data is even sparse, hierarchical linear transformation based adaptation may be used, where more complex analytical forms will be involved. However, given previous works on Bayesian linear transformation based adaptation [62], derivation of such algorithms is still straightforward.

Another direction of research is the inference of the knowledge source, in the form of decision rules. Research interests could be focused on posterior inference of splitting variables, which can be used to investigate knowledge variation from a linguistic-phonological perspective or to facilitate a rule-based decision system.

At last, BTIC can be used as a general model selection criterion for tree based models. In order to effectively apply BTIC based model selection, accurate representation for the prior probability of splitting variables need to be specified. This includes appropriate specification of knowledge source, and computational

efficient probability estimation algorithms. For the latter, existing tree lookahead algorithms might be a feasible choice [53].

Chapter 8

SUMMARY

This work has investigated optimization techniques of acoustic model training for large vocabulary speech recognition in a statistical learning framework. Its main contributions include two innovative machine learning algorithms, tailored for acoustic modeling:

- The knowledge-based adaptive decision tree algorithm was developed, where
    - A general Bayesian learning framework for PDTs was derived to incorporate prior knowledge on tree structure. The probability distribution of a decision tree is decomposed into probabilities on tree structure, which contains the tree topology and the tests carried out at internal nodes, and the observation distributions at leaf nodes. By making appropriate simplification, our tree priors mainly compose of prior probabilities of splitting variables at internal nodes.
    - A Bayesian tree information criterion (BTIC) was introduced as a splitting rule. Assuming informative prior on favored tree structures, BTIC was derived as an extension to the well-known Bayesian information criterion (BIC). The similarities and differences between BTIC and other information criterions including BIC and PIC were described.

- A computational efficient algorithm for prior probability induction was developed. The prior of splitting questions are implicitly represented by a decision tree growing process based on a large corpus. In general, considering the large number of possible realizations of a decision tree, direct computation of priors on tree structure is intractable. We proposed a novel solution to this problem by introducing an oracle process which provides recursive estimation of prior probability of splitting variables.

- The Gradient-Boosting function machine was developed for hidden Markov model with mixture of Gaussian observation densities, where

  - Gaussian mixture densities (GMDs) are recursively constructed in a greedy manner. An optimal new component is located and inserted to the mixture model, offering an efficient mechanism of allocating new components outside the local optimum regions.

  - A partial EM algorithm is developed for global component search based on the maximum likelihood (ML) criterion. By fixing the previously learned model, partial EM can be viewed as learning of two component models, and thus requires much less computation than full EM. This property is the key for developing a computational efficient global search algorithm for allocating optimal new components.

The value of this research is, firstly, it provides the theoretical investigation of applying Bayesian method (knowledge-based adaptive decision tree) and

ensemble learning (Gradient Boosting) to PDT-GMMHMM based acoustic modeling; secondly, the implementations on real data set showed that the proposed methods indeed lead to improved model quality and recognition accuracy in large vocabulary speech recognition tasks; and lastly, there are potential extensions of this work, including but not limited to, the following

- It is natural to extend the knowledge-based adaptive decision tree framework to decision tree model adaptation. Formulation of MAP adaptation has been briefly described in Chapter 6, and derivation of linear transformation based adaptation methods is also straightforward.

- One important part of the knowledge-based adaptive decision tree is the specification for the prior probability of the splitting variables. Future research interests could be focused on either posterior inference of splitting variables that may be used to facilitate a rule-based decision system, or on seeking of more accurate representation for the prior probability of splitting variables. Toward the latter task, one possibility we can think of is to incorporate the tree lookahead approaches which have been described in literature.

- Extension of the Gradient Boosting function learning is towards discriminative training. A novel concept of "large margin HMM" has been introduced in Chapter 4. The rational behind this method is to follow the aggressive weighting scheme on marginal data in existing Boosting procedures. The proposed "large margin HMM" criterion is just a modification of the MCE criterion, and there is a high possibility that this idea will work out.

BIBLIOGRAPHY

[1]. National Center for Health Statistics (NCHS), U.S. Department of Health and Human Services, *Data from the National Health Interview Survey*, Series 10, No. 188, Table 1, B, C,. 1994.

[2]. Y. Zhao, X. Zhang, R-S. Hu, J. Xue, X. Li, L. Che, R. Hu, L. Schopp, "An Automatic Captioning System for Telemedicine," *Proc. ICASSP06*

[3]. L. Deng and D. O'Shaughnessy, *Speech Processing-a Dynamic and Optimization-Oriented Approach*, Marcel Dekker, 2003.

[4]. D. Jurafsky and J.H. Martin, *Speech and Language Processing-an Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, Prentice Hall, 2000.

[5]. D. Povey, *Discriminative Training for Large Vocabulary Speech Recognition*, Ph. D Dissertation, Cambridge University, 2004.

[6]. H. Hermansky, "Perceptual Linear Predictive (PLP) Analysis of Speech," *J. Acoustic Society of America*, 87, pp. 1738-1752, 1990.

[7]. N. Kumar, A.G. Andreou, "Heteroscedastic Discriminant Analysis and Reduced Rank HMMs for Improved Speech Recognition," *Speech Communication*, vol 26, pp 283-297, 1998

[8]. A. Hyvarinen, E. Oja, "Independent Component Analysis: a Tutorial," *http://www.cis.hut.fi/aapo/papers/IJCNN99_tutorialweb/*

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

[10]. Y. Ephraim, N. Merhav, "Hidden Markov Processes", *IEEE Trans. Information Theory*, vol. 48, No. 6, pp. 1518-1569, June 2002.

[11]. S. Roweis, Z. Ghahramani, "A Unifying Review of Linear Gaussian Models", *Neural Computation*, 11, pp. 305-345, 1999.

[12]. Y. Bengio, "Markovian Models for Sequential Data," *Neural Computing Surveys*, 2, pp. 129-162, 1999, http://www.icsi.berkeley.edu/jagota/NCS

[13]. R. J. Elliott, L. Aggoun and J.B. Moore, *Hidden Markov Models: Estimation and Control.* Springer, NewYork, 1995.

[14]. R.W. Zhang and J. C. Hancock, "On Receiver Structures for Channels Having Memory," *IEEE Trans. Inform. Theory*, vol IT-12, pp. 463-468, Oct. 1966.

[15]. L.E. Baum, T. Petrie, G. Soules, and N. Weiss, "A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains," *Ann. Math. Statist.*, vol 41, pp. 164-171, 1970.

[16]. L. E. Baum, "An Inequality and Associated Maximization Technique in Statistical Estimation of Probabilistic Functions of Markov Processes," *Inequalities, III (Proc. 3rd Symp., Univ. Calif., Los Angeles, Calif., 1969; dedicated to the Memory of Theodore S. Motzkin).* New York: Academic, 1972, pp. 1-8.

[17]. A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm", *J. Roy. Statist. Soc. B*, vol. 39, no. 1, pp. 1-38, 1977.
[18]. R. Bellman, *Dynamic Programming.* NJ: Princeton University Press, 1957.
[19]. F. Jelinek, "Continuous Speech Recognition by Statistical Methods", *Proc. IEEE*, vol. 64, pp. 532-556, Apr. 1976.
[20]. L.R. Rabiner, J.G. Wilpon, and B.-H. Juang, "A Segmental *k*-means Training Procedure for Connected Word Recognition," *AT&T Tech. J.*, vol. 65, pp. 21-40, May-June 1986.
[21]. P. Bryant and J. A. Williamson, "Asymptotic Behavior of Classification Maximum Likelihood Estimates," *Biometrika*, vol. 65, no. 2, pp. 273-281, 1978.
[22]. N. Merhav and Y. Ephraim, "Hidden Markov Modeling Using a Dominant State Sequence with Application to Speech Recognition," *Computer, Speech, and Language*, vol. 5, pp. 327-339, Oct. 1991.
[23]. S. Roweis and Z. Ghahramani, "A Unifying Review of Linear Gaussian Models," *Neural Computation*, 11, pp. 305-345, 1999.
[24]. R.H. Shumway and D.S. Stoffer, "An Approach to Time Series Smoothing and Forecasting Using the EM Algorithm," *J. Time Ser. Anal*,. Vol. 3, no. 4, pp. 253-264, 1982.
[25]. Z. Ghahramani and G. Hinton, *Parameter Estimation for Linear Dynamic Systems* (Tech. Rep. CRG-TR-96-2). Toronto, 1996: Department of Computer Science, University of Toronto. Available from ftp://ftp.cs.toronto.edu/pub/zoubin/.
[26]. V. Digalakis, J.R. Rohlicek and M. Ostendorf, "ML Estimation of a Stochastic Linear System with the EM Algorithm and its Application to Speech Recognition," *IEEE Trans. Speech and Audio Proc.* Vol. 1, no. 4, pp. 431-442, 1993
[27]. R. Schluter, *Investigations on Discriminative Training Criteria,* Ph. D Dissertation, RWTH Aachen University, Aachen, Germany, 2000
[28]. B.-H. Juang, W. Chou, and C.-H. Lee, "Minimum Classification Error Rate Methods for Speech Recognition," *IEEE Trans. Speech Audio Proc.* vol. 5, May 1997.
[29]. V. Doumpiotis and W. Byrne. "Lattice segmentation and minimum Bayes risk discriminative training for large vocabulary continuous speech recognition," *Speech Communication*, (2):142–160, 2005.
[30]. V. N. Vapnik, "An Overview of Statistical Learning Theory," *IEEE Trans. Neural Net.*, vol. 10, no. 5, Sep. 1999.
[31]. S. Greenberg and E. Fosler-Lussier, "The Uninvited Guest:: Information's Role in Guiding the Production of Spontaneous Speech," *Proc. of CREST Workshop on Models of Speech Production: Motor Planning and Articulatory Modeling*, Kloster Seeon, Germany, 2000.

[32]. I. Shafran and M. Ostendorf, "Acoustic Model Clustering Based on Syllable Structure," *Computer Speech and Language*, vol. 17(4), pg. 311-328, 2003.
[33]. B.-H. Juang, "Detecting Based Processing for Speech Recognition and Understanding," in *NSF Symposium on Next Generation ASR*, Atlanta, GA, October 2003.
[34]. C.-H. Lee, "A New Collaborative ASR Paradigm: Is the Glass Half Full or Half Empty?" in *NSF Symposium on Next Generation ASR*, Atlanta, GA, October 2003.
[35]. K. Hacioglu, B. Pellom and W. Ward, "Parsing Speech into Articulatory Events," *Proc. IEEE ICASSP2004*, pp. SP-P14.9, May 2004.
[36]. S. Hiroya and M. Honda, "Estimation of Articulatory Movements from Speech Acoustics Using an HMM-Based Speech Production Model," *IEEE Trans. Speech & Audio Proc.* vol. 12, no. 2, March 2004.
[37]. T. Kaburagi and M. Honda, "Determination of Sagittal Tongue Shape from the Positions of Points on the Tongue Surface," *J. Acoust. Soc. Amer.*, vol. 96, pp. 1356-1366, 1994.
[38]. Y. Bengio and P. Frasconi, "An input/ouput HMM architecture," in *Advances in Neural Information Processing Systems 7* (G. Tesauro, D. Touretzky, and T. Leen, eds.), pp. 427-434, MIT Press, Cambridge, MA, 1995.
[39]. S. Greenberg, "Speaking in Shorthand-A Syllable-centric Perspective for Understanding Pronunciation Variation," *Speech Communication*, vol. 29, pp. 159-176, 1999.
[40]. L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, *Classification and Regression Trees*, Wadsworth International Group, Belmont, CA, 1984.
[41]. M.I. Jordan, "A Statistical Approach to Decision Tree Modeling," *Proc. 7th Annual ACM Conference on Computational Learning Theory*, 1994.
[42]. H. A. Chipman, E. I. George, R. E. McCulloch, "Bayesian CART Model Search," *JASA*, vol. 39, no. 443, pp. 935-948, 1998.
[43]. J.D. McAuliffe, M.I. Jordan, and L. Pachter, "Subtree Power Analysis and Species Selection for Comparative Genomics," *Proc. National Academy of Sciences*, vol. 102, pp. 7900-7905, 2005.
[44]. J. J. Odell, *The Use of Context in Large Vocabulary Speech Recognition*, Ph. D Dissertation, Cambridge University, 1995.
[45]. J.H. Friedman, "Greedy Function Approximation: a Gradient Boosting Machine," *Annals of Statist.* 29, pp. 1180, 2001.
[46]. J.J. Verbeek, N. Vlassis, "Efficient Greedy Learning of Gaussian Mixture Models," *Neural Comp.* 5(2): 468-485, 2003.
[47]. R.-S. Hu, X. Li and Y. Zhao, "Acoustic Model Training Using Greedy EM," *Proc. ICASSP05*, pp. I697-700, Philadelphia, PA, March 2005.
[48]. G. Schwarz, "Estimating the Dimension of a Model," *Ann. Statist.*, vol. 6, no. 2, pp. 461-464, 1978.

[49]. R.M. Neal, G.E. Hinton, "A View of the EM Algorithm that Justifies Incremental, Sparse, and Other Variants," *Learning in Graphic Models,* Kluwer Academic Publishers, Dordrecht, the Netherlands, 355-368, 1998.
[50]. Y. Tsao, J. Li, and C.-H. Lee, "A Study on Separation between Acoustic Models and its Applications," *Proc. InterSpeech2005,* pp. 1109-1112, Sep. 2005.
[51]. P.S. Gopalakrishnan, D. Kaneysky, A. Nadas, D. Nahamoo, and M.A. Picheny, "Decoder Selection Based on Cross-Entropies," *Proc. ICASSP88,* pp. 20-23, 1988.
[52]. Y. Normandin, "Optimal Splitting of HMM Gaussian Mixture Components with MMIE Training," *Proc. ICASSP95,* vol. 1, May 1995
[53]. W. Chou and L. Li, "A Minimum Classification Error (MCE) Framework for Generalized Linear Classifier in Machine Learning for Text Categorization/Retrieval," *Proc. ICMLA04,* Dec. 2004.
[54]. R. Zhang and A. I. Rudnicky, "Improving the Performance of an LVCSR System through Ensembles of Acoustic Models," *Proc. ICASSP03,* April 2003.
[55]. S. Seneff, "The use of subword linguistic modeling for multiple tasks in speech recognition," *Speech Communication,* vol. 42, pp. 373-390, April 2004.
[56]. D. Jurafsky, W. Ward, J. Zhang, K. Herold, X. Yu and S. Zhang, "What kind of pronunciation variation is hard for triphones to model?" *Proc. ICASSP01,* pp. 577-580, Salt Lake City, Utah, May 2001.
[57]. T. Hain, "Implicit modeling of pronunciation variation in automatic speech recognition," *Speech Communication,* vol. 26, pp. 171-188, 2005
[58]. M. Riley, B. Byrne, M. Finke, S. Khudanpur, A. Ljolje, J. McDonough, H. Nock, M. Saraclar, C. Wooters and G. Zavaliagkos, "Stochastic pronunciation modeling from hand-labeled phonetic corpra," *Speech Communication,* vol. 29, pp. 209-224, November 1999.
[59]. M. Saraclar, H.J. Nock, S. Khudarpur, "Pronunciation modeling by sharing Gaussian densities across phonetic models," *Computer Speech and Language,* vol. 14, pp. 137-160, 2000.
[60]. W. Reichl and W. Chou, "Robust decision tree state tying for continuous speech recognition," *IEEE Trans. Speech and Audio Proc.* Vol. 8, no. 5, pp. 555-566, September 2000.
[61]. J.-T. Chien and S. Furui, "Predictive hidden Markov model selection for speech recognition," *IEEE Trans. Speech and Audio Proc.* Vol. 13, no. 3, pp. 377-387, May 2005.
[62]. S. Wang and Y. Zhao, "Online Bayesian tree-structured transformation of HMMs with optimal model selection for speaker adaptation," *IEEE Trans. Speech and Audio Proc.* Vol. 9, no. 6, pp. 663-677, September 2001.
[63]. K. Shinoda and C.-H. Lee, "A structural Bayes approach to speaker adaptation," *IEEE Trans. Speech and Audio Proc.* Vol. 9, no. 3, pp. 276-287, March 2001.

[64]. H. Strik, C. Cucchiarini, "Modeling pronunciation variation for ASR: A survey of the literature," *Speech Communication*, vol. 29, pp. 225-246, 1999.
[65]. W. L. Buntine, *A Theory of Learning Classification Rules*, PhD thesis, School of Computing Science, University of Technology, Sydney, 1992.
[66]. H. A. Chipman, E. I. George and R. E. McCulloch, "Bayesian CART model search," *JASA*, vol. 93, no. 443, pp. 935-948, September 1998.
[67]. D. Denson, C. Holmes, B. Malick and A. Smith, *Bayesian Methods for Nonlinear Classification and Regression*, Willey, 2002.
[68]. Y. Zhao, X. Zhang, R_S. Hu, J. Xue, X. Li, L. Che, R. Hu and L. Schopp, "An automatic captioning system for telemedicine," *Proc. ICASSP06*, to appear.
[69]. The HTK Toolkit, http://htk.eng.cam.ac.uk/

V ITA

Rusheng Hu was born on March 20, 1971, in Cangzhou, Heibei, China. He has earned the following degrees: B.E. in Hydraulic Engineering from Tsinghua University (Beijing, China); M.S. in Civil Engineering, M.S. in Computer Science, and Ph.D in Computer Science, all from the University of Missouri at Columbia, Missouri. He is presently a member of the Capital One Financial Services, Richmond, Virginia, USA.