Friedrich-Alexander-Universität Erlangen-Nürnberg

Lehrstuhl für Multimediakommunikation und Signalverarbeitung

Masterthesis

Combined-Order Hidden Markov Models for Reverberation-Robust Speech Recognition

Sujan Reddy, Kotha

November 2011

Advisors: Prof. Dr.-Ing. Walter Kellermann,
Roland Maas, M.Sc.

Erklärung

Ich versichere, dass ich die vorliegende Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe, und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

_____ _____

Ort, Datum Unterschrift

Contents

Abstract

Automatic Speech Recognition (ASR) Systems work very reliably if close-talking microphones are used for speech input. If the distance between speaker and microphone increases, the recognition is often hampered by reverberation and other types of distortions. A major reason for this is that typical recognizers are based on first-order Hidden Markov Models (HMMs) assuming that the current speech feature vector is conditionally independent of the previous ones. Reverberation, however, has a dispersive effect on the feature, which significantly increases the inter-frame correlation and thus limits the performance of such recognizers.

In this thesis, the concepts of first- and second-order HMMs are combined to form a "combined-order" HMM (CO-HMM), such that the CO-HMM has transition probabilities dependent only on the previous state and each state is composed of different output probability density functions (PDFs) depending on its predecessor. For training, initially the Baum-Welch method is employed to set up a conventional first-order HMM. Then, the Information Combining Estimation With Non-reverberant Data (ICEWIND) approach is used to estimate predecessor dependent output PDFs. For recognition, the Viterbi decoder is adapted accordingly. Finally, connected-digit recognition experiments based on the TI digit corpus are carried out for three different recognition units, words, phonemes, and triphones, to assess the performance of the proposed concept.

Abbreviations

ASR	Automatic Speech Recognition
CDR	Connected Digit Recognition
CMS	Cepstral Mean Subtraction
CO-HMM	Combined-Order Hidden Markov Model
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DTFT	Discrete-Time Fourier Transform
EM	Expectation Maximization
FO-HMM	First-Order Hidden Markov Model
FVS	Feature Vector Sequence
HHI	Human/Human Interaction
HMI	Human/Machine Interaction
HMM	Hidden Markov Model
HO-HMM	Higher-Order HMM
HTK	HMM Tool Kit
ICEWIND	Information Combining Estimation With Non-reverberant Data
logmelspec	logarithmic mel-spectral
melspec	mel-spectral
MFCC	Mel-Frequency Cepstral Coefficient

ML	Maximum Likelihood
PDF	Probability Density Function
RIR	Room Impulse Response
SFA	State Frame Alignment
SNR	Signal-to-Noise Ratio
SRR	Signal-to-Reverberation Ratio
SS	Steady State
STFT	Short-Time Fourier Transform
TS	Transition State
WER	Word Error Rate

Symbols

t	continuous time index
n	discrete time index
f	index of DFT coefficients
k	frame index after feature extraction
m	reverberation frame index (frame lag)
j	index of current state
i	index of previous state
r	index of current HMM
K	length of FVS in numbers of frames
M	length of RVM in numbers of frames
I	number of ceptral coefficients $I < L$
$s(n)$	time-domain clean-speech signal
$b(n)$	time-domain signal of additive interferences,
	like background noise and competing speakers
$x(n)$	time-domain reverberant speech signal
$h(n)$	room impulse response in the time domain
$\mathbf{s}(k)$	clean-speech feature vector at frame k
	alternative use: complete feature vector sequence
$s(l,k)$	l-th element of $\mathbf{s}(k)$

$\mathbf{S}(1:K)$	feature vector sequence for frames $k = 1, ..., K$
$\mathbf{s}$	random process corresponding to $\mathbf{s}(k)$
$\mathbf{S}(k)$	random process modeling $\mathbf{s}(1:K)$
$\mathbf{S}$	FVS of undefined length. modeled as random process
$\mathbf{s}_m(k)$	feature vector at frame k, domain not specified
$\mathbf{s}_l(k)$	feature vector at frame k, melspec domain
$\mathbf{s}_c(k)$	feature vector at frame k, logmelspec domain
ω	word sequence of undefined length
W	random process modeling ω
$P(W = \omega \vert S = s)$	probability that the word sequences $W = \omega$,
	given the feature sequence $\mathbf{S} = \mathbf{s}$
$\mathbf{C}$	DCT matrix
$\mathbf{B}$	selection matrix
$q(k)$	state at frame k
	alternative use: state sequence
$q(1:K)$	state sequence for frames $k = 1, ..., K$
q	state squence of undefined length
$Q(k), Q(1:K), Q$	random process corresponding to $q(k), q(1:K), q$
λ	a single HMM
Λ	HMM sequence, i.e., concatenation of HMMs
$R(k)$	random process of HMM indices r
$f_X(x)$	PDF of random variable X as function of x
$f_{\mathbf{S}(k)\vert Q(k)=j}(\mathbf{s})$	conditional PDF of random process $\mathbf{S}(k)$, given
	$Q(k) = j$, i.e., output PDF of HMM state j
$\phi_j(k)$	Viterbi score of j at frame k
$\psi_j(k)$	backtracking pointer of state j at frame k
μ_X	(linear) mean of random variable X

Chapter 1

Motivation

The role of Human-to-Machine Interaction (HMI) in our everyday lifes is on the rise and the demand for more interfaces grows continuously with the availability of increasing powerful computational resources.Speech as a HMI is a very convenient and desirable application for human beings, as speech is a natural way of communication between humans.

HMI systems in an acoustic environment need to interpret and understand the sound captured by the machine through a single or several microphones and respond (sound production). These systems work very reliably if close-talking microphones are used for speech input. But for a truly natural HMI, the speaker need to enjoy the freedom of communicating via distant-talking microphone. If the distance between the speaker and the microphone increases, the microphone not only picks the desired signal, but also additive interferences, like background noise or undesired speaker signals, echoes of loudspeaker signals and reverberation of the desired signal.

Automatic Speech Recognition (ASR) is often a part of acoustic HMI systems, where the input signal has to be interpreted and understood. It is a technology that allows a machine to identify the words a speaker speaks into the input device and convert it to written text. Today, most practical speech recognition systems are based on the statistical framework. Based on major advances in statistical methods, most notably Hidden Markov Models (HMMs), sophisticated systems are developed that

respond to fluently spoken natural language. However, these systems are very sensitive to reverberation and their performance get degraded considerably with it.

The major reason for this is that typical recognizers are based on first-order HMMs assuming that the current speech feature vector is conditionally independent of the previous ones. Reverberation, however has a dispersive effect on the feature vector, which significantly increases the inter-frame correlation and thus limits the performance of such recognizers. Different techniques have already been investigated in order to model the inter-frame dependency, e.g., differential features [10] and frame-wise model adaptation [26,27].

The aim of this master thesis to implement a concept called Combined-Order Hidden Markov Model (CO-HMM), which is a model-based approach to reverberant robust speech recognition in distant talking scenarios. The concepts of first-order and second-order HMMs shall be merged to form a "combined-order" HMM. Such a CO-HMM is to be designed so that the transition probabilities are independent of the previous state, whereas each state is composed of different output PDFs depending on its predecessor.

The thesis is organized as follows: In chapter 2, basics of ASR like preprocessing, feature extraction, training and recognition, using HMMs are described in detail. In the next chapter, reverberation and its characteristics, its effects on the performance of ASR and model based approaches for reverberation compensation are elaborated. In chapter 3, the concept of CO-HMM and the procedures involved in building such a HMM are discussed in detail. chapter 4 presents the experimental setup, results and discussions. In the final chapter, conclusions are drawn from this thesis.

Chapter 2

Automatic Speech Recognition

Automatic speech recognition is interdisciplinary in nature. The different areas to be investigated in this context include the process of extracting useful information from the speech signal, the procedures for estimating the parameters of statistical models, the relationship between sounds, words and grammar rules in a language, understanding the relationship between the physical speech signal and the mechanisms that produce the speech, understanding how speech is perceived by human beings, implementing efficient algorithms in software or hardware and understanding the factors that help in applying this task in practical situations [2].

In this chapter, the fundamentals of speech recognition algorithms that make use of HMMs are described. In the first two sections, initial procedures in ASR, preprocessing and feature extraction are described. Next, a brief introduction to HMMs and then algorithms to train and recognize a HMM are described in detail. Then, the relation between isolated and continuous speech speech recognition is explained. At the end of this chapter, different recognition units and their advantages and disadvantages are discussed.

Figure 2.1 shows a block diagram of a typical speech recognition system. The stages involved in an ASR system are preprocessing, feature extraction, training an acoustic model and finally recognition based on this trained acoustic model and a language model. For increasing robustness against reverberation, measures can be embed-

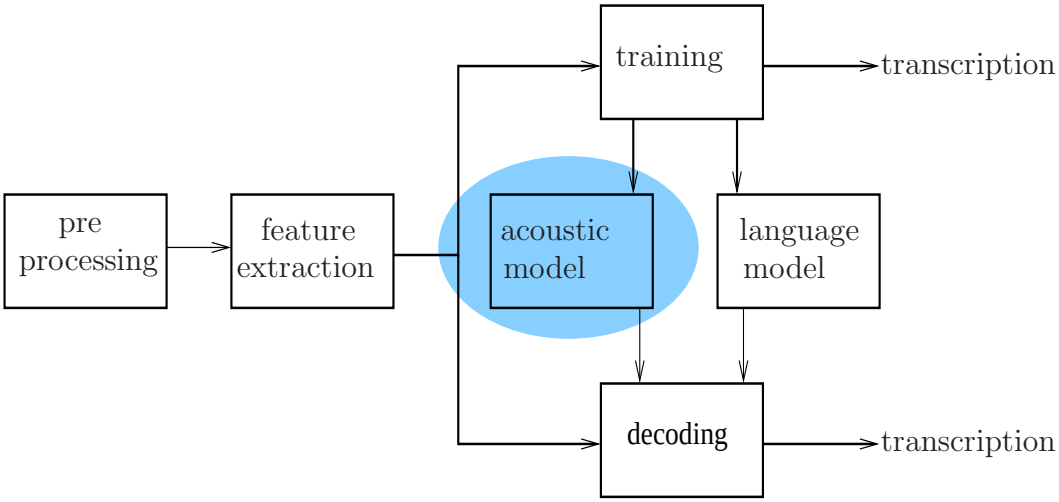


Figure 2.1: Block diagram of an ASR system

ded into the ASR system in the preprocessing, feature extraction, acoustic model and recognition. In this thesis, the focus is on the model based reverberation compensation techniques, which are described in section 3.5 and in chapter 4.

2.1 Preprocessing

Preprocessing involves the application of fundamental signal processing before the extraction of desired features from the speech signal. In a distant talking scenario, preprocessing includes removing additive interferences and dereverberation in the signal and frequency domain.

Noise can cause significant changes to the characteristics of the speech signal. If the additive noise and speech signals are statistically dependent, then there is a modification of the spectrum and characteristics of the speech signal [3]. To enhance the signal with additive interferences, noise reduction techniques [3] and beamforming techniques [3] can be applied.

Dereverberation techniques like reverberation cancellation [20], reverberation suppression [20] and beamforming can be applied in signal domain to remove the reverberation. Its main goal is estimation of clean-speech signal $s(n)$ from the microphone

signal $x(n)$.

2.2 Feature Extraction

Extracting features from speech is a fundamental necessity of any speech recognition system. The Goal is to classify the source files using a reliable representation that reflects the difference between utterances. Speech is non-stationary and to approximate the input speech signal to a quasi stationary signal, a window function is applied in the preprocessing stage to divide speech into small segments. A feature vector is usually computed from a window of speech signals (20...30 ms) in every short time interval (about 10 ms). An utterance is represented as a sequence of these feature vectors.

Mel-Frequency Cepstral Coefficients (MFCCs) [5] are the most commonly used features for human speech analysis and recognition. Since the human auditory system does not perceive the frequency on a linear scale, researchers have developed the Mel-scale in order to approximate the humans perception scale. The Mel-scale is a logarithmic mapping from physical frequency to perceived frequency [6]. The MFCCs are extracted using this frequency scale. Figure 2.2 shows the flow graph of MFCC extraction procedure. The steps involved in extracting MFCC features are as follows:

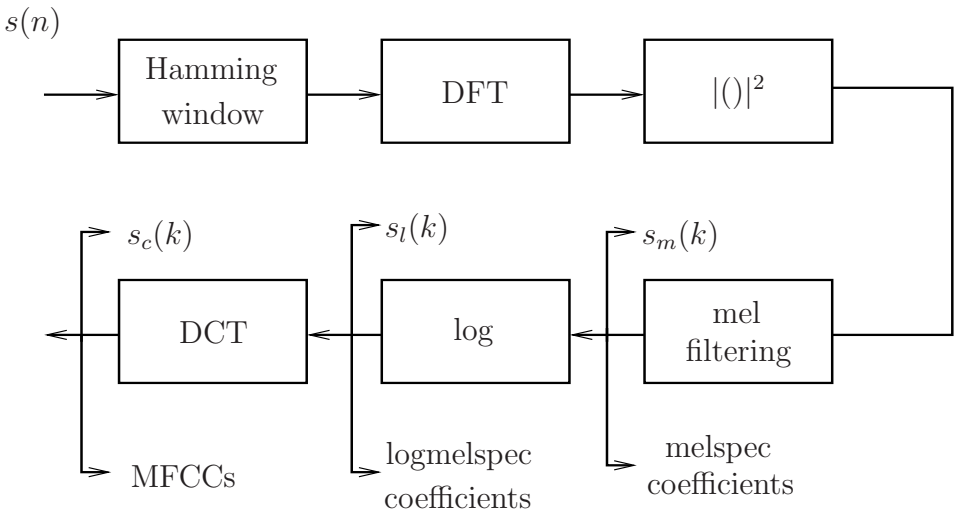


Figure 2.2: Block Diagram of MFCCs extraction

1) A short-time spectrum analysis is performed with a Hamming window.

$$w_H(m) = \begin{cases} (1-\alpha) - \alpha * cos[2\pi/(M-1)], & 0 \leq m \leq M-1, \\ 0, & m = else, \end{cases}$$

where M is the window length, $\alpha = 0.46$

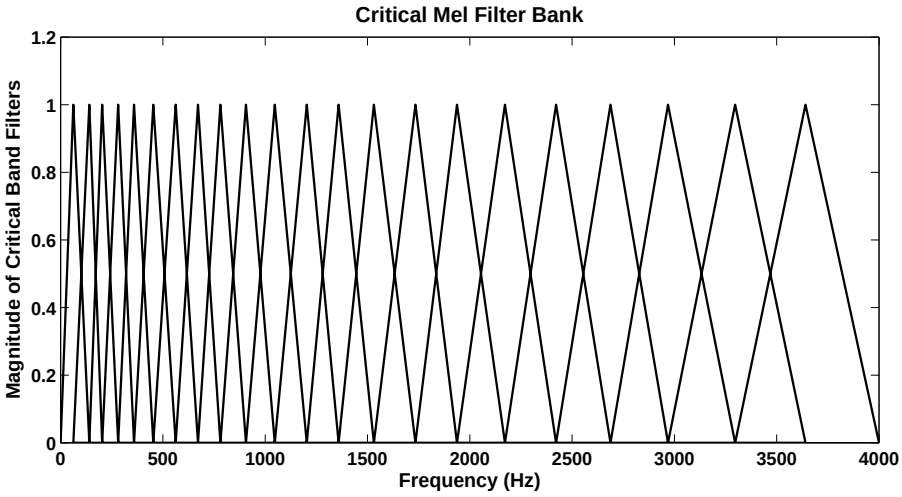
2) A f-point Discrete Fourier Transform (DFT) for a windowed input speech signal is applied

$$S(f,k) = \sum_{n=0}^{F-1} w_H(n) s(kN+n) e^{-j\frac{2\pi}{F}nf}, \tag{2.1}$$

where f is the index of the DFT bin, k is the frame index, $N \leq F$ is the frame shift.

3) The magnitude square of the filter coefficients $|S(f,k)|^2$ is calculated to obtain mel-spectral coefficients by applying a mel filter bank $C(l,f)$.

$$s_m(l,k) = \sum_{f=0}^{F/2} C(l,f)|S(f,k)|^2 \tag{2.2}$$



4) Next, logmelspec coefficients are obtained by applying the logarithm to melspec coefficients

$$s_l(k) = log s_m(k), \tag{2.3}$$

where

$$s_m(k) = [s_m(1,k), s_m(2,k), .., s_m(L,k)]^T,$$

T is a Matrix transpose, L is the number of mel-channels and l denotes "logmelspec domain"

5) Finally, Discrete Cosine Transform (DCT) is applied to the logmelspec features to obtain MFCCs:

$$s_c(k) = B.C.s_l(k), \tag{2.4}$$

$$C = C_{il} = \sqrt{\frac{2}{L}}.cos(\pi/L.i.(l+0.5)), \tag{2.5}$$

where C is the $L \times L$ DCT matrix, $B = [1_{I\times I} \quad 0_{I\times(L-1)}]$ is the selection matrix $I \times L$, where $I < L$.

The first I MFCCs are used as speech features to capture the spectral envelope of the time-frequency pattern.

In addition to static features (MFCCs), dynamic features (Δ and $\Delta\Delta$) features are added to exploit the temporal changes of short-time spectra [7]. They are the first and the second derivatives of Short-Time Fourier Transform (STFT) based features, like MFCCs, respectively. The derivatives are usually approximated by a simple differences, given by

$$\Delta s(\kappa) = s(k+\kappa) - s(k-\kappa), \tag{2.6}$$

or by linear regression calculations, given by

$$\Delta s(k) = \frac{\sum_{\kappa=1}^{K_\Delta} \kappa.(s(k+\kappa) - s(k-\kappa))}{2.\sum_{\kappa=1}^{K} \kappa^2} \tag{2.7}$$

where typical values for κ=1 or 2 and K_Δ={ 2, 3, 4}.

2.3 Hidden Markov Models

HMMs are widely used in an ASR because of its efficient implementation of the overall recognition system and its characteristic statistical framework. For estimating the

parameters of the models from a finite training sets of speech data, efficient algorithms are available and the recognition system obtained has the flexibility to change the size, type, or architecture of the models to suite particular words or sounds.

HMMs are commonly defined as stochastic finite state machines, which produce an observation by concatenation of two random experiments, the first one determines the unobservable state and the second one produces the given observation based on the chosen state [8].

Consider an example: Let N glass urns contain M colored balls in each of it. In an experiment, T balls are chosen randomly from a randomly selected urn and the color of the ball is noted and the ball is replaced again in the same urn. A new urn is then selected according to the random selection process associated with the current urn and the selection procedure of balls is repeated. This experiment generates finite observation sequence of colors.

A simple HMM corresponding to this model can be described as each urn corresponds to a state, the colored balls are the observations and for each color there is an observation probability depending on which state it belongs to. A state transition matrix has the probabilities for transition from urn (state) i to j. Figure 2.3 illustrates above example.

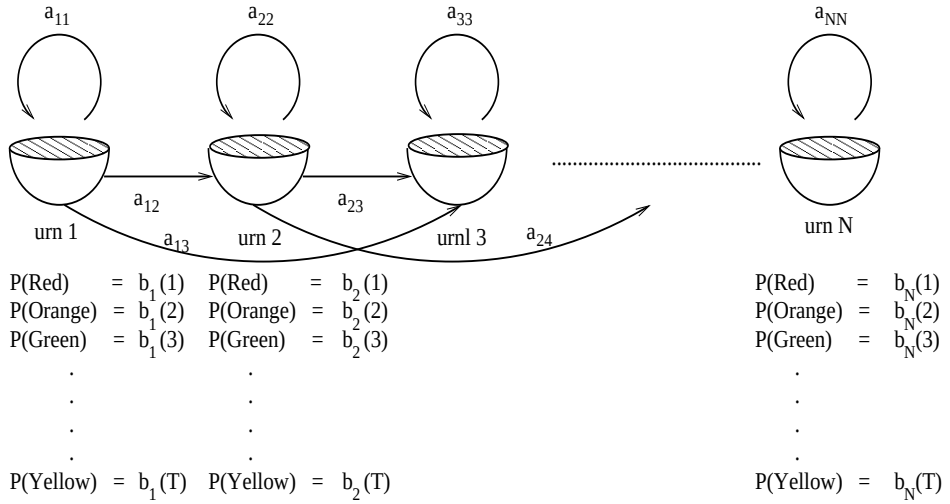


Figure 2.3: Example of a simple HMM

Figure 2.4 shows a pictorial representation of a HMM. A HMM can be characterized as follows

- It has a series of N states within the model which are responsible for representing the target which needs to be modeled.
- M, the number of observation symbols per state. O=$\{o_1, o_2....o_M\}$
- Each of the state has its associated output probability distribution B=$b_j(o_t)$, where o_t is the observation vector at time t and j is the state index, where

$$b_j(o_t) = P(o_t|q_t = j).$$

- Each pair of these states has a transition probability between them. A=$\{a_{ij}\}$ is a set of transition probabilities between states, where

$$a_{ij} = P(q_{t+1} = j|q_t = i).$$

- $\pi = \{\pi_i\}$ is the initial distribution

$$\pi_i = P(q_1 = i) \ \forall \ i\text{=}1,...,N$$

For convenience, a compact notation $\lambda = \{A, B, \pi\}$ is used to represent the complete parameter set of the model.

The two fundamental assumptions in a first-order HMM are the Markov assumption which states that the current state q_t depends only on the previous state $q_{(t-1)}$ and not on the earlier states, and the conditional independence assumption, which states that the output vector o_t for a given state q_t is independent of all the previous states and the output feature vectors. The Markov assumption and the conditional independence assumption can be written as follows
The Markov assumption:

$$a_{ij} = p(q_{t+1} = j|q_t = i). \tag{2.8}$$

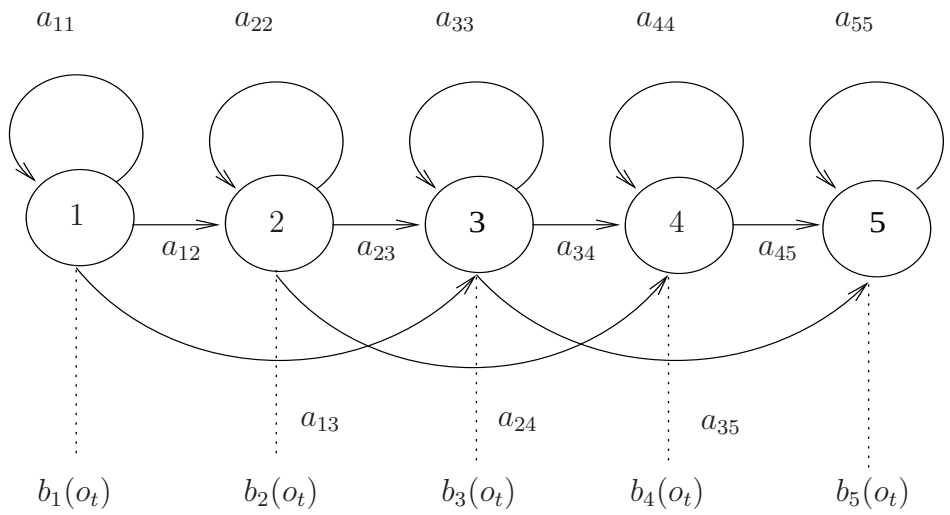


Figure 2.4: Five state Hidden Markov model

The Conditional independence assumption:

$$P(O|q_1, q_2, ..., q_T, \lambda) = \prod_{t=1}^{T} P(o_t|q_t, \lambda). \tag{2.9}$$

2.3.1 Training an HMM

The HMMs employed in practice are first-order HMMs. For convenience a first-order HMM is considered as HMM in this chapter and as FO-HMM in the later discussions for comparisons between different order HMMs. Training a HMM aims at finding optimized parameters of that model to maximize $P(O|\lambda)$. A standard algorithm used to train an HMM is Baum-Welch method, which is described as follows

Baum-Welch training:

Baum-Welch training is an iterative process of estimating the parameters of the model using the Maximum Likelihood (ML) estimation, until the optimized parameters found by some fixed number of iterations or some termination criteria is fulfilled.

In the training procedure, an initial model is taken and utilizing the input speech data and the associated transcription, parameters of this model are re-estimated, thus a new model is created. The aim of the training is to find the model, say $\hat{M}$, such that:

$$\hat{M} = \arg\max_{M} P(O|M), \tag{2.10}$$

where O is the given observation sequence and $P(O|M)$ is the likelihood of that sequence given the model. Figure 2.5 shows the flow graph of the Baum-Welch algorithm.

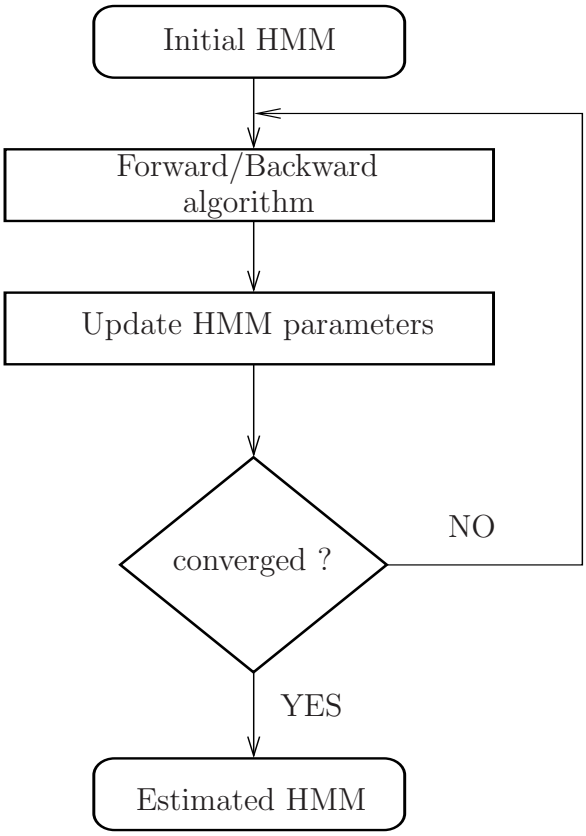


Figure 2.5: Baum-Welch Algorithm

The maximum likelihood estimates of mean μ_j and variance $\hat{\sum}_j$ of a HMM can be calculated by

$$\hat{\mu}_j = \frac{\sum_{t=1}^{T} L_j(t) o_t}{\sum_{t=1}^{T} L_j(t)}, \tag{2.11}$$

$$\hat{\sum_{j}} = \frac{\sum_{t=1}^{T} L_j(t)(o_t - \mu_j)(o_t - \mu_j)'}{\sum_{t=1}^{T} L_j(t)}, \tag{2.12}$$

where $L_j(t)$ denotes the probability of being in state j at time t. To apply equations 2.11 and 2.12 the probability of state occupation $L_j(t)$ must be calculated. This is done efficiently using the so-called Forward-Backward algorithm. Given a HMM, the forward probability $\alpha_j(t)$ is defined as the joint probability of having generated the partial observation sequence from time 1 to time t and having arrived state j at time t and a backward probability $\beta_j(t)$ is the probability of generating the partial observation sequence from time t to time T, such that the state sequence starts from state j at time t [1]. The forward probability can be calculated by the following recursion formula

$$\alpha_j(t) = [\sum_{i=2}^{N-1} \alpha_j(t-1).a_{ij}]b_j(o_t), \tag{2.13}$$

where N is the total number of states in the given HMM with 1 and N non emitting states. Initial conditions for above recursion are

$$\alpha_1(1) = 1 \tag{2.14}$$

$$\alpha_j(1) = a_{1j}b_j(o_1) \tag{2.15}$$

for $1 < j < N$. Similarly, the backward probability can be calculated by a backward recursion:

$$\beta_i(t) = \sum_{j=2}^{N-1} a_{ij}b_j(o_{t+1})\beta_j(t+1) \tag{2.16}$$

with initial condition

$$\beta_i(T) = a_{iN} \tag{2.17}$$

for $1 < i < N$. Therefore, the product of these two denotes the joint probability $\alpha_j(t)\beta_j(t)$ of generating the incoming observation sequence and arriving at state j at time t.

$$\alpha_j(t).\beta_j(t) = P(O, q_t = j|M). \tag{2.18}$$

Therefore, the probability of state occupation $L_j(t)$ can be calculated as follows

$$L_j(t) = P(q_t = j|O, M) = \frac{P(O, q_t = j|M)}{P(O|M)} = \frac{\alpha_j(t).\beta_j(t)}{P(O|M)}, \tag{2.19}$$

Note that at any time t, all possible state sequences must merge into one of the states. Thus the desired probability $P(O|M)$ is simply computed by summing all the forward and backward products as shown below

$$P(O|M) = \sum_{j=1}^{N} \alpha_j(t)\beta_j(t), \tag{2.20}$$

For a HMM with mixture components, means $\hat{\mu}_{im}$, covariance matrices $\hat{c}_{im}$, mixture weights w_{im} and transition probabilities $\hat{a}_{im}$ are re-estimated as follows.

$$\hat{\mu}_{im} = \frac{\sum_{t=1}^{T} \delta_{im}(t) o_t}{\sum_{t=1}^{T} \delta_{im}(t)} \tag{2.21}$$

$$\hat{c}_{im} = \frac{\sum_{t=1}^{T} \delta_{im}(o_t - \mu_{im})(o_t - \mu_{im})'}{\sum_{t=1}^{T} \delta_{im}(t)} \tag{2.22}$$

$$w_{im} = \frac{\sum_{t=1}^{T} \delta_{im}(t)}{\sum_{t=1}^{T} \delta_{im}(t)} \tag{2.23}$$

$$\hat{a}_{im} = \frac{\sum_{t=1}^{T} \alpha_i(t) a_{ij} b_j(o_{t+1}) \beta_j(t+1)}{\sum_{t=1}^{T} \alpha_i(t)\beta_i(t)} \tag{2.24}$$

Where $\delta_{im}(t)$ denotes the probability of the observation sequence occupying the m^{th} mixture component of state i at time t and denotes the probability of the observation sequence occupying the state i at time t. They can be expressed as follows:

$$\delta_i(t) = \sum_{m=1}^{M} \delta_{im}(t) = \sum_{m=1}^{M} \frac{1}{p} \sum_{j=1}^{N} \alpha_j(t-1) a_{ij} w_{im} b_{o_t} \beta_i(t) \tag{2.25}$$

where M is the total number of Gaussian mixture components in state i and N is the total number of states in the model.

2.3.2 Recognition

The task of a speech recognizer is to find the best estimate $\hat{w}$ of a true word sequence w corresponding to a certain utterance, from the respective speech feature vectors derived from the speech signal. The word sequence w is modeled as random process W and the sequence of observed speech vectors $s(k)$ is modeled by a vector-valued random process $S(K)$ [10]. Representing all observed vectors from frame $K = 1$ to $k = K$ as $S = S(1 : K)$, the recognition problem can be expressed as

$$\hat{w} = \arg\max_{w} P(W = w|S = s). \tag{2.26}$$

Using the Bayes theorem 2.20 can be written as

$$P(W = w|S = s) = \frac{P(S = s|W = w).P(W = w)}{P(S = s)} \tag{2.27}$$

Recognition problem now becomes maximizing the product of the likelihood $P(S = s|W = w)$ and the prior probability $P(W = w)$ as shown below

$$\hat{w} = \arg\max_{w} P(S = s|W = w).P(W = w). \tag{2.28}$$

Given a sequence of observation vectors, the recognizer, using the trained acoustic model, has to identify the correct word from available set of word models. Hence, the model which yields the maximum value of $P(O|M_i)$ has to be determined. But in practice, the recognition is based on the maximum likelihood state sequence to generalize for continuous speech [1]. For a given model M, let $\phi_j(t)$ represent the maximum likelihood of observing speech vectors o_1 to o_t and being in state j at time t. The partial likelihood is computed recursively using the Viterbi algorithm. It is a dynamic programming algorithm for finding the most likely sequence of hidden states called the Viterbi path, that results in a sequence of observed events.

Let us assume that the HMM starts in state 1 and ends in the last state N at the final frame T of the sequence $o_{(1:T)}$. The partial likelihood of observing speech vectors o_1 to o_t and being in state j at time t can be expressed as

$$\phi_j(t) = \max_i\{\phi_i(t-1).a_{ij}\}.b_j(o_t), \tag{2.29}$$

$$\psi_j(t) = \max_i\{\phi_i(t-1).a_{ij}\}, \tag{2.30}$$

where initial likelihood of observing speech vector o_1 and being in state 1 at time 1 is given by

$$\phi_1(1) = 1 \tag{2.31}$$

The likelihood of observing speech vector o_1 and being in state j at time 1 is given by

$$\phi_j(1) = a_{1j}.b_j(o_1). \tag{2.32}$$

for $1 < j < N$. The maximum likelihood state sequence representing a model M i.e., $\hat{P}(O|M)$, is given by

$$\phi_N(T) = \max_i \phi_i(t)a_{iN}. \tag{2.33}$$

This algorithm can be illustrated using a trellis diagram. For example, let us consider the recognition process using a five state HMM with words as recognition units, such that there are no skips between the states. The transition probability from initial non emitting state to the first state is one and the likelihood of observing speech vector o_1 and being in the first state is one.

This algorithm can be visualized as finding the best path through a matrix, where the vertical dimension represents the states of the HMM and the horizontal dimension represents the frames of speech (i.e. time), as shown in Figure 2.6. Every dot in the trellis diagram represents the Viterbi score $\phi_j(t)$ of the state j and the frame t and each arc between dots correspond to the non-zero transition probability between the respective states. From equation (2.29), we can infer that the Viterbi

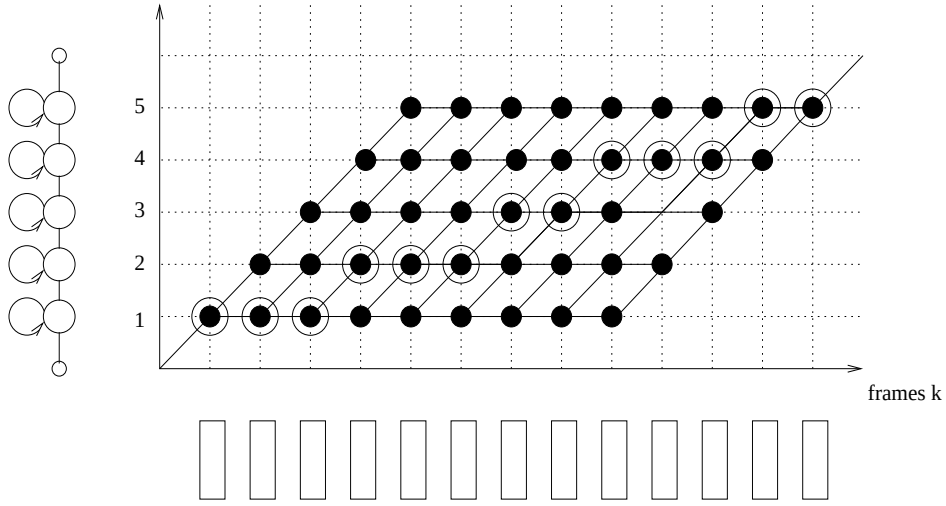


Figure 2.6: Viterbi Algorithm illustrated using Trellis diagram

scores are calculated step by step recursively by multiplying the score of the possible predecessor states with the corresponding transition probability, selecting maximum among all predecessors and then multiplying the output density of the current feature vector. Hence, in the recursion finding $\phi_5(T)$ gives the final acoustic score for a given model and the backtracking matrix stores the most likely state sequence through the HMM.

2.4 Continuous speech recognition

Let $O = o_1, o_2....o_T$ be a sequence of observations, o_t be the observation at time t. Then isolated word recognition problem can the be regarded as computing

$$\arg\max_i P(w_i|O), \tag{2.34}$$

where w_i is the i^{th} vocabulary word. This probability is computed by the Bayes Rule

$$P(w_i|O) = \frac{P(O|w_i)P(w_i)}{P(O)}. \tag{2.35}$$

For a given set of prior probabilities $P(w_i)$, the most probable spoken word depends only on the likelihood $P(O|w_i)$.

Recognition of continuous speech simply involves connecting HMMs together in sequence [1]. Each model in the sequence could be either whole a word for connected speech recognition or sub-words, such as phonemes and tiedstate triphones for

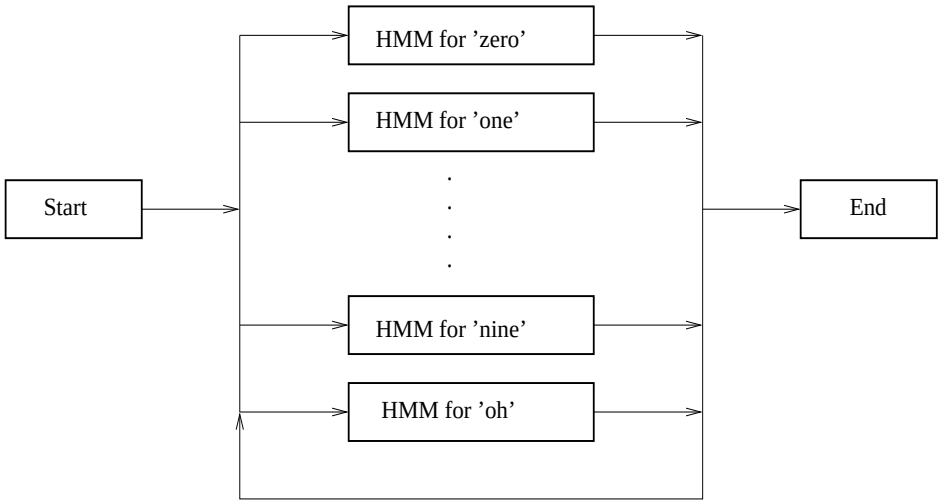


Figure 2.7: HMM Network for connected digit recognition

continuous speech recognition. In this case, the training data set contains continuous utterances. Hence, the boundaries dividing words or sub-words are unknown. A special training method called embedded training is used to solve the purpose. This method uses the Baum-Welch algorithm such that all models are trained in parallel. In recognition, the Viterbi algorithm is alternatively formulated by the Token Passing algorithm [11].

2.5 Recognition units and types

Recognition units play an important role in a speech recognizer. These units can be of different lengths and each choice has its own advantages and disadvantages. The longer the unit is, the more accurately it will model the effects of context dependency, but more training data will be required. The units decided should be consistent and trainable. In this section, a brief description of recognition units words, phonemes, tiedstate triphones is given.

2.5.1 Words

Words are the basic units to be recognized. They are considered to be context-dependent as the phonemes in a word are very likely the same in each utterance. However, using word models as recognition units for a very large vocabulary requires a large amount of training data, making them unusable. In the word models, each model is divided into many states and in the recognition, the best path between these states decides the recognized word. Figure 2.8 shows the representation of a HMM with words.

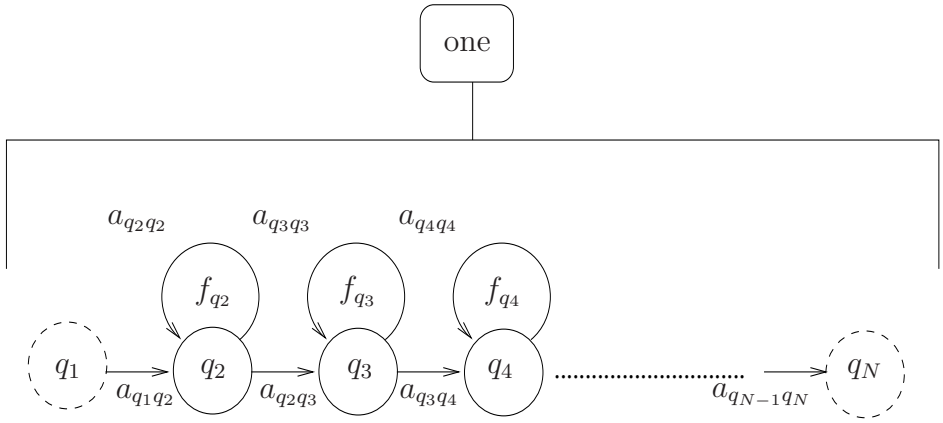


Figure 2.8: Representation of a word HMM

2.5.2 Phonemes

The phonemes are the basic sounds that form a word and the number of phonemes in most of the languages are moderate. Therefore, the phoneme models can be used as a recognition unit for a very large vocabulary as it is very easy to get enough training samples for each phoneme. However, the context in which the phoneme used is completely ignored. In a phoneme model, each phoneme is divided into states and in recognition the best path between the states decide the recognized phoneme and the combination of these phonemes decide the word recognized. Figure 2.9 shows the representation of a HMM with phonemes.

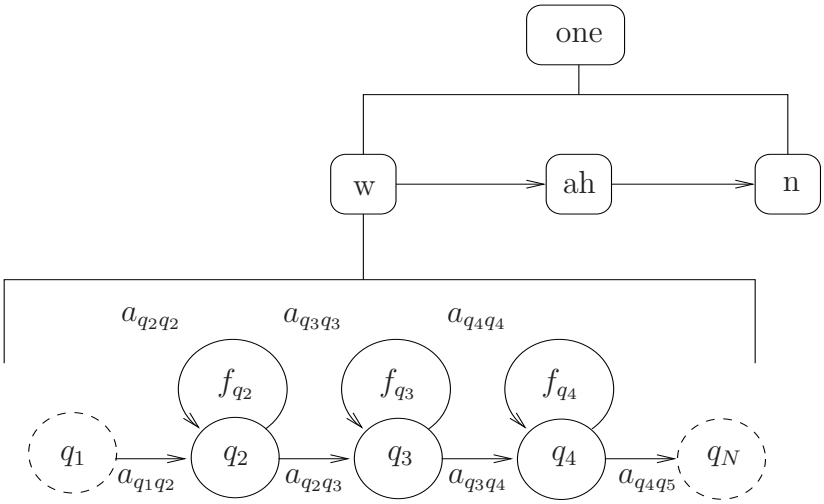


Figure 2.9: Representation of a phoneme HMM

2.5.3 Tiedstate triphones

The disadvantage with the phoneme level models can be compensated by using context-dependent phonemes called tiedstate triphones. A context dependent triphone

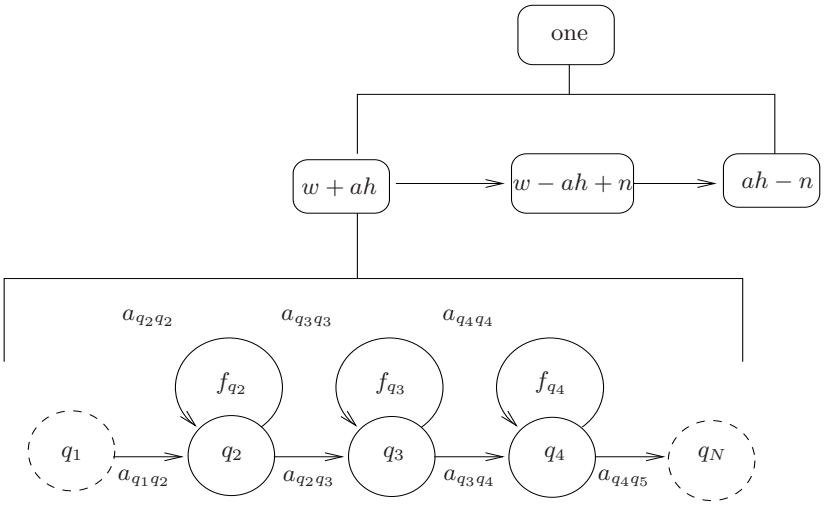


Figure 2.10: Representation of a triphone HMM

model can be built, either by word-internal or cross-word models. The word-internal model does not consider the context beyond the word borders, whereas a cross-word triphone model, considers the context of the neighboring words at the beginning and end of each phoneme of the current word to be recognized. In recognition, the process

is similar to phoneme level, here, the word recognized will be a combination of the tied state triphones. Figure 2.10 shows the representation of a HMM with triphones.

Chapter 3

Reverberation and its effects on ASR

In this chapter, the first few sections give an overview of reverberation, Room Impulse response (RIR), how reverberation can be modeled with RIRs and the measurements used for RIR and reverberation. Then, in the following sections, reverberation effects on ASR, change in the statistical characteristics of Feature Vector Sequences (FVSs) due to reverberation, and the model based approaches for this problem are described.

3.1 Reverberation

Reverberation is a phenomenon caused by the multiple reflections of the desired signal within a room. In a distant- talking scenario, the gain of the microphone amplifier has to be increased (compared to close-talking scenario) because of the longer distance between the desired speaker and the microphone. Due to this, the microphone picks delayed and attenuated copies of the desired signal, which are sensed as reverberation. Figure 3.1 shows a typical distant-talking scenario. The signals reaching the microphone by various paths can be categorized into three parts:

- direct signal: The direct signal takes a direct path to the microphone. The time

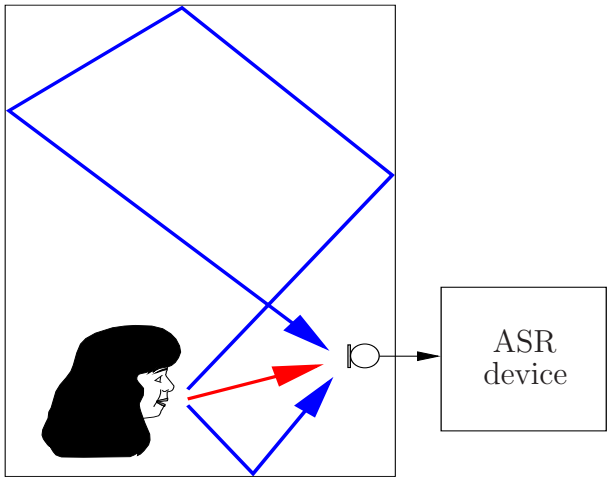


Figure 3.1: Distant talking scenario

delay between the source and its arrival at the microphone on the direct path can be calculated from the sound velocity c and the distance r from source to microphone.

- early reflections: The early reflections reach the microphone on the path of multiple reflections, approximately 50 to 100 ms after the direct signal and are relatively sparse.

- late reflections: The late reflections arrive at the microphone, following one another so closely that it is very hard to distinguish from one another and result in a diffused noise field.

Signal-to-reverberation ratio (SRR) is a useful measure for assessing the level of reverberation in a signal, which is defined as the ratio of a signal power to the reverberation power contained in a signal. It can be expressed as

$$SRR \triangleq 10log_{10}\frac{P_{signal}}{P_{reverberation}} = \varepsilon\{10log_{10}\frac{s^2}{(s * h_r)^2}\} \tag{3.1}$$

where s is the clean signal and h_r the impulse response of the reverberation.

3.2 Room Impulse Response

A natural approach to dereverberation, is the study of Acoustic Impulse Responses (AIRs) as they characterize the acoustics of a given enclosure. AIR limited to acoustics within a room are referred to as a RIR. In this thesis, the discussion is confined to RIRs.

The reverberation time T_{60} is defined as the time taken for the reverberant energy to decay by 60 dB once the sound source has been abruptly switched off [20]. It is an often used measure for RIRs and depends on room dimensions and reflecting coefficient of the surfaces. An RIR varies with the speaker, the microphones or other objects in the room, change location [20]. With the speaker-microphone separation, the relation between the energy of the direct-path component and the energy of the reflected components of the RIR changes. The critical distance is defined as the distance when the two energies, of the direct path and of the reflections are equal. Figure 3.2 shows an example of a RIR with T_{60} 600ms.

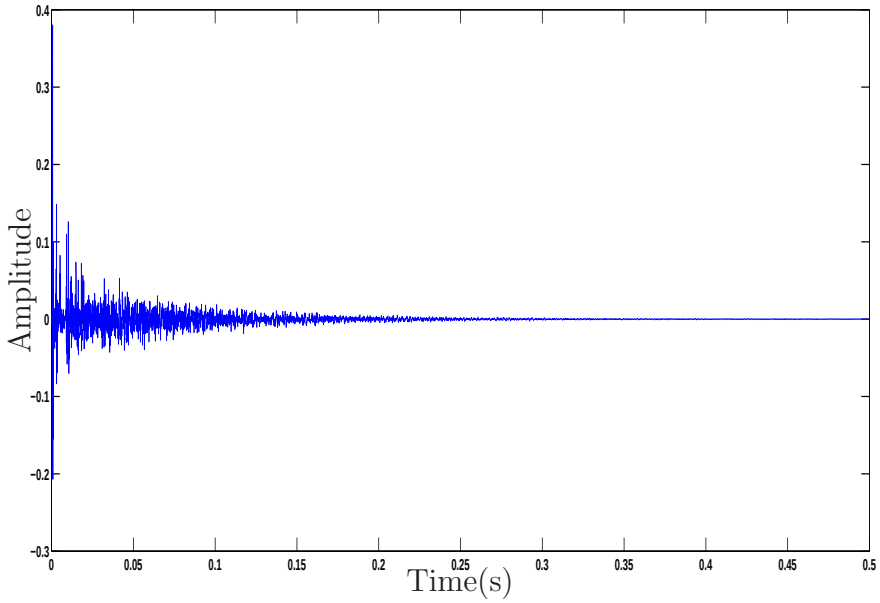


Figure 3.2: Room Impulse Response of a room with reverberation time $T_{60} = 600$ms

The initial short period of approximately zero amplitude is the propagation delay of direct signal from speaker to microphone. The peak followed by this short

period correspond to the direct signal. The early reflections are often taken as the first 50 ms of the impulse response [20], and constitute well-defined impulses of large magnitude relative to the smaller magnitude and diffuse nature of the late reflections. The late reflections are considered as the tail of the impulse response. They are distributed randomly with closely spaced and decaying impulses. This tail provides the major contribution to what is generally perceived as reverberation [22].

The reverberant data can be obtained by recording the microphone signals in that particular environment used for the application procedure. Such a large training data can be used for the estimation of the HMM model parameters using the Baum-Welch algorithm (see section 2.3 for details) as this data represents the statistical properties of the reverberant FVSs in the respective environments. But, the disadvantage with this procedure is recording a large training data in each application environment.

To reduce the effort in collecting the data, the reverberant training data can be generated by convolving clean-speech training utterances with RIRs measured in the environment of application [13]. But, the disadvantage with this procedure is that all the features of the reverberant data, like the Lombard effect [19] and significant change in the acoustic path between speaker and microphone, due to change in position of the speaker, change in the temperature and other effects, cannot be captured.

The propagation of signal from the speaker's lips to the microphone can be represented by the convolution of the speech signal with the RIR as shown below [20].

$$x(n) = h(n) * s(n) + b(n), \tag{3.2}$$

where $s(n)$ denotes desired speaker signal, $h(n)$ the room impulse response, $x(n)$ the microphone signal, $b(n)$ additive noise and n discrete time index. In the following discussions, the additive noise term is neglected, as the main focus of this thesis is reverberation. Therefore, the microphone signal can be represented as

$$x(n) = h(n) * s(n), \tag{3.3}$$

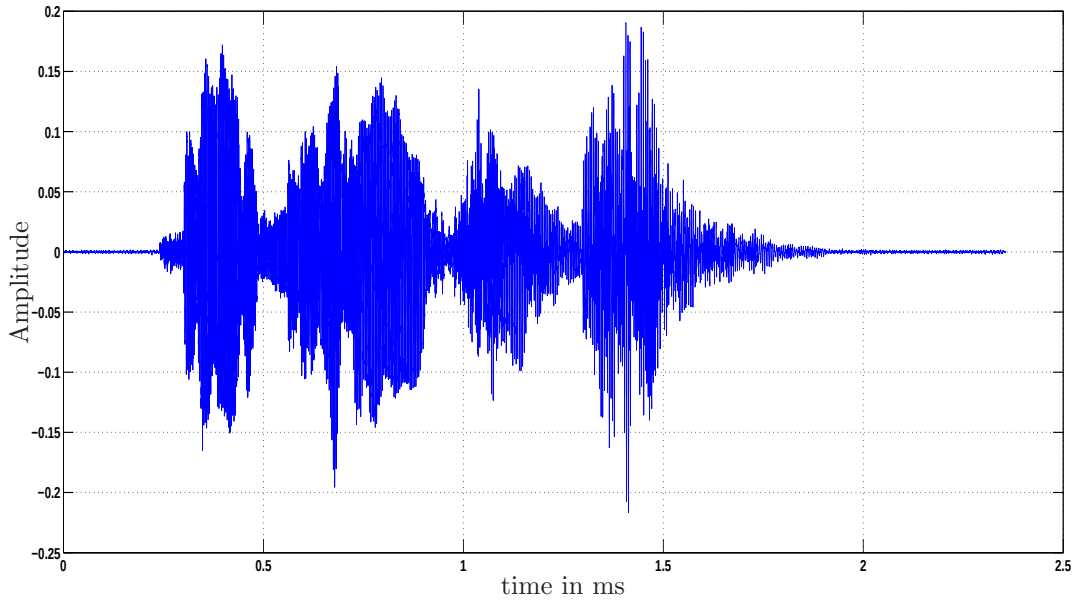


Figure 3.3: Microphone signal of utterance "two, three, oh, four" in a distant-talking scenario

3.3 Effects of Reverberation on ASR

When the speech signals received in a room by one or more microphones positioned at a distance from the speaker, the observed microphone signals consists of a superposition of many delayed and attenuated copies of the speech signal due to reverberation. The delay of the superimposed copies arises due to the fact that other propagation paths are longer than the direct-path and the additional attenuation occurs due to the frequency dependent absorption at each reflection [20].

When the reverberation effects are severe, the characteristics of the speech signal are altered, which in turn effects the speech recognition and significantly reduces the performance of algorithms developed without taking room effects into consideration. The destructive effects are magnified as the distance between the speaker and the microphones is increased.

Example: Fig 3.4 shows a plot of connected digit recognition experiment with increasing reverberation time T_{60}. For a HMM trained on clean speech, the reverberant test data

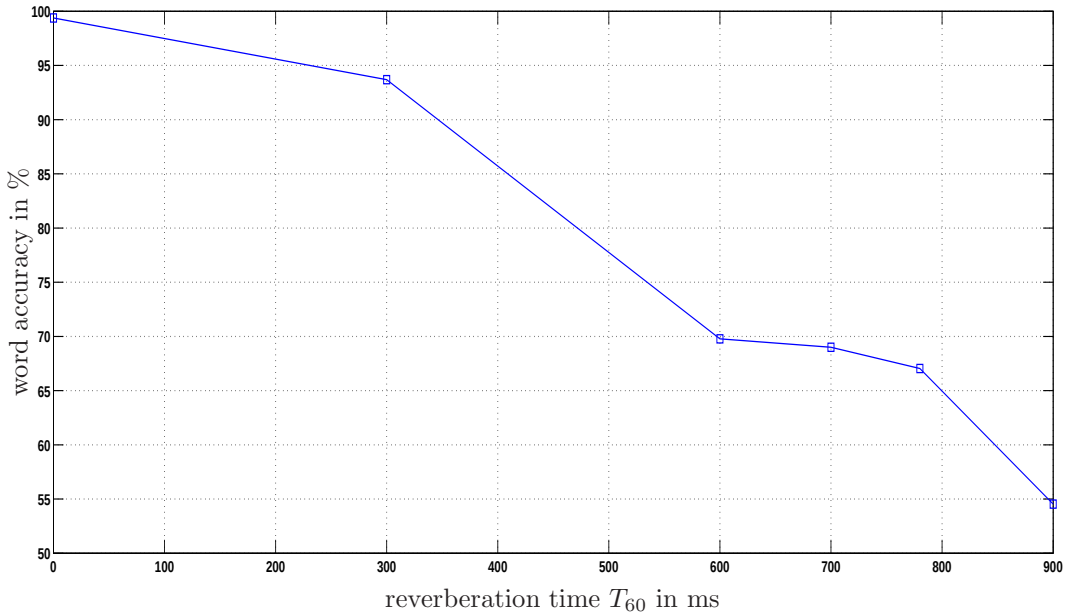


Figure 3.4: Word accuracy with increasing T_{60} for a HMM trained on clean data

is used to find the performance of the recognition system. The performance of the recognizer decreases as the reverberation effects increase.

3.4 Characteristics of Reverberation

In this section, characterization of a reverberant utterance in the time domain and logmelspec domain are discussed.

In the time domain, the reverberant speech signal can be modeled by convolving the clean-speech signal with RIR describing the acoustic path between speaker and microphone [20]. Figure 3.5 shows the time-domain representation of the microphone signal of the utterance "four, two, seven" in close-talking and distant-talking scenario. Figure 3.5 (b) shows the temporal smearing of the speech signal. Comparing Figure 3.5 (a) and 3.5 (b), it is noted that the reverberation of each phoneme extends to succeeding phoneme(s).

In STFT domain phonemes are more clearly distinguished. Figure 3.6 shows

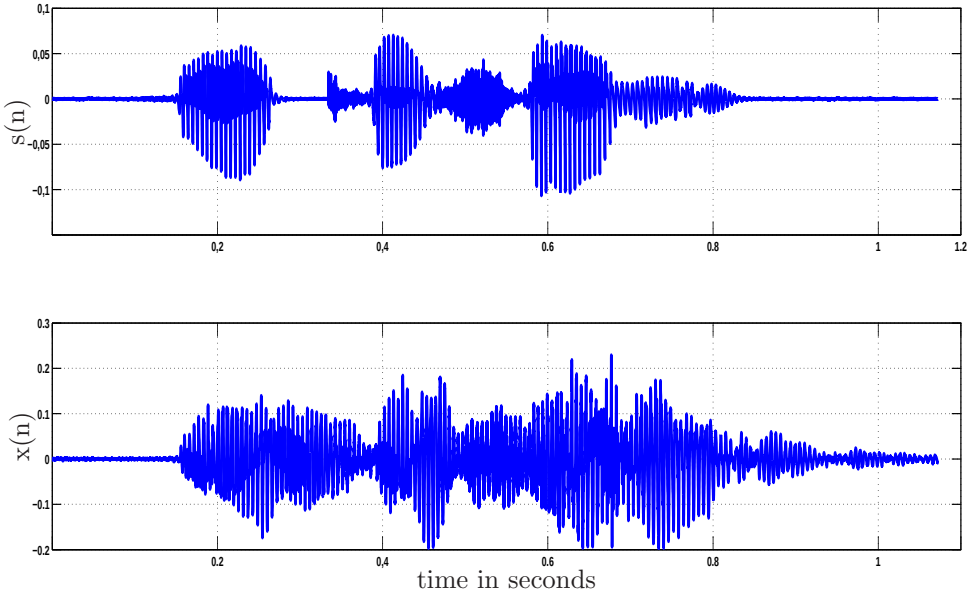


Figure 3.5: Time-domain signal of the utterance “four, two, seven” uttered by a female speaker a) close-talking recording, b) distant-talking recording in room B, $T_{60} = 700$ms, SRR = -4.0 dB loudspeaker/microphone distance 4m, $f_s = 20$ KHz.

the representation of the close-talking and distant-talking in STFT domain. Comparing 3.6 a) and 3.6 b), it is clearly seen that the harmonic spectogram of the vowel /ao/ fills the short pause before the plosive /t/.

The logmelspec domain and Δ coefficients representation of the utterance ”four, two, seven“ are shown in the Figure 3.7. The feature domain representation captures only the envelope of the speech signal’s time-frequency representation. Hence, useful representation for discriminating different phones, compared to the STFT representation containing the unnecessary information like pitch contour and other spectral details. Therefore, this representation is used in ASR.

Comparing 3.7 a) and 3.7 c), it is noted that the feature vectors are smeared across time in the reverberant case. For example, the short period of silence before plosive /t/ in ”two“ and the region of low energy during the fricative /s/ in ”seven“ are filled with the energies from the previous frame. This shows that, in the reverberant case, the current feature vector strongly depends on the previous feature vector. The

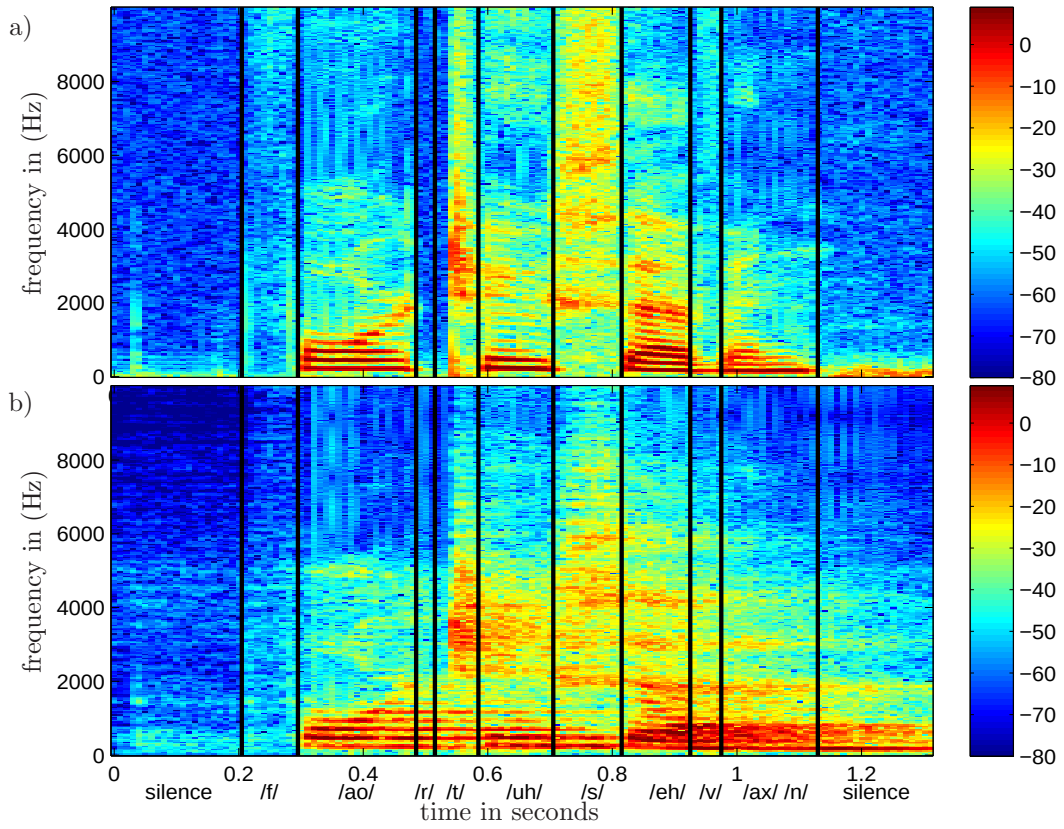


Figure 3.6: STFT representation of the utterance "four, two, seven" in dB color scale a) close-talking recording, b) distant-talking recording in room B, T_{60} = 700ms, SRR = -4.0 dB loudspeaker/microphone distance 4m, f_s = 20 KHz[10].

conditional independency assumption that the current feature vector is independent of the previous feature vectors for a conventionally used FO-HMM based recognizers, contradicts the above observation and therefore the performance of such recognizers decreases in the reverberant condition.

Figure 3.7 b) and Figure 3.7 d) are the logmelspec domain representation of Δ coefficients $\Delta s_l(k)$ and $\Delta x_l(k)$ with K_Δ=2. where $\Delta s(k)$ can be expressed by 2.7. These Δ coefficients capture the temporal changes of the logmelspec features. From the Figure 3.7 b), the following observations can be drawn. The Δ coefficients

- exhibit large positive values for the lower frequencies at the starting of vowels /ao/ and /eh/.

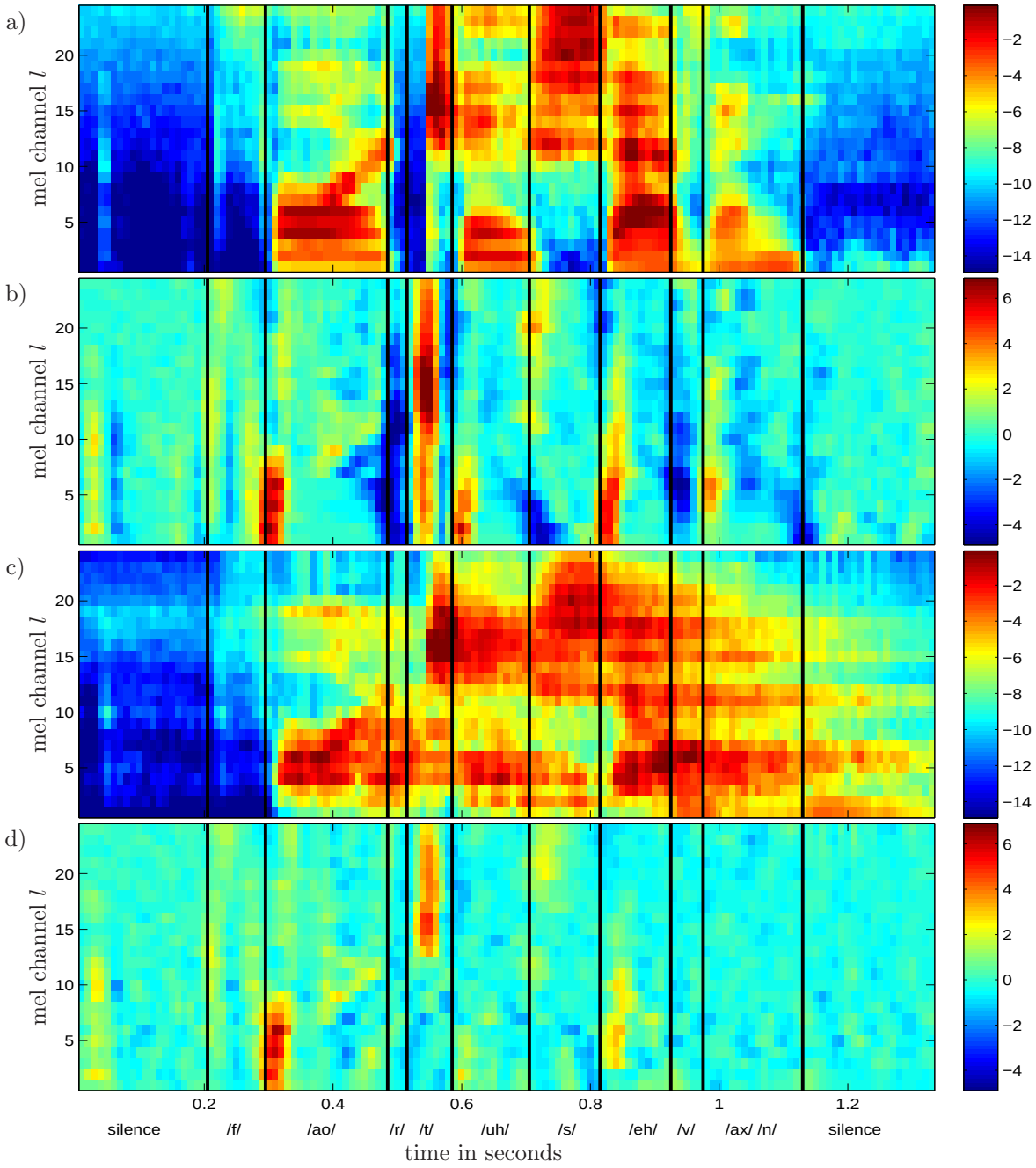


Figure 3.7: Logmelspec representation of the utterance “four, two, seven”,a) static features of close-talking recording, b) delta features of close-talking recording, c) static features of distant-talking recording with a loudspeaker/microphone distance 4 m, d) delta features of distant-talking recording[10].

- take close to zero values during the nearly stationary voiced articulation of the vowels.

- assume large negative values for rapid decrease of the sound energy. For example, at the end of the vowels and short period of silence before plosive /t/.

- assume large positive values across the entire range of frequencies, due to the plosive burst /t/.

However, due to the smearing effect of reverberation, there are no abrupt changes in the energy and therefore, the Δ coefficients do not take large negative values, compared to the clean speech values. This can be observed in the Figure 3.7 d), the magnitudes of Δ coefficients after the vowels /ao/,/uh/ and /eh/ in the reverberant case are lower than the clean case.

Due to the dispersion caused by reverberation, the shapes, the means, and the variance of the PDFs describing the reverberant features are changed in comparison to the PDFs of the corresponding clean-speech features [10]. Furthermore, the inter-frame correlation between reverberant feature vectors is significantly higher than that between clean-speech feature vectors [10].

3.5 Model Based Approaches

From the discussion in section 3.4, it is clear that the reverberation has a dispersive effect on FVSs and their statistical properties change significantly compared to clean speech FVSs. Hence, if a reverberant utterance is tested on an acoustic model trained on a clean speech data, there is a mismatch and to avoid that, the acoustic models have to be adjusted to reverberation in a distant talking scenario. In this section, model-based approaches, estimating the parameters of a HMM with reverberant training data, exploiting the inter-frame dependency using conditional HMMs and a novel approach for matched reverberant training of HMMs using data pairs are briefly described.

3.5.1 Estimation of Parameters of a HMM with Reverberant Training Data

The HMMs can be trained using reverberant data, as explained in section 3.2. The performance of HMMs trained on artificially reverberated training data is only slightly lower than that of HMMs trained on data recorded in the application environment [15]. Synthetically generated RIRs can be used instead of measured RIRs in generating the training data [16], reducing the effort to record reverberant data. In each new application environment, to avoid retraining the HMM, synthetically generated RIRs for different reverberation parameters are used to train the HMMs in advance and the HMM, that best suits the environment is selected [17].

3.5.2 Conditional HMMs

The conditional HMMs approach uses conditional output densities $f_{S(k)|S(k-1)=s_1,Q(k)=j}(s)$ instead of $f_{S(k)|Q(k)=j}(s)$ [10]. By using these conditional output densities on the previous frames, the conditional independency assumption can be overcome and the inter-frame dependencies can be modeled explicitly by HMMs [10]. The Baum-Welch training and Viterbi training formulae for determining the parameters of single-Gaussian conditional densities are derived in [18]. For Viterbi decoding, the conditionally independent output density $f_{S(k)|Q(k)=j}(s)$ is replaced by conditionally dependent output density on previous feature vector $f_{S(k)|S(k-1)=s_1,Q(k)=j}(s)$ [18].

3.5.3 Retraining a First-Order HMM

Information Combining Estimation With Non-reverberant Data (ICEWIND) is a novel approach, tailored particularly to training HMMs with stereo data consisting of clean and reverberated feature vectors [9]. In this algorithm, the temporal structure of speech is determined by hard aligning the clean-speech training data to the states of a well trained clean-speech HMM λ_s (Viterbi alignment). This State Frame Alignment (SFA)

is the optimum one that can be achieved, as the clean-speech signal is the actual source of information providing the most accurate picture of the temporal signal structure. The standard EM algorithm is applied to determine the parameters of the Gaussian-mixture densities to the predetermined training data of each state. The ICEWIND algorithm can be summarized as follows [9]:

- Step 1: Determine the SFA by hard-aligning $s(1:K)$ to λ_s:

$$\hat{q}(k) = \arg\max_j P(Q(k) = j | s(1:K), \lambda_s) \tag{3.4}$$

  If HMMs for several different reverberation conditions with identical clean-speech data are trained, the SFA has to be performed only once and can then be used for all conditions. Therefore, the state transition probabilities of the clean-speech HMM λ_s are simply copied to the reverberant HMM λ_x [9].

- Step 2: Determine the parameters of the Gaussian-mixture density for each state j applying the standard EM algorithm to the reverberant data [9]:
  a) E-step: Calculate the posterior mixture probability for each frame k and each mixture component m:

$$\gamma_{jm}(k) = P(R(k) = m | x(k), \hat{q}(k) = j, \lambda_x) = \frac{w_{jm}\mathcal{N}(x(k)|\mu_{jm}, C_{jm})}{\sum_{m'=1}^{M} w_{jm'}\mathcal{N}(x(k)|\mu_{jm'}, C_{jm'})} \tag{3.5}$$

$$\gamma_{jm} = \sum_{k=1}^{K} \gamma_{jm}(k) \tag{3.6}$$

  b) M-step: Estimate the mixture density parameters for each component m:

$$\hat{w}_{jm} = \frac{1}{K}\gamma_{jm} \tag{3.7}$$

$$\hat{\mu}_{jm} = \frac{1}{\gamma_{jm}} \sum_{k=1}^{K} \gamma_{jm} x(k) \tag{3.8}$$

$$\hat{C}_{jm} = \frac{1}{\gamma_{jm}} \sum_{k=1}^{K} \gamma_{jm}(x(k) - \hat{\mu}_{jm})(x(k) - \hat{\mu}_{jm})^T \tag{3.9}$$

where w_{jm}, μ_{jm}, and $\hat{C}_{jm}$ are the weight, mean vector, and covariance matrix of component m for state j, respectively, and $Q(k)$ is a random process of state indices.

The HMM parameters are updated with the estimates from step 2 so that a new parameter set λ_x is obtained, which is used for the following iteration. The steps a) and b) are repeated until some termination condition is fulfilled [9].

.

Chapter 4

Combined-Order HMM

In this chapter, first a brief introduction to Higher-Order HMMs (HO-HMMs) is given and the concept of CO-HMMs is explained. In the next following sections, the training and the recognition procedures are explained in detail.

4.1 Higher-Order HMM

In a FO-HMM the two fundamental assumptions as mentioned in section 2.3, the Markov assumption, which states that the current state q_t depends only on the previous state $q_{(t-1)}$, and not on the earlier states and the conditional independence assumption, which states that the output feature vector o_t for a given state q_t is independent of all the previous states and the output feature vectors. Figure 4.1 shows a FO-HMM.

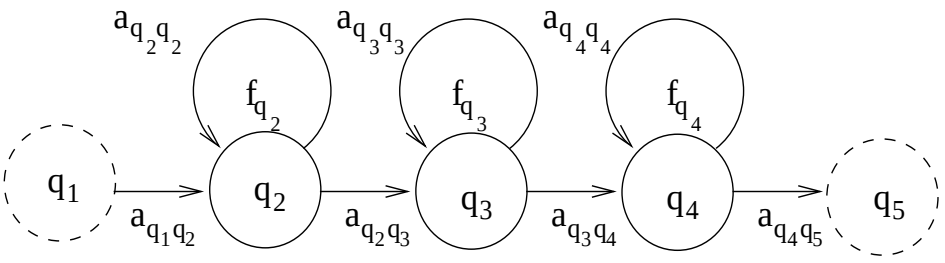


Figure 4.1: FO-HMM

In a HO-HMM, the Markov assumption is extended to k previous states i.e., a k^{th} order HMM assumes that the current state q_t depends on the k previous states $q_{(t-1)}$, $q_{(t-2)} \cdots\ q_{(t-k+1)}$, similarly the conditional independence assumption extends to k previous states. The Markov assumption assumption and the conditional independence assumption for a HO-HMM can be mathematically written as follows

$$a_{i_1 i_2 ..... i_k j} = P(q_{t+1} = j | q_t = i_1, q_{t-1} = i_2, ..., q_{(t-k+1)} = i_k) \tag{4.1}$$

$$P(O|q_1, q_2, ..., q_T, \lambda) = \prod_{t=1}^{T} P(o_t | q_t, q_{t-1} ... q_{(t-k+1)}, \lambda) \tag{4.2}$$

Figure 4.2 shows a second-order HMM.

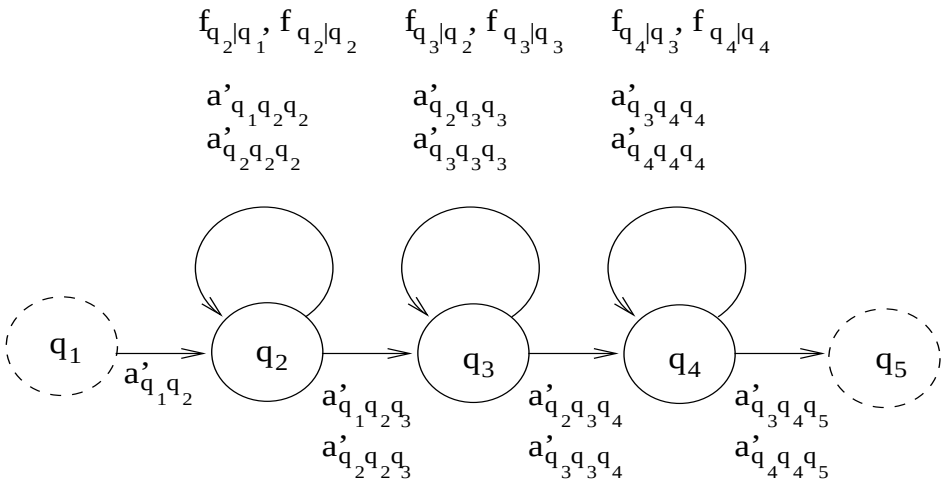


Figure 4.2: second-order HMM

4.2 Combined-Order HMM

From section 3.5, it is noted that reverberation has a dispersive effect on FVSs and due to this dispersive effect, the inter-frame correlation between FVSs is increased. This observation forms the basis for this concept called CO-HMM. To capture the inter-frame dependency and to build a reverberant robust speech recognizer, the concepts

of first and second-order HMMs are merged to form a "combined-order" HMM (CO-HMM).

In a CO-HMM the Markov assumption remains same as the first-order (2.8), but the conditional independence assumption is changed such that the observation feature vector o_t is dependent on the previous state. The conditional independence assumption of a CO-HMM can be stated as follows

$$P(O|q_1, q_2, ..., q_T, \lambda) = \prod_{t=1}^{T} P(o_t|q_t, q_{t-1}, \lambda) \tag{4.3}$$

where $O = o_1, o_2, o_3....., o_T$ and λ represents the model parameters.

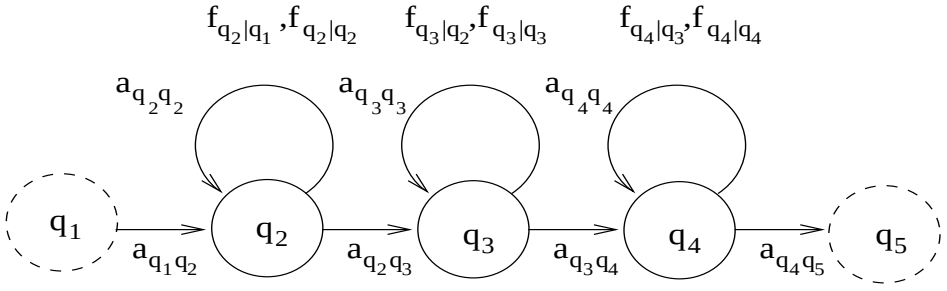


Figure 4.3: CO-HMM

From Figure 4.3, it can be observed that each state of a CO-HMM is associated with two predecessor dependent PDFs, Transition-State (TS) PDF $f_{qi|qj}$ and Steady-State (SS) PDF $f_{qi|qi}$. A TS PDF is defined as the output PDF of all the feature vectors whose predecessor belongs to previous state. A SS PDF is defined as the output PDF of all the feature vectors whose predecessor belongs to the same state.

4.3 Training a CO-HMM

Training a CO-HMM involves two main steps. First, a conventional FO-HMM is trained by the Baum-Welch method (section 2.3.1). Second, the HMM obtained from step one is used as input to the ICEWIND approach (section 3.5.3), to train a CO-HMM. Figure 4.4 shows the flow graph of the CO-HMM training procedure.

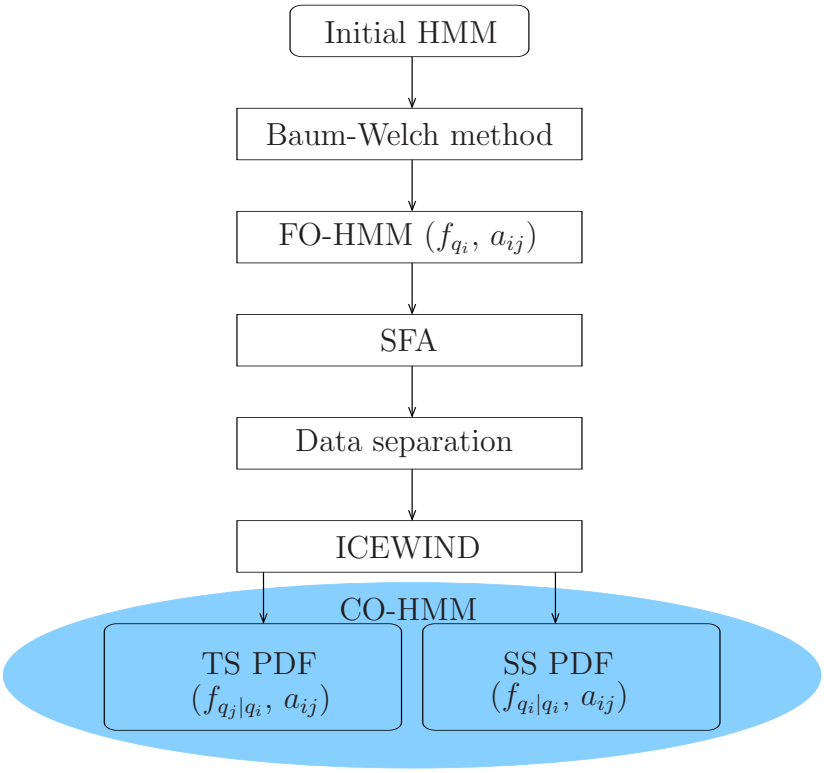


Figure 4.4: Algorithm to train a CO-HMM

In the ICEWIND approach the first step to determine the SFA remains same (refer section 3.5.3), which is shown in Figure 4.5.

However, in the second step finding the parameters of the Gaussian-mixture density for each state j applying the standard EM algorithm to the reverberant data has to be repeated twice. Ones for estimating TS output PDF and ones for SS output PDF.

The main difference in training a FO-HMM and CO-HMM using ICEWIND is that the sequence of observation feature vectors are split in two depending on its predecessor. All feature vectors with predecessor belonging to the same state are used to estimate the SS output PDF $f_{qi|qi}$. Similarly, all the feature vectors with predecessor belonging to another state are used to estimate the TS output PDF $f_{qi|qj}$. Figure 4.6 shows an example of this procedure.

This training procedure involves in a separation of the whole training set into two. Therefore, there is a decrease in the number of training samples for estimating

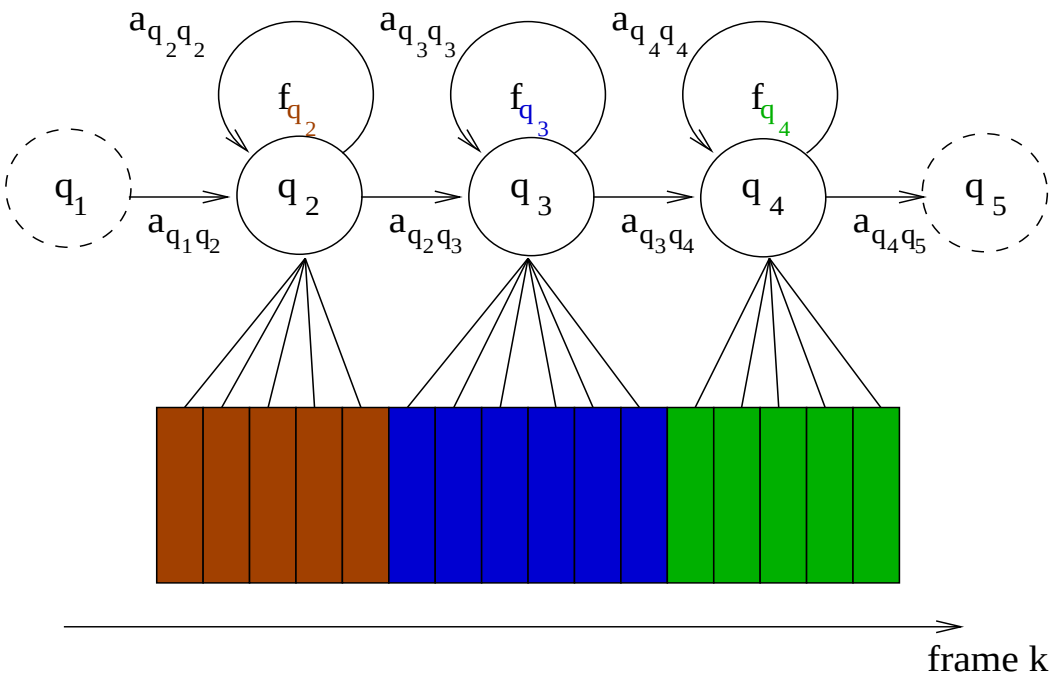


Figure 4.5: State Frame Alignment

the predecessor-dependent output PDFs for each state. If the number of samples are very less, then the estimated PDF is not accurate. One way to overcome this problem, which is applied in this thesis, is to decide a certain number of threshold samples N and if the number of training samples to estimate the PDF fell below this N, then all the samples are used to estimate one PDF used as TS and SS PDF.

In estimating the predecessor dependent PDFs, at the word boundaries there is no unique preceding state. Hence, it is assumed that for the first state, the preceding state could be the last emitting state from all the available models. Figure 4.7 shows the procedure at the word boundaries.

4.4 Recognition

A standard algorithm used for recognition in a FO-HMM is the Viterbi algorithm, which finds the maximum likelihood state sequence for a given sequence of observations. However, for a CO-HMM every state has the output probabilities depending upon its

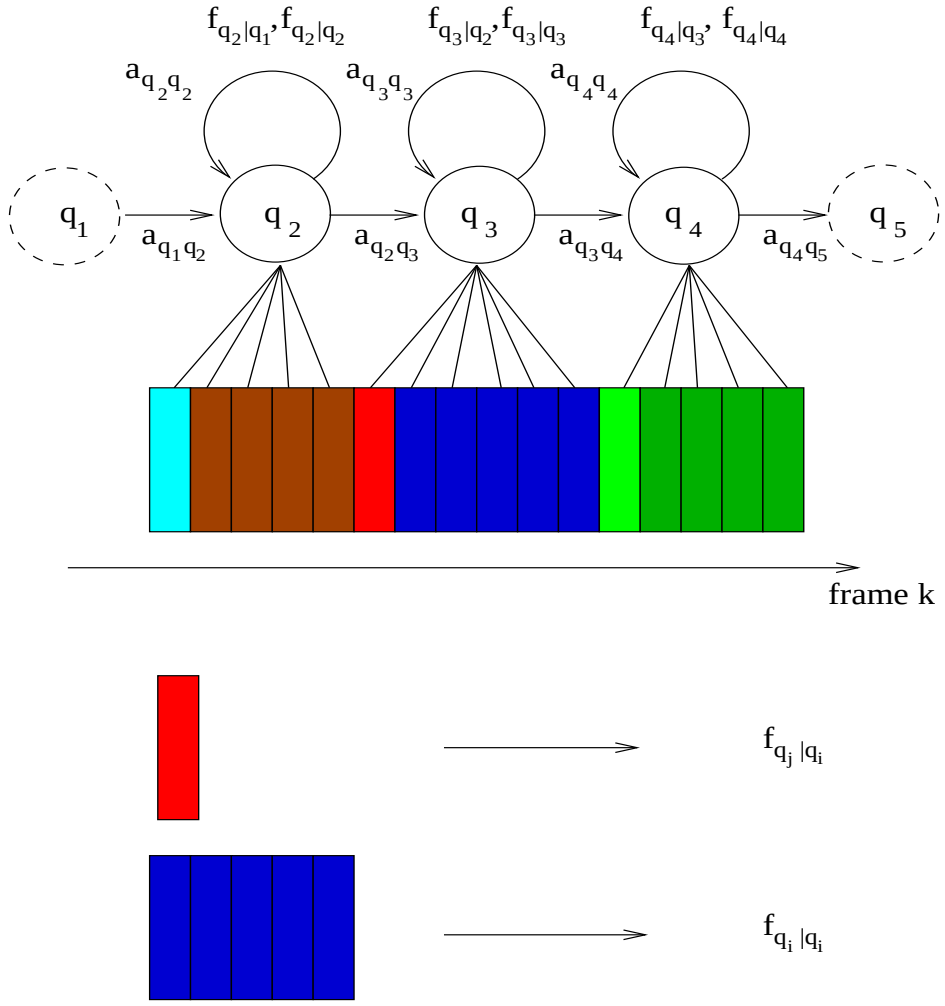


Figure 4.6: Separation of the PDF data in training a CO-HMM

predecessor. Hence, this algorithm has to be adapted accordingly.

For a given model M, let $\phi_j(t)$ represent the maximum likelihood of observing speech vectors o_1 to o_t and being in state j at time t. The partial likelihood for the CO-HMM is computed recursively using the adapted Viterbi algorithm.

Let us assume that the HMM starts in state 1 and ends in the last state N at the final frame T of the sequence $o_{(1:T)}$. Then the partial likelihood for the CO-HMM can be represented as follows

$$\phi_j(t) = \max_i\{\phi_i(t-1)\, a_{ij}\, b_{ij}(o_t)\}, \tag{4.4}$$

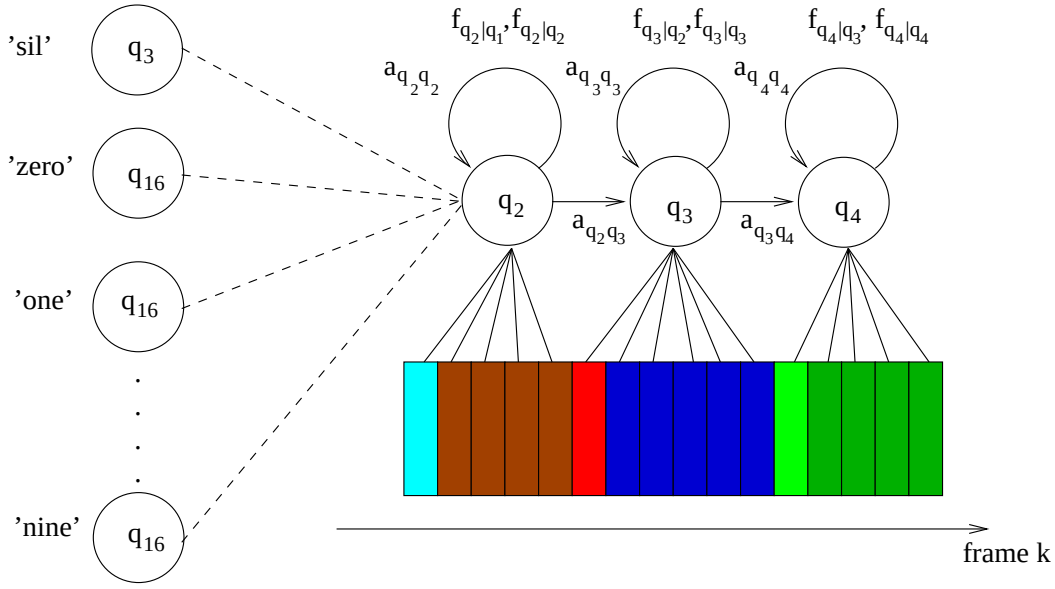


Figure 4.7: CO-HMM Boundaries

where the initial likelihood of observing speech vector o_1 and being in state 1 at time 1 is given by

$$\phi_1(1) = 1 \tag{4.5}$$

the likelihood of observing speech vector o_1 and being in state j at time 1 is given by

$$\phi_j(1) = a_{1j}\, b_{1j}(o_1) \tag{4.6}$$

for $1 < j < N$. The maximum likelihood state sequence representing a model M i.e., $\hat{P}(O|M)$, is given by

$$\phi_N(T) = \max_i \phi_i(t)\, a_{iN}\, b_{iN}(o_T) \tag{4.7}$$

Here, j is the current state and i represents all the previous states leading to j, $b_{ij}(o_t)$ represents the output probability of a observation vector at time t depending on its current and predecessor state, a_{ij} represents the transition probability from state i to state j.

This algorithm can be explained in detail using a trellis diagram as shown in Figure 4.8. For example, let us consider the recognition process using a five state

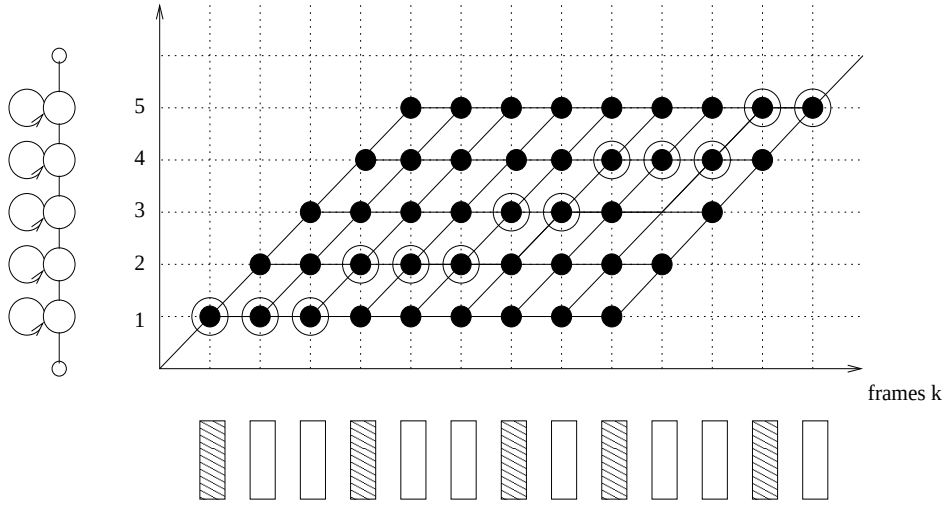


Figure 4.8: Adapted Viterbi algorithm for CO-HMM illustrated using Trellis diagram

HMM with words as recognition units, such that there are no skips between the states. The transition probability from initial non emitting state to first state is one and the likelihood of observing speech vector o_1 and being in first state is one.

From equation (4.4) the Viterbi scores are calculated step by step recursively by multiplying the score of the possible predecessor states with the corresponding transition probability and the output probability of the current feature vector, by selecting the transition probability, which is maximum among all predecessors and the maximum output probability among the predecessor dependent output probabilities. Hence, in recursion finding $\phi_5(T)$ gives the final acoustic score for a given CO-HMM model and backtracking matrix stores the most likely state sequence through the HMM.

Chapter 5

Experiments

The performance of CO-HMMs, is evaluated by Connected Digit Recognition (CDR) experiments based on TI digit corpus. The CDR task is selected so that the recognition rate does not depend on the language model and role of the acoustic model completely determines the performance. In this thesis, through CO-HMM, an attempt to improve the acoustic model is proposed. Hence, this task perfectly gives the environment for evaluating the performance of CO-HMM.

This chapter is organized as follows: The experimental setup, including the recognition system, the acoustic environment, the description of train and test data, different recognition units and parameters used are explained in section 5.1. In section 5.2, comparisons of different PDFs of FO-HMM and CO-HMM for different recognition units are discussed. The experimental results for approaches FO-HMM and CO-HMM are compared in section 5.3.

5.1 Experimental Setup

5.1.1 Baseline Recognition System

The HMM-Tool-Kit (HTK) is used for the CDR experiments. It is a software programming, that provide tools for different stages of ASR (data preparation, training,

recognition and analysis) in such a way that they are used together to construct and test HMM-based recognizers.

Data Preparation Tools:

The tool HCopy is used to convert the wav file into its parametric form. The 13 MFCC features are extracted from the process, shown in Figure 2.3, by decomposing the microphone signal into overlapping frames of length 25ms with a frame shift of 10ms by Hamming window. In this thesis, 13 MFCCs and 13 Δ Coefficients with Cepstral Mean Subtraction (CMS) are used. The dynamic coefficients Δ are calculated according to (2.7) and added to the extracted 13 MFCCs, so the length of the feature vector is 26.

Apart from that, the tools HList, HLEd, HLStats, HQuant are used to check contents of speech, to create and edit label files, to display statistics on label files and to build a VQ codebook respectively.

Training Tools:

Initially, once a prototype HMM to specify the overall characteristics and topology of the HMM is defined, the actual parameters are calculated by training tools. An acceptable and simple strategy for choosing the initial probabilities is to make all of the transitions out of any state equally likely [1].

Once an initial set of models has been created by HInit, HRes, if bootstrap data (the location of sub-word boundaries have been marked) is available and HCompV, if bootstrap data is not available, the tool HERest is used to perform embedded training using entire training set to perform single Baum-Welch re-estimation of the whole set of HMM phone models simultaneously [1].

Recognition Tools:

Hvite is the recognition tool used in HTK to perform Viterbi-based speech recognition using token passing algorithm. For the recognition a network describing the allowable word sequences, a dictionary defining how each word is pronounced and a set of HMMs are given as input [1].

Apart from that, HBuild, HParse, HDMan tools are used to generate word loops, to convert higher level grammar notation to equivalent word network notation and to generate dictionary respectively.

Analysis Tools:

In the analysis part, the performance of the recognizer is evaluated by HResults tool by matching the recognizer output with the correct reference transcriptions. It is a dynamic programming to align the two transcriptions and then count substitution, deletion and insertion errors [1].

Performance Measure: Speech recognition performance is measured by word accuracy, given by the formula

$$word\ accuracy = \frac{N_w - N_D - N_S - N_I}{N_W}.100\,\% \tag{5.1}$$

Where N_W is the total number of words in the reference transcription, N_S is the number of substitutions, N_D is the number of deletions, N_I is the number of insertions compared to the reference transcription.

The Word Error Rate (WER) is defined as

$$WER = \frac{N_D - N_S - N_I}{N_W}.100\,\% = 100\,\% - word\ accuracy \tag{5.2}$$

5.1.2 Acoustic Environment

The experiments are performed using RIRs measured in five different rooms, whose characteristics are defined in Table 5.1.

Table 5.1: characteristics of Application Environments

Room	mmr	jr1	ofc	il412	lr400
T_{60}	300ms	600ms	780ms	700ms	900ms
SRR	+4	0.5	-0.5	-4	-4

Two disjoint sets of RIRs are prepared by measuring RIRs for different loudspeaker and microphone positions. One of the sets is used for training and other for testing. This is to create different reverberation conditions for testing and training. Figure 5.1 shows RIRs of different rooms used as application environment in this thesis.

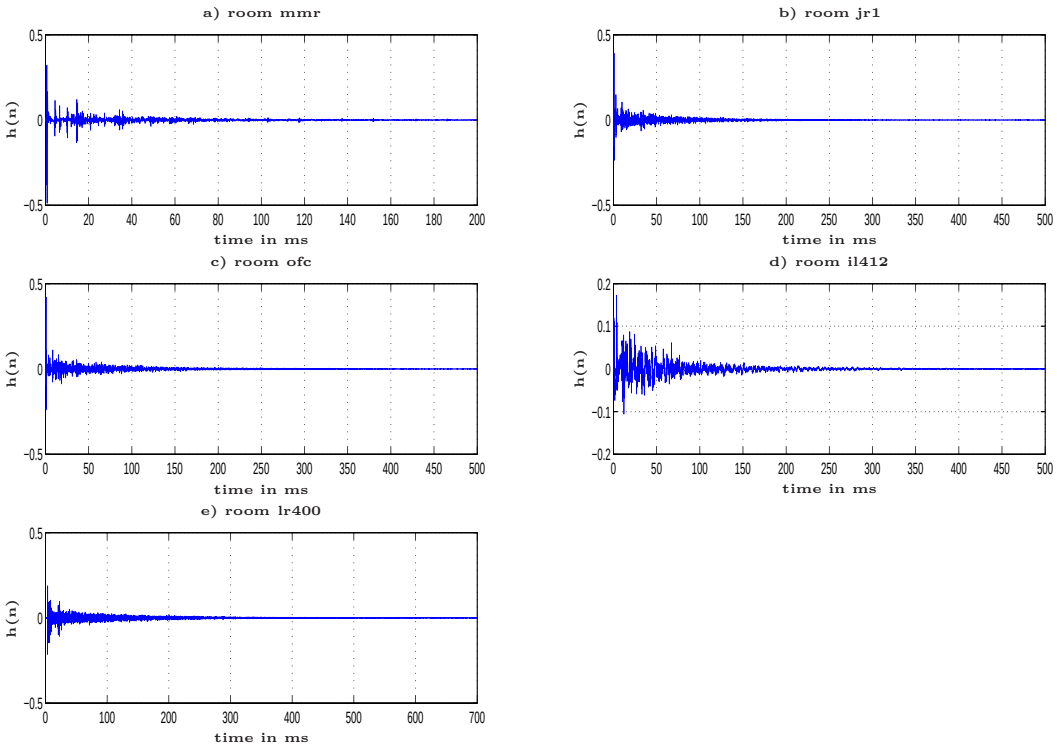


Figure 5.1: RIRs of different rooms with increasing T_{60}

5.1.3 Train and Test Data

The TI digits corpus [25] is used both for testing and training.

Train data:

A subset of the TI digits training set with 4579 connected digit utterances, corresponding to 1.5 hours of speech, is used for training. By convolving the clean-speech signal $s(n)$ from the training set with RIRs randomly selected from the RIR training set of the corresponding room according to Table 5.1, the reverberant signal $x(n)$ is obtained. From these signals, the stereo data $s(k)$, $x(k)$ are obtained by feature extraction using HTK.

Test Data:

A subset of 513 utterances randomly selected from the TI digits test set, corresponding to approximately 16 minutes of speech, is used for test. To obtain the reverberant test data, the clean data are convolved with RIRs randomly selected from the RIR test set of the corresponding rooms. By changing the RIRs for each utterance, the time-variance of the acoustic path between speaker and microphone is simulated.

5.1.4 Recognition Units

Word model:

A 16-state word-level HMM is used for each of the 11 digits
$'one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'zero', 'oh'$

For the first-order word-level HMM, the Baum-Welch training with 20 re-estimation iterations are performed with the HTK tool HERest, where a split of mixture components is performed after 10 and 15 iterations to go from 1 to 2 components and to go from 2 to 3 components, respectively.

Phoneme model:

A 3 state phoneme-level HMM is used for each of the 18 phonemes
$'f','k','n','r','s','t','v','w','z','ah','ay','eh','ey','ih','iy','ow','th','uw'$

For the first-order phoneme-level HMM, the Baum-Welch training with 18 re-estimation iterations are performed with the HTK tool HERest, where a split of mixture components is performed after 8 and 14 iterations to go from 1 to 2 components and to go from 2 to 3 components, respectively.

Triphone model:

A 3 state triphone-level HMM is used for each of the 33 triphones
$'w+ah','w-ah+n','ah-n','th+r','th-r+iy','r-iy','ey+t','ey-t','f+ow',$
$'f-ow+r','ow-r','ow','t+uw','t-uw','n+ay','n-ay+n','ay-n','z+iy',$
$'z-iy+r','iy-r+ow','r-ow','f+ay','f-ay+v','ay-v','s+ih','s-ih+k',$
$'ih-k+s','k-s','s+eh','s-eh+v','eh-v+ih','v-ih+n','ih-n'$

For the first-order triphone-level HMM, the Baum-Welch training with 25 re-estimation iterations are performed with the HTK tool HERest, where a split of mixture components is performed after 15 and 20 iterations to go from 1 to 2 components and to go from 2 to 3 components, respectively.

Additionally, a three-state silence model (sil) with a backward skip from state three to state one and a single state short pause model (sp)is used.

For the CO-HMM, the FO-HMM trained with Baum-Welch iterations is used as the starting point. Then ICEWIND approach is used to finally train the CO-HMM.

5.2 Differential Entropy as Sharpness Measure for GMMs

Entropy is a measure of the uncertainty associated with a random variable, which quantifies the expected value of the information contained in the realization

of a specific random variable [4]. This term usually refers to the Shannon entropy. If p denotes the probability mass function (function that gives the probability that a discrete random variable is exactly equal to some value.) of a discrete random variable $X = \{x_1, ..., x_n\}$, the entropy can be written as

$$H(X) = -\sum_{i=1}^{n} p(x_i)\ \log(p(x_i))dx \tag{5.3}$$

The differential entropy (also referred to as continuous entropy) is a concept in information theory that extends the idea of (Shannon) entropy, a measure of average surprisal of a random variable, to continuous probability distributions [12]. It can be expressed as

$$H(X) = -\int_X f(x)\ \log(f(x))dx \tag{5.4}$$

where X is a continuous random variable. $H(X)$, $f(x)$ are the entropy and PDF of the random variable X. Here the discussion is confined to Gaussian Mixture Models (GMMs). The integral of the differential entropy of a GMM can be approximated by sampling the x-axis, i.e., numerical integration. Therefore, 5.3 can be approximated as

$$H(X) = -\sum_{i=1}^{n} f(x_i)\ \log f(x_i)\Delta x, \tag{5.5}$$

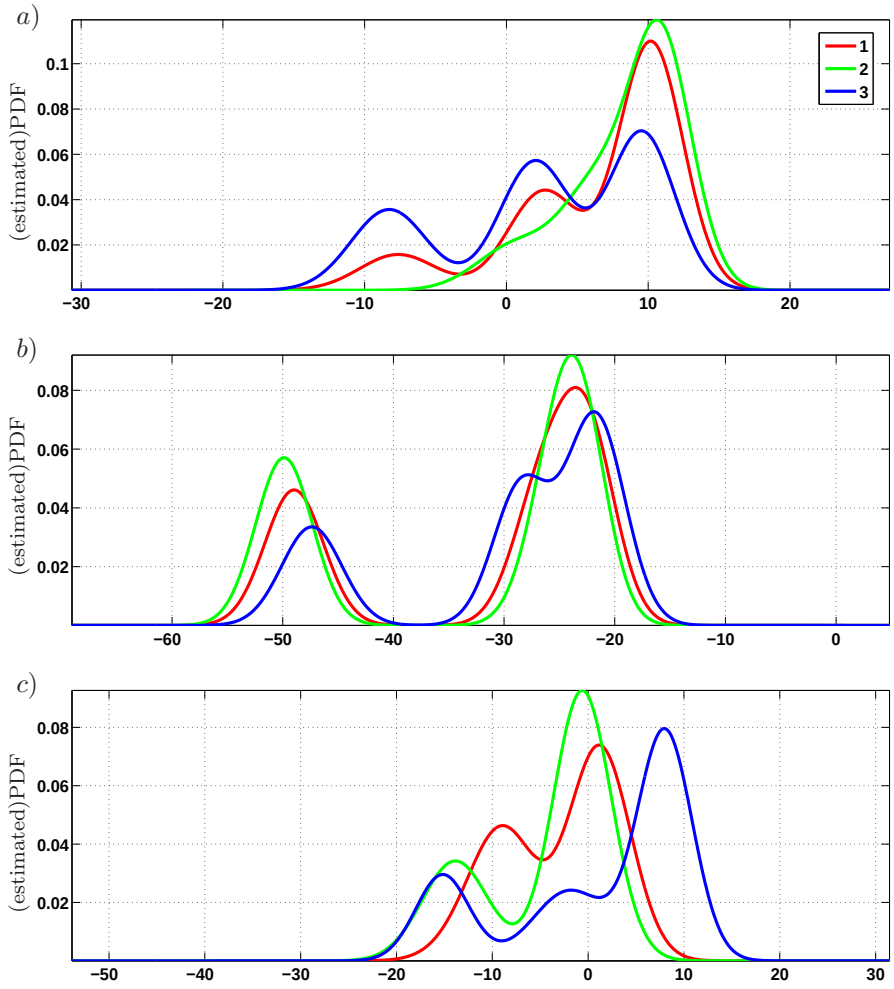
where $x_i = x_{min} + i.\Delta x$, Δx is the step size.

The differential entropy is referred to as entropy in the following discussions. This entropy can be used as the sharpness measure of estimated GMMs. If the entropy of a GMM is less, then the uncertainty associated with the random variable is also less and this model can give a sharper estimation than the model with higher entropy.

5.3 Comparison of Statistical Properties

From section 3.4, it has been shown that reverberation has a substantial effect on the statistical properties of FVSs. This section investigates how the proposed concept captures these changed statistical properties of reverberant FVSs, by comparing the

acoustic model of CO-HMM and FO-HMM (model based approach, ICEWIND according to section 3.5.1). These two models are compared because the procedure involved



1. PDF of FO-HMM, 2. SS PDF of CO-HMM, 3. TS PDF of CO-HMM

a) Room 'lr400', model 'three',state 16, MFCC channel 1.

b) Room 'lr400', model 'eh', state 1, MFCC channel 1.

c) Room 'lr400', model 'z+iy', state 1, MFCC channel 1.

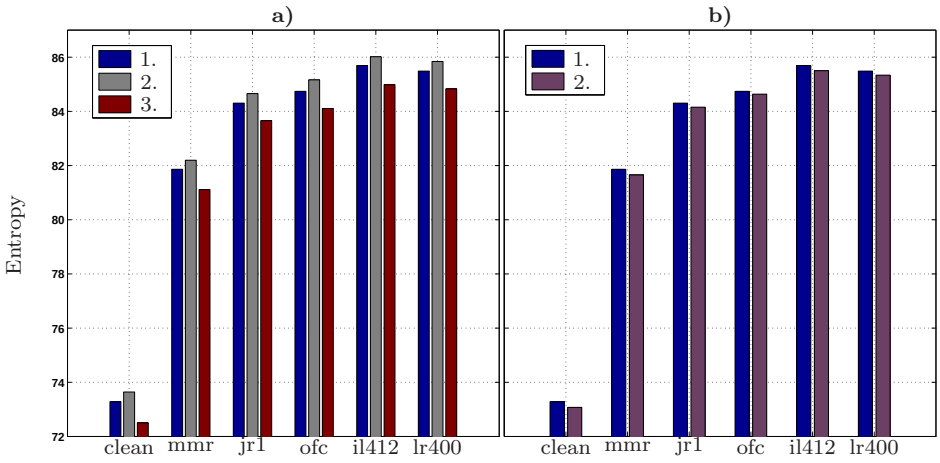
Figure 5.2: Comparison of PDFs of FO-HMM and CO-HMM.

in training both the models is the same.

The main motive of this thesis is to capture the inter-frame correlation of the feature vectors by CO-HMM. The TS output PDF has to capture the statistical properties of those feature vectors that have a higher diversity reverberation, where

as the SS output PDF has to capture the statistical properties of the feature vectors, which have a lower diversity reverberation.

Figure 5.2 shows the PDFs (GMMs, 3-mixtures) of FO-HMM and CO-HMM for different recognition units. The entropy associated with each GMM is calculated using 5.5. The TS output PDF of a CO-HMM has a higher entropy compared to output PDF of FO-HMM, as the TS output PDF models the feature vectors with higher diversity reverberation. The entropy of SS output PDF is less than that of PDF of FO-HMM, as the SS PDF models the feature vectors with lower diversity reverberation.



1.FO-HMM.

2.TS CO-HMM.

3.SS CO-HMM.

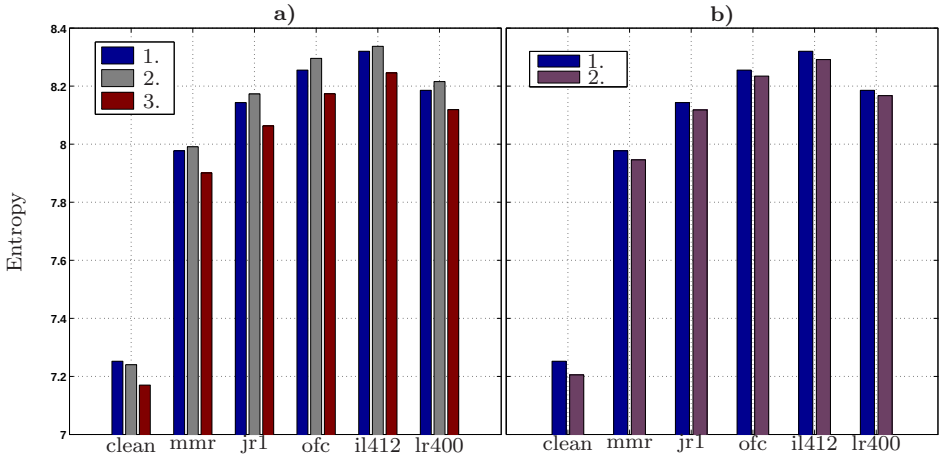
1.FO-HMM.

2.Average of SS and TS CO-HMM.

Figure 5.3: Global Entropy of the word-level HMM for different rooms

Figure 5.3, 5.4, 5.5 show, the global entropy of FO-HMM and CO-HMM for different acoustic environments and different recognition units. In the Figures 5.3 a), 5.4 a), 5.5 a), though the entropy of TS state is more than the PDF of FO-HMM, the average entropy of TS and SS PDF has a lower entropy than PDF of FO-HMM, which is shown in the Figure 5.3 b), 5.4 b), 5.5 b) for word, phoneme and triphone models, respectively.

These examples illustrate that the predecessor dependent output PDFs can



1.FO-HMM.

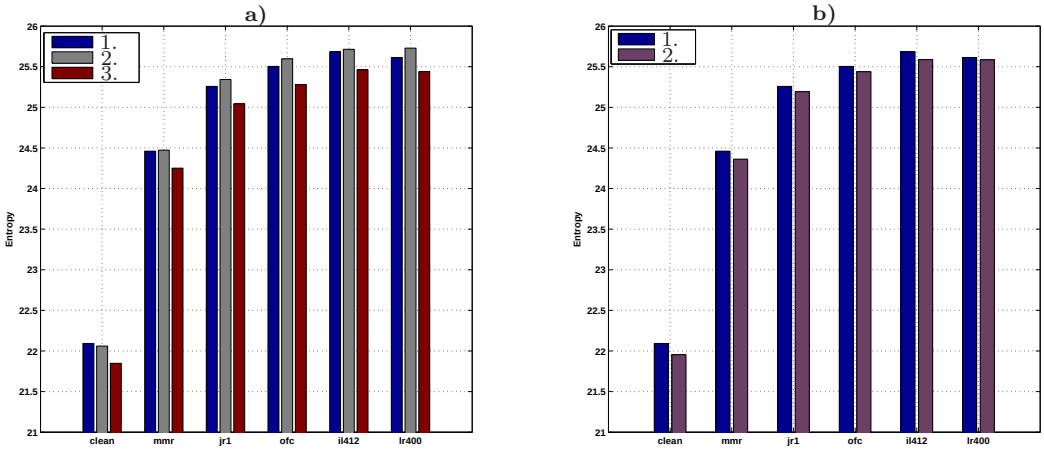
2.TS CO-HMM.

3.SS CO-HMM.

1.FO-HMM.

2.Average of SS and TS CO-HMM.

Figure 5.4: Global Entropy of the phoneme-level HMM for different rooms



1.FO-HMM.

2.TS CO-HMM.

3.SS CO-HMM.

1.FO-HMM.

2.Average of SS and TS CO-HMM.

Figure 5.5: Global Entropy of the triphone-level HMM for different rooms

model reverberant speech more accurately than the conditionally independent output PDF. The average lower entropy of the CO-HMM compared to FO-HMM suggests that, the proposed model provides a sharper model when compared to the conventional model. Thus, an increased discrimination capability resulting in lower WERs can be expected if predecessor-dependent PDFs are used.

5.4 Experimental Results

In this section, the recognition results for different reverberant conditions and different recognition units are displayed and compared.

Table 5.2: WER for Word Model

Room	clean	mmr	jr1	ofc	il412	lr400
First-Order %	0.45	0.98	2.21	2.91	2.62	5.08
Combined-Order %	0.21	0.78	1.89	2.5	2.26	4.88

Table 5.3: WER for Phoneme Model

Room	clean	mmr	jr1	ofc	il412	lr400
First-Order %	2.39	3.93	4.95	5.30	5.79	7.74
Combined-Order %	4.03	4.15	4.67	4.95	5.47	7.35

Table 5.4: WER for Triphone Model

Room	clean	mmr	jr1	ofc	il412	lr400
First-Order %	1.96	2.79	3.13	3.73	3.83	5.47
Combined-Order %	3.07	3	2.76	3.30	3.69	4.81

Tables 5.2, 5.3 and 5.4 show the recognition results of word, phoneme and triphone models for test data (section 5.1.3) in 6 different acoustic environments (section

5.1.2) respectively. The recognition rate is measured in WER. In the table, the results are arranged from left to right in decreasing order of SRRs of acoustic environments.

In all the cases, it is observed that, the CO-HMM gives a better recognition results than FO-HMM, except for clean and low reverberant (T_{60}=300ms) conditions of phonemes and triphones.

The improvement in recognition rate can be explained as follows:

The improvement in the recognition rate, is due to the formation of sharper acoustic models by capturing the inter-frame correlation using predecessor dependent output PDFs and thus increasing the discrimination capability of the acoustic model.

The decrement in recognition rate can be explained as follows:

There is a trade-off between the modeling precision and split of training data. In low reverberant conditions like clean and mmr, the loss of PDF data predominates the increase in modeling precision, whereas in moderate and high reverberant conditions the modeling precision predominates the loss of PDF data.

Chapter 6

Conclusions

In this thesis, a model based approach called 'CO-HMM' to achieve reverberation-robust speech recognition has been investigated. Reverberation has a dispersive effect on FVSs, thereby increasing the inter-frame correlation. The conventional FO-HMMs assume that the output probability of the current observation vector is conditionally independent of the previous feature vectors, which is not true in reverberant conditions. Therefore, the proposed concept, which is a combination of first-order and second-order HMM, attempts to characterize the statistical properties of the FVSs in reverberant conditions more closely by capturing inter-frame correlation using predecessor dependent output PDFs.

This model is evaluated by comparing its statistical properties with conventional FO-HMM. Entropy of a model is used as a measurement for the analysis. It has been shown that, the average entropy of CO-HMM is less than that of FO-HMM. This decrease in the entropy of the CO-HMM compared to conventional procedure suggests that the models built using the CO-HMM are sharper and have the property of higher discriminating capability.

The recognition results of word, phoneme, triphone models in different acoustic environments are published, which reflect the characteristic statistical analysis of the models, with few exceptions in phoneme and triphone models, which have a lower recognition rate than the FO-HMM in low reverberant conditions. This low recognition

rate is due to split of PDF data, which results insufficient training data to estimate the two PDFs (TS and SS) in CO-HMM. Hence, there is a trade-off between modeling precision and split data. The recognition results for low reverberant conditions could most likely be improved by using more training data. In highly reverberant conditions, however, the increase in modeling precision predominates the loss of PDF data.

List of Figures

List of Tables

Bibliography

[1] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev and P. Woodland. The HTK book (for HTK version 3.2), December 2002.

[2] L. Rebiner, B.H. Juang. Fundamentals of Speech Recognition. *Prentice-Hall International, Inc*, 1993.

[3] J. Benesty, J. Chen, Y. Huang and I. Cohen. Noise Reduction in Speech Processing. *Spriner Topic in Signal Processing* Vol. II, 2009.

[4] Wikipedia, Entropy (information theory), http://en.wikipedia.org/wiki/Entropy (information theory).

[5] S. Davis and P. Mermelstein. Comparison of parametric prepresentation for mono syllabic word recognition in continuous spoken sentences. *IEEE transactions on Acoustics, Speech and Signal processing*, 28(4), pp. 357-366, 1980.

[6] Stevens, S.S and J. Volkman. The relation of pitch to frequency. *American Journal of Psychology*, 1940, 53: p.329.

[7] S. Furui. On the role of Spectural transformation for speech perception. *Journal of the Acoustic Society, America*, 80(4), pp. 1016-1025, 1986.

[8] L. R. Raniner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceeding of the IEEE*, Vol. 77, No. 2, February 1989.

[9] A. Sehr, C. Hofmann, R. Maas and W. Kellermann. A Novel Approach for Matched

Reverberant Training of HMMs using Data Pairs. *INTERSPEECH*, 2010.

[10] A. M. Sehr. Reverberation Modeling for Robust Distant-Talking Speech Recognition (PhD Thesis), 2009.

[11] S.J. Young, N.H. Russell and J.H.S Thornton. Token Passing: a Simple Conceptual Model for Connected Speech Recognition Systems. *Cambridge University Engineering Department*, 1989.

[12] Wikipedia, Entropy estimation, http://en.wikipedia.org/wiki/Entropy estimation.

[13] Y. Gong. Speech recognition in noisy environments: A survey. *Speech Communication*, 16(3), pp. 261-291, April 1995.

[14] J.-C. Junqua. The lombard reflex and its role on human listeners and automatic speech recognizers. *The Journal of the Acoustic Society of America (JASA)*, 93(1), pp. 510-524, 1993.

[15] V. Stahl, A. Fischer, and R. Bippus. Acoustic synthesis of training data for speech recognition in living-room environments. *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 1, pp. 285-288, May 2001.

[16] A. Sehr, O. Gress, and W. Kellermann. Synthetisches Multicondition-Training zur robusten Erkennung verhallter Sprache. *Proc. ITG Fachtagnung Sprachkommunikation*, 2006.

[17] L. Couvreur and C. Couvreur. Robust automatic speech recognition in reverberant, environments by model selection. *Proceeding of International Workshop on Hands Free Speech Communication (HSC)*, pp. 147-150, April 2001.

[18] C. J. Wellekens. Explicit correlation in hidden Markov models for speech recognition. *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 384-387, 1987.

[19] J.-C. Junqua. The lombard reflex and its role on human listeners and automatic

speech recognizers. *The Journal of the Acoustic Society of America (JASA)*, 93(1), pp. 510-524, 1993.

[20] P. A. Naylor and N. D. Gaubitch. Speech Dereverberation. *Signals and Communication Technology*, Springer 2010.

[21] M. Wolfel and J. McDonough. Distant Speech Recognition, John Wiley & Sons, 2009.

[22] Wikipedia, Entropy (information theory), http://en.wikipedia.org/wiki/Entropy (information theory).

[23] J. P. Barker, M.P. Cooke, and D.P.W. Ellis. Decoding speech in the presence of other sources. *Speech Communication*, 45(1), pp. 5-25, January 2005.

[24] J. Ramirez, J.M. Gorriz and J. C. Segura. Voice Activity Detection. Fundamentals and Speech Recognition System Robustness. *I-Tech Education and Publishing, Vienna, Austria*, 2007.

[25] R.G. Leonard. A database for speaker-independent digit recognition. *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 42.11.1-42.11.4,1984.

[26] T.Takiguchi and M. Nishimura. Acoustic model Adaptation using first order prediction for reverberant speech. *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, I:869-872,2004.

[27] T.Takiguchi and M. Nishimura and Y. Ariki. Acoustic model Adaptation using first order prediction for reverberant speech. *IEICE Transactions on Information and Systems*, E89-D(3), pp.908-914, March 2006.