Friedrich-Alexander-Universität Erlangen-Nürnberg

Lehrstuhl für Multimediakommunikation und Signalverarbeitung

SIM Project

An EM Algorithm for Reverberation Model Estimation

Ali Khairat

April 2011

Supervisors:

Prof. Dr.-Ing. Walter Kellermann

Roland Maas

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

When a speaker is close to the microphone, Automatic Speech Recognition (ASR) systems work very reliably. In real world environments, when the distance becomes larger, some distortion is added to the captured speech signal, due to reverberation, which affects the recognition process. For the training of Hidden Markov Model (HMM)-based recognizers, recordings are usually taken from near distances. The recognized utterances may differ from the models, if they are captured by distant-talking microphones, since the reverberant speech feature vectors affect the performance of traditional HMM-based recognizers. The goal of model adaptation algorithms is to adapt the models to the target environment using only a few calibration utterances. A novel approach [5] for adaptation of the models to reverberation is based on first-order linear prediction. By adapting the HMMs of the recognizer in each frame, the dispersive character of reverberation can be accounted for. In this project, the Expectation Maximization (EM) algorithm for estimating the reverberation model is implemented. The adaptation approach was evaluated for different rooms based on connected-digit recognition experiments and showed better performance than Monte Carlo approach.

Chapter 1

Introduction

Modern digital media technologies have transmitted the world into a new era of automated services. Home automation is now considered as one of the richest topics of research specially, when it comes to speech recognition for home automated devices. Automatic speech recognition applications are now of crucial attraction for newly developed devices like mobile phones, interactive TVs, and other hands-free devices, which can be used in our daily lives.

When dealing with hands-free devices, the task of the speech recognizer is not as easy as it seems, one of the biggest challenges that faces speech recognition is reverberation. Reverberation is the reflection of speech signals due to hitting a barrier, not only the direct clean speech signal is the one that is captured by the microphone, but also the late reflections of the direct signal are captured by the microphone. Now, the information of the signal is not contained only in one frame, but in the preceding frames as well.

In order to overcome this problem, an approach for model adaptation to the speech signal was introduced, where the recognizer treats every observation of the speech frame as the direct signal added to it the early parts of the preceding frame. Another approach dealt with each observation by splitting hidden Markov model (HMM) state into substates to account for previous observations. In this work an approach is implemented, that uses first-order linear prediction of the spectral distortion of the reverberant speech, such that the observation signal is predicted from preceding frames.

The reverberation parameters are computed using the expectation maximization algorithm. The proposed approach has proved better performance, when compared to Monte Carlo approach.

The structure of this thesis is as follows. In the second chapter, the idea of hidden Markov model will be introduced, different approaches for hidden Markov model adaptation will be discussed, like model adaptation by adding attenuated excitations, model adaptation by state splitting of HMM, and model adaptation by using first-order linear prediction. In the third chapter, two approaches for the reverberation parameter estimation will be discussed, the Monte Carlo approach and the expectation maximization approach. In the last chapter, the implemented approach is evaluated and the accuracy of the two approaches is compared.

Chapter 2

HMM Adaptation

When using hands-free devices, a problem that affects the performance of the speech recognizer, is the multiple reflections of the signal on the walls of the room, known as reverberation. To overcome this problem, model adaptation of the clean speech HMM to the reverberant speech is performed, to increase the efficiency of the speech recognizer.

2.1 Hidden Markov Models

A hidden Markov model is a statistical Markov model, where the states of the Markov process in the modeled system are hidden or unobserved. As shown in Figure 2.1, a Markov model consists of states, observations and transitions. At discrete times, transitions between states take place, it is also possible that transitions happen back to the same state, according to a set of probabilities associated with the state. The time instants associated with state changes are denoted as $t = 1, 2, 3, ..$ and the actual state at time t as q_t. A full probabilistic description of the above system would require specification of the current state at time t, as well as all the predecessor states. This stochastic process could be called an observable Markov model, since the output of the process is the set of states at each instant of time, where each state corresponds to a physical (observable) event.

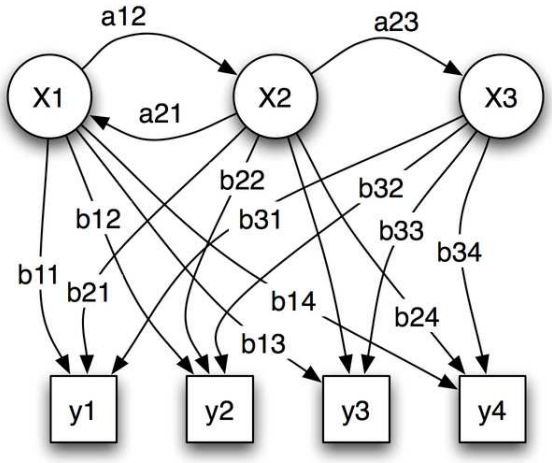
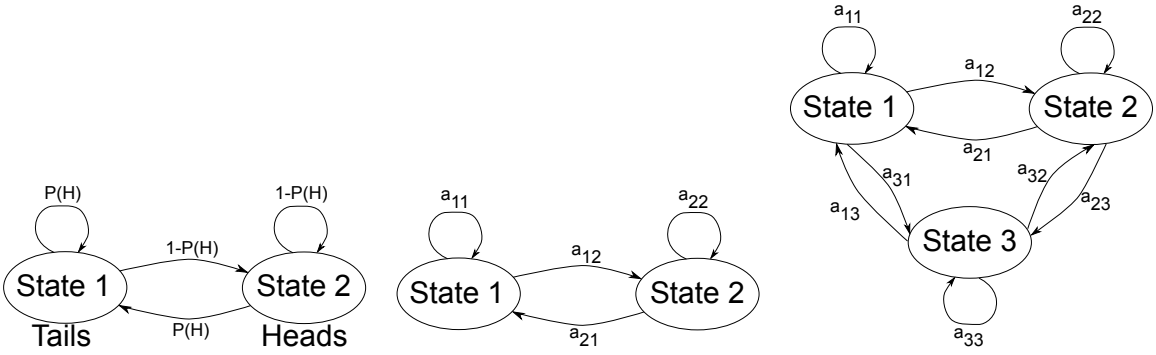


Figure 2.1: Markov Model Available [1]

HMM can be explained through the coin toss models, assuming that we have a person flipping a coin/multiple coins behind a barrier (e.g., a curtain) through which, what is happening behind the curtain cannot be seen. The person will not tell anything about what he is exactly doing, he will only tell the result of each coin flip. Thus, a sequence of hidden coin tossing experiments is performed, with the observation sequence consisting of a series of heads and tails, for example, a typical observation sequence would be $O = HTHTTTH$, where H stands for heads and T stands for tails Figure 2.2(a). Given this scenario, the problem of interest is how can a HMM be built to explain or model the observed sequence of heads and tails. The first problem one faces is deciding what the states in the model correspond to, and then deciding how many states should be in the model. One possible choice would be to assume that only a single biased coin was being tossed. In this case we could model the situation with a 2-state model, where each state corresponds to a side of the coin (i.e., heads or tails). In this case the Markov model is observable, and the only issue for complete specification of the model would be to decide on the best value for the bias (i.e., the probability of, say, heads). This model is a memory-less process and thus, is a degenerate case of a Markov model.

A second form of HMM for explaining the observed sequence of coin toss outcome as



(a) One coin (b) Two coins (c) Three coins

Figure 2.2: Coin Tossing Experiment

in Figure 2.2(b), the case where there are 2 coins and there are 2 states in the model and each state corresponds to a different coin being tossed. Each state is characterized by a probability distribution of heads and tails, and transitions between states are characterized by a state transition matrix. The physical mechanism which accounts for how the state transitions are selected could itself be a set of independent coin tosses, or some other probabilistic event. A third form of HMM for explaining the observed sequence of coin toss outcomes is given in Figure 2.2(c). This model corresponds to using 3 biased coins, and choosing from among the three, based on some probabilistic event.

A hidden Markov model is always characterized by some elements. First, the number of states in the model N. However, the states are not observable, yet there is still some physical significance to the set of states of the model, hence, in the coin tossing experiments, each state corresponded to a distinct biased coin. The individual states are denoted as $S = S_1, S_2, ..., S_N$, and the state at time t as q_t. Second, the number of distinct observation symbols per state M. The observation symbols correspond to the physical output of the system being modeled. For the coin toss experiments the observation symbols were simply heads or tails. Third parameter would be the state

transition probability distribution $A = \{a_{ij}\}$ where

$$a_{ij} = P\left[q_{t+1} = S_j | q_t = S_i\right] \qquad 1 \leq i, j \leq N \tag{2.1}$$

For the special case where any state can reach any other state in a single step, we have $a_{ij} > 0$ for all i, j. For other types of HMMs, we would have $a_{ij} = 0$ for one or more (i, j) pairs. Fourth parameter of a HMM is the the observation symbol output probability distribution in state j, $B = \{b_j(k)\}$, where

$$b_j(k) = p\left[v_k \; at \; t | q_t = S_j\right], \qquad 1 \leq j \leq N, \qquad 1 \leq k \leq M \tag{2.2}$$

Fifth is the initial state probability $\pi = \pi_i$ where

$$\pi = p\left[q_1 = S_i\right], \qquad 1 \leq i \leq N \tag{2.3}$$

Given appropriate values of N, M, A, B, and π, the HMM can be used as a generator to give an observation sequence $O = O_1 \, O_2 \, ... \, O_T$ (where each observation O, is one of the symbols from V, and T is the number of observations in the sequence). In order to build a HMM, these steps could be followed, first choose an initial state $q_1 = S_i$ according to the initial state probability π, then set $t = 1$, choose $0_t = v_k$ according to the symbol probability distribution in state S_i, then transit to a new state $q_{t+1} = S_j$, according to the state transition probability distribution a_{ij} for state S_i. Finally Set $t = t + 1$ and continue choosing $0_t = v_k$ until terminating the procedure if $t < T$. In order to indicate the complete parameter set of the model $\lambda = (A, B, \pi)$ [2].
The examples discussed so far illustrate the general characteristics of a HMM. Now, consider a model that generates acoustic parameter vectors.

For understanding the use of HMMs in ASR, assume that the acoustic parameter vectors are represented as codebook numbers, where the dictionary of the model output is a set of numbers. There is one acoustic vector per frame (e.g. each 10 ms interval of a signal), and transitions between states correspond to transitions from one frame to

the next, it is also possible that transitions are made to the same state, if the acoustic vector does change, or afterwards change the state again.

In the coin-tossing example, state transitions were permitted from each state to every other state. Such fully-connected models are called ergodic. When each state transitions corresponds to a change from one frame to the next, we are constrained by the laws of time, to which transitions are possible. The model in Figure 2.3 exemplifies a HMM called left-to-right or Bakis models, used in automatic speech recognition.

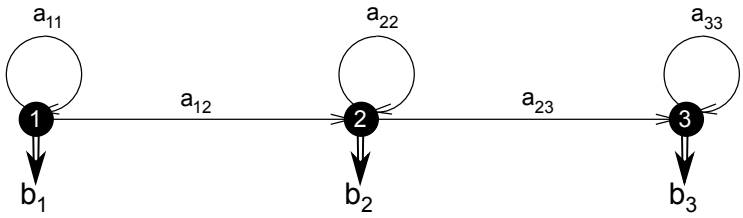


Figure 2.3: left-to-right or Bakis HMM

The hidden parameters (the transition probabilities and the output probabilities) are estimated by training the model on some training samples. The training samples can be considered as recordings of multiple pronunciations of each word. Improvement of the HMM is done by this training, dynamic time warping (DTW) based pattern matching can be considered for this operation. By training multiple examples of each word, where natural variations in the pronunciation of each word occur, can be incorporated in the training phase. Recognition, then, is the task of identifying, for a given sequence of input vectors (the observation sequence), which HMM best explains that sequence.

2.2 Model Adaptation

Hands-free speech input in a room environment has a bad effect on the performance of a recognition system. Adaptation of the HMM in a speech recognition system is done, in order to overcome the reverberation effect. The problem with reverberation, is the acoustic excitation of an artificial extension, also known as the reverberation tail when

observing the envelope of the short-term energy over the whole frequency range or in subbands.

Environments with reverberation time which is considered to be long, each frame of the speech is affected by reflected energy components from the preceding frames. For model adaptation of the parameters of a state, it becomes necessary to consider these frames, and compute their contributions to current state.

For model adaptation algorithms a complete set of reverberated training data is not required, only some features representations for reverberation and the clean speech model can be used to obtain the adapted HMM Figure 2.4.

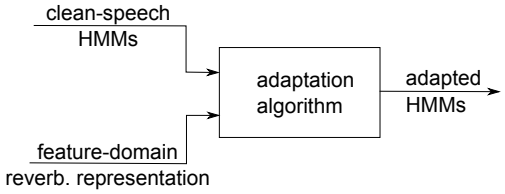


Figure 2.4: HMM adaptation

The idea is obtaining the HMM parameters, namely the means, variances, and the mixture weights of the output densities, by training HMMs on clean training data. Then, some parameters in a feature domain can be estimated to model or represent the distorted signal in a maximum likelihood sense by the Baum-Welch algorithm, by maximizing the probability that the set of training data has been produced by the HMM. The parameters of a clean-speech HMM then can be adapted to reverberation.

2.3 Model Adaptation by adding Attenuated Excitations

In this section the HMM adaptation approach from [3] is discussed. This approach is based on the assumption that the acoustic excitation of a speech segment modeled by a single HMM state, is considered as attenuated versions of previous HMM states. It

has been noticed that, adding attenuated excitations in the spectral domain at each HMM state, results in a significant improvement of the recognition performance.

In a room, the transmission between a speaker and a microphone can be modeled by the convolution between the speech signal and the room impulse response (RIR). However, the determination of the RIR is time variant, since it is subject to changes according to the room conditions, so factors like the speaker's position or opening a door or a window affects the RIR. Thus the estimation of the room impulse response is a quite difficult task, because of the high length of the impulse response and the time variant behavior.

The modeling of each speech segment is done with a single state HMM, which is described by a probability density function for the spectrum and frame energy. These HMM states model the speech segments by a set of Gaussian distribution for the acoustic parameters, where only the means of the parameters are of interest in this approach. The average duration of the speech segment is in the range between 20 and 100 ms.

> The idea of this approach is based on the existence of the acoustic excitation within a single HMM state as attenuated versions at later HMM states. These attenuated versions of the acoustic excitations from previous states will superpose the acoustic excitation of an observed HMM state [3].

For defining the individual attenuation, weighting coefficients are derived. These weighting coefficients describe the decaying characteristic of the room impulse response defined by the reverberation time of the room T_{60}. Figure 2.5 shows an exemplary curve of the decaying characteristic of the RIR for a value of 500 ms. Four HMM states are considered in this figure, where the average durations of these states can be estimated from the transition probabilities to remain in the corresponding state. The durations are used for defining the length of the corresponding segments in the exponential characteristic. In order to calculate these weighting coefficients, an exponential function is

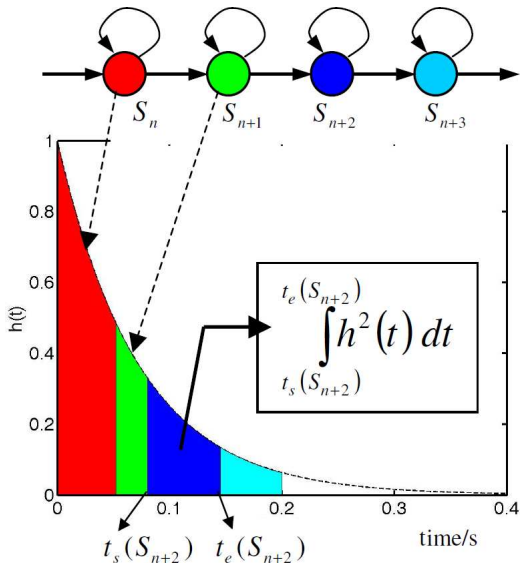


Figure 2.5: Determination of weighting coefficients [3]

used (equation 2.4)

$$w(i+n,i) = \int_{t_s(S_{i+n})}^{t_e(S_{i+n})} h^2(t)dt \tag{2.4}$$

where i is the index of an HMM state and $i+n$ is the index of the later state for which the energy contribution of the acoustic excitation at state i is calculated, $t_s(S_{i+n})$ and $t_e(S_{i+n})$ are the corresponding start and end time of state S_{i+n}, assuming a time measuring starting at the beginning of state S_i, and $h^2(t)$ is normalized so that $\int_0^\infty h^2(t)dt = 1$. The weighting coefficients for the example in Figure 2.5 describe in terms of spectral energy, how much of the acoustic excitation at state S_i will be seen in the later states S_{i+1} to S_{i+3}. The coefficients are calculated individually for each HMM state due to the different speech segments length.

After calculating the weighting coefficients, they can be used to adapt the frame energy of each HMM state by adding the corresponding contributions of the previous frames. In the same way the power density spectrum of the reverberant data X can be adapted after transforming back the cepstral coefficients to the Mel-spectral domain. The adaptation approach can be described as weighted sum of the spectrum $X(S_i)$

at HMM state S_i having an individual mixture component and the average spectra $\bar{X}(S_{i-n})$ of previous HMM states:
$\tilde{X}_{mel}(S_i) = w(i,i)\cdot X_{mel}(S_i) + w(i,i-1)\cdot \bar{X}_{mel}(S_{i-1}) + w(i,i-2)\cdot \bar{X}_{mel}(S_{i-2}) + \cdots$. This way the power density spectra in the Mel-spectral domain are individually adapted at each state and for each mixture component by taking into account the attenuated average spectra of previous HMM states. These average spectra are derived from a set of average cepstral coefficients, that are calculated as weighted sum over all mixture components.

As mentioned, the parameter that is needed for the adaptation is the reverberation time T_{60}. For estimating T_{60}, recognition of an utterance is performed with a set of adapted HMMs, then the clean HMM is adapted again by a varying value of T_{60} than that previously estimated. For each newly adapted model recognition a new T_{60} value is estimated and a new model is selected as well, which leads to maximum likelihood of the recognized sequence.

2.4 Model Adaptation by splitting State of HMM

Another approach for model adaptation, was provided in [4]. Most of the current model adaptation approaches perform well, when dealing with short reverberation time, but are unable to account the effect of preceding frames of speech effectively for long reverberation times. This method is based on the idea of representing speech frames using split-state HMM, instead of using the conventional single state HMM for representing a frame putting into account that at each frame observations from previous states occur.

In this approach, the reverberated signal is considered as the original signal convoluted with the impulse response. The reverberation signal with reverberation time (T_{60}) longer than the analysis window-length is expressed as $x(t) = h(t) * s(t)$, where x is the reverberated signal, s is the clean signal and h is the RIR. Since the impulse response of

the room is not directly given and its spectral parameters are not known, Mel-spectral domain parameters for reverberant speech are represented as weighted sums of the original speech frames $\hat{X}(t) = \alpha_0 S(t) + \alpha_1 S(t-1) + \alpha_2 S(t-2) + ... + \alpha_{N-1} S(t-N+1)$ where α_j is the reflection coefficient and N is selected according to the testing environment. The weight α_j is estimated for each Mel filter-bank from few seconds of adaptation data, calculated by minimizing mean-squared error (MMSE) $\|e(t)\|^2 = E(X(t) - \hat{X}(t))^2$. The adaptation data can be obtained through small amount of stereo recordings from close and far microphone. The coefficients can be estimated through other supervised or unsupervised methods. This formulation for the effect of preceding frames on the current frame will be used for the model adaptation purpose.

When considering the current speech at frame t, it contains spectral parameters from preceding frames at $t-1, t-2, ...$, so for the adaption of the output probability b_j at state $q_t = j$ of given HMM, the frames occurred at time $t-1, t-2, ...$ should also be taken into account. However this is not applicable in the conventional HMM used in most of speech recognition systems, where the observation can not be known deterministically and the state sequence preceding to a given state cannot be known. In a

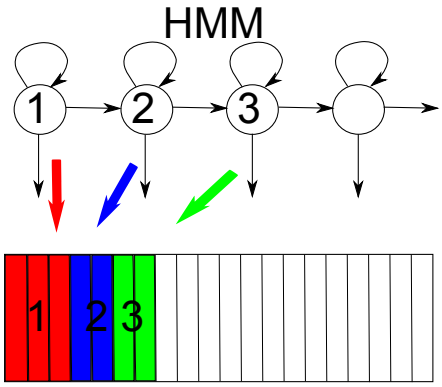


Figure 2.6: Conventional HMM where frames are aligned to a single state

conventional HMM Figure 2.6, if we consider at the first occurrence of state 3 there are frames from previous frames 1 and 2 which appear in the state sequence. For example we would have a state sequence at state 3 as {1, 1, 1, 2, 2} which can not be represented

by a single state HMM. In order to solve this, an expected number of occupation of preceding states is used to estimate the preceding state sequence, then composite mean from the output probability of the occurred state, as observation for the frame, is used. Now, with every new occurrence of a new state the adaptation will become harder, because the preceding state sequences are different, then there will be different values of compensation required at these different times. Its is difficult to accommodate the compensations into the single state of traditional HMM with static output probabilities.

The problem now is, we have several state sequences of previous states that can not be accounted by the conventional HMM to compensate for each state output.

The proposal is, to transform the conventional HMM into a split-state HMM (Figure 2.6) by splitting each state to a number of substates, each having a transition to another substate and the last substate has self transition loop. The transition probability from a substate to itself or another substate of its own parent state i is taken equal to self transition probability of the parent state a_{ii} , whereas from a substate of state i to a substate of state j, it is taken as the transition probability between 2 states a_{ij}. The output probability of each substate is initialized to be equal to that of its parent state. The number of substates under state i is taken proportional to expected occupation of the state in the way to minimize execution of self transition loop, and is constrained by compromise between complexity/speed and accuracy.

Due to the structure of split-state HMM and avoidance of self transition loops from states, there is no more ambiguity for number of frames coming from the same state. The preceding frames generated by the same state are essentially modeled by substates and are easily accounted for in the state sequence, while insuring that no substate except the last one (self loop) can occur more than once. Now considering the example in Figure 2.7 frames contributed from the same state can be accounted for estimation of preceding frames, the estimated state sequence for substate s23 will be {1, 1, 1, 1, 2, 2, 2}.

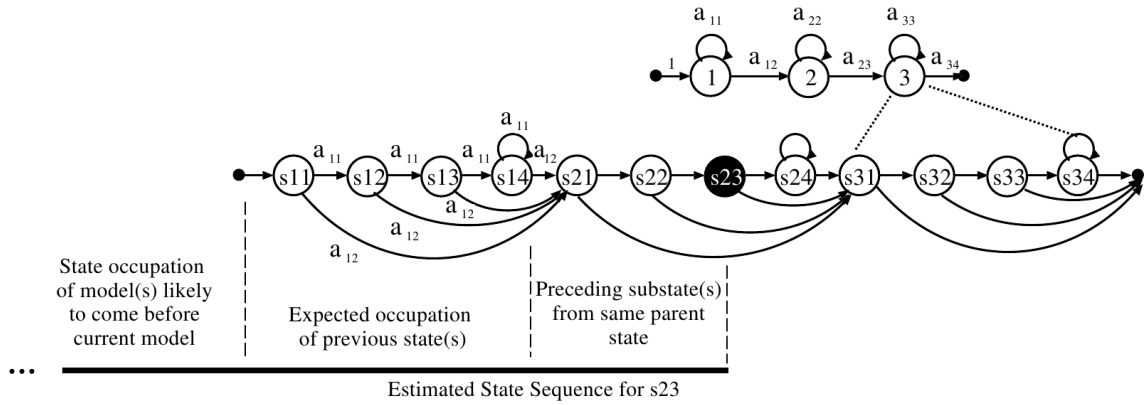


Figure 2.7: State Splitting [4]

For an estimated state sequence of length N the mean of a state is adapted by $\mu_{mel}^{(X)}(t) = \alpha_0\mu_{mel}^{(S)}(t) + \alpha_1\bar{\mu}_{mel}^{(S)}(t-1) + ... + \alpha_{N-1}\bar{\mu}_{mel}^{(S)}(t-N+1)$, where subscript mel represents the Mel-spectral domain parameters and $\bar{\mu}$ is the composite mean. The algorithm for the method states that once the split-state HMM is composed, and its output probabilites and transition matrix are initialized, preceding state sequence for each state is estimated. The composite means corresponding to output probabilites of preceding state sequence are transformed to Mel-spectral domain, and the adapted parameters for the state are computed, then the adapted parameters are transformed back to cepstral domain by applying the log and the discrete Cosine transform (DCT).

2.5 Model Adaptation by using First-Order Linear Prediction

As discussed previously in hands-free speech recognition, reverberation degrades the performance of recognition, since current frames are affected by preceding frames. For this approach proposed in [5], frame-by-frame adaptation method is introduced by adding the reflected frames to the means of the acoustic model. The reflection signal is approximated by a first-order linear prediction from the observation signal at the preceding frame, and the linear prediction coefficient is estimated with a maximum

likelihood of the adaptation data, using the EM algorithm.

Reverberation effect on the input speech appears as a convolution in the time domain, and it is represented as a multiplication in the Mel-spectral domain. For telephone channel or microphone that has short impulse responses conventional normalization techniques like CMS and RASTA proved their effectiveness, since the length of the analysis window is less than the length of the spectral analysis of speech. When having long reverberation time the efficiency is degraded due to the reflections of the preceding segments.

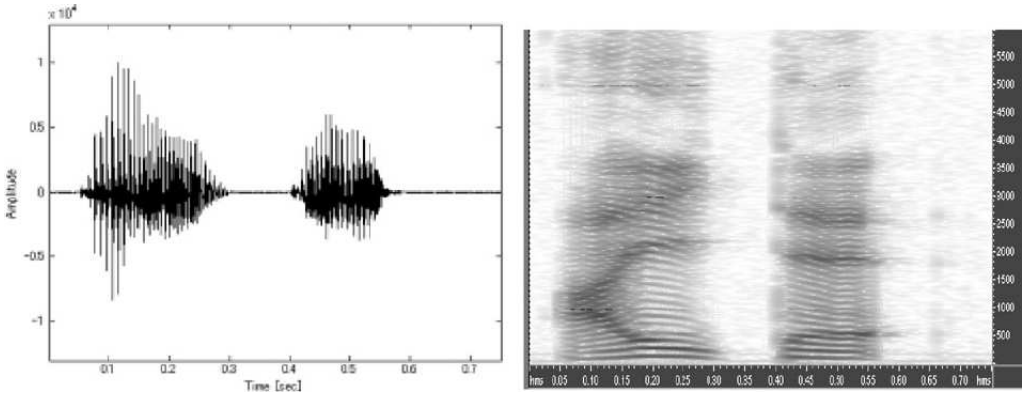
For this model adaptation technique for reverberant speech recognition, HMM composition using first-order linear prediction is performed. The reflection signal of the reverberant speech is approximated by the linear prediction from the observation signal at the preceding frame.

The observed signal is generally considered as the addition of the direct signal and the reflection signal

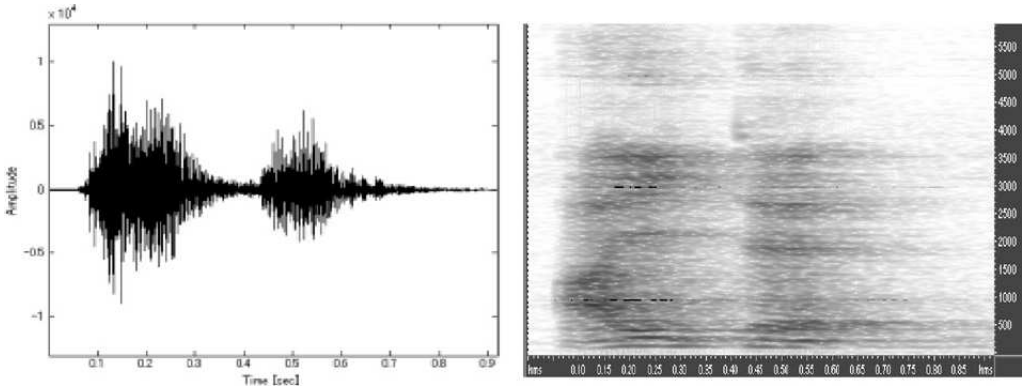
$$\begin{aligned} X(\omega;n) &\approx S(\omega;n)\cdot H_0(\omega) + \sum_{d=1} S(\omega;n-d)\cdot H_d(\omega) \\ &= \sum_{d=0} S(\omega;n-d)\cdot H_d(\omega) \end{aligned} \tag{2.5}$$

where $X(\omega;n)$ and $S(\omega;n)$ are the linear spectrum for the observed reverberant signal and the clean speech respectively of the frequency ω at the n-th frame, and $H(\omega)$ is the reflection factor modeling the RIR in the Mel-spectral domain. The reflected signal is represented as the summation of time delayed reflections, which may be longer than a phoneme interval, where the reflection signal can be seen as the overlapping segment from the previous segment. Reverberation effect can be clearly seen from Figures 2.8, where in Figure 2.8(b) the energy of the reflections appear as additive noise from the preceding segment of speech.

In the Mel-spectral domain, the reflection signal of the reverberant speech is represented as additive noise and approximated by a linear prediction from the observation signal



(a) Original Speech Signal



(b) Reverberant Speech Signal

Figure 2.8: Reverberation Effect [5]

at the preceding frame. The observed signal is represented as

$$X(\omega;n) \approx S(\omega;n)\cdot H(\omega) + \alpha(\omega)\cdot X(\omega;n-1)$$

where

$$\alpha(\omega)\cdot X(\omega;n-1) = \sum_{d=1} S(\omega;n-d)\cdot H_d(\omega) \tag{2.6}$$

where $\alpha(\omega)$ is the linear prediction coefficient for the frequency ω. The observation signal $X(\omega;n-1)$ includes besides the reflection signals, part of the direct signal. Adding the reflection signal to the means of the acoustic model, frame-by-frame adaptation is implemented for reverberant speech, which has the longer impulse response than the

analysis window. As shown in equation 2.6, the reverberation factor is approximated by the addition of the influence within the frame and outside of the frame. The former part in equation 2.6, is the compensation for the spectral distortion within each frame $H(\omega)$, and the latter is the reflection signal which is approximated by a first-order linear prediction from the observation signal at the preceding frame.

Figure 2.9 explains the procedure of model adaptation according to [5], where first the mean of the output probability density function (PDF) of the HMM of the clean speech and the mean of the spectral distortion within each frame are added in the cepstral domain:

$$\begin{aligned} \mu_{cep}^{(X)} &= \mu_{cep}^{(S)} + \mu_{cep}^{(H)}, \\ C_{cep}^{(X)} &= C_{cep}^{(S)} + C_{cep}^{(H)} \end{aligned} \tag{2.7}$$

where μ is the mean, C is the variance, $C_{cep}^{(H)}$ is considered to be set to zero.

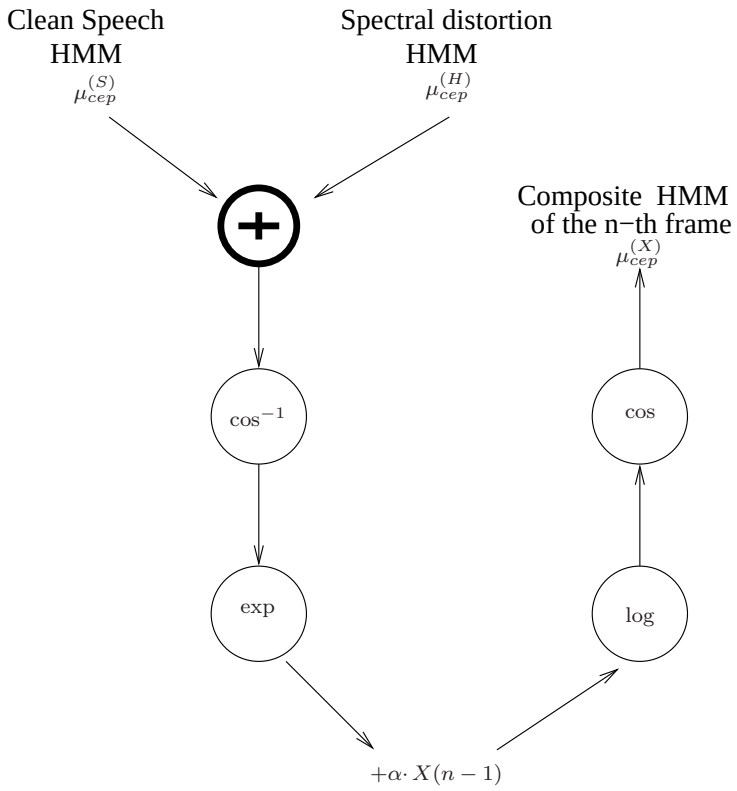


Figure 2.9: Frame-by-frame adaptation using a first-order linear prediction

Transformation of $\mu_{cep}^{(X)}, C_{cep}^{(X)}$ from the cepstral domain to the linear-spectral domain is done by computing the inverse cosine transform of each Gaussian probability density function of the HMM's $\mu_{log}^{(X)} = \Gamma^{-1}\mu_{cep}^{(X)}$, $C_{log}^{(X)} = (\Gamma^{-1})^T C_{cep}^{(X)}\Gamma^{-1}$, where Γ is a cosine transform matrix, $\mu_{log}^{(X)}$ and $C_{log}^{(X)}$ are the mean vector and covariance matrix of a Gaussian PDF in the log-power spectral domain, the transposition is denoted by T.
Afterwards, computation of the exponential transform to the Mel-spectral domain is done. The normal random vector is obtained by exponential transform, $Z = exp(Y)$.
The Idea of this adaptation approach is frame-by-frame adaptation to the reverberant speech using the preceding frame is done, where the reflection signal estimated by the linear prediction from the observation signal at the preceding frame is added to the means of the acoustic model, then transformation of the mean and the variance from the Mel-spectral domain to the cepstral domain is applied by computing first the log transform then computing the cosine transform to the cepstral domain.
Given the composite HMM for the reverberant speech, a speech recognition system estimates the word string associated with the test waveform. From (2.6) the following equation is obtained

$$\begin{aligned} X(\omega; n-1) &\approx \sum_{d=0} S(\omega; n-1-d)\cdot H_d(\omega) \\ \alpha(\omega)\cdot X(\omega; n-1) &\approx \sum_{d=0} S(\omega; n-1-d)\cdot \alpha(\omega)\cdot H_d(\omega) \end{aligned} \tag{2.8}$$

This means that the reverberation signal can be represented as weighted sum of previous distorted frames. Now the model parameter $\mu^{(H)}$ and α need to be estimated. Approches for model parameter estimation approaches will be discussed in the next chapter.

Chapter 3

Reverberation Model Estimation

In this chapter we will discuss how to estimate the adaptation model parameters using different approaches, like the Monte Carlo and Expectation Maximization.

3.1 Reverberation Model Estimation using Monte Carlo Approach

Monte Carlo Approach

Monte Carlo method is a type of algorithm based on the idea of performing random experiments several times and aggregating the result upon deterministic computations on the input. It is a simple method which is used for simulation in many mathematical and physical systems. Monte Carlo approach is often used when it is difficult to determine a value using deterministic algorithms, or for systems with significant uncertainty of the input values of the system. Monte Carlo method has several applications in the fields of physical sciences, engineering, computational biology, applied statistics, design and visuals, finance and business telecommunications, and even games.

Monte Carlo simulation methods always request a set of true random numbers, in order to be beneficial for some applications, such as primarily testing. It is easy for testing and re-running simulations when using deterministic or pseudo-random sequence tech-

niques. The important thing when applying this method is that the process used appears random.

Monte Carlo integration is considered to be a stochastic numerical integration method since it uses the randomness way for computing a result. It works well for high dimension sample space. By drawing support points for different evenly spaced samples $x(1), x(2), x(3), ..., x(N)$ from a function at random according to the probability $p(x)$ of each sample x, randomness is achieved. This way ensures that the samples are distributed and located according to their probabilities [6].

An example of the Monte Carlo method can be shown for estimating the value of π, the equation of the area of the circle can be used, since the circle equation is known and easy to use, in the same time it has the elements of more complex applications. First step, is to draw a square having length equal to the diameter of the circle, then draw a circle inside the square such that the center of the circle and the square is the same, then randomly spread dots on the whole area of the square Figure 3.1 knowing that the larger the number of dots, the greater the accuracy of the estimate. Last step

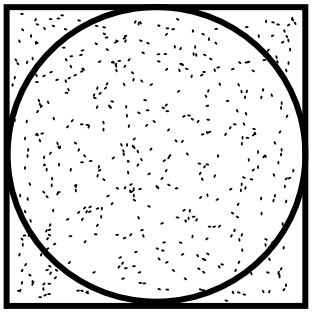


Figure 3.1: Monte Carlo example

to estimate the value of π, count the total number of the dots, then count the ones which are only inside the circle, using the area of the circle, π can be calculated as

$$\begin{aligned} \text{estimated circle area} &= \text{square area} \times \frac{\text{number of dots inside the circle}}{\text{total number of dots}} \\ \pi r^2 &= r^2 \times \frac{\text{number of dots inside the circle}}{\text{total number of dots}} \\ \pi &= \frac{\text{number of dots inside the circle}}{\text{total number of dots}} \end{aligned} \tag{3.1}$$

Parameter Estimation

Monte Carlo Method can be used for HMM adaptation techniques as well, Markov chain Monte Carlo (MCMC) sampling technique can be employed to mitigate the effects of noise in speaker identification systems while simultaneously enhancing the speech [7], Monte Carlo approach was also used in the investigation of the statistical properties of clean and reverberant data, including histograms of feature vector elements, joint densities, and correlations between different frames[8].

For estimating model parameters for the approach in [5], a sample of training data of digits was used, where recordings were captured as clean speech $s(n)$ and then convoluted with the room impulse response (RIR) $h(n)$ to obtain the reverberant speech $x(n)$.

For computing the parameters needed for the HMM adaptation (mean, variance, and reverberation parameter) for the reverberation models the RIR $h(n)$ is split into 2 parts $h_0(n)$ and $h_{rev}(n)$, where $h_0(n)$ is the early part of the distorted signal, basically, the part that contains low energy partitions of the RIR, and $h_{rev}(n)$ is the rest of the signal. The splitting part of the distorted signal to its early and late part, corresponds to the FFT frame shift interval. The convolution is done for the early part $s_0(n)$ and the later part $s_{rev}(n)$ as well. The Mel spectrum coefficients are extracted from the reverberant data, then the reverberation parameter $H_0(n)$ and the linear prediction coefficient α is calculated by the Monte Carlo method where

$$H_0(n) = \frac{X_0(n)}{S(n)} \tag{3.2}$$

and

$$\alpha = \frac{X_{rev}(n)}{X(n)} \tag{3.3}$$

Transformation of H_0 to the Mel-frequency cepstrum (MFCC) domain is done and the means and variances are calculated for H_0. For this HMM adaptation method α is expected to be in the Mel domain and $\mu^{(H_0)}$ is expected to be in the MFCC domain.

3.2 Reverberation Model Estimation using Expectation Maximization Algorithm

Expectation-maximization Algorithm

Expectation-maximization (EM) algorithm is a method used for parameter estimation based on maximum likelihood (ML) or maximum a posteriori (MAP) in statistical models that have hidden (unobservable, latent, and incomplete) data, and it is useful for high dimensional parameter spaces. In ML estimation all observations are assumed to be mutually statistically independent. The observations are kept fixed and the (log-)likelihood function is optimized regarding the parameters (Equation 3.4),

$$\hat{B} = \arg\max_{B} \; p(x; B) = \arg\max_{B} \; \log(x \mid B) \tag{3.4}$$

where x is the observed random variable and B is the parameter set. In MAP estimation, the probability density function of the parameters to be estimated are known (Equation 3.5),

$$\begin{aligned} \hat{B} &= \arg\max_{B} \; p(B \mid x) \\ &= \arg\max_{B} \; \frac{p(B)p(x \mid B)}{\sum_B p(B)p(x \mid B)} \\ &= \arg\max_{B} \; \log p(B) \log p(x \mid B) \end{aligned} \tag{3.5}$$

where B is considered as a random variable and its probability density function $p(B)$ is known.

EM algorithm consists of 2 steps expectation (E) step and maximization (M) step. In the expectation, or E-step, given the observed data and the current estimate or model parameters, we compute the expectation of the log-likelihood of the missing data which is achieved using the conditional expectation.

In the maximization, or M-step, the expected log-likelihood of the missing data found on the E-step is maximized. These parameter-estimates are then used to determine

the distribution of the latent variables in the next E-step. The convergence is assured since the algorithm is guaranteed to increase the likelihood at each iteration [9].
In order to derive the EM algorithm we need to explain first the missing information principle, where a colloquial formulation of the missing information principle (MIP) key equation can be explained as

$$\text{observable information} = \text{complete information} - \text{hidden information} \tag{3.6}$$

If we want to formalize the formula in 3.6 mathematically, we express the observable information as a random variable x, the hidden information as a random variable y, and the parameter set to be estimated B. The joint probability density of the events x (observation) and y (hidden) is

$$\begin{aligned} p(x; y; B) &= p(x; B)p(y \mid x; B) \\ \text{and thus:} & \\ p(x; B) &= \frac{p(x, y; B)}{p(y \mid x; B)} \end{aligned} \tag{3.7}$$

which can be expressed using the log as

$$-\log p(x; B) = -\log p(x, y; B) - (-\log p(y \mid x; B)) \tag{3.8}$$

We now consider the mathematical formulation of the key equation and derive an iterative parameter estimation scheme, where the iteration parameter is denoted by i and the key equation at $(i+1)st$ iteration as

$$\log p(x; \hat{B}^{(i+1)}) = \log p(x, y; \hat{B}^{(i+1)}) - \log p(y \mid x; \hat{B}^{(i+1)}) \tag{3.9}$$

where $\hat{B}^{(i+1)}$ denotes the estimation in iteration step $(i+1)$, next step we multiply both sides of equation 3.9 with $p(y \mid x; \hat{B}^{(i)})$ and integrate over the the hidden event y

$$\begin{aligned} \int p(y \mid x; \hat{B}^{(i)}) \log p(x; \hat{B}^{(i+1)})\, dy = &\int p(y \mid x; \hat{B}^{(i)}) \log p(x; y; \hat{B}^{(i+1)})\, dy \\ &- \int p(y \mid x; \hat{B}^{(i)}) \log p(y \mid x; \hat{B}^{(i+1)})\, dy \end{aligned} \tag{3.10}$$

Let's consider the left side of equation 3.10

$$\begin{aligned}\int p(y \mid x; \hat{B}^{(i)}) \log p(x; \hat{B}^{(i+1)})\, dy &= \int p(y \mid x; \hat{B}^{(i)})\, dy \log p(x; \hat{B}^{(i+1)}) \\ &= \log p(x; \hat{B}^{(i+1)})\end{aligned} \tag{3.11}$$

The left side of the key equation is now the log likelihood function of observations, this means that the maximization of the right hand side of the above key equation corresponds to a ML estimation.

As for the terms on the right hand side we introduce the following notation used for the two parts of the equation, the first part Kullback-Leibler Statistics $Q(\hat{B}^{(i)}; \hat{B}^{(i+1)})$ and the second part Entropy $H(\hat{B}^{(i)}; \hat{B}^{(i+1)})$, where

$$\begin{aligned}Q(\hat{B}^{(i)}; \hat{B}^{(i+1)}) &= \int p(y \mid x; \hat{B}^{(i)}) \log p(x; y; \hat{B}^{(i+1)})\, dy \\ H(\hat{B}^{(i)}; \hat{B}^{(i+1)}) &= -\int p(y \mid x; \hat{B}^{(i)}) \log p(y \mid x; \hat{B}^{(i+1)})\, dy\end{aligned} \tag{3.12}$$

Now we can write the key equation of the expectation maximization algorithm (EM algorithm) as

$$\log p(x; \hat{B}^{(i+1)}) = Q(\hat{B}^{(i)}; \hat{B}^{(i+1)}) + H(\hat{B}^{(i)}; \hat{B}^{(i+1)}) \tag{3.13}$$

The maximization of the Kullback-Leibler statistics can replace the optimization of the log-likelihood function [10]. The basic idea of the EM algorithm is instead of maximizing the log-likelihood function on the left side of the key equation, we maximize the Kullback-Leibler statistics iteratively, while ignoring the entropy term. The EM algorithm is done now through the 2 steps expectation and maximization. After initialization we calculate the Kullback-Leibler statistics as the expectation step $Q(\hat{B}^{(i)}; B)$, then the maximization step tries to select B which maximize the Kullback-Leibler statistics function $\hat{B}^{(i+1)} = \text{argmax}_B Q(\hat{B}^{(i)}; B)$, until $\hat{B}^{(i+1)} = \hat{B}^{(i)}$, which is the final parameter estimate $\hat{B}$.

Parameter Estimation

As explained in section 2.5 HMM model adaptation is done frame-wise, and for model parameter estimation like mean, variance, and linear prediction coefficient, the EM algorithm is used for this approach. Within each frame the estimation of the mean of the spectral distortion $\mu^{(H)}$ and the linear prediction coefficient α is done through the maximization of the likelihood of the adaptation data. The mean of the spectral distortion $\mu^{(H)}$ is estimated in the cepstral domain, in this case the linear prediction coefficient α is set to zero. The linear prediction coefficient α is estimated in the Mel-spectral domain.

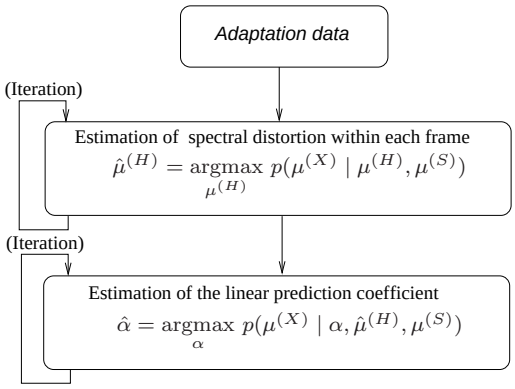


Figure 3.2: Estimation of Reverberant parameters using EM algorithm

As shown in Figure 3.2 the estimation of the mean of the spectral distortion parameters $\mu^{(H)}$ is performed using the EM algorithm, where we want to maximize the observation probability given the distorted signal and the clean signal

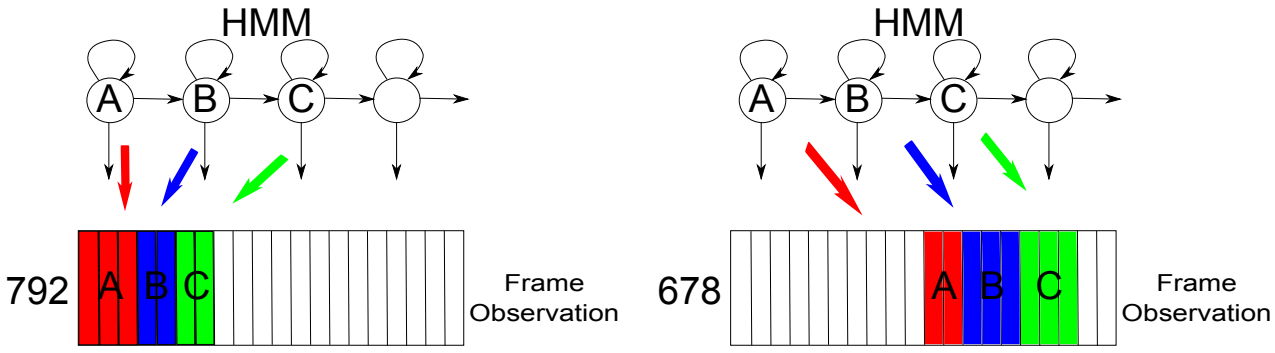
$$\hat{\mu}^{(H)} = \underset{\mu^{(H)}}{\text{argmax}}\ p(\mu^{(X)} \mid \mu^{(H)}, \mu^{(S)}, C^{(S)}) \tag{3.14}$$

The parameter that is used for the HMM separation is the mean vector where it is estimated as

$$\hat{\mu}^{(H)} = \frac{\sum_p \sum_j \sum_k \gamma_{p,j,k} \frac{\acute{\mu}^{(X)}_{p,j,k} - \mu^{(S)}_{p,j,k}}{C^{(S)}_{p,j,k}}}{\sum_p \sum_j \sum_k \frac{\gamma_{p,j,k}}{C^{(S)}_{p,j,k}}} \tag{3.15}$$

In this implementation, p is the HMM index, j is the state index, the HMM states have single Gaussian mixtures with mixture index k, $\mu^{(H)}$ is the mean vector of the distorted signal, and $\mu^{(S)}$ is the mean of the output PDF of the clean speech signal, $\acute{\mu}^{(X)}$ is calculated as the mean of the realizations frames aligned to a state, $C^{(S)}$ is the variance of the output PDF of the clean speech signal, and γ is the frame alignment parameter that defines which frames are aligned with which HMM state.

Hard alignment of frames is assumed, where for each frame, the HMM state connected to it, is previously known. From Figure 3.3, we can notice that the alignment of each frame is set to a state in the HMM. For example, in Figure 3.3(a), the HMM of the digit 7 "seven" is presented, and the utterance of the digit 792 "seven, nine, two" is aligned to it, where the first frames are connected to the states of the HMM. The same in Figure 3.3(b), for the utterance of the digit 678 "six, seven, eight", the middle frames are connected to the states of the HMM of the digit 7 "seven". The alignment parameter γ is expressed as a one or zero.



(a) Frame Alignment for 109

(b) Frame Alignment for 187

Figure 3.3: Frame HMM State Alignment

From equation 3.14, the mean value of the distorted signal is estimated in the cepstral domain using the EM algorithm, where the expectation function is $Q(\hat{\mu}^{(H)} \mid \mu^{(H)})$. In order to compute the Q-function, the steps explained above can be applied, where $\hat{B}^{(i)}$ is $\mu^{(H)}$, $\hat{B}^{(i+1)}$ is $\hat{\mu}^{(H)}$, x is $\acute{\mu}^{(X)}$, and y is $\mu^{(S)}$. As derived in [11], the E-step is expressed

in equation 3.16

$$\begin{aligned} Q(\hat{\mu}^{(H)} \mid \mu^{(H)}) &= \sum_p \sum_j \frac{p(\acute{\mu}_p^{(X)}, \mu^{(S)} \mid \mu^{(H)}, \mu^{(S)})}{p(\acute{\mu}_p^{(X)} \mid \mu^{(H)}, \mu^{(S)})} \cdot \log p(\acute{\mu}_p^{(X)}, \mu^{(S)} \mid \hat{\mu}^{(H)}, \mu^{(S)}) \\ &= \sum_p \sum_j \log N(\acute{\mu}^{(X)}; \mu^{(S)}, \mu^{(H)}) \\ &= \sum_p \sum_j \left\{ \frac{1}{2} \log(2\pi)^D C_j^{(S)} + \frac{(\acute{\mu}_{p,j}^{(X)} - \mu^{(S)} - \hat{\mu}^{(H)})^T (\acute{\mu}_{p,j}^{(X)} - \mu^{(S)} - \hat{\mu}^{(H)})}{2C^{(S)}} \right\} \end{aligned} \tag{3.16}$$

In the M-step we try to maximize $Q(\hat{\mu}^{(H)} \mid \mu^{(H)})$ with respect to $\hat{\mu}^{(H)}$, so we derive the Q-function with respect to $\hat{\mu}^{(H)}$ and set it to zero.

$$\frac{\partial Q(\hat{\mu}^{(H)} \mid \mu^{(H)})}{\partial \hat{\mu}^{(H)}} = \frac{\sum_p \sum_j \acute{\mu}_{p,j}^{(X)} - \mu^{(S)} - \hat{\mu}^{(H)}}{C^{(S)}} = 0 \tag{3.17}$$

From equation 3.17 we can calculate $\hat{\mu}^{(H)}$ as

$$\hat{\mu}^{(H)} = \frac{\sum_p \sum_j \frac{\acute{\mu}_{p,j}^{(X)} - \mu_{p,j}^{(S)}}{C_{p,j}^{(S)}}}{\sum_p \sum_j \frac{1}{C_{p,j}^{(S)}}} \tag{3.18}$$

In the implementation, for calculating the estimate of the mean vector of the distorted signal $\hat{\mu}^{(H)}$, we iterate over all HMMs in the digits dictionary and over all the states of each HMM then over all the mixtures of a state and calculate the formula.

Next step is to compose the HMMs of the mixed signal by adding the means of the clean speech $\mu^{(S)}$, and the spectral distortion $\mu^{(H)}$, in the cepstral domain according to equation 2.7, then transform the reverberant speech $\mu_{cep}^{(X)}$ from the cepstral domain to the Mel-spectral domain.

The estimation of the linear prediction coefficient α is computed by the maximum likelihood using the EM algorithm [5]. In the expectation step, we try to expect the conditional probability of the reverberant data X, the unobserved state sequence b and the unobserved mixture component labels corresponding to the observation sequence c given the estimated reverberation parameters $\hat{\alpha}$ and $\hat{\mu}^{(x)}$ equation 3.19

$$Q(\hat{\alpha} \mid \alpha) = E[\log p(X, b, c \mid \hat{\alpha}, \hat{\mu}^{(X)}) \mid \hat{\alpha}, \hat{\mu}^{(X)}] \tag{3.19}$$

The alignment for the adaptation data in the Mel-spectral domain is the same as that in the cepstral domain. The maximization step (M-step) in the EM algorithm becomes $\max Q(\hat{\alpha} \mid \alpha)$. To get the maximum $\hat{\alpha}$ we derivate $Q(\hat{\alpha} \mid \alpha)$ by $\hat{\alpha}$ and set it to zero $\frac{\partial Q(\hat{\alpha} \mid \alpha)}{\partial \hat{\alpha}} = 0$, then $\hat{\alpha}$ is calculated as

$$\hat{\alpha} = \frac{\sum_p \sum_\nu \sum_j \sum_k \sum_n \gamma_{p,\nu,j,k,n} \frac{X_{p,\nu,n-1}\{X_{p,\nu,n} - \hat{\mu}^{(X)}_{p,j,k}\}}{C^{(X)}_{p,j,k}}}{\sum_p \sum_\nu \sum_j \sum_k \sum_n \gamma_{p,\nu,j,k,n} \frac{X^2_{p,\nu,n-1}}{C^{(X)}_{p,j,k}}} \tag{3.20}$$

In equation 3.20, we iterate over the HMMs in the digits dictionary p, over all states j and mixtures k, for each utterance ν, and frame index n. $\mu^{(X)}$, $C^{(X)}$ and the realizations X are transformed to the Mel-spectral domain and the linear prediction coefficient $\hat{\alpha}$ is calculated. After applying frame alignment and setting $C^{(H)} = 0$, a simplified version of equation 3.20 will be

$$\hat{\alpha} = \frac{\sum_p \sum_j \sum_n \frac{X_{p,n-1}\{X_{p,n} - \hat{\mu}^{(X)}_{p,j}\}}{C^{(S)}_{p,j}}}{\sum_p \sum_j \sum_n \frac{X^2_{p,n-1}}{C^{(S)}_{p,j}}} \tag{3.21}$$

Chapter 4

Results

Experiments were conducted to assess the performance of the two techniques for reverberation model parameter estimation. Monte Carlo and expectation maximization methods for the HMM adaptation algorithm were applied on a connected-digit recognition task, to compare the efficiency of both approaches. The accuracy and mismatch of the recognizer as well as the performance of the estimated reverberation models is evaluated.

4.1 Experimental Setup

The connected-digit recognition task is chosen for evaluation since it can be considered as a generic example of continuous speech recognition [8]. It is supposed that, the probability of each digit is independent of preceding digits so no language model is required. It is also suited for the evaluation, since the recognition is dependent on the acoustic model quality. The EM approach was implemented by extending the decoding routine HVite of the HTK speech recognition toolkit [12]. For the feature representation 24 MFCC coefficients were used.

Continuous output density HMMs are used as acoustic models. A 16-state word-level HMM with no skips over states is used for each of the 11 digits

room	mmr	il412	lr400	ofc	jr1
type	lab	studio	seminar room	conference room	
T_{60}	300 ms	700 ms	900 ms	780 ms	600 ms
d	2.0 m	4.1 m	4.0 m	2.0 m	2.0 m
SRR	4.0 dB	-4.0 dB	-4.0 dB	0.5 dB	0.5 dB
N_{train}	36	18	44	36	36
N_{test}	18	6	22	18	18
M	20	50	70	50	50

Table 4.1: Rooms Characteristics

("zero"to "nine"and "oh "). Additionally, a three-state silence model with a backward skip from state three to state one is used. Single-Gaussian densities with diagonal covariance matrices are used as output pdfs. The HMMs are trained by ten iterations of Baum-Welch re-estimation. [8]

The HMM was trained on 4579 clean utterances based on a word-level, and for testing, 513 utterances were used for creating the reverberant test data which is produced by convolution of the speech signal with RIRs. The reverberation models are estimated based on 44 training utterances.

Table 4.1 shows the room characteristics were T_{60} is the reverberation time, d is the distance between the speaker and the microphone, SRR is the Signal-to-Reverberation Ratio, N_{train} and N_{test} are the number of RIRs in the training and test set, respectively, and M is the Length of the reverberation model.

4.2 Experimental Results

The adaptation techniques which were implemented using the 2 approaches Monte Carlo and expectation maximization were evaluated. The efficiency of both new techniques has been proved by some experiments on isolated and connected word recogni-

	Recognition accuracy %				
Room	mmr	il412	lr400	ofc	jr1
Monte Carlo using α	15.29	12.71	13.37	12.75	13.82
EM using α	39.89	14.84	27.06	36.61	43.01
Clean	85.53	55.92	48.42	61.75	62.73
Monte Carlo $\alpha = 0$	85.77	65.68	48.05	64.86	66.42
EM $\alpha = 0$	88.31	74.17	57.40	71.50	76.75

Table 4.2: Rooms word mismatch accuracy

tion of the TIDigits speech data base.

Simulations were tested for 5 different rooms, see Table 4.2. Room mmr has lower reverberation time than the other rooms, which explains why the recognition accuracy is better for this room than the other rooms.

In table 4.2, a comparison is made between the 2 techniques Monte Carlo and EM for the reverberation parameter estimation, when using α modeling the reverberation as spectral distortion and late reflection, or setting it to 0 compensating only for the spectral distortion, and the clean speech for the training of the HMM models.

When comparing the recognition accuracy between the adapted models using Monte Carlo or EM with model trained on clean speech, model adaptation in general performs better. Model adaptation emulates training the HMM on reverberant data after adding the part of the distorted signal to the clean speech, which enhances the performance of the recognizer, since the test data which are used are of modeled reverberant speech signals.

The recognition results for the EM approach, when using the linear prediction coefficient α or setting it to zero were compared. A remarkable degradation in performance was noticed when using α, an explanation for this could be, that the distinction between the HMM means is possible, as long as the recognizer tries to distinguish between

HMMs coming from the direct signal, however when the preceding frame is added to the direct signal, having dominant or larger mean value compared to the HMM means, the discrimination capability of the recognizer decreases.

EM shows better recognition accuracy rates than the Monte Carlo approach, EM estimates HMM parameters iteratively that converges to a maximum likelihood which is asymptotically unbiased and consistent (minimum variance estimator), besides EM takes into account the HMM and observations for estimation, while the Monte Carlo method estimates the parameters taking into account only data, which is not as optimum estimator as the EM.

Chapter 5

Conclusion

In this work, some model adaptation to reverberant speech signal approaches for speech recognition, as well as reverberation parameter estimation methods, were discussed. The main focus was on the approach suggested in [5] for acoustic model frame-wise adaptation for speech recognition based on first-order linear prediction approximation of the spectral distortion within each frame. The reverberation parameters like mean and linear prediction coefficient where estimated using the expectation maximization algorithm. Expectation maximization method for estimating the reverberation parameters was evaluated on speech recognition tasks and compared to the Monte Carlo method where it proved to have better performance.

Using model adaptation, it was shown that, compensation for the spectral distortion, improves the ASR performance. Approximation of reverberation tail by linear prediction, leads to degradation of ASR performance.

List of Figures

List of Tables

Bibliography

[1] Academic dictionaries and encyclopedias. [Online]. Available: http://en.academic.ru/pictures/enwiki/72/HiddenMarkovModel.png

[2] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[3] H. Hirsch and H. Finster, "A new HMM adaptation approach for the case of a hands-free speech input in reverberant rooms," in *Ninth International Conference on Spoken Language Processing*. ISCA, 2006.

[4] C. Raut, T. Nishimoto, and S. Sagayama, "Model adaptation by state splitting of HMM for long reverberation," in *Ninth European Conference on Speech Communication and Technology*. Citeseer, 2005.

[5] T. Takiguchi, M. Nishimura, and Y. Ariki, "Acoustic model adaptation using first-order linear prediction for reverberant speech," *IEICE Transactions on Information and Systems*, vol. 89, no. 3, p. 908, 2006.

[6] F. Faubel, "Speech Feature Enhancement for Speech Recognition by Sequential Monte Carlo Methods," 2006.

[7] C. wa Maina and J. Walsh, "Joint Speech Enhancement and Speaker Identification Using Monte Carlo Methods." Interspeech, 2009.

[8] A. Sehr, R. Maas, and W. Kellermann, "Reverberation Model-Based Decoding in the Logmelspec Domain for Robust Distant-Talking Speech Recognition," *Audio,*

Speech, and Language Processing, IEEE Transactions on, vol. 18, no. 7, pp. 1676–1691, 2010.

[9] S. Borman, "The Expectation Maximization Algorithm A short tutorial," *Submitted for publication*, 2004.

[10] A. Dempster, N. Laird, D. Rubin *et al.*, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[11] T. Takiguchi, S. Nakamura, and K. Shikano, "HMM-separation-based speech recognition for a distant moving speaker," *Speech and Audio Processing, IEEE Transactions on*, vol. 9, no. 2, pp. 127–140, 2002.

[12] Htk. [Online]. Available: http://htk.eng.cam.ac.uk/