Speech Recognition in Noisy Environments

Pedro J. Moreno
April 22, 1996

Department of Electrical
and Computer Engineering
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy in Electrical and Computer Engineering

Contents

List of Figures

Abstract

The accuracy of speech recognition systems degrades severely when the systems are operated in adverse acoustical environments. In recent years many approaches have been developed to address the problem of robust speech recognition, using feature-normalization algorithms, microphone arrays, representations based on human hearing, and other approaches.

Nevertheless, to date the improvement in recognition accuracy afforded by such algorithms has been limited, in part because of inadequacies in the mathematical models used to characterize the acoustical degradation. This thesis begins with a study of the reasons why speech recognition systems degrade in noise, using Monte Carlo simulation techniques. From observations about these simulations we propose a simple and yet effective model of how the environment affects the parameters used to characterize speech recognition systems and their input.

The proposed model of environment degradation is applied to two different approaches to environmental compensation, data-driven methods and model-based methods. Data-driven methods learn how a noisy environment affects the characteristics of incoming speech from direct comparisons of speech recorded in the noisy environment with the same speech recorded under optimal conditions. Model-based methods use a mathematical model of the environment and attempt to use samples of the degraded speech to estimate the parameters of the model.

In this thesis we argue that a careful mathematical formulation of environmental degradation improves recognition accuracy for both data-driven and model-based compensation procedures.The representation we develop for data-driven compensation approaches can be applied both to incoming feature vectors and to the stored statistical models used by speech recognition systems. These two approaches to data-driven compensation are referred to as RATZ and STAR, respectively. Finally, we introduce a new approach to model-based compensation with solution based on vector Taylor series, referred to as the VTS algorithms.

The proposed compensation algorithms are evaluated in a series of experiments measuring recognition accuracy for speech from the ARPA Wall Street Journal database that is corrupted by additive noise that is artificially injected at various signal-to-noise ratios (SNRs). For any particular SNR, the upper bound on recognition accuracy provided by practical compensation algorithms is the recognition accuracy of a system trained with noisy data at that SNR. The RATZ, VTS, and STAR algorithms achieve this bound at global SNRs as low as 15, 10, and 5 dB, respectively. The experimental results also demonstrate that the recognition error rate obtained using the algorithms proposed in this thesis is significantly better than what could be achieved using the previous state of the art. We include a small number of experimental results that indicate that the improvements in recognition accuracy provided by our approaches extend to degraded speech recorded in natural environments as well.

We also introduce a generic formulation of the environment compensation problem and its solution via vector Taylor series. We show how the use of vector Taylor series in combination with a Maximum Likelihood formulation produces dramatic improvements in recognition accuracy.

Acknowledgments

There are a lot of people I must recognize for their help in completing this project I started almost five years ago. First I must thank my thesis advisor, Professor Richard M. Stern. His scientific method has always been an inspiration for me. From him I have learned to ask the “whys” and “hows” in my research. Also, his excellent writing skills have always considerably improved my research papers (including this thesis!).

I must also recognize the other members of my thesis committee, Vijaya Kumar, Raj Reddy, Alejandro Acero and Bishnu Atal. I am indebted to Raj for creating the CMU SPHINX group and providing the research infrastructure used in this thesis. He has also generously provided the funding for my final year as a graduate student at CMU. Professor Kumar made excellent suggestions to improve this manuscript. Alejandro Acero is in part responsible for my joining CMU. He advised me in my early years at CMU and provided me with some of the very first robust speech recognition algorithms that are the seeds of the work presented here. He also carefully reviewed this manuscript and suggested several improvements. Finally, Bishnu Atal provided me with a more general perspective of my work and with valuable insights.

I must also thank the “*Ministerio de Educación y Ciencia*” from Spain and the Fulbright scholarship Program for their generous support during my first four years of stay at CMU.

During these five years at CMU I have had several colleagues and friends that have helped me in several ways. Evandro Gouvea has been willing to help in any kind of research experiment. The SNR plots I present in this thesis were pioneered by him in the summer of 1994. Matthew Siegler is one of the major contributors to the creation of the Robust Speech Group. His efforts in maintaining our software, directory structures, and other issues have made my experiments infinitely easier. I am also grateful to Sam-Joo Doh for his help in proofreading the final versions of this document. Eric Thayer and Ravi Mosur have been the core of the SPHINX-II system. It is only fair to say that without them the SPHINX-II system would not exist.

My good friend Bhiksha Raj deserves special mention. His arrival to our group made a big difference in my research. In a way my research changed dramatically (to the better) as a result of my collaboration and interactions with Bhiksha Raj. I have lost track of the many discussions we have had at 3 am. The algorithms in this thesis are the result of many discussions (over dinner and lots of beers) with him. He is also responsible for the real-time experiments reported in the thesis.

My friend Daniel Tapias, from Telefónica I+D, also has had a big influence in my work. His approach to research is relentless. He has showed me how little by little, any concept, no matter how difficult it is, can be mastered. Some of the derivations using the EM algorithm are based on a presentation he gave here at CMU in 1995.

I must mention too some other members of the SPHINX group for their occasional help and

advice. Bob Weide, Sunil Isar, Lin Chase, Uday Jain, and Roni Rosenfeld have always been there when needed. I am also thankful to my office mates, Rich Buskens, Mark Stahl and Mark Bearden for their friendship over the years.

John Hampshire and Radu Jasinschi deserve special mention. They have been a model of scientific honesty and integrity. I am lucky to have them as friends.

Finally, I want to dedicate this thesis to my parents, Pedro José and María Dolores, and sisters, Belén and Salud, for their love and support. They have always been there when I needed them the most. They have always encouraged me to follow my dreams.

Last but not least, I also want to dedicate this thesis to Carolina, my future wife, for her support and love. She has endured my busy schedule over the years always cheering me up when I felt disappointed. I could not think of a better person with whom to share the rest of my life.

Chapter 1
Introduction

The goal of errorless continuous speech recognition systems has remained unattainable over the years. Commercial systems have been developed to handle small to medium vocabularies with moderate performance. Large vocabulary systems able to handle 10,000 to 60,000 words have been developed and demonstrated under laboratory conditions. However, all these systems suffer substantial degradations in recognition accuracy when there is any kind of difference between the conditions in which the system is trained and the conditions in which the systems is finally tested.

Among other causes, differences between training and testing conditions can be due to:

- the speaking style
- the linguistic content of the task
- the environment

This dissertation focuses on the latter problem, environmental robustness.

1.1. Thesis goals

In recent years the field of environmental robustness has gained wide acceptance as one of the primary areas of research in the speech recognition field. Several approaches have been studied [*e.g.* 26, 52]. Microphone arrays [14, 53], auditory-based representations of speech features [51, 16], approaches based on filtering of features [2, 20] and other algorithms have been studied and showed to increase the recognition accuracy.

Some of the most successful approaches to environmental compensation have been based on modifying the feature vectors that are input to a speech recognition system or modifying the statistics that are at the heart of the internal models used by recognition systems. These modifications may be based on empirical comparisons of high-quality and degraded speech data, or they may be based on analytical models of the degradation. Empirically-based methods tend to achieve faster compensation while model-based methods tend to be more accurate.

In this dissertation we show that speech recognition accuracy can be further improved by making use of more accurate models of degraded speech than had been used previously. We apply our techniques to both empirically-based methods and model-based methods, using a variety of opti-

mal estimation procedures.

We also believe that the development of improved environmental compensation procedures is facilitated by a rigorous understanding of how noise and filtering affect the parameters used to characterize speech recognition systems and their input. Toward this end we attempt to elucidate the nature of these degradations in an intuitive fashion.

Another major effort in this thesis is that of experimentation at different signal-to-noise ratios (SNRs). Traditional environmental techniques have generally been tested at high SNRs where as we will show most environmental techniques achieve similar recognition results. Hence the relative merit of a particular environmental compensation technique can be better explored by looking at a complete range of SNRs.

The goals of this thesis include:

- Development of compensation procedures that approach the recognition accuracy of fully retrained systems (that have been trained with data from the same environment as the testing set).
- Development of a useful generic formulation of the problem of environmental robustness.
- Presentation of a generic characterization of the effects of the environment on the distributions of the cepstral vectors of clean speech along with a simple model to characterize these effects.
- Presentation of a unified formulation for data-driven compensation of incoming feature vectors as well as the internal statistical representation used by the recognition system.
- Presentation of applications of this unified formulation for two particular cases:
    - the Multivariate-Gaussian-Based Cepstral Normalization (RATZ) algorithm that compensates incoming feature vectors.
    - the Statistical Reestimation (STAR) compensation algorithm that compensates internal distributions.
- Presentation of an improved general approach to model-based compensation and its application, the Vector Taylor Series (VTS) compensation algorithm.

- Demonstration that a more detailed mathematical formulation of the problem of environment compensation results in greater recognition accuracy and flexibility in implementation than previous methods.

1.2. Dissertation Outline

The thesis outline is as follows. Chapter 2 provides a brief description of the CMU SPHINX-II speech recognition system, and it describes the databases used for training and experimentation.

In Chapter 3 we describe some relevant previously-developed environment compensation techniques, and we introduce the compensation techniques that form the core of this thesis. We also discuss the major differences between the compensation algorithms proposed in this thesis and the previous ones.

In Chapter 4 we study the effects of the environment on the distributions of log spectra of clean speech by using simulated data. We also discuss reasons for the degradation in recognition accuracy introduced by the environment.

In Chapter 5 we present a unified view of data-driven environmental compensation methods. We show how approaches that modify the feature vectors of noisy input cepstra and approaches that modify the internal distributions representing the cepstra of clean speech can be described within a common theoretical framework.

In Chapter 6 we present the RATZ family of algorithms, which modify incoming feature vectors. We describe in detail the mathematical structure of the algorithms, and we present experimental results exploring some of the dimensions of the algorithm.

In Chapter 7 we present the STAR algorithms, which modify the mean vectors and covariance matrices of the distributions used by the recognition system to model speech. We describe the algorithms and present experimental results. We present comparisons of the STAR and RATZ algorithms and we show that the STAR compensation results in greater recognition accuracy. We also explore the effect of initialization in the blind STAR algorithms.

In Chapter 8 we introduce the Vector Taylor Series (VTS) approach to robust speech recognition. We present a generic formulation for the problem of model-based environment compensation, and we introduce the use of vector Taylor series as a more tractable approximation to characterize

the environmental degradation. We present a mathematical formulation of the algorithm and conclude with experimental results.

Finally, Chapter 9 contains our results and conclusions as well as suggestions for future work.

Chapter 2
The SPHINX-II Recognition System

Since the environmental adaptation algorithms to be developed will be evaluated in the context of continuous speech recognition, this chapter provides an overview of the basic structure of the recognition system used for the experiments described in this thesis. Most of the algorithms developed in this thesis are independent of the recognition engine used, and in fact they can be implemented as completely separate modules. Hence, the results and conclusions of this thesis should be applicable to other recognition systems.

The most important topic of this chapter is a description of various aspects of the SPHINX-II recognition system. We also summarize the databases used for evaluation in the thesis.

2.1. An Overview of the SPHINX-II System

SPHINX-II is a large-vocabulary, speaker-independent, Hidden Markov Model (HMM)-based continuous speech recognition system, like its predecessor, the original SPHINX system. SPHINX was developed at CMU in 1988 [31, 32] and was one of the first systems to demonstrate the feasibility of accurate, speaker-independent, large-vocabulary continuous speech recognition.

Figure 2-1 shows the fundamental structure of the SPHINX-II [22] system. We describe the functions of each block briefly.

2.1.1. Signal Processing

Almost all speech recognition systems use a parametric representation of speech rather than the waveform itself as the basis for pattern recognition. The parameters usually carry the information about the short-time spectrum of the signal. SPHINX-II uses mel-frequency cepstral coefficients (MFCC) as static features for speech recognition [11]. First-order and second-order time derivatives of the cepstral coefficients are then obtained, and power information is included as a fourth feature.

In this thesis we will use the cepstrum and log spectrum signal feature representations for environment compensation procedure. In each section we will clearly define which features we use and the reason for them.

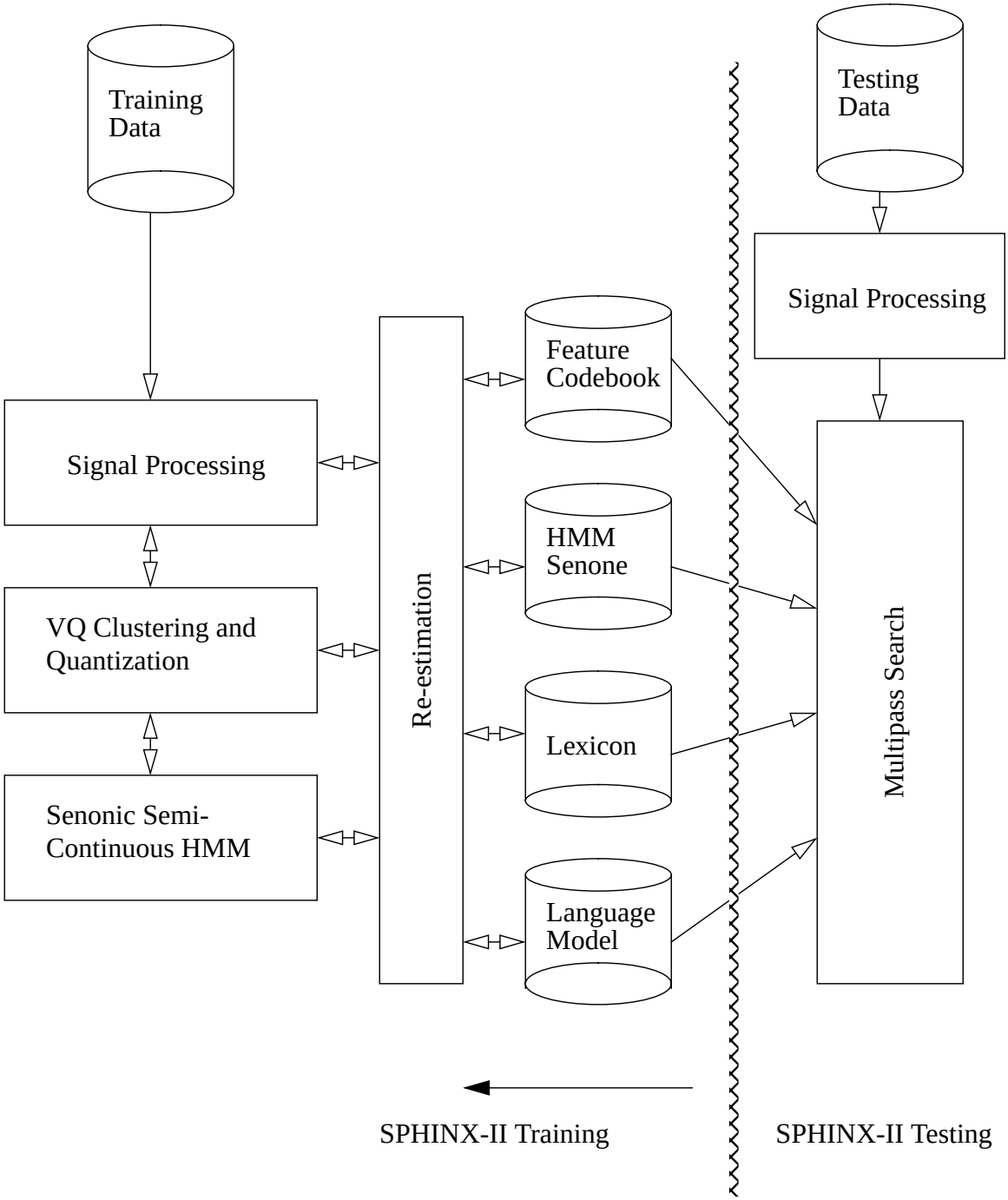

Figure 2-1. Block diagram of SPHINX-II.

The front end of SPHINX-II is illustrated in Figure 2-2. We summarize this feature extraction procedure as follows:

1. The input speech signal is digitized at a sampling rate of 16 kHz.

2. A pre-emphasis filter $H(z) = 1 - 0.97z^{-1}$ is applied to the speech samples. The pre-emphasis filter is used to reduce the effects of the glottal pulses and radiation impedance [38] and to focus on the spectral properties of the vocal tract.

3. Hamming windows of 25.6-ms duration are applied to the pre-emphasized speech samples at an analysis rate (frame rate) of 100 windows/sec.

4. The power spectrum of the windowed signal in each frame is computed using a 512-point DFT.

5. 40 mel-frequency spectral coefficients (MFSC) [11] are derived based on mel-frequency bandpass filters using 13 constant-bandwidth filters from 100 Hz to 1 kHz and 27 constant-Q filters from 1 kHz to 7 kHz.

6. For each 10-ms time frame, 13 mel-frequency cepstral coefficients (MFCCs) are computed using the cosine transform, as shown in Equation (2.1)

$$\boldsymbol{x}_t[k] = \sum_{i=0}^{39} X_t[i] \cos[k(i + 1/2)(\pi/40)] \qquad 0 \le k \le 12 \tag{2.1}$$

where $X_t[i]$ represents the log-energy output of the i^{th} mel-frequency bandpass filter at time frame t and $\boldsymbol{x}_t[k]$ represents the k^{th} cepstral vector component at time frame t. Note that unlike other speech recognition systems [*e.g.* 57], the $\boldsymbol{x}_t[0]$ cepstrum coefficient here is the sum of the log spectral band energies as opposed to the logarithm of the sum of the spectral band energies. The relationship between the cepstrum vector and the log spectrum vector can be expressed in matrix form as

$$\begin{bmatrix} \boldsymbol{x}_t[0] \\ \dots \\ \boldsymbol{x}_t[k] \\ \dots \\ \boldsymbol{x}_t[12] \end{bmatrix} = \begin{bmatrix} d_{k,i} \end{bmatrix} \begin{bmatrix} X_t[0] \\ \dots \\ X_t[i] \\ \dots \\ X_t[39] \end{bmatrix} \tag{2.2}$$

$$d_{k,i} = \cos[k(i + 1/2)(\pi/40)]$$

where $\left[d_{k,i}\right]$ is a 13x40 dimensional matrix.

7. The derivative features are computed from the static MFCCs as follows,

(a) Differenced cepstral vectors consist of 40-ms and 80-ms differences with 24 coefficients

$$\begin{aligned} \Delta \boldsymbol{x}_t[k] &= \boldsymbol{x}_{t+2}[k] - \boldsymbol{x}_{t-2}[k], 1 \le k \le 12 \\ \Delta \boldsymbol{x}'_t[k] &= \boldsymbol{x}_{t+4}[k] - \boldsymbol{x}_{t-4}[k], 1 \le k \le 12 \end{aligned} \tag{2.3}$$

(b) Second-order differenced MFCCs are then derived in similar fashion, with 12 dimensions.

$$\Delta\Delta \boldsymbol{x}_t[k] = \Delta \boldsymbol{x}_{t+1}[k] - \Delta \boldsymbol{x}_{t-1}[k], 1 \le k \le 12 \tag{2.4}$$

(c) Power features consist of normalized power, differenced power and second-order differenced power.

$$\begin{aligned} \bar{\boldsymbol{x}}_t[0] &= \boldsymbol{x}_t[0] - \max\{\boldsymbol{x}_i[0]\} \\ \Delta \boldsymbol{x}_t[0] &= \boldsymbol{x}_{t+2}[0] - \boldsymbol{x}_{t-2}[0] \\ \Delta\Delta \boldsymbol{x}_t[0] &= \Delta \boldsymbol{x}_{t+1}[0] - \Delta \boldsymbol{x}_{t-1}[0] \end{aligned} \tag{2.5}$$

Thus, the speech representation uses 4 sets of features including: (1) 12 Mel-frequency cepstral coefficients (MFCC); (2) 12 40-ms differenced MFCC and 12 80-ms differenced MFCC; (3) 12 second-order differenced cepstral vectors; and (4) power, 40-ms differenced power, and second-order differenced power. These features are all assumed to be statistically independent for mathematical and implementational simplicity.

2.1.2. Hidden Markov Models

In the context of statistical methods for speech recognition, hidden Markov models (HMM) have become a well known and widely used statistical approach to characterizing the spectral properties of frames of speech. As a stochastic modeling tool, HMMs have an advantage of providing a natural and highly reliable way of recognizing speech for a wide variety of applications. Since the HMM also integrates well into systems incorporating information about both acoustics and semantics, it is currently the predominant approach for speech recognition. We present here a brief summary of the fundamentals of HMMs. More details about the fundamentals of HMMs can be found in [6, 7, 25, 32, 34].

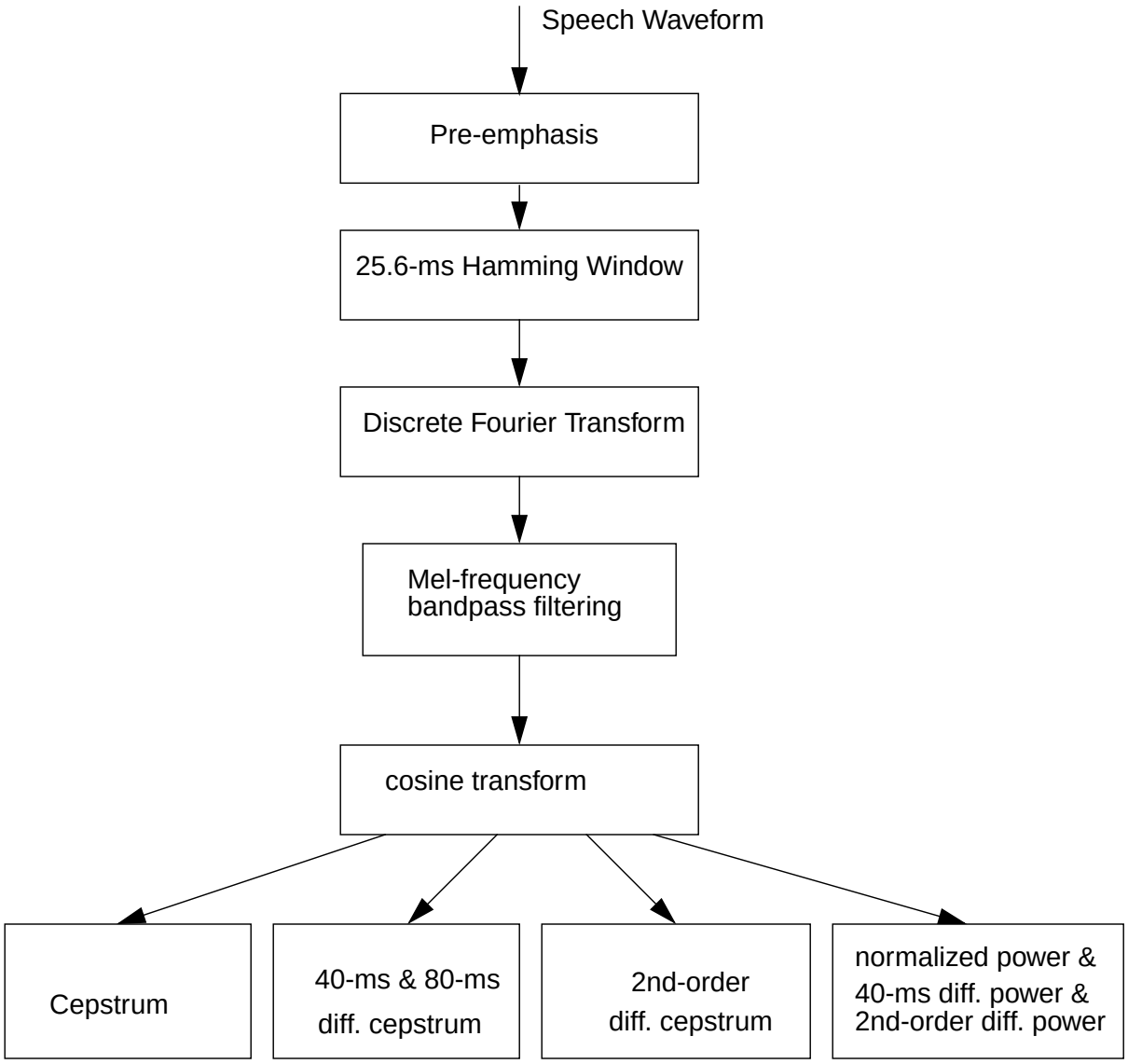

Figure 2-2. Block diagram of SPHINX-II's front end.

Hidden Markov models are a "doubly stochastic process" in which the observed data are viewed as the result of having passed the true (hidden) process through a function that produces the second process (observed). The hidden process consists of a collection of states (which are presumed abstractly to correspond to states of the speech production process) connected by transitions. Each transition is described by two sets of probabilities:

- A **transition probability**, which provides the probability of making a transition from one state to another.
- An **output probability** density function, which defines the conditional probability of observ-

ing a set of speech features when a particular transition takes place. For semicontinuous HMM systems (such as SPHINX-II) or fully continuous HMMs [27], pre-defined continuous distribution functions are used for observations that are multi-dimensional vectors. The continuous density function most frequently used for this purpose is the multivariate Gaussian mixture density function.

The goal of the decoding (or recognition) process in HMMs is to determine a sequence of (hidden) states (or transitions) that the observed signal has gone through. The second goal is to define the likelihood of observing that particular event given a state determined in the first process. Given the definition of hidden Markov models, there are three problems of interest:

- **The Evaluation Problem**: Given a model and a sequence of observations, what is the probability that the model generated the observations? This solution can be found using the forward-backward algorithm [47, 9].
- **The Decoding Problem**: Given a model and a sequence of observations, what is the most likely state sequence in the model that produced the observation? This solution can be found using the Viterbi algorithm [55].
- **The Learning Problem**: Given a model and a sequence of observations, what should the model's parameters be so that it has the maximum probability of generating the observations? This solution can be found using the Baum-Welch algorithm (or the forward-backward algorithm) [9, 8].

2.1.3. Recognition Unit

An HMM can be used to model a specific unit of speech. The specific unit of speech can be a word, a subword, or a complete sentence or paragraph. In large-vocabulary systems, HMMs are usually used to model subword units [6, 30, 10, 29] such as phonemes, while in small-vocabulary systems HMMs tend to be used to model the words themselves.

SPHINX-II is based on phonetic models because the amount of training data and storage required for word models is enormous. In addition, phonetic models are easily trainable. However, the phone model is inadequate to capture the variability of acoustical behavior for a given phoneme in different contexts. In order to enable detailed modeling of these co-articulation effects, triphone models were proposed [50] to account for the influence by the neighboring contexts.

Because the number of triphones to model can be too large and because triphone modeling does

not take into account the similarity of certain phones in their effect on neighboring phones, a parameter-sharing technique called distribution sharing [24] is used to describe the context-dependent characteristics for the same phones.

2.1.4. Training

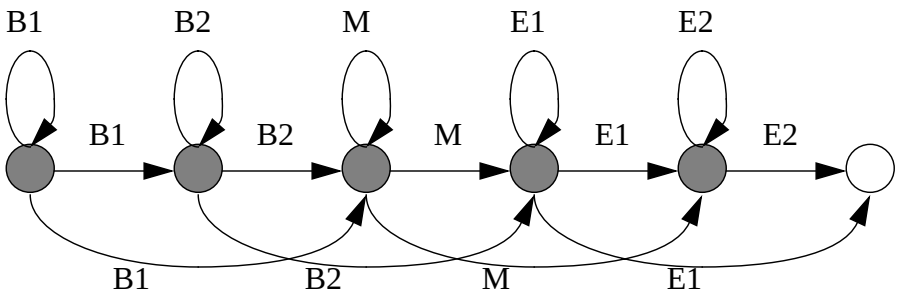

Figure 2-3. The topology of the phonetic HMM used in the SPHINX-II system.

SPHINX-II is a triphone-based HMM speech recognition system. Figure 2-3 shows the basic structure of the phonetic model for HMMs used in SPHINX-II. Each phonetic model is a left-to-right Bakis HMM [7] with 5 distinct output distributions.

SPHINX-II [22] uses a subphonetic clustering approach to share parameters among models. The output of clustering is a pre-specified number of shared distributions, which are called senones [24]. The senone, then, is a state-related modeling unit. By using subphonetic units for clustering, the distribution-level clustering provides more flexibility in parameter reduction and more accurate acoustic representation than the model-level clustering based on triphones.

The training procedure involves optimizing HMM parameters given an ensemble of training data. An iterative procedure, the Baum-Welch algorithm [9,47] or forward-backward algorithm, is employed to estimate transition probabilities, output distributions, and codebook means and variances under a unified probabilistic framework.

The optimal number of senones varies from application to application. It depends on the amount of available training data and the number of triphones present in the task. For the training corpus and experiments in this thesis which will be described in Section 2.2., we use 7000 senones

for the ARPA Wall Street Journal task with 7200 training sentences.

2.1.5. Recognition

For continuous speech recognition applied to large-vocabulary tasks, the search algorithm needs to apply all available acoustic and linguistic knowledge to maximize recognition accuracy. In order to integrate the use of all the lexical, linguistic and acoustic source of knowledge SPHINX-II uses a multi-pass search approach [5]. This approach uses the Viterbi algorithm [55] as a fast-match algorithm, and a detailed re-scoring approach to the N-best hypotheses [49] to produce the final recognition output.

SPHINX-II is designed to exploit all available acoustic and linguistic knowledge in three search phases. In Phase One a Viterbi beam search is applied in a left-to-right fashion, as a forward search, to produce best-matched word hypotheses, along with information about word ending times and associate scores, using detailed between-word triphone models and a bigram language model.

In Phase Two, a Viterbi beam search is performed in a right-to-left fashion, as a backward search, to generate all possible word beginning times and scores using the between-word triphone models and a bigram language model. In Phase Three, an A* search [44] is used to produce a set of N-best hypotheses for the test utterance by combining the results of Phases One and Phase Two for reescoring by a trigram language model.

2.2. Experimental Tasks and Corpora

To evaluate the algorithms proposed in this thesis, we have used the Wall Street Journal (WSJ) database. This database consists of several subsets encompassing different vocabulary sizes, environmental conditions, foreign accents, etc [45].

2.2.1. Testing databases

We have focused on the 5,000-word vocabulary size “clean” speech subset of the database submitted for evaluation on 1993. This subset was recorded over a Sennheiser close-talking, headset-mounted noise-cancelling microphone (HMD-410 or HMD-414). These data were contaminated with additive white Gaussian noise at several SNRs. The testing set contained 215 sentences with a total number of 4066 words belonging to 10 different native speakers of American English. The testing set contained five male and five female speakers. In all of our experiments we provide the

performance of the compensation algorithms at SNRs from zero to thirty decibels. A reasonable upper bound on the recognition accuracy of each compensation algorithm is the recognition accuracy of a fully retrained system. The expected lower bound on recognition accuracy is the accuracy of a system with no compensation enabled.

A second focus of our experiments was the Spoke 10 subset released in 1994 for study of environment compensation algorithms in the presence of automobile noise, which contains a vocabulary of 5,000-words. The testing set contained 113 sentences with a total number of 1937 words from 10 different speakers. The automobile noise was collected through an omni-directional microphone mounted on the drivers's side sun visor while the car was traveling at highway speeds. The windows of the car were closed and the air-conditioning was turned on. A single lengthy sample of the noise was collected so that it could span across the entire test set. The noise was scaled to three different levels so that the resulting SNR of the noise plus speech equalled three target SNRs picked by NIST and unknown to the speech recognition system. A one-minute sample of noise was provided for adaptation although in our experiments this was not used.

2.2.2. Training database

In our studies we use the official speaker-independent training corpus, referred to as "WSJ0-si_trn", supplied by the National Institute of Standards and Technology (NIST) containing 7240 utterances of read WSJ text. These sentences were recorded using a Sennheiser close-talking noise-cancelling headset. The training utterances are collected from 84 speakers. All these data are used to build a single set of gender independent HMMs to train the SPHINX-II system.

2.3. Statistical Significance of Differences in Recognition Accuracy

The algorithms we propose in this dissertation are evaluated in terms of recognition accuracy observed using a common standardized corpus of speech material for testing and training. Recognition accuracy is obtained by comparing the word-string output produced from the recognizer (hereafter referred to as the hypothesis) to the word string that had been actually uttered (hereafter referred to as the reference). Based on a standard nonlinear string-matching program, word error rate is computed as the percentage of errors including insertion, deletion and substitution of words.

It is important to know whether any apparent difference in performance of the algorithms is statistically significant in order to interpret experimental results in an objective manner. Gillick and

Cox [17] proposed the use of the McNemar's test and a matched-pairs test for determining the statistical significance of recognition results. Recognition errors are assumed to be independent in the McNemar's test or independent across different sentence segments in the matched-pairs test, respectively. Picone and Doddington [46] also advocated a phone-mediated alternative to the conventional alignment of reference and hypothesis word strings for the purpose of analyzing word errors. NIST has implemented several automated benchmark scoring programs to evaluate statistical significance of performance differences between systems.

Many results produced by different algorithms do not differ from each other by a very substantial margin, and it is to our interest to know whether these performance differences are statistically significant. A straightforward solution is to apply the NIST "standard" benchmark scoring program to compare a pair of results.

In general, the statistical significance of a particular performance improvement is closely related to the differences in error rates, and it also varies with the number of testing utterances, the task vocabulary size, the positions of errors, the grammar, and the range of overall accuracy. Nevertheless, for the ARPA WSJ0 task with the SPHINX-II system, a rule of thumb we have observed is that performance improvement is usually considered to be significant if the absolute difference in accuracy between two results is greater than 1%. There is usually no statistically significant difference if differences in error rate are less than 0.7%.

2.4. Summary

In this chapter, we reviewed the overall structure of SPHINX-II that will be used as the primary recognition system in our study. We also described the training and evaluation speech corpora that we employ to evaluate the performance of our algorithms in the following chapters. The primary vehicle for research of this thesis will be the WSJ0 5,000-word 7240 sentences training corpora from which a single set of gender independent 7000-senonic HMMs will be constructed. Using these models we will evaluate the environmental compensation algorithms proposed in this thesis with the 1993 WSJ0 5,000-word clean speech evaluation set, adding white noise at different SNRs

and with the 1994 5,000-word Spoke 10 multimicrophone evaluation set.

Chapter 3
Previous work in Environmental Compensation

In this section we discuss some of the latest and most relevant algorithms in environmental compensation that relate to those presented in this thesis. They all share similar assumptions, namely:

- they use a cepstrum feature vector representation of the speech signal
- they use a statistical characterization of the feature vector based on mixtures of Gaussians, Vector Quantization (VQ), or even a more detailed modelling provided by HMMs.

In this chapter we briefly review these algorithms and relate them to the algorithms proposed in this thesis. Finally, we also present a taxonomy of the algorithms presented in this dissertation.

3.1. Cepstral Mean Normalization

Cepstral mean normalization (CMN) [37] is perhaps one of the most effective algorithms considering its simplicity. It is a *de facto* standard in most large vocabulary speech recognition systems. The algorithm computes a long-term mean value of the feature vectors and subtracts this mean value from each of the vectors. In this way it assures that the mean value of the incoming feature stream is zero. This helps in reducing the variability of the data and also allows for a simple and yet effective channel and speaker normalization. The procedure is applied to both the training and testing data. In the experiments described in this thesis it is always used just before training or recognition but always after compensation.

However, the effectiveness of CMN is limited when the environment is not adequately modeled by a linear channel. For those situations more sophisticated algorithms are needed.

3.2. Data-driven compensation methods

As we will describe in Chapter 4, the effect of the environment on cepstra and log spectra of clean speech feature vectors can frequently be modeled by additive correction factors. These correction factors can be computed using “examples” of how clean speech vectors are affected by the environment. A simple and effective way of directly observe the effect of the environment on speech feature vectors is through the use of simultaneously recorded clean and noisy speech data, also known as “stereo-recorded” data.

In this section we describe the Probabilistic Optimum Filtering (POF) [43] and the Fixed Codeword Dependent Cepstral Normalization (FCDCN) [1] algorithms as primary examples of these approaches.

3.2.1. POF

The POF algorithm [43] uses a VQ description of the distribution of clean speech cepstrum combined with a codeword-dependent multidimensional transversal filter. The role of the multidimensional transversal filter is to capture temporal correlations across previous and past frame vectors.

POF learns the parameters of the VQ cell-dependent transversal filters (matrices) for each of the cells and for each environment through the minimization of an error function defined as the norm of the difference between the clean speech vectors and the noisy speech vectors. To do so it requires the use of stereo data.

One of the limitations of the POF algorithm is its dependency on stereo-recorded speech data. The use of a weak statistical representation of the clean speech cepstrum distributions as modeled by VQ makes the algorithm usable only when stereo-recorded data are available. Even if large amounts of noisy adaptation speech data are available, the algorithm cannot make use of them without parallel recordings of clean speech.

3.2.2. FCDCN

FCDCN [1] is similar in structure to POF. It uses a VQ representation for the distribution of clean speech cepstrum vectors and computes a codeword-dependent correction vector based on simultaneously recorded speech data. It suffers the same limitations as POF. The use of a weak statistical representation of cepstral vector distributions of clean speech based on VQ makes also the algorithm dependent on the availability of stereo-recorded data.

3.3. Model-based compensation methods

The previous compensation methods did not make any assumption about the environment since its effect on the cepstral vectors was directly modeled through the use of simultaneously-recorded clean and noisy speech. In this section we present methods that assume a model of the environment characterized by additive noise and linear filtering that do not require simultaneously recorded

speech data.

We describe the Codeword Dependent Cepstrum Normalization method (CDCN) [1] and the Parallel Model Combination method (PMC) [15]. Although strictly speaking it is not a model-based method, we also describe the Maximum Likelihood Linear Regression method (MLLR) [33] because of its similarity to PMC.

3.3.1. CDCN

CDCN [1] models the distributions of cepstra of clean speech by a mixture of Gaussian distributions. It analytically models the effect of the environment on the distributions of clean speech cepstrum. The algorithm works in two steps. The goal of the first step is to estimate the values of the environmental parameters (noise and channel vectors) that maximize the likelihood of the observed noisy cepstrum vectors. In the second step Minimum Mean Squared Estimation (MMSE) is applied to find the unobserved cepstral vector of clean speech given the cepstral vector of noisy speech.

The algorithm works on a sentence-by-sentence basis, needing only the sentence to be recognized to estimate environmental parameters.

3.3.2. PMC

The Parallel Model Combination approach [15] assumes the same model of the environment used by CDCN. Assuming perfect knowledge of the noise and channel vectors, it tries to transform the mean vectors and covariance matrices of the acoustical distributions of the HMMs to make them more similar to the ideal distributions of the cepstra of the noisy speech. Several possible alternatives exists to transform the mean vectors and covariance matrices.

However, all these versions of the PMC algorithm need previous knowledge of the noise and channel vectors. Its estimation is done beforehand using different approximations. Typically samples of isolated noise are needed to adequately estimate the parameters of PMC.

3.3.3. MLLR

MLLR [33] was originally designed as a speaker adaptation method but it has also proved to be effective for environment compensation [56]. The algorithm updates the mean vectors and covariance matrices of the distributions of cepstra for clean speech as modeled by the HMMs, given

a small amount of adaptation data. It finds a set of transformation matrices that maximize the likelihood of observing the noisy cepstrum vectors.

The algorithm does not make use of any implicit model of the environment. It only assumes that the mean vectors of the clean speech cepstrum distributions can be rotated and shifted by the environment.

3.4. Summary and discussion

The data-driven algorithms proposed in this thesis (RATZ and STAR) use richer and more detailed models of the distributions of cepstral vectors of speech than the POF and FCDCN algorithms. They generally use mixtures of Gaussian distributions in which **all** the parameters (priors, mean vectors, covariance matrices) are learned optimally from training data consisting of clean speech. The use of mixtures of Gaussians allows a representation of the feature space in which vectors are described as weighted sums of mixtures. In contrast, POF and FCDCN use a vector quantization representation of the feature space in which hard decisions are made to assign a vector as belonging to a particular codeword.

The data-driven methods we will propose are designed within a maximum likelihood framework. This framework facilitates easy extension to new conditions such as:

- unavailability of simultaneously recorded clean and noisy speech data.
- interpolation between different environmental conditions.

The model-based techniques proposed in this thesis (VTS-0 and VTS-1) also use a maximum likelihood framework. The formulation we introduce for these algorithms allows for an easy generalization to any kind of environment. They are able to work with a single sentence. Unlike PMC they do not need any extra information to estimate the noise or the channel.

Finally, all the algorithms proposed in this thesis assume a model of the effect of the environment on the distributions of cepstra or log spectra of clean speech that is closer to reality. Changes in both mean vectors and covariance matrices are modelled.

3.5. Algorithms proposed in this thesis

Figure 3-1 summarizes the environmental compensation techniques presented in this thesis. We first consider the data-driven compensation methods, RATZ and STAR, which make use of simul-

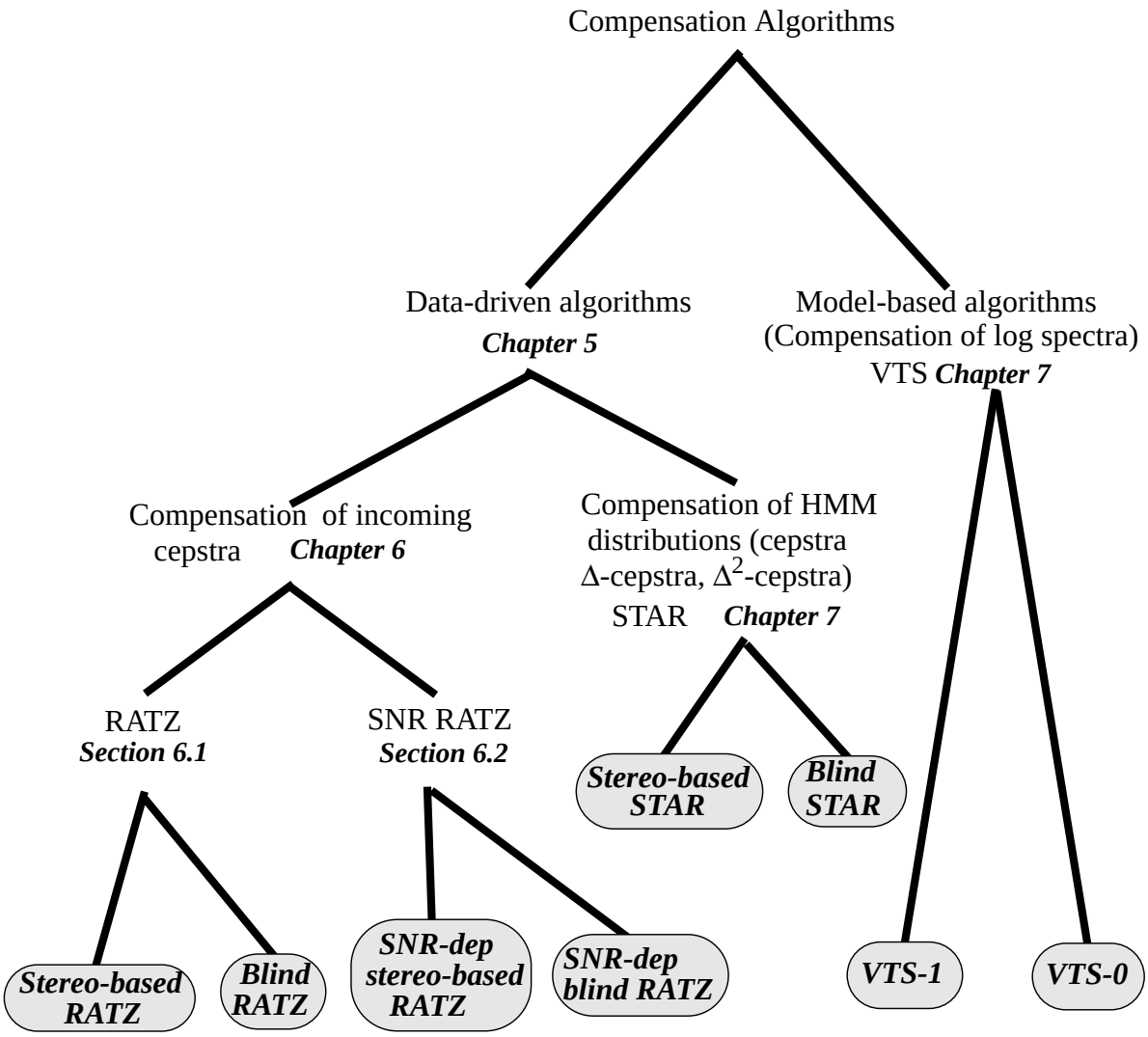

Figure 3-1 Outline of the algorithms for environment compensation presented in this thesis.

taneously-recorded clean and noisy speech data. In the case of RATZ we also explore the effect of SNR-dependent structures in modeling the distributions of clean speech cepstrum. In the case of STAR we explore the benefits of compensation of the distributions of cepstra of clean speech.

For the model-based VTS methods we will explore the use of vector Taylor series approximation to the complex analytical function describing the environments.

Chapter 4
Effects of the Environment on Distributions of Clean Speech

In this chapter we describe and discuss several sources of degradation that the environment imposes on speech recognition systems. We will analyze how these sources of degradation affect the statistics of clean speech and how they impact on speech recognition accuracy. We will also explain why speech recognition systems degrade in performance in the presence of unknown environments. Finally, based on this explanation we propose two generic solutions to the problem.

4.1. A Generic Formulation

Throughout this thesis we will assume that any environment can be characterized by the following equation

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{....}) \tag{4.1}$$

where $\boldsymbol{x}$ represents the clean speech log spectral or cepstral vector that characterize the speech, and $\boldsymbol{a}_1$, $\boldsymbol{a}_2$ and so on represent parameters (vectors, scalars, matrices,...) that define the environment.

While this generic mathematical formulation can be particularized for many cases, in the rest of this section we present a detailed analysis for the case of **convolutional and additive noise**. In this case we can assume the environment can be modeled as represented in Figure 4-1. This kind

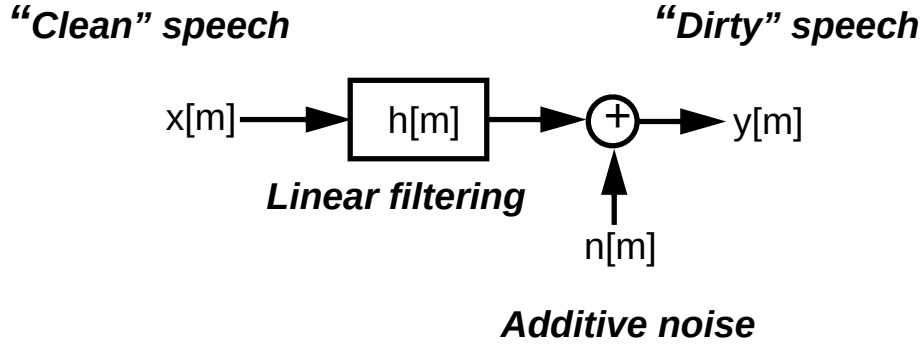

Figure 4-1: A model of the environment for additive noise and filtering by a linear channel. $x[m]$ represents the clean speech signal, $n[m]$ represents the additive noise and $y[m]$ represents the resulting noisy speech signal. $h[m]$ represents a linear channel

of environment was originally proposed by Acero [1] and later used by Liu [35] and Gales [15]. It is a reasonable model of the environment.

The effect of the noise and filtering on clean speech in the power spectral domain can be represented as

$$P_Y(\omega_k) = \left|H(\omega_k)\right|^2 P_X(\omega_k) + P_N(\omega_k) \tag{4.2}$$

where $P_Y(\omega_k)$ represents the spectra of the noisy speech $y[m]$, $P_N(\omega_k)$ represents the power spectra of the noise $n[m]$, $P_X(\omega_k)$ the power spectra of the clean speech $x[m]$, $\left|H(\omega_k)\right|^2$ the power spectra of the channel $h[m]$, and ω_k represents a particular mel-spectral band.

To transform to the log spectral domain we apply the logarithm operator at both sides of the expression (4.2) resulting in

$$10\ \log_{10}(P_Y(\omega_k)) = 10\ \log_{10}(\left|H(\omega_k)\right|^2 P_X(\omega_k) + P_N(\omega_k)) \tag{4.3}$$

and defining the noisy speech, noise, and clean speech,

$$\begin{aligned} y[k] &= 10\ \log_{10}(P_Y(\omega_k)) \\ n[k] &= 10\ \log_{10}(P_N(\omega_k)) \\ x[k] &= 10\ \log_{10}(P_X(\omega_k)) \\ h[k] &= 10\ \log_{10}(\left|H(\omega_k)\right|^2) \end{aligned} \tag{4.4}$$

results in equations

$$\begin{aligned} 10\ \log_{10}(P_Y(\omega_k)) &= 10\ \log_{10}\left(10^{\frac{x[k]+h[k]}{10}} + 10^{\frac{n[k]}{10}}\right) \\ y[k] &= x[k] + h[k] + 10\ \log_{10}\left(1 + 10^{\frac{n[k]-x[k]-h[k]}{10}}\right) \end{aligned} \tag{4.5}$$

Where $h[k]$ is the logarithm of $\left|H(\omega_k)\right|^2$, and similar relationships exist between $n[k]$ and $P_N(\omega_k)$, $x[k]$ and $P_X(\omega_k)$, and $y[k]$ and $P_Y(\omega_k)$.

Following our initial formulation proposed in Equation (4.1) for the case of additive noise and linear channel this expression can be written as

$$y[k] = x[k] + g(x[k], h[k], n[k]) \tag{4.6}$$

or in vector form

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{n}) \tag{4.7}$$

where

$$g(x[k], h[k], n[k]) = h[k] + 10\log_{10}\left(1 + 10^{\frac{n[k]-x[k]-h[k]}{10}}\right) \tag{4.8}$$

From these equations we can formulate a relationship between the log spectral features representing clean and noisy speech. However, the relation is cumbersome and not so easy to understand. A simple way to understand how noise and channel affect speech is by observing how the statistics of clean speech are transformed by the environment.

Let's assume that the log spectral vectors that characterize clean speech follow a Gaussian distribution $N_x(\mu_x, \Sigma_x)$ and that the noise and channel are perfectly known. These circumstances produce a transformation of random variables leading to a new distribution for the log spectra of noisy speech equal to

$$p(\boldsymbol{y}|\mu_{\boldsymbol{x}}, \Sigma_{\boldsymbol{x}}, \boldsymbol{n}, \boldsymbol{h}) = \left\{(2\pi|\Sigma_{\boldsymbol{x}}|)^{L/2}\left|\left(I - 10^{\frac{\boldsymbol{n}-\boldsymbol{y}}{10}}\right)\right|\right\}^{-1}$$
$$e^{-\frac{1}{2}\left(\boldsymbol{y}-\boldsymbol{h}-\mu_{\boldsymbol{x}}+10\log_{10}\left(\boldsymbol{i}-10^{\frac{\boldsymbol{n}-\boldsymbol{y}}{10}}\right)\right)^T\Sigma_x^{-1}\left(\boldsymbol{y}-\boldsymbol{h}-\mu_{\boldsymbol{x}}+10\left(10\log_{10}\left(\boldsymbol{i}-10^{\frac{\boldsymbol{n}-\boldsymbol{y}}{10}}\right)\right)\right)} \tag{4.9}$$

where L is the dimensionality of the log spectral vector random variable, i is the unitary vector, and I is the identity matrix.

The resulting distribution $p(\boldsymbol{y})$ is clearly non-Gaussian. However, since most speech recognition systems assume Gaussian distributions, a Gaussian distribution assigned to $p(\boldsymbol{y})$ can still capture part of the effect of the environment on speech statistics. To characterize the Gaussian distributions we need only compute the mean vector and covariance matrices of these new distributions. The new mean vector can be computed as

$$\begin{aligned} \mu_y &= E(\boldsymbol{x} + \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{n})) = \mu_{\boldsymbol{x}} + E(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{n})) \\ \mu_{\boldsymbol{y}} &= \mu_{\boldsymbol{x}} + \int_{\boldsymbol{X}} \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{n}) N_x(\mu_{\boldsymbol{x}}, \Sigma_{\boldsymbol{x}}) d\boldsymbol{x} \\ \mu_{\boldsymbol{y}} &= \mu_{\boldsymbol{x}} + \boldsymbol{h} + \int_{\boldsymbol{X}} 10 \log_{10}\left(\boldsymbol{i} + 10^{\frac{\boldsymbol{n}-\boldsymbol{x}-\boldsymbol{h}}{10}}\right) N_x(\mu_{\boldsymbol{x}}, \Sigma_{\boldsymbol{x}}) d\boldsymbol{x} \end{aligned} \tag{4.10}$$

and the covariance matrix can be computed as

$$\begin{aligned} \Sigma_{\boldsymbol{y}} &= E((\boldsymbol{x} + \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{n}))(\boldsymbol{x} + \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{n}))^T) - \mu_{\boldsymbol{y}}\mu_{\boldsymbol{y}}^T \\ \Sigma_{\boldsymbol{y}} &= E(\boldsymbol{x}\boldsymbol{x}^T) + E(\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{n})\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{n})^T) + 2E(\boldsymbol{x}\ \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{n})^T) - \mu_{\boldsymbol{y}}\mu_{\boldsymbol{y}}^T \\ \Sigma_{\boldsymbol{y}} &= \Sigma_{\boldsymbol{x}} + \mu_{\boldsymbol{x}}\mu_{\boldsymbol{x}}^T + \int_{\boldsymbol{X}} (\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{n})\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{n})^T) N_{\boldsymbol{x}}(\mu_{\boldsymbol{x}}, \Sigma_{\boldsymbol{x}}) d\boldsymbol{x} + \\ &\quad + 2\int_{\boldsymbol{X}} (\boldsymbol{x}\ \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{n})^T) N_{\boldsymbol{x}}(\mu_{\boldsymbol{x}}, \Sigma_{\boldsymbol{x}}) d\boldsymbol{x} - \mu_{\boldsymbol{y}}\mu_{\boldsymbol{y}}^T \end{aligned} \tag{4.11}$$

In both equations the integrals don't have a closed form solution. Therefore we must use numerical methods to estimate the mean vector and covariance matrix of the distribution.

Since in most cases the noise must be estimated and is not known *a priori*, a more realistic model would be to assign a Gaussian distribution $N_{\boldsymbol{n}}(\mu_{\boldsymbol{n}}, \Sigma_{\boldsymbol{n}})$ to the noise. To simplify the resulting equations we can also assume that the noise and the speech are statistically independent. The probability density function (pdf) of the log spectrum of the noisy speech under these assumptions cannot be computed analytically, but it can be estimated using Monte-Carlo methods.

The mean vector and the covariance matrix of the log spectrum of the noisy speech will have the form

$$\mu_{\boldsymbol{y}} = \mu_{\boldsymbol{x}} + \int_{\boldsymbol{X}} N_{\boldsymbol{x}}(\mu_{\boldsymbol{x}}, \Sigma_{\boldsymbol{x}}) \int_{\boldsymbol{N}} \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{n}) N_{\boldsymbol{n}}(\mu_{\boldsymbol{n}}, \Sigma_{\boldsymbol{n}}) \boldsymbol{dx dn} \tag{4.12}$$

$$\begin{aligned}\Sigma_{\boldsymbol{y}} = \Sigma_{\boldsymbol{x}} + \mu_{\boldsymbol{x}}\mu_{\boldsymbol{x}}^{T} + \int_{\boldsymbol{X}} N_{\boldsymbol{x}}(\mu_{\boldsymbol{x}}, \Sigma_{\boldsymbol{x}}) \int_{\boldsymbol{N}} (\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{n})\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{n})^{T}) N_{\boldsymbol{n}}(\mu_{\boldsymbol{n}}, \Sigma_{\boldsymbol{n}}) \boldsymbol{dn dx} + \\ + 2\int_{\boldsymbol{X}} N_{\boldsymbol{x}}(\mu_{\boldsymbol{x}}, \Sigma_{\boldsymbol{x}}) \int_{\boldsymbol{N}} (\boldsymbol{x}\ \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{n})^{T}) N_{\boldsymbol{n}}(\mu_{\boldsymbol{n}}, \Sigma_{\boldsymbol{n}}) \boldsymbol{dn dx} - \mu_{\boldsymbol{y}}\mu_{\boldsymbol{y}}^{T}\end{aligned} \tag{4.13}$$

Again, as in the previous case the resulting equations have no closed-form solution, and we can only estimate the resulting mean vector and covariance matrix through numerical methods.

4.2. One dimensional simulations using artificial data

To visualize the resulting distributions of noisy data we present results obtained with artificially produced one-dimensional data. These artificial data can simulate the simplified case of a log spectrum feature vector of speech using a single dimension.

Simulated clean data were produced according to a Gaussian distribution $N_x(\mu_x, \sigma_x^2)$ and contaminated with artificially produced noise according to a Gaussian distribution $N_n(\mu_n, \sigma_n^2)$. A channel h was also defined. The artificially-produced clean data, noise, and channel were combined according to Equation (4.5) producing a noisy data set $Y = \{y_0, y_1, ..., y_{N-1}\}$. From this noisy data set we directly estimated the mean and variance of a maximum likelihood (ML) fit as

$$\mu_{y,\,ML} = \frac{1}{N}\sum_{i=0}^{N-1} y_i \qquad \sigma_{y,\,ML}^2 = \frac{1}{N}\sum_{i=0}^{N-1} (y_i - \mu_{y,\,ML})^2 \tag{4.14}$$

We also computed a histogram of the noisy data set to estimate directly its real distribution. The contamination was performed at different speech-to-noise ratios defined by $\mu_x - \mu_n$.

Figure 4-2 shows an example of the original distribution of the clean signal, the original noisy signal after going through a transformation of Equation (4.5) and producing the distribution of

Equation (4.9), and the best Gaussian fit to the distribution of the noisy signal. The SNR for the noisy signal is 3 dB.

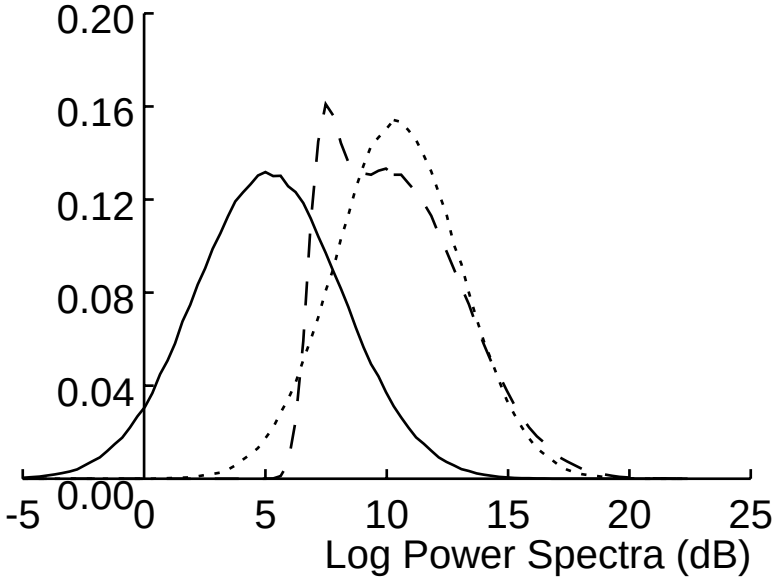

Figure 4-2: Estimate of the distribution of noisy data via Monte-Carlo simulations. The continuous line represents the pdf of the clean signal. The dashed line represents the real pdf of the noise-contaminated signal. The dotted line represents the Gaussian-approximated pdf of the noisy signal. The original clean signal had a mean of 5.0 and a variance of 3.0, the channel was set to 5.0. The mean of the noise was set to 7.0 and the variance of the noise was set to 0.5.

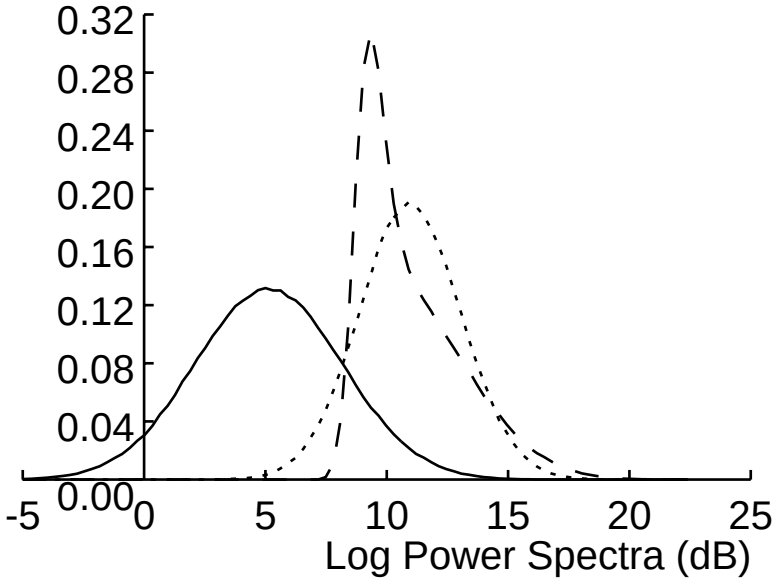

Figure 4-3: Estimate of the distribution of noisy signal at a lower SNR level via Monte-Carlo methods. The continuous line represents the pdf of the clean signal. The dashed line represents the real pdf of the noise-contaminated signal. The dotted line represents the Gaussian-approximated pdf of the noisy signal. The original clean signal had a mean of 5.0 and a variance of 3.0. Channel was set to 5.0. The mean of the noise was set to 9.0 and the variance of the noise was set to 0.5.

Figure 4-3 and Figure 4-4 show similar results as those of Figure 4-2 but with different SNRs.

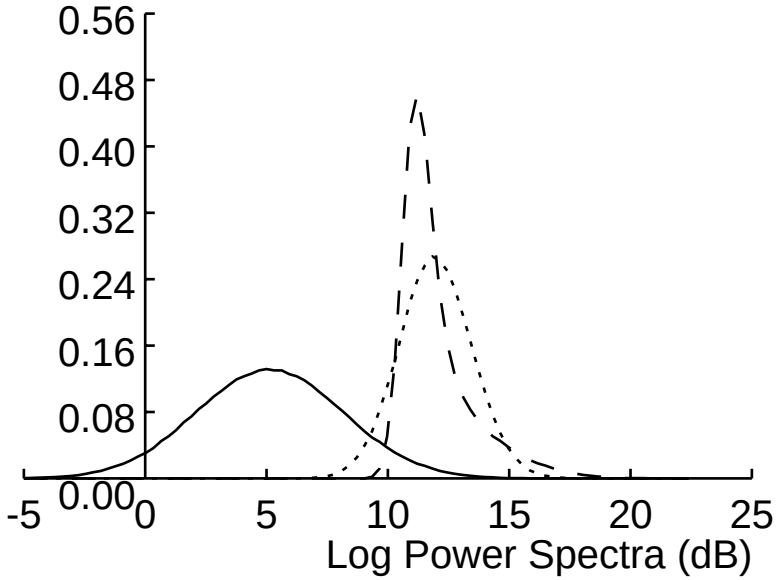

Figure 4-4:Estimate of the distribution of noisy signal at a lower SNR level via Monte-Carlo methods. The continuous line represents the pdf of the clean signal. The dashed line represent the real pdf of the noise-contaminated signal. The dotted line represents the Gaussian-approximated pdf of the noisy signal. The original clean signal had a mean of 5.0 and a variance of 3.0. The channel was set to 5.0. The mean of the noise was set to 11.0 and the variance of the noise was set to 0.5.

The noisy signal in Figure 4-3 had a SNR of 1 dB and in Figure 4-4 it had a SNR of -1 dB.

We can see how for some cases (*e.g*. Figure 4-2) the pdf of the resulting noisy signal can be bimodal and clearly non-Gaussian. However, if the noise mean is higher (9.0) (*e.g.* Figure 4-2) the bimodality of the resulting noisy signal pdf is lost and we can also observe the compression of the resulting noisy signal distribution. We also see that a Gaussian fit to the noisy signal captures some of the effect of this particular environment on the clean signal.

In general, the effect of this particular type of environment on speech statistics can be reasonably accurately modelled as a **shift** in the mean of the pdfs and a **decrease** in the variance of the resulting pdf. Notice, however, that this compression of the variance will happen only if the variance of the distribution of the noise is smaller than the variance of the distribution of the clean signal. The change in variance can be represented by a additive factor in the covariance matrix.

4.3. Two dimensional simulations with artificial data

Speech representation are normally multidimensional. In the case of the SPHINX-II system a 40-dimensional log spectral representation is created which is then transformed to a 13-dimension-

al cepstral representation. Because of the inherent multidimensionality of speech representations, it is worthwhile to visualize how the noise affects the multidimensional statistics of speech. Since log spectral features are highly correlated, the behavior of frequency band ω_i is very similar to that of bands ω_{i+1} and ω_{i-1}. As a result the covariance matrix of these features is non-diagonal.

A simple way to visualize the effects of noise on correlated distributions of speech features is to use a simplified representation with 2 dimensions. In the following sections we repeat the same simulations but with a set of simulated 2-dimensional artificial data where both the covariance matrices of both the signal and noise are non-diagonal.

Figure 4-5 shows the pdf of the clean signal assuming a two-dimensional Gaussian distribution

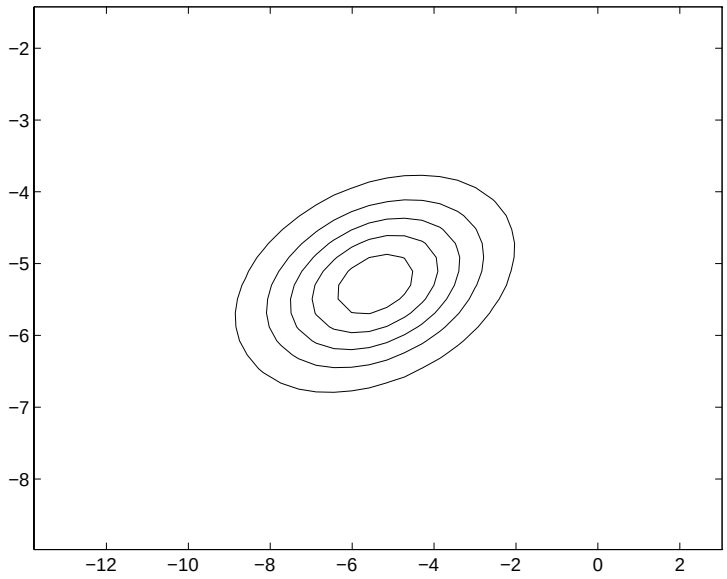

Figure 4-5: Contour plot of the distribution of the clean signal.

with mean and covariance matrix

$$\mu_{\boldsymbol{x}} = \begin{bmatrix} -5 \\ -5 \end{bmatrix} \qquad \Sigma_{\boldsymbol{x}} = \begin{bmatrix} 3 & 0.5 \\ 0.5 & 6 \end{bmatrix} \tag{4.15}$$

This signal is passed through a channel $\boldsymbol{h}$ and added to a noise with mean vector and covari-

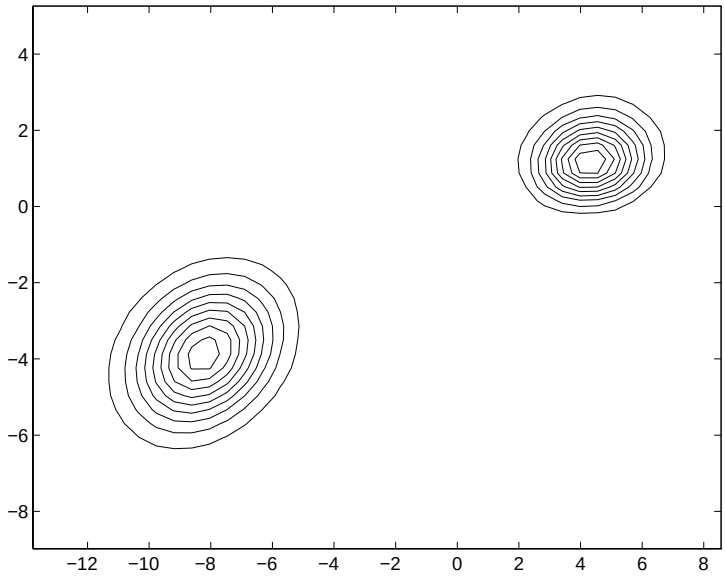

Figure 4-6: Contour plot of the distribution of the clean signal and of the noisy signal.

ance matrix

$$\boldsymbol{h} = \begin{bmatrix} 5 \\ 5 \end{bmatrix} \qquad \mu_{\boldsymbol{n}} = \begin{bmatrix} 5 \\ 5 \end{bmatrix} \qquad \Sigma_{\boldsymbol{n}} = \begin{bmatrix} 0.05 & -0.01 \\ -0.01 & 0.025 \end{bmatrix} \tag{4.16}$$

Figure 4-6 shows the pdf of the resulting noisy signal. As we can see the resulting distribution has been **shifted** and **compressed**. A maximum likelihood (ML) Gaussian fit to the resulting noisy data yields these estimates for the mean vector and covariance matrix

$$\mu_{\boldsymbol{y},\,ML} = \begin{bmatrix} 6.24 \\ 6.30 \end{bmatrix} \qquad \Sigma_{\boldsymbol{y},\,ML} = \begin{bmatrix} 0.35 & 0.006 \\ 0.006 & 0.53 \end{bmatrix} \tag{4.17}$$

The above conclusions apply for the case in which the variance of the distribution of the noise is smaller than the variance of the distribution of the clean signal. If the clean signal has a very narrow distribution or if the noise distribution is very wide, we will observe an **expansion** of the pdf of the resulting noisy signal.

4.4. Modeling the effects of the environment as correction factors

From the one- and two-dimensional simulations with artificial data sets we can clearly observe the following effects

- The pdf of the signal is shifted according to the SNRs.
- The pdf of the signal is expanded if $\Sigma_{\boldsymbol{x}} < \Sigma_{\boldsymbol{n}}$ or compressed if $\Sigma_{\boldsymbol{x}} > \Sigma_{\boldsymbol{n}}$ and this expansion/ compression depends on the SNR.
- The pdf of the noisy signal are clearly non Gaussian, under particular conditions they exhibit a bimodal shape.

However, since most speech recognition systems model speech statistics as mixtures of Gaussians it is still convenient to keep modelling these resulting noisy speech pdfs as Gaussians. A simple way to achieve this is by

- modelling the mean of the noisy speech distribution as the mean of the clean signal plus a correction vector

$$\mu_{\boldsymbol{y}} = \mu_{\boldsymbol{x}} + \boldsymbol{r} \tag{4.18}$$

- modelling the covariance matrix of the noisy speech distribution as the covariance matrix of the clean speech plus a correction covariance matrix

$$\Sigma_{\boldsymbol{y}} = \Sigma_{\boldsymbol{x}} + \boldsymbol{R} \tag{4.19}$$

The $\boldsymbol{R}$ matrix will be symmetric and will have positive or negative elements according to the value of the covariance matrix of the noise compared with that of the clean signal.

This approach will be extensively used in the RATZ family of algorithms (Chapter 6).

Another alternative is to model the environment effects by attempting to solve some of the equations presented in this chapter via Taylor series approximations (*e.g.* (4.10), (4.11), (4.12) and (4.13)). This kind of approach will be exploited in the VTS family of algorithms (Chapter 8).

4.5. Why do speech recognition system degrade in performance in the presence of unknown environments?

For completeness it is helpful to review some of the ways in which the degradations to the statistics of speech described earlier in this chapter can degrade recognition accuracy.

We frame the speech recognition problem as one of pattern classification [12]. The simplest type of pattern classification is illustrated in Figure 4-7. In this case we assume two classes, H_1 and H_2, each represented by a single Gaussian distribution, and each with equal *a priori* probabilities

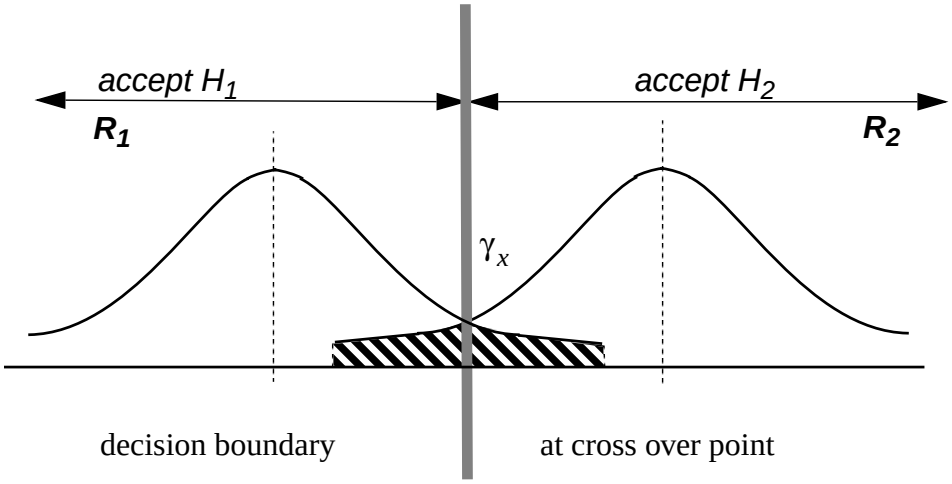

Figure 4-7: Decision boundary for a single two-class classification problem. The shaded region represents the probability of error. An incoming sample x_i will be classified as belonging to class H_1 or H_2 comparing it to the decision boundary γ_x. If x_i is less than γ_x it will be classified as belonging to class H_1, otherwise it will be classified as belonging to class H_2.

and variances. In this case the maximum *a posteriori* (MAP) decision rule is expressed as a ratio of likelihoods

$$\begin{aligned} &\text{choose } H_1 \text{ if} \qquad P[class{=}H_1 \,|\, x] \geq P[class{=}H_2 \,|\, x] \\ &\text{choose } H_2 \text{ if} \qquad P[class{=}H_1 \,|\, x] \leq P[class{=}H_2 \,|\, x] \end{aligned} \tag{4.20}$$

Solving the previous equation will yield a decision boundary of the form

$$\gamma_x = \frac{\mu_{x,H_1} + \mu_{x,H_2}}{2} \tag{4.21}$$

this decision boundary is guaranteed to minimize the probability of error P_e, and therefore it provides the optimal classifier. It specifies in effect a decision boundary that will be used by the system to classify incoming signals. However, as we have seen, the effects of the environment on that data are three fold:

- The resulting distributions change shape, becoming non-Gaussian.
- Even assuming a Gaussian shape, the means of the noisy distributions are shifted.
- Even assuming a Gaussian shape, the covariance matrices of the noisy distributions are compressed or expanded depending on the relation between noise and clean signal covariance

matrices.

For every particular environment the distributions of the noisy data change, and therefore the optimal decision boundaries also change. We call the optimal noisy decision boundary γ_y. If the classification is done using a decision boundaries γ_x that had been derived on the basis of the statistics of the clean signal, the result is suboptimal in that the minimal probability of error will not be obtained.

Figure 4-8 illustrates how the error region is composed of two areas, the optimal error region

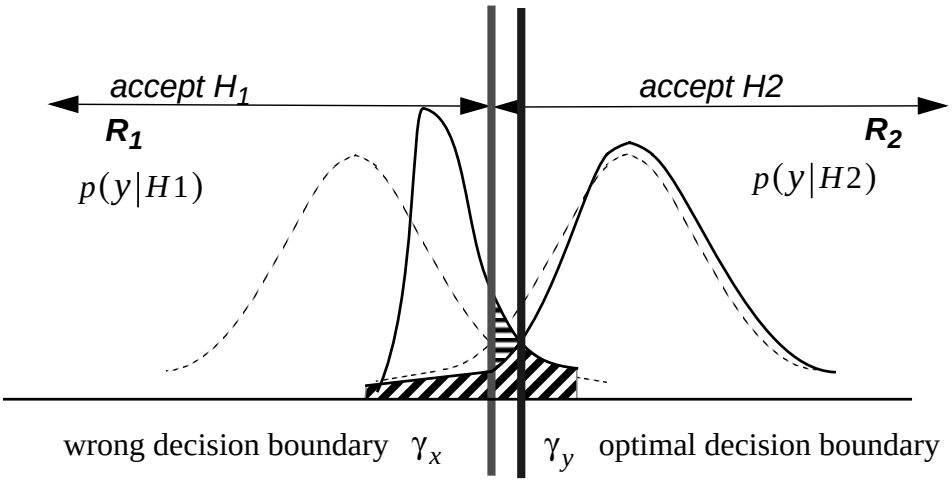

Figure 4-8: When the classification is performed using the wrong decision boundaries, the error region is composed of two terms, the optimal one assuming the optimal decision boundary is known (banded area above), and an addition term introduced by using the wrong decision boundary (shaded are above between γ_x and γ_y).

that would be obtained using the γ_y decision boundary, and a secondary error surface that is produced using the γ_x decision boundary.

This explanation suggests two possible ways to compensate for the effects of the environment on speech statistics:

- Modify the statistics of the speech recognition system (mean vectors and covariance matrices) to make them more similar to those of the incoming noisy speech. In other words, make sure that the “classifiers” as represented by the HMMs, use the optimal decision boundaries. This will be the **optimal** solution as it guarantees minimal probability of error.
- Modify the incoming noisy speech data to produce a pseudo-clean speech data set whose distributions resemble as much as possible the clean speech distributions. This solution does not modify any of the parameters (mean vectors and covariance matrices) of the HMMs. In ad-

dition, this will be a suboptimal solution since at the most it can yield similar results to the previous solution.

Notice however that the first approach would imply the use of non-Gaussian distributions and the solution of difficult equations. Therefore only approximations to the first approach will be proposed in this thesis.

In this thesis we provide several algorithms that attempt to approximate both approaches. In addition, Appendix A illustrates the difference between both types of approaches and discusses why algorithms that modify the statistics seem to perform better than those that attempt to compensate the incoming noisy speech data.

4.6. Summary

In this chapter we have analyzed the effect of the environment on the statistics of clean speech, considering the particular case of additive noise and linear filtering. We have provided several simulations using one- and two-dimensional data as a tool to explore how speech statistics are modified by the environment. From these experiments we have concluded that the effects of the environment on speech statistics are:

- The resulting distributions are not always Gaussian.
- The means of the resulting distributions are shifted.
- The covariance matrices of the resulting distributions are compressed or expanded.

As we have already mentioned the compression of the variances really depends on whether the variance of the distribution of the noise is smaller than that of the clean signal. In our experiments with real data we have observed the compression described in the simulation presented in this chapter. However, there might be situations in which this does not happen. The model of environmental degradation proposed in this chapter does indeed allow for an expansion of the variance.

It is also important to mention the fact that the resulting non-Gaussian distributions of the noisy signal are not necessarily problematical for speech modeling. The feature vectors representing the clean speech signal are probably not Gaussian in nature to begin with. We just fit a Gaussian model to them for convenience. Therefore applying a Gaussian model to the distributions of the feature vectors of noisy speech is as valid as when done with clean speech feature vectors.

The simulations described in this chapter have been done assuming log spectra feature vectors. It is important to mention that all the conclusions derived in this chapter are also valid for the case of cepstral feature vector as the relationship between cepstral vectors and log spectral vectors is linear.

Finally, we have introduced an explanation for why speech recognition systems fail in the presence of unknown environments using some examples with one dimensional data.

Chapter 5
A Unified View of Data-Driven Environment Compensation

In the previous chapter we have seen how the effect of the environment on the distributions of the log-spectra or cepstra of clean speech can be modeled as shifts to the mean vectors and compressions or expansions to the covariance matrices, or in more general terms as correction factors applied to the mean vectors and covariance matrices.

In this chapter we present some techniques that attempt to learn these effects directly from sample data. In other words, by observing sets of noisy and clean vectors these techniques try to learn the appropriate correction factors. This approach does not explicitly assume any model of the environment, but uses empirical observations to infer environmental characteristics. Any environment that only affects speech features by shifting their means and compressing their variances will be compensated by the techniques proposed in this chapter.

In previous years several techniques have been proposed to address environmental robustness using direct observation of the environment without structural assumptions about the nature of the degradation. Some of these techniques have attempted to address the problem by applying compensation factors to the cepstrum vectors (FCDCN [1], POF [43]) and others have applied compensation factors to the means and covariances of the distributions of HMMs [35]. However, both kind of approaches have been presented as separate techniques. In this chapter we present a unified view of data-driven compensation methods. We will argue that techniques that attempt to modify the incoming cepstra vectors and techniques that modify the parameters (means and variances) of the distributions of the HMMs are two aspects of the same theory. We will show how both these two approaches to environment compensation share the same basic assumptions and internal structure but differ in whether they modify the incoming cepstra vectors or the classifier statistics.

In this chapter we first introduce a unified view of environmental compensation and then provide solutions for the compensation factors. We then particularize these solutions for the case of stereo adaptation data. After this we particularize the generic solutions for two family of techniques; the Multivariate Gaussian Based Cepstral Normalization (RATZ) techniques [39, 40], and the Statistical Reestimation (STAR) techniques [40, 41]. Their performance for several databases

and experimental conditions is compared in the next chapters.

5.1. A unified view

We model the distribution of the t^{th} vector $\boldsymbol{x}_t$ of a cepstral vector sequence of length T, $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_T\}$, generically as

$$p(\boldsymbol{x}_t) = \sum_{k=1}^{K} a_k(t) N_{\boldsymbol{x}}(\mu_{\boldsymbol{x},k}, \Sigma_{\boldsymbol{x},k}) \tag{5.1}$$

i.e., as a summation of K Gaussian components with *a priori* probabilities that are time dependent. Assuming that each vector $\boldsymbol{x}_t$ is independent and identically distributed (i.i.d.) the overall likelihood for the full observation sequence $\boldsymbol{X}$ becomes

$$l(\boldsymbol{X}) = \prod_{t=1}^{T} p(\boldsymbol{x}_t) = \prod_{t=1}^{T} \sum_{k} a_k(t) N_{\boldsymbol{x}}(\mu_{\boldsymbol{x},k}, \Sigma_{\boldsymbol{x},k}) \tag{5.2}$$

The above likelihood equation offers a double interpretation. For the methods that modify the incoming features, we set the *a priori* probabilities $a_k(t)$ to be independent of t; this defines a conventional mixture of Gaussian distributions for the entire training set of cepstral vectors. Another possible interpretation is that the cepstral speech vectors are represented by a single HMM state with K Gaussians that transitions to itself with probability unity.

The interpretation is slightly different for the methods that modify the distributions representing speech (HMMs). We assume that the cepstral speech vectors are emitted by a HMM with K states in which each state emission p.d.f. is composed of a single Gaussian. In this case the $a_k(t)$ terms define the probability of being in state k at time t. Under these assumptions the expression of the likelihood for the full observation sequence is exactly as expressed in Equation (5.2).

The assumption of a single Gaussian per state is not limiting at all. Specifically, any state with a mixture of Gaussians for emission probabilities can also be represented by multiple states where the output distributions are single Gaussians and where the incoming transition probabilities of the state are the same as the *a priori* $a_k(t)$ probabilities of the Gaussians and the exiting transition probability is unity. The next figure illustrates this idea

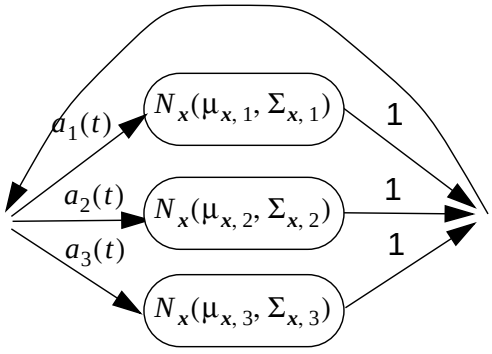

Figure 5-1A state with a mixture of Gaussians is equivalent to a set of states where each of them contains a single Gaussian and the transition probabilities are equivalent to the *a priori* probabilities of each of the mixture Gaussians.

These probabilities $a_k(t)$ depend only on the Markov chain topology and are represented in the form

$$a(t) = \left[a_1(t)\ a_2(t)\ ....\ a_K(t)\right]^T = A^t\pi \tag{5.3}$$

where A represents the transition matrix, A^t represents the transition matrix after t transitions and π represents the initial state probability vector of the HMM. The $N_x(\mu_{x,k}, \Sigma_{x,k})$ terms of Equation (5.1) refer to the Gaussian densities associated with each of the K states of the HMM.

As we have mentioned before the changes to the mean vectors and covariance matrices can be expressed as

$$\mu_{y,k} = \boldsymbol{r}_k + \mu_{x,k} \qquad \Sigma_{y,k} = \boldsymbol{R}_k + \Sigma_{x,k} \tag{5.4}$$

where $\boldsymbol{r}_k$ and $\boldsymbol{R}_k$ represent the corrections applied to the mean vector and covariance matrix respectively of the k^{th} Gaussian. These two correction factors account for the effect of the environment on the distributions of the cepstra of clean speech. Finding these two correction factors will be the first step of the RATZ and STAR algorithms.

5.2. Solutions for the correction factors $\boldsymbol{r}_k$ and $\boldsymbol{R}_k$

The solutions for the correction factors will depend upon the availability of stereo data, i.*e.*, simultaneous recordings of clean and noisy adaptation data. We first describe the generic solution for the case in which only samples of noisy speech are available, the so-called “blind” case. We

then describe how to particularize these solutions for the stereo case.

In this section we make extensive use of the EM algorithm [13]. Our goal is not to describe the EM algorithm itself but to show its use in the solution for the correction parameters $\boldsymbol{r}_k$ and $\boldsymbol{R}_k$. References [21] and [13] give a detailed and full explanation of the EM algorithm.

5.2.1. Non-stereo-based solutions

We begin with an observed set of T noisy vectors $\boldsymbol{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_T\}$, and assuming that these vectors have been produced by a probability density function

$$p(\boldsymbol{y}_t) = \sum_{k=1}^{K} a_k(t) N_{\boldsymbol{y}}(\mu_{\boldsymbol{y},k}, \Sigma_{\boldsymbol{y},k}) \tag{5.5}$$

which is a summation of K Gaussians where each component relates to the corresponding k^{th} Gaussian of clean speech according to Equation (5.4). We define a likelihood function $l(\boldsymbol{Y})$ as

$$l(\boldsymbol{Y}) = \prod_{t=1}^{T} p(\boldsymbol{y}_t) = \prod_{t=1}^{T} \sum_{k} a_k(t) N_{\boldsymbol{y}}(\mu_{\boldsymbol{y},k}, \Sigma_{\boldsymbol{y},k}) \tag{5.6}$$

We can also express $l(\boldsymbol{Y})$ in terms of the original parameters of clean speech and the correction terms $\boldsymbol{r}_k$ and $\boldsymbol{R}_k$

$$l(\boldsymbol{Y}) = l(\boldsymbol{Y}|\boldsymbol{r}_1, ..., \boldsymbol{r}_K, \boldsymbol{R}_1, ..., \boldsymbol{R}_K) = \prod_{t=1}^{T} p(\boldsymbol{y}_t) = \prod_{t=1}^{T} \sum_{k} a_k(t) N_{\boldsymbol{y}}(\boldsymbol{r}_k + \mu_{\boldsymbol{x},k}, \boldsymbol{R}_k + \Sigma_{\boldsymbol{x},k}) \tag{5.7}$$

For convenience we express the above equation in the logarithm domain defining the log likelihood $L(\boldsymbol{Y})$ as

$$L(\boldsymbol{Y}) = \log(l(\boldsymbol{Y})) = \sum_{t=1}^{T} \log(p(\boldsymbol{y}_t)) = \sum_{t=1}^{T} \log\left(\sum_{k} a_k(t) N_{\boldsymbol{y}}(\boldsymbol{r}_k + \mu_{\boldsymbol{x},k}, \boldsymbol{R}_k + \Sigma_{\boldsymbol{x},k})\right) \tag{5.8}$$

Our goal is to find the complete set of K terms $\boldsymbol{r}_k$ and $\boldsymbol{R}_k$ that maximize the likelihood (or log likelihood). As it turns out there is no direct solution to this problem and some indirect method is necessary. The Expectation-Maximization (EM) algorithm is one of this methods.

The EM algorithm defines a new auxiliary function $Q(\phi, \bar{\phi})$ as

$$Q(\phi, \bar{\phi}) = E[L(\boldsymbol{Y}, S|\bar{\phi})\big|\boldsymbol{Y}, \phi] \tag{5.9}$$

where the $(\boldsymbol{Y}, S)$ pair represent the *complete* data, composed of the *observed* data $\boldsymbol{Y}$ (the noisy vectors) and the *unobserved* data S (indicating which Gaussian/state produced an observed data vector). This equation can be easily related to the Baum-Welch equations used in Hidden Markov Modelling. The ϕ symbol represents the set of parameters (K correction vectors and K correction matrices) that maximize the observed data

$$\phi = \{\boldsymbol{r}_1, ..., \boldsymbol{r}_K, \boldsymbol{R}_1, ..., \boldsymbol{R}_K\} \tag{5.10}$$

The $\bar{\phi}$ symbol represents the same set of parameters as ϕ but with different values. The basis of the EM algorithm lies in the fact that given two sets of parameters ϕ and $\bar{\phi}$, if $Q(\phi, \bar{\phi}) \geq Q(\phi, \phi)$, then $L(\boldsymbol{Y}, \bar{\phi}) \geq L(\boldsymbol{Y}, \phi)$. In other words, maximizing $Q(\phi, \bar{\phi})$ with respect to the ϕ parameters is guaranteed to increase the likelihood $L(\boldsymbol{Y}, \bar{\phi})$.

Since the unobserved data S are represented by a discrete random variable (the mixture index in our case), Equation (5.9) can be expanded as

$$Q(\phi, \bar{\phi}) = E[L(\boldsymbol{Y}, S|\bar{\phi})\big|\boldsymbol{Y}, \phi] = \sum_{t=1}^{T}\sum_{k=1}^{K} \frac{p(\boldsymbol{y}_t, s_t(k)\big|\phi)}{p(\boldsymbol{y}_t\big|\phi)} \log(p(\boldsymbol{y}_t, s_t(k)\big|\bar{\phi})) \tag{5.11}$$

hence

$$\begin{aligned} Q(\phi, \bar{\phi}) = \sum_{t=1}^{T}\sum_{k=1}^{K} P[s_t(k)\big|\boldsymbol{y}_t, \phi]\{\log a_k(t) - \frac{L}{2}\log(2\pi) - \frac{L}{2}\log\big|\overline{\boldsymbol{R}_k} + \Sigma_{\boldsymbol{x},k}\big| + \\ -\frac{1}{2}(\boldsymbol{y}_t - \mu_{\boldsymbol{x},k} - \overline{\boldsymbol{r}_k})^T(\Sigma_{\boldsymbol{x},k} + \overline{\boldsymbol{R}_k})^{-1}(\boldsymbol{y}_t - \mu_{\boldsymbol{x},k} - \overline{\boldsymbol{r}_k})\} \end{aligned} \tag{5.12}$$

where L is the dimensionality of the cepstrum vector. The expression can be further simplified to

$$\begin{aligned} Q(\phi, \bar{\phi}) = \text{constant} + \sum_{t=1}^{T}\sum_{k=1}^{K} P[s_t(k)\big|\boldsymbol{y}_t, \phi]\{-\frac{L}{2}\log\big|\overline{\boldsymbol{R}_k} + \Sigma_{\boldsymbol{x},k}\big| + \\ -\frac{1}{2}(\boldsymbol{y}_t - \mu_{\boldsymbol{x},k} - \overline{\boldsymbol{r}_k})^T(\Sigma_{\boldsymbol{x},k} + \overline{\boldsymbol{R}_k})^{-1}(\boldsymbol{y}_t - \mu_{\boldsymbol{x},k} - \overline{\boldsymbol{r}_k})\} \end{aligned} \tag{5.13}$$

To find the ϕ parameters we simply take derivatives and set equal to zero,

$$\begin{aligned} \frac{d}{d\bar{\boldsymbol{r}}_k} Q(\phi, \bar{\phi}) &= \sum_{t=1}^{T} P[s_t(k)|\boldsymbol{y}_t, \phi](\Sigma_{\boldsymbol{x},k} + \overline{\boldsymbol{R}_k})^{-1}(\boldsymbol{y}_t - \mu_{\boldsymbol{x},k} - \overline{\boldsymbol{r}_k}) = 0 \\ \frac{d}{d\overline{\boldsymbol{R}}_k^{-1}} Q(\phi, \bar{\phi}) &= \sum_{t=1}^{T} P(\boldsymbol{y}_t|s_t(k), \phi)\{(\Sigma_{\boldsymbol{x},k} + \overline{\boldsymbol{R}_k}) - (\boldsymbol{y}_t - \mu_{\boldsymbol{x},k} - \overline{\boldsymbol{r}_k})(\boldsymbol{y}_t - \mu_{\boldsymbol{x},k} - \overline{\boldsymbol{r}_k})^T\} = 0 \end{aligned} \tag{5.14}$$

hence

$$\overline{\boldsymbol{r}_k} = \frac{\sum_{t=1}^{T} P[s_t(k)|\boldsymbol{y}_t, \phi]\boldsymbol{y}_t}{\sum_{t=1}^{T} P[s_t(k)|\boldsymbol{y}_t, \phi]} - \mu_{\boldsymbol{x},k} \tag{5.15}$$

$$\overline{\boldsymbol{R}_k} = \frac{\sum_{t=1}^{T} P[s_t(k)|\boldsymbol{y}_t, \phi]((\boldsymbol{y}_t - \mu_{\boldsymbol{x},k} - \overline{\boldsymbol{r}_k})(\boldsymbol{y}_t - \mu_{\boldsymbol{x},k} - \overline{\boldsymbol{r}_k})^T)}{\sum_{t=1}^{T} P[s_t(k)|\boldsymbol{y}_t, \phi]} - \Sigma_{\boldsymbol{x},k} \tag{5.16}$$

Equation (5.15) and Equation (5.16) for the basis of an iterative algorithm. The EM algorithm guarantees that each iteration increases the likelihood of the observed data.

5.2.2. Stereo-based solutions

When simultaneously recorded clean and noisy speech (also called stereo-recorded) adaptation data are available the information about the environment is encoded in the stereo pairs. By observing how each clean speech vector $\boldsymbol{x}_i$ is transformed into a noisy speech vector $\boldsymbol{y}_i$ we can learn the correction factors more directly.

We can readily assume that the *a posteriori* probabilities $P[s_t(k)|\boldsymbol{y}_t, \phi]$ can be directly estimated by $P[s_t(k)|\boldsymbol{x}_t]$. This is equivalent to assuming that the probabilities of a vector being produced by each of the underlying classes do not change due to the environment. We call this assumption ***a posteriori invariance***. This assumption, although it is not strictly correct, seems to

be a good approximation. If we expand the $P[s_t(k)|\boldsymbol{y}_t, \phi]$ and $P[s_t(k)|\boldsymbol{x}_t]$ terms we obtain

$$P[s_t(k)|\boldsymbol{y}_t, \phi] = \frac{P[s_t(k)]p(\boldsymbol{y}_t|s_t(k), \phi)}{\sum_{j=1}^{K} p(\boldsymbol{y}_t|s_t(j), \phi)P[s_t(j)]} \tag{5.17}$$

$$P[s_t(k)|\boldsymbol{x}_t, \phi] = \frac{P[s_t(k)]p(\boldsymbol{x}_t|s_t(k))}{\sum_{j=1}^{K} p(\boldsymbol{x}_t|s_t(j))P[s_t(j)]} \tag{5.18}$$

for the two above expressions to be equal each of the terms in the summation must be equal. This would imply that each Gaussian is shifted exactly the same amount and not compressed. However, at high SNR conditions the shift for each Gaussian is quite similar and the compression in the variances is almost non-existent. Therefore, at high SNR the ***a posteriori invariance*** is almost valid and at lower SNR conditions it is less valid. In addition, this assumption avoids the need to iterate in Equation (5.15) and Equation (5.16).

A second assumption we make is that the $\mu_{x,k}$ term can be replaced by $\boldsymbol{x}_t$. This change has been experimentally proven to improve recognition performance. After these changes the resulting estimates of the correction factors are

$$\boldsymbol{r}_k = \left(\sum_{t=1}^{T} P[s_t(k)|\boldsymbol{x}_t](\boldsymbol{y}_t - \boldsymbol{x}_t)\right)\left(\sum_{t=1}^{T} P[s_t(k)|\boldsymbol{x}_t]\right)^{-1} \tag{5.19}$$

$$\boldsymbol{R}_k = \frac{\sum_{t=1}^{T} P[s_t(k)|\boldsymbol{x}_t]((\boldsymbol{y}_t - \boldsymbol{x}_t - \bar{\boldsymbol{r}}_k)(\boldsymbol{y}_t - \boldsymbol{x}_t - \bar{\boldsymbol{r}}_k)^T)}{\sum_{t=1}^{T} P[s_t(k)|\boldsymbol{x}_t]} - \Sigma_{\boldsymbol{x},k} \tag{5.20}$$

5.3. Summary

In this chapter we have presented a unified view of adaptation based data-driven environmental compensation algorithms. We have showed how both approaches of environment compensation share the same algorithmic structure and differ only in minor details. Solutions have been provided

for the case of stereo- and non stereo-based solutions. In the next chapters we particularize this discussion for the RATZ and STAR family of algorithms.

Chapter 6
The RATZ Family of Algorithms

In this chapter we particularize the generic solutions described in Chapter 3 for the Multiva**r**iate-Gaussian-Based Cepst**r**al Normali**z**ation (RATZ) family of algorithms. We present an overview of the algorithms and describe in detail the steps followed in RATZ-based compensation. We describe the generic stereo based and blind versions of the algorithms as well as the SNR dependent versions of RATZ and Blind RATZ. In addition we also describe the interpolated versions of the RATZ algorithms. Finally, we provide some experimental results using several databases and environmental conditions, followed by our conclusions.

6.1. Overview of RATZ and Blind RATZ

The algorithms work in the three following stages which are described as follows:

- Estimation of the statistics of clean speech
- Estimation of the statistics of noisy speech (stereo and non stereo cases)
- Compensation of noisy speech

Estimation of the statistics of clean speech. The pdf for the features of clean speech is modeled as a mixture of multivariate Gaussian distributions. Under these assumptions the distribution of the cepstral vectors of clean speech can be written as

$$p(x_t) = \sum_{k=1}^{K} a_k \, N_x(\mu_{x,k}, \Sigma_{x,k}) \tag{6.1}$$

which is equivalent to Equation (5.1) for the case of $a_k(t)$ being time independent.

The a_k, $\mu_{x,k}$ and $\Sigma_{x,k}$ represent respectively the *a priori* probabilities, mean vector and covariance matrix of each multivariate Gaussian mixture element *k*. These parameters are learned through traditional maximum likelihood EM methods [21]. The covariance matrix is assumed to be diagonal.

Estimation of the statistics of noisy speech. As we mentioned in Chapter 4 we will assume that the effect of the environment on speech statistics can be accurately modeled by applying the proper correction factors to the mean vectors and covariance matrices. Therefore, our goal will be

to compute these correction factors to estimate the statistics of noisy speech.

If we particularize the solutions of Chapter 5 we will have several solutions, depending on whether or not stereo data are available for learning the correction factors.

If stereo data are not available we obtain these solutions

$$= \frac{\sum_{t=1}^{T} P[k|\boldsymbol{y}_t, \phi]\boldsymbol{y}_t - \mu_{\boldsymbol{x},k}}{\sum_{t=1}^{T} P[k|\boldsymbol{y}_t, \phi]} - \mu_{\boldsymbol{x},k} \tag{6.2}$$

$$\overline{\boldsymbol{R}_k} = \frac{\sum_{t=1}^{T} P[k|\boldsymbol{y}_t, \phi]((\boldsymbol{y}_t - \mu_{\boldsymbol{x},k} - \overline{\boldsymbol{r}_k})(\boldsymbol{y}_t - \mu_{\boldsymbol{x},k} - \overline{\boldsymbol{r}_k})^T)}{\sum_{t=1}^{T} P[k|\boldsymbol{y}_t, \phi]} - \Sigma_{\boldsymbol{x},k} \tag{6.3}$$

where the $P[k|\boldsymbol{y}_t, \phi]$ term represents the *a posteriori* probability of an observation noisy vector $\boldsymbol{y}_t$ being produced by Gaussian k given the set of estimated correction parameters ϕ. The solutions are iterative and each iteration guarantees greater likelihood.

If stereo data are available we obtain these solutions

$$\bar{r}_k = \frac{\sum_{t=1}^{T} P[k|\boldsymbol{x}_t](\boldsymbol{y}_t - \boldsymbol{x}_k)}{\sum_{t=1}^{T} P[k|\boldsymbol{x}_t]} \tag{6.4}$$

$$\overline{\boldsymbol{R}}_k = \frac{\sum_{t=1}^{T} P[k|\boldsymbol{x}_t]((\boldsymbol{y}_t - \mu_{\boldsymbol{x},k} - \boldsymbol{r}_k)(\boldsymbol{y}_t - \mu_{\boldsymbol{x},k} - \boldsymbol{r}_k)^T)}{\sum_{t=1}^{T} P[k|\boldsymbol{x}_t]} - \Sigma_{\boldsymbol{x},k} \tag{6.5}$$

In this case the solution is non iterative.

Compensation of noisy speech. The solution for the correction factors $\{\boldsymbol{r}_1, ..., \boldsymbol{r}_K, \boldsymbol{R}_1, ..., \boldsymbol{R}_K\}$ helps us learn the new distributions of noisy speech cepstral vectors. With this knowledge we can

estimate what correction factor to apply to each incoming noisy vector $\boldsymbol{y}$ to obtain an estimated clean vector $\hat{\boldsymbol{x}}$. To do so we use a Minimum Mean Squared Error (MMSE) estimator

$$\hat{\boldsymbol{x}}_{MMSE} = E(\boldsymbol{x}|\boldsymbol{y}) = \int_{\boldsymbol{X}} \boldsymbol{x}.p(\boldsymbol{x}|\boldsymbol{y})d\boldsymbol{x} \tag{6.6}$$

Since this equation requires the knowledge of the marginal distribution $p(\boldsymbol{x}|\boldsymbol{y})$ and this might be difficult or impossible to get in closed form (see Chapter 4), some simplifications are needed. In particular we will first assume that the $\boldsymbol{x}$ vector can be presented as $\boldsymbol{x} = \boldsymbol{y} - \boldsymbol{r}(\boldsymbol{x})$. In this case Equation (6.2) simplifies to

$$\begin{aligned}
\hat{\boldsymbol{x}}_{MMSE} &= \boldsymbol{y} - \int_{\boldsymbol{X}} \boldsymbol{r}(\boldsymbol{x}) p(\boldsymbol{x}|y)d\boldsymbol{x} = \boldsymbol{y} - \int_{\boldsymbol{X}} \sum_{k=1}^{K} \boldsymbol{r}(\boldsymbol{x}) p(\boldsymbol{x}, k|\boldsymbol{y})d\boldsymbol{x} \\
&= \boldsymbol{y} - \sum_{k=1}^{K} P[k|\boldsymbol{y}] \int_{\boldsymbol{X}} \boldsymbol{r}(\boldsymbol{x}) p(\boldsymbol{x}|k, \boldsymbol{y})d\boldsymbol{x} \\
&\cong \boldsymbol{y} - \sum_{k=1}^{K} \boldsymbol{r}_k \, P[k|\boldsymbol{y}] \int_{\boldsymbol{X}} p(\boldsymbol{x}|k, \boldsymbol{y})d\boldsymbol{x} \\
&\cong \boldsymbol{y} - \sum_{k=1}^{K} \boldsymbol{r}_k \, P[k|\boldsymbol{y}]
\end{aligned} \tag{6.7}$$

where we have further simplified the $\boldsymbol{r}(\boldsymbol{x})$ expression to $\boldsymbol{r}_k$. This is equivalent to assuming that the $\boldsymbol{r}(\boldsymbol{x})$ term can be well approximated by a constant value within the region in which $p(\boldsymbol{x}|k, \boldsymbol{y})$ has a significant value.

6.2. Overview of SNR-Dependent RATZ and Blind-RATZ

RATZ and Blind RATZ as described in [39] and [40] used a standard Gaussian mixture distribution as defined in Equation (5.1) where the $a_k(t)$ terms are taken to be independent of time to model the statistics of cepstra of clean speech. While this model is valid, it is constrained in that it resolves all the cepstral components equally, *i.e.*, into the same number of Gaussians. In particular, the frame energy parameter, $_0$, has the same resolution in terms of number of Gaussians as the other cepstral parameters.

Previous work by Acero [1] and Liu [35] suggests that a more fine modelling of $_0$ is necessary. In Acero's FCDCN algorithm [1] the covariance matrix used to model the distribution of the cep-

strum is diagonal. This gives higher importance to the $_0$ coefficient and makes a more detailed modeling of the $_0$ coefficient very helpful. Inspired by Liu and Acero's previous work, SNR-RATZ and SNR-BRATZ use a more structured model for the distribution whereby the number of Gaussians used to define the $_0$ statistics can be different from the number used for the other cepstral components.

Figure 6-1 illustrates this idea for a two-dimensional vector $\boldsymbol{x} = [x_0\ x_1]^T$. In this example the pdf has the following structure

$$p(\boldsymbol{x}) = \sum_{i=0}^{1} P[i] N_{x_0|i}(\mu_{x_0,i}, \sigma^2_{x_0,i}) \sum_{j=0}^{2} P[j|i] N_{x_1|i,j}(\mu_{x_1,i,j}, \sigma^2_{x_1,i,j}) \tag{6.8}$$

In this example there are two distributions for the $_0$ component and each $_0$ has three associated marginal distributions in the x_1 variable. Note that the means of the mixtures that comprise the pdf of $_1$ associated with each mixture component of $_0$ can take on any value, and they generally differ for different values of $_0$.

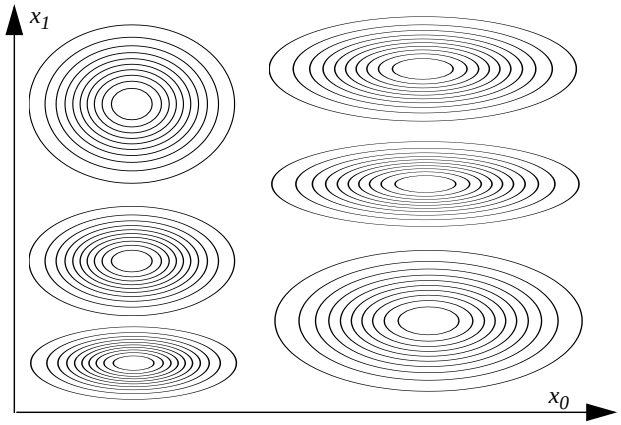

Figure 6-1: Contour plot illustrating joint pdfs of the structural mixture densities for the components $_0$ and $_1$

The SNR dependent versions of RATZ also work in the three basic steps mentioned in Section 4.1, namely

- Estimation of the statistics of clean speech
- Estimation of the statistics of noisy speech (stereo and non stereo cases)
- Compensation of noisy speech

Estimation of the statistics of clean speech. In our implementation of the SNR-RATZ and blind SNR-RATZ algorithms we split the cepstral vector in two parts, $\boldsymbol{x} = [x_0\ \boldsymbol{x}^T_1]^T$, where $\boldsymbol{x}_1$ is itself a vector composed of the $x_1, x_2 ..., x_{L-1}$ components of the original cepstral vector.

The resulting distribution for the clean cepstrum vectors has the following structure

$$p(\boldsymbol{x}) = \sum_{i=0}^{M} a_i\, N_{x_0|i}(\mu_{x_0,i}, \sigma^2_{x_0,i}) \sum_{j=0}^{N} a_{i,j}\, N_{\boldsymbol{x}_1|i,j}(\mu_{\boldsymbol{x}_1,i,j}, \Sigma_{\boldsymbol{x}_1,i,j}) \tag{6.9}$$

The means, variances, and *a priori* probabilities of the individual Gaussians are learned by standard EM methods [13]. Appendix C summarizes the resulting solutions.

Estimation of the statistics of noisy speech. As in the conventional RATZ algorithm we assume that the effect of the environment on the means and variances of the cepstral distributions of clean speech can be adequately modelled by additive correction factors.

The resulting means and variances of the statistics of noisy speech are

$$\begin{aligned} \mu_{y_0,i} &= r_i + \mu_{x_0,i} & \sigma^2_{y_0,i} &= R_i + \sigma^2_{x_0,i} \\ \mu_{\boldsymbol{y}_1,i,j} &= \boldsymbol{r}_{i,j} + \mu_{\boldsymbol{x}_1,i,j} & \Sigma_{\boldsymbol{y}_1,i,j} &= \boldsymbol{R}_{i,j} + \Sigma_{\boldsymbol{x}_1,i,j} \end{aligned} \tag{6.10}$$

where r_i, R_i, $\boldsymbol{r}_{i,j}$ and $\boldsymbol{R}_{i,j}$ represent the correction factors applied to the clean speech cepstrum distribution.

To obtain these correction factors we can use techniques very similar to those used in the case of non-SNR RATZ. Notice that the only difference lies in the structure of $p(\boldsymbol{x})$. Appendix B gives a detailed explanation of the procedure used to obtain the optimal correction factors for the stereo-based and non-stereo-based cases.

Compensation of noisy speech. Once the r_i, R_i, $\boldsymbol{r}_{i,j}$ and $\boldsymbol{R}_{i,j}$ correction factors are computed we can apply a MMSE procedure that is similar to the one used in the non-SNR-based RATZ case. The MMSE estimator will have the form

$$\hat{\boldsymbol{x}}_{MMSE} = E(\boldsymbol{x}|\boldsymbol{y}) = \int_{\boldsymbol{X}} \boldsymbol{x}.p(\boldsymbol{x}|\boldsymbol{y})d\boldsymbol{x} \tag{6.11}$$

As in the non-SNR-dependent case we will first assume that the vector $\boldsymbol{x}$ can be represented as $\boldsymbol{x} = \boldsymbol{y} - \boldsymbol{s}(\boldsymbol{x})$. In this case Equation (6.2) simplifies to

$$\begin{aligned}\hat{\boldsymbol{x}}_{MMSE} &= \boldsymbol{y} - \int_{\boldsymbol{X}} \boldsymbol{s}(\boldsymbol{x}) p(\boldsymbol{x}|\boldsymbol{y}) d\boldsymbol{x} = \boldsymbol{y} - \int_{\boldsymbol{X}} \sum_{i=0}^{M} \sum_{j=0}^{N} \boldsymbol{s}(\boldsymbol{x}) p(\boldsymbol{x}, i, k|\boldsymbol{y}) d\boldsymbol{x} \\ &= \boldsymbol{y} - \int_{\boldsymbol{X}} \sum_{i=0}^{M} \sum_{j=0}^{N} \boldsymbol{s}(\boldsymbol{x}) p(\boldsymbol{x}|\boldsymbol{y}, i, j) P[i, j|\boldsymbol{y}] d\boldsymbol{x} \\ &\cong \boldsymbol{y} - \int_{\boldsymbol{X}} \sum_{i=0}^{M} \sum_{j=0}^{N} \boldsymbol{s}_{i,j}\ p(\boldsymbol{x}|\boldsymbol{y}, i, j) P[i, j|\boldsymbol{y}] d\boldsymbol{x} \\ &\cong \boldsymbol{y} - \sum_{i=0}^{M} \sum_{j=0}^{N} \boldsymbol{s}_{i,j} P[i, j|\boldsymbol{y}]\end{aligned} \tag{6.12}$$

where we have further simplified the expression for $\boldsymbol{s}(\boldsymbol{x})$ to $\boldsymbol{s}_{i,j}$, a vector composed of the concatenation of the correction terms r_i and $\boldsymbol{r}_{i,j}$. This is equivalent to assuming that the $\boldsymbol{s}(\boldsymbol{x})$ term can be well approximated by a constant value within the region in which $p(\boldsymbol{x}|i, j, \boldsymbol{y})$ has its maximum, *i.e.*, the mean.

6.3. Overview of Interpolated RATZ and Blind RATZ

The previous versions of the RATZ algorithms assumed that the environment where the recognition is going to be performed is known, enabling the RATZ algorithm to make use of previously-learned correction factors. However, in more realistic conditions this might not be possible. Even though there might be enough adaptation data to learn the correction factors of a number of environments we might not know what environment is presented to us for recognition.

The basic idea of Interpolated RATZ is to estimate the *a posteriori* probabilities of each of E possible environments over the whole ensemble of cepstrum vectors for the utterance $\boldsymbol{Y}$

$$P[\textit{environment}{=}i|\boldsymbol{Y}] = \frac{P[i] \prod_{t=1}^{T} p(\boldsymbol{y}_t|i)}{\sum_{e=1}^{E} P[e] \prod_{t=1}^{T} p(\boldsymbol{y}_t|e)} \tag{6.13}$$

The *a priori* probability of each environment is $P[i]$. We normally assume that all the environments are equiprobable.

The $p(\boldsymbol{y}_t|i)$ terms are defined as

$$p(\boldsymbol{y}_t|i) = \sum_{k=1}^{K} a_k \, N_{\boldsymbol{y}}(\mu_{\boldsymbol{x},k} + \boldsymbol{r}_{\boldsymbol{x},k,i}, \Sigma_{\boldsymbol{x},k} + \boldsymbol{R}_{\boldsymbol{x},k,i}) \tag{6.14}$$

where $\boldsymbol{r}_{x,k,i}$ and $\boldsymbol{R}_{x,k,i}$ represent the correction terms for the environment i.

Once the *a posteriori* probabilities for each of the putative environments are computed we can use them to weight each of the environment dependent correction factors

$$\begin{aligned}
\hat{\boldsymbol{x}}_{MMSE} &= \boldsymbol{y} - \int_{\boldsymbol{X}} \sum_{e=1}^{E} \sum_{k=1}^{K} \boldsymbol{r}(\boldsymbol{x}).p(\boldsymbol{x},k,e|\boldsymbol{y})d\boldsymbol{x} \\
\hat{\boldsymbol{x}}_{MMSE} &= \boldsymbol{y} - \sum_{e=1}^{E} \sum_{k=1}^{K} \int_{\boldsymbol{X}} \boldsymbol{r}(\boldsymbol{x}).p(\boldsymbol{x}|k,e,\boldsymbol{y})P[k|e,\boldsymbol{y}]P[e|\boldsymbol{y}]d\boldsymbol{x} \\
\hat{\boldsymbol{x}}_{MMSE} &\cong \boldsymbol{y} - \sum_{e=1}^{E} P[e|\boldsymbol{Y}] \sum_{k=1}^{K} \boldsymbol{r}_{k,e} \, P[k|e,\boldsymbol{y}] \int_{\boldsymbol{X}} p(\boldsymbol{x}|k,e,\boldsymbol{y})d\boldsymbol{x} \\
\hat{\boldsymbol{x}}_{MMSE} &\cong \boldsymbol{y} - \sum_{e=1}^{E} P[e|\boldsymbol{Y}] \sum_{k=1}^{K} \boldsymbol{r}_{k,e} \, P[k|e,\boldsymbol{y}]
\end{aligned} \tag{6.15}$$

where we have approximated $P[e|\boldsymbol{y}]$ by $P[e|\boldsymbol{Y}]$, using all the cepstrum vectors in the utterance to compute the *a posteriori* probability of the environment.

Similar extensions are also possible for the case of the SNR-dependent RATZ algorithms.

6.4. Experimental Results

In this section we describe several experiments designed to evaluate the performance of the RATZ family of algorithms. We explore several of the dimensions of the algorithms, such as:

- the impact of SNR dependence on recognition accuracy
- the impact of the number of adaptation sentences on recognition accuracy
- the optimal number of Gaussians
- the impact of interpolation on the performance of the algorithm

The experiments described here are performed on the 5,000-word Wall Street Journal 1993 evaluation set with white Gaussian noise added at several SNR levels. The SNR is computed on a

sentence by sentence basis. For each sentence the energy is computed and white artificially generated Gaussian noise is added at the decided SNR below the signal energy level. In all the experiments with this database, the upper dotted line represents the performance of the system when fully trained on noisy data while the lower dotted line represents the performance of the system when no compensation is used.

6.4.1. Effect of an SNR-dependent structure

In this section we compare the possible benefits of an SNR-dependent structure as described in Section 4.2. To explore the effect of an SNR dependent structure in our experiments we compare the performance of the RATZ and SNR-RATZ algorithms using the same number of Gaussians.

In the case of SNR-RATZ we use two configurations. The first configuration which we call an 8.32 SNR-RATZ configuration contains 8 x_0 Gaussians and 32 x_0-dependent Gaussians while the second one called a 4.16 SNR-RATZ configuration contains 4 x_0-Gaussians and 16 x_0 dependent Gaussians. In the case of regular RATZ we also use two configurations. The first configuration contains 256 Gaussians and the second one 64 Gaussians. The correction factors were computed from the same stereo databases. One hundred adaptation sentences were used to learn the correction factors.

Figure 6-2 shows the performance for this particular database for all the four mentioned conditions.

As we can see, contrary to previous results by Liu and Acero, a SNR-based structure does not seem to provide clear benefits in reducing the error rate. In all four configurations the results are comparable. Perhaps this is due to the fact that Acero and Liu in their FCDCN [1, 35] algorithms use a weaker model to represent the distributions of clean speech cepstrum based on vector quantization (VQ). In particular their model does not take into account the compression of the distributions. Another possible reason is that FCDCN uses a diagonal covariance matrix in which all the elements have the same value. This weights too much the x_0 component of the cepstral vector in computing likelihoods. Under this conditions, a SNR-dependent structure might have significant benefits.

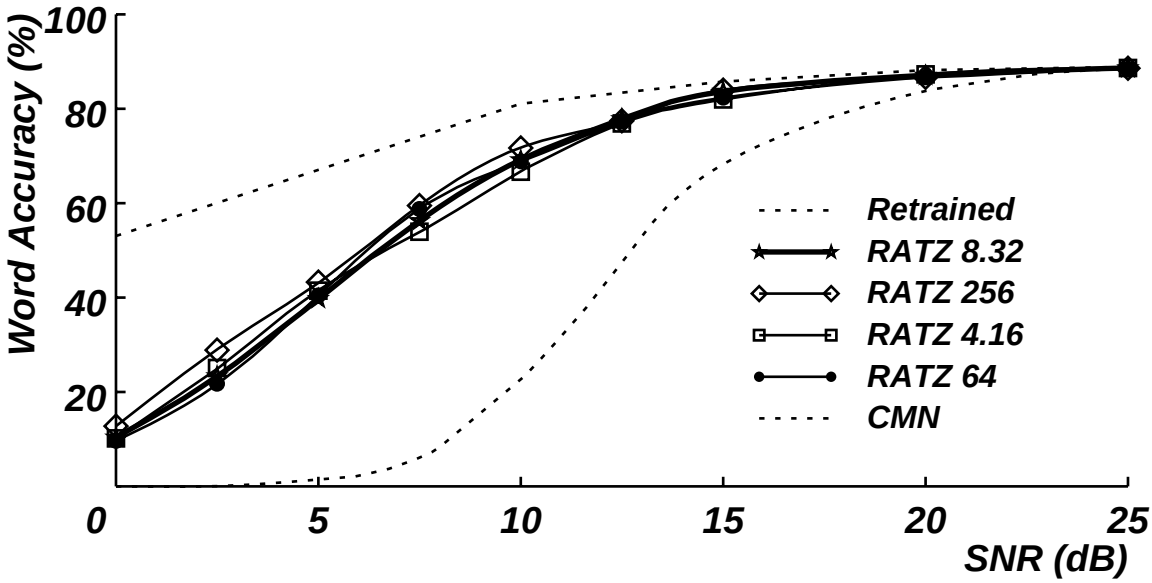

Figure 6-2. Comparison of RATZ algorithms with and without a SNR dependent structure. We compare an 8.32 SNR-RATZ algorithm with a normal RATZ algorithm with 256 Gaussians. We also compare a 4.16 SNR-RATZ algorithm with a normal RATZ algorithm with only 64 Gaussians.

6.4.2. Effect of the number of adaptation sentences

To explore the effect the number of adaptation sentences has on recognition accuracy we learned correction factors using a varied number of adaptation sentences. We present results for the 8.32 configuration of SNR-dependent RATZ.

Figure 6-3 demonstrates that even with a very small number of adaptation sentences the RATZ algorithm is able to compensate for the effect of the environment. In fact the performance seems to be quite insensitive to the number of adaptation sentences. Only when the number of sentences is lower than 10 do we observe a decrease in accuracy.

6.4.3. Effect of the number of Gaussian Mixtures

To explore the effect the number of Gaussians has on recognition accuracy we compared several RATZ algorithms. In particular we used 256, 64, and 16 stereo RATZ configurations with correction factor learned from 100 adaptation sentences for our study. Figure 4-4 shows that as the number of Gaussians increases, the recognition performance increases. The differences are more significant at lower SNR levels.

6.4.4. Stereo based RATZ vs. Blind based RATZ

In this section we consider not having stereo data to learn the correction factors on recognition

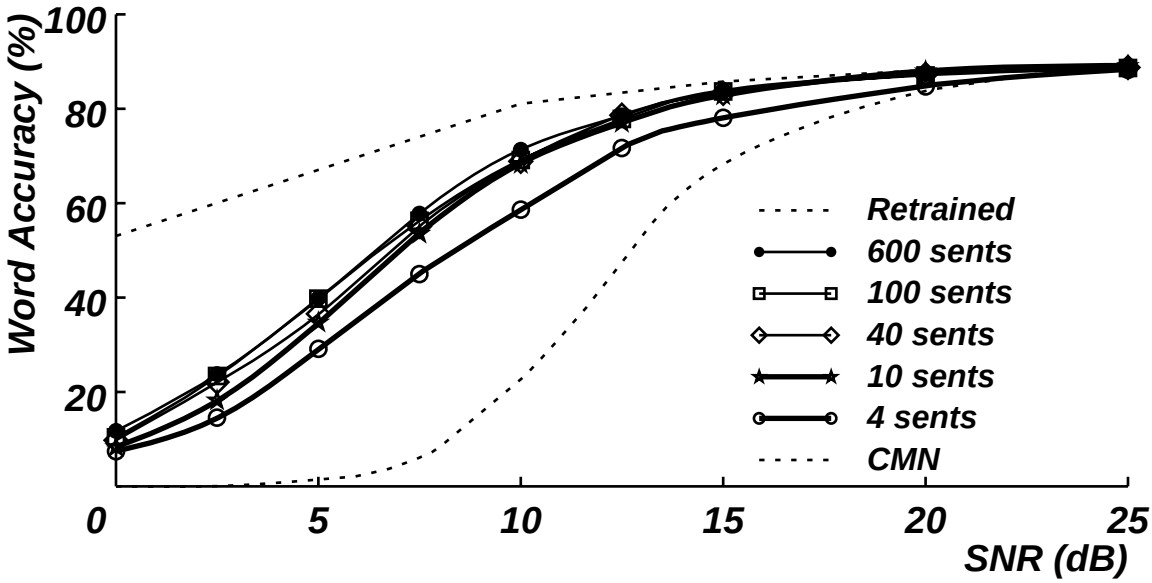

Figure 6-3. Study of the effect of the number of adaptation sentences on a 8.32 SNR dependent RATZ algorithm. We observe that even with only 10 sentences available for adaptation the performance of the algorithm does not seem to suffer.

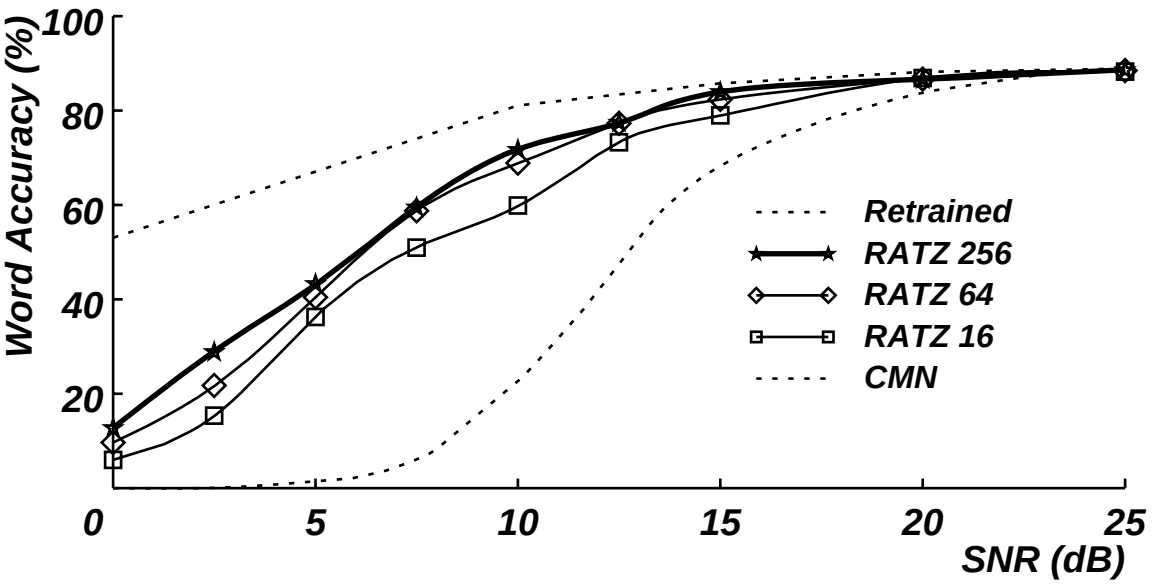

Figure 6-4. Study of the effect of the number of Gaussians on the performance of the RATZ algorithms. In general a 256 configuration seems to perform better than a 64 or 16.

performance. We compare two identical 4.16 SNR-dependent RATZ algorithms with the only difference being the presence or absence of simultaneously-recorded (“stereo”) clean and noisy sentences in learning the correction factors. The correction factors were trained using one hundred sentences in each case. Figure 6-5 shows that not having stereo data is detrimental to recognition

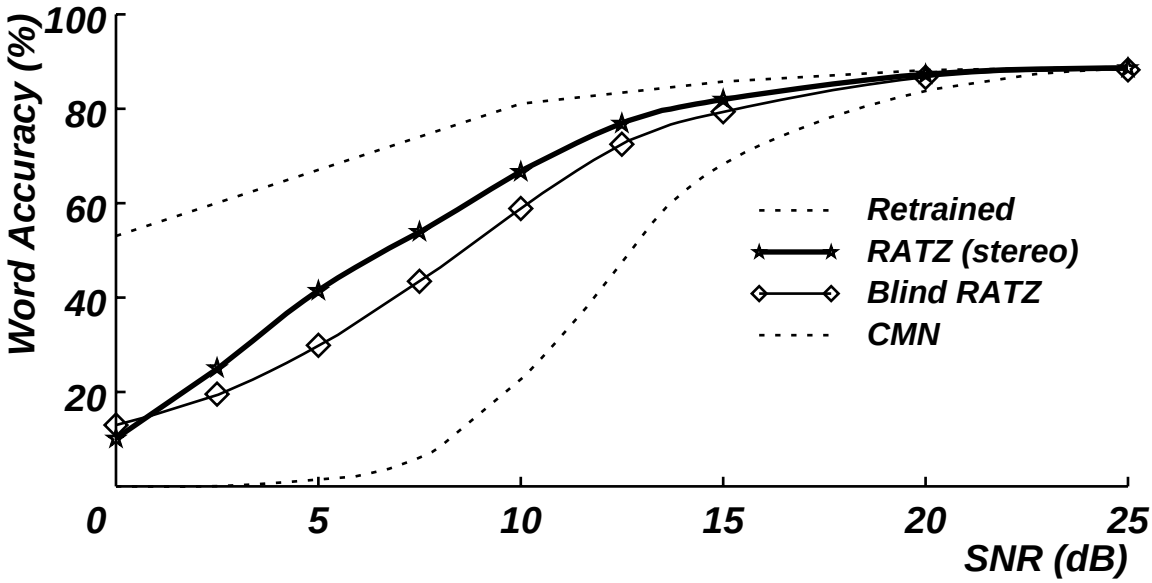

Figure 6-5. Comparison of a stereo based 4.16 RATZ algorithm with a blind 4.16 RATZ algorithm. The stereo-based algorithm outperforms the blind algorithm at almost all SNRs.

accuracy. However when compared to not doing any compensation at all (the CMN case) Blind RATZ still provides considerable benefits.

6.4.5. Effect of environment interpolation on RATZ

In this section we compare the performance of interpolated versions of RATZ with that of equivalent not interpolated versions. We compare two 8.32 SNR-dependent RATZ algorithms where the number of adaptation sentences used to learn the corrections was set to 100. In the standard version the correction factors are chosen from the correct environment. In the interpolated version the correction factors are chosen from a list of possible environments containing correction factors learned from data contaminated at different SNRs.

Figure 6-6 shows that not knowing the environment has almost no significant effect on the performance of the algorithm. In fact it seems to improve accuracy slightly. If the correct environment is removed from the list of environments at each SNR the algorithms does not seem to suffer either.

6.4.6. Comparisons with FCDCN

The FCDCN family of algorithms introduced by Acero [1] and further studied by Liu [35] is compared in this section with the RATZ family of algorithms. FCDCN can be considered to be a particular case of the RATZ algorithms where a VQ codebook is used to represent the statistics of

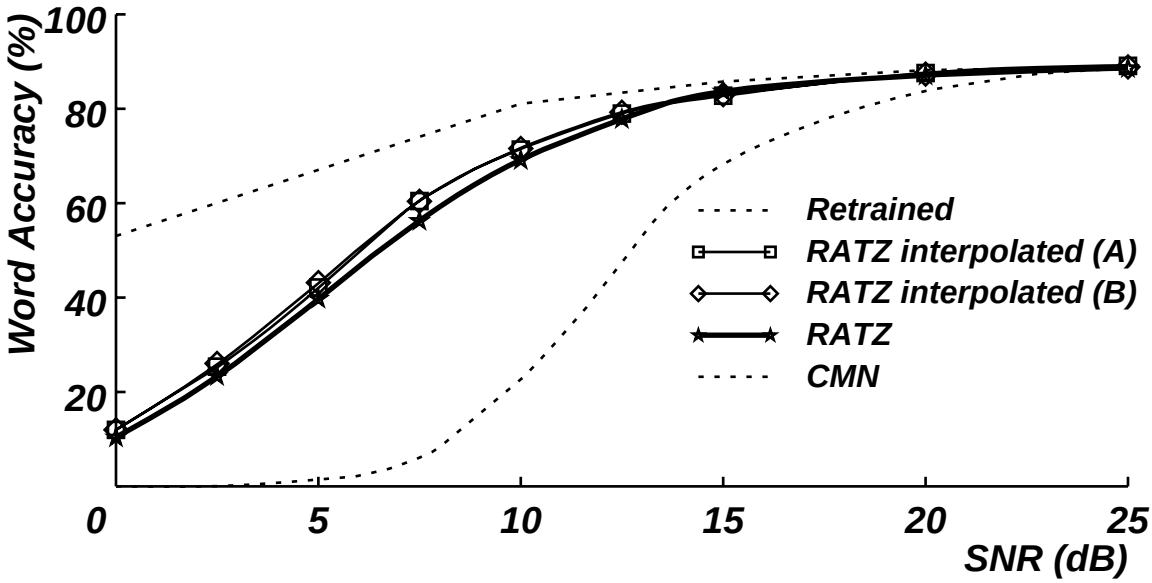

Figure 6-6. Effect of environment interpolation in recognition accuracy. The curve labeled RATZ interpolated (A) was computed excluding the correct environment from the list of environments. The curve labeled RATZ interpolated (B) was computing with all environments available for interpolation.

clean speech and the effect of the environment on this codebook is modelled by shifts in the centroids of the codebook. In FCDCN all the centroids in the codebook have the same variance. When compensation is applied a single correction factor is applied; the one that minimizes the VQ distortion.

Figure 6-7 compares a RATZ configuration with 256 Gaussians with an equivalent FCDCN configuration. The same sentences were used to learn the statistics or VQ codebook of clean speech and the same 100 stereo sentences were used to learn the correction factors. As we can see the RATZ algorithm outperforms FCDCN at all SNRs. This can be explained by the better representation used by RATZ to represent clean speech distributions and by the better model used by RATZ to represent the effect of the environment on clean speech distributions. The difference in accuracy is more obvious at lower SNRs.

The same experiment was reproduced with a SNR-dependent structure comparing FCDCN with RATZ and the same result was observed.

6.5. Summary

In this section we have presented the RATZ family of algorithms as a particular case of the unified approach described in the previous chapter. We have described the RATZ, SNR-dependent

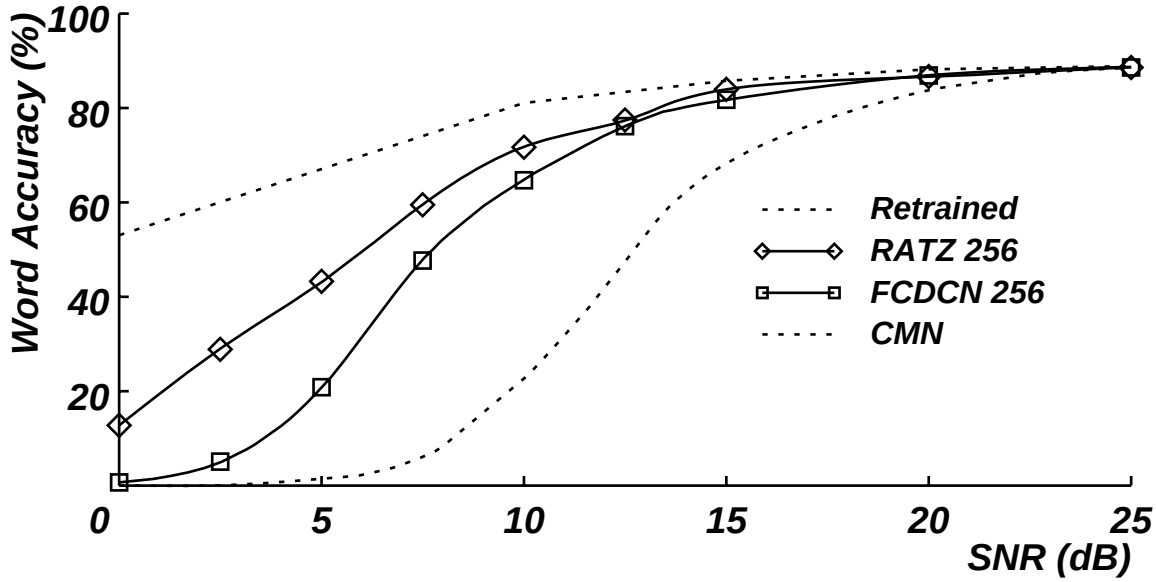

Figure 6-7. Effect of environment interpolation on the performance of the RATZ algorithm. In this case we compare the effect of removing the right environmental correction factors from the list of environments. We can observe that removing the right environment does not affect the performance of the algorithm.

RATZ, Blind RATZ, and Interpolated RATZ algorithms.

We have explored some of the dimensions of the algorithm including the number of distributions used to describe the clean speech cepstrum acoustics, the impact of a SNR dependent distribution, the effect of not having stereo data available to learn the environment correction factor, and the effect of interpolation of the correction factors on recognition performance. Finally we have compared the RATZ algorithm with the FCDCN algorithm developed by Acero [1].

From the experimental results presented in this chapter we conclude that contrary to previous results by Liu and Acero [1, 35], an SNR-dependent structure does not provide any improvement in recognition accuracy. An explanation for this difference in behavior is as follows. The use of diagonal covariance matrices in FCDCN in which all the elements are equal gives a disproportionate weight to the x_0 coefficient in its contribution to the likelihood. Therefore, a partition of the data according to the x_0 coefficient will reduce its variability and this in turn will reduce its contribution to the likelihood. In the RATZ family of algorithms we use diagonal covariance matrices with all the elements learned. This weights all the components of the cepstral vector equally making a explicit division of the data according to x_0 unnecessary.

Also, contrary to previous results with FCDCN [36], the RATZ family of algorithms seem to

be quite insensitive to the number of sentences used to learn the environmental correction factors. In fact, the algorithms seem to work quite well even with only 10 sentences (or about 70 seconds of speech). Perhaps the use of a Maximum Likelihood formulation, where each data sample contributes to learn all the parameters, is responsible for this. We have also shown that the use of a Maximum Likelihood formulation allows for a natural extension of the RATZ algorithms for the case in which only noisy data is available to learn the correction factors. Unlike FCDCN, which is based on VQ, the extension of RATZ to work without simultaneously-recorded clean and noisy data is quite natural. This Maximum Likelihood structure also allowed us to extend naturally the RATZ algorithms for the case of environment interpolation. The experiments with interpolated RATZ provided almost no loss in recognition accuracy when compared with the case of known environment.

In general we can observe how the RATZ algorithms are able to achieve the recognition performance of a fully retrained system up to a SNR of about 15 dB. For lower signal-to-noise ratios the algorithms provides partial recovery for the degradation introduced by the environment.

Chapter 7
The STAR Family of Algorithms

In this chapter we particularize the generic solutions described in Chapter 3 for the STAtistical Reestimation (STAR) algorithms. Unlike the previous RATZ family of algorithms that apply correction factors to the incoming cepstral vectors of noisy speech, STAR tries to modify some of the parameters of the acoustical distributions in the HMM structure. Therefore there is no need for "compensation" since the noisy speech cepstrum vectors are used for recognition. Because of this STAR is called a "distribution compensation" algorithm.

As we discussed before, there are theoretical reasons that support the idea that algorithms that attempt to adapt the distributions of the acoustics to those of the noisy target speech are optimal. The experimental results presented in this chapter support this conclusion.

We present an overview of the STAR algorithm and describe in detail all the steps followed in STAR based compensation. We also provide some experimental results on several databases and environmental conditions. Finally, we present our conclusions.

7.1. Overview of STAR and Blind STAR

The idea of data-driven algorithms that adapt the HMMs to the environment has been introduced before. For example, the Tied-mixture normalization algorithm proposed by Anastasakos [4] and the Dual Channel Codebook adaptation algorithm proposed by Liu [35] are similar in spirit to STAR. However, they are based on VQ indices rather than Gaussian *a posteriori* probabilities and use a weaker model of the effect of the environment on Gaussian distributions in which the covariance matrices are not corrected. Furthermore, they only model the effect of the environment on cepstrum distributions without modelling the effect on the other feature streams (see Section 2.1.1.) such as delta cepstrum, double delta cepstrum and energy.

Since the clean speech cepstrum is represented by the HMM distribution, in the STAR family of algorithms there is no need to explore a SNR-dependent structure as this would imply changing the underlying HMM structure to have SNR-dependencies. Furthermore, since in our experiments with the RATZ algorithms we did not find evidence to support the use of a SNR-dependent structure we decided not to explore this issue.

The algorithm works in the two following stages which are described as follows:

- Estimation of the statistics of clean speech
- Estimation of the statistics of noisy speech

In the next paragraphs we explain in detail each of the two stages of the STAR algorithm.

Estimation of the statistics of clean speech. The STAR algorithm uses the acoustical distributions modeled by the HMMs to represent clean speech. Therefore, strictly speaking this is not a step related to the STAR algorithm. The algorithm just takes advantage of the information contained on the HMMs.

In the SPHINX-II system the distributions representing the cepstra of clean speech are modeled as a mixture of multivariate Gaussian distributions. Under these assumptions the distribution for clean speech can be written as

$$p(\boldsymbol{x}_t) = \sum_{k=1}^{K} a_k(t)\ N_{\boldsymbol{x}}(\mu_{\boldsymbol{x},k}, \Sigma_{\boldsymbol{x},k}) \tag{7.1}$$

where the $a_k(t)$ term represents the *a priori* probability of each of the Gaussians of each of the possible states with the restriction that the total number of Gaussians is limited to 256 and shared across all states. Reference [23] describes the SPHINX-II HMM topology and acoustical modelling assumptions in detail.

The $\mu_{\boldsymbol{x},k}$ and $\Sigma_{\boldsymbol{x},k}$ terms represent the mean vector and covariance matrix of each multivariate Gaussian mixture element *k*. These parameters are learned through the well known Baum-Welch algorithm **[**25].

Estimation of the statistics of noisy speech. As we mentioned in Chapter 4, we assume that the effect of the environment on the distributions of speech cepstra can be well modeled by applying the proper correction factors to the mean vectors and covariance matrices. Therefore, our goal will be to compute these correction factors to estimate the statistics of noisy speech.

Using the methods described in Chapter 4 we have several solutions, depending on whether or not stereo data are available to learn the correction factors.

If stereo data are not available we obtain these solutions

$$\bar{\boldsymbol{r}}_k = \frac{\sum_{t=1}^{T} P[s_t(k)|\boldsymbol{y}_t, \phi]\boldsymbol{y}_t}{\sum_{t=1}^{T} P[s_t(k)|\boldsymbol{y}_t, \phi]} - \mu_{\boldsymbol{x},k} \tag{7.2}$$

$$\bar{\boldsymbol{R}}_k = \frac{\sum_{t-1}^{T} P[s_t(k)|\boldsymbol{y}_t, \phi]((\boldsymbol{y}_t - \mu_{\boldsymbol{x},k} - \bar{\boldsymbol{r}}_k)(\boldsymbol{y}_t - \mu_{\boldsymbol{x},k} - \bar{\boldsymbol{r}}_k)^T)}{\sum_{t=1}^{T} P[s_t(k)|\boldsymbol{y}_t, \phi]} - \Sigma_{\boldsymbol{x},k} \tag{7.3}$$

where the $P[s_t(k)|\boldsymbol{y}_t, \phi]$ term represent the *a posteriori* probability of an observation noisy vector $\boldsymbol{y}_t$ being produced by Gaussian k in state $s_t(k)$ given the set of estimated correction parameters ϕ. The solutions are iterative and each iteration guarantees higher likelihood. Notice that in this case the solutions are very similar to the Baum-Welch reestimation solutions commonly used for HMM training [21].

If stereo data are available we obtain these solutions

$$\bar{\boldsymbol{r}}_k = \frac{\sum_{t=1}^{T} P[s_t(k)|\boldsymbol{x}_t](\boldsymbol{y}_t - \boldsymbol{x}_t)}{\sum_{t=1}^{T} P[s_t(k)|\boldsymbol{x}_t]} \tag{7.4}$$

$$\bar{\boldsymbol{R}}_k = \frac{\sum_{t=1}^{T} P[s_t(k)|\boldsymbol{x}_t]((\boldsymbol{y}_t - \mu_{\boldsymbol{x},k} - \bar{\boldsymbol{r}}_k)(\boldsymbol{y}_t - \mu_{\boldsymbol{x},k} - \bar{\boldsymbol{r}}_k)^T)}{\sum_{t=1}^{T} P[s_t(k)|\boldsymbol{x}_t]} - \Sigma_{\boldsymbol{x},k} \tag{7.5}$$

where the solutions are non iterative. Notice that in the stereo case the substitution of $P[s_t(k)|\boldsymbol{y}_t]$ by $P[s_t(k)|\boldsymbol{x}_t]$ assumes implicitly that the *a posteriori* probabilities do not change due to the environment.

As we explained in Chapter 2, in the SPHINX-II system we use not only the cepstral vector as to represent the speech signal but also a first-difference cepstral vector, a second-difference cepstral vector and a fourth vector composed of three components: the energy, first-difference energy, and second-difference energy. Each of these four streams of vectors is modelled with a different set of 256 Gaussians whose probabilities are combined.

The STAR family of algorithms assumes that each of these four streams will be affected by the environment in a similar way as the cepstral stream. Even though we have not presented evidence to support this assumption our experimental results as well as results by Gales [15] support this assumption.

The STAR algorithm models the effect of the environment on the mean vectors and covariance matrices of the distributions by additional correction factors. We estimate each of the additional correction factors using formulas that are equivalent to those used to estimate the correction factors for the cepstral stream.

Once the correction factors are estimated we can perform recognition using the distributions of noisy speech estimated as distributions of clean speech corrected by the appropriate factors.

7.2. Experimental Results

In this section we describe several experiments designed to evaluate the performance of the STAR family of algorithms. We explore several of the dimensions of the algorithm, such as:

- the impact of the number of adaptation sentences on speech recognition performance
- the effect of having stereo data to learn the correction factors
- comparison of the STAR algorithm to other algorithms such as RATZ.

The experiments described here are performed on the 5,000-word Wall Street Journal 1993 evaluation set with white Gaussian noise added at several SNR levels. In all the following figures the upper dotted line represents the performance of the system when fully trained on noisy data while the lower dotted line represents the performance of the system when no compensation is used.

7.2.1. Effect of the number of adaptation sentences

In this section we study the sensitivity of the algorithm to the number of adaptation sentences used to learn the correction factors. The correction factors were learned from five sets of 10, 40,

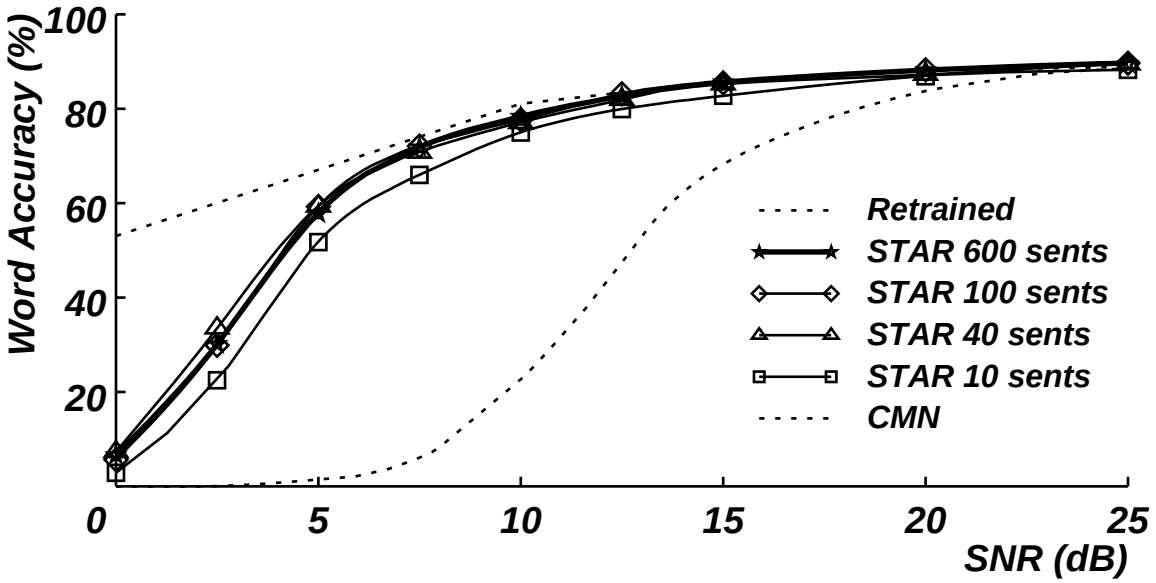

Figure 7-1. Effect of the number of adaptation sentences used to learn the correction factors r_k and R_k on the recognition accuracy of the STAR algorithm. The bottom dotted line represents the performance of the system with no compensation.

100, 200, and 600 stereo adaptation sentences at different signal-to-noise ratios.

Figure 7-1 shows the error rates for this particular database for all the aforementioned conditions. As we can see as the number of adaptation sentences grows the accuracy of the algorithm improves. However, with only 40 sentences the algorithm seems to capture all the needed information with no further improvements with more sentences. It is interesting to note that at SNRs larger than 12.5 dB the performance seems to be quite independent of the number of adaptation sentences even with only 10 adaptation sentences.

7.2.2. Stereo vs. non-stereo adaptation databases

In this section we explore the effect of not having stereo databases available for learning the correction factors. We also explore different alternatives to bootstrap the iterative learning equations when no stereo data are available.

Figure 7-2 shows the recognition accuracy for STAR and Blind STAR where the number of adaptation sentences used was 100. Ten iterations of the reestimation formulas were used for the Blind STAR experiments. To explore the effect the initial distributions have on the Blind STAR algorithm, we initialized the algorithm both from the distributions for clean speech and also from the closest distributions. We observe that when using the closest SNR distributions to bootstrap the

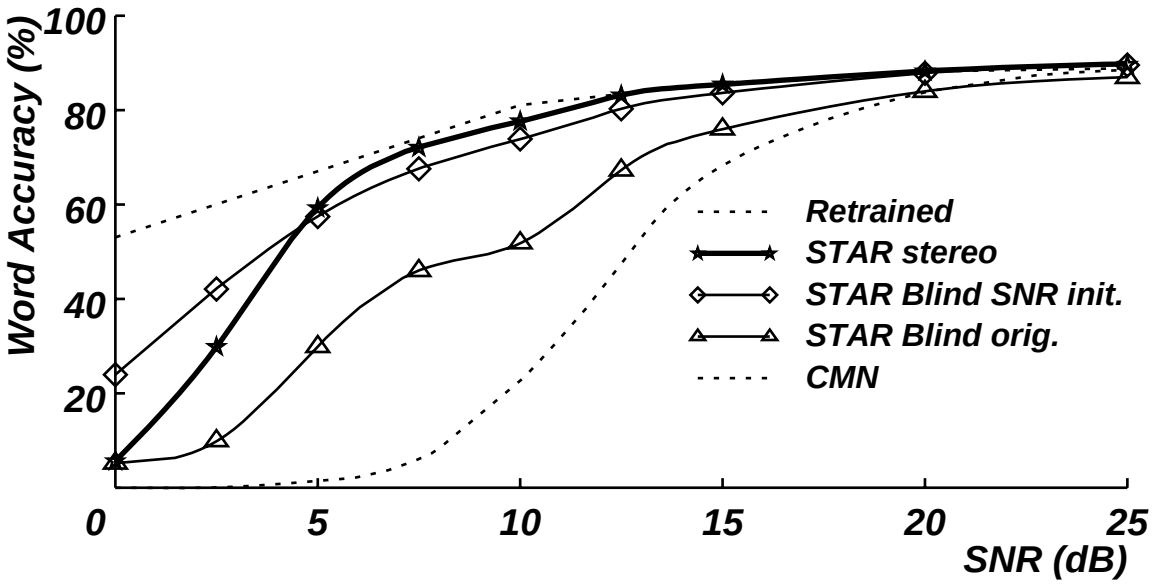

Figure 7-2. Comparison of the Blind STAR, and original STAR algorithms. The line with diamond symbols represents the original blind STAR algorithm while the line with triangle symbols represents the blind STAR algorithm bootstrapped from the closest in SNR sense distributions.

STAR algorithm its performance improves considerably. However, the stereo-based STAR seems to outperform the blind STAR versions at SNRs of 5 dB and above. For lower SNRs we hypothesize that a properly-initialized blind STAR algorithm might have some advantages over the stereo-based STAR. In particular, the assumption that the *a posteriori* probabilities $P[s_t(k)|\boldsymbol{y}_t]$ can be replaced by $P[s_t(k)|\boldsymbol{x}_t]$ is not completely valid at lower SNRs.

7.2.3. Comparisons with other algorithms

In this section we compare the performance of STAR with other previously developed algorithms. We make comparisons where the number of adaptation sentences is equivalent and where the availability of stereo data to learn the correction factors is also equivalent.

Figure 7-3 compares the STAR, blind STAR, RATZ, and blind RATZ algorithms using 100 sentences for learning the correction factors. In general the STAR algorithms always outperform the RATZ algorithms. This supports our claim that algorithms that modify the distributions of clean speech approach the ideal condition of minimization of the probability of error region much better than those that apply correction factors to the cepstrum data vectors.

The STAR algorithm is able to produce almost the same performance of a fully retrained system up to a SNR of 5 dB. For lower SNRs the assumptions made by the algorithm (***a posteriori***

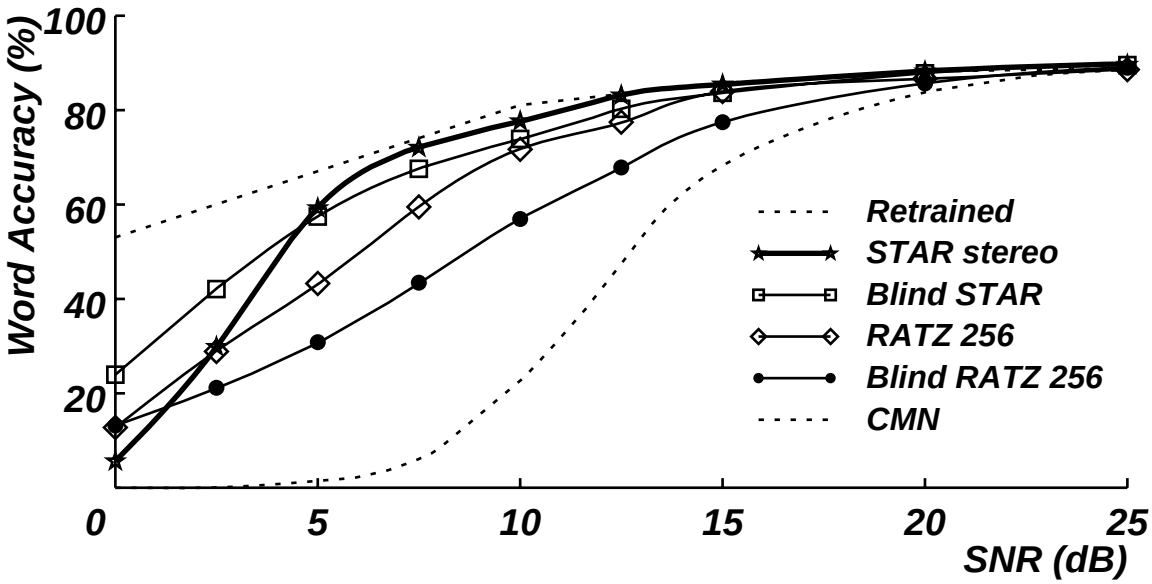

Figure 7-3. Comparison of the stereo STAR, blind STAR, stereo RATZ, and blind RATZ algorithms. The adaptation set was the same for all algorithms and consisted of 100 sentences. The dotted line at the bottom represents the performance of the system with no compensation.

invariance) are not appropriate and recognition accuracy suffers.

7.3. Summary

The STAR family of algorithms modify the mean vectors and covariance matrices of the distributions of clean speech to make them more similar to those of noisy speech. We observe that this family of algorithms outperforms the previously-described RATZ algorithms at almost all SNRs.

The results presented in this section suggest that data-driven compensation algorithm are better when applied to the distributions of clean speech cepstrum. These results support our earlier suggestion (see Appendix A) that compensation of the internal statistics approximates better the performance of a minimum error classifier than compensation of incoming features. In fact, our experimental results show that the STAR algorithm is able to produce almost the performance of a fully retrained system up to a SNR of 5 dB.

We also studied the effect of the initial distributions used for initializing the blind STAR algorithms and have observed that good initial distributions can radically change the performance of the algorithm.

Chapter 8
A Vector Taylor Series Approach to Robust Speech Recognition

In Chapter 4 we described how some of the parameters of the distributions of clean speech are affected by the environment. We have seen how the mean vectors are shifted and how the covariance matrices are compressed. Furthermore, we have shown how in some circumstances the resulting distributions representing noisy speech have no closed-form analytical solutions.

In Chapter 6 and 7 we have described methods that model the effects of the environment on the mean vectors and covariance matrices of clean speech distributions directly from observations of cepstral data, *i.e.*, no model assumptions are directly made. While these techniques, called RATZ and STAR, provide good performance, they are limited by the requirement of extra adaptation data. A sizable amount of noisy data must be observed before performing compensation.

In this chapter we present a new model-based approach to robust speech recognition. Unlike the previously-mentioned methods that learn correction terms directly from adaptation data, we present a group of algorithms that learn these correction terms analytically. These methods reduce the amount of adaptation data to a single sentence, namely, the sentence to be recognized. They take advantage of the extra knowledge provided by the model to reduce the data requirements. These methods are collectively referred to as the Vector Taylor Series (VTS) approach.

8.1. Theoretical assumptions[1]

We will assume that when a vector $\boldsymbol{x}$ representing clean speech is affected by the environment, the resulting vector $\boldsymbol{y}$ representing noisy speech can be described by the following equation

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{a}_1, \boldsymbol{a}_2, \ldots) \tag{8.1}$$

where the function $\boldsymbol{g}(\ )$ is called the environmental function and $\boldsymbol{a}_1$, $\boldsymbol{a}_2$ represents parameters (vectors, scalars, matrices,..) that define the environment. We will assume that the $\boldsymbol{g}(\ )$ function is perfectly known although we will not require knowledge of the environment parameters $\boldsymbol{a}_1$, $\boldsymbol{a}_2$.

Figure 4-1 showed a typical environmental function $\boldsymbol{g}(\ )$ and parameters. This kind of model

1. The notation used in this section as well as the derivation of some of these formulas was introduced in Section 4.1..

for the environment was originally proposed by Acero [1].

In this case the relationship between clean and noisy speech vectors can be expressed in the log-spectral domain as

$$y[k] = x[k] + g(x[k], h[k], n[k]) \tag{8.2}$$

or in vector notation as

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{n}) \tag{8.3}$$

where the environmental function term $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{n})$ is expanded as

$$\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{n}) = \boldsymbol{h} + 10 \log_{10}\left(\boldsymbol{i} + \mathbf{10}^{\frac{\boldsymbol{n}-\boldsymbol{x}-\boldsymbol{h}}{\mathbf{10}}}\right) \tag{8.4}$$

where *i* is a unity vector. The dimension[1] of all vectors is L.

In this case the environmental parameters are the vectors

$$\boldsymbol{h} = \begin{bmatrix} h[0] \\ \ldots \\ h[L-1] \end{bmatrix} \qquad \boldsymbol{n} = \begin{bmatrix} n[0] \\ \ldots \\ n[L-1] \end{bmatrix} \tag{8.5}$$

where each of the components $h[k]$ is the kth log spectral mel component of the power spectrum of the channel $|H(\omega_k)|^2$. Similarly each of the components $n[k]$ is the kth log spectra mel component of the power spectrum of the noise $N(\omega_k)$

As in the case of the RATZ family of algorithms our second assumption in this chapter will be that the clean speech log-spectrum random vector variable can be represented by a mixture of Gaussian distributions

$$p(\boldsymbol{x_t}) = \sum_{k=0}^{K-1} p_k \, N_{\boldsymbol{x}_t}(\mu_{\boldsymbol{x},k}, \Sigma_{\boldsymbol{x},k}) \tag{8.6}$$

---

1. In the SPHINX-II system L is set to 40. See Section 2.1.1.

8.2. Taylor series approximations

Given these assumptions, we would like to compute the distribution of the log spectral vectors of the noisy speech. If the pdf of $\boldsymbol{x}$ and the analytical relationship between the random variables $\boldsymbol{x}$ and $\boldsymbol{y}$ are known, it is possible to compute the distribution for $\boldsymbol{y}$. For the case in which the environmental parameters $\boldsymbol{h}$ and $\boldsymbol{n}$ are deterministic, we showed in Chapter 4 that the resulting distribution for $\boldsymbol{y}$ was non-Gaussian and had the form

$$p(\boldsymbol{y}|\mu_{\boldsymbol{x}}, \Sigma_{\boldsymbol{x}}, \boldsymbol{n}, \boldsymbol{h}) = \left\{(2\pi|\Sigma_{\boldsymbol{x}}|)^{L/2}\left|\left(\boldsymbol{I} - 10^{\frac{\boldsymbol{n}-\boldsymbol{y}}{10}}\right)\right|\right\}^{-1} e^{-\frac{1}{2}\left(\boldsymbol{y}-\boldsymbol{h}-\mu_{\boldsymbol{x}}+10\log_{10}\left(\boldsymbol{i}-10^{\frac{\boldsymbol{n}-\boldsymbol{y}}{10}}\right)\right)^{T}\Sigma_{x}^{-1}\left(\boldsymbol{y}-\boldsymbol{h}-\mu_{\boldsymbol{x}}+10\left(10\log_{10}\left(\boldsymbol{i}-10^{\frac{\boldsymbol{n}-\boldsymbol{y}}{10}}\right)\right)\right)} \tag{8.7}$$

For the more realistic case in which the noise itself is a random variable modeled with a Gaussian distribution $N_{\boldsymbol{n}}(\mu_{\boldsymbol{n}}, \Sigma_{\boldsymbol{n}})$, there is no closed-form solution for $p(\boldsymbol{y}|\mu_{\boldsymbol{x}}, \Sigma_{\boldsymbol{x}}, \boldsymbol{n}, \boldsymbol{h})$. In general, except for environments for which the environmental function $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{....})$ is very simple, the resulting distribution for the log spectral vectors of the noisy speech has no closed-form solution.

In order to obtain a solution for the pdf of $\boldsymbol{y}$, we make the further simplification that the resulting distribution is still Gaussian in nature. As we showed in Chapter 4, this assumption is not unreasonable and it makes the problem mathematically tractable. However, the resulting equations for the mean and covariance of $p(\boldsymbol{y}|\mu_{\boldsymbol{x}}, \Sigma_{\boldsymbol{x}}, \boldsymbol{n}, \boldsymbol{h})$ are still not solvable.

To simplify the problem even further we propose to replace the environmental vector function $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{....})$ by its vector Taylor series approximation. This simplification only requires that the environmental function $\boldsymbol{g}(\ )$ be analytical. Under this assumption the resulting relationship between the random variable $\boldsymbol{x}$ and $\boldsymbol{y}$ becomes

$$\begin{aligned}\boldsymbol{y} &= \boldsymbol{x} + \boldsymbol{g}(\boldsymbol{x}_0, \boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{....}) + \boldsymbol{g}'(\boldsymbol{x}_0, \boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{....})(\boldsymbol{x}-\boldsymbol{x}_0) \\ &\quad + \frac{1}{2}\boldsymbol{g}''(\boldsymbol{x}_0, \boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{....})(\boldsymbol{x}-\boldsymbol{x}_0)(\boldsymbol{x}-\boldsymbol{x}_0) + ...\end{aligned} \tag{8.8}$$

For the type of environment described in Figure 4-1 the Vector Taylor approximation is

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{x} + \boldsymbol{g}(\boldsymbol{x_0}, \boldsymbol{h}, \boldsymbol{n}) + \boldsymbol{g}'(\boldsymbol{x_0}, \boldsymbol{h}, \boldsymbol{n})(\boldsymbol{x} - \boldsymbol{x_0}) \\ &\quad + \frac{1}{2}\boldsymbol{g}''(\boldsymbol{x_0}, \boldsymbol{h}, \boldsymbol{n})(\boldsymbol{x} - \boldsymbol{x_0})(\boldsymbol{x} - \boldsymbol{x_0}) + \ldots \end{aligned} \tag{8.9}$$

where the term $\boldsymbol{g}(\boldsymbol{x_0}, \boldsymbol{h}, \boldsymbol{n})$ is expanded as

$$\boldsymbol{g}(\boldsymbol{x}_0, \boldsymbol{h}, \boldsymbol{n}) = \boldsymbol{h} + 10\log_{10}\left(\boldsymbol{i} + 10^{\frac{\boldsymbol{n} - \boldsymbol{x}_o - \boldsymbol{h}}{10}}\right) \tag{8.10}$$

i.e., the environment vector function $\boldsymbol{g}(.)$ evaluated at the vector point $\boldsymbol{x}_0$. The term $\boldsymbol{g}'(\boldsymbol{x_0}, \boldsymbol{h}, \boldsymbol{n})$ is the derivative of the environment vector function $\boldsymbol{g}(.)$ with respect to the vector variable $\boldsymbol{x}$ evaluated at the vector point $\boldsymbol{x}_0$

$$\boldsymbol{g}'(\boldsymbol{x_0}, \boldsymbol{h}, \boldsymbol{n}) = \boldsymbol{diag}\left(-\left(1 + 10^{\frac{x_{0,i} + h_i - n_i}{10}}\right)^{-1}\right) \tag{8.11}$$

i.e., a diagonal matrix with L entries in the main diagonal, each of the form $\left(1 + 10^{\frac{x_{0,i} + h_i - n_i}{10}}\right)^{-1}$.

Higher order derivatives result in tensors [3] of order three and higher. For example, the second derivative of the vector function $\boldsymbol{g}(.)$ with respect to the vector variable $\boldsymbol{x}$ evaluated at the vector point $\boldsymbol{x}_0$ would be

$$\boldsymbol{g}''(\boldsymbol{x_0}, \boldsymbol{h}, \boldsymbol{n}) = \boldsymbol{f}''_{\boldsymbol{ijk}} = \begin{cases} 10^{\frac{x_{0,i} + h_j - n_k}{10}}\left(1 + 10^{\frac{x_{0,i} + h_j - n_k}{10}}\right)^{-2}\frac{ln10}{10} & i = j = k \\ 0 & otherwise \end{cases} \tag{8.12}$$

i.e., a diagonal tensor with only the diagonal elements different from zero. In this particular type of environment higher order derivatives of $\boldsymbol{g}(.)$ always result in diagonal tensors.

8.3. Truncated Vector Taylor Series approximations

In this section we compute the mean vector and covariance matrix of the noisy vector $\boldsymbol{y}$ assuming that we have perfect knowledge of the environmental parameters $\boldsymbol{h}$ and $\boldsymbol{n}$.

The original expression for the mean vector is

$$\mu_y = E(\boldsymbol{y}) = E(\boldsymbol{x} + \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{n})) = E(\boldsymbol{x}) + E(\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{n})) \tag{8.13}$$

approximating the environmental function $\boldsymbol{g}(.)$ by its vector Taylor series the above expression is simplified to

$$\begin{aligned}\mu_y = E(\boldsymbol{y}) = E(\boldsymbol{x} + \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{n})) = E(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x_0}, \boldsymbol{h}, \boldsymbol{n}) + \boldsymbol{g}'(\boldsymbol{x_0}, \boldsymbol{n}, \boldsymbol{h})E(\boldsymbol{x} - \boldsymbol{x_0}) + \\ \frac{1}{2}\boldsymbol{g}''(\boldsymbol{x_0}, \boldsymbol{n}, \boldsymbol{h})E((\boldsymbol{x} - \boldsymbol{x_0})(\boldsymbol{x} - \boldsymbol{x_0})) + ...\end{aligned} \tag{8.14}$$

where each of the expected value terms can be easily computed.

For the covariance matrix the original expression is

$$\Sigma_y = E((\boldsymbol{y}\boldsymbol{y}^T)) - \mu_y\mu_y^T = E(\boldsymbol{x}\boldsymbol{x}^T) + E(\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{n})\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{n})^T) + 2E(\boldsymbol{x}\ \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{n})^T) - \mu_y\mu_y^T \tag{8.15}$$

approximating each of the $\boldsymbol{g}(.)$ terms by its vector Taylor series results also in a expression where each of the individual terms is solvable.

To simplify the expressions we retain terms of the Taylor series up to a certain order. This will result in approximate equations describing the mean and covariance matrices of the log spectrum random variable $\boldsymbol{y}$ for noisy speech. The more terms of the Taylor series we keep, the better the approximation.

For example, for a Taylor series of order zero the expressions for the mean vector and covariance matrix are

$$\mu_y = E(\boldsymbol{y}) \cong E(\boldsymbol{x} + \boldsymbol{g}(\boldsymbol{x}_0, \boldsymbol{h}, \boldsymbol{n})) = \mu_{\boldsymbol{x}} + \boldsymbol{g}(\boldsymbol{x}_0, \boldsymbol{h}, \boldsymbol{n}) \tag{8.16}$$

$$\begin{aligned}\Sigma_y = E\{(\boldsymbol{y} - \mu_y)(\boldsymbol{y} - \mu_y)^T\} \cong E\{&(\boldsymbol{x} + \boldsymbol{g}(\boldsymbol{x}_0, \boldsymbol{h}, \boldsymbol{n}) - \mu_{\boldsymbol{x}} - \boldsymbol{g}(\boldsymbol{x}_0, \boldsymbol{h}, \boldsymbol{n})) \\ &(\boldsymbol{x} + \boldsymbol{g}(\boldsymbol{x}_0, \boldsymbol{h}, \boldsymbol{n}) - \mu_{\boldsymbol{x}} - \boldsymbol{g}(\boldsymbol{x}_0, \boldsymbol{h}, \boldsymbol{n}))^T\}\end{aligned} \tag{8.17}$$

$$\Sigma_y \cong E\{(\boldsymbol{x} - \mu_{\boldsymbol{x}})(\boldsymbol{x} - \mu_{\boldsymbol{x}})^T\} = \Sigma_{\boldsymbol{x}}$$

From these expressions we conclude that a zeroth order vector Taylor series models the effect of the environment on clean speech distributions only as a shift of the mean.

For a Taylor series of order one, the expressions for the mean and covariance matrix are

$$\begin{aligned} \mu_y &= E(\boldsymbol{y}) \cong E(\boldsymbol{x} + \boldsymbol{g}(\boldsymbol{x}_0, \boldsymbol{h}, \boldsymbol{n}) + \boldsymbol{g}'(\boldsymbol{x}_0, \boldsymbol{h}, \boldsymbol{n})(\boldsymbol{x} - \boldsymbol{x}_0)) \\ \mu_y &\cong (\boldsymbol{I} + \boldsymbol{g}'(\boldsymbol{x}_0, \boldsymbol{h}, \boldsymbol{n}))\mu_x + \boldsymbol{g}(\boldsymbol{x}_0, \boldsymbol{h}, \boldsymbol{n}) - \boldsymbol{g}'(\boldsymbol{x}_0, \boldsymbol{h}, \boldsymbol{n})\boldsymbol{x}_0 \end{aligned} \tag{8.18}$$

$$\begin{aligned} \Sigma_y &= E\{\boldsymbol{y}\boldsymbol{y}^T\} - \mu_y\mu_y^T \\ E\{yy^T\} &\cong E\{(\boldsymbol{x} + \boldsymbol{g}(\boldsymbol{x}_0, \boldsymbol{h}, \boldsymbol{n}) + \boldsymbol{g}'(\boldsymbol{x}_0, \boldsymbol{h}, \boldsymbol{n})(\boldsymbol{x} - \boldsymbol{x}_0))(\boldsymbol{x} + \boldsymbol{g}(\boldsymbol{x}_0, \boldsymbol{h}, \boldsymbol{n}) + \\ &\qquad + \boldsymbol{g}'(\boldsymbol{x}_0, \boldsymbol{h}, \boldsymbol{n})(\boldsymbol{x} - \boldsymbol{x}_0))^T\} \\ \mu_y\mu_y^T &\cong ((\boldsymbol{I} + \boldsymbol{g}'(\boldsymbol{x}_0, \boldsymbol{h}, \boldsymbol{n}))\mu_x + \boldsymbol{g}(\boldsymbol{x}_0, \boldsymbol{h}, \boldsymbol{n}) - \boldsymbol{g}'(\boldsymbol{x}_0, \boldsymbol{h}, \boldsymbol{n})\boldsymbol{x}_0)((I + \boldsymbol{g}'(\boldsymbol{x}_0, \boldsymbol{h}, \boldsymbol{n}))\mu_x + \\ &\qquad + \boldsymbol{g}(\boldsymbol{x}_0, \boldsymbol{h}, \boldsymbol{n}) - \boldsymbol{g}'(\boldsymbol{x}_0, \boldsymbol{h}, \boldsymbol{n})\boldsymbol{x}_0)^T \\ \Sigma_y &\cong (\boldsymbol{I} + \boldsymbol{g}'(\boldsymbol{x}_0, \boldsymbol{h}, \boldsymbol{n}))\Sigma_x(\boldsymbol{I} + \boldsymbol{g}'(\boldsymbol{x}_0, \boldsymbol{h}, \boldsymbol{n}))^T \end{aligned} \tag{8.19}$$

From these expressions we conclude that a first order vector Taylor series models the effect of the environment on clean speech distributions as a shift of the mean and as a compression of the covariance matrix. This can be proved by realizing that each of the elements of the $(\boldsymbol{I} + \boldsymbol{g}'(\boldsymbol{x}_0, \boldsymbol{h}, \boldsymbol{n}))$ matrix that multiplies the covariance matrix Σ_x is smaller than one

$$(\boldsymbol{I} + \boldsymbol{g}'(\boldsymbol{x}_0, \boldsymbol{h}, \boldsymbol{n})) = \boldsymbol{diag}\left(1 - \left(1 + 10^{\frac{x_{0,i} + h_i - n_i}{10}}\right)^{-1}\right) = \boldsymbol{diag}\left(\left(1 + 10^{-\frac{x_{0,i} + h_i - n_i}{10}}\right)^{-1}\right) \tag{8.20}$$

8.3.1. Comparing Taylor series approximations to exact solutions

In this section we explore the accuracy of the Taylor series approximation for the kind of environment described in Figure 4-1. We compare the actual values of the means and variances of noisy distributions at different signal-to-noise ratios with those computed using Taylor series approximations of different orders.

The experiments were performed on artificially generated data in a similar way to the experi-

ments done on Chapter 4. A clean set $X = \{x_0, x_1, ..., x_{S-1}\}$ of one-dimensional data points was randomly generated according to a Gaussian distribution. Another set of noise data $N = \{n_0, n_1, ..., n_{S-1}\}$ was randomly produced according to a Gaussian distribution with a small variance. Both sets were combined according to the formula

$$y = x + 10 \log_{10}\left(1 + 10^{\frac{n-y}{10}}\right) \tag{8.21}$$

resulting in a noisy data set $Y = \{y_0, y_1, ..., y_{S-1}\}$. The mean and variance of this set was estimated directly from the data as

$$\mu_y = \frac{1}{S}\sum_{i=0}^{S-1} y_t \qquad \sigma_y^2 = \frac{1}{S-1}\sum_{i=0}^{S-1} (y_t - \mu_y)^2 \tag{8.22}$$

The mean and the variance were also computed using Taylor series approximations of different orders using the previously described formulas. The experiment was repeated at different signal to noise ratios[1].

Figure 8-1 compares Taylor series approximations of order zero and two with the actual values of the mean. As can be seen, a Taylor approximation of order zero seems to capture most of the effect of the environment on the mean. Figure 8-2 compares Taylor series approximations of order zero and one with the actual value of the variance of the noisy data. A Taylor series of order one seems to be able to capture most of the effect of the environment on the variance. From these simulations we conclude that a Taylor series approximation of order one might be enough to capture most of the effects of the environment of log spectral distributions of clean speech.

1. The SNR here is defined as $\mu_x - \mu_n$, the difference between the means of the distributions for clean and noisy speech.

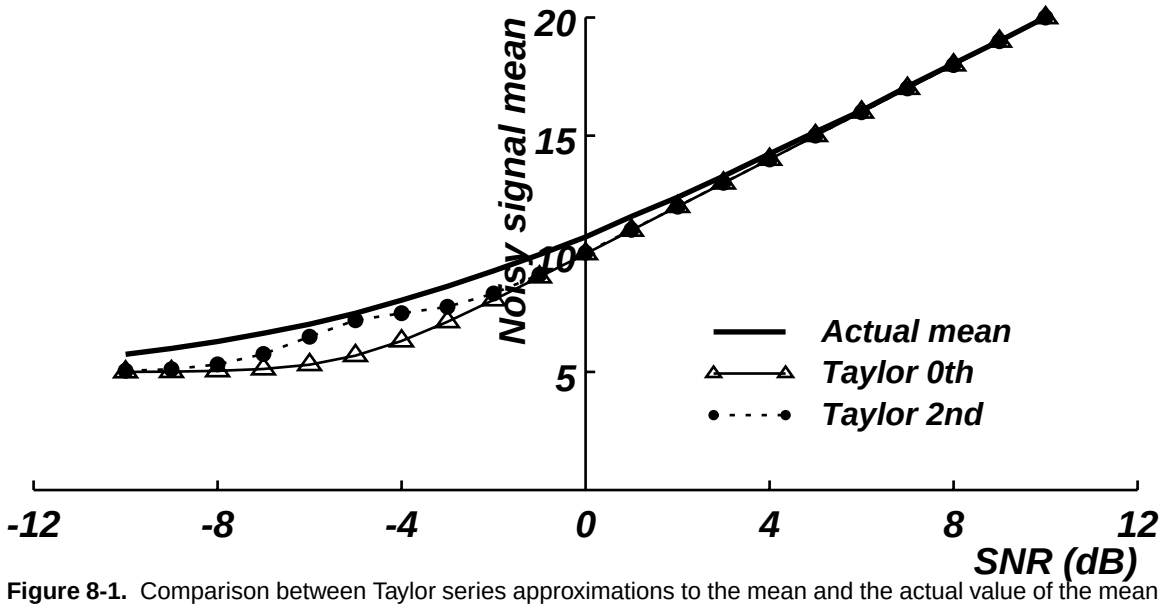

Figure 8-1. Comparison between Taylor series approximations to the mean and the actual value of the mean of noisy data. A Taylor series of order zero seems to capture most of the effect.

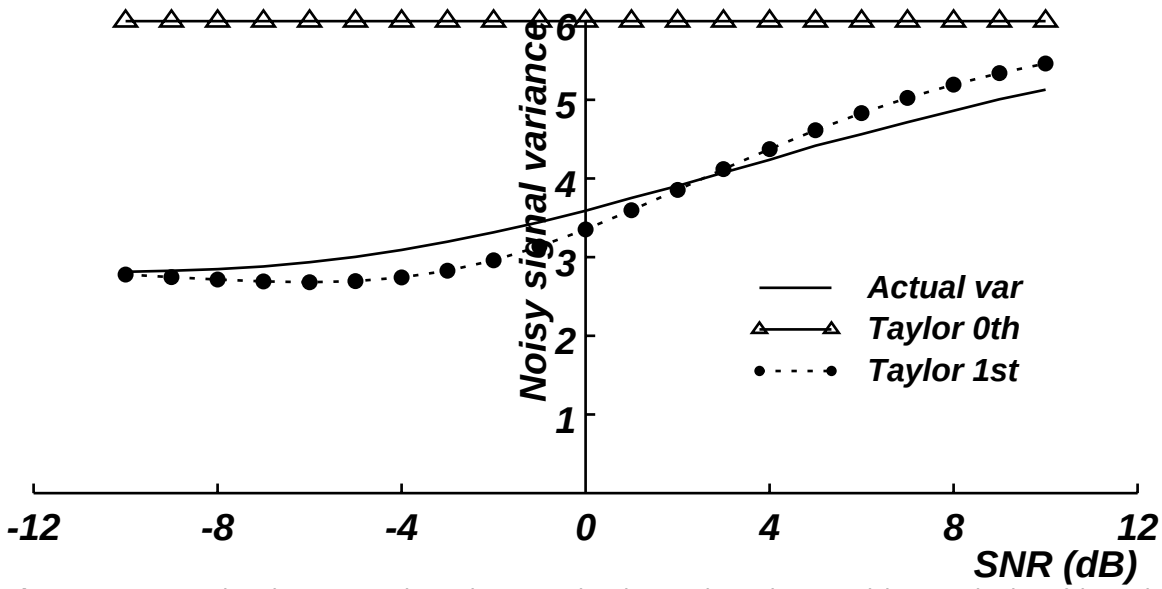

Figure 8-2. Comparison between Taylor series approximations to the variance and the actual value of the variance of noisy data. A Taylor series of order one seems to capture most of the effect.

8.4. A Maximum Likelihood formulation for the case of unknown environmental parameters

In the previous sections we described the generic use of vector Taylor series approximations as a way of solving for the means and covariances of the noisy speech log-spectrum distributions. However, knowledge of the environment parameters was assumed in the above discussion. In prac-

tical situations we might know the generic form of the environmental function $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{....})$ but not the exact values of the environmental parameters $\boldsymbol{a}_1$, $\boldsymbol{a}_2$. Therefore, a method that is able to estimate these parameters from noisy observations is necessary. Once these environmental parameters are estimated we can directly compute the log-spectrum mean vectors and covariance matrices for noisy speech using a vector Taylor series approximation as described in the previous sections.

In this section we outline an iterative procedure that combines a Taylor series approach with a maximum likelihood formulation to estimate the environmental parameters $\boldsymbol{a}_1$, $\boldsymbol{a}_2$ from noisy observations. The algorithm then estimates the mean vector and covariance matrix of the distributions of log spectral vectors for the noisy speech. The procedure is particularized for the type of environments described in Figure 4-1.

Given the following assumptions

- A set of noisy speech log-spectrum vectors $\boldsymbol{Y} = \{\boldsymbol{y}_0, \boldsymbol{y}_1, ..., \boldsymbol{y}_{S-1}\}$
- A distribution for the clean speech log-spectrum vector random variable $p(\boldsymbol{x}_t) = \sum_{k=0}^{K-1} p_k \, N_{\boldsymbol{x}_t}(\mu_{\boldsymbol{x},k}, \Sigma_{\boldsymbol{x},k})$
- A set of initial values for the environmental parameters $\boldsymbol{n}_0 = min\{\boldsymbol{Y}\}$, $\boldsymbol{h}_0 = mean\{\boldsymbol{Y}\} - \mu_{\boldsymbol{x}}$, and $\boldsymbol{x}_0 = \mu_{\boldsymbol{x}}$

we now define a vector Taylor[1] series around the set of points $\mu_{\boldsymbol{x}}$, $\boldsymbol{n}_0$ and $\boldsymbol{h}_0$

$$\begin{aligned}\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{n}, \boldsymbol{h}) \cong \boldsymbol{x} + \boldsymbol{g}(\mu_{\boldsymbol{x}}, \boldsymbol{n}_0, \boldsymbol{h}_0) + \nabla_{\boldsymbol{x}} \boldsymbol{g}(\mu_{\boldsymbol{x}}, \boldsymbol{n}_0, \boldsymbol{h}_0)\ (\boldsymbol{x} - \mu_{\boldsymbol{x}}) + \\ \nabla_{\boldsymbol{n}} \boldsymbol{g}(\mu_{\boldsymbol{x}}, \boldsymbol{n}_0, \boldsymbol{h}_0)\ (\boldsymbol{n} - \boldsymbol{n}_0) + \nabla_{\boldsymbol{h}} \boldsymbol{g}(\mu_{\boldsymbol{x}}, \boldsymbol{n}_0, \boldsymbol{h}_0)\ (\boldsymbol{h} - \boldsymbol{h}_0) + ...\end{aligned} \tag{8.23}$$

Given a particular order in the Taylor approximation we compute the mean vector and covariance matrix of noisy speech as functions of the unknown variables $\boldsymbol{n}$ and $\boldsymbol{h}$.

1. Since the environmental function is a vector function of vectors, *i.e.* a vector field or in more general terms a tensor field, we are forced to use a new notation for partial derivatives using the gradient operator $\nabla_{\boldsymbol{x}} \boldsymbol{g}(\ )$. Notice also that this is only valid for the first derivative. For higher order derivatives we would be forced to use tensor notation. See [3] for more details on tensor notation and tensor calculus.

For example, for the case of a first order Taylor series approximation we obtain

$$\begin{gathered}\mu_y \cong \mu_x + (\boldsymbol{I} + \nabla_{\boldsymbol{h}}\, \boldsymbol{g}(\mu_x, \boldsymbol{n}_0, \boldsymbol{h}_0))\boldsymbol{h} + \nabla_{\boldsymbol{n}}\, \boldsymbol{g}(\mu_x, \boldsymbol{n}_0, \boldsymbol{h}_0)\ \boldsymbol{n} + \\ \boldsymbol{g}(\mu_x, \boldsymbol{n}_0, \boldsymbol{h}_0) - \nabla_{\boldsymbol{h}}\, \boldsymbol{g}(\mu_x, \boldsymbol{n}_0, \boldsymbol{h}_0)\ \boldsymbol{h}_0 - \nabla_{\boldsymbol{n}}\, \boldsymbol{g}(\mu_x, \boldsymbol{n}_0, \boldsymbol{h}_0)\ \boldsymbol{n}_0 \\ \mu_y \cong \mu_x + \boldsymbol{w}(\boldsymbol{h}, \boldsymbol{n}, \mu_x, \boldsymbol{n}_0, \boldsymbol{h}_0)\end{gathered} \tag{8.24}$$

$$\begin{gathered}\Sigma_y \cong (\boldsymbol{I} + \nabla_{\boldsymbol{h}}\, \boldsymbol{g}(\mu_x, \boldsymbol{n}_0, \boldsymbol{h}_0))\Sigma_x(\boldsymbol{I} + \nabla_{\boldsymbol{h}}\, \boldsymbol{g}(\mu_x, \boldsymbol{n}_0, \boldsymbol{h}_0))^T \\ \Sigma_y \cong \boldsymbol{W}(\mu_x, \boldsymbol{n}_0, \boldsymbol{h}_0)\ \Sigma_x\ \boldsymbol{W}(\mu_x, \boldsymbol{n}_0, \boldsymbol{h}_0)^T\end{gathered} \tag{8.25}$$

In this case, the expression for the mean of the log-spectral distribution of noisy speech is a linear function of the unknown variables $\boldsymbol{n}$ and $\boldsymbol{h}$ can be rewritten as

$$\begin{gathered}\mu_y \cong \boldsymbol{a} + \boldsymbol{B}\ \boldsymbol{h} + \boldsymbol{C}\ \boldsymbol{n} \\ \boldsymbol{a} = \mu_x + \boldsymbol{g}(\mu_x, \boldsymbol{n}_0, \boldsymbol{h}_0) - \nabla_{\boldsymbol{h}}\, \boldsymbol{g}(\mu_x, \boldsymbol{n}_0, \boldsymbol{h}_0)\ \boldsymbol{h}_0 - \nabla_{\boldsymbol{n}}\, \boldsymbol{g}(\mu_x, \boldsymbol{n}_0, \boldsymbol{h}_0)\ \boldsymbol{n}_0 \\ \boldsymbol{B} = \boldsymbol{I} + \nabla_{\boldsymbol{h}}\, \boldsymbol{g}(\mu_x, \boldsymbol{n}_0, \boldsymbol{h}_0) \qquad \boldsymbol{C} = \nabla_{\boldsymbol{n}}\, \boldsymbol{g}(\mu_x, \boldsymbol{n}_0, \boldsymbol{h}_0)\end{gathered} \tag{8.26}$$

while the expression for the covariance matrix is dependent only on the initial values μ_x, $\boldsymbol{n}_0$, and $\boldsymbol{h}_0$, which are known.

Therefore, given the observed noisy data we can define a likelihood function

$$L(\boldsymbol{Y} = \{\boldsymbol{y}_0, \boldsymbol{y}_1, ..., \boldsymbol{y}_{S-1}\}) = \sum_{t=0}^{S-1} \log(p(\boldsymbol{y}_t | \boldsymbol{h}, \boldsymbol{n})) \tag{8.27}$$

where the only unknowns are the variables $\boldsymbol{n}$ and $\boldsymbol{h}$. To find these unknowns we can use a traditional iterative EM approach. Appendix D describes in detail all the steps needed to estimate these solutions.

Once we obtain the solutions for the variables $\boldsymbol{n}$ and $\boldsymbol{h}$ we readjust the Taylor approximations to the mean vectors and covariance matrices by substituting $\boldsymbol{n}_0$ by $\boldsymbol{n}$ and $\boldsymbol{h}_0$ by $\boldsymbol{h}$. Once the Taylor approximations are readjusted we iterate the procedure by defining again the new mean vectors and covariance matrices of the noisy speech and redefining a maximum likelihood function. The whole procedure is stopped when no significant change is observed in the estimated values of $\boldsymbol{n}$ and $\boldsymbol{h}$.

Figure 8-3 shows a block diagram of the whole estimation procedure.

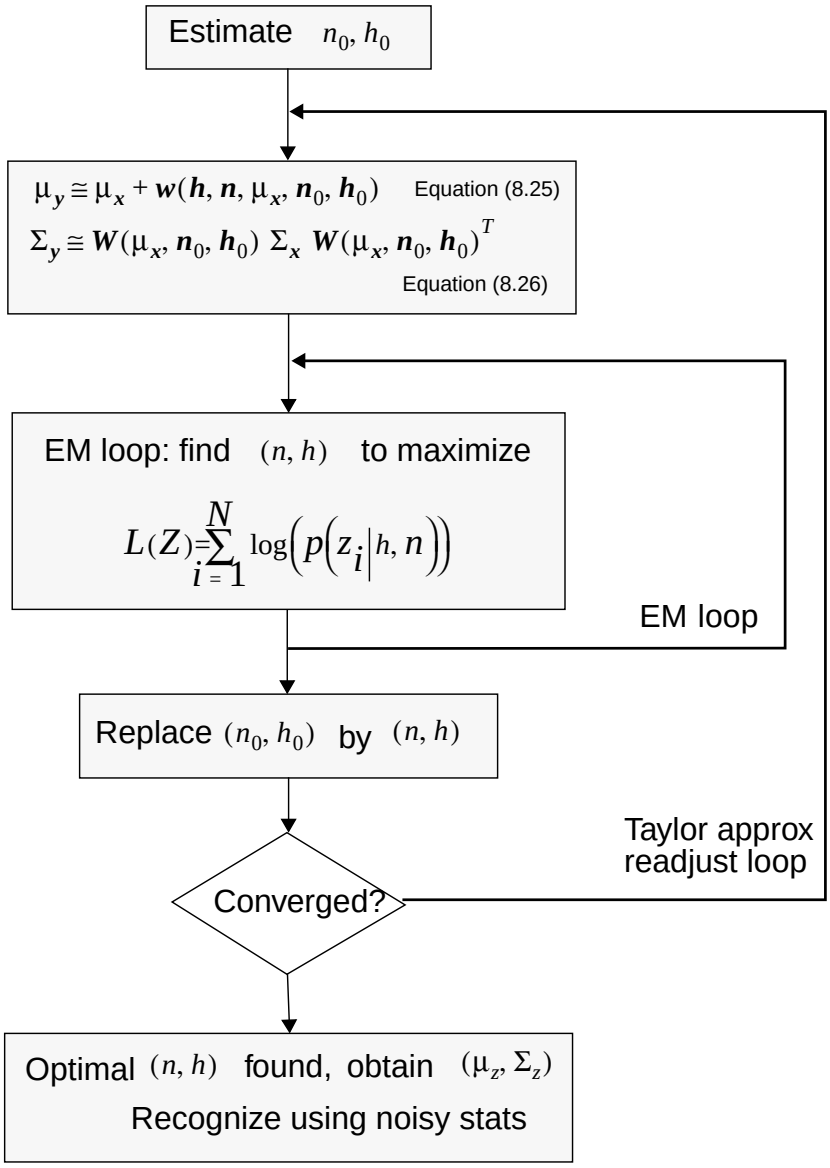

Figure 8-3. Flow chart of the vector Taylor series algorithm of order one for the case of unknown environmental parameters. Given a small amount of data the environmental parameters are learned in an iterative procedure.

8.5. Data compensation vs. HMM mean and variance adjustment

As we mentioned in Chapter 4, once the distributions of the noisy speech log-spectrum are known there are two choices for environment compensation. The first choice consists of performing the classification with the feature vectors directly and using the distributions of the noisy speech

feature vectors. The second choice consists of applying correction factors to features representing the noisy speech to "clean" them and performing the classification using distributions derived from clean speech.

The VTS algorithm as presented here can be used for both cases. However, because a speech feature compensation approach is easier to implement and can be implemented independently of the recognition engine the experimental results we will present correspond to the second case. Therefore an additional step is needed to perform data compensation once the distributions of the noisy speech are learned via the VTS method.

We propose to use an approximated Minimum Mean Squared Error (MMSE) method similar to the one used in the RATZ family of algorithms

$$\hat{\boldsymbol{x}}_{MMSE} = E(\boldsymbol{x}|\boldsymbol{y}) = \int_{\boldsymbol{X}} \boldsymbol{x}.p(\boldsymbol{x}|\boldsymbol{y})d\boldsymbol{x} \tag{8.28}$$

expressing $\boldsymbol{x}$ as a function of $\boldsymbol{y}$ and the environmental function $\boldsymbol{g}(\ )$

$$\begin{aligned} \hat{\boldsymbol{x}}_{MMSE} &= \boldsymbol{y} - \int_{\boldsymbol{X}} \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{n}, \boldsymbol{h}) p(\boldsymbol{x}|\boldsymbol{y}) d\boldsymbol{x} = \boldsymbol{y} - \int_{\boldsymbol{X}} \sum_{k=0}^{K-1} \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{n}, \boldsymbol{h}) p(\boldsymbol{x}, k|\boldsymbol{y}) d\boldsymbol{x} \\ \hat{\boldsymbol{x}}_{MMSE} &= \boldsymbol{y} - \sum_{k=0}^{K-1} P[k|\boldsymbol{y}] \int_{\boldsymbol{X}} \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{n}, \boldsymbol{h}) p(\boldsymbol{x}|k, \boldsymbol{y}) d\boldsymbol{x} \end{aligned} \tag{8.29}$$

and approximating $\boldsymbol{g}(\ )$ by its Taylor series approximation we will get different solutions. For example, for a Taylor series approximation of order zero we obtain the following result

$$\hat{\boldsymbol{x}}_{MMSE} \cong \boldsymbol{y} - \sum_{k=0}^{K-1} P[k|\boldsymbol{y}] \int_{\boldsymbol{X}} \boldsymbol{g}(\mu_{k,\boldsymbol{x}}, \boldsymbol{n}, \boldsymbol{h}) p(\boldsymbol{x}|k, \boldsymbol{y}) d\boldsymbol{x} = \boldsymbol{y} - \sum_{k=0}^{K-1} P[k|\boldsymbol{y}] \boldsymbol{g}(\mu_{k,\boldsymbol{x}}, \boldsymbol{n}, \boldsymbol{h}) \tag{8.30}$$

For a Taylor series of order one we obtain the following result

$$y \cong x + \boldsymbol{g}(\mu_{k,\boldsymbol{x}}, \boldsymbol{n}, \boldsymbol{h}) + \boldsymbol{g}'(\mu_{k,\boldsymbol{x}}, \boldsymbol{n}, \boldsymbol{h})(\boldsymbol{x} - \mu_{k,\boldsymbol{x}}) \tag{8.31}$$

$$\hat{\boldsymbol{x}}_{MMSE} \cong \sum_{k=0}^{K-1} P[k|\boldsymbol{y}] \int_{\boldsymbol{X}} (y - \boldsymbol{g}(\mu_{k,\boldsymbol{x}}, \boldsymbol{n}, \boldsymbol{h})) p(\boldsymbol{x}|k, \boldsymbol{y}) dx$$
$$\hat{\boldsymbol{x}}_{MMSE} = \boldsymbol{y} - \sum_{k=0}^{K-1} P[k|\boldsymbol{y}] \boldsymbol{g}(\mu_{k,\boldsymbol{x}}, \boldsymbol{n}, \boldsymbol{h}) \tag{8.32}$$

Solutions for higher order approximations are also possible.

8.6. Experimental results

In this section we compare the VTS algorithms of order zero and one with other environmental compensation algorithms. The experiments described here were performed on the 5,000-word Wall Street Journal 1993 evaluation set with white Gaussian noise added at several SNRs. As in the previous experiments with this database, the upper dotted line represents the performance of the system when fully trained on noisy data while the lower dotted line represents the performance of the system when no compensation is used.

Figure 8-4 compares the VTS algorithms with the CDCN algorithm. All the algorithms used

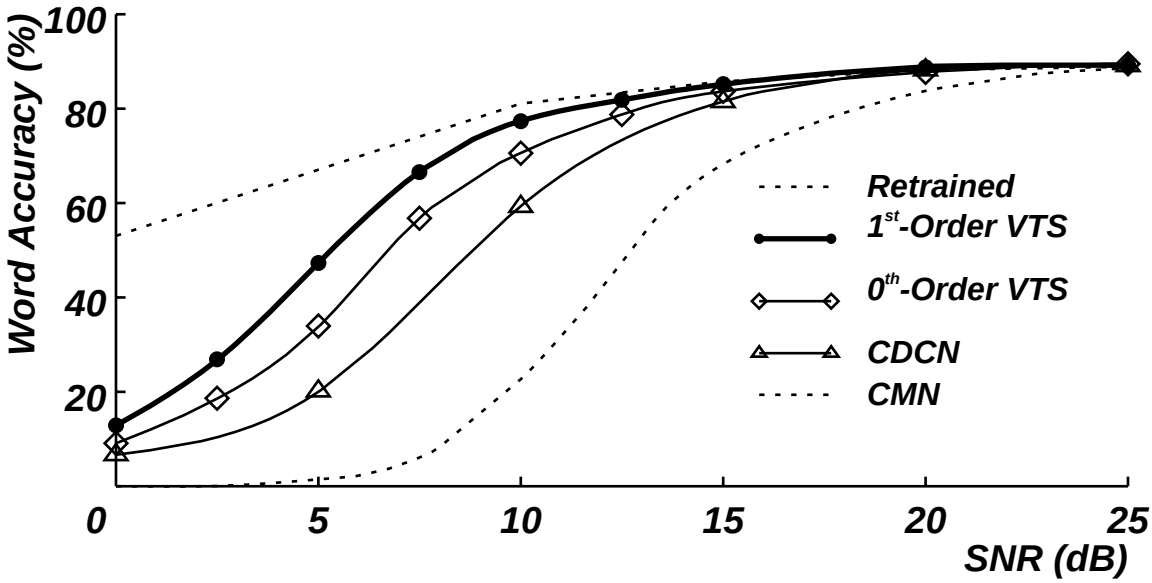

Figure 8-4. Comparison of the VTS algorithms of order zero and one with CDCN. The VTS algorithms outperform CDCN at all SNRs.

256 Gaussians to represent the distributions of clean speech feature vectors. The VTS algorithms used a 40-dimensional log spectral vector as features. The CDCN algorithm used a 13-dimensional cepstral vector. The VTS algorithm of order one outperforms the VTS algorithm of order zero

which in turn it outperforms the CDCN algorithm at all SNRs. The approximations of the VTS algorithm model the effects of the environment on the distributions of clean speech log spectrum better than those of CDCN.

Figure 8-5 compares the VTS algorithms with the best configurations of the RATZ and STAR

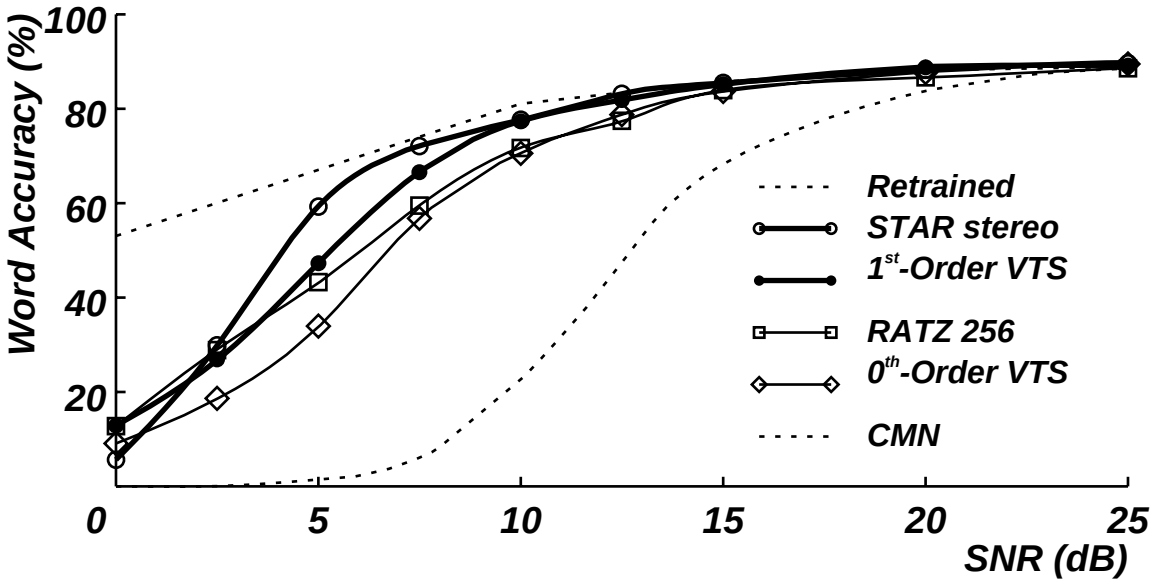

Figure 8-5. Comparison of the VTS algorithms of order zero and one with the stereo-based RATZ and STAR algorithms. The VTS algorithm of order one performs as well as the STAR algorithm up to a SNR or 10 dB. For lower SNRs only the STAR algorithm produces lower error rates.

algorithm. For this experiments the STAR and RATZ algorithm were trained with one hundred adaptation sentences per SNR. The environment was perfectly known. The VTS algorithms worked on a sentence by sentence level using the same sentence for learning the distributions of the noisy speech log spectrum and then compensating the same sentence.

We can observe how the VTS algorithm of order one yields similar recognition accuracy as the STAR algorithm or a fully retrained system. Only at lower than 10 dB the STAR algorithm outperforms the VTS algorithm of order one. Perhaps this is due to the fact that it is at this lower SNRs where algorithms that modify the distributions of clean speech cepstrum approximate better the optimal minimum error classifier.

8.7. Experiments using real data

In this section we describe a series of experiments in which we study the performance of the algorithms proposed in this thesis with real databases.

The first experiment compares the performance of the CDCN and the VTS of order one algorithms. The task consists of one hundred sentences collected at a seven different distances between the speaker and a desktop microphone. Each of the one hundred sentences subsets is different in terms of speech and noise. However, the same speakers read the same sentences at each mike-to-mouth distance. The vocabulary size of the task was about 2,000 words. The sentences were recorded in an office environment with computer background noise as well as some impulsive noises. The next figure illustrates our experimental results.

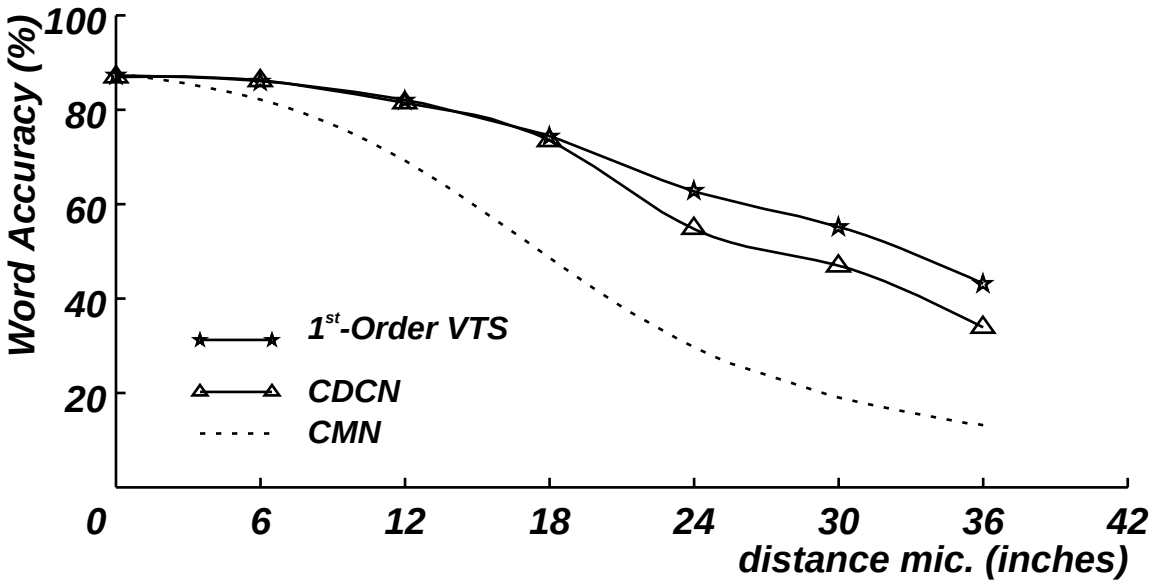

Figure 8-6. Comparison of the VTS of order one algorithm with the CDCN algorithm on a real database. Each points consists of one hundred sentences collected at different distances from the mouth of the speaker to the microphone.

As we can see both algorithms behave similarly up to a distance of 18 inches. At that point the VTS algorithm improves the recognition rate.

The second experiment described in this section compares the performance of all the algorithms introduced in this thesis and the CDCN algorithm. The database used was the 1994 Spoke 10 evaluation set described in Section 2.2.1. This database consists of three sets of 113 sentences contaminated with car-recorded noise at different signal-to-noise levels. The recognition system used was trained on the whole Wall Street Journal corpora consisting of about 37,000 sentences or 57 hours of speech. Male and female models were produced with about 10,000 senonic clusters each.

Each sentences was recognized with the male and female models and the most likely candidate chosen as the correct one.

The CDCN and VTS algorithms of order zero and one were ran on a sentence by sentence basis. Even though the Spoke 10 evaluations conditions allow for the use of some extra noise samples to correctly estimate the noise, we did not make use of those samples.

The RATZ algorithm used a statistical representation of the clean speech based on 256 Gaussians. Fifteen sets of correction factors were learned from stereo-recorded data distributed by NIST. The data consisted of sets of 53 utterances contaminated with noise collected from three different cars and added at five different SNRs. An interpolated version of the RATZ algorithm was used.

Using the same stereo-recorded data fifteen different sets of STAR-based correction factors were also estimated. For each of the three evaluation sets the three most likely correction sets were applied to the HMM means and variances and the resulting hypothesis produced by the decoder were combined and the most likely chosen.

Figure 8-7 presents our experimental results. As we can see most of the algorithms perform quite well and differences in recognition performance are only clear and apparent at the lowest SNR. In that case the interpolated RATZ algorithm exhibits the greatest accuracy.

Contrary with our results based on artificial data the STAR algorithm does not outperform the results achieved with RATZ. The lack of a proper mechanism for interpolation in STAR might be responsible for this lower performance. In general, all the algorithms perform similarly perhaps because the SNR of each of the testing sets is high enough.

8.8. Computational complexity

In a very informal study of the computational complexity of the compensation algorithms proposed in this thesis we observed how long it took to compensate a set of five noisy sentences using the RATZ, CDCN, VTS-0, and VTS-1 algorithms.

No optimization effort was made to improve the performance of each of the algorithms. The experiments were done on a DEC Alpha workstation model 3K600. The numbers we report in Figure 8-8 represent the number of seconds it took to compensate the five sentences divided by the duration of the five sentences.

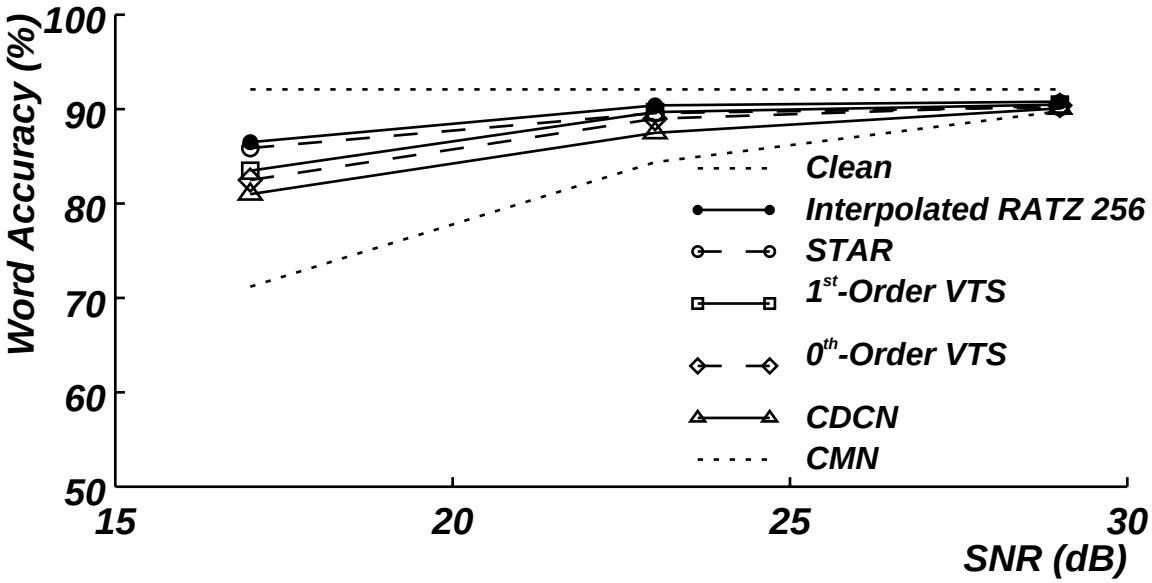

Figure 8-7. Comparison of several algorithms on the 1994 Spoke 10 evaluation set. The upper line represents the accuracy on clean data while the lower dotted line represents the recognition accuracy with no compensation. The RATZ algorithm provides the best recognition accuracy at all SNRs.

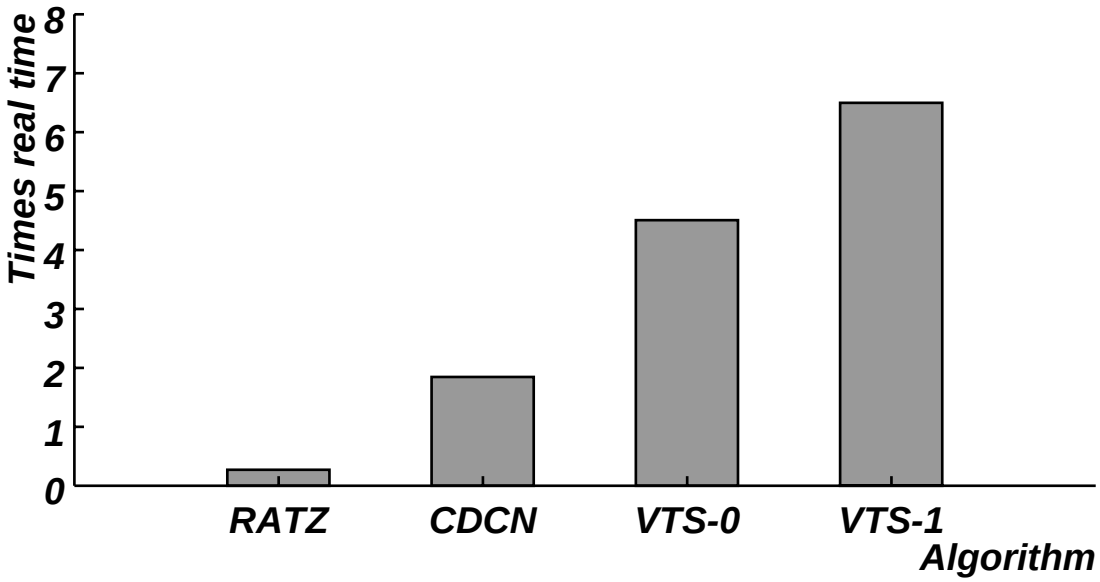

Figure 8-8. Comparison of the real time performance of the VTS algorithms with the RATZ and CDCN compensation algorithms. VTS-1 requires about 6 times the computational effort of CDCN.

The RATZ experiment assumes that the environment (and its correction parameters) is known. If an interpolated version of RATZ is used, the load of the algorithm increases linearly with the number of environments. Therefore, if the number of environments to interpolate is about twenty, the computational load of RATZ reaches the level of VTS-1. Even though VTS-0 and CDCN are

very similar algorithms, the use of log spectra as the feature representation of VTS-1, makes the algorithm more computationally intensive as the dimensionality of a log spectral vector is three times that of a cepstral vector.

It is interesting to note that the increase in computational load going from VTS-0 to VTS-1 is not that great.

8.9. Summary

In this chapter we have introduced the Vector Taylor Series (VTS) algorithms. We have shown how this approach allows us to estimate simultaneously the parameters defining the environment as well as the mean vectors and covariance matrices of log spectral distributions of noisy speech.

We also have shown the algorithm performance as a feature-vector compensation method. It compares favorably with other model-based algorithms such as CDCN as well as with algorithms that learn the effect of the environment on speech distributions by adaptation sets, such as RATZ. In fact the VTS algorithms perform as well as fully retrained systems up to a SNR of 10 dB.

We have explored the computational load of the VTS algorithms and shown to be higher than RATZ or CDCN.

Finally, we have studied the performance of all the algorithms proposed on this thesis on the 1994 Spoke 10 evaluation database. We have observed how all the algorithms proposed in this thesis provide significant improvements in recognition accuracy.

Chapter 9
Summary and Conclusions

This dissertation addresses the problem of environmental robustness using current speech recognition technology. Starting with a study of the effects of the environment on speech distributions we proposed a mathematical framework based on the EM algorithm for environment compensation. Two generic approaches have been proposed. The first approach uses data that is simultaneously recorded in the training and testing environments to learn how speech distributions are affected by the environment, and the second approach uses a Taylor series approximation to model the effects of the environment using an analytical vector function.

In this chapter we summarize our conclusions and findings based on our simulations with artificial data and experiments using real and artificially-contaminated data containing real speech. We review the major contributions of this work and present several suggestions for future work.

9.1. Summary of Results

We performed a series of simulations using artificial data to study in a controlled manner to study the effects of the environment on speech-like log spectral distributions. From these simulations we draw the following conclusions:

- the distributions of log spectra of speech are no longer Gaussians when submitted to additive noise and linear channel distortions.
- the means of the resulting noisy distributions are shifted
- the variances of the resulting noisy distributions are compressed

Based on this simulations we modeled the effects of the environment on Gaussian speech distributions as correction factors to be applied to the mean vectors and covariance matrices.

We developed two families of algorithms for data-driven environmental compensation. The first set of algorithms specified the use of correction factors to be applied to the incoming vector representation of noisy speech. The second family of algorithms provided corrections to the speech distributions that represent noisy speech in the HMM classifier.

The values of the correction factors were learned in three different ways:

- Using simultaneously-recorded clean and noisy speech databases ("stereo" databases) to learn correction factors directly from data (stereo RATZ and STAR).
- Iteratively learning the correction factors directly from noisy data alone (blind RATZ and STAR).
- Using only a small amount of noisy data (*i.e.*, the same sentence to be recognized) in combination with an analytical model of the environment to learn iteratively the parameters of the environment and the correction factors (VTS).

We also presented a unified framework for the RATZ and STAR algorithms showing that techniques that attempt to modify the incoming cepstra vectors and techniques that modify the parameters of the distributions of the HMMs can be described by the same theory.

Our experimental results demonstrate that all techniques proposed in this thesis produce significant improvements in recognition accuracy. In agreement with our predictions, the STAR techniques that modify the parameters of the distributions that internally represent speech outperform the RATZ techniques that modify the incoming cepstral vectors of noisy speech, given equivalent experimental conditions.

We have also shown how data-driven techniques seem to perform quite well even with only ten sentences of adaptation data. Contrary to previous results by Liu and Acero using the FCDCN algorithm [1], the use of an SNR-dependent structure did not help to reduce the error rate for our data-driven algorithms. Perhaps the use of a better mathematical model makes it unnecessary to partition distributions according to SNR, as had been done by the FCDCN algorithm. We have also shown how this mathematical structure allows for a natural extension to incorporate the concept of environment interpolation. Our results using environment interpolation show that not knowing the target environment might not be very detrimental.

When comparing the proposed algorithms with the performance of a fully retrained system we can conclude that they provide the same performance for SNRs as low as

- 15 dB for the RATZ family of algorithms

- 10 dB for the VTS family of algorithms
- 5 dB for the STAR family of algorithms

Finally, we have shown that an analytical characterization of the environment can enable meaningful compensation even with very limited amounts of data. Our results with the VTS algorithm are better than equivalent model-based algorithms such as CDCN [1].

9.2. Contributions

We summarize below the major contributions of this thesis.

- We used simulations with artificial data as a way of learning the effects of the environment on the distributions of the log spectra of clean speech. We believe that experimentation in controlled conditions is key to understanding the problem in hand and to obtaining important insights into the nature of the effects of the environment on the distributions of clean speech. Based on these insights we modeled the effects of the environment on the distributions of cepstra of clean speech as shifts of the mean vectors and compressions of the covariance matrices.

- We showed that compensation of the distributions of clean speech is the optimal solution for speech compensation as it minimizes the probability of error. Our results with STAR seem to support this assertion, since the STAR algorithm (which modifies internal distributions) outperforms all other algorithms that compensate incoming features. However, it is important to note that algorithms such as STAR are only approximations to the theoretical optimum as they make several assumptions that result in distributions that do not exactly minimize the probability of error.

- We presented a unified statistical formulation of the data-driven noise compensation problem. We argued that compensation methods that modify incoming data and compensation methods that modify the parameters of the distributions of the HMMs share the same common assumptions and differ primarily in how the correction factors are applied. This statistical for-

mulation has been naturally extended to the cases of SNR-dependent distributions and environment interpolation.

- We also presented a generic formulation of the problem of environment compensation when the environment is known through an analytical function. We introduced the use of vector Taylor series as a generic formulation for solving analytically for the correction factors to be applied to the distributions of the log spectra of clean speech. Our results are consistently better than the previous best-performing model-based method, CDCN.

9.3. Suggestions for Future Work

The majority of the current environmental compensation techniques are limited by the feature representations used to characterize the speech signal and by the model assumptions made in the classifier itself. As we have shown in our experimental results, even when the system is fully retrained at each of the SNRs used in the experiments, the performance of the system degrades.

The major reason for this degradation is that as the noise becomes more and more dominant the inherent differences between the classes become smaller, and the classification error increases. It is not possible to recover from the effects of additive noise at extremely low SNRs, so our only choice is to keep improving the performance of speech recognition systems at medium SNRs by changes to the feature representation and/or the classifier itself.

We would like to use feature representations that are inherently more resilient to the effects of the environment. We certainly would like to avoid the compressions produced by the use of a logarithmic function. Perhaps the use of mel-cepstral vectors should be reconsidered. In this sense techniques motivated by our knowledge of speech production and perception mechanisms have great potential, as confirmed by recent results using the Perceptual Linear Prediction technique (PLP) [18] in large vocabulary systems [56] [28].

With respect to the classifier, we would like to use more contextual and temporal information. Certainly the signal contains more structure and detailed information than is currently used in speech recognition systems. The use of feature trajectories [18] instead of feature vectors is a possibility. Also, the current recognition paradigm focuses more on the stationary parts of the signal

than on the transitional parts. Techniques that model the transitional part of the signal seem also promising [42].

In this thesis we have presented two data-driven approaches to environment compensation (RATZ and STAR) and a model-based approach (VTS). The data-driven approaches make minimal assumptions about the environment. The model-based approaches apply "structural" knowledge of the environmental degradation on the distributions of log spectrum of clean speech, thus reducing the need for large data sets. Perhaps there is scope for hybrid approaches between VTS and RATZ that assume some initial model of the environment and make a more efficient use of the data as these are available. In this sense MAP approaches are worth exploring.

Another possibility for improvement of the data-driven methods is to explore the ***a posteriori invariance*** assumption proposed in Section 5.2.2. We know from our results and other results [15] that this assumption is not accurate at lower SNRs. A study of ways to learn how the *a posteriori* probabilities are changed by the effects of the environment might be useful.

The VTS approach was only introduced in this thesis. We believe this compensation algorithm should be further explored in the following ways:

- Although we have presented a generic formulation for any kind of environment, the VTS algorithm should be modified for testing in different kinds of environments such as telephone channels, radio broadcast channels, etc. This would involve the exploration of different environmental functions.

- The VTS algorithms should be extended to compensate the HMM acoustical distributions as does STAR. We would expect further improvements when used in this manner.

- If perfect analytical knowledge of the environment is not available, perhaps methods that attempt to learn the environmental function and its derivatives should be explored.

- For a given order in the polynomial series expansion there might be more optimal power series than the Taylor series. The use of optimal Vector Power Series should be explored.

Another topic worth exploring would be to study the extent to which the ideas proposed in this

thesis can be applied to the area of speaker adaptation. Both problem domains share similar assumptions and can probably be unified.

Finally, to avoid the complexity that the use of Gaussian distributions introduces it is important to explore the use of other distributions that either result in equations with analytical solutions or that result in equations where simpler approximations can be used.

Appendix A
Comparing Data Compensation to Distribution Compensation

In this appendix, we provide a simple explanation of why data compensation methods such as RATZ, VTS-0th and VTS-1st provide worse performance than those that modify some of the parameters of the speech distributions such as STAR.

A.1. Basic Assumptions

We frame our problem as a simple binary detection problem. We assume that there are two classes (classes H_1 and H_2) with *a priori* probabilities $P(H_1)$ and $P(H_2)$ respectively. Our goal will be that of deriving a decision rule that maximizes a performance measure (likelihood) based on the probabilities of correct and incorrect decisions.

Both classes have Gaussian pdf's

$$p_{x|H_1}(x|H_1) = N_x(\mu_{x,H_1}, \sigma_{x,H_1}) \qquad p_{x|H_2}(x|H_2) = N_x(\mu_{x,H_2}, \sigma_{x,H_2}) \tag{A.1}$$

Our decision rule will have the form

$$\begin{aligned} &\text{choose } H_1 \text{ if} \quad (P[class{=}H_1|x] \geq P[class{=}H_2|x]) \\ &\text{choose } H_2 \text{ if} \quad (P[class{=}H_1|x] \leq P[class{=}H_2|x]) \end{aligned} \tag{A.2}$$

which is the maximum *a posteriori* (MAP) decision rule. From this decision rule we can divide the space into decision regions. For example, for the case where the two variances are equal and the two *a priori* probabilities are not equal the decision region is

$$x \geq \frac{\mu_{x,H_1} + \mu_{x,H_2}}{2} - \frac{\sigma^2_{x,H} \log\left(\frac{P(H_2)}{P(H_1)}\right)}{(\mu_{x,H_2} - \mu_{x,H_1})} = \gamma_x \tag{A.3}$$

Or presenting this graphically for a one dimensional random variable

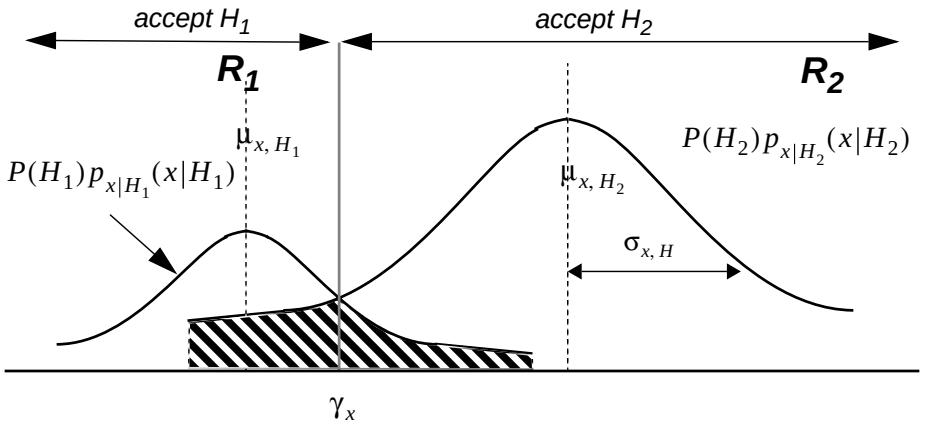

Figure A-1: The shadowed area represents the probability of making an error when classifying an incoming signal *x* as belonging to class H_1 or H_2. The line at the middle point divides the space into regions R_1 and R_2.

The shadowed area represents the probability of making a wrong decision when classifying an incoming signal *x* as belonging to classes H_1 or H_2. The space is divided into two regions, R_1 and R_2. Depending on what region the vector *x* is located we will classify the signal as belonging to class H_1 or H_2.

The probability of error can be computed as

$$Pe = P(H_1)\int_{\gamma_x}^{\infty} p_{x|H_1}(x|H_1)dx + P(H_2)\int_{-\infty}^{\gamma_x} p_{x|H_2}(x|H_2)dx \qquad \text{(A.4)}$$

where γ_x is the decision threshold.

In the two following sections we compare model compensation with data compensations for the simple case of additive noise, *i.e.*, the signal *x* has been contaminated by a noise *n* with pdf $N_n(\mu_n, \sigma_n)$, resulting in an observed noisy signal *y*. We also assume that the noise pdf is perfectly known.

A.2. Speech Distribution parameter Compensation

In this case a model compensation method would modify the parameters in the distributions, mean vectors and covariance matrices and perform classification with the noisy signal directly.

For the simple case of additive noise the new mean and covariance matrices can be easily calculated as

$$\begin{aligned} \mu_{y,H_1} &= \mu_{x,H_1} + \mu_n \qquad \Sigma_{y,H_1} = \Sigma_{x,H_1} + \Sigma_n \\ \mu_{y,H_2} &= \mu_{x,H_2} + \mu_n \qquad \Sigma_{y,H_2} = \Sigma_{x,H_2} + \Sigma_n \end{aligned} \tag{A.5}$$

For the previous case where the two covariance matrices where equal and the *a priori* probabilities were different, the new decision rule with the noisy data example will be

$$y \geq \frac{\mu_{y,H_1} + \mu_{y,H_2}}{2} - \frac{(\mu_{y,H_2} - \mu_{y,H_1})}{(\mu_{y,H_2} - \mu_{y,H_1})^t \Sigma_{y,H}^{-1} (\mu_{y,H_2} - \mu_{y,H_1})} \log\left(\frac{P(H_2)}{P(H_1)}\right) = \gamma_y \tag{A.6}$$

And the new probability of error will be

$$Pe = P(H_1) \int_{\gamma_y}^{\infty} p_{y|H_1}(y|H_1)dy + P(H_2) \int_{-\infty}^{\gamma_y} p_{y|H_2}(y|H_2)dy \tag{A.7}$$

For example, for the case where the two covariance matrices of the clean signal classes are equal and the *a priori* probabilities of both classes are also equal Figure A-2 shows how the error as represented by the shadowed area increases as the variance of the noise increases. The relative distance between the two classes remains constant as both are shifted the same distance, μ_n. However, the two distributions are wider as their width increases due to the effect of the noise covariance matrix Σ_n

A.3. Data Compensation

In this case a data compensation method would apply a correction term to the noisy data producing an estimated “clean” data vector. In most of the techniques proposed in this thesis the clean data are obtained using a Minimum Mean Squared Error Estimation technique (MMSE).

$$\hat{x} = E\{x|y\} = E\{y - n|y\} = y - \mu_n \tag{A.8}$$

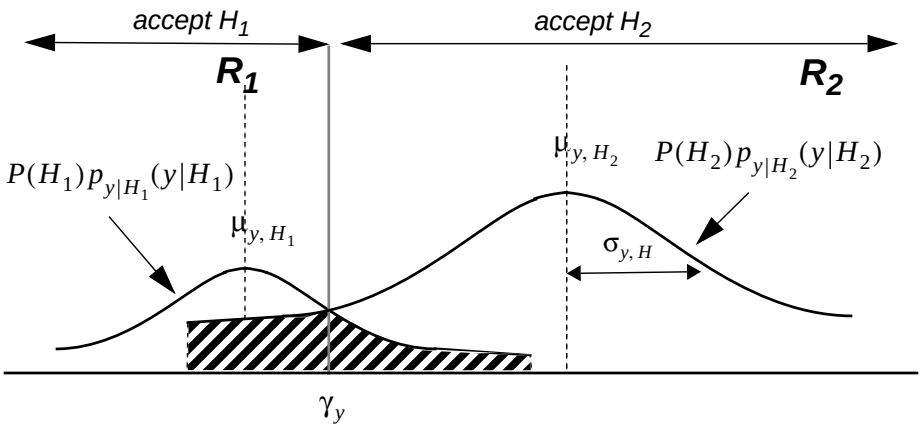

Figure A-2: The shadowed area represents the probability of making an error when classifying an incoming signal *y* as belonging to class H_1 or H_2. The line at the middle point divides the space into regions R_1 and R_2.

The pdf of the estimated clean data would be also Gaussian with these mean and variance parameters

$$\begin{aligned} \mu_{\hat{x},H_1} &= \mu_{y,H_1} - \mu_n = \mu_{x,H_1} \qquad \Sigma_{\hat{x},H_1} = \Sigma_{y,H_1} = \Sigma_{x,H_1} + \Sigma_n \\ \mu_{\hat{x},H_2} &= \mu_{y,H_2} - \mu_n = \mu_{x,H_2} \qquad \Sigma_{\hat{x},H_2} = \Sigma_{y,H_2} = \Sigma_{x,H_2} + \Sigma_n \end{aligned} \tag{A.9}$$

From these pdf's the decision threshold for our graphical example would be

$$\gamma_{\hat{x}} \geq \frac{\mu_{x,H_1} + \mu_{x,H_2}}{2} - \frac{(\mu_{x,H_2} - \mu_{x,H_1})}{(\mu_{x,H_2} - \mu_{x,H_1})^t (\Sigma_{x,H} + \Sigma_n)^{-1} (\mu_{x,H_2} - \mu_{x,H_1})} \log\left(\frac{P(H_2)}{P(H_1)}\right) \tag{A.10}$$

However, the classification would be done using the clean signal statistics which are different from the estimated clean statistics in the covariance terms. Using these clean signal pdf's would yield a decision threshold

$$\gamma_x \geq \frac{\mu_{x,H_1} + \mu_{x,H_2}}{2} - \frac{(\mu_{x,H_2} - \mu_{x,H_1})}{(\mu_{x,H_2} - \mu_{x,H_1})^t \Sigma_{x,H}^{-1} (\mu_{x,H_2} - \mu_{x,H_1})} \log\left(\frac{P(H_2)}{P(H_1)}\right) \tag{A.11}$$

Performing the classification with the clean signal distributions would introduce an additional error due to using the wrong decision threshold. In the next figure this additional error is marked as a small shadowed surface between the two thresholds γ_x and $\gamma_{\hat{x}}$.

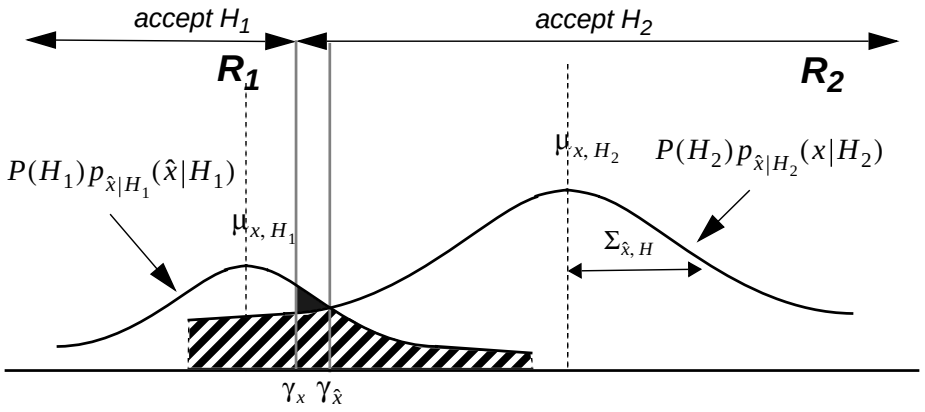

Figure A-3: The shadowed area represents the probability of making an error when classifying an incoming signal *y* as belonging to class H_1 or H_2. The area is split into two regions, the stripped one represent the normal error due to the classifier, the shadowed one (smaller and above the other one) represents the additional error due to improper modeling of the effect of the environment on the variances of the signal distributions.

In general data compensation methods incur greater errors compared to model compensation methods, due to improper modelling of the effects of the environment in the variances of the distributions.

Appendix B
Solutions for the SNR-RATZ Correction Factors

In this appendix we provide solutions for the SNR-RATZ correction factors $_i$, R_i, $\boldsymbol{r}_{i,j}$ and $\boldsymbol{R}_{i,j}$. The solutions depend on the availability of simultaneous clean and noisy (stereo) recordings. We first describe the generic solutions for the case in which only samples of noisy speech are available (the blind case) and we then describe how to particularize the previous solutions for the stereo case.

B.1. Non stereo based solutions

Given an observed noisy set of vectors $\boldsymbol{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_T\}$ of length T, and assuming that these vectors have been produced by the probability density function

$$p(\boldsymbol{y}) = \sum_{i=0}^{M} a_i \, N_{y_0|i}(\mu_{y_0,i}, \sigma^2_{y_0,i}) \sum_{j=0}^{N} a_{i,j} \, N_{\boldsymbol{y}_1|i,j}(\mu_{\boldsymbol{y}_1,i,j}, \Sigma_{\boldsymbol{y}_1,i,j}) \tag{B.1}$$

i.e., a double summation of Gaussians where each of them relates to the corresponding clean speech Gaussian according to

$$\begin{aligned} \mu_{\boldsymbol{y}_0,i} &= r_i + \mu_{x_0,i} & \sigma^2_{y_0,i} &= R_i + \sigma^2_{x_0,i} \\ {}_{\boldsymbol{y}_1,i,j} &= \boldsymbol{r}_{i,j} + \mu_{\boldsymbol{x}_1,i,j} & \Sigma_{\boldsymbol{y}_1,i,j} &= \boldsymbol{R}_{i,j} + \Sigma_{\boldsymbol{x}_1,i,} \end{aligned} \tag{B.2}$$

we can define a log likelihood function $L(\boldsymbol{Y})$ as

$$L(\boldsymbol{Y}) = \log\left(\prod_{t=1}^{T} p(y_t)\right) = \sum_{t=1}^{T} \log\left(\sum_{i=0}^{M} a_i \, N_{y_0|i}(\mu_{y_0,i}, \sigma^2_{y_0,i}) \sum_{j=0}^{N} a_{i,j} \, N_{\boldsymbol{y}_1|i,j}(\mu_{\boldsymbol{y}_1,i,j}, \Sigma_{\boldsymbol{y}_1,i,j})\right) \tag{B.3}$$

We can also express it in terms of the original clean speech parameters and the correction terms $_i$, R_i, $\boldsymbol{r}_{i,j}$ and $\boldsymbol{R}_{i,j}$

$$L(\boldsymbol{Y}) = \sum_{t=1}^{T} \log\left(\sum_{i=0}^{M} a_i \, N_{y_0|i}(r_i + \mu_{x_0,i}, R_i + \sigma^2_{x_0,i}) \sum_{j=0}^{N} a_{i,j} \, N_{\boldsymbol{y}_1|i,j}(\boldsymbol{r}_{i,j} + \mu_{\boldsymbol{x}_1,i,j}, \boldsymbol{R}_{i,j} + \Sigma_{\boldsymbol{x}_1,i,j})\right) \tag{B.4}$$

Our goal is to find all the terms $_i$, R_i, $\boldsymbol{r}_{i,j}$ and $\boldsymbol{R}_{i,j}$ that maximize the log likelihood. For this problem we can also use the EM algorithm defining a new auxiliary function $Q(\phi, \bar{\phi})$ as

$$Q(\phi, \bar{\phi}) = E[L(\boldsymbol{Y}, S|\bar{\phi}) \big| \boldsymbol{Y}, \phi] \tag{B.5}$$

where the $(\boldsymbol{Y}, S)$ pair represents the *complete* data, composed of the *observed* data $\boldsymbol{Y}$ (the noisy vectors) and the *unobserved* data S (it indicates what Gaussian produced an observed data vector). The ϕ symbol represents the correction terms $_i$, R_i, $\boldsymbol{r}_{i,j}$ and $\boldsymbol{R}_{i,j}$.

Equation (B.5) can we expanded as

$$Q(\phi, \bar{\phi}) = E[L(\boldsymbol{Y}, S|\bar{\phi})\big|\boldsymbol{Y}, \phi] = \sum_{t=1}^{T}\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{p(\boldsymbol{y}_t, i, j|\phi)}{p(\boldsymbol{y}_t|\phi)}\log(p(\boldsymbol{y}_t, i, j|\bar{\phi})) \tag{B.6}$$

hence,

$$\begin{aligned} Q(\phi, \bar{\phi}) = \sum_{t=1}^{T}\sum_{i=1}^{M}\sum_{j=1}^{N} P[i, j|\boldsymbol{y}_t, \phi]\{\log a_i - \frac{1}{2}\log(2\pi) - \frac{1}{2}\log\left|\bar{R}_i + \sigma^2_{x_0,i}\right| + \\ -\frac{1}{2}(y_{0,t} - (\bar{r}_i + \mu_{x_0,i}))^2/(\bar{R}_i + \sigma^2_{x_0,i}) + \log a_{i,j} - \frac{L-1}{2}\log(2\pi) - \frac{L-1}{2}\log\left|\bar{\boldsymbol{R}}_{i,j} + \Sigma_{\boldsymbol{x}_1,i,j}\right| + \\ -\frac{1}{2}(\boldsymbol{y}_{1,t} - (\bar{\boldsymbol{r}}_{i,j} + \mu_{\boldsymbol{x}_1,i,j}))^T(\bar{\boldsymbol{R}}_{i,j} + \Sigma_{\boldsymbol{x}_1,i,j})^{-1}(\boldsymbol{y}_{1,t} - (\bar{\boldsymbol{r}}_{i,j} + \mu_{\boldsymbol{x}_1,i,j}))\} \end{aligned} \tag{B.7}$$

where L is the dimensionality of the cepstrum vector. The expression can be further simplified to

$$\begin{aligned} Q(\phi, \bar{\phi}) = \textit{constants} + \sum_{t=1}^{T}\sum_{i=1}^{M}\sum_{j=1}^{N} P[i, j|\boldsymbol{y}_t, \phi]\{-\frac{1}{2}\log\left|\bar{R}_i + \sigma^2_{x_0,i}\right| + \\ -\frac{1}{2}(y_{0,t} - (\bar{r}_i + \mu_{x_0,i}))^2/(\bar{R}_i + \sigma^2_{x_0,i}) - \frac{L-1}{2}\log\left|\bar{\boldsymbol{R}}_{i,j} + \Sigma_{\boldsymbol{x}_1,i,j}\right| + \\ -\frac{1}{2}(\boldsymbol{y}_{1,t} - (\bar{\boldsymbol{r}}_{i,j} + \mu_{\boldsymbol{x}_1,i,j}))^T(\bar{\boldsymbol{R}}_{i,j} + \Sigma_{\boldsymbol{x}_1,i,j})^{-1}(\boldsymbol{y}_{1,t} - (\bar{\boldsymbol{r}}_{i,j} + \mu_{\boldsymbol{x}_1,i,j}))\} \end{aligned} \tag{B.8}$$

B.1.1 Solutions for the $\bar{r}_i$ and $\bar{R}_i$ parameters

To find the $\bar{r}_i$ and $\bar{R}_i$ parameters we simply take derivatives and equate to zero,

$$\nabla_{r_i} Q(\phi, \bar{\phi}) = \sum_{t=1}^{T}\sum_{j=1}^{N} P[i, j|\boldsymbol{y}_t, \phi](y_{0,t} - (\bar{r}_i + \mu_{x_0,i}))/(\bar{R}_i + \sigma^2_{x_0,i}) = 0 \tag{B.9}$$

$$\nabla_{\left(\bar{R}_i + \sigma^2_{x_0,i}\right)^{-1}} Q(\phi, \bar{\phi}) = -\sum_{t=1}^{T}\sum_{j=1}^{N} P[i, j|\boldsymbol{y}_t, \phi]\frac{1}{2}((\bar{R}_i + \sigma^2_{x_0,i}) - (y_{0,t} - (\bar{r}_i + \mu_{x_0,i}))^2) = 0 \tag{B.10}$$

hence,

$$\bar{r}_i = \frac{\sum_{t=1}^{T}\sum_{j=1}^{N} P[i, j|\boldsymbol{y}_t, \phi](y_{0,t} - \mu_{x_0,i})}{\sum_{t=1}^{T}\sum_{j=1}^{N} P[i, j|\boldsymbol{y}_t, \phi]} = \frac{\sum_{t=1}^{T} P[i|\boldsymbol{y}_t, \phi](y_{0,t} - \mu_{x_0,i})}{\sum_{t=1}^{T} P[i|\boldsymbol{y}_t, \phi]} \tag{B.11}$$

$$\begin{aligned}\bar{R}_i &= \frac{\sum_{t=1}^{T}\sum_{j=1}^{N} P[i, j|\boldsymbol{y}_t, \phi]((y_{0,t} - (\bar{r}_i + \mu_{x_0,i}))^2)}{\sum_{t=1}^{T}\sum_{j=1}^{N} P[i, j|\boldsymbol{y}_t, \phi]} - \sigma^2_{x_0,i} \\ &= \left(\sum_{t=1}^{T} P[i|\boldsymbol{y}_t, \phi]((y_{0,t} - (\bar{r}_i + \mu_{x_0,i}))^2)\right) \Big/ \left(\sum_{t=1}^{T} P[i|\boldsymbol{y}_t, \phi]\right) - \sigma^2_{x_0,i}\end{aligned} \tag{B.12}$$

B.1.2 Solutions for the $\bar{r}_{i,j}$ and $\bar{R}_{i,j}$ parameters

To find the $\bar{\boldsymbol{r}}_{i,j}$ and $\bar{\boldsymbol{R}}_{i,j}$ parameters we simply take derivatives and equate to zero,

$$\nabla_{\bar{\boldsymbol{r}}_{i,j}} Q(\phi, \bar{\phi}) = \sum_{t=1}^{T} P[i, j|\boldsymbol{y}_t, \phi](\bar{\boldsymbol{R}}_{i,j} + \Sigma_{\boldsymbol{x}_1,i,j})^{-1}(\boldsymbol{y}_{1,t} - (\bar{\boldsymbol{r}}_{i,j} + \mu_{\boldsymbol{x}_1,i,j})) = 0 \tag{B.13}$$

$$\begin{aligned}\nabla_{(\bar{\boldsymbol{R}}_{i,j} + \Sigma_{\boldsymbol{x}_1,i,j})^{-1}} Q(\phi, \bar{\phi}) = -\sum_{t=1}^{T} P[i, j|\boldsymbol{y}_t, \phi]\{&(\bar{\boldsymbol{R}}_{i,j} + \Sigma_{\boldsymbol{x}_1,i,j}) - \\ &(\boldsymbol{y}_{1,t} - (\bar{\boldsymbol{r}}_{i,j} + \mu_{\boldsymbol{x}_1,i,j}))(\boldsymbol{y}_{1,t} - (\bar{\boldsymbol{r}}_{i,j} + \mu_{\boldsymbol{x}_1,i,j}))^T\}\end{aligned} \tag{B.14}$$

resulting in the following solutions

$$\bar{\boldsymbol{r}}_{i,j} = \frac{\sum_{t=1}^{T} P[i, j|\boldsymbol{y}_t, \phi](\boldsymbol{y}_{1,t} - \mu_{\boldsymbol{x}_1,i,j})}{\sum_{\boldsymbol{t}=\boldsymbol{1}}^{\boldsymbol{T}} \boldsymbol{P}[i, j|\boldsymbol{y}_{\boldsymbol{t}}, \phi]} \tag{B.15}$$

$$\bar{\boldsymbol{R}}_{i,j} = \frac{\sum_{t=1}^{T} P[i, j|\boldsymbol{y}_t, \phi]((\boldsymbol{y}_{1,t} - (\bar{\boldsymbol{r}}_{i,j} + \mu_{\boldsymbol{x}_1,i,j}))(\boldsymbol{y}_{1,t} - (\bar{\boldsymbol{r}}_{i,j} + \mu_{\boldsymbol{x}_1,i,j}))^T)}{\sum_{t=1}^{T} P[i, j|\boldsymbol{y}_t, \phi]} - \Sigma_{\boldsymbol{x}_1,i,j} \tag{B.16}$$

These equations for the basis of an iterative algorithm. The EM algorithm guarantees that each iteration produces better estimates in a ML sense.

B.2. Stereo based solutions

When stereo adaptation data is available we can readily assume that the *a posteriori* probabilities $P[i, j|\boldsymbol{y}_t, \phi]$ can be directly estimated by $P[i, j|\boldsymbol{x}_t, \phi]$. We call this assumption ***a posteriori invariance***.

B.2.1 Solutions for the $\bar{r}_i$ and $\overline{R_i}$ parameters

The resulting estimates for the $\bar{r}_i$ and $\bar{R}_i$ parameters are

$$\bar{r}_i = \frac{\sum_{t=1}^{T} P[i|\boldsymbol{x}_t, \phi](y_{0,t} - \mu_{x_{0,i}})}{\sum_{t=1}^{T} P[i|\boldsymbol{x}_t, \phi]} \tag{B.17}$$

$$\bar{R}_i = \frac{\sum_{t=1}^{T} P[i|\boldsymbol{x}_t, \phi]((y_{0,t} - (r_i + \mu_{x_{0,i}}))^2)}{\sum_{t=1}^{T} P[i|\boldsymbol{x}_t, \phi]} - \sigma^2_{x_{0,i}} \tag{B.18}$$

Further simplification can be achieved by substituting the $(y_{0,t} - \mu_{x_{0,i}})$ term by $(y_{0,t} - x_{0,t})$ resulting in the formulas

$$\bar{r}_i = \frac{\sum_{t=1}^{T} P[i|\boldsymbol{x}_t, \phi](y_{0,t} - x_{0,t})}{\left(\sum_{t=1}^{T} P[i|\boldsymbol{x}_t, \phi]\right)} \tag{B.19}$$

$$\bar{R}_i = \frac{\sum_{t=1}^{T} P[i|\boldsymbol{x}_t, \phi]((y_{0,t} - (x_{0,t} + r_i))^2)}{\sum_{t=1}^{T} P[i|\boldsymbol{x}_t, \phi]} - \sigma^2_{x_{0,i}} \tag{B.20}$$

B.2.2 Solutions for the $\bar{r}_{i,j}$ and $\bar{\bar{r}}_{i,j}$ parameters

Using the ***a posteriori invariance*** property we obtain the following formulas

$$\bar{\boldsymbol{r}}_{i,j} = \frac{\sum_{t=1}^{T} P[i,j|\boldsymbol{x}_t,\phi](\boldsymbol{y}_{1,t} - \mu_{\boldsymbol{x}_1,i,j})}{\sum_{\boldsymbol{t}=\boldsymbol{1}}^{\boldsymbol{T}} \boldsymbol{P}[i,j|\boldsymbol{x}_{\boldsymbol{t}},\phi]} \tag{B.21}$$

$$\bar{\boldsymbol{R}}_{i,j} = \frac{\sum_{t=1}^{T} P[i,j|\boldsymbol{x}_t,\phi]((\boldsymbol{y}_{1,t} - (\bar{\boldsymbol{r}}_{i,j} + \mu_{\boldsymbol{x}_1,i,j}))(\boldsymbol{y}_{1,t} - (\bar{\boldsymbol{r}}_{i,j} + \mu_{\boldsymbol{x}_1,i,j}))^T)}{\sum_{t=1}^{T} P[i,j|\boldsymbol{x}_t,\phi]} - \Sigma_{\boldsymbol{x}_1,i,j} \tag{B.22}$$

Further simplification can be achieved by substituting the $(\boldsymbol{y}_{1,t} - \mu_{\boldsymbol{x}_1,i,j})$ term by $(\boldsymbol{y}_{1,t} - \boldsymbol{x}_{1,t})$ resulting in

$$\bar{\boldsymbol{r}}_{i,j} = \frac{\sum_{t=1}^{T} P[i,j|\boldsymbol{x}_t,\phi](\boldsymbol{y}_{1,t} - \boldsymbol{x}_{1,t})}{\sum_{\boldsymbol{t}=\boldsymbol{1}}^{\boldsymbol{T}} \boldsymbol{P}[i,j|\boldsymbol{x}_{\boldsymbol{t}},\phi]} \tag{B.23}$$

$$\bar{\boldsymbol{R}}_{i,j} = \frac{\sum_{t=1}^{T} P[i,j|\boldsymbol{x}_t,\phi]((\boldsymbol{y}_{1,t} - \boldsymbol{x}_{1,t} - \bar{\boldsymbol{r}}_{i,j})(\boldsymbol{y}_{1,t} - \boldsymbol{x}_{1,t} - \bar{\boldsymbol{r}}_{i,j})^T)}{\sum_{t=1}^{T} P[i,j|\boldsymbol{x}_t,\phi]} - \Sigma_{\boldsymbol{x}_1,i,j} \tag{B.24}$$

This concludes the derivation of the correction factors for the stereo and non stereo based SNR RATZ compensation algorithms.

Appendix C
Solutions for the Distribution Parameters for Clean Speech using SNR-RATZ

In this appendix we provide the EM solutions for the parameters of the SNR-RATZ clean speech cepstrum distributions.

It is important to notice also that our goal is only to show the use of the Expectation-Maximization (EM) algorithm in solving the Maximum Likelihood equations that will appear in this section. The reader is referred to [9][13] for a detailed explanation of the EM algorithm.

C.1. Basic Assumptions

The SNR-RATZ distribution for the clean speech cepstrum vectors has the following structure

$$p(\boldsymbol{x}|\phi) = \sum_{i=1}^{M} a_i \, N_{x_0|i}(\mu_{x_0,\,i}, \sigma^2_{x_0,\,i}) \sum_{j=1}^{N} a_{i,\,j} \, N_{\boldsymbol{x}_1|i,\,j}(\mu_{\boldsymbol{x}_1,\,i,\,j}, \Sigma_{\boldsymbol{x}_1,\,i,\,j}) \tag{C.1}$$

where we define ϕ as

$$\begin{aligned} \phi = \{ & a_1, .., a_M, \mu_{x_0,\,1}, ..., \mu_{x_0,\,M}, \sigma^2_{x_0,\,1}, ..., \sigma^2_{x_0,\,M}, \\ & a_{1,\,1}, ..., a_{M,\,N}, \mu_{\boldsymbol{x}_1,\,1,\,1}, ..., \mu_{\boldsymbol{x}_1,\,M,\,N}, \Sigma_{\boldsymbol{x}_1,\,1,\,1}, ..., \Sigma_{\boldsymbol{x}_1,\,M,\,N} \} \end{aligned} \tag{C.2}$$

i.e., the set of parameters that are unknown, and where the cepstrum vector $\boldsymbol{x} = [x_0 \; \boldsymbol{x}^T_1]^T$ is split in two parts, the energy component x_0, and $\boldsymbol{x}_1$, a vector composed of the x_1, x_2...,x_{L-1} components of the original cepstrum vector.

As in many other Maximum Likelihood problems given an ensemble of T clean vectors or observations $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_T\}$, we can define a log likelihood function,

$$L(\boldsymbol{X}|\phi) = \sum_{t=1}^{T} \log(p(\boldsymbol{x}_t|\phi)) \tag{C.3}$$

Our goal is to find the set of parameters ϕ that maximize the log likelihood of the observed data $\boldsymbol{X}$.

C.2. EM solutions for the $\bar{\mu}_{x_0,i}, \bar{\sigma}^2_{x_0,i}, \bar{\mu}_{\boldsymbol{x}_1,i,j}, \bar{\Sigma}_{\boldsymbol{x}_1,i,j}$ parameters

As it turns out there is no direct solution to this problem and indirect methods are necessary. The Expectation-Maximization (EM) algorithm is one of these methods. The EM algorithm defines a new auxiliary function $Q(\phi, \bar{\phi})$ as

$$Q(\phi, \bar{\phi}) = E[L(\boldsymbol{X}, S|\bar{\phi})\big|\boldsymbol{X}, \phi] \tag{C.4}$$

where the (X, S) pair represents the *complete* data, composed of the *observed* data $\boldsymbol{X}$ (the clean cepstrum vectors) and the *unobserved* data S (it indicates what two Gaussians produced an observed data vector).

The basis of the EM algorithm lies in the fact that given two sets of parameters ϕ and $\bar{\phi}$, if $Q(\phi, \bar{\phi}) \geq Q(\phi, \phi)$, then $L(\boldsymbol{X}, \bar{\phi}) \geq L(\boldsymbol{X}, \phi)$. In other words, maximizing $Q(\phi, \bar{\phi})$ with respect to the ϕ parameters is guaranteed to increase the likelihood $L(\boldsymbol{X}, \bar{\phi})$.

Since the unobserved data S are described by a discrete random variable (the mixture index in our case), Equation (C.4) can we expanded as

$$Q(\phi, \bar{\phi}) = E[L(\boldsymbol{X}, S|\bar{\phi})\big|\boldsymbol{X}, \phi] = \sum_{t=1}^{T}\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{p(\boldsymbol{x}_t, i, j|\phi)}{p(\boldsymbol{x}_t\big|\phi)}\log(p(\boldsymbol{x}_t, i, j|\bar{\phi})) \tag{C.5}$$

This can be further expanded to

$$\begin{aligned} Q(\phi, \bar{\phi}) = \sum_{t=1}^{T}\sum_{i=1}^{M}\sum_{j=1}^{N} P[i, j|\boldsymbol{x}_t, \phi]\{&\log a_i - \frac{1}{2}\log(2\pi) - \frac{1}{2}\log\left|\overline{\sigma^2_{x_0,i}}\right| + \\ -\frac{1}{2}(x_{0,t} - \bar{\mu}_{x_0,i})^2 / \bar{\sigma}^2_{x_0,i} &+ \log a_{i,j} - \frac{L-1}{2}\log(2\pi) - \frac{L-1}{2}\log\left|\bar{\Sigma}_{\boldsymbol{x}_1,i,j}\right| + \\ -\frac{1}{2}(\boldsymbol{x}_{1,t} - \bar{\mu}_{\boldsymbol{x}_1,i,j})^T &\bar{\Sigma}^{-1}_{\boldsymbol{x}_1,i,j}(\boldsymbol{x}_{1,t} - \bar{\mu}_{\boldsymbol{x}_1,i,j})\} \end{aligned} \tag{C.6}$$

where L is the dimensionality of the cepstrum vector. The expression can be further simplified to

$$Q(\phi, \bar{\phi}) = \text{constants} + \sum_{t=1}^{T} \sum_{i=1}^{M} \sum_{j=1}^{N} P[i, j|\boldsymbol{x}_t, \phi]\{\log a_i - \frac{1}{2}\log\left|\overline{\bar{\sigma}^2_{x_0, i}}\right| + \\ -\frac{1}{2}(x_{0,t} - \bar{\mu}_{x_0, i})^2 / \bar{\sigma}^2_{x_0, i} + \log a_{i,j} - \frac{L-1}{2}\log\left|\bar{\Sigma}_{\boldsymbol{x}_1, i, j}\right| + \\ -\frac{1}{2}(\boldsymbol{x}_{1,t} - \bar{\mu}_{\boldsymbol{x}_1, i, j})^T \bar{\Sigma}_{\boldsymbol{x}_1, i, j}^{-1}(\boldsymbol{x}_{1,t} - \bar{\mu}_{\boldsymbol{x}_1, i, j})\} \tag{C.7}$$

To find the ϕ parameters we simply take derivatives and equate to zero. The solutions for the $\bar{\mu}_{x_0, i}$ and $\bar{\sigma}^2_{x_0, i}$ parameters are

$$\nabla_{\bar{\mu}_{x_0, i}} Q(\phi, \bar{\phi}) = \sum_{t=1}^{T} \sum_{j=1}^{N} P[i, j|\boldsymbol{x}_t, \phi](x_{0,t} - \bar{\mu}_{x_0, i}) / \bar{\sigma}^2_{x_0, i} = 0 \tag{C.8}$$

$$\nabla_{\bar{\sigma}^{-2}_{x_0, i}} Q(\phi, \bar{\phi}) = -\sum_{t=1}^{T} \sum_{j=1}^{N} P[i, j|\boldsymbol{x}_t, \phi]\frac{1}{2}(\bar{\sigma}^2_{x_0, i} - (x_{0,t} - \bar{\mu}_{x_0, i})^2) = 0 \tag{C.9}$$

hence,

$$\bar{\mu}_{x_0, i} = \frac{\sum_{t=1}^{T} \sum_{j=1}^{N} P[i, j|\boldsymbol{x}_t, \phi] x_{0,t}}{\sum_{t=1}^{T} \sum_{j=1}^{N} P[i, j|x_t, \phi]} = \frac{\sum_{t=1}^{T} P[i|\boldsymbol{x}_t, \phi] x_{0,t}}{\sum_{t=1}^{T} P[i|\boldsymbol{x}_t, \phi]} \tag{C.10}$$

$$\bar{\sigma}^{-2}_{x_0, i} = \frac{\sum_{t=1}^{T} \sum_{j=1}^{N} P[i, j|\boldsymbol{x}_t, \phi](x_{0,t} - \bar{\mu}_{x_0, i})^2}{\sum_{t=1}^{T} \sum_{j=1}^{N} P[i, j|\boldsymbol{x}_t, \phi]} = \frac{\sum_{t=1}^{T} P[i|\boldsymbol{x}_t, \phi](x_{0,t} - \bar{\mu}_{x_0, i})^2}{\sum_{t=1}^{T} P[i|\boldsymbol{x}_t, \phi]} \tag{C.11}$$

Similarly, the solutions for the $\mu_{\boldsymbol{x}_1, i, j}$ and $\Sigma_{\boldsymbol{x}_1, i, j}$ parameter are

$$\nabla_{\bar{\mu}_{\boldsymbol{x}_1, i, j}} Q(\phi, \bar{\phi}) = \sum_{t=1}^{T} P[i, j|\boldsymbol{x}_t, \phi] \bar{\Sigma}_{\boldsymbol{x}_1, i, j}^{-1}(\boldsymbol{x}_{1,t} - \bar{\mu}_{\boldsymbol{x}_1, i, j}) = 0 \tag{C.12}$$

$$\nabla_{\bar{\Sigma}_{\boldsymbol{x}_1, i, j}^{-1}} Q(\phi, \bar{\phi}) = -\sum_{t=1}^{T} P[i, j|\boldsymbol{x}_t, \phi] \frac{1}{2} (\bar{\Sigma}_{\boldsymbol{x}_1, i, j} - (\boldsymbol{x}_{1,t} - \bar{\mu}_{\boldsymbol{x}_1, i, j})(\boldsymbol{x}_{1,t} - \bar{\mu}_{\boldsymbol{x}_1, i, j})^T) = 0 \tag{C.13}$$

hence,

$$\bar{\mu}_{\boldsymbol{x}_1, i, j} = \frac{\sum_{t=1}^{T} P[i, j|\boldsymbol{x}_t, \phi] \boldsymbol{x}_{1,t}}{\sum_{t=1}^{T} P[i, j|\boldsymbol{x}_t, \phi]} \tag{C.14}$$

$$\bar{\Sigma}_{\boldsymbol{x}_1, i, j} = \frac{\sum_{t=1}^{T} P[i, j|\boldsymbol{x}_t, \phi](\boldsymbol{x}_{1,t} - \bar{\mu}_{\boldsymbol{x}_1, i, j})(\boldsymbol{x}_{1,t} - \bar{\mu}_{\boldsymbol{x}_1, i, j})^T}{\sum_{t=1}^{T} P[i, j|\boldsymbol{x}_t, \phi]} \tag{C.15}$$

C.3. EM solutions for the $\bar{}_i, \bar{a}_{i,j}$ parameters

The solutions for the $\bar{a}_i$ and the $\bar{a}_{i,j}$ parameters cannot be obtained by simple derivatives. These parameters have the following additional constraints

$$\sum_{=1}^{M} \bar{a}_i = 1 \qquad \sum_{j=1}^{N} \bar{a}_{ij} = 1 \tag{C.16}$$

therefore to find the solutions for these parameters we need to use a Lagrange multiplier.

We can build two auxiliary functions f_{aux} and g_{aux} as

$$f_{aux} = \alpha \sum_{i=1}^{M} \bar{a}_i + Q(\phi, \bar{\phi}) \qquad g_{aux} = \beta \sum_{j=1}^{N} \bar{a}_{ij} + Q(\phi, \bar{\phi}) \tag{C.17}$$

where α and β are the Lagrange multipliers.

Taking the partial derivative of f_{aux} and g_{aux} with respect to $\bar{a}_i$ and $\bar{a}_{ij}$ respectively and equating to zero

$$\frac{\partial f_{aux}}{\partial \bar{a}_i} = \alpha + \sum_{t=1}^{T}\sum_{j=1}^{N} P[i, j|\boldsymbol{x}_t, \phi]\frac{1}{a_i} = 0 \quad \Rightarrow \quad \alpha\bar{a}_i + \sum_{t=1}^{T}\sum_{j=1}^{N} P[i, j|\boldsymbol{x}_t, \phi] = 0 \tag{C.18}$$

$$\frac{\partial g_{aux}}{\partial \bar{a}_{ij}} = \beta + \sum_{t=1}^{T} P[i, j|\boldsymbol{x}_t, \phi]\frac{1}{\bar{a}_{ij}} = 0 \quad \Rightarrow \quad \beta\bar{a}_{ij} + \sum_{t=1}^{T} P[i, j|\boldsymbol{x}_t, \phi] = 0 \tag{C.19}$$

summing over i and j respectively

$$\alpha\sum_{i=1}^{M}\bar{a}_i + \sum_{t=1}^{T}\sum_{j=1}^{N}\sum_{i=1}^{M} P[i, j|\boldsymbol{x}_t, \phi] = 0 \quad \Rightarrow \quad \alpha = -\sum_{t=1}^{T}\sum_{j=1}^{N}\sum_{i=1}^{M} P[i, j|\boldsymbol{x}_t, \phi] = -T \tag{C.20}$$

$$\beta\sum_{j=1}^{N}\bar{a}_{ij} + \sum_{t=1}^{T}\sum_{j=1}^{N} P[i, j|\boldsymbol{x}_t, \phi] = 0 \quad \Rightarrow \quad \beta = -\sum_{t=1}^{T}\sum_{j=1}^{N} P[i, j|\boldsymbol{x}_t, \phi] \tag{C.21}$$

entering the value of α in Equation (C.16) and the value of β in Equation (C.17)

$$-T\bar{a}_i + \sum_{t=1}^{T}\sum_{j=1}^{N} P[i, j|\boldsymbol{x}_t, \phi] = 0 \quad \Rightarrow \quad \bar{a}_i = \frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{N} P[i, j|\boldsymbol{x}_t, \phi] = \frac{1}{T}\sum_{t=1}^{T} P[i|\boldsymbol{x}_t, \phi] \tag{C.22}$$

$$-\bar{a}_{ij}\sum_{t=1}^{T}\sum_{j=1}^{N} P[i, j|\boldsymbol{x}_t, \phi] + \sum_{t=1}^{T} P[i, j|\boldsymbol{x}_t, \phi] = 0 \quad \Rightarrow \quad \bar{a}_{ij} = \frac{\sum_{t=1}^{T} P[i, j|\boldsymbol{x}_t, \phi]}{\sum_{t=1}^{T}\sum_{j=1}^{N} P[i, j|\boldsymbol{x}_t, \phi]} \tag{C.23}$$

This concludes the derivation of the EM solutions for the parameters of the SNR-RATZ clean speech cepstrum distributions.

Appendix D
EM Solutions for the *n* and *q* Parameters for the VTS Algorithm

In this appendix, we provide the EM solutions for the environmental parameters for the case of an additive noise and linear channel environment. We also provide solutions for the Vector Taylor series or order zero and one.

D.1. Solutions for the vector Taylor series of order one

As we mentioned in Section 8.4. for the case of a first order Taylor series approximation the mean vector and covariance matrix of each of the individual mixtures of the Gaussian mixture can be expressed as

$$\begin{aligned}\mu_{k,\,\boldsymbol{y}} \cong \mu_{k,\,\boldsymbol{x}} + (\boldsymbol{I} + \nabla_{\boldsymbol{h}}\, \boldsymbol{g}(\mu_{k,\,\boldsymbol{x}}, \boldsymbol{n}_0, \boldsymbol{h}_0))\ \boldsymbol{h} + \nabla_{\boldsymbol{n}}\, \boldsymbol{g}(\mu_{k,\,\boldsymbol{x}}, \boldsymbol{n}_0, \boldsymbol{h}_0)\ \boldsymbol{n} + \\ \boldsymbol{g}(\mu_{k,\,\boldsymbol{x}}, \boldsymbol{n}_0, \boldsymbol{h}_0) - \nabla_{\boldsymbol{h}}\, \boldsymbol{g}(\mu_{k,\,\boldsymbol{x}}, \boldsymbol{n}_0, \boldsymbol{h}_0)\ \boldsymbol{h}_0 - \nabla_{\boldsymbol{n}}\, \boldsymbol{g}(\mu_{k,\,\boldsymbol{x}}, \boldsymbol{n}_0, \boldsymbol{h}_0)\ \boldsymbol{n}_0\end{aligned} \tag{D.1}$$

$$\Sigma_{k,\,\boldsymbol{y}} \cong (\boldsymbol{I} + \nabla_{\boldsymbol{h}}\, \boldsymbol{g}(\mu_{k,\,\boldsymbol{x}}, \boldsymbol{n}_0, \boldsymbol{h}_0))\Sigma_{k,\,\boldsymbol{x}}(\boldsymbol{I} + \nabla_{\boldsymbol{h}}\, \boldsymbol{g}(\mu_{k,\,\boldsymbol{x}}, \boldsymbol{n}_0, \boldsymbol{h}_0))^T \tag{D.2}$$

In this case, the expression for the mean of the noisy speech log-spectrum distribution is a linear function of the unknown variables $\boldsymbol{n}$ and $\boldsymbol{h}$ can be rewritten as

$$\begin{gathered}\mu_{k,\,\boldsymbol{y}} \cong \boldsymbol{a}_k + \boldsymbol{B}_k\, \mathbf{h} + \boldsymbol{C}_k\, \boldsymbol{n} \\ \boldsymbol{a}_k = \mu_{k,\,\boldsymbol{x}} + \boldsymbol{g}(\mu_{k,\,\boldsymbol{x}}, \boldsymbol{n}_0, \boldsymbol{h}_0) - \nabla_{\boldsymbol{h}}\, \boldsymbol{g}(\mu_{k,\,\boldsymbol{x}}, \boldsymbol{n}_0, \boldsymbol{h}_0)\ \boldsymbol{h}_0 - \nabla_{\boldsymbol{n}}\, \boldsymbol{g}(\mu_{k,\,\boldsymbol{x}}, \boldsymbol{n}_0, \boldsymbol{h}_0)\ \boldsymbol{n}_0 \\ \boldsymbol{B}_k = \boldsymbol{I} + \nabla_{\boldsymbol{h}}\, \boldsymbol{g}(\mu_{k,\,\boldsymbol{x}}, \boldsymbol{n}_0, \boldsymbol{h}_0) \qquad \boldsymbol{C}_k = \nabla_{\boldsymbol{n}}\, \boldsymbol{g}(\mu_{k,\,\boldsymbol{x}}, \boldsymbol{n}_0, \boldsymbol{h}_0)\end{gathered} \tag{D.3}$$

while the expression for the covariance matrix is dependent only of the initial values $\mu_{\boldsymbol{x}}$, $\boldsymbol{n}_0$ and $\boldsymbol{h}_0$ which are known. Therefore, given the observed noisy data we can define a likelihood function

$$L(\boldsymbol{Y} = \{\boldsymbol{y}_0, \boldsymbol{y}_1, ..., \boldsymbol{y}_{S-1}\}) = \sum_{t=0}^{S-1} \log(p(\boldsymbol{y}_t | \boldsymbol{h}, \boldsymbol{n})) \tag{D.4}$$

where the only unknowns are the $\boldsymbol{n}$ and $\boldsymbol{h}$ variables. To find these unknowns we can use a traditional iterative EM approach.

We define an auxiliary function $Q(\phi, \bar{\phi})$ as

$$Q(\phi, \bar{\phi}) = E[L(\boldsymbol{Y}, S|\bar{\phi})\big|\boldsymbol{Y}, \phi] \tag{D.5}$$

where the $(\boldsymbol{Y}, S)$ pair represents the *complete* data, composed of the *observed* data $\boldsymbol{Y}$ (the noisy vectors) and the *unobserved* data S (it indicates what Gaussian produced an observed data vector). The ϕ symbol represents the environmental parameters $\boldsymbol{n}$ and $\boldsymbol{h}$.

Equation (D.5) can we expanded as

$$Q(\phi, \bar{\phi}) = E[L(\boldsymbol{Y}, S|\bar{\phi})\big|\boldsymbol{Y}, \phi] = \sum_{t=0}^{S-1}\sum_{k=0}^{K-1}\frac{p(\boldsymbol{y}_t, k|\phi)}{p(\boldsymbol{y}_t|\phi)}\log(p(\boldsymbol{y}_t, k|\bar{\phi})) \tag{D.6}$$

hence,

$$\begin{aligned} Q(\phi, \bar{\phi}) = E[L(\boldsymbol{Y}, S|\bar{\phi})\big|\boldsymbol{Y}, \phi] = \sum_{t=0}^{S-1}\sum_{k=0}^{K-1} P[k|\boldsymbol{y}_t, \phi]\{\log p_k + \frac{L}{2}\log 2\pi + \\ -\frac{L}{2}\log|\Sigma_{k,\boldsymbol{y}}| - \frac{1}{2}(\boldsymbol{y}_t - (\boldsymbol{a}_k + \boldsymbol{B}_k\,\bar{\mathbf{h}} + \boldsymbol{C}_k\,\bar{\boldsymbol{n}}))^T\Sigma_{k,\boldsymbol{y}}^{-1}(\boldsymbol{y}_t - (\boldsymbol{a}_k + \boldsymbol{B}_k\,\bar{\mathbf{h}} + \boldsymbol{C}_k\,\bar{\boldsymbol{n}}))\} \end{aligned} \tag{D.7}$$

where L is the dimension of the log-spectrum vector and the terms $\boldsymbol{a}_k$, $\boldsymbol{B}_k$ and $\boldsymbol{C}_k$ are the terms described in Equation (D.3) particularize for each of the individual Gaussians of $p(\boldsymbol{x}_t)$. The expression can be further simplified to

$$\begin{aligned} Q(\phi, \bar{\phi}) = constants - \frac{1}{2}\sum_{t=0}^{S-1}\sum_{k=0}^{K-1} P[k|\boldsymbol{y}_t, \phi] \\ (\boldsymbol{y}_t - (\boldsymbol{a}_k + \boldsymbol{B}_k\,\bar{\boldsymbol{h}} + \boldsymbol{C}_k\,\bar{\boldsymbol{n}}))^T\Sigma_{k,\boldsymbol{y}}^{-1}(\boldsymbol{y}_t - (\boldsymbol{a}_k + \boldsymbol{B}_k\,\bar{\boldsymbol{h}} + \boldsymbol{C}_k\,\bar{\boldsymbol{n}})) \end{aligned} \tag{D.8}$$

To find the $\boldsymbol{n}$ and $\boldsymbol{h}$ parameters we simply take derivatives and equate to zero,

$$\nabla_{\bar{\boldsymbol{n}}}\, Q(\phi, \bar{\phi}) = \sum_{t=0}^{S-1}\sum_{k=0}^{K-1} P[k|\boldsymbol{y}_t, \phi]\{\boldsymbol{C}_k^T\Sigma_{k,\boldsymbol{y}}^{-1}(\boldsymbol{y}_t - (\boldsymbol{a}_k + \boldsymbol{B}_k\,\bar{\boldsymbol{h}} + \boldsymbol{C}_k\,\bar{\boldsymbol{n}})) = 0 \tag{D.9}$$

$$\nabla_{\bar{\boldsymbol{h}}} Q(\phi, \bar{\phi}) = \sum_{t=0}^{S-1} \sum_{k=0}^{K-1} P[k|\boldsymbol{y}_t, \phi]\{\boldsymbol{B}_k^T \Sigma_{k,\boldsymbol{y}}^{-1}(\boldsymbol{y}_t - (\boldsymbol{a}_k + \boldsymbol{B}_k \bar{\boldsymbol{h}} + \boldsymbol{C}_k \bar{\boldsymbol{n}})) = 0 \tag{D.10}$$

The above two vector equations can be simplified to

$$\begin{aligned} \boldsymbol{d} - \boldsymbol{E}\,\bar{\boldsymbol{h}} - \boldsymbol{F}\,\bar{\boldsymbol{n}} &= 0 \\ \boldsymbol{g} - \boldsymbol{H}\,\bar{\boldsymbol{h}} - \boldsymbol{J}\,\bar{\boldsymbol{n}} &= 0 \end{aligned} \tag{D.11}$$

where each of the *d, E, F, g, H,* and *J* terms is expanded as

$$\begin{aligned} \boldsymbol{d} &= \sum_{t=0}^{S-1} \sum_{k=0}^{K-1} P[k|\boldsymbol{y}_t, \phi]\{\boldsymbol{C}_k^T \Sigma_{k,\boldsymbol{y}}^{-1}(\boldsymbol{y}_t - \boldsymbol{a}_k) & \boldsymbol{g} &= \sum_{t=0}^{S-1} \sum_{k=0}^{K-1} P[k|\boldsymbol{y}_t, \phi]\{\boldsymbol{B}_k^T \Sigma_{k,\boldsymbol{y}}^{-1}(\boldsymbol{y}_t - \boldsymbol{a}_k) \\ \boldsymbol{E} &= \sum_{t=0}^{S-1} \sum_{k=0}^{K-1} P[k|\boldsymbol{y}_t, \phi]\{\boldsymbol{C}_k^T \Sigma_{k,\boldsymbol{y}}^{-1} \boldsymbol{B}_k & \boldsymbol{H} &= \sum_{t=0}^{S-1} \sum_{k=0}^{K-1} P[k|\boldsymbol{y}_t, \phi]\{\boldsymbol{B}_k^T \Sigma_{k,\boldsymbol{y}}^{-1} \boldsymbol{B}_k \\ \boldsymbol{F} &= \sum_{t=0}^{S-1} \sum_{k=0}^{K-1} P[k|\boldsymbol{y}_t, \phi]\{\boldsymbol{C}_k^T \Sigma_{k,\boldsymbol{y}}^{-1} \boldsymbol{C}_k & \boldsymbol{J} &= \sum_{t=0}^{S-1} \sum_{k=0}^{K-1} P[k|\boldsymbol{y}_t, \phi]\{\boldsymbol{B}_k^T \Sigma_{k,\boldsymbol{y}}^{-1} \boldsymbol{C}_k \end{aligned} \tag{D.12}$$

Equation (D.5) can be rewritten as

$$\begin{bmatrix} \boldsymbol{d} \\ \boldsymbol{g} \end{bmatrix} = \begin{bmatrix} \boldsymbol{E} & \boldsymbol{F} \\ \boldsymbol{H} & \boldsymbol{J} \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{h}} \\ \bar{\boldsymbol{n}} \end{bmatrix} \tag{D.13}$$

where we have created an expanded matrix composed of the *E, F, H,* and *J* matrices. We also have created an expanded vector composed of the concatenation of the *d* and *g* vectors. The above linear system yields the following solutions

$$\begin{aligned} \bar{\boldsymbol{h}} &= (\boldsymbol{H} - \boldsymbol{J}\,\boldsymbol{F}^{-1}\,\boldsymbol{E})^{-1}(\boldsymbol{g} - \boldsymbol{J}\,\boldsymbol{F}^{-1}\,\boldsymbol{d}) \\ \bar{\boldsymbol{n}} &= (\boldsymbol{J} - \boldsymbol{H}\,\boldsymbol{F}^{-1}\,\boldsymbol{E})^{-1}(\boldsymbol{g} - \boldsymbol{H}\,\boldsymbol{F}^{-1}\,\boldsymbol{d}) \end{aligned} \tag{D.14}$$

There is no solutions strictly speaking if the extended matrix $\begin{bmatrix} \boldsymbol{E} & \boldsymbol{F} \\ \boldsymbol{H} & \boldsymbol{J} \end{bmatrix}$ is not invertible. We might be faced with a situation where there are no solution or there are an infinite number of solutions. This occurs when the solutions obtained for the $\bar{\boldsymbol{h}}$ and $\bar{\boldsymbol{n}}$ vectors converge to ± infinity. To avoid

this behavior we impose the empirical constraint in the space of solutions that any log-spectrum component $\bar{h}_i$ or $\bar{n}_i$ can only exist in the range $h_{i,\,max} \le \bar{h}_i \le h_{i,\,min}$ or $n_{i,\,max} \le \bar{n}_i \le n_{i,\,min}$. The upper and lower boundaries are set experimentally.

Once the solutions for $\bar{\boldsymbol{h}}$ and $\bar{\boldsymbol{n}}$ are found we can substitute them for $\boldsymbol{h}$ and $\boldsymbol{n}$ and iterate the procedure until convergence is obtained.

D.2. Solutions for the vector Taylor series of order one

For the case of a zeroth-order Taylor series approximation the mean vector and covariance matrix of each of the individual mixtures of the Gaussian mixture can be expressed as

$$\mu_{k,\,\boldsymbol{y}} \cong \mu_{k,\,\boldsymbol{x}} + \boldsymbol{h} + \boldsymbol{g}(\mu_{k,\,\boldsymbol{x}}, \boldsymbol{n}_0, \boldsymbol{h}_0) \tag{D.15}$$

$$\Sigma_{k,\,\boldsymbol{y}} \cong \Sigma_{k,\,\boldsymbol{x}} \tag{D.16}$$

In this case, the expression for the mean of the log-spectral distribution of the noisy speech is a linear function of the unknown variable $\boldsymbol{h}$ and can be rewritten as

$$\begin{aligned} \mu_{k,\,\boldsymbol{y}} &\cong \boldsymbol{a}_k + \boldsymbol{h} \\ \boldsymbol{a}_k &= \mu_{k,\,\boldsymbol{x}} + \boldsymbol{g}(\mu_{k,\,\boldsymbol{x}}, \boldsymbol{n}_0, \boldsymbol{h}_0) \end{aligned} \tag{D.17}$$

Therefore, given the observed noisy data we can define a likelihood function

$$L(\boldsymbol{Y} = \{\boldsymbol{y}_0, \boldsymbol{y}_1, ..., \boldsymbol{y}_{S-1}\}) = \sum_{t=0}^{S-1} \log(p(\boldsymbol{y}_t | \boldsymbol{h})) \tag{D.18}$$

where the only unknown is the $\boldsymbol{h}$ variable. To find $\boldsymbol{h}$ we can use again a traditional iterative EM approach. We define an auxiliary function $Q(\phi, \bar{\phi})$ as

$$Q(\phi, \bar{\phi}) = E[L(\boldsymbol{Y}, S|\bar{\phi}) \big| \boldsymbol{Y}, \phi] \tag{D.19}$$

that can we expanded as

$$\begin{aligned} Q(\phi, \bar{\phi}) = E[L(\boldsymbol{Y}, S|\bar{\phi}) \big| \boldsymbol{Y}, \phi] = \sum_{t=0}^{S-1} \sum_{k=0}^{K-1} P[k|\boldsymbol{y}_t, \phi]\{\log p_k + \frac{L}{2}\log 2\pi + \\ -\frac{L}{2}\log|\Sigma_{k,\,y}| - \frac{1}{2}(\boldsymbol{y}_t - (\boldsymbol{a}_k + \bar{\boldsymbol{h}}))^T \Sigma_{k,\,\boldsymbol{y}}^{-1}(\boldsymbol{y}_t - (\boldsymbol{a}_k + \bar{\boldsymbol{h}}))\} \end{aligned} \tag{D.20}$$

where L is the dimension of the log-spectrum vector and the term $\boldsymbol{a}_k$ is the term described in Equation (D.3). The expression can be further simplified to

$$Q(\phi, \bar{\phi}) = constants - \frac{1}{2}\sum_{t=0}^{S-1}\sum_{k=0}^{K-1} P[k|\boldsymbol{y}_t, \phi]\ ((\boldsymbol{y}_t - (\boldsymbol{a}_k + \bar{\boldsymbol{h}}))^T \Sigma_{k,\boldsymbol{x}}^{-1}(\boldsymbol{y}_t - (\boldsymbol{a}_k + \bar{\boldsymbol{h}}))) \tag{D.21}$$

To find the $\boldsymbol{h}$ parameter we simply take derivative and set equal to zero,

$$\nabla_{\bar{\boldsymbol{h}}} Q(\phi, \bar{\phi}) = \sum_{t=0}^{S-1}\sum_{k=0}^{K-1} P[k|\boldsymbol{y}_t, \phi]\{\Sigma_{k,\boldsymbol{x}}^{-1}(\boldsymbol{y}_t - (\boldsymbol{a}_k + \bar{\boldsymbol{h}})) = 0 \tag{D.22}$$

The above vector equation yields the following solution for $\boldsymbol{h}$

$$\bar{\boldsymbol{h}} = \left(\sum_{t=0}^{S-1}\sum_{k=0}^{K-1} P[k|\boldsymbol{y}_t, \phi]\{\Sigma_{k,\boldsymbol{x}}^{-1}\right)^{-1}\left(\sum_{t=0}^{S-1}\sum_{k=0}^{K-1} P[k|\boldsymbol{y}_t, \phi]\{\Sigma_{k,\boldsymbol{x}}^{-1}(\boldsymbol{y}_t - \boldsymbol{a}_k)\right) \tag{D.23}$$

Once the $\bar{\boldsymbol{h}}$ variable is found we can substitute it for $\boldsymbol{h}$ and iterate the procedure until convergence is obtained.

As we can see a zeroth-order Taylor approximation does not provide solution for the parameter *n*. To remedy this problem we can redefine the relationship between noisy speech and clean speech as

$$\boldsymbol{y} \cong \boldsymbol{n} + \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{n}, \boldsymbol{h}) \tag{D.24}$$

With this new environmental equation we can define a zeroth-order Taylor expansion

$$\boldsymbol{y} \cong \boldsymbol{n} + \boldsymbol{f}(\boldsymbol{x}_0, \boldsymbol{n}_0, \boldsymbol{h}_0) \tag{D.25}$$

Taking expected values in both sides of the equation yields the expression

$$\mu_{\boldsymbol{y}} \cong \boldsymbol{n} + \boldsymbol{f}(\mu_{\boldsymbol{x}}, \boldsymbol{n}_0, \boldsymbol{h}_0) \tag{D.26}$$

We now can use this expression to define a likelihood function that can be maximized via the EM algorithm resulting in the following iterative solution

$$\bar{n} = \left(\sum_{t=0}^{S-1}\sum_{k=0}^{K-1} P[k|\boldsymbol{y}_t, \phi]\{\Sigma_{k,\boldsymbol{x}}^{-1}\right)^{-1}\left(\sum_{t=0}^{S-1}\sum_{k=0}^{K-1} P[k|\boldsymbol{y}_t, \phi]\{\Sigma_{k,\boldsymbol{x}}^{-1}(\boldsymbol{y}_t - \boldsymbol{f}(\mu_{k,\boldsymbol{x}}, \boldsymbol{n}_0, \boldsymbol{h}_0))\right) \tag{D.27}$$

Although this is not a rigorous solution it works reasonably well in practice and solves the problem of not having a proper way of estimating the noise vector for zeroth-order Taylor approximations.

REFERENCES

[1] A. Acero, “Acoustical and Environmental Robustness in Automatic Speech Recognition”, Ph.D. Thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, Sept. 1990.

[2] K. Aikawa, H. Singer, H. Kawahara, Y. Tohkura, “A dynamic cepstrum incorporating time-frequency masking and its application to continuous speech recognition”, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, May, 1993

[3] A. Akivis & V. V. Goldberg, An introduction to Linear Algebra and Tensors. Dover Publications, 1990.

[4] A. Anastasakos, F. Kubala, J. Makhoul and R. Schwartz, “Adaptation to new microphones using Tied-Mixtures Normalization”. *Proceedings of the Spoken Language Technology Workshop*, March, 1994.

[5] F. Alleva, X. Huang, and M. Hwang, “An Improved Search Algorithm for Continuous Speech Recognition”, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. II 307-310, May, 1993.

[6] J. Baker, “Stochastic Modeling as a Means of Automatic Speech Recognition”, Ph.D. Thesis, Computer Science Department, Carnegie Mellon University, April 1975.

[7] R. Bakis, “Continuous Speech Recognition via Centisecond Acoustic States”, *91st Meeting of the Acoustical Society of America*, April, 1976.

[8] L. Bahl, F. Jelinek, and R. Mercer, “A Maximum Likelihood Approach to Continuous Speech Recognition” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(2):179-190, March 1983.

[9] L. Baum, “An Inequality and Associated Maximization Technique in Statistical Estimation of Probabilistic Functions of Markov Processes”, *Inequalities* 3:1-8, 1972.

[10] Y. Chow, M. Dunham, O. Kimball, M. Krasner, F. Kubala, J. Markoul, S. Roucos, and R. Schwartz, “BYBLOS: The BBN Continuous Speech Recognition System”, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp.89-92, April, 1987.

[11] S. Davis, and P. Mermelstein, “Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences”, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-28, No. 4, pp. 357-366, August 1980.

[12] R. O. Duda & P. E. Hart, Pattern Classification and Scene Analysis, John Wiley and Sons, 1973.

[13] A. P. Depster, N. M. Laird & D. B. Rubin, “Maximum Likelihood from incomplete data via the EM algorithm (with discussion)”, Journal of the Royal Stat. Society, Series B, Vol.39, pp.1-38.

[14] J. Flanagan, J. Johnston, R. Zahn, and G. Elko, “Computer-steered Microphone Arrays for Sound Transduction in Large Rooms”, *the Journal of Acoustical Society of America*, Vol. 78, pp. 1508-1518, Nov. 1985.

[15] M. F. Gales, “Model-Based Techniques for Noise Robust Speech Recognition”. Ph.D. Thesis, Engineering Department, Cambridge University, Sept. 1995.

[16] O. Ghitza, “Auditory Nerve Representation as a Front-End for Speech Recognition in a Noisy En-

vironment", *Computer Speech and Language*, Vol. 1, pp. 109-130, 1986.

[17] L. Gillick, and S. Cox, "Some Statistical Issues in the Comparison of Speech Recognition Algorithms", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 532-535, May 1989.

[18] Y. Gong and J. P. Haton, "Stochastic trajectory modeling for speech recognition", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. I-57 - 60, April, 1994.

[19] H. Hermansky, "Perceptual Linear Prediction (PLP) Analysis for Speech". *J. Acoust. Soc. Amer.* Vol. 87, pp. 1738-1752.

[20] H. Hermansky, N. Morgan, and H. Hirsch, "Recognition of Speech in Additive and Convolutional Noise Based on RASTA Spectral Processing", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. II-83 - 86, April, 1993.

[21] Huang, Ariki & Jack, Hidden Markov Models for Speech Recognition, Edinburgh University Press, 1990.

[22] X. Huang, F. Alleva, H. Hon, M. Hwang, K. Lee, R. Rosenfeld, "The SPHINX-II Speech Recognition System: An Overview", *Computer Speech and Language*, vol. 2, pp. 137-148, 1993.

[23] M. Hwang, "Subphonetic Acoustic Modeling for Speaker-Independent Continuous Speech Recognition", Ph.D. Thesis, School of Computer Science, Carnegie Mellon University, Dec. 1993.

[24] M. Hwang, and X. Huang, "Shared-Distribution Hidden Markov Models for Speech Recognition", *IEEE Transactions on Speech and Audio Processing*, vol. 1, pp. 414-420, 1993.

[25] F. Jelinek, "Continuous Speech Recognition by Statistical Methods", *Proceedings of the IEEE* 64(4):532-556, April 1976.

[26] B. Juang, "Speech Recognition in Adverse Environments", *Computer Speech and Language*, Vol. 5, pp. 275-294, 1991.

[27] B. Juang, and L. Rabiner, "Mixture Autoregressive Hidden Markov Models for Speech Signals", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-33, pp. 1404-1413, 1985.

[28] D. J Kershaw, A. J. Robinson & S. J. Renals, "The 1995 Abbot Hybrid Connectionist-HMM Large-Vocabulary Recognition System", *Proceeding of the 1996 ARPA Speech Recognition Workshop*, Feb. 1996.

[29] C. Lee, L. Rabiner, R. Pieraccini, and J. Wilpon, "Acoustic Modeling for Large Vocabulary Speech Recognition" *Computer Speech and Language*, vol. 4, 1990.

[30] K. Lee and H. Hon, "Large-Vocabulary Speaker-Independent Continuous Speech Recognition", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 123-126, April 1988.

[31] K. Lee, H. Hon, and R. Reddy, "An Overview of the SPHINX Speech Recognition", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, pp. 35-45, Jan. 1990.

[32] K. Lee, "Large-Vocabulary Speaker-Independent Continuous Speech Recognition: The SPHINX System", Ph.D. Thesis, School of Computer Science, Carnegie Mellon University, April. 1988.

[33] C. J. Leggeter and P. C. Woodland, "Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov Models". *Computer Speech & Language*, Vol. 9, pp.

171-185.

[34] S. Levinson, L. Rabiner, M. Sondhi, "An Introduction to the Application of the Theory of Probabilistic Function on a Markov Process to Automatic Speech Recognition", *the Bell System Technical Journal* 62(4), April, 1983.

[35] F.H. Liu, "Environment Adaptation for Robust Speech Recognition". Ph.D. Thesis, dept. of ECE, Carnegie Mellon University, June 1994.

[36] F. H. Liu, Personal Communication.

[37] F.H. Liu, A. Acero, and R. Stern, "Efficient Joint Compensation of Speech For the Effects of Additive Noise and Linear Filtering", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. I-257 - I-260, March, 1992

[38] J. Markel, and A. Gray, *Linear Prediction of Speech*, Springer-Verlag, 1976.

[39] P. J. Moreno, B. Raj, R. M. Stern, "Multivariate Gaussian-Based Cepstral normalization", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, May, 1995.

[40] P. J. Moreno, B. Raj, R. M. Stern, "A Unified Approach to Robust Speech Recognition", *Proceedings of Eurospeech 1995,* Madrid, Spain.

[41] P. J. Moreno, B. Raj, R. M. Stern, "Approaches to Environment Compensation in Automatic Speech Recognition", *Proceeding of the 1995 International Conference in Acoustics ICA'95,* Throndhein, Norway, June 1995.

[42] N. Morgan, H. Boulard, S. Greenberg and H. Hermansky, "Stochastic Perceptual Auditory-based models for speech recognition", *Proceedings of the 1994 International Conference on Spoken Language Processing (ICSLP)*. Vol. 4, pp. 1943-6, Yokohama, Japan.

[43] L. Neumeyer and M. Weintraub, "Probabilistic Optimum Filtering for Robust Speech Recognition", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. I417-I420, May 1994

[44] N. Nilsson, *Principles of Artificial Intelligence*, Tioga Publishing Co., 1980.

[45] D. Paul, and J. Baker, "The Design of the Wall Street Journal-based CSR Corpus", *Proceedings of ARPA Speech and Natural Language Workshop*, pp. 357-362, Feb., 1992.

[46] J. Picone, G. Doddington, and D. Pallett, "Phone-mediated Word Alignment for Speech Recognition Evaluation", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-38, pp. 559-562, March 1990.

[47] L. Rabiner, and B. Juang, "An Introduction to Hidden Markov Models", *IEEE ASSP Magazine* 3(1):4-16, Jan. 1986.

[48] B. Raj, *Personal Communication*.

[49] R. Schwartz, and Y. Chow, "The Optimal N-Best Algorithm: An Efficient Procedure for Finding Multiple Sentence Hypotheses", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, April 1990.

[50] R. Schwartz, Y. Chow, S. Roucos, M. Krasner, J. Makhoul, "Improved Hidden Markov Modeling of Phonemes for Continuous Speech Recognition", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1984

[51] S. Seneff, “A Joint Synchrony/Mean-Rate Model of Auditory Speech Processing”, *Journal of Phonetics,* Vol. 16, pp. 55-76, January 1988.

[52] R. M. Stern, A. Acero, F. H. Liu, Y. Ohshima, “Signal Processing for Robust Speech Recognition”, in *Automatic Speech and Speaker Recognition,* edited by Lee, Soong and Paliwal, Kluwer Academic Publishers, 1996.

[53] T. Sullivan, and R. Stern, “Multi-Microphone Correlation-Based Processing For Robust Speech Recognition”, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 91-94, April, 1993.

[54] D. Tapias-Merino. *Personal Communication*.

[55] A. Viterbi, “Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm”, *IEEE Transactions on Information Theory,* vol. IT-13, pp. 260-269, 1967.

[56] P. C. Woodland, M. J. F. Gales, D. Pye & V. Valtchev, “The HTK Large Vocabulary Recognition System for the 1995 ARPA H3 Task”. *Proceeding of the 1996 ARPA Speech Recognition Workshop,* Feb. 1996.

[57] S. J. Young & P. C. Woodland, HTK Version 1.5: User, Reference and Programmer Manual, Cambridge University Engineering Dept., Speech Group, 1993.