University of California
Los Angeles

Low Complexity Spectral Imputation for Noise Robust Speech Recognition

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Science in Electrical Engineering

by

Julien van Hout

2012

Abstract of the Thesis

Low Complexity Spectral Imputation for Noise Robust Speech Recognition

by

Julien van Hout

Master of Science in Electrical Engineering

University of California, Los Angeles, 2012

Professor Abeer Alwan, Chair

With the recent push of Automatic Speech Recognition (ASR) capabilities to mobile devices, the user's voice is now recorded in environments with a potentially high level of background noise. To reduce the sensitivity of ASR performance to these distortions, techniques have been proposed that preprocess the speech waveforms to remove noise effects while preserving discriminative speech information. At the expense of increased complexity, recent algorithms have significantly improved recognition accuracy but remain far from human performance in highly noisy environments.

With a concern for both complexity and performance, this thesis investigated ways to reduce the corruptive effect of noise by directly weighting the power-spectrum (SMF_{pow}) or log-spectrum (SMF_{log}) of speech by a mask whose values are within [0,1] and are indexed on the local relative prominence of speech and noise energy. Additional contributions include a low-complexity approach to mask estimation and the use of spectral flooring for matching the dynamic range of clean and noisy spectra. These two techniques are evaluated on two standard noisy ASR databases: the Aurora-2 connected digits recognition task with 11

words, and the Aurora-4 continuous speech recognition task with 5000 words.

On the Aurora-2 task, the SMF_{log} algorithm leads to state-of-the-art performance, with a limited complexity compared to existing techniques. The SMF_{pow} technique, however, results in many insertions that we attribute to the rather weak language model present in the Aurora-2 setup. On the Aurora-4 task, both algorithms show significant improvements over the un-enhanced baselines. In particular, word-accuracies obtained with SMF_{pow} approach those of a state-of-the-art front-end algorithm, on half of the noise types. Yet, the performances are heavily noise dependent, suggesting that the proposed technique is effective only given a good initial mask estimation.

This study confirms the potential of techniques that are based on direct spectrum masking, and proposes a framework for doing so. Future work will need to consider more elaborate mask estimation techniques to further improve on the performance.

The thesis of Julien van Hout is approved.

Gregory Pottie

Jennifer Wortman Vaughan

Abeer Alwan, Committee Chair

University of California, Los Angeles

2012

to my parents and my sister for their love and support,

Table of Contents

List of Figures

List of Tables

Acknowledgments

First, I would like to express my gratitude to Professor Abeer Alwan, my advisor at UCLA and chair of my thesis committee. She has provided me with her trust, support and guidance through my graduate research at UCLA. I owe a great part of my accumulated scientific knowledge to her as well as my exposure to the Speech research community. She is the main reason why obtaining this Master's of Science degree was a truly rich and rewarding experience.

I wish to acknowledge Professor Gregory Pottie and Professor Jennifer Wortman Vaughan from UCLA for agreeing to serve as members of my thesis committee. I greatly appreciate their time, effort, and curiosity about this research.

As a recent graduate of UCLA's Speech laboratory, Dr. Jonas Borgstrom has been following my work closely since I first began graduate research in speech recognition. His responsiveness and experience has inspired and greatly impacted the quality of this work. I am very grateful to him for this.

I would also like to thank all SPAPLers, my co-workers at UCLA, for many fruitful discussions, but above all, for fostering a great atmosphere within the lab. They made research a truly pleasant as well as enriching experience.

Finally, I could not thank enough all my old and new friends from all over the world for two delightful years here in Los Angeles, and lastly Caitlin for being an everyday source of love, motivation and support.

This research is funded in part by the DARPA RATS award via SRI International.

CHAPTER 1

Introduction

This chapter provides background and motivation for the work presented in this thesis. We first introduce the need for a low-complexity noise robust Automatic Speech Recognition (ASR) system. Next, we present the basic components of an ASR system and review relevant work on noise robust preprocessing. Lastly, we provide a thesis outline.

1.1 Motivation

For the past 60 years, research in ASR resulted in steady incremental improvements that now enable widespread use of this technology for applications such as dictation, automated call-centers, domotics or military purposes. One of the factors that most influences recognition accuracy is the possibility that the ASR system is used in a noisy environment. In this case, there might be a mismatch between the recorded waveforms and the waveforms used to train the recognizer, causing a steep drop in recognition rate as the Signal-to-Noise Ratio (SNR) decreases.

Even though this mismatch can be reduced by training the system with noisy waveforms, this approach is still limited in that there is a potentially large number of noise characteristics. This is especially true with the recent increase of ASR-powered mobile devices, often used in highly noisy and non-stationary en-

vironments such as cars or restaurants. Also, mobile devices suffer from limited computational capabilities that restricts the complexity of the processing that can be performed. These observations motivated the development of low-complexity methods for preprocessing the speech waveforms in a way that attenuates the mismatch between clean and noisy recordings.

In the last 15 years, efforts to reduce this noise mismatch have produced large accuracy gains in adverse conditions over the traditional non-robust approaches, while preserving good performance in clean conditions. Yet, current state-of-the-art techniques still fall far behind the computational capability of the human brain, especially for non-stationary noises or when the SNR lies below 5dB SNR. In short, effective compensation for real-life acoustic noise in speech recognition is still an open problem, and any improvements in that direction would lead to direct benefits for currently deployed systems.

The next sections will provide an overview of the traditional HMM-based approach to ASR, and will then describe the ideas behind some state-of-the-art noise robust ASR front-end algorithms.

1.2 Background on Automatic Speech Recognition

An ASR system automatically performs transcription of speech to text. In an offline phase, the system is trained with thousands of hours of speech waveforms and their corresponding text transcriptions. During this phase, the system builds a model for each acoustic unit (phonemes, triphones, or words) seen in training. In the online phase, the system matches the received waveform to the acoustic models developed in training. After using syntax and grammar information via the language model, the system outputs the most likely word sequence. Figure

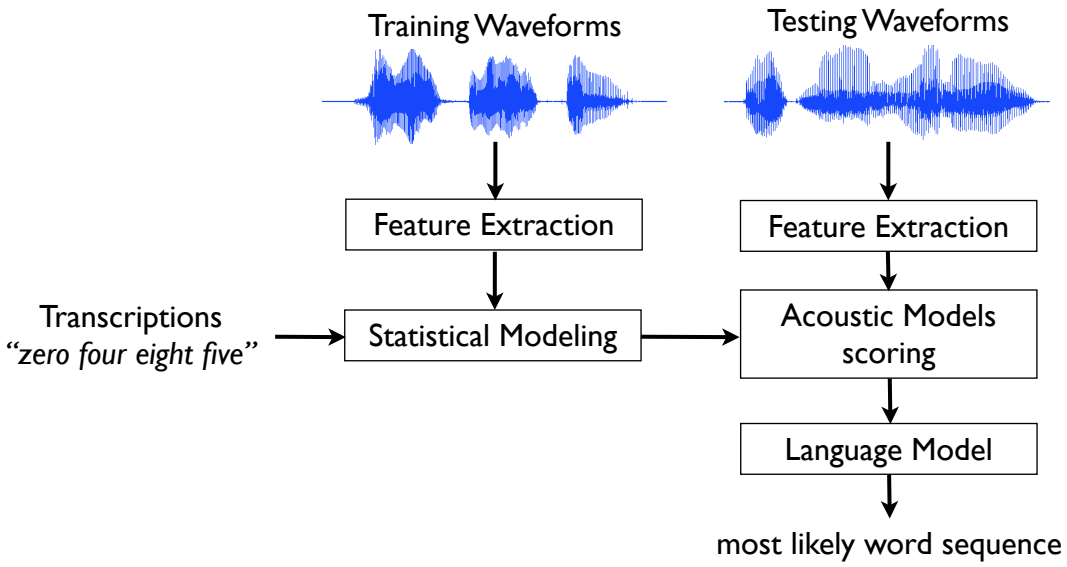


Figure 1.1: A traditional ASR system is composed of a preprocessing algorithm (feature extraction), statistical acoustic models and a language model

1.1 shows the diagram of a traditional ASR system.

In this section, we present some background on a popular ASR structure, namely the Mel-Filtered Cepstral Coefficients (MFCC) preprocessing followed by an HMM-based statistical model.

1.2.1 Front-end Feature Extraction

Speech waveforms cannot be fed directly to a statistical engine for recognition. Some dimensionality reduction has to be performed first. Waveforms are digital representations of the pressure variations, and are sampled at 8kHz or 16kHz in most applications. Since discriminative speech information has been found to lie at frequencies up to at least 4000Hz, time samples are highly correlated and thus are not well suited for building discriminative models for recognition. The first step, before doing any statistical learning, is to convert these waveforms

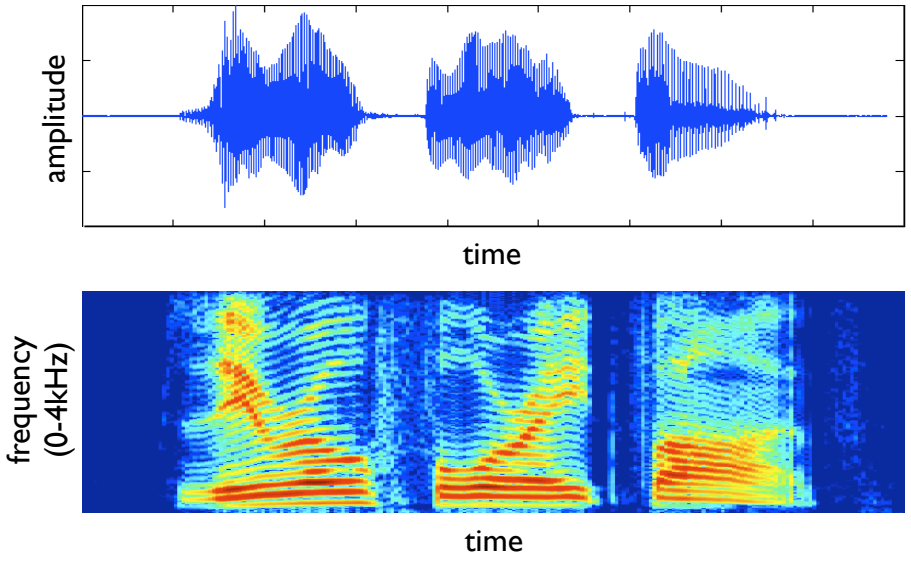


Figure 1.2: The spectral representation (bottom) captures which time-frequency components of the original waveform (top) have the most (in red) or the least energy (in blue). The sampling rate is 8kHz and the number of FFT channels is 512

into time-varying features that carry discriminative information about the speech signal while reducing the redundancy of the original data.

A successful approach to feature extraction that is now widespread in the ASR community is called Mel-Frequency Cepstral Coefficients (MFCC), and will be introduced as described in [3]. Like most feature extraction methods, MFCCs are computed via a spectro-temporal representation of speech called a spectrogram, obtained by taking the short-time Fourrier transform of the waveform using a 25ms Hamming window and a 10ms shift between each frame. The resulting spectrogram shows which time-frequency bins carry the energy of the speech waveform (See Figure 1.2). Since most of the speech information is thought to be carried by the vocal tract resonances, also referred to as formants, this frequency domain representation is essential to capture discriminative information for ASR.

The MFCC algorithm then groups the frequency channels together to further reduce the dimension via a non-linear filterbank inspired by the human cochlea. Different auditory-inspired filterbanks have been proposed to this end, but they all share the characteristics of using several narrow filters at low frequencies and few wide filters at high frequencies. MFCCs use a Mel-filterbank, with triangular shaped filters and a number of channels between 25 to 40. The energy at the output of each filter is then compressed using a log operator, also inspired by the auditory system. Then, a Discrete Cosine Transform (DCT) is taken on each time-frame of the Mel-spectrum as a way to reduce the remaining cross-channel correlation. For each frame, only the first 13 DCT coefficients are kept since they account for most of the signal energy. At this point, it is usually found helpful to perform weighting along the frequency axis (liftering) and to normalize each time-frame by subtracting its mean for each of the 13 coefficients (cepstral mean normalization). In a last step, the dynamic nature of speech and the particularly discriminative information about energy onsets is taken into account. To this end, first and second derivatives of the DCT coefficients are computed. These 13+13 coefficients are appended to the original 13 coefficients, for a total of 39 coefficients per 10ms time-frame.

These MFCC features have a lower dimension and are less correlated than the original spectrum or waveform. Such a preprocessing is an essential first step to build discriminative statistical models.

1.2.2 Acoustic Modeling

Statistical modeling using Hidden Markov Models (HMMs) is a widely used approach in the field of automatic speech recognition [4]. This section introduces the basic ideas behind the use of HMMs for acoustic modeling, but the specifics

of each HMM configuration used in our experiments are left for Chapter 3.

A HMM is a statistical Markov model used to model a sequence of observations as the output of a network of hidden states. Each state has its own output probability density function, but transfers between states are made possible by the mean of transition probabilities. Since the observation sequence is the only information available to the user, these output distributions and transition probabilities are the parameters that are used to parametrize the HMM.

Formally, let us define a continuous-valued HMM Λ comprised of N discrete states $(s_1, \ldots, s_N)$ and an observation space $O \subset \mathbb{R}^n$ (see Figure 1.3). Suppose this HMM generates the occupied state sequence $\mathbf{S}_{1:T} = (\mathbf{s}(1), \ldots, \mathbf{s}(T))$ and the observation sequence $\mathbf{O}_{1:T} = (\mathbf{o}_1, \ldots, \mathbf{o}_T) \in O^T$. Using these notations, Λ can be entirely described by its parameters $\lambda = (\pi, \mathbf{A}, \mathbf{B})$ where:

- $\pi = (\pi_1, \ldots, \pi_N)$ are the initial state probabilities, namely $\pi_i = \Pr(\mathbf{s}(1) = s_i)$
- $\mathbf{A} = (a_{ij})_{1 \leq i,j \leq N}$ are the state transition probabilities, namely $\forall t \in [2, T]$:

$$a_{ij} = \Pr(\mathbf{s}(t) = s_j \mid \mathbf{s}(t-1) = s_i)$$

- $\mathbf{B} = (b_i)_{1 \leq i \leq N}$ are the state output distributions, namely $\forall t \in [1, T]$:

$$b_i(\mathbf{o}_t) = \Pr(\mathbf{o}_t \mid \mathbf{s}(t) = s_i)$$

In ASR, left-to-right HMMs are used to model the speech acoustic units that will be later connected to form the word transcriptions. Each HMM can be chosen to model units like phonemes, triphones, or even whole words depending on the size of the vocabulary and the amount of data available for training. Using HMMs with multiple states allow to model the time dynamics in the observations of these acoustic units. The various output distributions at each state model the dynamic

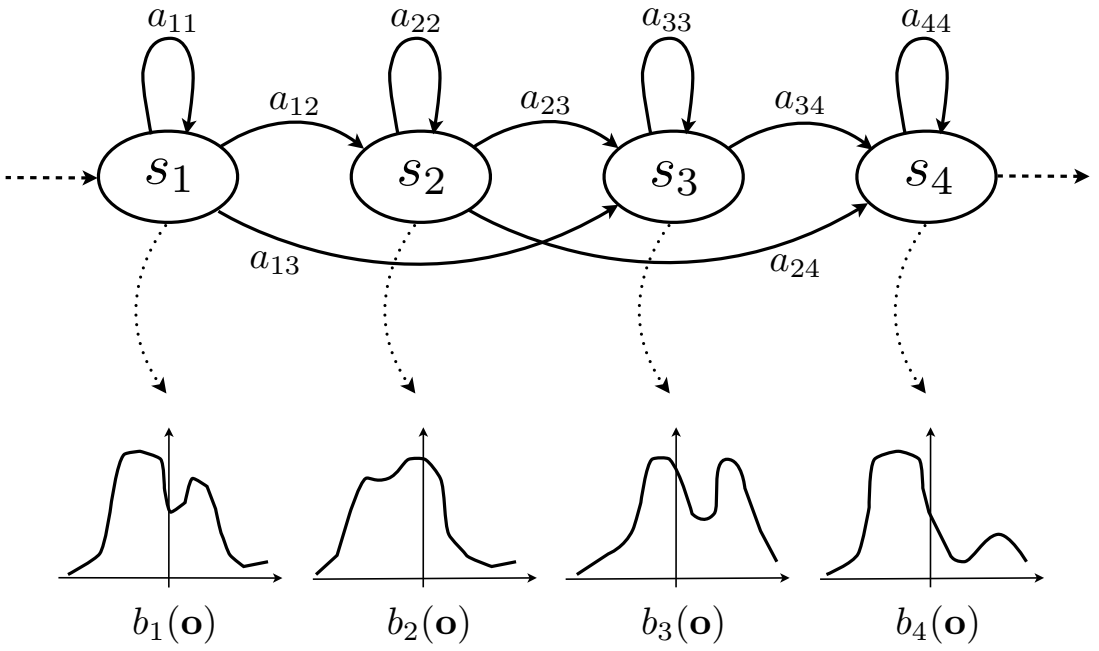


Figure 1.3: Diagram of a causal four-states HMM, for an observation space $O \subset \mathbb{R}$

in the observed vector, and allow for flexibility in the duration of these patterns via the probabilistic nature of state-transitions. In ASR, state-specific output distributions are generally modeled as Gaussian Mixture Models (GMM), where the number of Gaussians per mixture can be tuned depending of the amount of available training data. These HMMs are then connected to model longer speech segments, thus providing statistical information at the sentence level.

Three main problems must be solved in order to use HMMs for acoustic modeling (see [4]):

Problem 1 Given a sequence of observations $\mathbf{O}_{1:T} = (\mathbf{o}_1, \ldots, \mathbf{o}_T)$, how do we find the parameters $\lambda = (\pi, \mathbf{A}, \mathbf{B})$ of the HMM that bests models these observations?

This problem pertains to the training process, where the observations and the transcriptions are known. In training, we are trying to find the best

HMM for each of the acoustic units present in the transcriptions. Problem 1 is solved by the Baum-Welch Reestimation algorithm.

Problem 2 Given an HMM with parameters $\lambda = (\pi, \mathbf{A}, \mathbf{B})$, how do we compute the probability of a given observation sequence $\mathbf{O}_{1:T} = (\mathbf{o}_1, \ldots, \mathbf{o}_T)$?

This problem is related to the recognition process, in which we try to decide which model is most likely to explain a given sequence of observations. This problem can be solved using the Forward-Backward Procedure.

Problem 3 Given an HMM with parameters $\lambda = (\pi, \mathbf{A}, \mathbf{B})$ and an observation sequence $\mathbf{O}_{1:T} = (\mathbf{o}_1, \ldots, \mathbf{o}_T)$, how do we find the most likely state sequence $\mathbf{S}_{1:T} = (\mathbf{s}(1), \ldots, \mathbf{s}(T))$?

This problem also pertains to the recognition process, and can be solved using the Viterbi algorithm.

Further optimizations like tying states from different HMMs so that they share the same observation pdfs allows for better acoustic modeling with limited data. Moreover, creating specific models for silences and short pauses has also been shown helpful to model observations from between the words.

1.2.3 Language Modeling

The last element of the back-end ASR system, the language model, allows us to add prior information about the structure of the target language. This information can be thought as a grammar that helps avoid unlikely acoustic unit combinations and correct errors such as *"Luke likes to eat arts"* to *"Luke likes to eat tarts"*. Using a good language model provides a large gain in accuracy, especially in large vocabulary recognition tasks where HMM fails to take into account the strong a-priori correlation between neighboring words. Among the

most popular language models are statistical models like N-grams, which model the conditional probability of a word given the past N words. Designing good language models is a research field on its own, and we suggest the interested reader to refer to [3] for more details.

1.3 ASR in Adverse Noise Conditions

In this section, we present the challenges posed by using ASR when the speech signal is corrupted by additive background noise. First, we review the difficulties posed by additive background noise with the traditional MFCC preprocessing. Then, we present some alternative front-end techniques that have been used for increasing the recognition accuracy in noise. Last, we describe the Missing Feature framework, a group of techniques for noise robust ASR that has received attention in the last 15 years.

1.3.1 Difficulties in Noise

When noise is present in the background, the speech waveform is distorted. As a result, the spectrogram of noisy speech no longer matches the spectrogram of clean speech, since the spectral components of the noise are now also present (Figure 1.4). Therefore, the common scenario of training the ASR system with MFCCs computed on clean speech and testing this system on noisy speech will be very challenging for the back-end when matching the noisy features with the clean models.

In fact, experiments show that even adding noise at a relatively high SNR (20dB) increases the word-error rate by a factor of 6 when using MFCC preprocessing (Table 1.1). At low SNRs (0-5dB), recognition rates drop below 50%

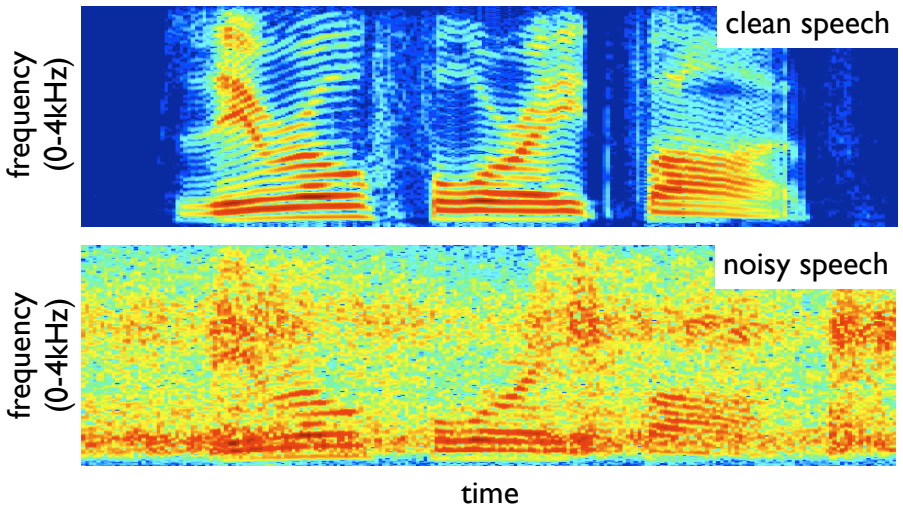


Figure 1.4: The spectrogram of an utterance corrupted by artificially adding subway noise at 0dB SNR (bottom) is highly distorted compared to the spectrogram of the corresponding clean speech utterance (top)

while human recognition performance remains good down to -10dB SNR. This steep accuracy drop using MFCCs is even more severe on a large-vocabulary task, pressing the need for noise robust feature extraction algorithms.

1.3.2 Review of Noise Robust Techniques

This subsection describes some approaches to feature extraction that resulted in improved performance for ASR in noisy environments.

Compensation for the noise distortions can be done at the waveform or spectrogram level, to make the noisy speech signal resemble the clean speech signal via speech enhancement, before using traditional features like MFCCs. Efforts in this area have attracted attention since [5, 6] introduced a framework for Minimum Mean-Square Error (MMSE) based spectral amplitude estimation in 1984. Many approaches to speech enhancement have been developed since then, and

SNR (dB)	**clean**	**20**	**15**	**10**	**5**	**0**
Word-Accuracy (%)	99.6	97.6	93.6	78.7	45.8	11.9

Table 1.1: The Percent Word-Accuracy of ASR using MFCC features degrades steeply as the SNR decreases. Experiments have been carried out on the Aurora-2 connected digits recognition task as described in Chapter 3. Numbers shown are averaged over 8 types of additive noises

were recently summarized in [7].

It was also shown successful to perform compensation as part of the feature extraction process itself, by modifying steps of the MFCC processing to retain less variability from the noise. To this end, the Mean Variance Arma (MVA) technique [8] performs mean and variance normalization on the MFCC coefficients before smoothing the features with an Auto-Regressive Mean Averaging (ARMA) filter. As another example, the recently introduced Power Normalized Cepstral Coefficient (PNCC) algorithm [9] suggests to replace some steps of the MFCC computation by an auditory inspired and more noise robust processing. Its major differences with the MFCC algorithm are the use of a Gammatone auditory filterbank [10] in lieu of a Mel-filterbank, a power compression instead of a log-compression, and a noise compensation step on the spectrogram based on tracking and subtracting the noise floor.

The idea of emphasizing the information from the spectral peaks rather than the valleys has also been shown to help retain discriminative speech information. In the auditory inspired front-end described in [11], the authors introduce a peak isolation technique (PK-ISO) based on liftering, half-wave rectification and peak normalization. An additional step to PK-ISO was proposed in [12] that consists in deciding on a constant peak-to-valley ratio to further increase the similarity

between clean and noisy spectra. These ideas have also been shown to help build robust features when used in conjunction with noise-suppression algorithms. In [13], the noisy spectrogram is enhanced via spectral imputation (cf. Section 1.3.3) before isolating the peaks via Log-Spectral FLooRing (LS-FLR), a technique similar to [11] but applied in the log-spectral domain. Lastly, tracking spectral peaks across time has been shown in [14] to provide information that can be used to build low-dimensional noise-robust features for a digit recognition task.

Several other characteristics of human speech have been successfully exploited to build noise robust preprocessing algorithms. Among these, the forward-masking characteristics of the human auditory system have inspired several authors into building front-ends that emphasize signal onsets rather than slowly varying amplitudes. For instance, the study from [11] coupled their peak isolation algorithm with a auditory-derived model of forward-masking and obtained significant improvements in recognition accuracy in noisy conditions. Other discriminative characteristics of speech lies in its temporal dynamics, as rapid energy variations like plosive consonants could also carry useful information. Yet, computing frames every 10ms is sometimes not sufficient to capture these quick variations. This motivated [15] to introduce a variable frame rate for analysis, based on a measure of local Euclidian distances on MFCC vectors. This algorithm uses as many frames as the fixed-rate method, but allocates more frames to the quickly varying parts of the speech and less to the slowly varying parts, like steady vowels. Further investigations have found other distance measures such as the entropy between MFCC vectors [16] to further increase gains in accuracy.

Lastly, some approaches do not fall into the standard MFCC framework, either because they use an alternative representation to the spectrogram, or because they exploit spectrographic information differently. Analysis of the spectral

modulations of speech [17] reveals that noise can be attenuated by focusing on amplitude modulations around 4Hz, where most of the speech energy is present. These approaches to computing the modulation spectrogram can be successfully combined with previously described frame-level noise compensation techniques [18]. With a different scope, features based on spectro-temporal filtering of the spectrogram by 2-d Gabor wavelets have been successfully applied to small vocabulary noisy ASR [19]. This technique generates high-dimensional features that capture local spectrographic modulations along several directions on the time-frequency plane. After dimensionality reduction using techniques based on neural networks, these multi-stream features are appended to MFCCs to provide additional discriminative power in both clean and noisy conditions.

1.3.3 The Missing-feature Approach to Noise Robust ASR

The Missing-Feature (MF) approach to noise robust ASR is a framework that has become increasingly popular over the last decade [20]. This subsection presents a brief introduction to MF for ASR, and reviews some popular algorithms that have influenced the proposed approach, described in Chapter 2.

The MF framework is based on the idea that if one could compute a mask labeling each time-frequency bin of the spectrogram as reliable or unreliable, then this information could be helpful to the recognition process. Two ways have been proposed to account for the reliability information: marginalization and imputation. The first technique reduces the weight of the unreliable bins in the back-end at the recognition stage, whereas the second performs estimation of the unreliable parts of the spectrogram in the front-end, before computing traditional MFCCs (see Figure 1.5). While the marginalization technique is optimal in theory, it is usually discarded for computational reasons, and because it precludes the use

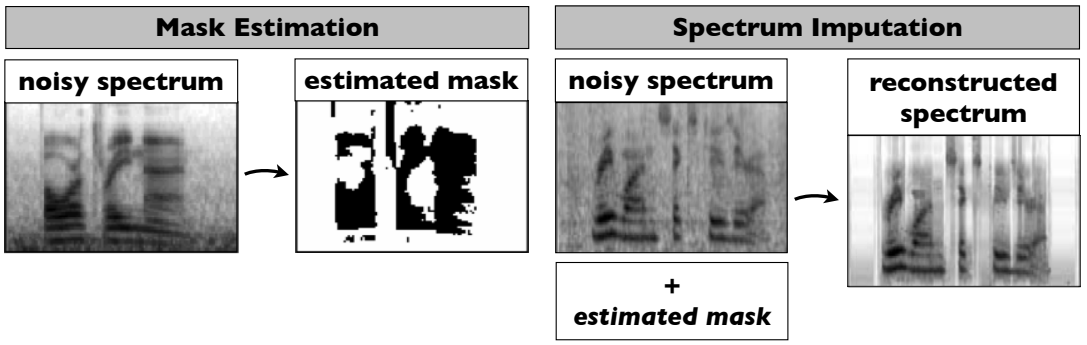


Figure 1.5: In the MF framework, a mask is first estimated to label the time-frequency bins of the noisy spectrum as reliable (black) or unreliable (white). Then, the information from this mask is used by the imputation algorithm to infer the unreliable parts of the noisy spectrogram.

of the cepstrum for feature representation. The following will describe several techniques that have been used for mask estimation and spectral imputation.

Mask estimation aims at labeling time-frequency bin of the spectrogram as reliable if the speech energy is dominant, or as unreliable if the noise energy is dominant. An accurate mask provides highly valuable information to the recognizer, since it enables it to focus on the reliable components, the most discriminative for speech recognition. Estimating a good mask is a difficult problem, as it requires a criterion to tell apart speech from noise energy. Techniques have been proposed that look at clues based on spectral-subtraction [21], SNR [22], and the harmonic structure of voiced speech [23]. Other authors successfully used statistical classifiers with two-classes, based on Bayes rule [24] or HMMs [25].

Spectral imputation is the process of estimating the value of unreliable components of the spectrogram based on the value of the reliable bins. The resulting reconstructed spectrogram then is fed to a traditional front-end feature extraction scheme like MFCCs. The first approaches that have been proposed to perform

spectral imputation, such as conditional mean imputation [26] and MAP inference [27] are based on modeling statistical information of the clean speech spectrum. Some recent approaches feature the use of compressive sensing as a means for data recovery under the assumption of sparsity of the clean speech signal. In [28, 29], a basis for sparsity is obtained by accumulating a large dictionary of exemplars whereas [30, 13] exploit the time-frequency correlation of speech and use an image processing inspired two-dimensional Haar transform on the spectrographic data. Lastly, a recent study [31] suggests that a simple imputation technique, directly using the mask as a set of multiplicative coefficients on the power spectrum followed by variance normalization leads to state-of-the-art results on a large vocabulary task, if the mask is using oracle information about the speech location. This pilot study suggests that on tasks requiring a strong language model, large improvements will more likely originate from more accurate mask estimation than sophisticated imputation. Yet, such an imputation technique by mask weighting hasn't been successful with more realistic estimated masks.

This thesis investigates the idea of weighting the spectrum with an estimated mask, by proposing a novel mask estimation procedure as well as two different frameworks for mask weighting.

1.4 Organization of the Thesis

Following the background information and the motivations presented above, Chapter 2 describes the proposed noise robust front-end algorithms for feature extraction. Then, Chapter 3 presents the evaluation setup and comments on the results. Finally, Chapter 4 offers a conclusion and directions for future work.

CHAPTER 2

Proposed Missing-feature based Feature Extraction Approach

In this chapter, the proposed low complexity feature extraction approach is described. In order to follow the Missing-Feature framework, we first introduce our approach to estimating a soft-decision mask on the spectrum. In the second part of this chapter, we present and justify our approach to spectral imputation.

2.1 Mask Estimation

In this section, we describe the steps in the computation of the SNR-based soft decision mask. Our goal is to determine in which areas of the noisy speech spectrogram does the energy originate from the speech rather than the background noise. We output a mask that will have a value close to 1 in these areas, and a value close to 0 where the noise is prominent. A flowchart for the algorithm is shown in Figure 2.1. The following section assumes that a Mel-filterbank is used, but the same ideas could be applied using a Gammatone filterbank.

2.1.1 Noise Modeling

This first subsection lays some theoretical ground on modeling the corruptive effect of the noise.

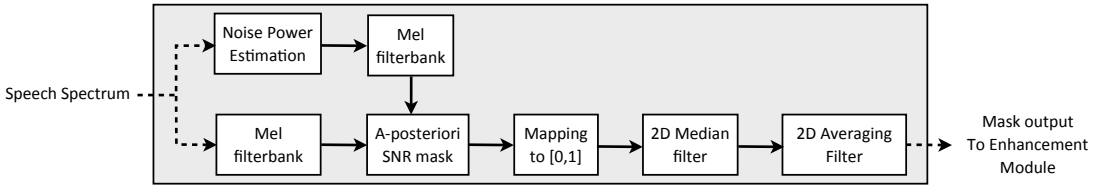


Figure 2.1: Flowchart of the proposed mask estimation technique

In this study, we will be considering specific noisy conditions under which the Missing-Feature framework has been shown to be more appropriate. Specifically, we will assume that the observed segment of speech has been additively corrupted by noise that is Gaussian, zero-mean and uncorrelated with the speech signals. These assumptions are reasonable when considering speech corrupted by background noise such as speech recorded by smartphones in a car. Sometimes, however, the distortions will be speech dependent, for instance when recording in a room with high reverberation. In that case, our assumptions may no longer hold and the proposed method might fail to effectively tell apart the distortions from the actual speech.

Following these assumptions, we perform short-term spectral analysis of the observed noisy waveform to obtain:

$$Y(i,k) = S(i,k) + N(i,k)$$

where $Y(i,k)$ is the amplitude of the observed noisy spectrum at time frame i and frequency channel k, $S(i,k)$ is the amplitude of the clean speech and $N(i,k)$ models additive corruptive noise. In order to compute the power spectrum, it is commonly assumed that the signals are in phase. In this case we can write:

$$|Y(i,k)|^2 \simeq |S(i,k)|^2 + |N(i,k)|^2 . \tag{2.1}$$

Following the Gaussian framework described in [32] and assuming that the noise variance is reasonably constant over the short duration of this analysis, $|N(i,k)|^2$

can be modeled as an exponential random variable with probability density function:

$$p(|N(i,k)|^2) = \frac{1}{\sigma_N^2(k)} \exp\left(-\frac{|N(i,k)|^2}{\sigma_N^2(k)}\right)$$

where $\sigma_N^2(k)$ is the noise variance for DFT channel k. The latter stationarity assumption will be essential in our analysis and implies that while the noise variance is allowed to gradually increase or decrease over time, we won't expect rapid variations in the background noise like bursts or other highly non-stationary noises.

After applying the non-linear Mel-filterbank on the original power spectrum, the Mel-filtered power spectrum is defined by:

$$|Y(i,m)|^2 = \sum_{k=c_{m-1}}^{c_{m+1}} w_m(k) \, |Y(i,k)|^2 \tag{2.2}$$

where c_m denotes the center frequency of the m^{th} Mel-filter and $w_m(k)$ represents its weighting across frequencies. As a direct consequence of (2.1) the resulting Mel-spectral power is given by:

$$|Y(i,m)|^2 = |S(i,m)|^2 + |N(i,m)|^2 \tag{2.3}$$

where we define

$$\begin{aligned} |S(i,m)|^2 &= \sum_{k=c_{m-1}}^{c_{m+1}} w_m(k) \, |S(i,k)|^2 \\ |N(i,m)|^2 &= \sum_{k=c_{m-1}}^{c_{m+1}} w_m(k) \, |N(i,k)|^2 . \end{aligned}$$

Because a weighted sum of exponentially-distributed random variables with possibly unequal variances follows a generalized χ^2 distribution, [32] found that the distribution for $|N(i,m)|^2$ can be approximated by:

$$p_m^{\chi^2}(|N(i,m)|^2) = A_m \left(\frac{|N(i,m)|^2}{\sigma_N^2(m)}\right)^{(k_m/2-1)} \exp\left(-\frac{|N(i,m)|^2}{\sigma_N^2(m)}\right) \tag{2.4}$$

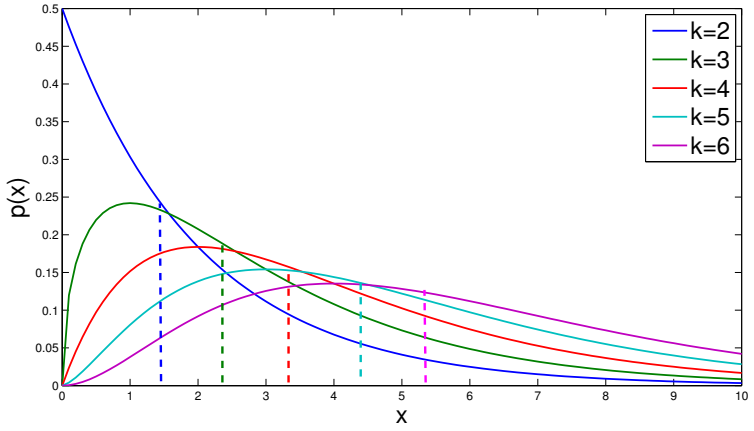


Figure 2.2: Probability density function of a χ^2 distribution for various degrees of freedom. The median value is shown with a dashed line for each value of k

where $\sigma_N^2(m)$ is the size parameter of the χ^2 distribution, k_m the channel-specific number of degrees of freedom, and A_m a normalizing factor. Theoretically, k_m corresponds to the number of random variables averaged in Eq. 2.2, that is, to the width of the m^{th} Mel-filter. Since the most narrow filter has a bandwidth of approximately 100Hz, a typical value for k_m should be above 5, and will increase as we move up in frequency and use wider filters. In practice though, the value for k_m is more reliably estimated on a subset of the data using the ratio of non-central moments of the observed power distribution. We won't need to obtain an actual estimate of k_m in the current study, but an interested reader can refer to [32] for more details.

A direct consequence of (2.3) and (2.4) is that for a fixed $|S(i,m)|^2$ we have:

$$p(|Y(i,m)|^2) = \begin{cases} 0 & if \ \ |Y(i,m)|^2 < |S(i,m)|^2 \\ p_m^{\chi^2}\left(|Y(i,m)|^2 - |S(i,m)|^2\right) & otherwise \end{cases}$$

By looking at a typical χ^2 distribution (Fig. 2.2), we find that while it is not

perfectly Gaussian, it still is roughly symmetrical and puts most of its weight in the neighborhood of the median, especially in the present case where $k \geq 5$. This observation motivates the idea that taking the median of the noise $|N(i,m)|^2$ over several frames would effectively reduce the fluctuations and leave us with a good indicator (maybe biased) of $\sigma_N^2(m)$. This fact will be useful later in supporting our approach to canceling the SNR distortions caused by high bin-to-bin variability by using a moving median filter.

2.1.2 Noise Variance Cancellation via Spectro-Temporal Filtering

The *reliable/unreliable* criterion we wish to derive, the soft-valued mask, should be based on a long-term estimate of the noise power rather than depend on local variations due to the noise variance. Yet, as described in Chapter 1, traditional SNR-based mask estimation techniques like [33] fail to take into account the noise variability in deciding wether a bin is *reliable* or *unreliable*. This subsection will propose additional steps based on two-dimensional median filtering and blurring to remove these fluctuations from the mask, resulting in more spatially coherent decision regions.

The traditional method to compute SNR-based soft masks is fairly intuitive. The time-frequency bins where the Signal-to-Noise Ratio is high have higher speech energy than noise energy. Therefore, they are tagged as *reliable* and the corresponding mask value is set close to 1. Conversely, if the SNR is low because the noise is in higher proportions than the speech then the bin value is deemed *unreliable*, and is given a mask value close to 0. In practice, this approach requires an estimate of the noise power at each bin in order to compute the SNR. We will first assume that we are given such an estimate $\tilde{\sigma}_N^2(i,m)$ and will describe the steps in computing the SNR-based mask. Then, Sections 2.1.3 and 2.1.4 will

describe two methods to actually estimate the noise power.

The bin-wise SNR $\gamma_{i,m}$ is defined by:

$$\gamma_{i,m} = 10 \cdot \log \left(\max \left(\rho_{min}, \frac{|Y(i,m)|^2}{\tilde{\sigma}_N^2(i,m)} \right) \right)$$

where ρ_{min} is a flooring threshold set to 0.5 that avoids unnecessarily high amplitude negative values for the resulting SNR. The soft-mask is generated as in [33] by mapping the SNR estimate $\gamma_{i,m}$ to the interval $[0, 1]$ using a simple sigmoid function. The mapping is of the form:

$$f_{\alpha,\beta}(x) = \frac{1}{1 + \exp(-\alpha(x - \beta))}$$

where α sets the sigmoid slope and β sets the sigmoid center. In other words, if we define

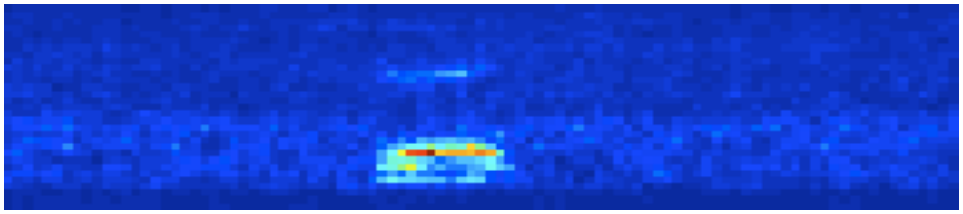
$$M_{i,m}^{(1)} = f_{\alpha,\beta}(\gamma_{i,m}) \in [0, 1]$$

to be a first estimate of the soft-mask values, then α allows to set the sharpness of the mapping while β is a tunable offset for the SNR estimate $\gamma_{i,m}$. Tuning for α allows to decide for soft or hard $[0, 1]$ decisions, while tuning for β will allow to compensate for any bias in the SNR estimate $\gamma_{i,m}$. Since there is no objective criterion to help us decide on those parameters, we tune α and β empirically like in [33], by picking the values that maximize our recognition accuracy jointly with the enhancement technique that will be introduced in Section 2.2. For the current setup, we have found $\alpha = 0.2$ and $\beta = 4dB$ to give the best results and will be using these values in the following. It is important to note that if we were to use a different enhancement technique, the optimal values for α and β may be different. Figure 2.3 shows the first three steps in the mask computation, from the observed spectral power $|Y(i,m)|^2$ to the SNR $\gamma_{i,m}$ and finally the first mask $M_{i,m}^{(1)}$.

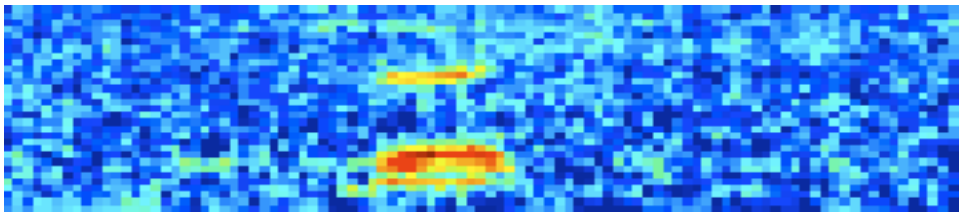
Because we use a noise estimate $\tilde{\sigma}_N^2(i,m)$ that is relatively smooth in time and frequency and carries information about the average noise power, the noise fluctuations present in $|Y(i,m)|^2$ will propagate to $\gamma_{i,m}$ and finally to $M_{i,m}^{(1)}$. As can be verified in Figure 2.3c, these fluctuations are heavily corrupting the initial mask estimate, especially in the silence regions where some coefficients are very close to 1 even though no actual speech is present. One way to attenuate this effect could be to change the mapping $f_{\alpha,\beta}(x)$ so as to compress those values closer to zero. While this would work in non-speech regions, it would also attenuate useful lower-energy speech information that might originate from unvoiced sounds like the phoneme /*f*/ in *five*.

These observations motivate us to use an alternative approach, based on two-dimensional median filtering, as a way to remove these fluctuations while leaving speech information intact. This approach can be justified in several ways. First, since the noise power estimate $\tilde{\sigma}_N^2(i,m)$ is assumed to be smooth both in time and frequency, applying the median filter on $M_{i,m}^{(1)}$ or $|Y(i,m)|^2$ will lead to similar final outputs. Now, the above study of the properties of the χ^2 distribution from which $|Y(i,m)|^2$ originates shows that the median provides a good estimate of $|S(i,n)|^2 + \sigma_N^2(m)$ and thus will effectively eliminate the noise fluctuations. Second, the median has similar smoothing properties to the mean while being less sensitive to the common outliers that emerge from a χ^2 distribution. Third, two-dimensional median filters tend to be more preserving of the abrupt edges of the speech regions while a blurring effect could occur if we were to use a Gaussian filter, or another local averaging technique.

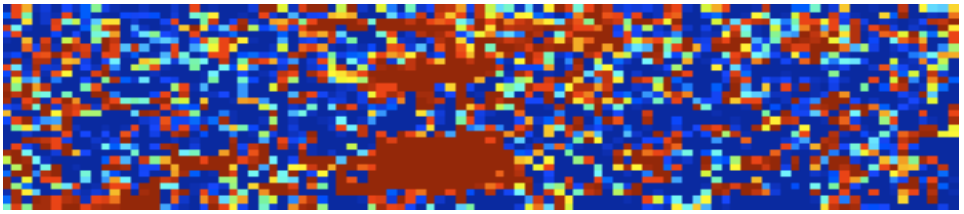
With these ideas in mind, we apply a 3×5 median filter to our mask to obtain $M_{i,m}^{(2)}$. As shown in Figure 2.4a, $M_{i,m}^{(2)}$ exhibits less noise fluctuations and shows an increased spectro-temporal coherence: *reliable* and *unreliable* bins are



(a) Mel-power spectrum $|Y(i,m)|^2$

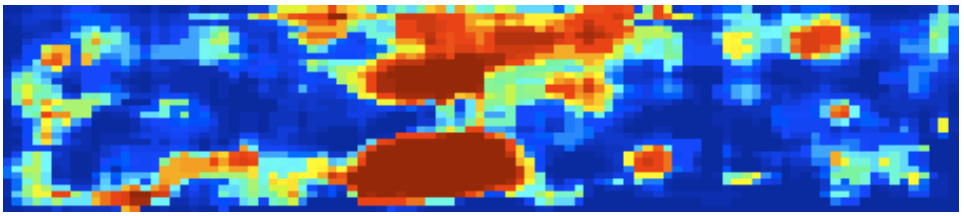


(b) Local bin-wise SNR $\gamma_{i,m}$

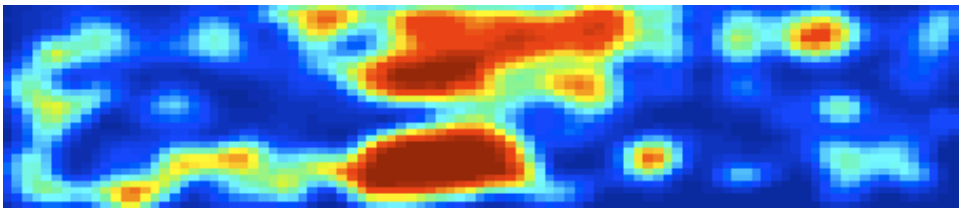


(c) Soft-mask $M_{i,m}^{(1)}$ after mapping of the SNR to [0,1] using $\alpha = 0.2$ and $\beta = 4dB$

Figure 2.3: Output at the first three steps of the processing of the soft mask for the spoken digit *six* corrupted by car noise at a global SNR of 5dB. The x-axis corresponds to time and the y-axis to the Mel channel. While the dynamic ranges of these plots differ, blue always corresponds to a value at the lowest-end of the range while red is at the highest-end of the range



(a) Soft-mask $M_{i,m}^{(2)}$ after applying a 3×5 median filter



(b) Final soft-mask $M_{i,m}^{final}$ after smoothing

Figure 2.4: After 2-d median filtering (a) and smoothing (b), the soft mask is not as sensitive as before to the noise fluctuations

now grouped together in the time-frequency domain. The second step aims at smoothing the rather sharp and piecewise-constant decision regions created by the median filter. Spatial averaging is performed with a constant disk of radius 2. This smoothing serves the same purpose of noise fluctuation cancellation as the median filter, but acts in a complementary way, by outputting a smooth, yet well-segmented soft-decision mask $M_{i,m}^{final}$ (see Fig. 2.4b).

Filter parameters have been optimized empirically, and will depend on the frame rate, window size and type, as well as on the number of Mel channels. In our experiments, we used a Hamming window of length 25ms that we shift by 10ms between subsequent frames. The number of Mel channels used is 32.

2.1.3 Naive approach to Noise Estimation

In this section, we propose a naive method to obtain the smooth noise power spectral density estimate $\tilde{\sigma}_N^2(i,m)$ that is used to compute the mask.

Let us assume that the noise is reasonably stationary and exhibits constant power through the whole utterance. If we also make the assumption that the first few milliseconds are composed solely of noise, then we can obtain a reliable estimate for $\tilde{\sigma}_N^2(i,m)$ by taking an average of the observed power $|Y(i,m)|^2$ over the first few frames. Formally, we derive $\tilde{\sigma}_{naive}^2(i,m)$ by:

$$\tilde{\sigma}_{naive}^2(i,m) = \frac{1}{2L}\left[\sum_{i<L}|Y(i,m)|^2 + \sum_{i>T-L+1}|Y(i,m)|^2\right]$$

where L will depend on the available data. In practice, a choice of L from 10 to 15 proves to be enough to reliably estimate the noise power. With the current window length and shift, this corresponds to a non speech segment of length about 175ms.

As stated above, this method will fail as soon as the noise power fades, rises or is non-stationary in nature, but the technique provides a good oracle estimate when the noise is fairly stationary and a few non-speech frames are available at both ends of the recording.

2.1.4 Adaptive Mask Estimation via Minimum Statistics Noise Power Tracking

In this section, we propose a strategy to overcome the rather limiting stationarity assumptions of the noise estimate introduced in Section 2.1.3.

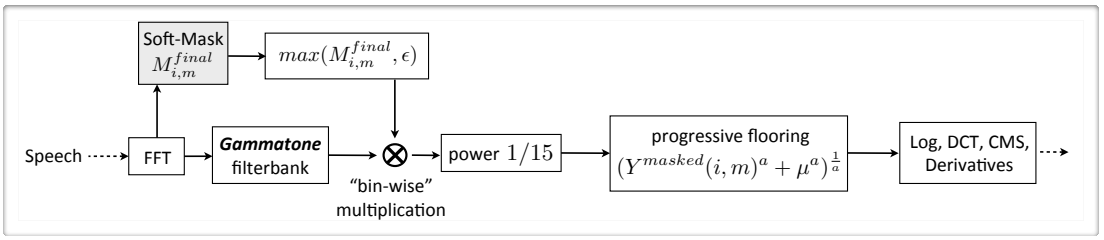
To this end, we use the Minimum Statistics-based (MS) noise power spectral density estimator from [34]. The MS algorithm tracks the minima values of a

smoothed power estimate of the noisy signal. By compensating the inherent bias, it derives an estimate of the noise PSD. Even though many subsequent authors have claimed to obtain more reliable noise estimators, a recent study ([35]) has shown that [34] still stands among the best algorithms across many noise conditions, especially in low-SNR and when facing varying noise. An implementation for this algorithm is available through the Voicebox Toolkit for Matlab [36] and allows a smooth integration within our framework via a routine called *estnoisem*.

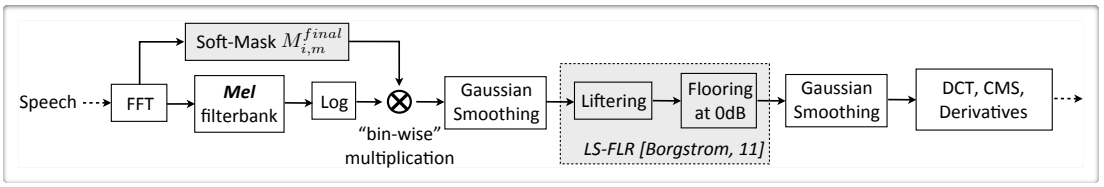
This function takes the un-warped spectrum $|Y(i,k)|^2$ as an input, and outputs an estimate $\tilde{\sigma}^2_{MS}(i,k)$ of the noise power in the linear frequency domain. Because this estimate has a tendency to produce outliers at the beginning of the speech segment as well as sharp transitions in the estimated noise magnitude across time, we found it useful to smooth the output by applying a median filter of length 50 frames, equivalent to approximately 0.5 sec. The resulting noise power estimate $\tilde{\sigma}^2_{MS,med}(i,k)$ has the desired time-smoothness of $\sigma^2_N(i,k)$ and automatically tracks varying powers of noise.

In order to obtain a noise estimate in the Mel-domain, we apply the Mel-filterbank to the previous estimate to obtain $\tilde{\sigma}^2_{MS}(i,m)$. After comparison with the oracle noise estimate derived in Section 2.1.3 for stationary noises, we noticed that the noise power was over-estimated by a factor of 2.5 to 4 on average. This bias is compensated for by multiplying the latter estimate by a factor of 0.36 before computing the SNR. We believe that this bias originates from the differences between the naive estimation that is performed directly in the Mel-domain and the current estimation that is performed in the linear domain before being filtered into the Mel-domain. This procedure is necessary though, as the algorithm from [34] is not designed to operate in the Mel-domain directly.

After this final bias correction, we obtain a noise estimate that we will refer



(a) Feature extraction using SMF_{pow}



(b) Feature extraction using SMF_{log}

Figure 2.5: Flowcharts for the proposed SMF_{pow} and SMF_{log} algorithms

to as $\tilde{\sigma}^2_{MS,\,unbiased}(i,m)$. This estimate has the double advantage that it does not require an oracle knowledge of speech presence and can adapt to varying powers of noise, as opposed to $\tilde{\sigma}^2_{naive}(i,m)$ that was derived in Section 2.1.3.

2.2 Spectral Imputation

This section introduces two Spectral Masking and Flooring (SMF) imputation techniques called SMF_{pow} and SMF_{log}. These techniques extracts information from the latter soft-mask in order to enhance the observed speech spectrum. Both techniques use the soft-mask as a set of multiplicative coefficients, but while SMF_{pow} performs enhancement on the power spectrum, SMF_{log} is designed to work on the log-spectrum. Since they rely on similar ideas (spectro-temporal weighting, dynamic range matching and noise variance cancellation) and only differ in the specifics of their implementation, the two algorithms will be presented in parallel. Flowcharts are shown in Figure 2.5.

2.2.1 Imputation via Spectrum Weighting

In this section, we motivate and describe the idea at the core of our imputation method: using the soft-mask as a set of multiplicative coefficients to discard the corrupted parts of the spectrum.

The proposed algorithm is based on the idea that, if the soft mask already contains some information based on the spectro-temporal correlation of speech, then the imputation can be made quite easily. Using such a soft mask, the filled-in values of *unreliable* bins could originate from a weighted value of the original *unreliable* bin. In other words, the neighboring bins to an *unreliable* one help decide what proportion of its power should be retained. For instance, suppose a bin tagged as *unreliable* has many neighbors tagged as *reliable*. Then, because speech is known to be so highly time-frequency correlated, we might consider using a fraction of the noisy value of that bin instead of setting it to zero or to some interpolated value from the neighboring *reliable* bins, as done in traditional imputation techniques introduced in Chapter 1 [37, 30, 13]. With this in mind, we propose the two following algorithms for imputation:

- Weighting the observed spectral power $|Y(i,m)|^2$ by its corresponding soft decision mask $M_{i,m}^{final}$. We refer to this algorithm as SMF_{pow}.
- Weighting the log spectral power $Y_{log}(i,m) = \log\left(|Y(i,m)|^2\right)$ by the soft decision mask $M_{i,m}^{final}$. We refer to this algorithm as SMF_{log}.

While it is not included in Figure 2.5, SMF_{pow} and SMF_{log} include a normalization step at the utterance level. For SMF_{log}, the utterance is normalized by the maximum of the time waveform. For the SMF_{pow} setup, the normalization follows the normalization from the PNCC framework [9]: after Gammatone filtering, the spectral power energy is computed on a frame basis, and the whole

spectrogram is normalized by 10^{15} times the 95^{th} percentile of this frame-by-frame power. Both of those normalization allow the front-end parameters to be independent of the frame energy.

As can be seen from Figure 2.5, the SMF_{pow} system is designed with a Gammatone filterbank and a 1/15 power compression while SMF_{log} is used in combination with a Mel-filterbank and a logarithmic compression. This choice is primarily motivated by the fact that we wanted to try our enhancement technique both in the case of the MFCC framework — which use the Mel-filterbank and log-compression — and with the recently introduced PNCC framework [9], which shows good performance using a Gammatone filterbank combined with root compression. Out of the four possible combinations, we present the two that have shown to perform best in our evaluations, namely:

- SMF_{log} with the MFCC framework: Mel filters and log
- SMF_{pow} with the PNCC framework: Gammatone filters and $(\cdot)^{1/15}$

Finally, since weighting the power spectrum and the log-spectrum is such a different operation in nature, SMF_{pow} and SMF_{log} might have very different optimal soft-mask parameters α and β.

2.2.2 Dynamic Range Matching

The proposed approach of discarding the likely non-speech components of the spectrum efficiently removes most of the corruptive noise energy. Yet, both methods create artifacts in the non-speech regions of clean and high-SNR spectra. This section describes the techniques that were used in both SMF_{pow} and SMF_{log} to obtain matching dynamic ranges across SNRs.

In the case of SMF_{log}, the log-spectrum of clean speech after mask weighting

exhibits a higher dynamic range than the enhanced log-spectrum of noisy speech. This is primarily due to the components with negative values, representative of low-energy speech and silence. Such negative values are seldom observed in the enhanced log-spectrum because the additive noise makes the observed signal inherently higher in energy. Moreover, the few remaining low energy regions will likely be tagged as *unreliable* and weighted with coefficients close to 0. Since speech recognition statistical models are generally trained using clean speech segments, this dynamic range mismatch between clean-speech and noisy-speech spectra might harm recognition accuracy. To alleviate this issue, we use a technique called Log-Spectral Flooring (LS-FLR), that was recently introduced in [13] to provide a lower bound on the reconstructed spectral energy of *unreliable* bins. In LS-FLR, we compute the liftered log-spectrum and set a flooring threshold, empirically optimized at 0dB. Formally, the liftered log-spectrum is obtained by:

$$\widetilde{Y}_{log} = \mathcal{C}^{-1}\mathcal{L}_{cep} \otimes \mathcal{C}\left[M_{i,m}^{final} \otimes Y_{log}\right]$$

where $\mathcal{C}$ is the discrete Cosine transform operator , $\mathcal{L}_{cep}$ is the cepstral lifter and $\otimes$ represent the element-by-element multiplication for vectors or matrices. The liftering step, equivalent to applying a band-pass filter in the frequency domain, tends to enhance the contrasts between the spectral peaks and valleys. Then, the flooring step builds upon the common belief that spectral valleys carry little discriminative energy while potentially resulting in unbounded values for $\widetilde{Y}_{log}$. As suggested in [13], we define the floored liftered spectrum as

$$\widetilde{Y}_{log}^{fl} = \max(\delta_{log}^{fl}, \widetilde{Y}_{log})$$

where the flooring threshold δ_{log}^{fl} is empirically set to 0dB. The latter processing helps matching the dynamic range of clean-speech and noisy-speech log-spectra while relying on the fact that discriminative information for ASR is more likely

to be found in the peaks of the spectrum than in the valleys. Indeed, a clean-speech silence whose power was $-3dB$ after masking will be floored up to 0 while a noisy-speech silence with high initial energy will be reduced down to 0 by the mask. Values of $\widetilde{Y}_{log}^{fl}$ lying significantly above 0 are expected to originate from speech energy only, which thereby makes the pattern matching easier for the recognition engine.

In the case of SMF_{pow}, similar problems are encountered. After applying the $(\cdot)^{1/15}$ operator on the weighted power spectrum, we obtain the compressed weighted power spectrum

$$Y_{pow}(i,m) = \left(M_{i,m}^{final} \otimes Y\right)^{1/15}.$$

This $(\cdot)^{1/15}$ operator has the neat property that it maps the bin values to a minimum of $0dB$, eliminating the inconvenience of very low energy bins that become highly negative after taking the log. Yet, a mismatch remains between clean-speech and noisy-speech spectra in low-amplitude regions. An additional problem is that the $(\cdot)^{1/15}$ operator exhibits a sharp transition from 0 to 1 for bins with very low energy. Such a transition will create high-amplitude fluctuations in the low-energy parts of the spectrum and therefore, should be avoided. A progressive flooring technique called Small Power Boosting, that aims at solving this dynamic range mismatch was first introduced in [38]. Inspired by this technique, we define:

$$Y_{pow}^{fl}(i,m) = \left[\left(Y_{pow}(i,m)\right)^a + \left(\delta_{pow}^{fl}\right)^a\right]^{1/a}$$

where δ_{pow}^{fl} is the power floor and a is a parameter representing the smoothness of the mapping. This mapping function is almost constant for powers significantly below δ_{pow}^{fl} and exhibits a linear behavior at powers significantly above δ_{pow}^{fl}. By setting the appropriate values for δ_{pow}^{fl} and a, this soft-flooring helps solve the two dynamic range issues outlined above.

2.2.3 Noise Variance Cancellation

According to the additive noise model presented in Section 2.1.1, the noise power equally corrupts the high and low energy parts of the speech power spectrum $|S(i,m)|^2$. The proposed spectral weighting techniques aimed at reducing the noise effect in the *unreliable* bins of the spectrum, by doing some noise cancellation. However, the high energy parts have also been corrupted, but since the mask weights at these *reliable* bins was hopefully all 1's, such distortions have not been accounted for in the above. In order to prevent from a spectral mismatch of clean-speech and noisy-speech spectral peaks, this section introduces a simple additional smoothing step.

In the case of SMF_{log}, we perform smoothing with a two-dimensional low-pass Gaussian filter of size 5×5 and standard deviation of $\sigma = 0.7$ bins. This rather sharp filter helps remove the remaining noise variability on the parts of the log-spectrum that have been preserved by the mask multiplication step. As can be seen in Figure 2.5b, the smoothing is done twice: once right after multiplication by the mask, to avoid this variance to be enhanced by the liftering step and once right after the flooring, to smooth the liftered spectrum. In the case of SMF_{pow}, we perform smoothing on Y_{pow}^{fl} using a similar filter.

2.3 Summary

In this chapter, we introduced a SNR-based soft-mask estimation technique along with two spectral imputation algorithms, SMF_{log} and SMF_{pow}, designed to attenuate the distortions caused by additive background noise, with applications to noise robust ASR.

The proposed soft-mask estimation technique is based on modeling the dis-

tortions of the noise as additive in the power spectral domain. In order to perform noise estimation, the minimum statistics-based algorithm from [34] is used for its robustness against non-stationary noise conditions, when compared to a naive noise estimator based on power averaging. This noise estimate provides a SNR-based soft-mask that is enhanced by two-dimensional median filtering and smoothing to cancel the noise fluctuations.

The proposed SMF_{pow} algorithm uses this mask as a set of multiplicative coefficients on the power spectrum, while the proposed SMF_{log} algorithm proceeds with mask weighting using the log-spectrum. Both algorithms implement an additional flooring step to better match the dynamic ranges of clean and noisy spectra. Lastly, the SMF_{log} algorithm is enhanced with power smoothing to cancel the remaining noise variance on the spectral peaks. The influence of mask weighting using the power or the log-power domain on speech recognition accuracy will be discussed in Chapter 3.

CHAPTER 3

Experimental Validation

In this chapter, the proposed approaches to feature extraction are evaluated in two different Automatic Speech Recognition (ASR) setups. In the first part, we introduce the two selected databases along with the back-end configuration for each setup. In the second part of this chapter, we present and discuss our experimental findings, and compare the performances of the proposed technique versus state-of-art algorithms.

3.1 Data and Experimental Setup

In this section, two speech recognition tasks are presented for evaluation of the SMF_{log} and SMF_{pow} techniques. We describe the databases, noise conditions and back-end configurations for each case.

3.1.1 Small Vocabulary ASR

The first task that has been selected is the Aurora-2 noisy digit recognition task [39]. Since its release in 2000, this database has been widely reported in the ASR literature when evaluating noise robust algorithms.

The Aurora-2 database consists of utterances of connected spoken digits artificially corrupted by background noise. The noise samples have been recorded in

the following real-life environments: subway, babble, car, exhibition hall, restaurant, street, airport and train station. All speech signals have been downsampled to 8kHz, and pre-filtered to conform to the standardized IUT-T G.712 frequency characteristics for telephone speech, with a passband between 300Hz and 3400Hz.

The training set consists of 8440 utterances containing recordings from 55 male and 55 female adult speakers. These noise-free utterances are used to train the ASR statistical models. The original test set consists of 4 subsets of 1001 utterances containing recordings from 52 male and 52 female speakers not seen in the training set. For each type of noise, one of these subsets is corrupted with noise added at 5 different SNR conditions (20dB, 15dB, 10dB, 5dB, 0dB). Since there are 8 noise types, we use a total of $8 \times 5 \times 1001$ utterances for testing.

The recognition engine is implemented using the Hidden Markov Models Toolkit (HTK) software package v3.4.1 [40]. The Aurora-2 scripts are used to prepare speech signals for processing as well as to train and test the ASR system. The HMM-based recognizer is configured with 11 single-word models: the digits "oh" and "zero" to "nine", plus two silence models "sil" and "sp". Each digit is modeled with a 16-states HMM with a 3-mixture Gaussian Mixture Model (GMM) per state. The silence "sil" models the pauses at the beginning and the end of an utterance and is modeled by a 3-state HMM with 6 Gaussian Mixtures per state. The silence model "sp" stands for "short-pause" and models the short silences between words. It is modeled by a 1-state HMM tied to the center state of the "sil" HMM.

Finally, the language model used in this experiment is shown in Figure 3.1. It constrains the ASR output to be an arbitrary long sequence of digits, and makes it possible to have a "sil" at the beginning and the end, and a "sp" after each digit.

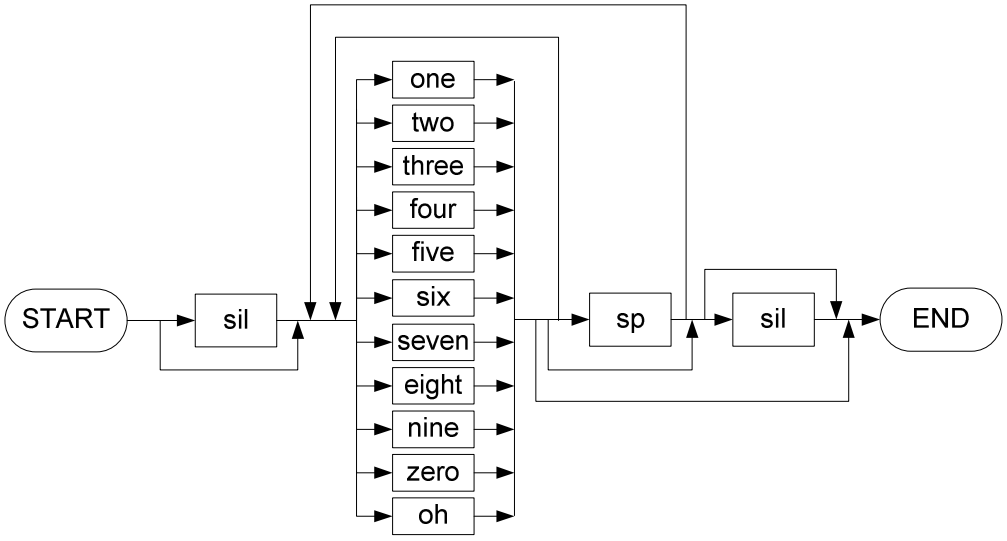


Figure 3.1: Language Model used for the Aurora-2 experiment. *Figure from* [1]

3.1.2 Large Vocabulary ASR

The second task that has been selected is the large vocabulary continuous speech recognition (LVCSR) Aurora-4 task[1], as described in [2]. Because it is a large vocabulary database with various noise conditions, experiments on Aurora-4 have been increasingly reported in the noise robust ASR literature in the last few years.

The Aurora-4 database was created as part of an effort of the Aurora Working Group from the European Telecommunications Standards Institute (ETSI) to evaluate the robustness of different front-ends for LVCSR in noisy conditions. It is constructed as a subset of the DARPA Wall Street Journal (WSJ0) Corpus and features a 5000-word vocabulary for both the training and the testing set. Speech signals used for these experiments were recorded by a head-mounted Sennheiser HMD-414 close-talking microphone providing a good quality signal. A second set

[1]While the Aurora-4 task is sometimes refered to as Large-Vocabulary in the ASR litterature, 5000 words is at the lower end of LVCSR as some tasks use vocabularies with more than 60,000 words.

of recordings from lower-quality microphones also exist for this database but was not used in this study since this set introduces distortions that do not fall into our additive noise model. Speech signals are down-sampled to 8kHz and noise is digitally added at SNRs from 5 to 15dB. The noise types are similar to those used in the Aurora-2 database, with 6 different environments: street traffic, train station, car, babble, restaurant and airport.

The training set contains 7,138 clean utterances from 83 speakers, totaling 14 hours of speech data. These recordings are obtained from speakers reading articles from the Wall Street Journal, and fall within a vocabulary of 5000 words. The clean test set contains 330 utterances from 8 speakers, recorded using the same microphone as in training, and without any Out of Vocabulary (OOV) words. Using this clean set, six noisy test sets were created by separately adding each of the 6 noise types at randomly chosen SNRs between 5 and 15 dB. The total number of utterances used for testing is thus 7×330, with an average SNR level of 10dB and an equal representation for each noise type.

The recognition engine is implemented using HTK [40], as in the case of Aurora-2. The scripts from [41] for the WSJ0 database were used for training and testing. The HMM-based recognizer is configured using context-dependent word-internal triphone models. Each triphone is modeled by a 3-state HMM as shown in Fig. 3.2 (a). As in Aurora-2, potentially long silences are modeled by the HMM "sil" whereas short pauses accounting for inter-words silences are modeled by the HMM "sp", as shown in Fig. 3.2 (b) and (c). Each state of these HMMs initially contains a single mixture GMM, that is increased to up to 16 mixtures during training. The pronunciations for each of the words were obtained using the publicly available CMU dictionary [42]. Training of the triphone models was performed as follows:

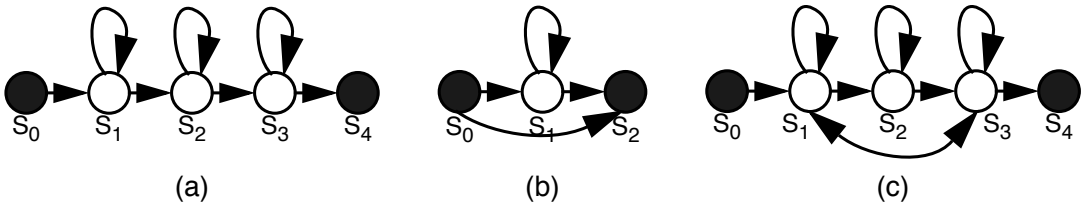


Figure 3.2: HMM models used for acoustic modeling for the Aurora-4 task: (a) typical triphone, (b) short pause, and (c) silence. The shaded states denote the start and stop states for each model. *Figure from* [2]

1. Learn flat start monophones with single mixture Gaussians using the Baum-Welch reestimation algorithm. The Gaussian means and variances are initially set to the average mean and variance over a subset of training data.

2. Use the above models and the Viterbi algorithm to force-align the phone-level transcriptions to the acoustic data.

3. Learn the monophones on the training set with aligned labels using the Baum-Welch algorithm.

4. Combine the monophone models into word-internal triphone models. Reestimate their parameters using the Baum-Welch algorithm.

5. Tie the states of some triphones using decision tree clustering to alleviate the lack of training data and improve the reliability of the models. Tied states share the same Gaussian mixtures parameters.

6. Progressively increase the number of mixtures from 1 to 2, 4, 8 and 16 by splitting existing mixtures at each step, and reestimating their parameters with the Baum-Welch algorithm.

Finally, we use the WSJ standard 5K non-verbalized closed bigram Language Model (LM). A bigram language model provides the probabilities of occurrence for each triphone conditioned on all pairs of past triphones. The insertion penalty, that limits short word insertions in place of noise, is set to -4 while the LM scale factor, that balances the importance of the LM over the acoustic models, is set to 15.

3.2 Results and Discussion

In this section, we present and discuss the evaluation results of the selected ASR tasks using the front-end algorithms SMF_{pow} and SMF_{log} proposed in Chapter 2. In a first part, we present the setups that gave the best recognition results on each task. In the subsequent parts, we discuss the influence of various parameters in recognition performance. Lastly, we compare the performance and the complexity of the proposed method to state-of-the-art noise robust front-end algorithms.

3.2.1 Main Results on Aurora-2

In this subsection, we present the performances for SMF_{pow} and SMF_{log} on the Aurora-2 task and discuss the contribution of the different steps to the overall improvement in accuracy over the baselines. We will present the results as we obtained them chronologically, namely SMF_{log} and then SMF_{pow}.

The SMF_{log} front-end was first evaluated on the Aurora-2 task, with the parameters mentioned in Chapter 2: $\alpha = 2$, $\beta = 4\,dB$ and $\delta_{log}^{fl} = 0\,dB$. Table 3.1 shows the accuracy on the Aurora-2 task for SMF_{log} under the following configurations:

- Mel-Filtered Cepstral Coefficients (MFCC baseline)

SNR (dB)	20	15	10	5	0	**Avg.**
MFCC	97.6	93.6	78.7	45.8	11.9	**65.5**
SMF_{log} (no Masking, Smoothing)	97.5	94.7	86.2	64.8	29.5	**74.5**
SMF_{log} (no Flooring)	96.9	93.7	86.6	71.0	43.4	**78.3**
SMF_{log} (no Masking)	97.6	94.7	86.6	66.7	33.7	**75.8**
SMF_{log}	98.4	97.1	93.7	83.1	58.6	**86.2**

Table 3.1: Percent Word-Accuracies per SNR on Aurora-2 for the SMF_{log} algorithm

- SMF_{log} with no Masking and Smoothing (MFCC + LS-FLR)
- SMF_{log} with no Flooring (MFCC + Masking + Liftering + Smoothing)
- SMF_{log} with no Masking (MFCC + LS-FLR + Smoothing)
- SMF_{log} as in Chapter 2 (MFCC + Masking + LS-FLR + Smoothing)

Improvements in accuracy with the SMF_{log} front-end are observed at all SNRs over the non-robust MFCC baseline. It is also interesting to note the contributions of various components of the proposed algorithm to this gain in performance. Comparison between the last two rows show that the masking step alone accounts for about 50% of the improvements. The prominent role of this step was expected as masking removes most of the noise distortions. Comparison between rows 3 and 5 show the essential role of the flooring step, as a tool to adjust the spectrum's dynamic range that accounts for about 40% of the gain. Finally, the difference in accuracy between rows 2 and 4 shows that the smoothing step plays a small but significant role in improving the accuracy, by about 1.3% on average.

The SMF_{pow} front-end technique was also evaluated on the Aurora-2 setup, with the following parameters: $\alpha = 0.5$, $\beta = 6\,dB$, $\delta_{log}^{fl} = 4$ and $a = 5$. Note

SNR (dB)	20	15	10	5	0	**Avg.**
MFCC	97.6	93.6	78.7	45.8	11.9	**65.5**
SMF_{pow} (no Masking/Flooring)	98.0	95.6	87.9	66.9	33.3	**76.3**
SMF_{pow}	96.9	92.2	82.5	64.5	37.6	**74.7**

Table 3.2: Percent Word-Accuracies per SNR on Aurora-2 for the SMF_{pow} algorithm

that these parameters were optimized for the large-vocabulary Aurora-4 task, but results are also shown on the Aurora-2 task for consistency. Word-Accuracies are shown in Table 3.2 for the following configurations:

- MFCC baseline
- SMF_{pow} with no masking and flooring, equivalent to PNCC without Power Bias Subtraction (PBS). This serves as a second baseline, more robust as MFCC, that SMF_{pow} should improve on.
- SMF_{pow} as in Chapter 2

Surprisingly, the proposed setup SMF_{pow} performs worse than the un-enhanced baseline shown in the second row. Yet, a more careful analysis shows that the two algorithms have different error patterns. Table 3.3 details the performances of both algorithms under the following two error measures:

- The Word-Accuracy (%) accounts for all three types of errors : Insertion, Deletion and Substitution. It is defined by:

$$Acc = \frac{\#Words - (\#Del + \#Sub + \#Ins)}{\#Words}$$

- The Word-Correct (%) does not take into account the insertions. It it is

Noise type	Sub.	Bab.	Car	Exh.	Rest.	Str.	Airp.	Train	**Avg.**
Word-Accuracy (%)									
SMF_{pow} no Mask/Floor	68.5	65.3	67.2	60.4	64.5	69	71.2	69.3	**66.9**
SMF_{pow}	65.1	53.6	80.0	58.8	52.0	68.0	65.3	73.3	**64.5**
Word-Correct (%)									
SMF_{pow} no Mask/Floor	69.1	70.3	67.4	62.6	72.1	70.8	74.9	72.7	**70**
SMF_{pow}	78.4	74.5	84.2	71.4	72.2	79.3	80.7	83.1	**78.0**

Table 3.3: Percent Word-Accuracies and Word-Correct at 5dB SNR, on Aurora-2

defined by:

$$Cor = \frac{\#Words - (\#Del + \#Sub)}{\#Words}$$

From Table 3.3, we can see that SMF_{pow} has a systematically higher Word-Correct than the baseline without masking or flooring. This means that masking and flooring can correct many deletion and substitution errors but also introduce many insertions, mostly from the short digits *'eight'* and *'oh'*. These insertion errors greatly impact the overall Word-Accuracy previously shown in Tables 3.2 and 3.3. Such errors were not observed in the experiments with SMF_{log}, for which masking and flooring were shown to have a positive impact on Word-Accuracy. Since masking on the spectrum and on the log-spectrum are different operations, our guess is that masking using the log-spectrum removed more noise components that might not have been removed by simply masking using the power spectrum. This in turn could result in more insertions for SMF_{pow}, due to the statistical engine misclassifying noise chunks as short digits. We also think that these errors

Noise Type	Clean	Airp.	Babble	Car	Rest.	Street	Train	**Avg.**
MFCC	90.2	55.5	49.4	72.7	51.4	40.8	40.9	**51.8**
SMF_{log}	81.0	54.8	58.9	72.41	56.2	58.7	59.3	**60.0**
SMF_{pow} no Mask/Floor	88.4	56.8	56.5	78.5	55.6	54.5	52.9	**59.1**
SMF_{pow} no Masking	88.9	56.9	56.6	81.2	54.8	56.2	54.7	**60.1**
SMF_{pow} no Flooring	88.9	56.9	60.8	82.6	55.6	61.1	63.0	**63.3**
SMF_{pow}	88.4	58.1	61.3	84.4	56.7	64.4	65.1	**65.0**

Table 3.4: Percent Word-Accuracies for SMF_{log} and SMF_{pow} on Aurora-4

are amplified by the rather weak language model used on the Aurora-2 task. As we will show in the next subsection, such insertion errors using SMF_{pow} are no longer prominent on the Aurora-4 task, where we use a stronger bigram language model.

3.2.2 Main Results on Aurora-4

In this subsection, we present the performances for SMF_{pow} and SMF_{log} on the Aurora-4 task and discuss the contribution of the different steps to the overall improvement in accuracy over the baselines. For consistency, these experiments have been run with the same parameters previously used on the Aurora-2 setup. Results are shown in Table 3.4.

First, a few comments can be made on the SMF_{log} versus the MFCC setup. On clean data, the accuracy of the proposed SMF_{log} technique falls 10% below

the accuracy of the baseline MFCC. This large accuracy loss was not previously observed on the limited vocabulary Aurora-2 task. We believe that denoising by masking on the log-spectrum, combined with an imperfect estimated mask and optimizing the parameters on a small vocabulary setup lead to a rather strong noise suppression that might have eroded low energy speech components, even in clean conditions where the masking effect should have been rather weak. While this low-energy speech information may not be needed to discriminate 11 digits, it is essential when dealing with a large vocabulary task like in Aurora-4, and explains why such an accuracy drop is observed. On noisy utterances however, the proposed SMF_{log} technique leads to an 8% average improvement over MFCC. Such improvements are obtained despite the observed bad performance on clean speech, and confirms that the proposed masking and flooring technique is indeed helpful in attenuating the noise distortions.

The SMF_{pow} method, in contrast to SMF_{log}, proposes a more forgiving approach to noise suppression that will hopefully improve the performances in both clean and noise conditions. The following approaches are evaluated, with gradual enhancements:

- SMF_{pow} without Masking and Flooring, equivalent to PNCC without Power Bias Subtraction (PBS). This provides a non-robust baseline similar in essence to MFCC.
- SMF_{pow} without Masking, equivalent to PNCC no PBS + Flooring.
- SMF_{pow} without Flooring, equivalent to PNCC no PBS + Masking.
- SMF_{pow} with all the steps presented in Chapter 2, equivalent to PNCC no PBS + Masking + Flooring.

First, we observe that the two non-robust baselines, MFCC and SMF_{pow} without masking and flooring (equivalent to PNCC without PBS), have similar performance when tested on clean data, with recognition accuracy of 90.2% and 88.4% respectively. In noise though, SMF_{pow} without masking and flooring outperforms MFCC by more than 7% on average across noise types. This motivates our choice to base SMF_{pow} on the PNCC framework (Gammatone filterbank, power compression) instead of the MFCC framework (Mel filterbank, log compression) which seems less noise robust in the present large vocabulary scenario. Further addition of masking and flooring on top of the PNCC no PBS baseline improves the accuracy in noise by 4% and 1% respectively, and in clean by 0.5% for both cases. Lastly, combining masking with flooring lead to a total absolute improvement of 6% in noise with no change in clean, well above the performances of SMF_{log}. As was suggested in Section 3.2.1, we believe that this performance gap is due to a lighter noise suppression coupled with a stronger language model that avoids errors due to short insertions.

3.2.3 Influence of the Noise Estimation Technique

In all previously reported results, we used the adaptive noise estimate from [34] to compute our SNR-based mask. In Chapter 2, we also presented a more naive technique for noise estimation based on a channel-wise time-averaging of the noise power over a few frames, at the beginning and the end of the utterance. In this subsection, we present experiments to assess the effect on the recognition accuracy when choosing either method.

Table 3.5 presents the performances obtained with the SMF_{log} algorithm on the Aurora-2 task, when noise estimation is performed in the naive fashion and with the adaptive estimate from [34], as previously presented in Section

Noise type	Sub.	Babble	Car	Exhib.	Rest.	Street	Airp.	Train	**Avg.**
SMF_{log} (no Masking)	77.7	72.7	78.1	72.0	72.1	78.0	78.1	77.9	**75.8**
SMF_{log} (naive N. Est.)	83.7	84.8	88.4	86.6	82.3	83.0	87.1	87.1	**85.4**
SMF_{log}	83.7	84.3	89.1	87.2	82.9	87.4	87.6	87.3	**86.2**

Table 3.5: Percent Word-Accuracies for SMF_{log} on Aurora-2, with various noise estimation techniques

3.2.1. What we observe is that the adaptive noise estimate, besides making no hypothesis about silence in the first and last few frames as well as about the noise stationarity, actually performs better on average than the naive noise estimate. When looking at the improvements broken up by noise type, we notice that the absolute improvements lie within 0 to 1% for all noises, except for the street noise where an absolute 4.4% improvement in achieved. The street noise has been recorded on a busy street, and is particularly non-stationary. As shown on Figure 3.3, the short-time power of street noise varies sharply, with swings of more than 20dB across intervals of a few seconds. In these environmental conditions, the stationarity hypothesis of the naive noise estimator is violated and obtaining a time-varying estimate becomes essential to properly compute the mask, and thus remove the right amount of noise from the signal. Such an estimate is used in the proposed SMF_{log} algorithm, with a beneficial effect on recognition accuracy.

3.2.4 Influence of Mask Weighting on the Training Data

For both SMF_{log} and SMF_{pow}, no difference in processing is made when handling clean and noisy data. While masking makes sense when dealing with noisy data,

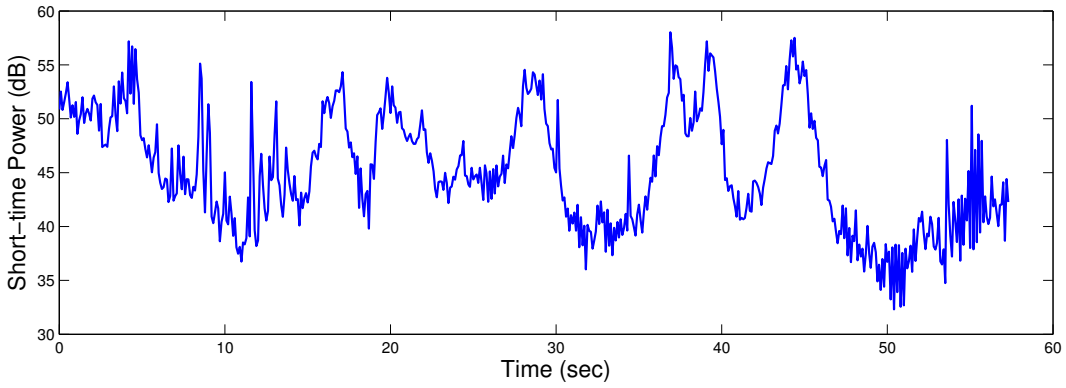


Figure 3.3: The short-time spectral power of pure street noise is highly non-stationary

it is not clear whether doing so when dealing with clean data would not distort the original spectrum. This is a concern since the noise estimated on a clean utterance might very well be non-zero due to the imperfect behavior of [34] at very high SNRs. An erroneous noise estimate could in turn lead to discarding some important parts of the clean speech spectrum, which would hurt recognition accuracy. Sections 3.2.1 and 3.2.2 already showed that no such degradation was observed for SMF_{log} on Aurora-2 and SMF_{pow} on Aurora-4, the best performing setups. Yet, one might wonder what would happen if no such masking was performed on data labelled as clean, in order to preserve the spectral information of speech. The choice to apply masking or not depending on the noise conditions is not so realistic on the testing utterances, since the environment is assumed to be unknown. The training utterances, on the other hand, are recorded in clean conditions and lends themselves well to bypassing the Mask weighting step. This subsection presents a set of experiments where no masking is performed on the data used for training, and compares the results to the original setup where no such clean/noisy distinction is made.

Table 3.6 shows results obtained on the Aurora-2 task for the SMF_{log} al-

Noise type	Sub.	Bab.	Car	Exhib.	Rest.	Str.	Airp.	Train	**Avg.**
Training and Testing with Masking									
SMF_{log} (naive N. Est.)	83.7	84.8	88.4	86.6	82.3	83.0	87.1	87.1	**85.4**
SMF_{log}	83.7	84.3	89.1	87.2	82.9	87.4	87.6	87.3	**86.2**
Training without Masking and Testing with Masking									
SMF_{log} (naive N. Est.)	84.9	86.3	88.3	86.7	84.8	84.5	87.9	87.5	**86.4**
SMF_{log}	83.5	84.3	88.9	86.8	82.5	87.2	87.4	87.1	**86.0**

Table 3.6: Influence of training the clean models with and without masking, on the Aurora-2 task. Percent Word-Accuracies are shown, averaged over 5-15 dB SNR

gorithm, both with the adaptive and the naive noise estimates. In the case of SMF_{log} with the adaptive estimate, no significant difference in accuracy is observed both across noise types and on average. This demonstrates that the adaptive noise estimation is good enough that using masking on the clean training utterances is not harmful to building discriminative word models. For SMF_{log} with the naive estimation method, the accuracy when no masking is performed for training is 1% higher on average than with masking. This trend is stronger for noises such as *street* or *restaurant*, confirming our doubts in the potential of the naive averaging technique to capture information from non-stationary environments.

Table 3.7 shows results obtained on the Aurora-4 task for the SMF_{log} and SMF_{pow} algorithms, both using the adaptive noise estimator. We observe a clear trend on both algorithms and across all noise types: training and testing with masking leads to improved results over a mismatched scenario where no masking

Noise type	Clean	Airp.	Babble	Car	Rest.	Street	Train	**Avg.**
Training and Testing with Masking								
SMF_{log}	81.0	54.8	58.9	72.41	56.2	58.7	59.3	**60.0**
SMF_{pow}	88.4	58.1	61.3	84.4	56.7	64.4	65.1	**65.0**
Training without Masking and Testing with Masking								
SMF_{log}	80.3	51.6	54.2	66.8	51.9	51.4	52.8	**54.8**
SMF_{pow}	87.6	54.2	58.7	82.1	53.1	64.2	64.6	**62.8**

Table 3.7: Influence of training the clean models with and without masking, on the Aurora-4 task. Percent Word-Accuracies are shown, averaged over 5-15 dB SNR

is performed on the training data. In the clean testing case, the gain is modest, but the reasons for this behavior are straightforward since doing a similar processing on both data will increase the feature similarity and thus the recognition accuracy. In the noisy case, we observe an absolute difference in accuracy of 6% for SMF_{log} and 2% for SMF_{pow}, higher than the previous 0.8% difference in clean for both SMF_{log} and SMF_{pow}, and than the 0.2% average difference in noise observed for SMF_{log} on Aurora-2 (Table 3.6). While the amplitude of these differences are most likely only artifacts of the masking technique and the recognition task, it becomes clear that applying the same processing on clean and noisy data consistently provides a gain in accuracy for the two proposed techniques, when using the adaptive noise estimation method. This is good news for another reason, because it means that we could use a single algorithm to deal with a mismatched and a multi-condition setup, where noisy utterances are also available in the training corpus.

Noise Type	Clean	Airp.	Babble	Car	Rest.	Street	Train	**Avg.**
Word-Accuracy (%) using SMF_{pow} with flooring exponent $a = 5$								
$\delta_{pow}^{fl} = 2$	89.3	57.2	60.8	83.4	55.7	61.3	62.1	**63.4**
$\delta_{pow}^{fl} = 3$	88.8	57.3	61.1	83.8	56.8	62.8	63.8	**64.3**
$\delta_{pow}^{fl} = 4$	88.4	58.1	61.3	84.4	56.7	64.4	65.1	**65.0**
$\delta_{pow}^{fl} = 5$	87.4	57.3	61.7	83.1	56.0	64.1	65.5	**64.6**
$\delta_{pow}^{fl} = 6$	86.4	57.2	61.0	82.1	56.5	64.9	65.7	**64.6**

Table 3.8: Influence of the Power Spectrum flooring parameter δ_{pow}^{fl} for SMF_{pow}, on the Aurora-4 task. Percent Word-Accuracies are shown, averaged over 5-15 dB SNR

3.2.5 Influence of the Flooring Parameters

This subsection explores the effect of various flooring parameters. Experiments will be presented for the SMF_{pow} algorithm on the Aurora-4 task, where the flooring is done using the soft-flooring function introduced in Section 2.2.2 as:

$$Y_{pow}^{fl}(i,m) = \left[(Y_{pow}(i,m))^a + \left(\delta_{pow}^{fl}\right)^a\right]^{1/a}$$

where δ_{pow}^{fl} is the power floor and a is an exponent that sets the smoothness of the mapping.

Table 3.8 shows the effect of increasing and decreasing the power floor on recognition accuracy, with a fixed flooring exponent set at $a = 5$. As expected, setting the floor too low ($\delta_{pow}^{fl} = 2$) decreases the word-accuracy as it increases the dynamic range of the spectral powers, thereby introducing more variability into the spectral valleys, which are known to carry little discriminative information. On the other hand, increasing the power floor too much ($\delta_{pow}^{fl} = 6$) tends to erase discriminative low-energy speech information and impact the recognition accuracy. A good middle ground for setting this parameter is found at $\delta_{pow}^{fl} = 4$.

Noise Type	Clean	Airp.	Babble	Car	Rest.	Street	Train	**Avg.**
Word-Accuracy (%) using SMF_{pow} with power floor $\delta_{pow}^{fl} = 4$								
$a = 3$	88.8	58.2	61.9	84.0	56.6	64.0	64.4	**64.8**
$a = 4$	88.2	58.0	61.5	83.6	56.5	64.7	65.3	**64.9**
$a = 5$	88.4	58.1	61.3	84.4	56.7	64.4	65.1	**65.0**
$a = 7$	88.5	56.6	59.7	83.6	55.5	62.8	63.6	**63.6**
$a = 9$	88.2	55.6	58.0	83.5	54.4	62.2	63.7	**62.9**

Table 3.9: Influence of the Power Spectrum flooring parameter a for SMF_{pow}, on the Aurora-4 task. Percent Word-Accuracies are shown, averaged over 5-15 dB SNR

Table 3.9 shows the effect of increasing and decreasing the flooring exponent on recognition accuracy, with a fixed power floor set at $\delta_{pow}^{fl} = 4$. When the exponent is low and gets close to $a = 1$, the mapping becomes more linear, weakly compressing the dynamic range while preserving some of the variability of the spectral valleys. As a increases, the mapping becomes similar to a hard thresholding of the type $Y_{pow}^{fl}(i, m) = \max(Y_{pow}(i, m), \delta_{pow}^{fl})$, where all the information about spectral powers less than δ_{pow}^{fl} is erased. The optimal tradeoff is found with an exponent value of $a = 5$.

3.2.6 Comparison versus State-of-the-art front-ends

As described in Chapter 1, many other algorithms have been previously introduced to perform front-end processing for noise robust speech recognition. In this subsection, we present a performance comparison of our methods to some of the state-of-the-art techniques for each task.

Table 3.10 shows a SNR-wise comparison of word-accuracies for the SMF_{log}

SNR (dB)	20	15	10	5	0	**Avg.**
MFCC	97.6	93.6	78.7	45.8	11.9	**65.5**
SMF_{log}	98.4	97.1	93.7	83.1	58.6	**86.2**
MFCC + MVA	97.9	96.1	91.6	81.0	59.2	**85.1**
PNCC	98.7	97.3	93.3	81.1	53.7	**84.8**
ETSI-AFE	98.1	96.7	92.8	83.2	59.8	**86.1**

Table 3.10: Comparison of Percent Word-Accuracies for several state-of-the-art techniques on Aurora-2

technique to algorithms reported as state-of-art on the Aurora-2 database:

- MFCC + Mean Variance ARMA filtering (MVA) as introduced in [8] is a low-complexity normalization scheme that reduces the variability between the clean and noisy MFCCs at the cepstrum level.
- Power Normalized Cepstral Coefficients (PNCC) as described in [9], with the Power Bias Subtraction (PBS) algorithm.
- ETSI-AFE as described in [43] is the current state-of-art for single-channel noise robust front-end processing on Aurora-2. It performs denoising via a two-stage Wiener filtering approach.

As we can see from this first set of experiments, the proposed SMF_{log} 's performances are in par with all the other techniques, at every SNR level. On average over all SNRs, the proposed technique does slightly better than the best performing algorithm ETSI-AFE. This validates our approach of masking and flooring to reduce the variability between features computed on clean and noisy utterances.

An evaluation of the computational cost of these techniques has been per-

Algorithm	Language	Running Time
MFCC	Matlab	30s
SMF_{log} (naive N. Est.)	Matlab	40s
SMF_{log} (adaptive N. Est.)	Matlab	150s
MFCC + MVA	Matlab	30s
PNCC	Matlab	1200s
ETSI-AFE	C	50s

Table 3.11: Running time of various state-of-the-art front-ends, for a feature extraction task of 1001 utterances from the test set of the Aurora-2 database

formed on 1001 utterances from the testing data. The scripts were evaluated in the programming language used by the authors, namely Matlab for all the algorithms but ETSI-AFE, which was written in C. Results displayed in Table 3.11 show that the good accuracy of the proposed method is achieved with a reasonable computational cost (150s), when compared to PNCC (1200s) or ETSI-AFE (50s in C). Also, we notice that replacing the adaptive noise estimation method by the naive averaging technique cuts the computational cost of our processing by a factor of 12, from +120s to only +10s on top of the traditional MFCC processing. This shows that the proposed masking and flooring techniques are computationally efficient, and that the biggest load in complexity is introduced by the external noise estimation algorithm from [34].

Table 3.12 shows the word-accuracy of the SMF_{log} and SMF_{pow} algorithms versus the previously introduced PNCC technique, on the Aurora-4 database. On this task, PNCC proves to perform better on average than the proposed SMF_{pow} approach, where we attempted to replace the PBS processing by our conceptually simpler masking and flooring steps. Yet, the proposed algorithm performs as well

Noise type	Clean	Airp.	Babble	Car	Rest.	Street	Train	**Avg.**
MFCC	90.2	55.5	49.4	72.7	51.4	40.8	40.9	**51.8**
SMF_{log}	81.0	54.8	58.9	72.41	56.2	58.7	59.3	**60.0**
SMF_{pow}	88.4	58.1	61.3	84.4	56.7	64.4	65.1	**65.0**
PNCC	88.0	67.2	68.3	83.4	64.3	66.7	66.4	**69.4**

Table 3.12: Percent Word-Accuracies for SMF_{pow} and SMF_{log} compared to PNCC on Aurora-4

as PNCC on 4 out of the 7 test conditions (clean, car, street and train). Only in the remaining three noise conditions (airport, babble and restaurant) is there a performance gap. These observations as well as the upper bound using an oracle mask obtained by [31] suggest that our mask estimation technique is not perfect for some noise types, and that obtaining a more accurate mask could improve recognition performance. In this sense, the proposed SMF_{pow} technique can be seen as a successful framework to apply direct masking on the power spectrum, for noise robust ASR front-end processing.

3.3 Summary

In this Chapter, we introduced two ASR setups for evaluation of the proposed algorithms: a small-vocabulary connected words recognition task (Aurora-2) and a large vocabulary continuous speech recognition task (Aurora-4). In both cases, we presented the characteristics of the speech data and noise conditions, as well as the back-end configurations with the acoustic and language models.

The experiments showed that the SMF_{log} algorithm performs better than SMF_{pow} and other state-of-the-art algorithms like ETSI-AFE or PNCC on the

Aurora-2 task. We have shown that masking and flooring account both for about 50% of the gain in performance over an un-enhanced baseline. On the Aurora-4 task, we found that the SMF_{pow} algorithm performs better than SMF_{log}, yet still worse than the performance of the PNCC technique on three out of six noise types. As suggested by [31], the present thesis confirms the importance of an accurate mask estimation to perform direct masking.

On both ASR setups, using a mask computed with the adaptive noise power estimator from [34] to process clean and noisy speech spectra, both in training and testing, lead to better results than when using a naive noise estimator.

Lastly, the influence of varying the flooring parameters of the SMF_{pow} algorithm was studied, confirming our interpretation of flooring as a tool to match the dynamic ranges of clean and noisy spectra while discarding the non-discriminative information carried by spectral valleys.

CHAPTER 4

Conclusion and Perspectives

In this thesis, we introduce two front-end algorithms for ASR: SMF_{log} and SMF_{pow}. These algorithms, presented in Chapter 2, attenuate the distortive effect of additive background noise by means of mask weighting and dynamic range matching. In doing so, we further elaborate on previous work on incorporating direct binary mask weighting to ASR by [31]. That work suggested that such an approach would only work with an ideal mask and on tasks with a strong language model. To asses the validity of our approach to mask weighting with an estimated mask, Chapter 3 presents an evaluation setup that includes ASR tasks with both a small vocabulary (Aurora-2) and a large vocabulary (Aurora-4). It was found that while SMF_{log} was more adapted to the Aurora-2 task, SMF_{pow} performed significantly better on the Aurora-4 task.

First, experiments performed on the Aurora-2 connected digit recognition task demonstrated that mask weighting combined with flooring could lead to state-of-the-art results, even with an estimated mask and no language model. Indeed, the SMF_{log} algorithm performed at least as well as ETSI-AFE, the best performing algorithm on Aurora-2. Contributions of the SMF_{log} algorithm include an improved soft-mask estimation algorithm integrating the adaptive noise estimator from [34], the idea to perform masking on the log-spectrum, the integration of Log-Spectral Flooring as an alternative to variance normalization, and the final spectral smoothing.

Further, experiments performed on the Aurora-4 continuous speech recognition task demonstrated that mask weighting and flooring with SMF_{pow} could approach the performance of state-of-the-art algorithms like PNCC in some noise conditions, thus replacing a costly processing with the proposed low-complexity procedure. This result is an improvement over [31], which suggested that only ideal masks could lead to such gains in accuracy. We believe that this gain is the combined result of our mask estimation technique with soft values in [0,1] and the proposed dynamic range matching technique using progressive flooring, as compared to the simple variance normalization procedure used in [31]. On the other hand, the limited accuracy gains observed in half of the noise types confirm the high dependability of mask weighting techniques on a reliable initial mask estimate.

These results show that techniques that perform severe denoising like SMF_{log} are better fitted to small vocabulary tasks like Aurora-2, where the language model cannot prevent a large amount of insertions errors. With a larger vocabulary task, like Aurora-4, insertions are more likely to be avoided thanks to a stronger bigram language model. On such a task, severe denoising techniques like SMF_{log} tend to erase discriminative speech information, thus impacting the performance of ASR. Therefore, smoother denoising techniques such as SMF_{pow} or PNCC have been shown to preserve more discriminative speech components, with a positive impact on word-accuracy.

Future investigations could study the potential of mask weighting approaches using more sophisticated mask estimation techniques, especially if they can account not only for background noise but also other effects such as reverberation or channel distortions. Also, since flooring has been shown to account for about 50% of the accuracy gains in the above experiments, a natural extension of this study

would look at the performance of flooring compared to variance normalization, as an alternative dynamic range matching technique when using ideal masks. If flooring performed better than variance normalization with ideal masks, then a new upper bound on the performance of mask weighting techniques would be found. Finally, better estimates of the power spectral density of non-stationary noises are needed, as an improved tracking of the noise would directly translate into a more precise mask estimate.

References

[1] Ngee Tan, L., *Voice Activity Detection using Harmonic Frequency Components in Likelihood Ratio Tests*, Master's thesis, University of California, Los Angeles, USA, 2010

[2] Parihar, N. and Picone, J., "DSR Front End LVCSR Evaluation - AU/384/02," *European Telecommunications Standards Institute*, 2002

[3] Rabiner, L.R. and Juang, B.H., *Fundamentals of Speech Recognition*, PTR Prentice Hall, ISBN 9780130151575, 1993

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

[5] Ephraim, Y. and Malah, D., "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," *IEEE Transactions on Acoustics Speech and Signal Processing*, volume 32, no. 6, pp. 1109–1121, 1984

[6] Ephraim, Y. and Malah, D., "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," *IEEE Transactions on Acoustics Speech and Signal Processing*, volume 33, no. 2, pp. 443–445, 1985

[7] Loizou, P.C., *Speech Enhancement: Theory and Practice*, Taylor and Francis, 2007

[8] Chen, C.P., Bilmes, J., and Kirchhoff, K., "Low-Resource Noise-Robust Feature Post-Processing On Aurora 2.0," *Proceedings of ICSLP*, pp. 2445–2448, 2002

[9] Kim, C. and Stern, R.M., "Feature extraction for robust Speech recognition based on maximizing the sharpness of the power distribution and on power flooring," *IEEE International Conference on Acoustics Speech and Signal Processing ICASSP*, pp. 4574 –4577, 2010

[10] Moore, B.C.J. and Glasberg, B.R., "A revision of Zwicker's loudness model," *Acustica united with Acta acustica*, volume 82, pp. 335–345, 1996

[11] Strope, B. and Alwan, A., "A model of dynamic auditory perception and its application to robust word recognition," *IEEE Transactions on Speech and Audio Processing*, volume 5, no. 5, pp. 451–464, 1997

[12] Zhu, Q., Iseli, M., Cui, X., and Alwan, A., "Noise Robust Feature Extraction for ASR using the Aurora 2 Database," *Proceedings of EuroSpeech*, pp. 185–188, 2001

[13] Borgstrom, B. J. and Alwan, A., "Missing Feature Imputation of Log-Spectral Data For Noise Robust ASR," *Workshop on DSP in Mobile and Vehicular Systems*, 2009

[14] Strope, B. and Alwan, A., "Robust Word Recognition using Threaded Spectral Peaks," *IEEE International Conference on Acoustics Speech and Signal Processing ICASSP*, pp. 625–628 vol.2, 1998

[15] Zhu, Q. and Alwan, A., "On the use of variable frame rate analysis in Speech recognition," *IEEE International Conference on Acoustics Speech and Signal Processing ICASSP*, pp. 1783–1786, 2000

[16] You, H., Zhu, Q., and Alwan, A., "Entropy-based variable frame rate analysis of speech signals and its application to ASR," *IEEE International Conference on Acoustics Speech and Signal Processing ICASSP*, volume 1, pp. 549–552, 2004

[17] Hermansky, H., "History of modulation spectrum in ASR," *IEEE International Conference on Acoustics Speech and Signal Processing ICASSP*, pp. 5458–5461, 2010

[18] Mitra, V., Franco, H., Graciarena, M., and Mandal, A., "Normalized amplitude modulation features for large vocabulary noise-robust Speech recognition," *IEEE International Conference on Acoustics Speech and Signal Processing ICASSP*, pp. 4117–4120, 2012

[19] Zhao, S.Y., Ravuri, S., and Morgan, N., "Multi-Stream to Many-Stream: Using Spectro-Temporal Features for ASR," *Proceedings of Interspeech Brighton United Kingdom*, pp. 2951–2954, 2009

[20] Raj, B. and Stern, R.M., "Missing-feature approaches in Speech recognition," *IEEE Signal Processing Magazine*, volume 22, no. 5, pp. 101–116, 2005

[21] Drygajlo, A. and El-Maliki, M., "Speaker verification in noisy environments with combined spectral subtraction and missing feature theory," *IEEE International Conference on Acoustics Speech and Signal Processing ICASSP*, volume 1, pp. 121–124, 1998

[22] Vizinho, A., Green, P., Cooke, M., and Josifovski, L., "Missing Data Theory, Spectral Subtraction And Signal-To-Noise Estimation For Robust ASR: An Integrated Study," *Proceedings of Eurospeech*, pp. 2407–2410, 1999

[23] Barker, J., Cooke, M., and Green, P. D., "Robust ASR based on clean Speech models: an evaluation of missing data techniques for connected digit recognition in noise." *Proceedings of Interspeech*, pp. 213–217, 2001

[24] Seltzer, M., Raj, B., and Stern, R.M., "A Bayesian classifier for spectrographic mask estimation for missing feature Speech recognition," *Speech Communication*, volume 43, no. 4, pp. 379–393, 2004

[25] Borgstrom, B.J. and Alwan, A., "Improved Speech Presence Probabilities Using HMM-Based Inference, With Applications to Speech Enhancement and ASR," *IEEE Journal of Selected Topics in Signal Processing*, volume 4, no. 5, pp. 808 –815, 2010

[26] Cooke, M., Morris, A., and Green, P., "Recognising occluded Speech," *Workshop on the Auditory Basis of Speech Perception*, pp. 297–300, 1996

[27] Raj, B., Singh, R., and Stern, R.M., "Inference of missing spectrographic features for robust Speech recognition," *Proceedings of ICSLP*, pp. 1491–1494, 1998

[28] Gemmeke, J. F., Van Hamme, H., Cranen, B., and Boves, L., "Compressive Sensing for Missing Data Imputation in Noise Robust Speech Recognition," *IEEE Journal of Selected Topics in Signal Processing*, volume 4, no. 2, pp. 272–287, 2010

[29] Gemmeke, J. F., Virtanen, T., and Hurmalainen, A., "Exemplar-Based Sparse Representations for Noise Robust Automatic Speech Recognition," *IEEE Transactions on Audio Speech and Language Processing*, volume 19, no. 7, pp. 2067–2080, 2011

[30] Borgstrom, B. J. and Alwan, A., "Utilizing Compressibility in Reconstructing Spectrographic Data, With Applications to Noise Robust ASR," *IEEE Signal Processing Letters*, volume 16, no. 5, pp. 398–401, 2009

[31] Hartmann, W. and Fosler-Lussier, E., "Investigations into the incorporation of the Ideal Binary Mask in ASR," *IEEE International Conference on Acoustics Speech and Signal Processing ICASSP*, pp. 4804–4807, 2011

[32] Borgstrom, B. J. and Alwan, A., "A Statistical Approach to Mel-Domain Mask Estimation for Missing-Feature ASR," *Signal Processing Letters IEEE*, volume 17, no. 11, pp. 941–944, 2010

[33] Barker, J., Josifovski, L., Cooke, M., and Green, P., "Soft Decisions In Missing Data Techniques For Robust Automatic Speech Recognition." *Proceedings of ICSLP*, pp. 373–376, 2000

[34] Martin, R., "Noise power spectral density estimation based on optimal smoothing and minimum statistics," *IEEE Transactions on Speech and Audio Processing*, volume 9, no. 5, pp. 504 –512, 2001

[35] Taghia, J., Taghia, J., Mohammadiha, N., Sang, Jinqiu, Bouse, V., and Martin, R., "An evaluation of noise power spectral density estimation algorithms in adverse acoustic environments," *IEEE International Conference on Acoustics Speech and Signal Processing ICASSP*, pp. 4640 –4643, 2011

[36] Brookes, M., *Voicebox, Speech Processing Toolbox for MATLAB*, Department of EE, Imperial College, London, www.ee.ic.ac.uk/hp/staff/dmb/voicebox/

[37] Raj, B., Seltzer, M.L., and Stern, R.M., "Reconstruction of missing features for robust Speech recognition," *Speech Communication*, volume 43, no. 4, pp. 275–296, 2004

[38] Kim, C., Kumar, K., and Stern, R.M., "Robust Speech recognition using a Small Power Boosting algorithm," *IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 243 –248, 2009

[39] Pearce, D. and Hirsch, H.-G., "The Aurora Experimental Framework for the Performance Evaluation of Speech Recognition Systems under Noisy Conditions," *ISCA ITRW ASR2000*, pp. 29–32, 2000

[40] Young, S., Kershaw, D., Odell, J., Ollason, D., Valtchev, V., and Woodland, P., *The HTK Book Version 3.0*, Cambridge University Press, 2000

[41] Vertanen, K., *HTK Wall Street Journal Training Recipe*, Department of Computer Science, Montana Tech of The University of Montana, http://www.keithv.com/software/htk/, 2008

[42] *The CMU Pronouncing Dictionary*, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, http://www.speech.cs.cmu.edu/cgi-bin/cmudict, 2001

[43] ETSI, "Speech Processing, Transmission and Quality Aspects; Distributed Speech recognition; Advanced front-end feature extraction algorithm; Compression algorithms ETSI ES 202 050," Technical report, 2007, http://www.etsi.org/