CAMBRIDGE UNIVERSITY

ENGINEERING DEPARTMENT

Uncertainty Decoding for Noise Robust Automatic Speech Recognition

H. Liao and M.J.F. Gales

CUED/F-INFENG/TR.499

October 2004

Cambridge University Engineering Department

Trumpington Street

Cambridge. CB2 1PZ

United Kingdom

E-mail: {hl251, mjfg}@eng.cam.ac.uk

http://mi.eng.cam.ac.uk/~{hl251, mjfg}

Abstract

This report presents uncertainty decoding as a method for robust automatic speech recognition for the Noise Robust Automatic Speech Recognition project funded by Toshiba Research Europe Limited. The effects of noise on speech recognition are reviewed and a general framework for noise robust speech recognition introduced. Common and related noise robustness techniques are described in the context of this framework. Uncertainty decoding is also presented in this framework with the goal of providing fast noise compensation through the propagation of uncertainty to the decoder. Two forms are discussed, the `Joint` and `SPLICE` methods, and evaluated on the medium vocabulary Resource Management corpus at a range of artificially produced noise levels. It was found that the uncertainty decoding algorithms did not meet the performance of a matched system, but were more accurate than the baseline `SPLICE` enhancement technique and low numbers of CMLLR transforms.

Glossary

BW	Baum-Welch
CDCN	Codeword Dependent Cepstral Normalisation
CMLLR	Constrained MLLR
CMN	Cepstral Mean Normalisation
CVN	Cepstral Variance Normalisation
DCT	Discrete Cosine Transform
DPMC	Data-Driven PMC
EM	Expectation Maximisation
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
HTK	HMM Toolkit
IDCT	Inverse DCT
IPMC	Iterative PMC
LVCSR	Large Vocabulary Continuous Speech Recognion
MAP	Maximum A Posteriori
MFCC	Mel-Frequency Cepstral Coefficients
MLLR	Maximum Likelihood Linear Regression
MMSE	Minimum Mean Squared Error
PLP	Perceptual Linear Prediction
PMC	Parallel Model Combination
POF	Probabilistic Optimal Filtering
RASTA	Relative Spectra
SDCN	SNR Dependent Cepstral Normalisation
SNR	Signal-to-Noise Ratio
SPLICE	Stereo Piece-wise Linear Compensation for Environments
VAD	Voice Activity Detector
VQ	Vector Quantisation
VTS	Vector Taylor Series
WER	Word Error Rate

Notation

$\boldsymbol{A}$	matrix of arbitrary dimensions
$\boldsymbol{A}^{\mathsf{T}}$	transpose of matrix $\boldsymbol{A}$
$\lvert\boldsymbol{A}\rvert$	determinant of matrix $\boldsymbol{A}$
$\boldsymbol{A}^{-1}$	inverse of matrix $\boldsymbol{A}$
$\boldsymbol{I}$	identity matrix
x	scalar quantity
$\hat{x}$	estimate of the true value of x
$\boldsymbol{x}$	vector of arbitrary dimensions
T	number of frames in a sequence of observations
M	number of GMM components in the acoustic model
N	number of GMM components in the front-end model
$\check{s}_n$	GMM front-end component n
$\check{c}_n$	mixture weight associated with front-end component $\check{s}_n$
$x^{(n)}$	parameter x is associated with front-end component $\check{s}_n$
s_m	GMM acoustic model component m
c_m	mixture weight associated with acoustic model component s_m
$x^{(m)}$	parameter x is associated with acoustic model component s_m
$x^{(r_m)}$	parameter x is associated with regression class r_m
$\mathcal{M}$	set of clean speech acoustic model parameters
$\hat{\mathcal{M}}$	set of estimated corrupted speech acoustic model parameters
$\check{\mathcal{M}}$	set of front-end model parameters
$\tilde{\mathcal{M}}$	set of noise model parameters
$\mathcal{E}\{f(x)\}$	the expected value of $f(x)$, where x is a random variable
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	multivariate Gaussian distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$
$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$	likelihood of vector $\boldsymbol{x}$ given a multivariate Gaussian distribution
$\boldsymbol{X}$	sequence of clean speech observations $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\}$
$\boldsymbol{Y}$	sequence of corrupted speech observations $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T\}$
$\boldsymbol{\theta}$	sequence of discrete clean speech states $\{\theta_1, \ldots, \theta_T\}$
$\boldsymbol{\theta}^n$	sequence of discrete noise speech states $\{\theta_1^n, \ldots, \theta_T^n\}$
$\mathcal{R}^d$	d-dimensional Euclidean space

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Speech recognition has improved markedly over the years such that it has gained adoption in consumer goods, call center applications and desktop personal computer software. The benchmarks for speech recognition continue to become more difficult with origins in yes/no and digit recognition tasks, to the thousand word Resource Management database, and now towards telephone conversation and broadcast news transcriptions. The standard HTK release [60] without language modeling yields a word error rate of 5.2% on the 1000 word ARPA Resource Management database. This approaches the human 2.0% error rate on nonsense sentences found in [54]. However once a nominal amount of background noise is added, the machine error rate rises to an unusable 65.5% – an order of magnitude greater. This noise at 20 dB SNR is hardly a distraction for the human ear[1]. This susceptibility to environmental noise is primarily due to the mismatch between the original conditions of the data used to *train* the recogniser and the actual data used to *test* it.

This was recognised early on, thus most noise robustness methods can be classified under several standard approaches, namely using *inherently robust front-ends*, *front-end compensation* and *model compensation*. The first seeks parameterisations that are fundamentally immune to noise. While effective in some circumstances with low-levels of noise, a generic inherently robust front-end has yet to be developed that can handle higher and varied noise levels. Hence, research has turned to feature enhancement or cleaning whereby noise is removed from the observed speech, yielding an estimate of the clean speech for decoding. Alternatively, the models can be compensated by incorporating the effects of the noise into the acoustic models. This is a far more powerful technique in that the model variances can be adapted to account for the noise, however the improved results come with a significant computational cost that is usually impractical for commercial LVCSR systems.

Recently, research has been directed at incorporating the uncertainty in speech recognition that noise causes. The uncertain observation method [6] formulates a method of incorporating

[1] In the author's opinion.

the uncertainty in decoding when the SNR decreases. This is different than the dynamic Bayesian network inference approaches found in the soft-information paradigm [39] and uncertainty decoding [15]. All these approaches seek to incorporate the frame-level uncertainty caused by noise into the decoding process to achieve accuracy comparable to model-based techniques at a speed similar to enhancement style schemes. This amounts to finding tractable forms of representing the uncertainty such that the model variances updates are fast to compute, yet effective.

1.1 Report Organisation

Following this introduction, a model of the environment is presented along with the effects that noise has on ASR, and a general statistical framework for reviewing past and current work in this field. In Chapter 3, classic and current noise robustness approaches are reviewed. Uncertainty decoding is formally presented in Chapter 4 and Chapter 5 discusses practical, implementation issues. In Chapter 6, initial experimental results from related robustness techniques and a novel `Joint` distribution uncertainty decoding method are presented and discussed. Finally, conclusions and future work directions are presented in Chapter 7.

Chapter 2

Speech Recognition in Noise

Research in increasing the robustness of automatic speech recognition systems to noise has been on-going for many decades. It is important to understand the particular difficulties noise presents to current algorithms in order to begin to address the problem. In this section, a general model of how environment noise affects the features used in LVCSR systems is described. The empirical effects are simulated, presented and discussed. Finally, a general framework for noise robust speech recognition is introduced.

2.1 Model of the Environment

It is impossible to name and describe all the possible noises that a speech recogniser could encounter. Fortunately, general sources and types can be categorised and their influence grouped into a general model of the environment. Acoustic degradation of the speech signal is typically understood as a complex process that initially suffers from changes to speaker articulation due to task stress, emotion or the Lombard effect [45]. This signal can then be coloured by additive background noise, channel distortions either due to the microphone or network with channel noise added, and finally possible noise at the near end of the speech recognition system. This is summarised in a model from [29]

$$y(m) = \Big[\Big(\Big\{\big[x(m)\Big|_{stress}^{Lombard}\big] + n_1(m)\Big\} * h_{mike}(m) + n_2(m)\Big) * h_{channel}(m)\Big] + n_3(m)$$

The main sources of concern are usually additive environmental noise, $n_1(m)$, present when the user is speaking and spectral tilt due to microphone mismatches. This yields the standard model of the noisy acoustic environment in the time domain, used by Acero [1], Gales [20] and later Moreno [48]

$$y[m] = x[m] * h[m] + n[m] \tag{2.1}$$

where $y[m]$ is the corrupted speech, $x[m]$ the "clean" speech, $h[m]$ the linear channel filter, and $n[m]$ the additive noise.

After applying the DFT the spectrum can be sampled using Mel-spaced filterbanks

$$Y(f_k) = X(f_k)H(f_k) + N(f_k) \tag{2.2}$$

where $k = \{1, \ldots, K\}$ denoting the filterbank bin. To compress the dynamic range, the natural logarithm is often applied

$$\log(Y(f_k)) = \log(X(f_k)) + \log(H(f_k)) + \log\bigl(1 + \exp\bigl(\log(N(f_k)) - \log(X(f_k)) - \log(H(f_k))\bigr)\bigr) \tag{2.3}$$

yielding a complex non-linear model of the acoustic environment in the log spectral domain. Cepstral parameters are the most popular features used in speech recognisers, thus it is of interest to examine the effect of noise in the cepstrum. First, the following cepstral vectors are defined[1]

$$\begin{aligned}
\boldsymbol{y}^{(c)} &= \boldsymbol{C}\bigl[\log(Y(f_0))\ \ \log(Y(f_1)) \ldots \log(Y(f_M))\bigr]^{\mathsf{T}} \\
\boldsymbol{x}^{(c)} &= \boldsymbol{C}\bigl[\log(X(f_0))\ \ \log(X(f_1)) \ldots \log(X(f_M))\bigr]^{\mathsf{T}} \\
\boldsymbol{h}^{(c)} &= \boldsymbol{C}\bigl[\log(H(f_0))\ \ \log(H(f_1)) \ldots \log(H(f_M))\bigr]^{\mathsf{T}} \\
\boldsymbol{n}^{(c)} &= \boldsymbol{C}\bigl[\log(N(f_0))\ \ \log(N(f_1)) \ldots \log(N(f_M))\bigr]^{\mathsf{T}}
\end{aligned} \tag{2.4}$$

and if we define a function $\boldsymbol{g}(\boldsymbol{a})$ of the form

$$\boldsymbol{g}(\boldsymbol{a}) = \boldsymbol{C}\log\bigl(\mathbf{1} + \exp\bigl(\boldsymbol{C}^{-1}\boldsymbol{a}\bigr)\bigr) \tag{2.5}$$

where $\boldsymbol{C}$ and $\boldsymbol{C}^{-1}$ are the discrete cosine transform matrix and it's inverse. Liftering equation 2.3 into the cepstral domain yields

$$\boldsymbol{y}^{(c)} = \boldsymbol{x}^{(c)} + \boldsymbol{h}^{(c)} + \boldsymbol{g}(\boldsymbol{n}^{(c)} - \boldsymbol{x}^{(c)} - \boldsymbol{h}^{(c)}) \tag{2.6}$$

This is the model of the corrupted speech environment for the commonly used MFCC parameterisation [11]. Though this, along with dynamic coefficients, is the most common form of feature vector, the optimality has been questioned[32]. Equation 2.6 clearly shows that the corrupted speech is a complex non-linear function of the channel, noise and normally Gaussian distributed clean speech.

2.2 The Effect of Noise on Speech Distributions

To more clearly understand the effects of noise, a simulation of it's influence on a Gaussian distribution can be conducted through the model of the environment. Figure 2.1 shows how a

[1] The use of the $^{(c)}$ is used for clarity here. In the following chapters, it should be assumed, unless otherwise noted, that vectors are cepstral.

single Gaussian representing the clean speech in the log spectrum, is affected by additive noise at different levels. If noise and clean speech are considered to be Gaussian distributed in the log-spectral domain, but noise additive in the spectral the following equation can be used to draw random vectors to plot a histogram of the corrupted speech

$$y_t = f(x_t, y_t) = \log(\exp(x_t) + \exp(n_t)) \tag{2.7}$$

where the noise is a randomly generated Gaussian variable all with mean 1, and the clean speech is also a randomly generated Gaussian variable with mean 10 and variance 5. At first there is a distinct bimodal distribution, but as the noise energy increases, the separability is lost and the distribution is once again unimodal with a strong skew. Also, there is a shift in the mean and a sharp decrease in variance. The same trend can be seen with actual data as shown in figure 2.2. This is a plot of the distribution of the 0th cepstral coefficient from a sample from the Resource Management corpus contaminated with various levels of artificially added noise. Models that are trained on the clean speech are highly tuned to the broad space across the range of the plot, but clearly with the addition of noise, the distributions share less and less area. Also, the silence models would be estimated on the peak on the left in clean models. However, these models would be ill-adapted for recognition in noise, since with added noise, the entire probability of the 0th cepstral coefficient falls outside of the clean silence model area. This is one example of mismatch due to noise.

2.3 A Framework for Noise Robust ASR

With these effects in mind, it is clear a general framework for robust speech recognition that explicitly accounts for the presence of noise is needed. To begin, the overall aim is to robustly determine what a speaker has said from a sequence of corrupted speech observations $\boldsymbol{Y} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T\}$. A corrupted speech observation can be thought of as the sum of hidden noise and clean speech variables where it is assumed that observations are conditionally independent given the clean speech $\boldsymbol{x}_t$ and the corrupting noise $\boldsymbol{n}_t$ at that time instance. Thus the clean speech $\boldsymbol{X}$ and noise sequence $\boldsymbol{N}$ can be considered independent, each generated by a hidden first-order Markov processes. The likelihood of the observation sequence is then expressed as

$$p(\boldsymbol{Y}|\mathcal{M}, \tilde{\mathcal{M}}) = \int_{2\mathcal{R}^{dT}} p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{N}, \mathcal{M}, \tilde{\mathcal{M}}) p(\boldsymbol{X}|\mathcal{M}) p(\boldsymbol{N}|\tilde{\mathcal{M}}) d\boldsymbol{X} d\boldsymbol{N} \tag{2.8}$$

$$\approx \sum_{\boldsymbol{\theta}, \boldsymbol{\theta}^n \in \boldsymbol{\Theta}} P(\boldsymbol{\theta}|\mathcal{M}) P(\boldsymbol{\theta}^n|\tilde{\mathcal{M}}) \prod_{t=1}^{T} \int_{2\mathcal{R}^d} p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{n}_t) p(\boldsymbol{x}_t|\mathcal{M}, \theta_t) p(\boldsymbol{n}_t|\tilde{\mathcal{M}}, \theta^n_t) d\boldsymbol{x}_t d\boldsymbol{n}_t \tag{2.9}$$

where $\boldsymbol{\Theta}$ is the set of all possible sequences of length T through the state space, $\mathcal{M}$ the speech model and $\tilde{\mathcal{M}}$ the noise model. This is compactly described in the Dynamic Bayesian network

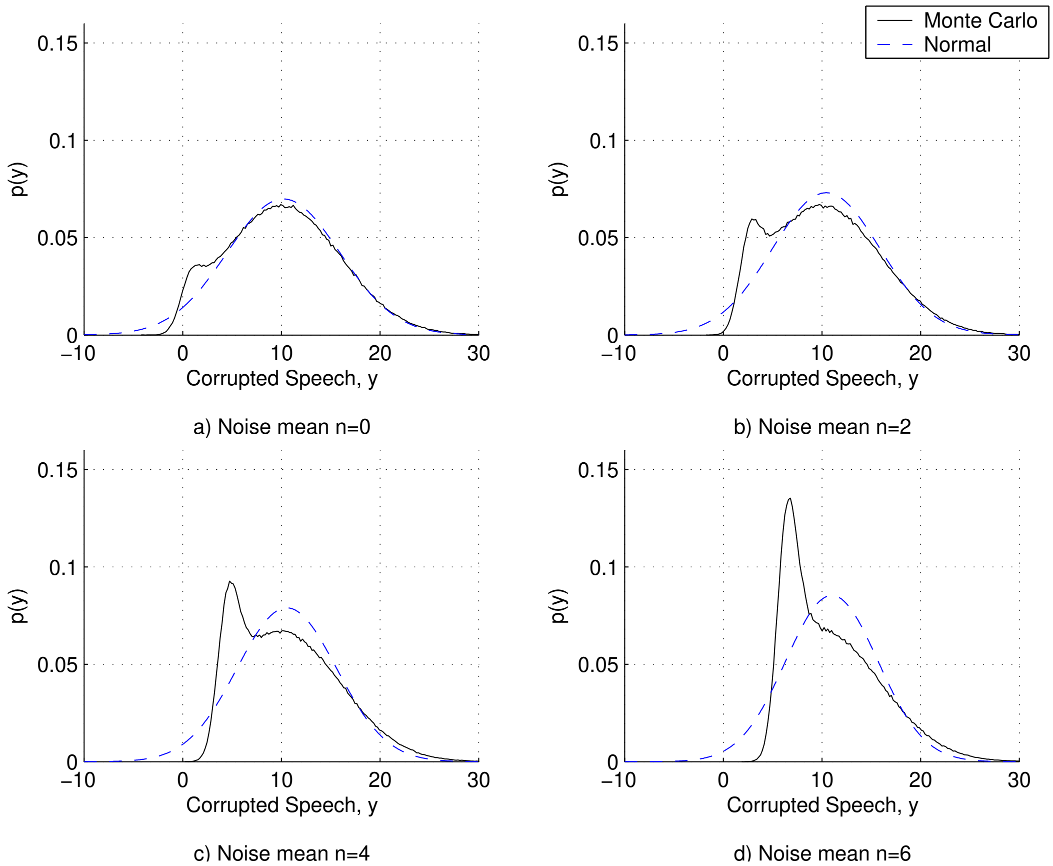

Figure 2.1: Corrupted speech distribution with clean speech of mean 10, variance 5, and ML estimate of Gaussian distribution.

shown in Figure 2.3. The circles indicate continuous variables and squares discrete with the arrows marking dependencies.

This framework is an extension of the typical application of hidden Markov models (HMMs) in speech recognition. Since HMMs have shown to be the best means of representing the time varying characteristics of speech, it makes sense to continue to leverage this representation. However, some of the assumptions in using HMMs that are tolerable with clean speech, may result in increased fragility to noise, such as the conditional independence of observations and the lack of explicit duration modeling in state transitions. Hermansky contends that the frailty of current ASR in realistic situations is due to broad across frequency processing, excessive attention to spectral structure, and poor modeling of the temporal structure of speech signals [31]. A frequent comparison is made to the robustness of human perception to speech that has the features of limited spectral resolution, broad temporal memory of larger acoustic segments, and the ability to mask unreliable features in the signal. The assumption that the clean speech is independent

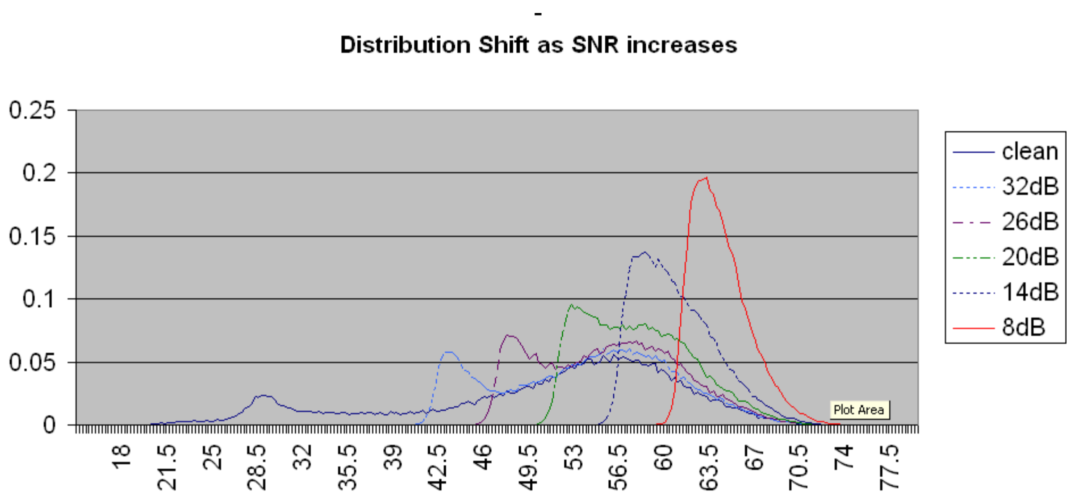

Figure 2.2: Corrupted speech distribution as SNR decreases.

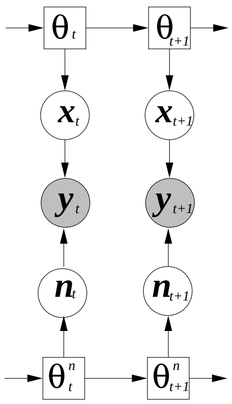

Figure 2.3: Dynamic Bayesian network for robust speech recognition. Emitting states are shaded, non-emitting hidden variables are unshaded.

of the noise is not true as demonstrated by the Lombard effect; however, it is assumed because of the lack of an accurate and efficient means to model this effect. Speech signal production has strong constraints that could be exploited for more robust recognition that are not exploited by the 1st order Markov assumption. For example recent work has looked into using a switching linear dynamic model to take advantage of the smooth time varying qualities of speech [14] for speech enhancement.

Chapter 3

Techniques for Noise Robust ASR

There are many approaches to robustly recognising noise corrupted speech. Ideally, a robust noise immune parameterisation could be found such that the recogniser would inherently be unaffected by noise. So far this has not been possible, hence techniques focus on reducing the mismatch between the training and usage conditions. These can be grouped into two distinct paradigms as shown in figure 3.1. The front-end in speech recognition systems is responsible for capturing and process the speech signal into a lower-dimensional feature vector for recognition. The acoustic models represent the speech itself and is used by the decoding to make a hypothesis of what is said. Front-end compensation seeks to correct the corrupted observation into an estimate that more closely resembles clean speech. These estimates can then be decoded using the clean acoustic models. Acoustic model compensation aims to adapt or transform the clean acoustic models to a corrupted set that better matches the noise corrupted observations. These are further discussed in detail.

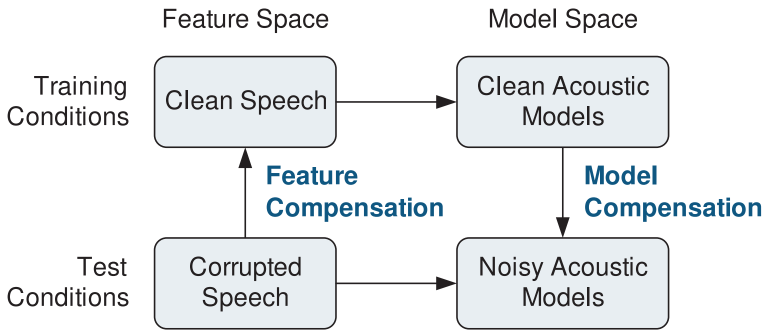

Figure 3.1: Methods of reducing the acoustic mismatch.

3.1 Inherently Robust Front-Ends

A straightforward response to the problem of environmental noise is to build a system that is not susceptible to it. The move from using log-spectral features to MFCCs could be considered one of moving towards a more robust parameterisation. Still, widely used MFCC[11] and PLP[30] parameters on their own are not immune to the affects of noise. In this framework for noise robust speech recognition, this would be the equivalent of completely ignoring the noise process and following the standard approach for HMM decoding using the corrupted observations directly

$$p(\boldsymbol{Y}|\mathcal{M}) = \sum_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} P(\boldsymbol{\theta}|\mathcal{M}) \prod_{t=1}^{T} p(\boldsymbol{y}_t|\mathcal{M}, \theta_t) \tag{3.1}$$

3.1.1 RASTA-PLP and J-RASTA-PLP

Perceptual linear predictor co-efficients have been studied for the use speech recognition and then extended with relative spectral (RASTA) processing to yield RASTA-PLP coefficients that are perceptually motivated [33, 37]. The bandpass filtering in RASTA is motivated by the fact that modulations in the spectrum below 1 Hz and above 12 Hz are usually noise and best removed. The integration over several frames of speech yielding smoothing over 150-170 ms simulates the human feature of incorporating information over time. The net effect is enhancement of dynamic features and the suppression of static or slowly changing ones. The addition of a parameterised log-J function gives rise to J-RASTA that can handle both additive and convolutional noise.

It has been shown how RASTA processing effectively mitigates convolutional noise and the handles additive noise with the use of J-RASTA [37]. However the J parameter effectively is a compromising value between the degree of convolutional and additive noise removal. Even with the variance of this parameter on different noise conditions, the compromise results in degradation of clean speech recognition with convolution noise to incorporate the ability to handle additive noise. Still, it is a promising development, and other groups such as [53] have used RASTA-PLP parameterisations with neural network observation modeling to achieve phone accuracy similar to levels using MFCC and Gaussian mixture models with the assumed (but unmeasured) benefit of improved robustness. Also, PLP brings recognition accuracy of child speech closer to that of adults as example of its robustness [31].

3.1.2 Cepstral Normalisation

An extremely effective method to address channel mismatch is cepstral mean normalisation (CMN); the removal of the cepstral bias that results from slowly changing convolutional noise sources. One issue with this is the estimation of the bias. Using the entire utterance yields significant improvements; but for real-time use, there is an issue of initial estimate of the bias, and limiting the estimation to a certain time window so that the delay introduced is tolerable. The merit of this algorithm is demonstrated in its widespread application in the majority of papers surveyed. Reports [43, 59] clearly show the gains from incorporating CMN to handle channel mismatch noise.

A natural extension is cepstral variance normalisation (CVN). Along with CMN, this represents linear shifts in the mean and variance to address the effects of noise. However, the effects of additive noise are non-linear and so these techniques are not entirely successful. Another step is to warp the feature vector in a non-linear fashion to "Gaussianise it" or histogram equalisation[58] or normalisation[47]. The warping of the feature space is through a transform based on the cumulative histograms of the noisy and clean speech. The reduction in error rate in using histogram equalisation over CMN and CVN is similar to the gains CMN and CVN provide over baseline clean results.

3.2 Feature Compensation

From figure 3.1, one approach is to process the incoming observations $\boldsymbol{Y}$ to better resemble the features the original clean speech acoustic model was trained on

$$\hat{\boldsymbol{X}} = \mathcal{F}(\boldsymbol{Y}, \mathcal{M}, \tilde{\mathcal{M}}) \tag{3.2}$$

where $\hat{\boldsymbol{X}} = \{\hat{\boldsymbol{x}}_1, \ldots, \hat{\boldsymbol{x}}_T\}$ and represents the set of estimated clean speech observations computed from the noise corrupted observations $\boldsymbol{Y}$ and the clean and noise models. For enhancement, it is often the case that the corrupted speech is mapped deterministically to a clean speech estimate, given some estimate of the noise

$$\int_{\mathcal{R}^{dT}} p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{n}_t)\, p\Big(\boldsymbol{n}_t|\tilde{\mathcal{M}}, \theta_t^n\Big)\, d\boldsymbol{n}_t = p\big(\boldsymbol{y}_t|\boldsymbol{x}_t, \check{\mathcal{M}}\big) = \alpha_t \delta(\hat{\boldsymbol{x}}_t - \boldsymbol{x}_t) \tag{3.3}$$

Here the marginalisation over the unknown noise state using noise models $\tilde{\mathcal{M}}$ is replaced by a probabilistic distribution conditioned on parameters $\check{\mathcal{M}}$ assuming a certain noise condition; for deterministic enhancement algorithms this probability distribution is a Dirac delta function. This substituted into equation 2.9 yields the front-end compensation framework where the estimate of the clean speech is directly used for decoding

$$p(\boldsymbol{Y}|\mathcal{M}, \tilde{\mathcal{M}}) = \sum_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} P(\boldsymbol{\theta}|\mathcal{M}) \prod_{t=1}^{T} \alpha_t p(\hat{\boldsymbol{x}}_t|\mathcal{M}, \theta_t) \tag{3.4}$$

There are various methods to compute $\hat{\boldsymbol{x}}$; these can be broadly classified into those that enhance the spectral domain, and those that compensate the cepstral parameters. Figure 3.2 outlines the standard feature compensation process.

3.2.1 Spectral Subtraction

This technique [8] is often quoted as a baseline algorithm for comparisons and is widely used to successfully mitigate additive noise. The noise power spectrum can be estimated from frames that

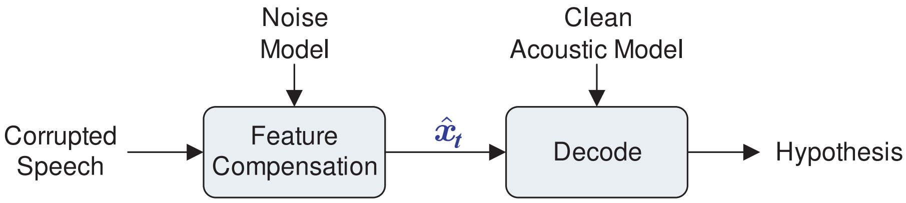

Figure 3.2: The standard feature compensation process

are classified as not having speech. This estimate of the noise can then be subtracted from the corrupted signal to yield an enhanced feature vector

$$\hat{X}(f_k) = \sqrt{|Y(f_k)|^2 - |N(f_k)|^2} \tag{3.5}$$

This assumes the noise is additive in the time domain, is uncorrelated with the speech, and varies slowly in time. Practically, this technique has shown good results even with these simple assumptions on the noise and the need for a VAD to provide a background estimate. Later algorithmic improvements [57] can remove the need for VAD by using estimated background noise through histograms of the energy of the observed signal in several frequency bands; the distribution will tend to be bimodal, and the noise spectrum can be estimated by using the appropriate quantile over time.

3.2.2 State-Based Speech Enhancement

The original spectral subtraction technique assumes stationary noise. Promising results can be attained by aligning a simple front-end HMM to the corrupted speech and using the state statistics to more informatively enhance the speech using Weiner filters. The corrupted speech models of the front-end HMM can be recursively estimated from a combination of the clean and noise models using an EM algorithm as suggested in [18]. Since the corrupted state sequence should map to the clean in a one-to-one fashion, the clean speech state sequence can be obtained. This allows for better estimates of the clean and noise speech statistics which can be used in the enhancement process. Enhancement with auto-regressive, hidden Markov models of speech are studied in [17, 40, 44] and cepstral domain HMMs in [55].

3.2.3 Codeword Dependent Cepstral Normalisation

CDCN attempts to learn a mapping from the corrupted speech domain to the clean speech [1]. The estimate of the clean speech is a weighted sum of the bias vectors for each VQ region in the code-book

$$\hat{\boldsymbol{x}}_t = \boldsymbol{y}_t - \hat{\boldsymbol{h}} - \sum_{i=1}^{I} \boldsymbol{f}(i)\boldsymbol{b}(i) \tag{3.6}$$

where $\hat{\boldsymbol{h}}$ is an estimate of the constant channel noise, $\boldsymbol{f}(i)$ is interpreted as the a posteriori probability of region i given the observed corrupted speech and the environmental parameters and $\boldsymbol{b}(i)$ the correction vector for the associated region. CDCN depends on the online estimation of the channel and additive noise through an iterative EM approach. Since this can be somewhat intensive to compute, an environment specific form was introduced called SNR-dependent cepstral normalisation or SCDN [3, 4]. Here, the correction is dependent on the SNR and trained using stereo data

$$\hat{\boldsymbol{x}}_t = \boldsymbol{y}_t - f(SNR) \tag{3.7}$$

These forms can be considered as a more sophisticated cepstral normalisation, where the cepstral bias is based on regions of space or the SNR. These appear to be the first instances of algorithms to partition the acoustic space and apply different MMSE correction factors to separate regions.

3.2.4 Probabilistic Optimal Filtering

Probabilistic optimal filtering (POF) can be considered a generic piece-wise minimum squared error approach to enhancement [49]. It is similar to CDCN in that it specifies linear transforms learnt in a MMSE fashion for regions of the acoustic space. Each VQ region is partitioned by a Gaussian mixture model where each component has a corresponding transform. Succinctly, the estimate of the clean speech vector from [49] is

$$\begin{aligned}
\hat{\boldsymbol{x}}_n &= \sum_{i=1}^{I}\Big\{\boldsymbol{W}_i^{\mathsf{T}} P(g_i|\boldsymbol{z}_n)\Big\}\boldsymbol{Y}_n && (3.8)\\
\boldsymbol{W}_i^{\mathsf{T}} &= \Big[\boldsymbol{A}_{i,-p}\dots\ \boldsymbol{A}_{i,-1}\ \ \boldsymbol{A}_{i,0}\ \ \boldsymbol{A}_{i,1}\dots\ \boldsymbol{A}_{i,p}\ \ \boldsymbol{b}_i\Big] && (3.9)\\
\boldsymbol{Y}_n^{\mathsf{T}} &= \Big[\boldsymbol{y}_{n-p}^{\mathsf{T}}\cdots\ \boldsymbol{y}_{n-1}^{\mathsf{T}}\ \ \boldsymbol{y}_n^{\mathsf{T}}\ \ \boldsymbol{y}_{n+1}^{\mathsf{T}}\cdots\ \boldsymbol{y}_{n+p}^{\mathsf{T}}\ \ 1\Big] && (3.10)
\end{aligned}$$

where i is the VQ region index, n the frame, p the filter delay, and $\boldsymbol{z}_n$ the conditioning vector. The conditioning vector, which selects the appropriate transform to apply, can span multiple frames, contain extra parameters such as local SNR, and thus is not necessarily the feature vector.

3.2.5 SPLICE

SPLICE [12], descendant from FCDCN[4], has shown extremely good results in AURORA testing and can be considered as a special case of POF. `SPLICE` uses a probabilistic approach where the corrupted observations and the clean speech given the noisy are both modeled by Gaussian mixture models. With a GMM partitioning the acoustic into N regions, each region has an associated linear compensation bias to map the observed corrupted speech vector to an estimated clean. Thus, the corrupted space is modeled by a GMM

$$p\big(\boldsymbol{y}_t|\check{\mathcal{M}}\big) = \sum_{n=1}^{N} P\big(\check{s}_n|\boldsymbol{y}_t,\check{\mathcal{M}}\big)\,\mathcal{N}\Big(\boldsymbol{y}_t;\boldsymbol{\mu}_y^{(n)},\boldsymbol{\Sigma}_y^{(n)}\Big) \tag{3.11}$$

where the posterior of component $\check{s}_n$ is given by

$$P\big(\check{s}_n|\boldsymbol{y}_t, \check{\mathcal{M}}\big) = \frac{\check{c}_n p(\boldsymbol{y}_t|\check{s}_n)}{\sum_{i=1}^{N} \check{c}_i p(\boldsymbol{y}_t|\check{s}_i)} \tag{3.12}$$

The a posteriori probability of the clean speech, for a component $\check{s}_n$, is given by

$$p(\boldsymbol{x}_t|\boldsymbol{y}_t, \check{s}_n) = \mathcal{N}\big(\boldsymbol{x}_t; \boldsymbol{y}_t + \check{\boldsymbol{\mu}}^{(n)}, \check{\boldsymbol{\Sigma}}^{(n)}\big) \tag{3.13}$$

The correction vectors are estimated using stereo data in the following manner

$$\check{\boldsymbol{\mu}}^{(n)} = \mathcal{E}\left\{\boldsymbol{x}_t - \boldsymbol{y}_t|\check{s}_n\right\} \tag{3.14}$$

$$\check{\boldsymbol{\Sigma}}^{(n)} = \mathcal{E}\left\{(\boldsymbol{x}_t - \boldsymbol{y}_t)(\boldsymbol{x}_t - \boldsymbol{y}_t)^{\mathsf{T}}|\check{s}_n\right\} - \check{\boldsymbol{\mu}}^{(n)}\check{\boldsymbol{\mu}}^{(n)\mathsf{T}} \tag{3.15}$$

The term $\check{\boldsymbol{\Sigma}}^{(n)}$ can be interpreted as the expected square error of the estimation. Thus, the MMSE estimate of the clean speech is

$$\hat{\boldsymbol{x}}_t = \int_{\mathcal{R}^d} \boldsymbol{x}_t p(\boldsymbol{x}_t|\boldsymbol{y}_t, \check{s}_n)\, d\boldsymbol{x}_t = \sum_{n=1}^{N} P\big(\check{s}_n|\boldsymbol{y}_t, \check{\mathcal{M}}\big)\left(\boldsymbol{y}_t + \check{\boldsymbol{\mu}}^{(n)}\right) \tag{3.16}$$

This involves computing a weighted clean speech estimate for each region and then summing them for the the final estimate which is referred to *s*oft `SPLICE` enhancement. Alternatively, the most probable component $\check{s}_n^*$ can be used in place of the soft weighted estimate

$$\check{s}_{n^*} = \arg\max_{\check{s}_n}\Big[\check{c}_n P\big(\boldsymbol{y}_t|\check{s}_n, \check{\mathcal{M}}\big)\Big] \tag{3.17}$$

This *h*ard estimate yields a more efficient version of `SPLICE` enhancement

$$\hat{\boldsymbol{x}}_t = \boldsymbol{y}_t + \check{\boldsymbol{\mu}}^{(n)^*} \tag{3.18}$$

Similar in form to POF, `SPLICE` is an efficient front-end noise robustness scheme where the corrupted speech vector is compensated by a linear bias $\check{\boldsymbol{\mu}}^{(n)}$ varying on the region of the acoustic space the observation resides in. Figure 3.3 depicts this operation for a four component front-end GMM.

3.2.6 Feature Domain Vector Taylor Series

The environmental model, presented in section 2.1, can be used to compensate the corrupted cepstral feature vectors by providing an estimate of the unobserved clean speech vector [48] in the log spectral domain. Later a cepstral form of the environment was derived [35]. This can also be used to form a cepstral domain MMSE estimate of the clean speech.

Based on the 0th-order VTS approximation of the corrupted speech, as in equation 2.6

$$\boldsymbol{x}_t = \boldsymbol{y}_t - g(\boldsymbol{n}_t - \boldsymbol{x}_t - \boldsymbol{h}_t) \tag{3.19}$$

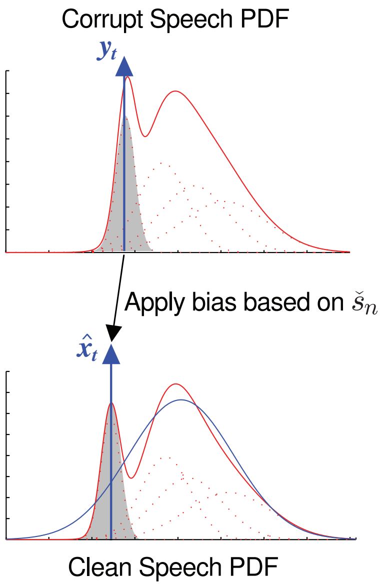

Figure 3.3: SPLICE feature enhancement

a MMSE estimate of the clean speech can be derived

$$\hat{\boldsymbol{x}}_t = \int_{\mathcal{R}^d} \boldsymbol{x}_t p(\boldsymbol{x}_t|\boldsymbol{y}_t) d\boldsymbol{x}_t \tag{3.20}$$

$$= \boldsymbol{y}_t - \int_{\mathcal{R}^d} g(\boldsymbol{n}_t - \boldsymbol{x}_t - \boldsymbol{h}_t) p(\boldsymbol{x}_t|\boldsymbol{y}_t) d\boldsymbol{x}_t \tag{3.21}$$

If a front-end GMM is used to partition the acoustic the space such that a set of compensation parameters is estimated for each partition, a constant channel noise μ_h determined, and a deterministic mapping between $\boldsymbol{y}_t$ and $\boldsymbol{x}_t$ is used then

$$\hat{\boldsymbol{x}}_t = \boldsymbol{y}_t - \int_{\mathcal{R}^d} \sum_{n=1}^{N} g(\boldsymbol{n}_t - \boldsymbol{\mu}_x^{(n)} - \boldsymbol{\mu}_h) p(\boldsymbol{x}_t|s_n, \boldsymbol{y}_t) d\boldsymbol{x}_t \tag{3.22}$$

$$= \boldsymbol{y}_t - \sum_{n=1}^{N} P(s_n|\boldsymbol{y}_t) g(\boldsymbol{n}_t - \boldsymbol{\mu}_x^{(n)} - \boldsymbol{\mu}_h) \tag{3.23}$$

The statistics of the varying additive noise and constant channel noise are often estimated online through an iterative EM framework.

3.2.7 Uncertain Observations

Arrowood in [6] discusses an intuitive method of decoding with uncertain observations due to noise. A probability density function is produced by the front-end instead of just a point observation to the decoding process as shown in figure 4.2 and in contrast to figure 3.2. This is to reflect the uncertainty in the removal of noise from the feature vector. The bias and variance of the observation distribution are second-order polynomial functions of the estimated SNR based on the filterbank parameters

$$\hat{\boldsymbol{\mu}}_t(f_k) = a_0(f_k) + a_1(f_k)\boldsymbol{w}_t(f_k) + a_2(f_k)\boldsymbol{w}_t(f_k) \tag{3.24}$$

$$\hat{\boldsymbol{\mu}}_t(f_k) = b_0(f_k) + b_1(f_k)\boldsymbol{w}_t(f_k) + b_2(f_k)\boldsymbol{w}_t(f_k) \tag{3.25}$$

and

$$\boldsymbol{w}_t(f_k) = \boldsymbol{y}_t(f_k) - \hat{\boldsymbol{x}}_t(f_k) \tag{3.26}$$

at frame t. This observation uncertainty, interpreted as a single multivariate, clean speech posterior Gaussian distribution, can then be propagated to the decoder as biases to the corrupted speech and the acoustic model variance. For cepstral coefficients, these filterbank parameters can be liftered. Results comparable to PMC using the log-add approximation are reported in [7] in white Gaussian additive and channel varying noise.

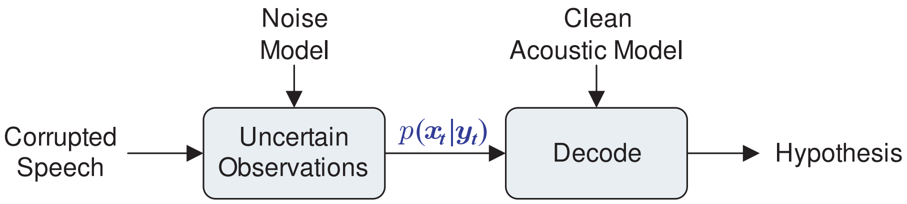

Figure 3.4: Feature compensation with uncertain observations

3.2.8 Missing Feature Theory

Another approach, inspired from vision[5], has been to treat certain elements of the feature vector as unreliable or missing[10]. It has been noted that listeners naturally handle missing data in everyday communications. Detecting unreliable areas of speech is done at a spectral level using local SNR as a measure. Once parameters have been partitioned into reliable and unreliable, unreliable ones can be marginalised over[10, 16] or restored[51, 42]. Marginalisation requires changes to the recogniser whereas restoration, or data imputation, can be used as a general front-end enhancement system.

3.3 Model Compensation

As figure 3.1 also indicates, the acoustic models can be updated for the noise condition such that they better match the incoming corrupted speech observations

$$p(\boldsymbol{Y}|\hat{\mathcal{M}}) = \sum_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} P(\boldsymbol{\theta}|\hat{\mathcal{M}}) \prod_{t=1}^{T} p(\boldsymbol{y}_t|\hat{\mathcal{M}}, \theta_t) \tag{3.27}$$

where $\hat{\mathcal{M}}$ are the compensated noisy acoustic models. The noisy acoustic models can be derived from the observed corrupted speech, the clean acoustic models, and perhaps noise models

$$\hat{\mathcal{M}} = \mathcal{G}(\boldsymbol{Y}, \mathcal{M}, \tilde{\mathcal{M}}) \tag{3.28}$$

There are two general approaches to compensating acoustic models: *adaptive*, where sufficient corrupted data is available to update the acoustic models directly to match the noisy speech observations; and *predictive*, where a noise model is combined with the clean speech models to provide a corrupted speech acoustic model. Retraining of the models, MAP, and MLLR-style transforms can be considered adaptive forms, whilst PMC and VTS are predictive techniques.

3.3.1 Training on Corrupted Speech

The most obvious approach to handling mismatched training and test conditions is to retrain the acoustic models in the new environment. While this usually yields the best results in a variety of papers surveyed [28, 23, 59], it is not very practical, as collecting large quantities of speech data at varying noise conditions is time-consuming. Artificial methods of corrupting the training data have been explored which also yield good results. Samples, such as those from NOISEX-92 can be added to the clean training data utterances to provide corrupted training data. This provides good results for levels of noise down to 6-10dB, but with varying SNR or increased levels, the mismatch problem arises and degradation occurs [28]. Adding a variety of noise samples to clean training data is known as multi-style training [52, 12] and is generally assumed to be the theoretical upper limit for speech recognisers, although humans tend to still do better than this [34].

The artificial addition of noise does not account for changes in speech production that occur in such conditions. Training on speech transformed to artificially add stress improves isolated word recognition rates[9]. Such a technique in concert with artificially added background noise or spectral subtraction could achieve even better results than independently, however the end result is still a system highly tuned towards a certain noise characteristic.

3.3.2 Single Pass Retraining

Re-training acoustic models directly on the corrupted speech data typically trains state-level alignments from the noisy speech. This reduces the performance of the system since the state posteriors become more unreliable as noise increases. Thus model estimation using well-trained clean state posteriors such as with Single Pass Retraining (SPR)[20] represents an ideal model-based compensation scheme. With SPR though, the corrupted speech distributions are still badly modeled. The corrupted output distributions can be retrained by further BW iterations, with the state level posteriors fixed to the clean using two-model re-estimation[60]. This assumes the state posteriors in clean and corrupted speech are constant, which is accurate for artificially corrupted data, but not for natural situations as it well known speech production changes in high noise – the Lombard effect. As well, each Gaussian component is updated to fit the corrupted speech. From table 2.1, it is clear that the dashed maximum likelihood estimate fails to accurately represent the corrupted distribution. This is a general problem for most model compensation techniques that use the clean speech posteriors and update on a component by component basis.

Moreover, SPR and two-model estimation are offline compensation techniques not suitable for varying acoustic environments. There are other methods of dynamically updated the state distributions of the recogniser in an online system. Typically speech recognition use GMMs to model output distributions. The non-linear combination of $\boldsymbol{x}$,$\boldsymbol{h}$, and $\boldsymbol{n}$ makes it difficult to derive a closed form solution for modeling the distribution of the corrupted speech vector $\boldsymbol{y}$ in equation 2.6. Moreno [48] solves this in the log-spectral domain by using a vector Taylor series approximation of the non-linearity. Alternatively with PMC [24], Gales applies various approximations in the log-spectral domain to combine models of clean speech and noise to yield transformed to the cepstral domain to yield corrupted models of speech. These techniques still do not ideally model the corrupted speech distribution since each Gaussian component in the original clean speech model is still represented with only one Gaussian component in the corrupt model, when it has been shown that with the introduction of noise, the distribution because distinctly bi-modal and non-Gaussian [23].

3.3.3 Adaptation of Acoustic Models

So far, there has been little discussion on dealing with previously described speech production changes in the presence of noise known as the Lombard effect. One method to mitigate such effects is to adapt the durational parameters of the phone models on observed Lombard speech data[56]. Results showed this could improve recognition accuracy, however this fails to address the environmental noise in the signal, and large amounts of data are required to approach the MLE of the corrupted speech.

Adaptation techniques like MAP and MLLR can be used to fully adapt acoustic models to both speech production changes due to the Lombard effect and stress, and to stationary additive or convolutional noise. These approaches can bring performance close to the matched condition. However the amount of data MAP requires to adapt makes it impractical for online adaptation; the need for transcription versus unsupervised adaptation a concern in noise; and the storage and propagation of adapted models in real-world server-based telephony applications a problem. The adapted models also suffer from being specific to the new adaptation data, and any change in noise will require further adaptation.

The use of linear model-space transformations has been explored in depth [41, 22] although mostly in the context of speaker adaptation. In noise robustness, MLLR transforms have been successfully applied to adapt models to noise within a session on the AURORA 2.0 digits corpus [43], but in general many transforms are necessary to model the non-linear affects of noise [34]. These transforms can be classified as either unconstrained, where the compensated mean and variance are given by

$$\hat{\boldsymbol{\mu}}^{(m)} = \boldsymbol{A}\boldsymbol{\mu}^{(m)} + \boldsymbol{b} \tag{3.29}$$

$$\hat{\boldsymbol{\Sigma}}^{(m)} = \boldsymbol{H}\boldsymbol{\Sigma}^{(m)}\boldsymbol{H}^{\mathsf{T}} \tag{3.30}$$

or constrained

$$\hat{\boldsymbol{\mu}}^{(m)} = \boldsymbol{A}\boldsymbol{\mu}^{(m)} + \boldsymbol{b} \tag{3.31}$$

$$\hat{\boldsymbol{\Sigma}}^{(m)} = \boldsymbol{A}\boldsymbol{\Sigma}^{(m)}\boldsymbol{A}^{\mathsf{T}} \tag{3.32}$$

where the transformation matrix of the mean and variance parameters is the same. Often, the mean vector update is written as

$$\hat{\boldsymbol{\mu}}^{(m)} = \boldsymbol{W}\boldsymbol{\xi}^{(m)} \tag{3.33}$$

where $\boldsymbol{\xi}^{(m)}$ is the extended mean vector $\begin{bmatrix} 1 & \hat{\boldsymbol{\mu}}^{(m)\mathsf{T}} \end{bmatrix}^{\mathsf{T}}$ and $\boldsymbol{W}$ the extended transform $\begin{bmatrix} \boldsymbol{b}^{\mathsf{T}} & \boldsymbol{A}^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}$.

A benefit of the constrained transform, or CMLLR, is that with some matrix algebra, the transformation can be efficiently applied in feature space

$$\hat{\boldsymbol{x}}_t = \boldsymbol{A}'\boldsymbol{x}_t + \boldsymbol{b}' \tag{3.34}$$

where $\boldsymbol{A}' = \boldsymbol{A}^{-1}$ and $\boldsymbol{b}' = \boldsymbol{A}^{-1}\boldsymbol{b}$. A normalisation term of $\log(|\boldsymbol{A}'|)$ is required during the likelihood calculation [22]. The form of the transformation has also been studied where it has been reported that a block transform of the mean vector is more effective than the diagonal [22], but for the variance transform the diagonal form just as effective as the block [25] with less computational cost.

Linear transforms mapping the output distributions of the clean speech models to the corrupted environment can be robustly estimated on less data than MAP and similar models can be grouped together into classes using a regression tree, sharing the transform and adaptation data [21]. Regression class trees group models either by decision trees using phonetic knowledge, or by comparing how close models are in the acoustic space. Each leaf of the tree has a transform trained for it unless there is insufficient data for a robust estimation. In this case, it can regress to the parent node's transform estimated from aggregating the data of it's leaves. This gives an elegant means to scale the number of transforms to the available adaptation data.

3.3.4 Parallel Model Combination

PMC combines separate noise and speech models to form a corrupted speech model directly for use in the recognition process [26]. The composition of the noise and speech is done through a mismatch function describing how the noise and speech are combined to form the corrupted signal. Specific additive, convolutional, additive and convolutional, and bandwidth limited channel mismatch functions can be found in [23]. The log-normal approximation is a popular and efficient choice that assumes the sum of two log-normal distributions is approximately log-normal, however cannot be applied with delta and delta-delta parameters due to the resulting complexity of the forms [27]. PMC can restore performance in a 10 dB SNR environment to a level comparable to training models with actual noisy speech [24].

The transform of the parameters of each Gaussian component in the clean model to reflect the noise does not model the overall corrupted speech distribution well as seen in figure 2.1. Iterative PMC (IPMC) and data-driven PMC (DPMC) aim to resolve this problem [20]. Iterative PMC (IPMC) addresses this issue by representing each component with multiple components, iteratively re-estimating the GMM modeling the corrupted speech, still based on alignments from the clean speech posteriors. This increases the number of components in the overall system. Alternatively, data-driven PMC directly estimates the corrupted speech distribution by drawing sample corrupted speech vectors from combinations of the clean and noise models to re-estimate the GMM on a per state basis. The efficient log-add approximation can be used to combine the model and the overall number of components can remain unchanged, however anywhere from 25-1000 observations need to be generated per Gaussian in the system [23]. These approaches should match a SPR with two-model re-estimation system discussed earlier and represent ideal forms of model-based compensation. However, they came at a high cost, with the iterative re-estimation of each state distribution extremely computationally expensive.

As in the use of PMC for speech enhancement, these model-based schemes depend on the quality of the noise models used [23] and the model accuracy of the interaction of the noisy environment with speech. Single state noise models are fast and efficient, but can only handle slowly changing noise statistics. The assumption of independence between noise and speech production is not a good one as speech changes with volume of noise. To handle rapidly changing noise, more states are required in the noise model, significantly increasing computational complexity in the decoding to find the optimal combination of speech and noise. Also the training of the noise models is not trivial. For additive noises a variety of sources are available like the NOISEX-92 database, but convolutional noise samples are more difficult to obtain. The performance of PMC depends on having appropriate noise models, but as Bishnu Atal is purported to have said, "We call it noise because we know nothing about it!" - unknown sources of noise arise. Despite these drawbacks, PMC has shown to work fairly well.

3.3.5 Model Domain Vector Taylor Series

Typically speech recognition uses GMMs to model the output distributions. However the non-linear combination of $\boldsymbol{x}$, $\boldsymbol{h}$, and $\boldsymbol{n}$ make it intractable to calculate a closed form solution for modeling the distribution of the corrupted speech vector $\boldsymbol{y}$ in equation 2.6. Moreno [48] solves this by using a vector Taylor series approximation of the non-linearity in the log spectral domain. This has also been applied in the cepstral domain [2, 36] with results close to the matched system. To do so, a first-order approximation of the environment is made, assuming independence of $\boldsymbol{x}$, $\boldsymbol{h}$ and $\boldsymbol{n}$, evaluated about $\boldsymbol{\mu} = \boldsymbol{\mu}_n - \boldsymbol{\mu}_x - \boldsymbol{\mu}_h$

$$\boldsymbol{y} = \boldsymbol{\mu}_x + \boldsymbol{\mu}_h + g(\mu) + \boldsymbol{A}(\boldsymbol{x} - \boldsymbol{\mu}_x) + \boldsymbol{A}(\boldsymbol{h} - \boldsymbol{\mu}_h) + (\boldsymbol{I} - \boldsymbol{A})(\boldsymbol{n} - \boldsymbol{\mu}_n) \tag{3.35}$$

with

$$\boldsymbol{A} = \boldsymbol{C}\boldsymbol{F}\boldsymbol{C}^{-1} \tag{3.36}$$

and $\boldsymbol{F}$ a diagonal matrix with the following elements from vector $f(\boldsymbol{\mu})$

$$f(\boldsymbol{\mu}) = \frac{1}{1 + e^{C^{-1}\mu}} \tag{3.37}$$

giving the following estimates of the mean and variance of the corrupted speech

$$\begin{aligned} \boldsymbol{\mu}_y &\approx \boldsymbol{\mu}_x + \boldsymbol{\mu}_h + g(\boldsymbol{\mu}_n - \boldsymbol{\mu}_x - \boldsymbol{\mu}_h) \\ \boldsymbol{\Sigma}_y &\approx \boldsymbol{A}\boldsymbol{\Sigma}_x\boldsymbol{A}^\mathsf{T} + \boldsymbol{A}\boldsymbol{\Sigma}_h\boldsymbol{A}^\mathsf{T} + (\boldsymbol{I} - \boldsymbol{A})\boldsymbol{\Sigma}_n(\boldsymbol{I} - \boldsymbol{A})^\mathsf{T} \end{aligned} \tag{3.38}$$

assuming diagonal covariance matrices.

The results in equation 3.38 show how the mean is shifted non-linearly by the noise. The matrix $\boldsymbol{A}$ varies proportionally with the level of noise [48], increasing in magnitude towards identity with greater noise. When the noise is dominant, the variances of the model take on the variance of the noise. A 0th-order VTS approximation is sufficient to represent the affect of the environment on the mean, but a 1st order VTS approximation is needed to reflect the reduced variance. It has been observed in Monto Carlo simulations in the log-spectrum that 1st-order VTS provides a better approximation to the mean and variance of the corrupted distribution than log-normal PMC [2].

3.3.6 Algonquin

Uncertainty decoding, as described in [38], attempts to directly compute state conditional likelihoods using variational methods. Gaussian mixture models are used to model both the clean speech, noise and channel distributions. The posterior of the joint distribution of the clean speech, noise, channel and state variables are approximated by a simpler parameterised distribution, which is also a GMM. The variational parameters of this simpler distribution are optimised per frame in an iterative EM fashion. While the results are promising, Algonquin requires the update of every component of every state in the acoustic model using computationally expensive forms, iteratively obtaining linear Gaussian approximations of the posterior. This makes it as computationally expensive as other model compensation schemes. A different from of Algonquin has also been derived as a MMSE feature enhancement scheme.

3.4 Summary

There has been much research to solve the problem of ASR in noise. Front-end techniques tend to be computationally efficient, and responsive to changing conditions, but fail after moderate amounts of noise. Model compensation is more powerful, but requires numerous mean and variance parameter updates in the acoustic models at a significant computational cost. Many techniques require a priori knowledge of noise such as stereo-databases, MMSE techniques, and forms of PMC. For general noise robustness, such assumptions seem counter-intuitive since noise is inherently unknown. For unseen conditions, these schemes are not applicable and degrade.

Thus techniques that can either adequately adapt or predict and compensate for noise conditions or systems that are inherently immune to noise, such as robust parameterisations or models are of great interest. While some inherently robust front-ends have shown promise, they typically degrade under medium levels of noise. Unsupervised techniques to precisely estimate the noise and associated schemes to accurately compensate for it show promise in viably operating ASR in unknown, adverse environments.

Chapter 4

Uncertainty Decoding

The term "uncertainty" has been loosely applied in a variety of contexts to various robustness techniques for ASR. In this work, the concept of uncertainty decoding is distinct from the soft-information paradigm presented in the Algonquin framework [38, 39], uncertain observation decoding [6, 7] and the notion of uncertainty in missing feature theory [5, 10].

The goal of uncertainty decoding is to achieve a fast compensation scheme under the assumption that estimates of clean speech are not exact in noise. Intuitively, the presence of noise in the environment results in uncertainty of the true value of the clean speech, thus the variance of the model increases with the noise. This implies model compensation, however ways of decoupling the front-end processing from the acoustic model to increase efficiency are explored. In this chapter, uncertainty decoding is presented in a rigorous framework with key approximations noted and contrasted with previous work. Once the theoretical basis has been presented, various modeling aspects are discussed and two forms of uncertainty decoding advanced.

4.1 Theoretical Framework

Formally, uncertainty decoding begins from the Bayesian inference of the optimal state sequence given the graph in figure 2.3. This requires marginalising out the latent hidden clean speech and noise variables. Recall equation 2.9 where the likelihood of the corrupted observation sequence is dependent on models of the clean speech and noise

$$p(\boldsymbol{Y}|\mathcal{M},\tilde{\mathcal{M}}) \approx \sum_{\boldsymbol{\theta},\boldsymbol{\theta}^n \in \boldsymbol{\Theta}} P(\boldsymbol{\theta}|\mathcal{M})P(\boldsymbol{\theta}^n|\tilde{\mathcal{M}}) \prod_{t=1}^{T} \int_{2\mathcal{R}^d} p(\boldsymbol{y}_t|\boldsymbol{x}_t,\boldsymbol{n}_t)p(\boldsymbol{x}_t|\mathcal{M},\theta_t)p(\boldsymbol{n}_t|\tilde{\mathcal{M}},\theta_t^n)d\boldsymbol{x}_t d\boldsymbol{n}_t$$

The double integration can be simplified by assuming a certain noise condition captured in a front-end model $\check{\mathcal{M}}$. That is

$$\int_{\mathcal{R}^d} p(\boldsymbol{y}_t|\boldsymbol{x}_t,\boldsymbol{n}_t)\, p(\boldsymbol{n}_t|\tilde{\mathcal{M}},\theta_t^n)d\boldsymbol{n}_t \approx p\big(\boldsymbol{y}_t|\boldsymbol{x}_t,\check{\mathcal{M}}\big) \tag{4.1}$$

This allows the likelihood calculation of this conditional to depend on a simple front-end noise model, which may also be made independent of the acoustic model complexity. Since the conditional is computed in the front-end, this minimises the computational load the compensation process has on search. Contrast this with computationally three-dimensional decoding using models of different noises $\tilde{\mathcal{M}}$ and clean speech. Given this, the final decoding likelihood is approximated by

$$p(\boldsymbol{Y}|\mathcal{M}, \check{\mathcal{M}}) \approx \sum_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} P(\boldsymbol{\theta}|\mathcal{M}) \prod_{t=1}^{T} \int_{\mathcal{R}^d} p(\boldsymbol{y}_t|\boldsymbol{x}_t, \check{\mathcal{M}})\, p(\boldsymbol{x}_t|\mathcal{M}, \theta_t)\, d\boldsymbol{x}_t \tag{4.2}$$

The solution to the integral is of course highly dependent on the form of the two parts of the integrand. If the corrupted speech conditional, which is dependent on the noise, is deterministic, then the integral resolves to the clean speech model prior. However, if the conditional takes a Gaussian distributed form, then the integral is also a Gaussian distribution with a variance that is the sum of the variances of the two parts of the integrand. Algonquin approximates the integral for each component of each model using expensive iterative variational estimation. Scaling this approach to LVCSR is not straightforward. In contrast, the focus of this work is to find practical forms of $p(\boldsymbol{y}_t|\boldsymbol{x}_t, \check{\mathcal{M}})$ such that tractable solutions are available. In this framework, uncertainty decoding can be interpreted as passing the corrupted condition density function to the decoder as shown in figure 4.1. The form of the conditional distribution should be efficient to compute like feature enhancement schemes, yet minimise the cost of updating the acoustic model parameters.

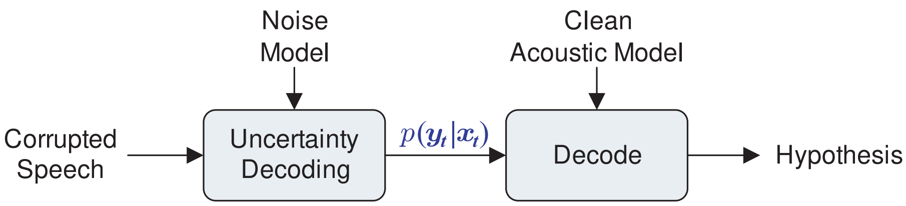

Figure 4.1: Feature compensation with uncertainty decoding

4.2 The Conditional Corrupted Speech Distribution

Given that the clean speech prior in equation 4.2 is usually a GMM in most recognisers, the focus of uncertainty decoding is to find an efficient and accurate representation of the conditional corrupted speech distribution. The main difficulty is that $p(\boldsymbol{y}_t|\boldsymbol{x}_t, \check{\mathcal{M}})$ is highly complex as seen through a numerical simulation of the joint clean and corrupted speech distribution in figure 4.2 using the same equation from section 2.2

$$y = \log(\exp(n) + \exp(x))$$

but now with x as a point input variable and the noise again generated from a Gaussian distribution with a mean 1 on the left, and 4 on the right and the variance constant at 1. The joint distribution is highly non-linear and difficult to characterise parametrically.

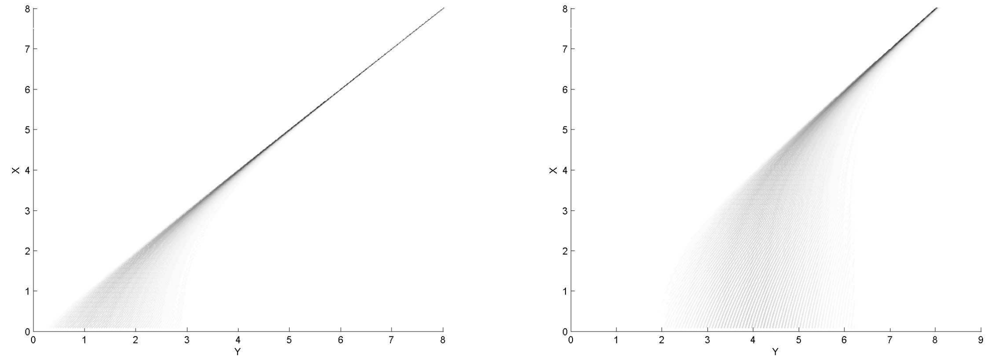

Figure 4.2: Joint clean and corrupted speech distribution with a noise source of mean 1 (left) and mean 4 (right), both with a variance of 1

It is useful to examine how the conditional distribution shifts and skews as a function of the clean speech through the same simulation. Normalised histograms in figure 4.3 show the distribution varying from a relatively Gaussian distribution, matching the noise source, when the noise mean is comparable to the clean speech mean, to a strongly deterministic one, as the difference between the speech and noise means increases.

Observe that at a high SNR, where the corrupted speech is approximately equal to the clean, the conditional distribution of the corrupted speech is relatively deterministic, whereas for a low SNR it is roughly Gaussian approaching the distribution of the noise. Thus the effective local form of the conditional corrupted speech varies significantly depending on the clean speech prior. From figures 4.2 and 4.3 it is clear that approximating the conditional with a constant density function independent of the clean speech would be poor.

4.3 Gaussian Mixture Model Approximations

A standard approach to modeling complex distributions is to use a mixture model

$$p\big(\boldsymbol{y}_t|\boldsymbol{x}_t,\check{\mathcal{M}}\big)=\sum_{n=1}^{N}P\big(\check{s}_n|\boldsymbol{x}_t,\check{\mathcal{M}}\big)\,p\big(\boldsymbol{y}_t|\boldsymbol{x}_t,\check{\mathcal{M}},\check{s}_n\big) \tag{4.3}$$

The final decoding likelihood, equation 4.2, involves a marginalisation of two distributions – the conditional corrupted speech and the clean speech model. As the clean speech model is typically a Gaussian mixture model, if the conditional corrupted speech distribution is also Gaussian then deriving an analytical form is trivial since the integration becomes the convolution of two Gaussians. Therefore represent the conditional as a Gaussian distribution

$$p\big(\boldsymbol{y}_t|\boldsymbol{x}_t,\check{\mathcal{M}},\check{s}_n\big)=\mathcal{N}(\boldsymbol{y}_t;f_\mu(\boldsymbol{x}_t,\check{s}_n),f_\Sigma(\boldsymbol{x}_t,\check{s}_n)) \tag{4.4}$$

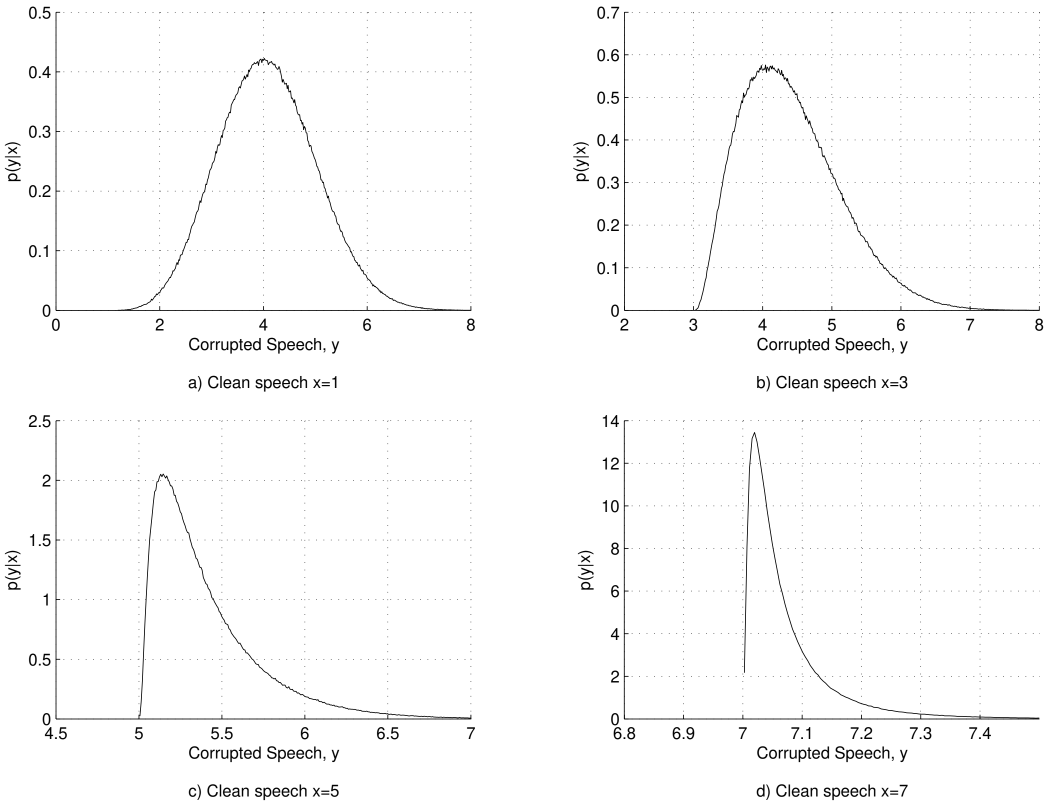

Figure 4.3: Conditional corrupted speech distribution with a noise source of mean 4, variance 1

With this form of the conditional distribution of the corrupted speech there are several issues:

- the component posterior $P(\check{s}_n|\boldsymbol{x}_t, \check{\mathcal{M}})$ is conditional on the clean speech;
- the number of component in the front-end affects the total number of effective components evaluated;
- and determining the form of the component conditional distribution $p(\boldsymbol{y}_t|\boldsymbol{x}_t, \check{\mathcal{M}}, \check{s}_n)$.

The component posterior is conditional on the hidden "clean speech" variable which depends entirely on the state of the clean speech model. Ideally the front-end compensation should be as independent of the acoustic models as possible. Directly using a GMM requires the marginalisation of each Gaussian in the front-end with each in the acoustic model. Effectively, this multiplies the number of components in the system by the number in the GMM which greatly increases the computational cost. These issues can be overcome by some approximations as discussed in the sub-sections where two different forms of the conditional corrupted speech distribution are presented.

4.3.1 SPLICE Form

The conditional corrupted speech distribution can be transformed to the clean speech posterior, through the application of Bayes' rule as suggested in [38]. This yields the following form of the conditional corrupted speech posterior

$$p\big(\boldsymbol{y}_t|\boldsymbol{x}_t,\check{\mathcal{M}}\big) = \frac{\sum_{n=1}^{N} p\big(\boldsymbol{x}_t|\boldsymbol{y}_t,\check{s}_n,\check{\mathcal{M}}\big)\, p\big(\boldsymbol{y}_t|\check{s}_n,\check{\mathcal{M}}\big)\, \check{c}_n}{p\big(\boldsymbol{x}_t|\check{\mathcal{M}}\big)} \tag{4.5}$$

where the clean speech posterior and corrupted speech model are Gaussian mixture models of n components, and weighted by $\check{c}_n$. Many different forms of the clean speech posterior using a GMM have been investigated such as CDCN, POF, and VTS. Here, the `SPLICE` MMSE estimate is examined [15] as reviewed in section 3.2.5.

Modeling the denominator with a GMM would make marginalisation complex, thus a simple, single Gaussian approximation can be used instead. This is a rather crude assumption as a single Gaussian does not represent the clean speech distribution well, hence the common use of multiple Gaussian distributions in current speech recognisers. Nevertheless, a single clean speech Gaussian is used

$$p\big(\boldsymbol{x}_t|\check{\mathcal{M}}\big) \approx \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x) \tag{4.6}$$

where the parameters are estimated from the corrupted speech GMM, compensated using the SPLICE parameters, and the individual components combined

$$\boldsymbol{\mu}_x = \sum_{n=1}^{N} \check{c}_n \left(\boldsymbol{\mu}_y^{(n)} + \check{\boldsymbol{\mu}}^{(n)}\right) \tag{4.7}$$

$$\boldsymbol{\Sigma}_x = \sum_{n=1}^{N} \check{c}_n \left(\boldsymbol{\mu}_y^{(n)}\boldsymbol{\mu}_y^{(n)\mathsf{T}} + \check{\boldsymbol{\mu}}^{(n)}\check{\boldsymbol{\mu}}^{(n)\mathsf{T}} + \boldsymbol{\Sigma}_y^{(n)} + \check{\boldsymbol{\Sigma}}^{(n)}\right) - \boldsymbol{\mu}_x\boldsymbol{\mu}_x^{\mathsf{T}} \tag{4.8}$$

Given the `SPLICE` form of the clean speech posterior, a front-end GMM modeling the corrupted speech and the simplified denominator, and diagonal covariance matrices as in [15], the conditional takes the form

$$p\big(\boldsymbol{y}_t|\boldsymbol{x}_t,\check{\mathcal{M}}\big) = \frac{\sum_{n=1}^{N} p\big(\boldsymbol{x}_t|\boldsymbol{y}_t,\check{s}_n,\check{\mathcal{M}}\big)\, p\big(\boldsymbol{y}_t|\check{s}_n,\check{\mathcal{M}}\big)\, \check{c}_n}{p\big(\boldsymbol{x}_t|\check{\mathcal{M}}\big)} \tag{4.9}$$

$$\approx \frac{\sum_{n=1}^{N} \mathcal{N}\Big(\boldsymbol{x}_t; \boldsymbol{y}_t + \check{\boldsymbol{\mu}}^{(n)}, \check{\boldsymbol{\Sigma}}^{(n)}\Big)\, p\big(\boldsymbol{y}_t|\check{s}_n,\check{\mathcal{M}}\big)\, \check{c}_n}{\mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)} \tag{4.10}$$

$$= \sum_{n=1}^{N} p\big(\boldsymbol{y}_t|\check{s}_n,\check{\mathcal{M}}\big)\, \check{c}_n \alpha^{(n)} \mathcal{N}\Big(\boldsymbol{A}^{(n)}\boldsymbol{y}_t + \boldsymbol{b}^{(n)}; \boldsymbol{x}_t, \hat{\boldsymbol{\Sigma}}_x^{(n)}\Big) \tag{4.11}$$

The following are the elements of the diagonal matrix $\boldsymbol{A}^{(n)}$ and vector $\boldsymbol{b}^{(n)}$ and the associated uncertainty variance $\hat{\boldsymbol{\Sigma}}_x^{(n)}$

$$
\begin{aligned}
a_{ii}^{(n)} =& \frac{\sigma_{xi}^2}{\sigma_{xi}^2 - \breve{\sigma}_i^{(n)2}} \\
b_i^{(n)} =& a_{ii}^{(n)} \left(\breve{\mu}_i^{(n)} - \frac{\breve{\sigma}_i^{(n)2}}{\sigma_{xi}^{(n)2}} \mu_{xi} \right) \\
\hat{\boldsymbol{\Sigma}}_x^{(n)} =& \boldsymbol{A}^{(n)} \breve{\boldsymbol{\Sigma}}^{(n)}
\end{aligned}
\tag{4.12}
$$

and $\breve{\mu}_i^{(n)}$ and $\breve{\sigma}_i^{(n)2}$ the compensation parameters from standard `SPLICE` enhancement. See appendix A for a more detailed derivation and definition of $\alpha^{(n)}$. The denominator of the $a_{ii}^{(n)}$ term in equation 4.12 should be forced to remain positive. As in [15], this can be done by constraining the simplified clean speech variance to be greater than the uncertainty of the estimation by some factor

$$
\sigma_{xi}^2 \geq \breve{\sigma}_i^{(n)2} + \epsilon \tag{4.13}
$$

The system was found to be fairly insensitive to the value of ϵ however a value of ten percent of the global clean speech variance was used. In the limit, when ϵ is very large, the uncertainty aspect is ignored, returning processing to the standard `SPLICE` enhancement scheme.

With this form of the conditional, the integral from equation 4.2 becomes

$$
p\big(\boldsymbol{y}_t|\mathcal{M}, \breve{\mathcal{M}}, \theta_t\big) = \int_{\mathcal{R}^d} p\big(\boldsymbol{y}_t|\boldsymbol{x}_t, \breve{\mathcal{M}}\big)\, p(\boldsymbol{x}_t|\mathcal{M}, \theta_t)\, d\boldsymbol{x}_t \tag{4.14}
$$

$$
\approx \int_{\mathcal{R}^d} \sum_{n=1}^{N} \breve{c}_n p\big(\boldsymbol{y}_t|\breve{s}_n, \breve{\mathcal{M}}\big)\, \alpha^{(n)} \mathcal{N}\Big(\boldsymbol{A}^{(n)}\boldsymbol{y}_t + \boldsymbol{b}^{(n)}; \boldsymbol{x}_t, \hat{\boldsymbol{\Sigma}}_x^{(n)}\Big) \times \sum_{m\in\theta_t} c_m \mathcal{N}\Big(\boldsymbol{x}_t; \boldsymbol{\mu}^{(m)}, \boldsymbol{\Sigma}^{(m)}\Big)\, d\boldsymbol{x}_t \tag{4.15}
$$

$$
= \sum_{n=1}^{N} \sum_{m\in\theta_t} \breve{c}_n c_m p\big(\boldsymbol{y}_t|\breve{s}_n, \breve{\mathcal{M}}\big)\, \alpha^{(n)} \times \int_{\mathcal{R}^d} \mathcal{N}\Big(\boldsymbol{A}^{(n)}\boldsymbol{y}_t + \boldsymbol{b}^{(n)}; \boldsymbol{x}_t, \hat{\boldsymbol{\Sigma}}_x^{(n)}\Big) \mathcal{N}\Big(\boldsymbol{x}_t; \boldsymbol{\mu}^{(m)}, \boldsymbol{\Sigma}^{(m)}\Big)\, d\boldsymbol{x}_t \tag{4.16}
$$

$$
= \sum_{n=1}^{N} \sum_{m\in\theta_t} \breve{c}_n c_m p\big(\boldsymbol{y}_t|\breve{s}_n, \breve{\mathcal{M}}\big) \alpha^{(n)} \mathcal{N}\Big(\boldsymbol{A}^{(n)}\boldsymbol{y}_t + \boldsymbol{b}^{(n)}; \boldsymbol{\mu}^{(m)}, \boldsymbol{\Sigma}^{(m)} + \hat{\boldsymbol{\Sigma}}_x^{(n)}\Big) \tag{4.17}
$$

With the two parts of the integrand being Gaussian distributions, the integral simplifies into a single Gaussian distribution[1]. It is clear from this form, that the number of components in the front-end GMM directly multiplies the number of effective components in the acoustic model. As suggested in [15] and with standard `SPLICE` enhancement, this can be avoided by using the *h*ard approximation in using only the most probable component $\breve{s}_{n^*}$'s compensation parameters

$$
\breve{s}_{n^*} = \arg\max_{\breve{s}_n} \Big[\breve{c}_n p\big(\boldsymbol{y}_t|\breve{s}_n, \breve{\mathcal{M}}\big)\Big]
$$

[1] See appendix B for derivation.

where n^* is the index of the most probable component. This allows the effective number of components per state in the acoustic model to remain unchanged, resulting in a more computationally efficient form

$$p\big(\boldsymbol{y}_t|\mathcal{M},\check{\mathcal{M}},\theta_t\big) \approx \sum_{m\in\theta_t} c_m \mathcal{N}\Big(\hat{\boldsymbol{\mu}}_x^{(n^*)};\boldsymbol{\mu}^{(m)},\boldsymbol{\Sigma}^{(m)}+\hat{\boldsymbol{\Sigma}}_x^{(n^*)}\Big) \tag{4.18}$$

As a result, the complexity of the front-end processing is independent of the model; the number of Gaussian evaluations in the decoder is not affected by the number of components in the front-end GMM. The terms $\check{c}_n$, $p\big(\boldsymbol{y}_t|\check{s}_n,\check{\mathcal{M}}\big)$ and $\alpha^{(n)}$ do not actually need to be included in the Gaussian evaluations during decoding since these are the same for every model and evaluation during a frame.

4.3.2 Joint Distribution Form

Another approach to modeling the corrupted speech component conditional distribution is to use a joint distribution of the clean and corrupted speech to derive the conditional for each region of the acoustic space. To resolve the issue of the component posterior, it can be approximated by making it conditional upon the corrupted speech

$$P\big(\check{s}_n|\boldsymbol{x}_t,\check{\mathcal{M}}\big) \approx P\big(\check{s}_n|\boldsymbol{y}_t,\check{\mathcal{M}}\big) \tag{4.19}$$

This has a very coarse effect of passing the same component, and thus conditional distribution, to the decoder, regardless of the state in the clean model. As seen from 4.3, the conditional should have a smaller variance for some models, and larger variance for others more confusable by the noise.

If the corrupted speech is modeled by a single GMM

$$p\big(\boldsymbol{y}_t|\check{\mathcal{M}}\big) \approx \sum_{n=1}^{N} P\big(\check{s}_n|\boldsymbol{y}_t,\check{\mathcal{M}}\big)\,\mathcal{N}\Big(\boldsymbol{y}_t;\boldsymbol{\mu}_y^{(n)},\boldsymbol{\Sigma}_y^{(n)}\Big) \tag{4.20}$$

then the component posterior is given by

$$P(\check{s}_n|\boldsymbol{y}_t,\check{\mathcal{M}}) = \frac{\check{c}_n p(\boldsymbol{y}_t|\check{s}_n)}{\sum_{i=1}^{N}\check{c}_i p(\boldsymbol{y}_t|\check{s}_i)} \tag{4.21}$$

Like POF and `SPLICE` , the acoustic space is partitioned by the front-end GMM, and a conditional distribution estimated for each component. Next the form of equation 4.4 can be derived using the joint distribution of the clean and corrupted speech, which can be readily estimated using stereo data

$$\begin{bmatrix} \boldsymbol{y}_t \\ \boldsymbol{x}_t \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_y^{(n)} \\ \boldsymbol{\mu}_x^{(n)} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_y^{(n)} & \boldsymbol{\Sigma}_{yx}^{(n)} \\ \boldsymbol{\Sigma}_{xy}^{(n)} & \boldsymbol{\Sigma}_x^{(n)} \end{bmatrix}\right) \tag{4.22}$$

the conditional distribution can be derived from the joint[1]

$$p(\boldsymbol{y}_t|\boldsymbol{x}_t, \check{\mathcal{M}}, \check{s}_n) \approx \mathcal{N}\Big(\boldsymbol{y}_t; \boldsymbol{\mu}_y^{(n)} + \boldsymbol{\Sigma}_{yx}^{(n)}\boldsymbol{\Sigma}_x^{(n)\text{-}1}\Big(\boldsymbol{x}_t - \boldsymbol{\mu}_x^{(n)}\Big), \boldsymbol{\Sigma}_y^{(n)} - \boldsymbol{\Sigma}_{yx}^{(n)}\boldsymbol{\Sigma}_x^{(n)\text{-}1}\boldsymbol{\Sigma}_{xy}^{(n)}\Big) \tag{4.23}$$

$$= \alpha^{(n)}\mathcal{N}\Big(\boldsymbol{\Sigma}_x^{(n)}\boldsymbol{\Sigma}_{yx}^{(n)\text{-}1}\Big(\boldsymbol{y}_t - \boldsymbol{\mu}_y^{(n)}\Big) + \boldsymbol{\mu}_x^{(n)}; \boldsymbol{x}_t, \boldsymbol{\Sigma}_x^{(n)}\boldsymbol{\Sigma}_{yx}^{(n)\text{-}1}\boldsymbol{\Sigma}_y^{(n)}\boldsymbol{\Sigma}_x^{(n)}\boldsymbol{\Sigma}_{yx}^{(n)\text{-}1} - \boldsymbol{\Sigma}_x^{(n)}\Big) \tag{4.24}$$

$$= \alpha^{(n)}\mathcal{N}\Big(\boldsymbol{A}^{(n)}\boldsymbol{y}_t + \boldsymbol{b}^{(n)}; \boldsymbol{x}_t, \hat{\boldsymbol{\Sigma}}_x^{(n)}\Big) \tag{4.25}$$

where $\alpha^{(n)} = \left|\boldsymbol{A}^{(n)}\right|$. Thus the transform of feature vector and the uncertainty are

$$\begin{aligned} \boldsymbol{A}^{(n)} &= \boldsymbol{\Sigma}_x^{(n)}\boldsymbol{\Sigma}_{yx}^{(n)\text{-}1} \\ \boldsymbol{b}^{(n)} &= \boldsymbol{\mu}_x^{(n)} - \boldsymbol{A}^{(n)}\boldsymbol{\mu}_y^{(n)} \\ \hat{\boldsymbol{\Sigma}}_x^{(n)} &= \boldsymbol{A}^{(n)}\boldsymbol{\Sigma}_y^{(n)}\boldsymbol{A}^{(n)\mathsf{T}} - \boldsymbol{\Sigma}_x^{(n)} \end{aligned} \tag{4.26}$$

These can be compared to the forms of equation 4.12 where explicit flooring of the variances is required and diagonalised covariance matrices necessary.

With the approximations for the component posterior and Gaussian form of the component conditional corrupted speech distribution, the decoding likelihood from equation 4.2 becomes

$$p(\boldsymbol{y}_t|\mathcal{M}, \check{\mathcal{M}}, \theta_t) = \int_{\mathcal{R}^d} p(\boldsymbol{y}_t|\boldsymbol{x}_t, \check{\mathcal{M}})\, p(\boldsymbol{x}_t|\mathcal{M}, \theta_t)\, d\boldsymbol{x}_t \tag{4.27}$$

$$\approx \int_{\mathcal{R}^d} \sum_{n=1}^{N} \check{c}_n P(\check{s}_n|\boldsymbol{y}_t, \check{\mathcal{M}})\, \alpha^{(n)}\mathcal{N}\Big(\boldsymbol{A}^{(n)}\boldsymbol{y}_t + \boldsymbol{b}^{(n)}; \boldsymbol{x}_t, \hat{\boldsymbol{\Sigma}}_x^{(n)}\Big) \times \tag{4.28}$$

$$= \sum_{n=1}^{N}\sum_{m\in\theta_t} \check{c}_n c_m P(\check{s}_n|\boldsymbol{y}_t, \check{\mathcal{M}})\alpha^{(n)}\mathcal{N}\Big(\boldsymbol{A}^{(n)}\boldsymbol{y}_t + \boldsymbol{b}^{(n)}; \boldsymbol{\mu}^{(m)}, \boldsymbol{\Sigma}^{(m)} + \hat{\boldsymbol{\Sigma}}_x^{(n)}\Big) \tag{4.29}$$

Again, the marginalisation of two Gaussian distributions yields a single distribution[2]. Note the similarity to the `SPLICE` with uncertainty decoding equation 4.17. Using a similar approximation in the `SPLICE` formulation, a most probable component can be used in place of the weighted sum

$$\check{s}_{n^*} = \arg\max_{\check{s}_n}\Big[P(\check{s}_n|\boldsymbol{y}_t, \check{\mathcal{M}})\Big] \tag{4.30}$$

This yields the same form of likelihood calculation as derived in the `SPLICE` form that is equation 4.18, but with forms of the transforms in equation 4.26. Figure 4.4 demonstrates the operation of the `SPLICE` and `Joint` forms of uncertainty decoding. It shows how the compensation parameters are computed in the front-end and how they are used during decoding.

[1]See appendix C for more details.
[2]See appendix B for derivation.

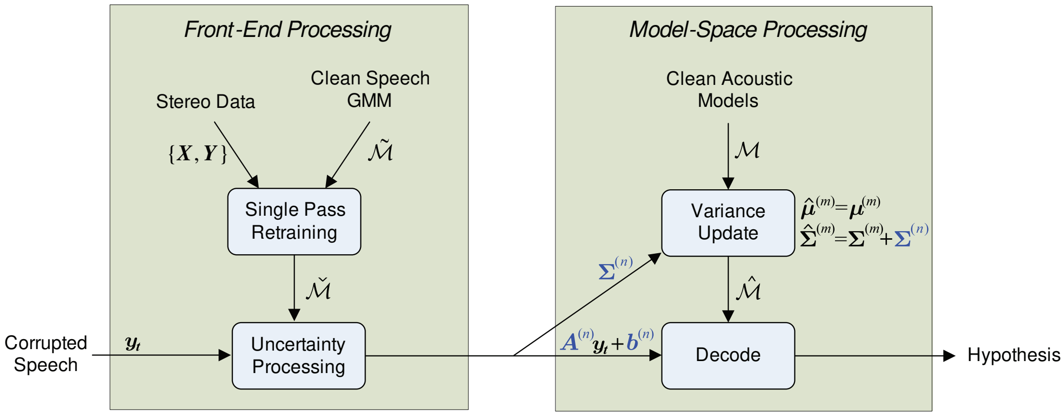

Figure 4.4: Uncertainty decoding processing

4.4 Model-Space Uncertainty Transforms

One crude approximation made was to use the corrupted speech component posterior rather than the clean as stated in equation 4.19. A way to explore the impact of this is to contrast this with a model-based approach where the form of the component conditional corrupted speech distribution is dependent on the clean speech model component, instead of the observed corrupted speech. One method of doing so, is to cluster the model components with a GMM. Instead of passing a single Gaussian chosen by the front-end, a GMM is embedded in the acoustic model which groups the model components. For each region a conditional distribution is estimated based on the joint distribution of clean and corrupted speech and the approximation in equation 4.19 is not required. Specifically, represent the clean acoustic space by a GMM

$$p(\boldsymbol{x}_t|\check{\mathcal{M}}) \approx \sum_{n=1}^{N} P\big(\check{s}_n|\boldsymbol{x}_t,\check{\mathcal{M}}\big)\,\mathcal{N}\Big(\boldsymbol{x}_t;\boldsymbol{\mu}_x^{(n)},\boldsymbol{\Sigma}_x^{(n)}\Big) \tag{4.31}$$

where the conditional corrupted speech distribution is still based on the joint distribution of the clean and corrupted speech, again possibly trained on stereo data, but with the clean speech posterior partitioning the acoustic space. The observation likelihood of a given state with a "hard" approximation is then

$$p\big(\boldsymbol{y}_t|\mathcal{M},\check{\mathcal{M}},\theta_t\big) = \sum_{m\in\theta_t} c_m P\Big(\check{s}_{n^*}|\boldsymbol{\mu}^{(m)},\check{\mathcal{M}}\Big)\alpha^{(n^*)}\mathcal{N}\Big(\boldsymbol{A}^{(n^*)}\boldsymbol{y}_t+\boldsymbol{b}^{(n^*)};\boldsymbol{\mu}^{(m)},\boldsymbol{\Sigma}^{(m)}+\hat{\boldsymbol{\Sigma}}_x^{(n^*)}\Big) \tag{4.32}$$

During decoding, $p(\boldsymbol{y}_t|\boldsymbol{x}_t,\check{\mathcal{M}},\check{s}_{n^*})$ and therefore parameters $\alpha^{(n^*)}$, $\boldsymbol{A}^{(n^*)}$, $\boldsymbol{b}^{(n^*)}$ and $\hat{\boldsymbol{\Sigma}}_x^{(n^*)}$ vary by the model component mean $\boldsymbol{\mu}^{(m)}$ and are chosen in the following manner

$$\check{s}_{n^*} = \arg\max_{\check{s}_n}\Big[P\Big(\check{s}_n|\boldsymbol{\mu}^{(m)},\check{\mathcal{M}}\Big)\Big] \tag{4.33}$$

Partitioning the model-space with a GMM requires N evaluations of the posterior per component in the acoustic model. This is a form of model-compensation since n^* is not a function of the acoustic signal. The conditional distribution does not change for a component in the acoustic model with time; it varies only with the noise model $\check{\mathcal{M}}$ and each component in the model is individually compensated. This gives an upper bound on the performance obtainable by using a GMM approximation to the conditional corrupted speech distribution.

An alternate way of examining this is to group the components using a regression tree as applied with MLLR transforms and described in section 3.3.3. Model components are grouped into classes according to acoustic similarity. For each class, a joint distribution of the clean and corrupted speech can be estimated, and thus a conditional corrupted speech distribution determined. In this form, the marginalisation of the conditional corrupted speech with the clean speech prior could be considered a transform. For example, with the `Joint` form applied to different model classes

$$p\big(\boldsymbol{y}_t|\boldsymbol{\theta}_t, \check{\mathcal{M}}\big) = \sum_{m\in\theta_t} c_m \left|\boldsymbol{A}^{(r_m)}\right| \mathcal{N}\Big(\boldsymbol{A}^{(r_m)}\boldsymbol{y}_t + \boldsymbol{b}^{(r_m)}; \boldsymbol{\mu}^{(m)}, \boldsymbol{\Sigma}^{(m)} + \boldsymbol{\Sigma}_x^{(r_m)}\Big) \tag{4.34}$$

where $\check{\mathcal{M}} = \Big\{\mathbf{A}^{(1)}, \mathbf{b}^{(1)}, \boldsymbol{\Sigma}_x^{(1)}, \ldots, \mathbf{A}^{(R)}, \mathbf{b}^{(R)}, \boldsymbol{\Sigma}_x^{(R)}\Big\}$, r_m denotes the model class, and R the total number of classes. Contrast this with constrained MLLR

$$p\big(\boldsymbol{y}_t|\boldsymbol{\theta}_t, \check{\mathcal{M}}\big) = \sum_{m\in\theta_t} c_m \left|\boldsymbol{A}^{(r_m)}\right| \mathcal{N}\Big(\boldsymbol{A}^{(r_m)}\boldsymbol{y}_t + \boldsymbol{b}^{(r_m)}; \boldsymbol{\mu}^{(m)}, \boldsymbol{\Sigma}^{(m)}\Big) \tag{4.35}$$

CMLLR is an efficient form of model adaptation. The `Joint` form is similar, but with a uncertainty offset to the model variances. Typically, the transforms in CMLLR are based on adaptation data, which can be limited. The noise models in the `Joint` form have been described as being estimated from a large amount of stereo data but need not be as discussed in the following chapter. Figure 4.5 shows the general method for estimating the compensation parameters for a regression class. With a model of the clean speech per class, a joint distribution can be estimated per class with only an estimate of the noise statistics and not actual corrupted speech data. Since the models are grouped according to acoustic similarity according to the mean, this also provides a method to see if making the conditional corrupted speech distribution dependent on the state in acoustic model would improve performance. Running parallel front-ends to compute the compensated feature vector and the associated uncertainty per model is far more efficient than grouping the model components using a GMM. However, the form in equation 4.32 is theoretically better as the Gaussian evaluation is weighted by the component posterior. This allows different acoustic regions or classes of models to be weighted as well as factoring in the likelihood of how well an acoustic model fits in the region partitioned by the GMM.

4.5 Non-Gaussian Distributions

As demonstrated in figures 4.2 and 4.3, when the speech energy is strong, the conditional corrupted speech distribution is deterministic, and when the energy is low relative to the noise, it is Gaussian.

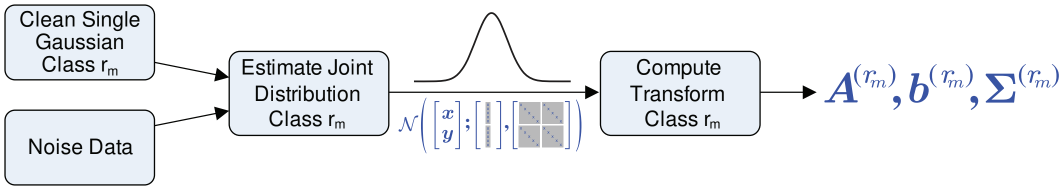

Figure 4.5: Model `Joint` uncertainty decoding

Thus it may be questionable to use a mixture of Gaussians, as in equations 4.3 and 4.4, to represent the conditional corrupted speech distribution where a mixture of another form of distribution would be more appropriate. Alternate forms such as the Weibull or Gamma distribution are more representative than the normal distribution assumed as shown in figure 4.6; the mode is closer and the skewing eliminates the left tail of the distribution.

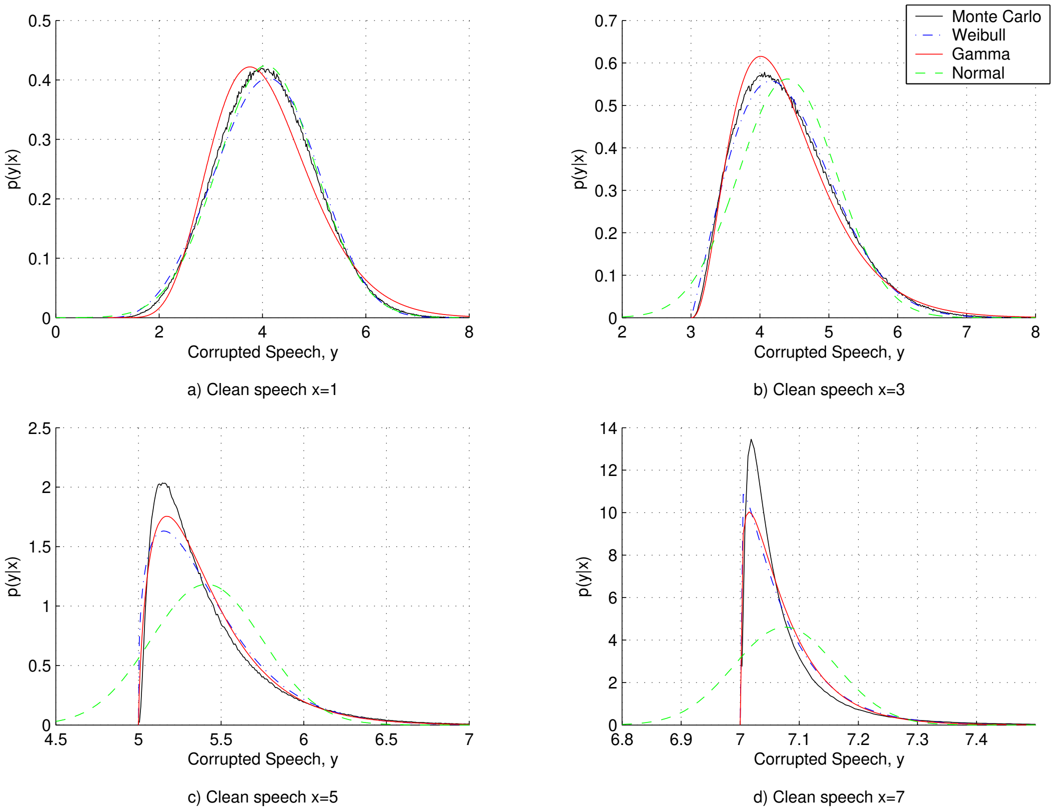

Figure 4.6: Conditional corrupted speech distribution with noise of mean 4, variance 1. Various distributions are fitted to the corrupted speech data.

The formula for the gamma distribution is

$$p(y|x) = \frac{\left(\frac{y-x}{b}\right)^{(a-1)} e^{\left(-\frac{y-x}{b}\right)}}{b\Gamma(a)} \tag{4.36}$$

where

$$\Gamma(a) = \int_0^\infty t^{(a-1)} e^{-t} dt \tag{4.37}$$

and a and b are parameters of the distribution.

The formula for the Weibull distribution is

$$p(y|x) = ab(y-x)^{(b-1)} e^{(-a(y-x)^b)} \tag{4.38}$$

where a and b are parameters of the distribution. Maximum likelihood estimates of the parameters of these skewed distributions can be obtained and mixtures of skewed distributions may be a more accurate representation of the conditional corrupted speech distribution. However analytic solutions of the integral in equation 4.2 are difficult to derive if the conditional takes these forms. This marginalisation is similar to what is required in Bayesian parameter estimation, where a variable has a distribution to be parameterised, and the parameter to be estimated also has a prior distribution. There are natural forms for these distributions that provide manageable solutions, together called conjugate distributions or pairs such as the Gaussian-Gaussian, or the gamma and exponential.

Using numerical integration on simulated data, as shown in figure 4.7 shows that even by using a Weibull distribution instead of a Gaussian, the end resulting distribution varies little. The histogram plots are of the corrupted speech distribution

$$p(y|\theta) = \int_{\mathcal{R}} p(y|x)p(x|\theta)dx \tag{4.39}$$

with $p(y|x)$ as either a constant Weibull distribution or a Gaussian distribution, with optimal parameters given Gaussian distributed noise of mean 4 and variance 1 and Gaussian distributed clean speech $p(x|\theta)$ with mean 5 and variance 1 , as shown in graph c of figure 4.6. The mode is only very slightly shifted with the better fitting Weibull distribution. With the clean speech prior being of a Gaussian form in practically all LVCSR systems, and the minimal observable difference when using a skewed distribution, it makes sense to use a Gaussian representation of the conditional corrupted speech for tractable solutions. The use of a mixture of non-Gaussian distributions does not mitigate the larger problems of the conditional varying on the clean speech, and component selection with mixture models.

4.6 Summary

In this section, uncertainty decoding is formally introduced in the context of the noise robust ASR framework described in section 2.3. Given that the form of the speech prior in recognition systems is typically Gaussian, the research focus of uncertainty decoding is to find tractable and accurate forms of the conditional corrupted speech distribution. While this posterior can be highly

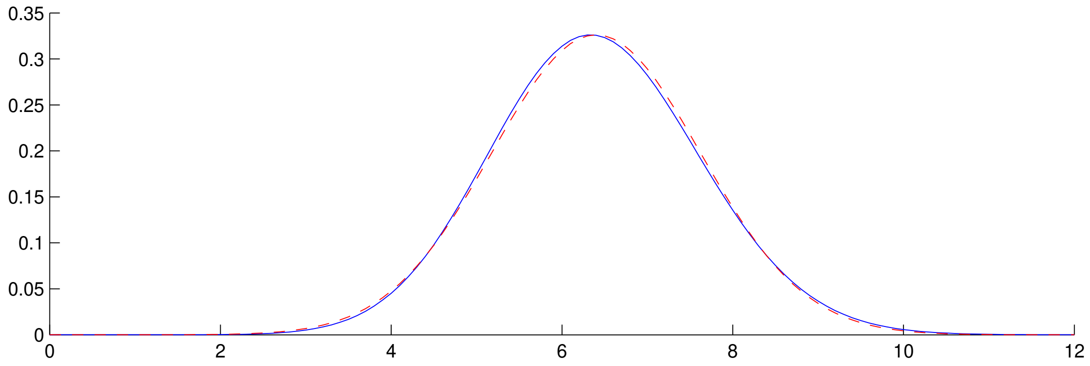

Figure 4.7: Resulting corrupt speech distribution using Weibull form of $p(y|x)$ in dashed red, Gaussian form in solid blue.

skewed, modeled it with a Gaussian form provides a Gaussian result when used with the Gaussian speech prior. Two forms of the conditional distribution of the corrupted speech are presented: `SPLICE` , a derivative of prior work, and a novel `Joint` distribution form. The various assumptions implicit in these two forms are discussed. The uncertainty decoding framework as described has the attributes of decoupling the front-end processing from the acoustic model complexity, yet provides a model variance offset — the uncertainty. This should allow fast feature compensation, with better performance boosted by a simple model variance update.

Chapter 5

Implementation Issues

The investigative work described so far relies on front-end models trained offline using stereo data and SPR for specific conditions. However, a truly robust system flexibly responds to a changing acoustic environment in the same way CMN gracefully handles different channel mismatches. Standard model based techniques can be used to estimate the distributions required in a predictive fashion by combining a model of the noise with a well trained model of the clean speech. Two dominant approaches to do so are Parallel Model Combination (PMC) [24] and Vector Taylor Series (VTS) [48] as described in section 3.3. The effects of using these approximations to the compensation parameters should be investigated and compared to the performance of the ideal SPR baseline.

5.1 Environment Estimation

The first step for an online, adaptive system is to characterise the acoustic environment. Specifically this entails estimating the noise statistics. This is typically done in an iterative fashion using an EM framework as first proposed for this purpose in [48]. The general process is as follows:

1. Initialise estimates of $\boldsymbol{\mu}_n$, $\boldsymbol{\Sigma}_n$ and $\boldsymbol{h}$
2. Expand the 1st order VTS model of the environment in equation 3.35 around initial estimates
3. Perform single step of EM to re-estimate $\boldsymbol{\mu}_n$, $\boldsymbol{\Sigma}_n$ and $\boldsymbol{h}$
4. If the likelihood of the observation has not converged, iterate by going to step 2
5. Estimate the clean speech feature vector

Here $\boldsymbol{\mu}_n$ and $\boldsymbol{\Sigma}_n$ are the statistics of the additive noise and $\boldsymbol{h}$ the mean of the channel noise which assumed to be deterministic without variance. For the alignment stage of the EM step, a GMM could be used to model the clean speech [36]. This allows fast compensation and alignment in the EM step. The re-estimation can take place over a very small number of initial frames [36] or over one or more utterances [46]. Kim et al. achieved very good performance using such little data to estimate the noise statistics. This would indicate that using VTS to continually provide updated noise statistics even within an utterance is viable. Thus research has focused in this area recently with Algonquin using VTS to supply statistics for the variational estimation of the model distributions [19] and a noise-normalised version of `SPLICE` using VTS for the noise normalisation mean [13].

5.2 Parameter Estimation

In the model-based enhancement frameworks, discussed earlier, MMSE based Wiener enhancement is typically used once the clean and noise statistics are determined [17, 44]. For this work, in addition to the clean speech estimate, an uncertainty estimate is required for the front-end processing. Two different combination methods are available: PMC and VTS. In the next sections, estimates of the `Joint` parameters for uncertainty decoding are presented using these techniques.

5.2.1 The Corrupted Speech Distribution

Given an estimate of the noise models and a well trained clean speech GMM, a corrupted speech GMM can be estimated using the 1st-order VTS approximation from equation 3.38. The mean and variance of each component n in the corrupted speech distribution are given by

$$\boldsymbol{\mu}_y^{(n)} \approx \boldsymbol{\mu}_x^{(n)} + g(\boldsymbol{\mu}_n - \boldsymbol{\mu}_x^{(n)}) \tag{5.1}$$

$$\boldsymbol{\Sigma}_y^{(n)} \approx \boldsymbol{A}\boldsymbol{\Sigma}_x^{(n)}\boldsymbol{A}^{\mathsf{T}} + (\boldsymbol{I} - \boldsymbol{A})\boldsymbol{\Sigma}_n(\boldsymbol{I} - \boldsymbol{A})^{\mathsf{T}} \tag{5.2}$$

This ignores the effects of the channel. The corrupted speech distribution could also be obtained by using PMC.

5.2.2 The Cross-Moment

Currently there is no analytic method to compute the covariance of the clean and corrupted speech given the noise statistics and a clean speech model. Thus, a data-driven approach can be take similar to DPMC. Given a log-add approximation using static MFCC parameters, the corrupted speech can be combined with this mismatch function

$$\boldsymbol{y}_t = f(\boldsymbol{x}_t, \boldsymbol{n}_t) = \boldsymbol{C}\left[\exp(\boldsymbol{C}^{-1}\boldsymbol{x}_t) + \exp(\boldsymbol{C}^{-1}\boldsymbol{n}_t)\right] \tag{5.3}$$

where $\boldsymbol{C}$ is the discrete cosine transform matrix. Mismatch functions to compute the dynamic coefficients can be found in [20]. For `Joint` uncertainty decoding, in order to obtain the cross-moment of the clean and corrupted speech, it is necessary to estimate $\check{\boldsymbol{\Sigma}}_{yx}^{(n)}$ which can be computed as

$$\mathcal{E}\left\{\boldsymbol{y}_t\boldsymbol{x}_t^{\mathsf{T}}|\check{s}_n\right\} = \mathcal{E}\left\{\boldsymbol{C}\left[\exp(\boldsymbol{C}^{-1}\boldsymbol{x}_t) + \exp(\boldsymbol{C}^{-1}\boldsymbol{n}_t)\right]\boldsymbol{x}_t^{\mathsf{T}}|\check{s}_n\right\} \tag{5.4}$$

Multiple corrupted speech observation can be generated using the mismatch function above, allowing statistics required for the cross covariance to accumulate for estimation in a data-driven PMC fashion. The log-add approximation itself is very efficient to compute, since the effects of the variances are ignored; however there is a cost in inverse liftering and applying the DCT again since the log-add operation takes place in the spectral domain. Also, estimation of the front-end noise model in this way would be computationally proportional to the number of components in the front-end model since more vectors would need to be drawn to accurately compute each component. The number of sample vectors that need to be computed to provide a good estimate of the cross-moment should be explored especially as it relates to performance.

The incorporation of dynamic estimation of the compensation parameters into the process in figure 4.4 is shown in figure 5.1.

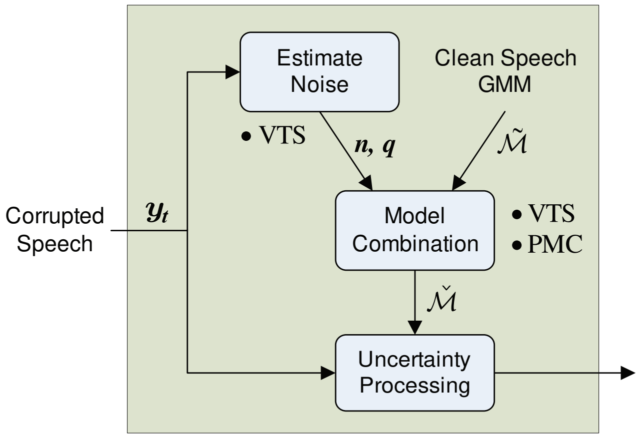

Figure 5.1: Dynamic compensation parameter estimation in the front-end

5.3 Computational Load

The computational cost of any algorithm is always a concern especially in the field of speech recognition with its practical applications on servers where processing power is carefully provisioned or on hand-held devices with minimal capabilities. The aim of uncertainty decoding is to provide robustness at a minimal cost similar to other front-end techniques, rather than examine more extensive model compensation methods. From section 4.3 it was found that both the `SPLICE` and `Joint` forms modified the calculation of the output probability in the decoding in this way

$$p(\boldsymbol{y}_t|\theta_t, \mathcal{M}, \check{\mathcal{M}}) \approx \sum_{m \in \theta_t} c_m \mathcal{N}\left(\hat{\boldsymbol{\mu}}_x^{(n^*)}; \boldsymbol{\mu}^{(m)}, \boldsymbol{\Sigma}^{(m)} + \hat{\boldsymbol{\Sigma}}_x^{(n^*)}\right) \tag{5.5}$$

Typically the cost of the front-end processing to calculate $\hat{\boldsymbol{\mu}}_x^{(n^*)}$ is negligible compared to the Gaussian evaluations in the actual recogniser. This would be the case in the forms of uncertainty decoding introduced here as the feature processing consists of a simple Gaussian calculation and some simple matrix operations. In contrast, the state-based enhancement schemes in section 3.2.2 require decoding of a simple HMM in the front-end.

Once the the feature vector is updated, this is used directly in the decoding process. Thus the main cost in uncertainty decoding is the addition of the uncertainty to the variances in the acoustic models. This involves a sum, division and re-computation of the `GConst` normalisation term in HTK for each Gaussian evaluation. The increase in the model variance should also increase the number of active models being evaluated since the competing acoustic are compressed; this is the equivalent of increasing the pruning threshold. In similar work, it was found that this form of processing increased the recognition time by about 33%.

It would be useful to explore optimisations or alternate forms that achieve the same goal of propagating uncertainty to the recogniser, but with less impact on performance. For example, scaling the variances instead of adding a bias would be more efficient since the `GConst` is then easily computed. It may be possible to compute an optimal variance scale factor given the bias for each front-end component. Since the models are constantly being updated, it would be useful to permanently update the models with the most frequently added bias, and normalise the rest of the values; this would be more effective for low numbers of components in the front-end GMM.

This section discussed techniques to actively estimate the environment and predictively update the front-end model when parallel corpora are not available. Since the estimation is usually iterative, this can be a substantial computational cost and depending on the number of frames required for accurate estimate, introduce a lag for real-time systems. The cost is a function of the form of the front-end model, particularly the number of components N and the number if iterations required for the estimate to converge.

5.4 Summary

The previous chapter introduced uncertainty decoding, and in this chapter, methods to dynamically estimate the front-end model in an unsupervised manner were presented. This is important for practical systems since stereo data is not always readily available and increases the robustness of uncertainty decoding in unknown environments. Environmental estimation usually requires an iterative process to accurately provide the parameters of the noise. Once the noise model is estimated, PMC or VTS can be used to predictively compute the front-end model. Also, the computational cost of uncertainty decoding was discussed. The main overhead in propagating the uncertainty to the recogniser is the variance update and recalculation of the Gaussian normalisation term.

Chapter 6

Preliminary Experimental Results

This chapter presents preliminary results from experiments designed to explore the effectiveness of uncertainty decoding in noisy conditions. The evaluation is based on the medium vocabulary Resource Management (RM) task with artificially added noise. Various baseline and existing noise robustness algorithms are evaluated to provide a contrast with uncertainty decoding. Two forms of uncertainty coding are presented: `SPLICE` with uncertainty and `Joint` distribution uncertainty decoding. The `Joint` uncertainty decoding is examined as an entirely feature-based form or operating with a different for each class of acoustic models. These can be compared to a fast state-of-the-art feature compensation system `SPLICE` and an efficient and powerful model adaptation scheme in CMLLR.

6.1 Resource Management Task

These results are based on the 1000 word naval ARPA Resource Management (RM) database [50] with noise artificially added at the waveform level from the NOISEX-92 database. The clean RM data was recorded in a sound-isolated room using a head mounted Sennheiser HMD414 noise-canceling microphone yielding a high signal-to-noise ratio of 49 dB[1]. The speech was recorded with 16 bit resolution at 20 kHz and down-sampled subsequently to 16kHz. The speaker independent training data for this task consists of 109 speakers reading 3990 sentences of prompted script. The utterances vary in length from about 3 to 5 seconds totaling 3.8 hours of data.

The NATO NOISEX-92 database provides recording samples of various artificial, pedestrian and military noise environments recorded at 20 kHz with 16 bit resolution. The Destroyer Operations Room noise was sampled at random intervals and added to the clean speech data at the waveform level prior to parameterisation. A range of environments were simulated from 32 dB to 8dB SNR. Figure 6.1 shows the affect of the noise on one of the RM sentences "Clear all

[1] The `wavmd` tool from the NIST Speech Quality Assurance Package v2.3 was used to determine the SNR.

windows". The noise itself has a dominant low frequency background hum, an unknown repetitive 6 Hz broadband noise of a machine, and intermittent speech.

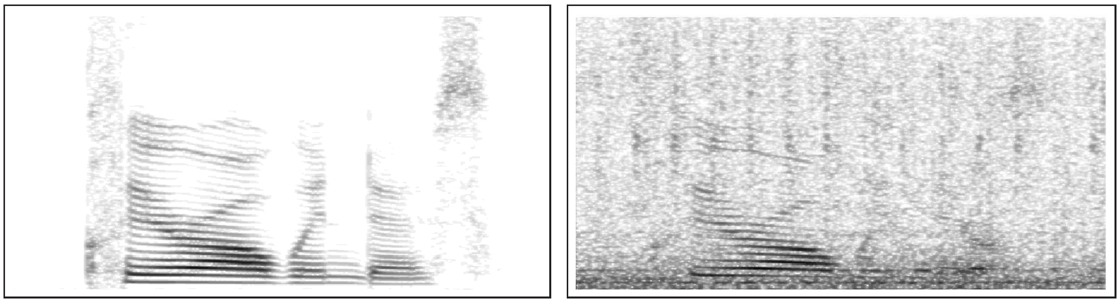

Figure 6.1: Clean spectrum (left) compared to with Operating Room noise at 8 dB SNR (right) "Clear all windows"

The baseline recogniser was built using the RM recipe distributed with HTK [60]. The 39 dimensional feature vector consists of 12 MFCCs appended with the log energy and delta and delta-delta coefficients. The cross-word, state-clustered triphone system with six components per continuous output distribution was used along with a word pair grammar. All results are quoted as an average of three of the four available test sets, `Feb'89`, `Oct'89` and `Feb'91`, unless otherwise stated; the `Sep'92` test data was not used. This gave a total of 30 test speakers and 900 utterances. All decoding experiments were run using this system as the standard RM configuration unless otherwise stated.

The parameters of the front-end GMMs required for some forms of decoding were trained using iterative mixture splitting on either the clean data or artificially corrupted data. At each step the number of components was doubled and then four iterations of Baum-Welch estimation performed. The corresponding corrupted or clean GMM was then trained using SPR with stereo data.

Modifications to the HTK front-end in `HParm.c` were required to support the update of the feature vectors and compute the uncertainty values. In `HModel.c`, the output probability methods were updated to support the variance offset and dynamic recalculation of the `GConst` normalisation term. The BW re-estimation was updated to support the training of the various models and transforms in `HERest.c` and `HFB.c`.

6.2 Baseline Systems Performance

A variety of baselines are presented. First the clean system is evaluated on the clean data, and across a range of SNRs. These results along with simple cepstral normalisation represent a lower baseline compared to the upper bound that a matched system provides.

SNR	Test Set Feb'89	Oct'89	Feb'91	Average
Clean	2.8	3.8	3.2	3.3
32 dB	5.4	6.0	4.7	5.4
26 dB	10.9	11.3	10.2	10.8
20 dB	37.6	32.6	32.2	34.1
14 dB	77.5	79.0	75.0	77.2
8 dB	97.2	97.1	98.4	97.5

Table 6.1: Word error rates (%) of a Clean RM system as SNR decreases

6.2.1 Clean System

Table 6.1 demonstrates the performance of the clean system for a range of SNR. As expected, similar to real conditions, the performance of the baseline, uncompensated system rapidly deteriorates as the level noise increases.

6.2.2 Cepstral Normalisation

Most recognisers will apply CMN and CVN as described in section 3.1.2 for some level of intermediate robustness. In table 6.2 the use of cepstral mean and normalisation variance normalisation is shown. As expected, a simple linear offset of the cepstral mean is effective and additional normalisation of the variance provides gains across all conditions. Also, applying normalisation at the speaker level is more powerful than the simple global estimates; the use of speaker level statistics over global gives a gain from 18.2% to 16.8% compared to the clean baseline of 34.1%. The best normalisation scheme halves the overall error rate compared to the uncompensated system. However, the recognition accuracy is still not very good, and degrades rapidly as the SNR decreases further. A simple bias or single linear transform of the cepstral space cannot address the non-linear influence of noise in this domain.

SNR	CMN Global	CMN Speaker	CMN+CVN Global	CMN+CVN Speaker
32 dB	5.2	4.6	4.5	3.8
26 dB	9.2	8.8	7.5	6.7
20 dB	25.8	24.0	18.2	16.8
14 dB	62.4	62.8	50.0	47.1
8 dB	92.1	92.1	88.2	84.4

Table 6.2: Word error rates (%) of CMN and CVN systems with normalisation at a global and speaker level

6.2.3 Matched Systems

The clean uncompensated system and cepstral normalisation front-end baselines provide lower bounds for compensation schemes. A widely accepted upper bound is a matched system. Two forms were discussed in section 3.3: SPR and SPR with two-model re-estimation. Table 6.3 shows the results from using a matched system trained using SPR. Such a system represents a reasonable upper limit for non-iterative model-based compensation techniques such as PMC and model domain VTS. As expected, and seen in other research, testing and training on matched conditions yields good performance. At 20 dB SNR, the error rate is halved again from the speaker normalised features 16.8% to 7.1% and much better than the 34.1% with the uncompensated system. With a further 4 iterations of two-model re-estimation using clean speech state alignments, the recognition accuracy improves slightly from 11.8% to 11.4% averaged across the range of SNR evaluated. Still for these matched systems at lower SNR, the accuracy is unacceptably poor at less than 90%; for humans, an SNR of 8 dB hardly affects the intelligibility [1]. Thus from these baseline results, the 20 dB SNR environment seems the most balanced condition to investigate with a reasonable but not overwhelming difficulty. Thus most results reported focus on this level of noise.

SNR	Test Set			Average
	`Feb'89`	`Oct'89`	`Feb'91`	
32 dB	3.2	4.6	3.7	3.8
26 dB	4.4	6.4	5.3	5.4
20 dB	6.5	8.0	6.6	7.1
14 dB	12.8	14.2	12.8	13.3
8 dB	32.6	27.2	27.8	29.2

Table 6.3: Word error rates (%) of a SNR matched RM system as SNR decreases

6.3 Standard SPLICE Performance

The `SPLICE` algorithm requires the a GMM of the corrupted environment and MMSE estimates of the offsets. Two forms of the GMM were studied. The first was the standard form where the GMM is trained directly on noisy speech – this is referred to as the *noisy* front-end model. Alternatively, the front-end model can be estimated by first training on clean speech data, and then ideally compensating it using SPR. This is a more realistic scenario as data from the corrupted environment is not always readily available. This form will be called the *clean* front-end model.

Table 6.4 presents the `SPLICE` results. As expected, with an increase in the number components in the front-end GMM, the error rate is reduced. It is halved when 256 components are used.

[1]Based on the author's sampling of test data.

System	Front-End Model	Number of Components				
		1	4	16	64	256
Clean	—	34.1				
SPLICE	Noise	25.9	20.8	17.3	15.1	14.1
	Clean		20.4	16.8	14.6	12.9
Matched	—	7.1				

Table 6.4: Word error rates (%) comparing SPLICE with clean and noise front-end models, varying the number of components at 20 dB SNR

Unexpectedly, the clean front-end performs better than the noisy; it is unclear why this is the case. The noisy front-end should better model the corrupted acoustic space than the clean. The single component front-end is equivalent to CMN applied at a global level in that both are static biases applied to the feature vector. The hard approximation was found to be effective; a soft weighting improved results only slightly for low numbers of components and provided negligible gains at higher numbers of components. Overall, `SPLICE` provides good robustness, significantly better than speaker level CMN plus CVN results of 16.8% compared to 12.9%, but still far from the matched performance of 7.1%.

6.4 Uncertainty Decoding

The results presented in section 6.3 are for the baseline `SPLICE` system with no uncertainty decoding. This section gives results for systems using uncertainty decoding. Results are presented for both `SPLICE` with uncertainty (section 4.3.1) and the feature-based `Joint` scheme (section 4.3.2).

System	Front-End Model	Number of Components				
		1	4	16	64	256
Clean	—	34.1				
SPLICE	Noise	10.8	10.8	10.7	10.2	9.8
	Clean		13.1	12.8	11.4	11.2
Feature-Based Joint	Noise	10.6	11.4	11.8	11.2	11.8
	Clean		9.5	10.1	9.3	9.3
Matched	—	7.1				

Table 6.5: Word error rates (%) comparing uncertainty algorithm implementations at 20 dB SNR

Table 6.5 shows the performance of `SPLICE` with uncertainty on the 20dB SNR noise corrupted data. As in table 6.4, both noise and clean front-end forms were examined. It clear that for all conditions, when comparing these results with those without uncertainty that uncertainty decoding improves recognition accuracy. It is strange that with uncertainty, for `SPLICE` the noise front-end performs better than the clean, opposite to the results seen without uncertainty. It was found excessive insertion errors results with the addition of uncertainty with the clean front-end.

Tuning the inter-model insertion penalty improved the results, but still did not match the noise front-end performance. Surprisingly, the systems performed quite well with few components in the front-end. Only a small gain is had from one component at 10.8%, to 9.8% with 256 for the `SPLICE` system with the noise front-end. With a single component, a constant variance is propagated to the decoding process.

The `Joint` distribution uncertainty decoding algorithm performed well, generally better than the `SPLICE` form. In the best configurations this was 9.8 % compared to 9.3%. These are worse than the matched result of 7.1%. For the `Joint` system, the clean front-end was better than the noise. Again, the single component version performed surprisingly well. The single component forms of decoding with uncertainty surpassed all the cepstral normalisation and `SPLICE` systems without uncertainty. Also, the results show that the variance offset improves performance from 16.5%, applying a non-homogeneous transform to the feature vector as in global CMN plus CVN, to 10.6% with the single transform `Joint` form. Overall the addition of uncertainty to decoding proves beneficial.

Lastly, the behaviour of these forms can be examined for higher SNRs. In table 6.6, it can be seen that the `Joint` form is consistently slightly better than `SPLICE` , and uncertainty decoding in general significantly reduces the WER compared to an uncompensated system.

System	Front-End Model	SNR 32 dB	26 dB	20 dB	14 dB
Clean	—	5.4	10.8	34.1	77.2
`SPLICE`	Noise	4.1	6.0	9.8	18.6
Feature-Based `Joint`	Clean	4.0	5.5	9.3	19.3
Matched	—	3.8	5.4	7.1	13.3

Table 6.6: Word error rates (%) comparing uncertainty algorithms with 256 components

6.4.1 Qualitative Comparison of Forms

Since the forms of uncertainty decoding can be interpreted as providing a clean speech estimate and associated uncertainty, a plot can be produced to examine the signal processing. Figure 6.2 shows the C_0 plot over time for the clean, noise corrupted, estimated clean speech and another plot of the estimate with uncertainty for the sentence “Clear all windows”. As expected, the areas where the clean speech signal is strong, such as voiced regions, have a good estimates of speech, with low variance. In non-speech areas or those with less speech energy, such as the plosive n in *windows* around the 80th frame, the clean speech estimate is poorer with greater uncertainty. This is consistent with the expected operation of the algorithm.

The operation of the `Joint` algorithm can also be examined in the same way as shown in figure 6.3 with the same sentence. For high energy regions of speech the same behaviour is observed

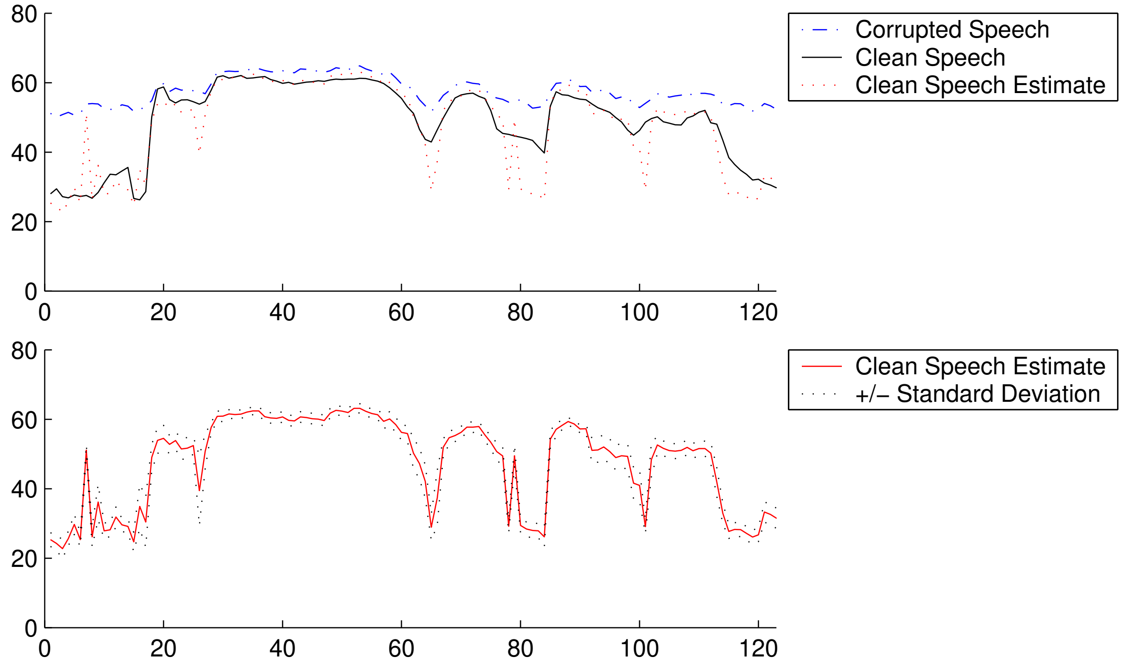

Figure 6.2: Comparing C_0 for clean speech, with Operating Room noise at 20 dB SNR and the SPLICE speech estimate above using a 256 component noisy front-end. The distribution of $\hat{\boldsymbol{x}}$ is plotted below.

as for the `SPLICE` scheme. However, for low speech energy regions, the estimates are more erratic with very large uncertainties. Since the variances are so large, frames with such estimates have little discriminative power and hence do not influence the decoding. An interesting finding was that for both the `SPLICE` and `Joint` uncertainty decoding techniques, a single component front-end performed very well. Figure 6.4 illustrates a plot similar to figure 6.3 for the `Joint` scheme with a single component clean front-end. The clean speech estimate is poorer than for the 256 component system. However the variance is better behaved; it has greater uncertainty of high speech regions compared to the 256 system, but lacks the erratic variance spikes in the low energy areas.

6.5 Model Adaptation

Two forms of model adaptation were examined, CMLLR and a `Joint` transform, with regression classes, operating as described in section 4.4. CMLLR provides a useful baseline as a linear transform based compensation methods. With only a global regression class, it represents a linear transform of the feature vector trained on the noise condition. With multiple regression classes, it functions conceptually as parallel front-ends, each with a different transform for different classes of models. The `Joint` transforms derived are similar to CMLLR with diagonal matrices, but has

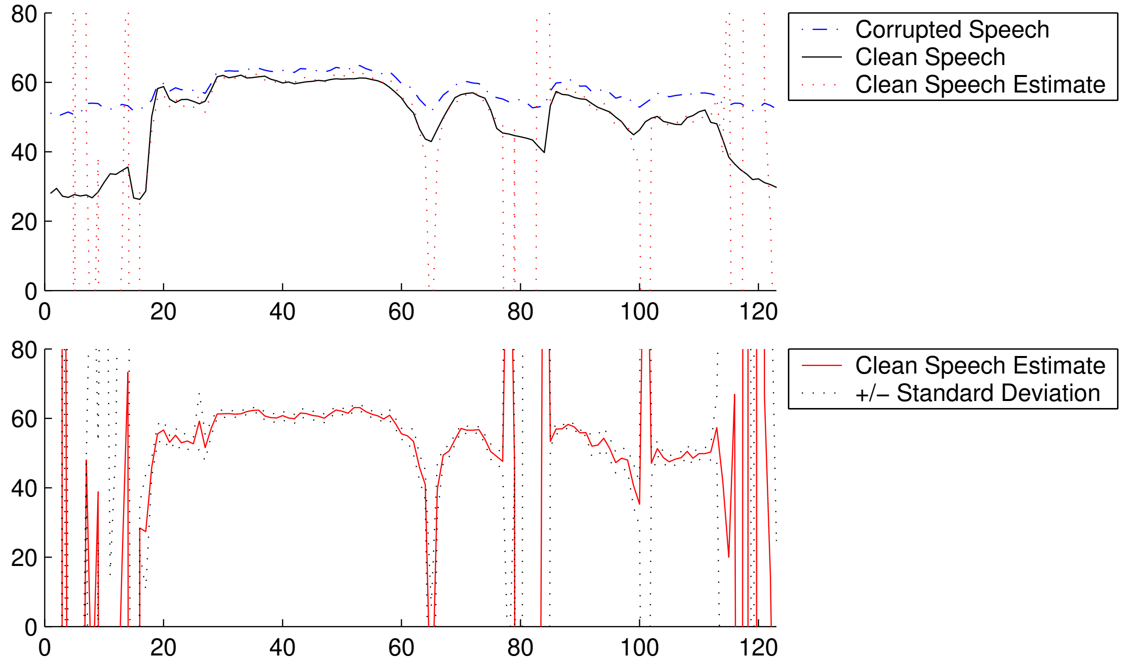

Figure 6.3: Comparing C_0 for clean speech, with Operating Room noise at 20 dB SNR and the Joint speech estimate above using a 256 component clean front-end. The distribution of $\hat{\boldsymbol{x}}$ is plotted below.

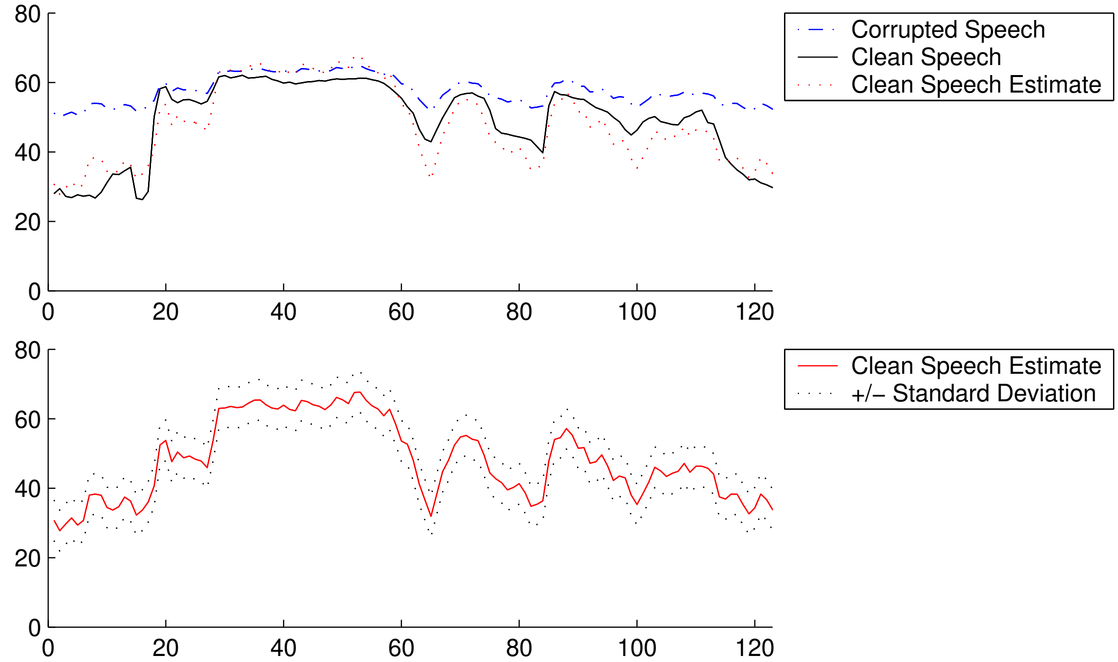

Figure 6.4: Comparing C_0 for clean speech, with Operating Room noise at 20 dB SNR and the `Joint` speech estimate above using a single component front-end. The distribution of $\hat{\boldsymbol{x}}$ is plotted below.

an added variance offset to the models. Models are classified according to a Euclidean measure of closeness of their means. The effect of the number of classes on accuracy is explored. Model `Joint` transforms also provide an interesting contrast to the feature-based `Joint` scheme as it does not have this front-end component selection problem. Table 6.7 presents the model-based `Joint` scheme compared with the performance of CMLLR and feature-based `Joint` .

	Number of Classes						
Transform	1	2	4	8	16	32	64
CMLLR – Diagonal	16.5	15.5	14.9	11.6	10.4	9.6	8.9
CMLLR – Block-Diagonal	15.8	19.4	18.3	10.0	9.2	8.8	8.7
CMLLR – Full	17.0	17.0	15.0	10.0	9.2	9.0	8.4
Model-Based `Joint`	10.6	10.0	9.9	9.1	8.9	9.2	9.2

	Number of Components				
System	1	4	16	64	256
Feature-Based `Joint`	10.6	9.5	10.1	9.3	9.3

Table 6.7: Word error rates (%) comparing different model transforms and Feature-Based `Joint`, varying the number of classes/components at 20 dB SNR

It was found that performance leveled off after 64 classes and the addition of a silence class was only slightly better for low numbers of regression classes. The CMLLR transforms were estimated using two iterations of BW. The 2 and 4 class block-diagonal transforms were found to have not converged thus giving the aberrant rates of 19.4% and 18.3%. It was found that full transforms took much longer to train than block-diagonal transforms with little to no benefits, while the block-diagonal transform did perform better than the diagonal transform as expected, indicating that correlations within blocks are important, whereas correlations between the static, delta and delta-delta coefficients are less so. Generally there was a sharp improvement in accuracy at eight regression classes; thereafter, the accuracy only improves slightly. At eight classes, CMLLR with block-diagonal transforms performed markedly better than the best `SPLICE` system, halving the word error rate from 12.9% to 10.0 relative to the matched system at 7.1%. With 64 regression classes, the accuracy nears the matched at 8.1%, however at a high computational cost. The diagonal transform performs well at 11.6% better than `SPLICE` but worse than the block transform with eight regression classes.

It is interesting to compare the global transform cases of CMLLR with `SPLICE` using only a single component in the front-end GMM. As one would expect, with both the block and diagonal transforms, the use of both a transform and a bias is far more effective than just a bias, as `SPLICE` amounts to with one component – 15.8% and 16.5% compared to 25.9%. However, CMLLR with a single class and a diagonal transform, is similar to CMN and CVN at a global level, giving comparable performance(16.8%). Since both transform and offset the feature vector in the same manner this is expected.

The model-based `Joint` transform is equivalent to the feature-space version with one front-end GMM component, sharing the same unexpectedly robust performance of 10.6%. From there, there is a small incremental improvement, but after eight classes the performance levels. For the fewer numbers of classes, the model-based `Joint` transform outperforms CMLLR, however with large numbers of transforms CMLLR does slightly better than `Joint` . The relative improvement moving from 1 to 8 or more regression classes is only about 15% or a gain of about one and half percent absolute WER% whereas for the CMLLR transforms there is a substantial relative gains of over 40%.

The diagonal CMLLR and the model-based `Joint` transforms provide a good comparison since they both apply similar non-homogeneous transforms to the feature vector, but the `Joint` scheme adds a variance offset. Also the `Joint` transform, as implemented, only uses diagonal covariances. Since the diagonal CMLLR converges to the `Joint` performance with higher numbers of regression classes, this suggests that either for more model specific transforms, finer estimation of the means is more important than the estimation of the uncertainty offsets. Further to this point, the block transform exceeds the performance of the uncertainty decoding when there are more than 32 regression classes. It would be interesting to explore the performance of the `Joint` transform with a block form to see if the improved estimates translate to better a WER. The variances could be constrained to diagonal or even a single global matrix to determine the effect coarser measures of uncertainty have on recognition.

The model-based `Joint` transform and the feature-based version both plateau in performance at just over 9% WER. It is unclear whether this is due to the forms which estimate the clean speech or uncertainty, or if it is an inherent limitation in uncertainty decoding. It would be useful to conduct experiments using the true uncertainty and this is possible using stereo data. The magnitude-square error between the estimated cepstral vectors from the front-end and the true clean speech can be used as Oracle uncertainty values[15].

6.6 Computational Load

By increasing the model variances with uncertainty decoding, given a set beam width, the number of active models will naturally rise perhaps reducing the number of search errors, but definitely increase the computational load. Hence, it is worth investigating the sensitivity of the results to the number of active models. Table 6.8 shows the average number of active models during decoding at different pruning thresholds, for a variety of schemes, and the associated WER for reference on the Feb'89 test set. The `SPLICE` baseline is for a 256 component system and shows how the refined clean speech estimate allows the recogniser to more efficiently perform model pruning during the search. The single component `Joint` configuration was of interest because of its unexpectedly robust performance. At the standard pruning threshold of 300, the `Joint` uncertainty decode causes a large increase in the number of models active in the recogniser. But it can be seen that despite a two fold reduction in the pruning threshold and a significant drop in the number of models evaluated to below the matched condition, the WER is only slightly affected. Thus the `Joint` distribution uncertainty decoding is inherently sound, and gains are found even when the number of active models are reduced to below standard levels.

System	With Uncertainty?	Pruning Threshold	WER	# Active Models
Clean	—	300	37.6	10153
		150	33.9*	1632
`SPLICE` (Baseline)	No	300	14.4	4306
Feature-Based `Joint` 1 Comp.	Yes	300	11.3	19680
		150	11.4	4096
		100	11.7	1144
Feature-Based `Joint` 256 Comp.	Yes	300	8.9	16600
		150	9.1	3445
		100	9.6	1037
Matched	—	300	6.5	5535
		150	7.1	865

Table 6.8: Word error rates (%) and active models at 20 dB SNR as pruning decreases on the `Feb'89` test set only. *Not all sentences yielded a hypothesis.

6.7 Summary of Results

Figure 6.5 shows the performance of some of the compensation configurations over a range of SNRs. As expected the compensation schemes examined are bounded in performance by the clean system performance as a lower bound and the matched, SPR, system as an upper bound. On this task with the range of SNRs considered the two uncertainty decoding schemes performed significantly better than `SPLICE` without uncertainty.

The preliminary results provide uncertainty decoding figures for two different forms of the conditional corrupted speech distribution; one modeled directly using the joint distribution, the other based on the Bayes equivalent, with the clean posterior using the SPLICE form. Both yielded positive results compared to the baseline and cepstral mean and variance normalisation, across a range of SNRs. The benefit of uncertainty decoding over standard decoding was clear. There was a definite gain in the `SPLICE` form with uncertainty than without. With lower numbers of model transforms, the addition of uncertainty gave the `Joint` scheme a clear gain over the CMLLR transforms.

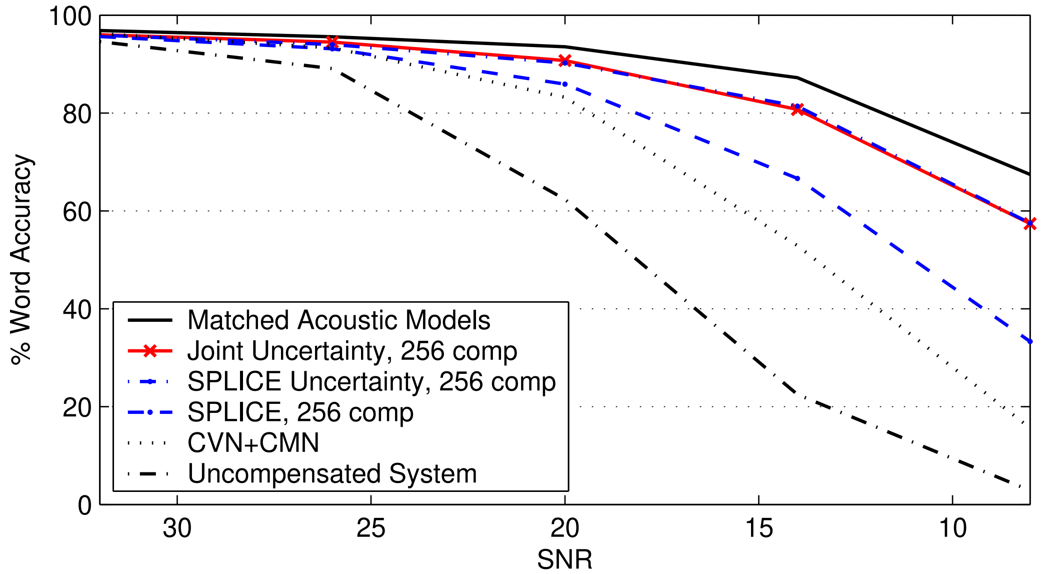

Figure 6.5: Comparing overall performance of different noise robustness techniques

Chapter 7

Conclusions

The overall aim of this work was to develop a form of fast noise compensation through the propagation of uncertainty to the recogniser without the associated cost of model-based approaches. To begin, a formal framework for noise robust automatic speech recognition was presented. Past and present robustness techniques are reviewed in this framework. This inclues recent work in uncertainty observations and decoding and its relation to this preliminary work. A framework for uncertainty decoding was then presented in this context. Uncertainty decoding is based on passing the uncertainty of the decoding through an approximation of the conditional corrupted speech distribution to the decoder. Tractable and efficient forms are the research focus, and two forms of uncertainty decoding presented. One is based on the past work with `SPLICE`, the other is a novel `Joint` distribution uncertainty decoding. This framework characterises the noise environment in a front-end model, decoupling the front-end processing from the acoustic model complexity, but still propagating the uncertainty of the current frame. The front-end model of these forms are estimated from stereo data. Methods to dynamically estimate the front-end models in an unsupervised fashion through iterative environment and predictive front-end model estimation are discussed.

The Cambridge University HMM Toolkit was updated to support uncertainty decoding and the estimation of the front-end models from stereo data. The `SPLICE` and `Joint` forms were implemented and tested on the medium vocabulary Resource Management task across a range of SNRs using artificially added NOISEX Operating Room noise. The addition of uncertainty to the decoding gave definite gains in performance, although did not meet the matched condition. Surprisingly, a single component front-end yielded extremely good performance, with other more sophisticated configurations only marginally improving on it. This amounts to some feature processing and a static variance offset. Several approximations were made in the derivation of these forms, and are felt to contribute to some loss in performance. It would be useful to run Oracle experiments with more ideal form of uncertainty decoding, relaxing some constraints, and using stereo data to determine theoretical upper bounds in this form of decoding. The run-time computational load was also examined. The variance expansion resulted in an increased number of active models being evaluated, but a decrease in the pruning threshold increased the speed, without adversely affecting performance.

7.1 Future Work

The work so far has provided a sound framework for further research. First, the AURORA corpus should be obtained to allow better comparisons with contemporary systems. Next, experimentation to explore the approximations made and investigate the upper bounds with uncertainty decoding, and modified `SPLICE` and `Joint` forms should be conducted. The `Joint` form can be freed from using stereo data, by actively estimating the noise and dynamically compensating the front-end models. Lastly, a noise adaptive framework can be developed where uncertainty transforms are estimated and applied during training and testing against a canonical acoustic model set similar to speaker adaptive training. The final system should automatically recognise speech from unknown environments in a robust and efficient manner through uncertainty decoding.

Appendix A

SPLICE Conditional Corrupted Speech Derivation

In section 4.3.1 the `SPLICE` form of uncertainty decoding is derived. With the simplified clean speech prior the conditional corrupted speech distribution takes the form

$$p\big(\boldsymbol{y}_t|\boldsymbol{x}_t,\check{\mathcal{M}}\big) = \frac{\sum_{n=1}^{N} p\big(\boldsymbol{x}_t|\boldsymbol{y}_t,\check{s}_n,\check{\mathcal{M}}\big)\, p\big(\boldsymbol{y}_t|\check{s}_n,\check{\mathcal{M}}\big)\,\check{c}_n}{p\big(\boldsymbol{x}_t|\check{\mathcal{M}}\big)} \tag{A.1}$$

$$\approx \frac{\sum_{n=1}^{N} \mathcal{N}\Big(\boldsymbol{x}_t;\boldsymbol{y}_t+\check{\boldsymbol{\mu}}^{(n)},\check{\boldsymbol{\Sigma}}^{(n)}\Big)\, p\big(\boldsymbol{y}_t|\check{s}_n,\check{\mathcal{M}}\big)\,\check{c}_n}{\mathcal{N}(\boldsymbol{x}_t;\boldsymbol{\mu}_x,\boldsymbol{\Sigma}_x)} \tag{A.2}$$

$$= \sum_{n=1}^{N} p\big(\boldsymbol{y}_t|\check{s}_n,\check{\mathcal{M}}\big)\,\check{c}_n\alpha^{(n)}\mathcal{N}\Big(\boldsymbol{A}^{(n)}\boldsymbol{y}_t+\boldsymbol{b}^{(n)};\boldsymbol{x}_t,\hat{\boldsymbol{\Sigma}}_x^{(n)}\Big) \tag{A.3}$$

The division of each component normal clean speech posterior distribution by the simplified single Gaussian clean speech prior can be further detailed. If the covariance matrices are assumed to be diagonal, the conditional can be derived per dimension i

$$\frac{p\big(x_{ti}|y_{ti},\check{s}_n,\check{\mathcal{M}}\big)}{p\big(x_{ti}|\check{\mathcal{M}}\big)} \approx \frac{\mathcal{N}\Big(x_{ti};y_{ti}+\check{\mu}_i^{(n)},\check{\sigma}_i^{(n)2}\Big)}{\mathcal{N}(x_{ti};\mu_{xi},\sigma_{xi}^2)} \tag{A.4}$$

$$= \frac{\frac{1}{\sqrt{2\pi}\check{\sigma}_i^{(n)}}\exp\left[-\frac{1}{2}\frac{(x_{ti}-(y_{ti}+\check{\mu}_i^{(n)}))^2}{\check{\sigma}_i^{(n)2}}\right]}{\frac{1}{\sqrt{2\pi}\sigma_{xi}}\exp\left[-\frac{1}{2}\frac{(x_{ti}-\mu_{xi})^2}{\check{\sigma}_{xi}^2}\right]} \tag{A.5}$$

$$= \frac{\sigma_{xi}}{\check{\sigma}_i^{(n)}}\exp\left[-\frac{1}{2}\left\{\left(\frac{(x_{ti}-(y_{ti}+\check{\mu}_i^{(n)}))^2}{\check{\sigma}_i^{(n)2}}\right)-\left(\frac{(x_{ti}-\mu_{xi})^2}{\sigma_{xi}^2}\right)\right\}\right] \tag{A.6}$$

$$= \frac{\sigma_{xi}}{\check{\sigma}_i^{(n)}}\exp\left[-\frac{1}{2}\left\{\frac{(x_{ti}-(y_{ti}+\check{\mu}_i^{(n)}))^2\sigma_{xi}^2-(x_{ti}-\mu_{xi})^2\check{\sigma}_i^{(n)2}}{\sigma_{xi}^2\check{\sigma}_i^{(n)2}}\right\}\right] \tag{A.7}$$

The difference of squares can be expanded and the terms collected as follows

$$\left\{\frac{(x_{ti}-(y_{ti}+\check{\mu}_i^{(n)}))^2\sigma_{xi}^2-(x_{ti}-\mu_{xi})^2\check{\sigma}_i^{(n)2}}{\sigma_{xi}^2\check{\sigma}_i^{(n)2}}\right\}$$

$$=\frac{1}{\sigma_{xi}^2\check{\sigma}_i^{(n)2}}\left\{(\sigma_{xi}^2-\check{\sigma}_i^{(n)2})x_{ti}^2-2\left\{(y_{ti}+\check{\mu}_i^{(n)})\sigma_{xi}^2-\mu_{xi}\check{\sigma}_i^{(n)2}\right\}x_{ti}+(y_{ti}+\check{\mu}_i^{(n)})^2\sigma_{xi}^2+\mu_{xi}^2\check{\sigma}_i^{(n)2}\right\} \tag{A.8}$$

$$=\frac{\sigma_{xi}^2-\check{\sigma}_i^{(n)2}}{\sigma_{xi}^2\check{\sigma}_i^{(n)2}}\left\{x_{ti}^2-2\left\{\frac{(y_{ti}+\check{\mu}_i^{(n)})\sigma_{xi}^2-\mu_{xi}\check{\sigma}_i^{(n)2}}{\sigma_{xi}^2-\check{\sigma}_i^{(n)2}}\right\}x_{ti}+\frac{(y_{ti}+\check{\mu}_i^{(n)})^2\sigma_{xi}^2+\mu_{xi}^2\check{\sigma}_i^{(n)2}}{\sigma_{xi}^2-\check{\sigma}_i^{(n)2}}\right\} \tag{A.9}$$

Now it becomes clear that the mean and variance of the normal distribution that is sought are

$$\hat{\mu}_{xi}^{(n)}=\frac{(y_{ti}+\check{\mu}_i^{(n)})\sigma_{xi}^2-\mu_{xi}\check{\sigma}_i^{(n)2}}{\sigma_{xi}^2-\check{\sigma}_i^{(n)2}} \tag{A.10}$$

$$\hat{\sigma}_{xi}^{(n)2}=\frac{\sigma_{xi}^2\check{\sigma}_i^{(n)2}}{\sigma_{xi}^2-\check{\sigma}_i^{(n)2}} \tag{A.11}$$

Thus equation A.7 can now be written as

$$\frac{p\big(x_{ti}|y_{ti},\check{s}_n,\check{\mathcal{M}}\big)}{p\big(x_{ti}|\check{\mathcal{M}}\big)}\approx\frac{\sigma_{xi}}{\check{\sigma}_i^{(n)}}\exp\left[-\frac{1}{2}\left\{\frac{(x_{ti}-(y_{ti}+\check{\mu}_i^{(n)}))^2\sigma_{xi}^2-(x_{ti}-\mu_{xi})^2\check{\sigma}_i^{(n)2}}{\sigma_{xi}^2\check{\sigma}_i^{(n)2}}\right\}\right] \tag{A.12}$$

$$=\frac{\sigma_{xi}}{\check{\sigma}_i^{(n)}}\exp\left[-\frac{1}{2}\frac{\left\{x_{ti}^2-2\hat{\mu}_{xi}^{(n)}x_{ti}+S^{(n)}\right\}}{\hat{\sigma}_{xi}^{(n)2}}\right] \tag{A.13}$$

where

$$S^{(n)}=\frac{(y_{ti}+\check{\mu}_i^{(n)})^2\sigma_{xi}^2-\mu_{xi}^2\check{\sigma}_i^{(n)2}}{\sigma_{xi}^2-\check{\sigma}_i^{(n)2}} \tag{A.14}$$

This can now be written in the form of a normal distribution

$$\frac{\sigma_{xi}}{\check{\sigma}_i^{(n)}}\exp\left[-\frac{1}{2}\frac{\left\{x_{ti}^2-2\hat{\mu}_{xi}^{(n)}x_{ti}+S^{(n)}\right\}}{\hat{\sigma}_{xi}^{(n)2}}\right] \tag{A.15}$$

$$=\frac{\sigma_{xi}}{\check{\sigma}_i^{(n)}}\exp\left[-\frac{1}{2}\left(\frac{S^{(n)}-\hat{\mu}_{xi}^{(n)2}}{\hat{\sigma}_{xi}^{(n)2}}\right)\right]\exp\left[-\frac{1}{2}\frac{(x_{ti}-\hat{\mu}_{xi}^{(n)})^2}{\hat{\sigma}_{xi}^{(n)2}}\right] \tag{A.16}$$

$$=\alpha^{(n)}\mathcal{N}\Big(x_{ti};\hat{\mu}_{xi}^{(n)},\hat{\sigma}_{xi}^{(n)2}\Big) \tag{A.17}$$

where

$$\alpha^{(n)}=\frac{\sigma_{xi}}{\check{\sigma}_i^{(n)}}\exp\left[-\frac{1}{2}\left(\frac{S^{(n)}-\hat{\mu}_{xi}^{(n)2}}{\hat{\sigma}_{xi}^{(n)2}}\right)\right]\sqrt{2\pi}\hat{\sigma}_{xi}^{(n)2} \tag{A.18}$$

Thus with the assumption of diagonal covariance matrices it can be concluded that

$$\frac{p\big(\boldsymbol{x}_t|\boldsymbol{y}_t,\check{s}_n,\check{\mathcal{M}}\big)}{p\big(\boldsymbol{x}_t|\check{\mathcal{M}}\big)}\approx\frac{\mathcal{N}\Big(\boldsymbol{x}_t;\boldsymbol{y}_t+\check{\boldsymbol{\mu}}^{(n)},\check{\boldsymbol{\Sigma}}^{(n)}\Big)}{\mathcal{N}(\boldsymbol{x}_t;\boldsymbol{\mu}_x,\boldsymbol{\Sigma}_x)} \tag{A.19}$$

$$=\alpha^{(n)}\mathcal{N}\Big(\boldsymbol{x}_t;\hat{\boldsymbol{\mu}}_x^{(n)},\hat{\boldsymbol{\Sigma}}_x^{(n)}\Big) \tag{A.20}$$

where the elements of the mean vector are given in equation A.10 and the diagonal elements in the covariance matrix in equation A.11. With further manipulation, it can be shown that

$$\mathcal{N}\Big(\boldsymbol{x}_t; \hat{\boldsymbol{\mu}}_x^{(n)}, \hat{\boldsymbol{\Sigma}}_x^{(n)}\Big) = \mathcal{N}\Big(\boldsymbol{A}^{(n)}\boldsymbol{y}_t + \boldsymbol{b}^{(n)}; \boldsymbol{x}_t, \hat{\boldsymbol{\Sigma}}_x^{(n)}\Big) \tag{A.21}$$

with

$$\begin{aligned} a_{ii}^{(n)} &= \frac{\sigma_{xi}^2}{\sigma_{xi}^2 - \check{\sigma}_i^{(n)2}} \\ b_i^{(n)} &= a_{ii}^{(n)}\left(\check{\mu}_i^{(n)} - \frac{\check{\sigma}_i^{(n)2}}{\sigma_{xi}^{(n)2}}\mu_{xi}\right) \\ \hat{\boldsymbol{\Sigma}}_x^{(n)} &= \boldsymbol{A}^{(n)}\check{\boldsymbol{\Sigma}}^{(n)} \end{aligned} \tag{A.22}$$

Appendix B

Convolution of Two Gaussian Distributions

In the marginalisation across the clean speech variable $\boldsymbol{x}_t$ this integral appears

$$p\big(\boldsymbol{y}_t|\mathcal{M},\check{\mathcal{M}},\check{s}_n,s_m,\theta_t\big)=\alpha^{(n)}\int_{\mathcal{R}^d}\mathcal{N}\Big(\boldsymbol{A}^{(n)}\boldsymbol{y}_t+\boldsymbol{b}^{(n)};\boldsymbol{x}_t,\hat{\boldsymbol{\Sigma}}_x^{(n)}\Big)\mathcal{N}\Big(\boldsymbol{x}_t;\boldsymbol{\mu}^{(m)},\boldsymbol{\Sigma}^{(m)}\Big)\,d\boldsymbol{x}_t \tag{B.1}$$

This integration can also be considered the convolution of the two normal distributions or the sum of two independent random vectors $\boldsymbol{w}\sim\mathcal{N}\Big(0,\hat{\boldsymbol{\Sigma}}_x^{(n)}\Big)$ and $\boldsymbol{x}\sim\mathcal{N}\Big(\boldsymbol{\mu}^{(m)},\boldsymbol{\Sigma}^{(m)}\Big)$. Since the mean and variance of the sum of two independent normally distributed random vectors is another normally distributed random vector whose mean is the sum of the means and covariance matrix is the sum of the covariances, the resulting distribution is

$$\boldsymbol{y}=\boldsymbol{w}+\boldsymbol{x}\sim\mathcal{N}\Big(\boldsymbol{\mu}^{(m)},\boldsymbol{\Sigma}^{(m)}+\hat{\boldsymbol{\Sigma}}_x^{(n)}\Big) \tag{B.2}$$

Thus

$$p\big(\boldsymbol{y}_t|\mathcal{M},\check{\mathcal{M}},\check{s}_n,s_m,\theta_t\big)=\alpha^{(n)}\mathcal{N}\Big(\boldsymbol{A}^{(n)}\boldsymbol{y}_t+\boldsymbol{b}^{(n)};\boldsymbol{\mu}^{(m)},\boldsymbol{\Sigma}^{(m)}+\hat{\boldsymbol{\Sigma}}_x^{(n)}\Big) \tag{B.3}$$

A similar result can be found in [16].

Appendix C

The Conditional Multivariate Gaussian

Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be multivariate Gaussian pdfs of dimensions p and q, mean vectors of $\boldsymbol{\mu}_x$ and $\boldsymbol{\mu}_x$, and covariance matrices of $\boldsymbol{\Sigma}_x$ and $\boldsymbol{\Sigma}_y$ respectively. The joint distribution of these two random vectors can be considered Gaussian distributed

$$p(\boldsymbol{x}, \boldsymbol{y}) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{C.1}$$

where the mean and variance are given by

$$\begin{aligned} \boldsymbol{\mu} &= \begin{bmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y \end{bmatrix} \\ \boldsymbol{\Sigma} &= \begin{bmatrix} \boldsymbol{\Sigma}_x & \boldsymbol{\Sigma}_{xy} \\ \boldsymbol{\Sigma}_{yx} & \boldsymbol{\Sigma}_y \end{bmatrix} \end{aligned} \tag{C.2}$$

and $\boldsymbol{\Sigma}_{xy}$ and $\boldsymbol{\Sigma}_{yx}$ are the cross covariances between $\boldsymbol{x}$ and $\boldsymbol{y}$, and $\boldsymbol{\Sigma}_{xy} = \boldsymbol{\Sigma}_{yx}^{\mathsf{T}}$.

Bayes' rule dictates that

$$p(\boldsymbol{x}|\boldsymbol{y}) = \frac{p(\boldsymbol{x}, \boldsymbol{y})}{p(\boldsymbol{y})} \tag{C.3}$$

Since both $p(\boldsymbol{x}, \boldsymbol{y})$ and $p(\boldsymbol{y})$ are Gaussian distributed, the conditional pdf of $\boldsymbol{x}$ given $\boldsymbol{y}$ is also Gaussian distributed

$$p(\boldsymbol{x}|\boldsymbol{y}) \sim \mathcal{N}\Big(\boldsymbol{x}; \boldsymbol{\mu}_{x|y}, \boldsymbol{\Sigma}_{x|y}\Big) \tag{C.4}$$

where

$$\boldsymbol{\mu}_{x|y} = \boldsymbol{\mu}_x + \boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_y^{-1}\left(\boldsymbol{y} - \boldsymbol{\mu}_y\right) \tag{C.5}$$

$$\boldsymbol{\Sigma}_{x|y} = \boldsymbol{\Sigma}_x - \boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_y^{-1}\boldsymbol{\Sigma}_{yx} \tag{C.6}$$

A similar form can also be derived for $p(\boldsymbol{y}|\boldsymbol{x})$. $\boldsymbol{\Sigma}_{x|y}$ is also referred to as the Schur decomposition of $\boldsymbol{\Sigma}$ with respect to $\boldsymbol{\Sigma}_y$ and may be written as $\boldsymbol{\Sigma}_{|\Sigma_y}$.

Bibliography

[1] A. Acero. *Acoustical and Environmental Robustness in Automatic Speech Recognition*. PhD thesis, Carnegie Mellon University, 1990.

[2] A. Acero, L. Deng, T. Kristjansson, and J. Zhang. HMM Adaptation Using Vector Taylor Series for Noisy Speech Recognition. In *Proceedings of the International Conference on Spoken Language Processing*, Beijing, China, October 2000.

[3] A. Acero and R. M. Stern. Environmental robustness in automatic speech recognition. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Albuquerque, New Mexico, 1990.

[4] A. Acero and R. M. Stern. Robust speech recognition by normalization of the acoustic space. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Toronto, Ontario, 1991.

[5] S. Ahmad and V. Tresp. Some solutions to the missing feature problem in vision. In *Advances in Neural Information Processing Systems 5*, pages 393–400, San Mateo, California, 1993.

[6] J.A. Arrowood. *Using Observation Uncertainty for Robust Speech Recognition*. PhD thesis, Georgia Institute of Technology, 2003.

[7] J.A. Arrowood and M.A. Clements. Using Observation Uncertainty In HMM Decoding. In *Proceedings of ICSLP*, Denver, Colorado, September 2002.

[8] S.F. Boll. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions ASSP*, 27:113–120, 1979.

[9] S.E. Bou-Ghazale and J.H.L. Hansen. Duration and spectral based stress token generation for HMM speech recognition under stress. In *Proc. IEEE ICASSP*, Adelaide, Australia, April 1994.

[10] M. Cooke, P. Green, L. Josifovski, and A. Vizinho. Robust automatic speech recognition with missing and unreliable acoustic data. *Speech Communication*, 34(3):267–285, June 2001.

[11] S.B. Davis and P. Mermelstein. Comparison of parametric representations for monosyllable word recognition in continuously spoken sentences. *IEEE Transactions on Speech and Audio Processing*, 28(4):357–366, 1980.

[12] L. Deng, A. Acero, M. Plumpe, and X.D. Huang. Large vocabulary speech recognition under adverse acoustic environments. In *Proceedings of the International Conference on Speech and Language Processing*, pages 806–809, Beijing, China, October 2000.

[13] L. Deng, J. Droppo, and A. Acero. Recursive estimation of nonstationary noise using iterative stochastic approximation for robust speech recognition. *IEEE Trans. on Speech and Audio Processing*, 11(6), November 2003.

[14] J. Droppo and A. Acero. Noise robust speech recognition with a switching linear dynamic model. In *Proc. ICASSP*, Montreal, Canada, May 2004.

[15] J. Droppo, A. Acero, and L. Deng. Uncertainty decoding with splice for noise robust speech recognition. In *Proc. ICASSP*, Orlando, Florida, May 2002.

[16] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley-Interscience, 2nd edition, 2001.

[17] Y. Ephraim. A Bayesian estimation approach for speech enhancement using hidden Markov models. *IEEE Transactions ASSP*, 40:725–735, 1992.

[18] Y. Ephraim, D. Malah, and B.-H. Juang. On the application of hidden Markov models for enhancing noisy speech. *IEEE Transactions ASSP*, 37:1846–1856, December 1989.

[19] B. Frey, T.T. Kristjansson, L. Deng, and A. Acero. ALGONQUIN – learning dynamic noise models from noisy speech for robust speech recognition. In *Proc NIPS*, 2001.

[20] M.J.F. Gales. *Model-Based Techniques for Noise Robust Speech Recognition*. PhD thesis, Cambridge University, 1995.

[21] M.J.F. Gales. The generation and the use of regression class trees for MLLR adaptation. Technical Report CUED/F-INFENG/TR263, University of Cambridge, 1996. Available via anonymous ftp from: svr-www.eng.cam.ac.uk.

[22] M.J.F. Gales. Maximum Likelihood Linear Transformations For HMM-Based Speech Recognition. *Computer Speech and Language*, 12, January 1998.

[23] M.J.F. Gales. Predicative model based compensation schemes for robust speech recognition. *Speech Communication*, 25, 1998.

[24] M.J.F. Gales and S.J.Young. Robust continuous speech recognition using parallel model combination. *IEEE Trans. on Speech and Audio Processing*, 1996.

[25] M.J.F. Gales and P.C. Woodland. Mean and variance adaptation within the MLLR framework. *Computer Speech and Languages*, 10:249–264, 1996.

[26] M.J.F. Gales and S. J. Young. Robust speech recognition in additive and convolutional noise using parallel model combination. *Computer Speech and Language*, 9:289–307, 1995.

[27] M.J.F. Gales and S.J. Young. An improved approach to hidden Markov model decomposition of sleep and noise. In *Proc. ICASSP*, 1992.

[28] Y. Gong. Speech recognition in noisy environments. a survey. *Speech Communication*, 16:261–291, 1995.

[29] J.H.L. Hansen. Analysis and compensation of speech under stress and noise for environmental robustness in speech recognition. *Speech Communication*, 20(2):151–170, November 1996.

[30] H. Hermansky. Perceptual Linear Predictive (PLP) analysis of speech. *Journal of the Acoustic Society of America*, 87(4):1738–1752, 1990.

[31] H. Hermansky. Should Recognizers Have Ears? In *Proceedings of ESCA Tutorial and Research Workshop on Robust Speech Recognition for Unknown Communication Channels*, pages 1–10, France, 1997.

[32] H. Hermansky. Mel cepstrum, deltas, double-deltas,... - what else is new? In *In Robust Methods for Speech Recognition in Adverse Conditions*, Tampere, Finland, 1999.

[33] H. Hermansky and N. Morgan. RASTA processing of speech. *IEEE Transactions on Speech and Audio Processing*, 2(4), October 1994.

[34] X.D. Huang, A. Acero, and H.W. Hon. *Spoken Language Processing*. Prentice Hall, 2001.

[35] D.Y. Kim, N.S. Kim, and C.K. Un. Model-based approach for robust speech recognition in noisy environments with multiple noise sources. In *Proc. Eurospeech*, 1997.

[36] D.Y. Kim, C.K. Un, and N.S. Kim. Speech recognition in noisy environments using first-order vector Taylor series. *Speech Communication*, 24(1):39–49, June 1998.

[37] J. Koehler, N. Morgan, H. Hermansky, H. Gunter-Hirsh, and G. Tong. Integrating RASTA-PLP into speech recognition. In *Proc. ICASSP*, volume 1, pages 421–424, Albuquerque, New Mexico, 1994.

[38] T.T. Kristjansson. *Speech Recognition in Adverse Environments: a Probabilistic Approach*. PhD thesis, University of Waterloo, Waterloo, Canada, 2002.

[39] T.T. Kristjansson and B.J. Frey. Accounting for uncertainty in observations: A new paradigm for robust speech recognition. In *Proc. ICASSP*, Orlando, Florida, May 2002.

[40] K. Lee, B. Lee, I. Song, and J. Yoo. Recursive speech enhancement using the EM algorithm with initial conditions trained by HMM's. In *Proc. ICASSP*, 1996.

[41] C. Leggetter and P.C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density HMMs. *Computer Speech and Language*, 9:171–186, 1995.

[42] X. Li and R.M. Stern. Feature Generation Based on Maximum Classification Probability for Improved Speech Recognition. In *Proc. Eurospeech*, Geneva, Switzerland, September 2003.

[43] M. Lieb and A. Fischer. Experiments with the Philips continuous ASR system on the AURORA noisy digits database. In *Proc. Eurospeech*, Aalborg, Denmark, September 2001.

[44] B. Logan and A. Robinson. Enhancement and recognition of noisy speech within an autoregressive hidden Markov model framework using estimates from the noisy signal. In *Proc. ICASSP*, 1997.

[45] J. Lunqua. The Lombard reflex and its role on human listeners and automatic speech recognizers. In *Proc. JASA*, pages 510–524, January 1993.

[46] H. Mehanna. Estimating noise models using noise corrupted data. Master's thesis, University of Cambridge, Cambridge, UK, July 2004.

[47] Sirko Molau, Florian Hilger, and Hermann Ney. Feature space normalization in adverse acoustic conditions. In *Proc. ICASSP*, 2003.

[48] P. Moreno. *Speech Recognition in Noisy Environments*. PhD thesis, Carnegie Mellon University, 1996.

[49] L. Neumeyer and M. Weintraub. Probabilistic optimum filtering for robust speech recognition. In *Proceedings ICASSP*, volume 1, pages 417–420, 1994.

[50] P. Price, W.M. Fisher, J. Bernstein, and D.S. Pallett. The DARPA 1000-word resource management database for continuous speech recognition. In *Proc. ICASSP*, 1988.

[51] B. Raj, M. L. Seltzer, and R.M. Stern. Robust Speech Recognition: The Case for Restoring Missing Features. In *Proc. of Eurospeech, The Workshop on Consistent and Reliable Acoustic Cues*, Aalborg, Denmark, September 2001.

[52] R.Lippman, E.A. Martin, and D.B. Paul. Multi-style training for robust isolated-word speech recognition. In *Proc. ICASSP*, 1987.

[53] D. Roy. Integration of speech and vision using mutual information. In *Proc. ICASSP*, Istanbul, Turkey, 2000.

[54] R.P.Lippman. Speech recognition by machines and humans. *Speech Communication*, 22(1-15), 1997.

[55] C.W. Seymour and M. Niranjan. An HMM based cepstral-domain speech enhancement scheme. In *Proc. ICSLP*, pages 1595–1598, 1994.

[56] O. Siohan, Y. Gong, and J. Haton. A Bayesian approach to phone duration adaptation for Lombard speech recognition. In *Proc. European Con. Speech Communication Technology*, volume 3, pages 1639–1642, Berlin, September 1993.

[57] V. Stahl, A. Fischer, and R. Bippus. Quantile based noise estimation for spectral subtraction and wiener filtering. In *Proc. ICASSP*, pages 1875–1878, 2000.

[58] A. de la Torre, J.C. Segura, C. Benítez, A.M. Peinado, and A.J. Rubio. Non-linear transformations of the feature space for robust speech recognition. In *Proc. ICASSP*, Orlando, Florida, May 2002.

[59] U. Yapanel, J.H.L. Hansen, R.Sarikaya, and B.Pellom. Robust Digit Recognition in Noise: An Evaluation using the AURORA Corpus. In *Proc. of Eurospeech*, Aalborg, Denmark, September 2001.

[60] S.J. Young, G.Evermann, D.Kershaw, G.Moore, J.Odell, D.Ollason, D.Povey, V.Valtchev, and P.Woodland. *The HTK Book (for HTK Version 3.2).* University of Cambridge, December 2002.