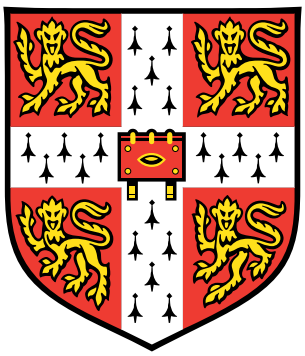
Uncertainty Decoding for Noise Robust Speech Recognition

Hank Liao

Sidney Sussex College
University of Cambridge



September 2007

This dissertation is submitted for the degree of
Doctor of Philosophy to the University of Cambridge.

Declaration

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration. It has not been submitted in whole or in part for a degree at any other university. Some of the work has been published previously in conference proceedings [93, 94, 95] and technical reports [90, 91, 92]. The length of this thesis including appendices, references, footnotes, tables and equations is approximately 53,000 words and contains 38 tables and 41 figures.

Summary

It is well known that the performance of automatic speech recognition degrades in noisy conditions. To address this, typically the noise is removed from the features or the models are compensated for the noise condition. The former is usually quite efficient, but not as effective as the latter, often computationally expensive, approach. This thesis examines a hybrid form of noise compensation called uncertainty decoding that is characterised by transforming the features and a simple acoustic model update that increases the model variances in proportion to the noise level. In particular, a novel approach called joint uncertainty decoding (JUD) is introduced. JUD compensation parameters are derived from the joint distribution between the training and test conditions. Two forms of uncertainty decoding are presented: front-end and model-based joint uncertainty decoding (FE-Joint and M-Joint). An important contribution is it is shown that front-end uncertainty decoding forms, like SPLICE with uncertainty and FE-Joint, can exhibit problems in low SNR that do not occur with model-based forms. Furthermore, M-Joint is as efficient as FE-Joint for the same number of transforms. Thus JUD provides forms that are fast like feature compensation, yet more efficient than standard model-based techniques.

Some common shortcomings of noise robustness techniques are that they only work with stereo data, on small vocabulary systems, are difficult to integrate with other acoustic modelling techniques and are evaluated on artificial data. These are all addressed in this work for JUD. An EM-based ML noise model estimation technique allows JUD transforms to be generated given a sample of the noisy speech from the test environment. An ML approach may update the noise model during speech, can be optimised for the noise compensation type and provide a suitable noise model for multistyle-trained acoustic models. In addition, it is shown how JUD can be combined with CMLLR or semi-tied covariance modelling.

The last main contribution is noise adaptive training using JUD transforms called joint adaptive training (JAT). Instead of forcing the acoustic models to represent extraneous variability introduced by noise in the training data, as is the case for multistyle training, the noise effect is modelled by JUD transforms. Adaptive training with CMLLR or normalisation updates the features and subsequently treats cleaner observations the same as noisier ones. In contrast, during acoustic model training, JAT directly takes into account the noise level of observations by de-weighting them in proportion to the uncertainty. In this way, noisier observations contribute less to the estimation of the canonical model parameters than clean ones. The resulting acoustic models are then purer representations of the speech variability.

JUD is evaluated on small, medium, and large vocabulary tasks, over a wide range of SNR, and artificially corrupted databases as well as actual recorded noisy speech data. The results show that JUD is a flexible, fast, yet powerful noise robustness technique for ASR.

Keywords: speech recognition; noise robustness; hidden Markov models; uncertainty decoding; model-based noise compensation; adaptation; adaptive training.

Acknowledgments

First and foremost, I would like to thank my supervisor Mark Gales for his always insightful suggestions and expert guidance. His unwavering commitment to his students and constant demand for excellence really helped me bring this work to fruition. It has been a privilege and memorable experience to work with Mark.

Secondly, I would like to express my gratitude to Toshiba Research Europe, and in particular Drs. Masami Akamine and Kate Knill, for providing the generous funding that made this research possible. I am grateful to Steve Young and Phil Woodland for providing the excellent research facilities here in the Machine Intelligence Laboratory at Cambridge University. To the many people who have contributed to developing and maintaining HTK, I am indebted to you for providing such useful software for conducted this work. I am most obliged to Anna Langley and Patrick Gosling for their excellent work in managing the computing facilities here in the lab and quickly dealing with spontaneous shutdowns, overheating processors and the incessant demands on space, bandwidth and memory.

I would like to thank Mitch Weintraub and Brian Strope for their early encouragement in the field of speech recognition and noise robustness. I thank Matt Stuttle for his help in preparing the RM corpus in the initial stages of my research. I would also like to thank James Nealand for his diligent assistance in setting up the Toshiba corpus. I appreciate the discussions on my work with those at various conferences in particular Jasha Droppo, Michael Picheny, Li Deng, and Dan Povey; their views gave me different perspectives on my work. I also appreciate Sharmaine and Vidura's kind help at the last stages of writing up this thesis. I acknowledge Sarah Airey, Rogier van Dalen, Darren Green, Andrew Liu, Chris Longworth and Kai Yu for going over various sections in this thesis; special thanks to Mark and Catherine for proofreading and providing useful feedback on large portions of this work. I will indeed miss the supervisions with Mark, discussions about MBR with Catherine, adaptation and notation with Kai, learning about decoders from Andrew, and numerous chats about kernels with Chris.

Thanks to my friends and acquaintances from college, the MCR social circle, volleyball, hockey, the CCC, the CHC and HRM for your support and making my time in Blighty most enjoyable. To my friends back at home and in the Bay Area thanks for always making it feel like I'd never left, despite my short visits and poor attempts at keeping in touch.

And lastly, but most of all, thanks to my family for their unconditional love and support.

Acronyms

ASR	Automatic Speech Recognition
BN	Broadcast News
CMLLR	Constrained MLLR
CMN	Cepstral Mean Normalisation
CVN	Cepstral Variance Normalisation
DBN	Dynamic Bayesian Network
DCT	Discrete Cosine Transform
DPMC	Data-driven PMC
EM	Expectation Maximisation
FFT	Fast Fourier Transform
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
HTK	HMM Toolkit
IDCT	Inverse DCT
IPMC	Iterative PMC
JUD	Joint Uncertainty Decoding
LVCSR	Large Vocabulary Continuous Speech Recognition
MAP	Maximum A Posteriori
MFCC	Mel-Frequency Cepstral Coefficients
ML	Maximum Likelihood
MLLR	Maximum Likelihood Linear Regression
MMSE	Minimum Mean Squared Error
PDF	Probability Density Function
PMC	Parallel Model Combination
POF	Probabilistic Optimal Filtering
RM	Resource Management
SAT	Speaker Adaptive Training
SNR	Signal-to-Noise Ratio
SPLICE	Stereo Piece-wise Linear Compensation for Environments
STC	Semi-tied Covariance
UD	Uncertainty Decoding
VTS	Vector Taylor Series
WER	Word Error Rate
WSJ	Wall Street Journal

Notation

These are terms and notation used throughout this work.

Variables, Symbols and Operations

$\approx$	approximately equal to
$\propto$	proportional to
x	scalar quantity
$\hat{x}$	estimate of the true value of x
$\underset{x}{\text{argmax}}\, f(x)$	value of x that maximises $f(x)$
$\max_x f(x)$	value of $f(x)$, when x maximises $f(x)$
$\log(x)$	natural logarithm of x
$\exp(x)$	exponential of x
$\mathcal{E}\{f(x)\}$	the expected value of $f(x)$, where x is a random variable
$\texttt{Var}\{f(x)\}$	the variance of $f(x)$, where x is a random variable
$f(x)\big\|_{\mu_0}$	evaluate function f(x) at the point μ_0
$h(t) * x(t)$	convolution operator—that is, $\int_{-\infty}^{\infty} h(\tau)x(t-\tau)d\tau$

Vectors and Matrices

$\boldsymbol{x}$	vector of arbitrary dimensions
$\mathcal{R}^D$	D-dimensional Euclidean space
$\boldsymbol{A}$	a matrix
$\boldsymbol{A}_{[p]}$	a projection matrix where the number of rows p is less than the number of columns
$\boldsymbol{A}^\mathsf{T}$	transpose of matrix $\boldsymbol{A}$
$\text{diag}\{\boldsymbol{A}\}$	a diagonalised version of matrix $\boldsymbol{A}$
$\lvert\boldsymbol{A}\rvert$	determinant of matrix $\boldsymbol{A}$

$\boldsymbol{A}^{-1}$	inverse of matrix $\boldsymbol{A}$
$\boldsymbol{a}_i$	column vector that is the ith column of $\boldsymbol{A}$
$\boldsymbol{a}_{\bar{i}}$	row vector that is the ith row of $\boldsymbol{A}$
a_{ij}	scalar value that is the element in row i and column j of $\boldsymbol{A}$
$\boldsymbol{I}$	identity matrix
$\mathbf{1}$	column vector of 1's
Δ_{ij}	all-zero matrix, except for a 1 in row i and column j
$\boldsymbol{b}$	column vector
$\boldsymbol{a} \circ \boldsymbol{b}$	element-wise product of $\boldsymbol{a}$ and $\boldsymbol{b}$ yielding a column vector
$\boldsymbol{a} \bullet \boldsymbol{b}$	dot product of $\boldsymbol{a}$ and $\boldsymbol{b}$, yielding a scalar value

Observations

T	number of frames in a sequence of observations
t	time frame index
D	number of dimensions of full feature vector
D_s	number of dimensions of static, delta, or delta-delta components of static features—therefore $3 \times D_s = D$
d	dimension index
$\boldsymbol{S}$	sequence of clean speech vectors $[\boldsymbol{s}_1\ \boldsymbol{s}_2\ \cdots\ \boldsymbol{s}_T]$
$\boldsymbol{s}_t$	complete clean speech vector, comprised of static, delta and delta-delta clean speech vectors—that is $\boldsymbol{s}_t = [\boldsymbol{x}_t^\mathsf{T}\ \ \Delta\boldsymbol{x}_t^\mathsf{T}\ \ \Delta^2\boldsymbol{x}_t^\mathsf{T}]^\mathsf{T}$
$\boldsymbol{O}$	sequence of noise-corrupted speech vectors $[\boldsymbol{o}_1\ \boldsymbol{o}_2\ \cdots\ \boldsymbol{o}_T]$
$\boldsymbol{o}_t$	complete noise-corrupted speech vector, comprised of static, delta and delta-delta noise-corrupted speech vectors—that is $\boldsymbol{o}_t = [\boldsymbol{y}_t^\mathsf{T}\ \ \Delta\boldsymbol{y}_t^\mathsf{T}\ \ \Delta^2\boldsymbol{y}_t^\mathsf{T}]^\mathsf{T}$
$\boldsymbol{n}_t$	complete additive noise vector, comprised of static, delta and delta-delta additive noise vectors—that is $\boldsymbol{n}_t = [\boldsymbol{z}_t^\mathsf{T}\ \ \Delta\boldsymbol{z}_t^\mathsf{T}\ \ \Delta^2\boldsymbol{z}_t^\mathsf{T}]^\mathsf{T}$
$\boldsymbol{h}$	convolutional noise vector
$\boldsymbol{C}$	discrete cosine transform matrix
$\boldsymbol{C}^{-1}$	inverse discrete cosine transform matrix

Probability and Distributions

$\mathrm{P}(\cdot)$	probability mass function
$\mathrm{p}(\cdot)$	probability density function

$\mathrm{p}(x, y)$	joint probability density function—that is, the probability density of having both x and y
$\mathrm{p}(x\|y)$	conditional probability density function of having x given y
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	multivariate Gaussian distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$
$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$	probability of vector $\boldsymbol{x}$ given a multivariate Gaussian distribution
$\delta(x)$	Dirac delta function, which has value of 0 for $x \neq 0$, integrates to 1
δ_{ij}	Kronecker delta symbol, which equals 1 when $i = j$ and is 0 otherwise
$\Gamma(\cdot)$	Gamma function

HMM Parameters

$\mathcal{M}$	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
$\mathcal{M}_n$	set of noise model parameters
$\boldsymbol{\Theta}$	set of all possible state sequences $\boldsymbol{\theta}$ for a transcription $\mathcal{W}_\mathsf{r}$
$\boldsymbol{\theta}$	sequence of discrete clean speech states $[\theta_1\ \theta_2\ \cdots\ \theta_T]$
$\boldsymbol{\theta}^n$	sequence of discrete noise speech states $[\theta_1^n\ \theta_2^n\ \cdots\ \theta_T^n]$
$\boldsymbol{M}$	set of all possible component sequences $\boldsymbol{m}$ for a transcription $\mathcal{W}_\mathsf{r}$
K	number of GMM components in the front-end model
M	number of GMM components in the full acoustic model
R	number of regression classes—that is the number of clusters of acoustic model components
r_m	regression class for component m
$\check{a}^{(k)}$	parameter a is associated with front-end component k
$a^{(m)}$	parameter a is associated with acoustic model component m
$a^{(r_m)}, a^{(r)}$	parameter a is associated with regression class r_m or just class r
$\check{c}^{(k)}$	component prior associated with front-end component k
$c^{(m)}$	component prior associated with acoustic model component m
$\boldsymbol{\mu}_x^{(m)}, \boldsymbol{\Sigma}_x^{(m)}$	static clean speech mean and variance of component m
$\boldsymbol{\mu}_{\Delta x}^{(m)}, \boldsymbol{\Sigma}_{\Delta x}^{(m)}$	delta clean speech mean and variance of component m
$\boldsymbol{\mu}_{\Delta^2 x}^{(m)}, \boldsymbol{\Sigma}_{\Delta^2 x}^{(m)}$	delta-delta clean speech mean and variance of component m

$\boldsymbol{\mu}_s^{(m)}, \boldsymbol{\Sigma}_s^{(m)}$	complete clean speech mean and variance of component m—that is $\boldsymbol{\mu}_s^{(m)} = \left[\boldsymbol{\mu}_x^{(m)\mathsf{T}} \boldsymbol{\mu}_{\Delta x}^{(m)\mathsf{T}} \boldsymbol{\mu}_{\Delta^2 x}^{(m)\mathsf{T}}\right]^{\mathsf{T}}$
$\boldsymbol{\mu}_y^{(m)}, \boldsymbol{\Sigma}_y^{(m)}$	static noise corrupted speech mean and variance of component m
$\boldsymbol{\mu}_{\Delta y}^{(m)}, \boldsymbol{\Sigma}_{\Delta y}^{(m)}$	delta noise corrupted speech mean and variance of component m
$\boldsymbol{\mu}_{\Delta^2 y}^{(m)}, \boldsymbol{\Sigma}_{\Delta^2 y}^{(m)}$	delta-delta noise corrupted speech mean and variance of component m
$\boldsymbol{\mu}_o^{(m)}, \boldsymbol{\Sigma}_o^{(m)}$	complete noise corrupted speech mean and variance of component m—that is $\boldsymbol{\mu}_o^{(m)} = \left[\boldsymbol{\mu}_y^{(m)\mathsf{T}} \boldsymbol{\mu}_{\Delta y}^{(m)\mathsf{T}} \boldsymbol{\mu}_{\Delta^2 y}^{(m)\mathsf{T}}\right]^{\mathsf{T}}$

Parameter Estimation

$\boldsymbol{\mu}_h$	static channel mean
$\boldsymbol{\mu}_n$	additive noise mean, which is an extended static additive noise mean vector—that is $\boldsymbol{\mu}_n = [\boldsymbol{\mu}_z^{\mathsf{T}}\ \mathbf{0}^{\mathsf{T}}\ \mathbf{0}^{\mathsf{T}}]^{\mathsf{T}}$
$\boldsymbol{\Sigma}_n$	additive noise variance and is comprised of static $\boldsymbol{\Sigma}_z$, delta $\boldsymbol{\Sigma}_{\Delta z}$ and delta-delta additive noise $\boldsymbol{\Sigma}_{\Delta^2 z}$ variances
$\gamma_{s,t}^{(m)}$	posterior probability of component m at time t given the complete clean observation sequence
$\gamma_s^{(m)}$	posterior probability of component m given the complete clean observation sequence, $\gamma_s^{(m)} = \sum_{t=1}^{T} \gamma_{s,t}^{(m)}$
$\gamma_{o,t}^{(m)}$	posterior probability of component m at time t given the complete noisy observation sequence
$\gamma_o^{(m)}$	posterior probability of component m given the complete noisy observation sequence, $\gamma_o^{(m)} = \sum_{t=1}^{T} \gamma_{o,t}^{(m)}$
$\mathcal{Q}(\mathcal{M}; \hat{\mathcal{M}})$	auxiliary function where component posteriors are computed using parameter set $\mathcal{M}$ and output distribution probabilities with $\hat{\mathcal{M}}$
$\mathcal{W}_{\mathsf{r}}$	sequence of words that are the reference transcription of data
$\mathcal{W}_{\mathsf{h}}$	sequence of words that are the hypothesised transcription from a decoding pass over data

Table of Contents

List of Tables

List of Figures

Chapter 1
Introduction

Automatic speech recognition (ASR) has improved markedly over the last decade such that it can be used to transcribe speech in a variety of domains such as consumer goods[1], call centre applications[2] and desktop personal computer software[3]. However, recognition accuracy is still far from human levels. Humans make mistakes at a rate of less than one hundredth of a percent [97] when recognising strings of digits, while the best machine error rates have only advanced from 0.72% to 0.55% over the last decade [155]. For more difficult tasks the difference narrows: for example on telephone conversation transcription [56] the human word error rate is about 4% while state-of-the-art automatic transcription systems rates are still over three times worse [18, 32]. The difference between human and machine performance has been attributed to a variety of causes including: the immense variability of speech [105], poor modelling of spontaneous speech [97, 112], fundamental limitations in conventional speech feature extraction [107] and the statistical framework [13]. Despite this "performance gap", basic ASR technology has advanced to a level where it may be applied in a variety of commercial applications. However, a major problem is robustness to noise. Despite decades of research on noise robustness, leading researchers in the field such as Nelson Morgan and Sadaoki Furui have called on a serious effort to improve recogniser performance in noise [38]. The reason for poor accuracy in noise is a mismatch between the original conditions of the data used to *train* the recogniser and the actual noisy environment it is

[1] For example, voice-dialling in mobile phones or controlling toys such as the robotic dog AIBO (1999), or interactive doll Amazing Amanda (2005). Visit the Saras institute (http://www.sarasinstitute.org/) for an extensive history.

[2] Examples include Charles Schwab's stock trading and lookup system (Nuance), Cineworld's Movieline for movie information and booking (Telephonetics), or Verizon's 411 directory assistance (Microsoft/Tellme).

[3] Such as Dragon NaturallySpeaking, IBM ViaVoice or Microsoft's Whisper ASR Engine for its Windows operating system.

tested in. While human speech recognition degrades only slightly in noise, machine error rates increase dramatically even with noise compensation [97].

There are many challenges when building a speech recogniser that is robust to environmental noise. Noise is unpredictable and has a variety of properties: additive, e.g. car or fan noise; convolutional, e.g. different microphones; or non-stationarity, e.g. other people talking, keyboard clicks, lip smacks, or doors slams. Estimating noise accurately is not trivial and its effects on speech are complex. Even if an accurate model of the noise is available, there is the classic trade-off between computational efficiency and performance. Noise compensation algorithms may be used in small, embedded devices where memory and computational power are constrained. This limits the range of robustness techniques that can be employed. Though large call centre deployments use a computing grid to serve thousands of calls simultaneously, an increase in computational cost directly affects the size of the grid. Poor performance of speech recognition may be blamed on noise, even when it may not be heard[1]. Undoubtedly though, improving speech recognition performance in noise will help the adoption of this enabling technology.

Most noise robustness methods can be classified by how they address this problem. Standard approaches are: *inherently robust front-ends*, *front-end compensation* and *model-based compensation*. The first seeks speech features that are immune to noise. While performance may be acceptable in low-levels of background noise [67], an inherently robust front-end has yet to be developed that can handle higher and varied noise levels. Hence, the focus of much research has turned to feature enhancement, or cleaning, whereby noise is explicitly removed from the observed speech to better match the clean models of speech. Alternatively, the acoustic model parameters can be updated to reflect the effects of noise. This is a more powerful technique since each component can be individually adapted to account for the degree to which the noise affects its mean and variance. However improved results typically come with a significant computational cost. Hence model-based approaches may be impractical for some commercial LVCSR applications, let alone embedded devices.

Recently, research has been directed at incorporating the uncertainty due to noise into ASR. The "observation uncertainty" method [4, 5] incorporates the variance of the feature enhancement process into the decoding step representing the residual enhancement uncertainty. This is different to "uncertainty decoding" [26, 85], which is based on a dynamic Bayesian network inference approach[2]. Both these techniques 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 representations of the uncertainty such that the model variance updates are fast to compute, yet effectively compensate for noise. Ideally, uncertainty decoding will be efficient like feature-based compensation forms, yet as powerful as more computationally expensive model-based adaptation techniques. This thesis examines new approaches to uncertainty decoding for noise robust speech recognition.

[1] See this ASR demonstration gone awry http://www.youtube.com/watch?v=IkeC7HpsHxo

[2] *Observation uncertainty* has also been referred to as *uncertainty decoding*, but for clarity this work distinguishes these two approaches.

1.1 Organisation of Thesis

Following this introduction, a brief overview of automatic speech recognition using hidden Markov models (HMMs) is given in chapter 2 along with semi-tied covariance modelling, adaptation and adaptive training, which will all be evaluated in this work. A model of the noisy acoustic environment is presented, along with the effects that noise has on ASR, in chapter 3. Chapter 4 reviews some relevant noise robustness techniques. In particular, observation uncertainty, SPLICE with and without certainty, and VTS compensation are discussed because they provide interesting theoretical and practical comparisons for uncertainty decoding. Joint uncertainty decoding is formally presented in chapter 5. A method for estimating a model of the noise to predict JUD transforms is given in chapter 6. These Joint transforms are applied in an adaptive training framework described in chapter 7. In chapter 8, experimental results on artificially corrupted corpora, Aurora2 and Resource Management, are presented. The next chapter looks at evaluating the various techniques on speech recorded in noisy conditions such as Broadcast News and Toshiba Research Europe's internal collection of in-car speech data. Finally, conclusions and future research directions are presented in chapter 10.

Chapter 2
Hidden Markov Model Speech Recognition

Automatic speech recognition is a classic pattern recognition problem where the goal is to automatically produce a text transcription of spoken words. Major concerns are finding compact set of classification features and determining a suitable means of recognising words from these features. The features should be a compact representation of the audio signal that is optimal for discrimination. The majority of recognisers use HMMs as models of speech although how they are trained can vary. Finding the actual transcription should be efficient as well as accurate and is also known as decoding. This chapter describes this standard approach to automatic speech recognition in detail.

2.1 Overview of ASR

The main components of a generic speech recognition system, or recogniser, and how they interact are shown in figure 2.1. The input speech, captured by some transducer, is processed by the front-end to provide a compact and effective set of features for recognition. The front-end may perform some speech detection, also known as endpointing, to remove background silence or noise reduction before passing feature vectors to the decoder. Given the features provided by the front-end, the goal is to classify or "recognise" the speech uttered. This amounts to "decoding" the most likely word sequence $\mathcal{W}_{\mathsf{h}}$, i.e. the hypothesis, given the observation sequence $\boldsymbol{S}$ and a set of model parameters $\bar{\mathcal{M}}$

$$\mathcal{W}_{\mathsf{h}} = \underset{\mathcal{W}}{\operatorname{argmax}} \, \mathrm{P}(\mathcal{W}|\boldsymbol{S}; \bar{\mathcal{M}}) \tag{2.1}$$

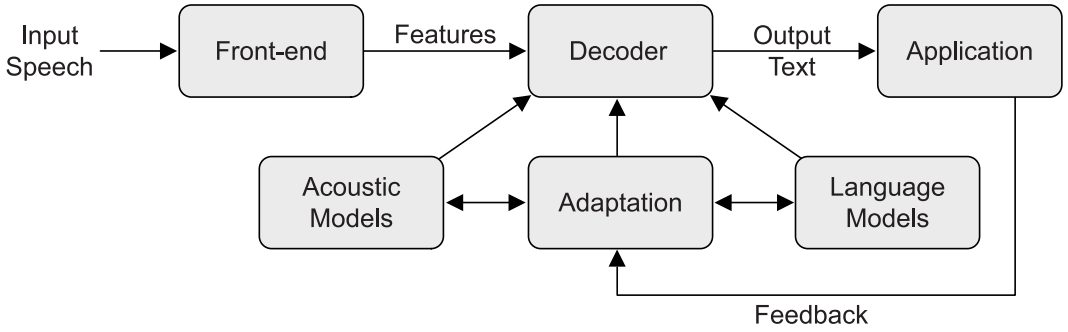


Figure 2.1: Architecture of a speech recognition system.

Rather than directly estimate the posterior probability of a word sequence, Bayes' rule may be applied to compute the posterior as the product of the class conditional distribution and the prior

$$\begin{aligned} \mathcal{W}_{\mathsf{h}} &= \underset{\mathcal{W}}{\operatorname{argmax}}\, \mathrm{P}(\mathcal{W}|\boldsymbol{S};\bar{\mathcal{M}}) \\ &= \underset{\mathcal{W}}{\operatorname{argmax}}\, \frac{\mathrm{p}(\boldsymbol{S}|\mathcal{W};\mathcal{M})\,\mathrm{P}(\mathcal{W};\dot{\mathcal{M}})}{\mathrm{p}(\boldsymbol{S};\bar{\mathcal{M}})} \\ &= \underset{\mathcal{W}}{\operatorname{argmax}}\, \mathrm{p}(\boldsymbol{S}|\mathcal{W};\mathcal{M})\,\mathrm{P}(\mathcal{W};\dot{\mathcal{M}}) \end{aligned} \tag{2.2}$$

where the complete set of model parameters $\bar{\mathcal{M}}$ are comprised of the acoustic model $\mathcal{M}$ and language model $\dot{\mathcal{M}}$. Often $\mathrm{p}(\boldsymbol{S}|\mathcal{W};\mathcal{M})$ is referred to as the acoustic score and $\mathrm{P}(\mathcal{W};\dot{\mathcal{M}})$ the language score. The normalising probability of the observation sequence $\mathrm{p}(\boldsymbol{S};\bar{\mathcal{M}})$ is not needed since it is independent of the word sequence. Hence the decoder incorporates acoustic and language models to produce a word sequence that maximises the posterior probability of the feature sequence.

The recogniser is limited to only recognising words that exist in a known vocabulary although tests may be conducted where the words are known beforehand, i.e. *closed* vocabulary, or not, i.e. *open* vocabulary. Words that are not in the vocabulary are labelled *out-of-vocabulary* (OOV), and depending on scoring rules, may not be counted in error rates. There is a direct impact on the number of words in the vocabulary on the computational cost of searching for the optimal word sequence. Often, recognition tasks are categorised as small vocabulary, when the number of words is less than 100; medium vocabulary when the number of words is greater than 100, but less than 5000; and large vocabulary for 5000 or more words. The number of words is rather arbitrary in the definitions, but they give a sense of task complexity.

The optimal word sequence, or sometimes a word lattice [72], confusion network [31], or list of possible transcriptions, is then passed to the application. The application may simply provide transcriptions, where post-processing could be required to add punctuation and capitalisation, or might be a call center application in which case an intention may need to be discerned from the speech to direct a dialog interaction. Most ASR systems will also use some form of adaptation to increase accuracy by improving acoustic or language modelling. The focus of this work is on ASR noise robustness and improving the core acoustic modelling aspects: the front-end, decoder, acoustic models and adaptation. The following sections discuss these in more detail.

2.2 Front-end Processing

The front-end processes audio to produce or "extract" features that ideally are optimal for speech recognition and invariant to extraneous factors such as different speakers, microphones or environmental noise. The front-end stage may also be divided into two steps: segmentation and feature extraction [27]. The first involves isolating relevant speech segments from the background. For example, in dialogue systems a speech detector, or end-pointer, senses the beginning and end points of speech from the background. Or in broadcast news transcription, a segmentation stage may precede the front-end to remove the opening titles, musical interludes and commercials [131]. Once these segments are identified, they are processed to yield salient features for classification.

In the feature extraction stage, the speech signal captured by the microphone is sampled and digitised into discrete samples over time. A popular feature representation is mel frequency cepstrum coefficients (MFCC) [20], which arise from a homomorphic transform of the short-term spectrum expressed on a mel frequency scale. Figure 2.2 shows how they may be computed. Their use is motivated by both perceptual and performance aspects. In speech production, the vocal tract may be viewed as a filter acting on a sound source, such as the glottis—this is the source-filter model [57, 72, 125]. In continuous speech, it has been noted that the vocal tract changes shape slowly in continuous speech; therefore at small enough time scales, on the order of 10 ms, it may be considered a filter of fixed characteristics [125]. Hence, a short-time Fourier transform is applied, converting the time domain signal into the frequency or spectral domain. A first-order pre-emphasis filter is usually applied to accen-

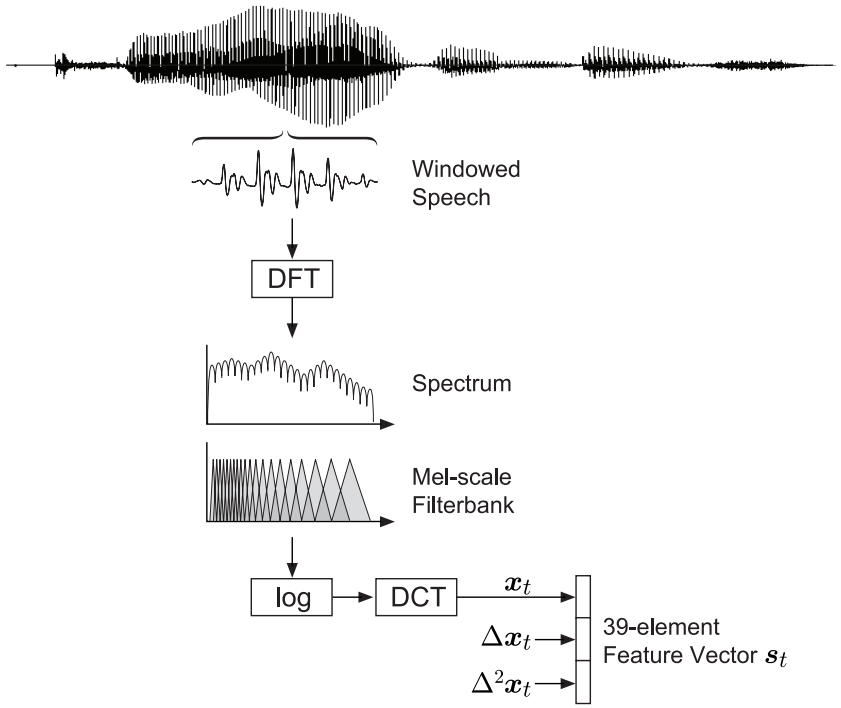


Figure 2.2: Front-end processing for MFCC. Speech waveform converted to smoothed short-term log spectrum every 10 ms. Discrete cosine transform is applied and dynamic terms appended to produce the complete feature vector $\boldsymbol{s}_t$.

tuate the higher frequencies in the formant structure. The signal is windowed at intervals to produce frames of speech. A typical frame rate is 10 ms, with a window size of 25 ms. Overlapping the windows and using a Hamming window smooths transitions between frames and reduces frame edge discontinuities respectively. The discrete Fourier transform (DFT) is applied to compute the spectrum. Because of the short time steps, each frame can be a considered stationary signal.

Normally as the frequency rises, greater increases in frequency are necessary to double the *perceived* pitch—the mel scale [20] warps the frequency scale by logarithmically compressing it to linearise with relation to perceived pitch. The spectral envelope may be sampled by a series of triangular band-pass filters, to produce a set of $\boldsymbol{m}$ filter bank coefficients. The number of filters can vary, usually between 20-40, where more filters are used with a larger bandwidth. Each filter captures the spectral energy for its particular band or "bin". The filters are spaced according to the mel scale

$$f_{\mathsf{mel}} = 1127 \log\left(1 + \frac{f_{\mathsf{Hz}}}{700}\right) \tag{2.3}$$

where f_{mel} is the mel frequency and f_{Hz} is standard frequency. For low frequencies, the spacing is approximately linear, but at high frequencies it becomes logarithmic. Hence the filterbank reduces the normal FFT window of 256 points to a set of 20-40 smoothed filterbank coefficients or channels. The dynamic range of each filterbank output is also compressed using the natural logarithm to give log-spectral filterbank coefficients.

Although these log-spectral coefficients may be used directly as features for speech recognition, they are highly correlated, e.g. loudness is reflected in most parameters, and hence a poor representation. This motivates applying the discrete cosine transform (DCT) to decorrelate the features and compact information into lower-frequency cepstral coefficients. Cepstral processing is also homomorphic, allowing separation of the source excitation from the vocal tract filter. Cepstral coefficients may be derived as follows, where $x_{t,d}$ is the coefficient at time t for dimension d

$$x_{t,d} = \sqrt{\frac{2}{N}} \sum_{i=1}^{N} \log(x_{f_i,t}) \cos\left(\frac{\pi d}{N}(i - 0.5)\right) \tag{2.4}$$

and $x_{f_i,t}$ is the energy output of filter i [152]. Thus equation (2.4) transforms the log-spectral features into MFCC. MFCC are not strictly homomorphic, since the natural log is applied after the filterbank smoothing, however they are approximately homomorphic for filters with smooth transfer functions [72], e.g. the vocal tract response. Due to the energy compacting effect of the DCT, the number of coefficients can be limited to 13; increasing the number has shown not to reduce the error rate [72]. The 0th cepstral coefficient is sometimes replaced with a normalised log energy coefficient [152]. Although MFCC are a widely used speech parameterisation, its optimality has been questioned [63, 65, 67]. Alternatively, perceptual linear prediction (PLP) coefficients have been used [64] giving similar performance to MFCC [69]. An extensive review of speech signal representations can be found in Huang et al. [72] or Gold and Morgan [57].

2.2.1 Dynamic Features

The set of "static" MFCC features are often appended with additional coefficients to explicitly model the changing speech signal at each time instance [37]. This is done to overcome the limited temporal modelling of current HMMs, specifically breaking the conditional independence assumption discussed in the next section. Dynamic features may be computed by using simple differences, e.g. $\Delta \boldsymbol{x}_t = \boldsymbol{x}_{t+2} - \boldsymbol{x}_{t-2}$, or linear regression

$$\Delta \boldsymbol{x}_t = \frac{\sum_{\delta=1}^{\Delta} \delta(\boldsymbol{x}_{t+\delta} - \boldsymbol{x}_{t-\delta})}{2\sum_{\delta=1}^{\Delta} \delta^2} \tag{2.5}$$

where $\boldsymbol{x}_t$ is the vector of static cepstral features, and $\Delta \boldsymbol{x}_t$ the dynamic features. A window size of $\Delta = 1$ gives coefficients that are simply the difference between the previous and following frame. A large window size of $\Delta = 2$ gives a more robust estimate of dynamic coefficients. As noted in [39], delta parameters may be considered an approximation to the first derivative of the static parameters; hence in the Continuous-Time approximation, time derivatives of the static coefficients may be used as dynamic coefficients. Higher order delta-delta acceleration coefficients may be computed in similar manner using simple differences or linear regression. For a front-end that produces 13 static MFCC or PLP coefficients, the addition of these velocity and acceleration parameters results in a feature vector of 39 elements with the feature vector structured as follows

$$\boldsymbol{s}_t = \begin{bmatrix} \boldsymbol{x}_t \\ \Delta \boldsymbol{x}_t \\ \Delta^2 \boldsymbol{x}_t \end{bmatrix} \tag{2.6}$$

It is shown in Huang et al. [72] that a greater number of cepstral coefficients or third-order dynamic coefficients do not improve system accuracy.

2.3 Acoustic Modelling

HMMs have proven to be a powerful means of representing time varying signals, such as speech, as a parametric random process [72, 118]. In ASR, an HMM is used to model the acoustics of each word, syllable or phone to generate the acoustic score $\mathrm{p}(\boldsymbol{S}|\mathcal{W}, \mathcal{M})$ in equation (2.2). The HMM is a first-order, discrete-time Markov chain, as depicted in figure 2.3,

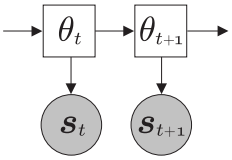


Figure 2.3: Dynamic Bayesian network for speech recognition. Arrows indicate dependencies, observed variables are shaded, and hidden variables unshaded. Circles represent continuous variables, squares discrete.

with a hidden state sequence. Two key assumptions are made. With discrete time, the *first-order Markov assumption* assumes the probability of a state θ_t at time t is only dependent on the previous state θ_{t-1}

$$\mathrm{P}(\theta_t|\boldsymbol{\theta}_1^{t-1}) \approx \mathrm{P}(\theta_t|\theta_{t-1}) \tag{2.7}$$

where $\boldsymbol{\theta}_1^{t-1} = \theta_1, \theta_2, \ldots, \theta_{t-1}$. This provides some memory of past events, but does not explicitly require the storage of all past states. In figure 2.3 this is reflected by the left-to-right dependency arrow. The hidden aspect of HMMs is that the state sequence is not actually observed and thus unknown. Rather the hidden state at time t, θ_t, emits an observation $\boldsymbol{s}_t$. This leads to the second assumption, that an observation is *conditionally independent* of all other observations given the state

$$\mathrm{p}(\boldsymbol{s}_t|\boldsymbol{S}_1^{t-1}, \boldsymbol{\theta}_1^t) \approx \mathrm{p}(\boldsymbol{s}_t|\theta_t) \tag{2.8}$$

In figure 2.3 this is reflected by the downward pointing arrows that indicate the observation is only dependent on the hidden state, and hence conditionally independent of all other observations. The dynamic features previously discussed in section 2.2.1 violate this assumption, but are useful in modelling the continuous nature of speech by incorporating neighbouring frames. These assumptions imply that transitions between states are *instantaneous* and whilst in a state, observations are *stationary*. Given that speech itself is continuous in nature, not piece-wise stationary and can exhibit long-term dependencies, these assumptions are poor, although HMMs continue to be the dominant acoustic model form for ASR.

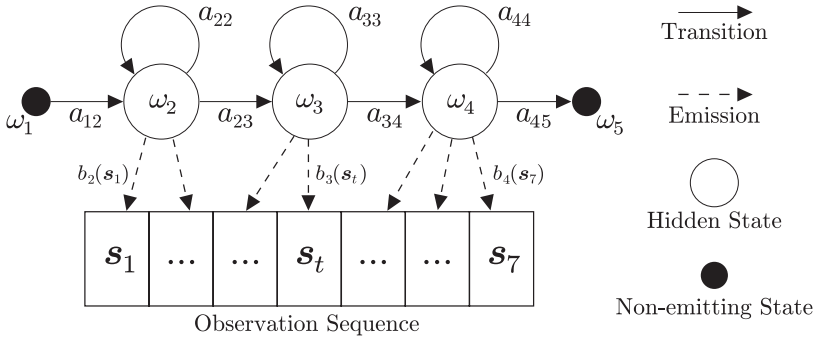


Figure 2.4: First-order HMM with left-to-right topology and three emitting states.

A three emitting state HMM with a left-to-right topology and no skips is shown in figure 2.4. The use of a start/initial state and end/absorbing state allow HMMs to be easily concatenated. In general HMMs may have initial probabilities of each state, typically denoted by π, but this notion is not relevant by incorporating these two non-emitting states. The observed speech sequence of T feature vectors is $\boldsymbol{S} = [\boldsymbol{s}_1 \cdots \boldsymbol{s}_t \cdots \boldsymbol{s}_{T=7}]$ while the hidden state sequence is $\boldsymbol{\theta} = [\theta_1 = \omega_2 \cdots \theta_t = \omega_3 \cdots \theta_{T=7} = \omega_4]$. The transition probability between the first state and the second is always one, i.e. $a_{12} = 1$. Valid transitions for emitting states are either to the next state, or self loop. The transition probability from state i to j is defined as

$$a_{ij} = \mathrm{P}(\theta_t = \omega_j|\theta_{t-1} = \omega_i; \mathcal{A}) \tag{2.9}$$

and for every state ω_i, all the transitions out should sum to one, $\sum_{j=1}^{J} a_{ij} = 1$, except the last, non-emitting, absorbing state. The output or observation emission probability for state ω_j is defined as

$$b_j(\boldsymbol{s}_t) = \mathrm{p}(\boldsymbol{s}_t|\theta_t = \omega_j; \mathcal{B}) \tag{2.10}$$

The set of transition probabilities $\mathcal{A}$, also expressible as a matrix, and output distribution parameters $\mathcal{B}$, constitute the acoustic model $\mathcal{M}$.

Now that the HMM topology for speech recognition has been described, the three main issues in using them, as discussed in Rabiner [118], are:

- **Likelihood Evaluation**: Given an acoustic model $\mathcal{M}$, what is the likelihood that it generated the observed speech $\boldsymbol{S}$?
- **Parameter Estimation**: Given training observations $\boldsymbol{S}$, their transcription $\mathcal{W}_\mathsf{r}$, and the topology of the HMM, including the number of states and output distribution form, how are the acoustic model parameters $\mathcal{M}$ to be estimated?
- **Decoding**: Given an acoustic model $\mathcal{M}$, what is the most likely hidden state sequence $\hat{\boldsymbol{\theta}}$ that generated a segment of observed speech $\boldsymbol{S}$?

The following subsections give solutions to the first two issues, while the last is covered in section 2.4.2.

2.3.1 Likelihood Evaluation

The probability of the observed clean speech sequence $\boldsymbol{S}$, for a transcription $\mathcal{W}_\mathsf{r}$ and the HMM parameters $\mathcal{M}$, is

$$\mathrm{p}(\boldsymbol{S}|\mathcal{W}_\mathsf{r};\mathcal{M}) = \sum_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \mathrm{p}(\boldsymbol{S}|\boldsymbol{\theta};\mathcal{M})\,\mathrm{P}(\boldsymbol{\theta}|\mathcal{W}_\mathsf{r};\mathcal{M}) \tag{2.11}$$

where $\boldsymbol{\Theta}$ is the set of all possible state sequences $\boldsymbol{\theta}$ for the transcription $\mathcal{W}_\mathsf{r}$. This may be approximated by applying the first-order Markov and conditional independence assumptions

$$\mathrm{p}(\boldsymbol{S}|\mathcal{W}_\mathsf{r};\mathcal{M}) \approx \sum_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \prod_{t=1}^{T} \mathrm{p}(\boldsymbol{s}_t|\theta_t;\mathcal{M})\,\mathrm{P}(\theta_t|\theta_{t-1};\mathcal{M}) \tag{2.12}$$

As given in Rabiner [118], this probability may be recursively computed efficiently by using a forward probability, defined as the joint probability of the partial observation sequence $\boldsymbol{s}_1\boldsymbol{s}_2\cdots\boldsymbol{s}_t$ *and* state ω_j at time t given the transcription $\mathcal{W}_\mathsf{r}$

$$\begin{aligned}
\alpha_j(t) &= \mathrm{p}(\boldsymbol{s}_1\boldsymbol{s}_2\cdots\boldsymbol{s}_t, \theta_t = \omega_j|\mathcal{W}_\mathsf{r};\mathcal{M}) = \mathrm{p}(\boldsymbol{S}_1^t, \theta_t = \omega_j|\mathcal{W}_\mathsf{r};\mathcal{M}) \\
&= \mathrm{p}(\boldsymbol{s}_t|\theta_t = \omega_j;\mathcal{M}) \sum_{i=2}^{J-1} \mathrm{P}(\theta_t = \omega_j|\theta_{t-1} = \omega_i;\mathcal{M})\,\mathrm{p}(\boldsymbol{S}_1^{t-1}, \theta_{t-1} = \omega_i|\mathcal{W}_\mathsf{r};\mathcal{M}) \\
&= b_j(\boldsymbol{s}_t) \sum_{i=2}^{J-1} a_{ij}\alpha_i(t-1)
\end{aligned} \tag{2.13}$$

for $1 < t \leq T$, where T is the index of the last frame of the observation sequence $\boldsymbol{S}$, and J is the number of states in the HMM, and the following two initial conditions

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

$$\alpha_j(1) = a_{1j}b_j(\boldsymbol{s}_1), \text{ for } 1 < j < J \tag{2.15}$$

where $\alpha_1(1)$ represents the probability at the start state θ_1 and $a_{1j} = 0$ for all j, except when $j = 2$ where $a_{12} = 1$, for the topology shown in figure 2.4. The terminating step at time T is

$$\alpha_J(T) = \sum_{i=2}^{J-1} \alpha_i(T)a_{iJ} \tag{2.16}$$

Equation (2.12) may now simply be expressed as

$$\mathrm{p}(\boldsymbol{S}|\mathcal{W}_\mathsf{r};\mathcal{M}) = \alpha_J(T) \tag{2.17}$$

This is the forward probability of being in the final state ω_J. This recursive form is far more efficient at $\mathcal{O}\{(J)^2T\}$ rather than $\mathcal{O}\{(J)^TT\}$ to exhaustively evaluate each possible state sequence individually.

A backward probability [118], may be similarly defined. It is the conditional probability the model will generate the rest of the observation sequence from time t, *given* $\theta_t = \omega_i$ and transcription $\mathcal{W}_\mathsf{r}$

$$\begin{aligned}\beta_i(t) &= \mathrm{p}(\boldsymbol{s}_{t+1}\boldsymbol{s}_{t+2}\cdots\boldsymbol{s}_T|\theta_t=\omega_i,\mathcal{W}_\mathsf{r};\mathcal{M}) = \mathrm{p}(\boldsymbol{S}_{t+1}^T|\theta_t=\omega_i,\mathcal{W}_\mathsf{r};\mathcal{M}) \\ &= \sum_{j=2}^{J-1}\mathrm{P}(\theta_{t+1}=\omega_j|\theta_t=\omega_i;\mathcal{M})\,\mathrm{p}(\boldsymbol{s}_{t+1}|\theta_{t+1}=\omega_j;\mathcal{M})\,\mathrm{p}(\boldsymbol{S}_{t+2}^T|\theta_{t+1}=\omega_j,\mathcal{W}_\mathsf{r};\mathcal{M}) \\ &= \sum_{j=2}^{J-1} a_{ij}b_j(\boldsymbol{s}_{t+1})\beta_j(t+1)\end{aligned} \tag{2.18}$$

for $1 \leq t < T$ and initial conditions

$$\beta_J(T) = 1 \tag{2.19}$$

$$\beta_i(T) = a_{iJ}, \text{ for } 1 < i < J \tag{2.20}$$

and terminating condition at $t = 1$

$$\beta_1(1) = \sum_{j=2}^{J-1} a_{1j}b_j(\boldsymbol{s}_1)\beta_j(1) \tag{2.21}$$

The backward probability may now be used in equation (2.17): $\mathrm{p}(\boldsymbol{S}|\mathcal{W}_\mathsf{r};\mathcal{M}) = \beta_1(1)$.

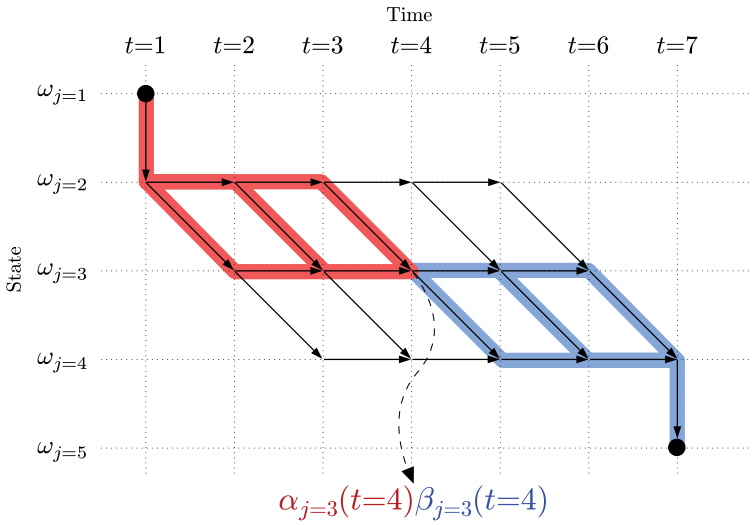


Figure 2.5: The relationship between joint probability $\alpha_i(t)$ and the conditional probability $\beta_i(t)$ in the forward-backward algorithm. The forward path is highlighted in red and backward in blue. The likelihood of state ω_j at time t is given by $\gamma_{s,t=4}^{(j=3)} = \alpha_{j=3}(t=4)\beta_{j=3}(t=4))$.

Figure 2.5 illustrates the relationship between the forward and backward probabilities. Thus, the forward and backward probabilities can be combined to give the likelihood of state ω_j at time t, given $\boldsymbol{S}$ and $\mathcal{W}_{\mathsf{r}}$. This probability may then be simply expressed as

$$\begin{aligned}\gamma_{s,t}^{(j)} &= \mathrm{P}(\theta_t{=}\omega_j|\boldsymbol{S},\mathcal{W}_{\mathsf{r}};\mathcal{M}) \\ &= \frac{\mathrm{P}(\theta_t{=}\omega_j,\boldsymbol{S}|\mathcal{W}_{\mathsf{r}};\mathcal{M})}{\mathrm{P}(\boldsymbol{S}|\mathcal{W}_{\mathsf{r}};\mathcal{M})} = \frac{\alpha_j(t)\beta_j(t)}{\mathrm{P}(\boldsymbol{S}|\mathcal{W}_{\mathsf{r}};\mathcal{M})}\end{aligned} \tag{2.22}$$

such that $\sum_{j=1}^{J}\gamma_{s,t}^{(j)}=1$. The subscript s reinforces the fact that $\gamma_{s,t}^{(j)}$ is computed from the training data $\boldsymbol{S}$. Though the forward probability $\alpha_j(t)$ and backward probability $\beta_j(t)$ may be considered reversed counterparts, there is asymmetry: the forward probability includes the output probability at time t, $b_j(\boldsymbol{s}_t)$, whereas the backward does not.

The form of the output distribution has so far not been discussed. The majority of continuous density HMM-based speech recognisers use a multivariate Gaussian mixture model (GMM) to model the state emission probability, although mixtures of Laplacian distributions have also been used successfully [61].

$$\begin{aligned}b_j(\boldsymbol{s}_t) &= \mathrm{p}(\boldsymbol{s}_t|\theta_t=\omega_j;\mathcal{M}) \\ &= \sum_{m=1}^{M} c^{(jm)}\,\mathrm{p}(\boldsymbol{s}_t|\theta_t=\omega_j,m_t=m;\mathcal{M}) \\ &= \sum_{m=1}^{M} c^{(jm)}\mathcal{N}\big(\boldsymbol{s}_t;\boldsymbol{\mu}_s^{(jm)},\boldsymbol{\Sigma}_s^{(jm)}\big)\end{aligned} \tag{2.23}$$

where m indexes a model component Gaussian in the GMM and m_t denotes the component at time t. The component prior weights $c^{(jm)}$ are constrained to be positive and sum to one

$$\sum_{m=1}^{M} c^{(jm)} = 1 \tag{2.24}$$

The subscript s notation indicates the parameters are derived from the training data $\boldsymbol{S}$[1]. Each component of the GMM is a multivariate Gaussian of the form

$$\mathcal{N}\big(\boldsymbol{s}_t;\boldsymbol{\mu},\boldsymbol{\Sigma}\big) = \frac{1}{(2\pi)^{\frac{D}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}}\exp\left\{-\frac{1}{2}\big(\boldsymbol{s}_t-\boldsymbol{\mu}\big)^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\big(\boldsymbol{s}_t-\boldsymbol{\mu}\big)\right\} \tag{2.25}$$

where D is the number of feature dimensions. Since the term outside the exponential is not dependent on the observation, it may be pre-computed and cached for efficiency. As discussed in section 2.3.4, feature dimensions may be correlated, therefore full covariances should be used. However, there are two issues with using full covariance matrices. There may not be sufficient data to robustly estimate them is the first. Secondly, there is a significant increase in the number of operations necessary to compute equation (2.23): full covariances are $\mathcal{O}\{D(D-1)\}$ compared to diagonal ones at $\mathcal{O}\{2D\}$. Hence diagonal covariance matrices are often used. A GMM with diagonal covariances can model some of these intra-frame correlations as well as more complex multimodal and skewed distributions compared to using a single Gaussian output distribution. More advanced covariance modelling methods [130], that are not as expensive as using full covariances, can give improved results, e.g. semi-tied covariance matrices [41] discussed in section 2.3.4

[1] Later it will be important to distinguish these parameters from those associated with mismatched test data $\boldsymbol{O}$ which are subscripted with o.

2.3.2 Parameter Estimation

The HMM parameters may be estimated using the Maximum Likelihood (ML) criterion. The optimal set of parameters should maximise the log likelihood of the training data for the reference transcription $\mathcal{W}_\mathsf{r}$

$$\mathcal{L}(\mathcal{M}) = \log \mathrm{p}(\boldsymbol{S}|\mathcal{W}_\mathsf{r}; \mathcal{M}) \tag{2.26}$$

where $\mathcal{L}$ is the log-likelihood function. There is no known way to analytically solve for the acoustic model which globally maximises this function [118]. One iterative method to find a local maximum is the Baum-Welch (BW) or forward-backward algorithm [9]. This is an example of the expectation maximisation (EM) algorithm [21]. Instead of maximising $\mathcal{L}$, an auxiliary function $\mathcal{Q}$ is optimised which guarantees the log-likelihood will not decrease

$$\mathcal{L}(\hat{\mathcal{M}}) - \mathcal{L}(\mathcal{M}) \geq \mathcal{Q}(\mathcal{M}; \hat{\mathcal{M}}) - \mathcal{Q}(\mathcal{M}; \mathcal{M}) \tag{2.27}$$

which is derived using Jensen's inequality [21] and $\mathcal{M}$ is the current model and $\hat{\mathcal{M}}$ the re-estimated model. The auxiliary function is defined as

$$\begin{aligned}
\mathcal{Q}(\mathcal{M}; \hat{\mathcal{M}}) &= \mathbb{E}_{\mathcal{M}}\Big[\log \mathrm{p}(\boldsymbol{S}, \boldsymbol{\Theta}, \boldsymbol{M}|\hat{\mathcal{M}})\Big] \\
&= \sum_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \sum_{\boldsymbol{m}\in\boldsymbol{M}} \frac{\mathrm{p}(\boldsymbol{S}, \boldsymbol{\theta}, \boldsymbol{m}|\mathcal{W}_\mathsf{r}; \mathcal{M})}{\mathrm{p}(\boldsymbol{S}|\mathcal{W}_\mathsf{r}; \mathcal{M})} \log \mathrm{p}(\boldsymbol{S}, \boldsymbol{\theta}, \boldsymbol{m}|\mathcal{W}_\mathsf{r}; \hat{\mathcal{M}}) \\
&= \sum_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \sum_{\boldsymbol{m}\in\boldsymbol{M}} \frac{\mathrm{p}(\boldsymbol{S}, \boldsymbol{\theta}, \boldsymbol{m}|\mathcal{W}_\mathsf{r}; \mathcal{M})}{\mathrm{p}(\boldsymbol{S}|\mathcal{W}_\mathsf{r}; \mathcal{M})} \Big(\log \mathrm{p}(\boldsymbol{S}|\boldsymbol{\theta}, \boldsymbol{m}, \mathcal{W}_\mathsf{r}; \hat{\mathcal{B}}) + \log \mathrm{P}(\boldsymbol{\theta}, \boldsymbol{m}|\mathcal{W}_\mathsf{r}; \hat{\mathcal{A}})\Big)
\end{aligned} \tag{2.28}$$

where $\mathbb{E}_{\mathcal{M}}$ denotes the expectation over all possible hidden state sequences $\boldsymbol{\Theta}$ and for all possible hidden component sequences $\boldsymbol{m}$ in $\boldsymbol{M}$ for the transcription $\mathcal{W}_\mathsf{r}$, given the observation sequence $\boldsymbol{S}$ computed with parameter set $\mathcal{M}$. Since the transition parameters $\hat{\mathcal{A}}$ and the output distribution parameters $\hat{\mathcal{B}}$ are separate in the summation they may be estimated separately. The overall EM algorithm may be described in the following manner

Initialise $\mathcal{M}^i$, $i = 0$
Do
  E-step: compute $\mathcal{Q}(\mathcal{M}^i; \mathcal{M}^i)$
  M-step: estimate $\mathcal{M}^{i+1} = \mathrm{argmax}_{\mathcal{M}}\, \mathcal{Q}(\mathcal{M}^i; \mathcal{M})$
While $\mathcal{Q}(\mathcal{M}^i; \mathcal{M}^{i+1}) - \mathcal{Q}(\mathcal{M}^i; \mathcal{M}^i) >$ threshold.

Figure 2.6: The EM algorithm.

The E-step requires the calculation of the joint probability $\mathrm{p}(\boldsymbol{S}, \boldsymbol{\theta}, \boldsymbol{m}|\mathcal{W}_\mathsf{r}; \mathcal{M})$ for possible state sequences and components. The forward-backward algorithm gives an efficient means to compute this using the forward and backward probabilities $\alpha_j(t)$ and $\beta_i(t)$ as shown in the derivation of equation (2.22).

Two other terms are necessary to estimate the HMM parameters. The first is the probability of transitioning from state ω_i to state ω_j given the complete observation sequence and

model parameters

$$\begin{aligned}\zeta_t^{(ij)} &= \mathrm{P}(\theta_{t-1}=\omega_i,\theta_t=\omega_j|\boldsymbol{S},\mathcal{W}_\mathsf{r};\mathcal{M}) \\ &= \frac{\mathrm{p}(\theta_{t-1}=\omega_i,\theta_t=\omega_j,\boldsymbol{S}|\mathcal{W}_\mathsf{r};\mathcal{M})}{\mathrm{p}(\boldsymbol{S}|\mathcal{W}_\mathsf{r};\mathcal{M})} \\ &= \frac{\alpha_i(t-1)a_{ij}b_j(\boldsymbol{s}_t)\beta_j(t)}{\alpha_J(T)}\end{aligned} \tag{2.29}$$

The second quantity is the probability that component m, of state ω_j, generated the observation at time t given the compete observation sequence

$$\begin{aligned}\gamma_{s,t}^{(jm)} &= \mathrm{P}(\theta_t=\omega_j,m_t=m|\boldsymbol{S},\mathcal{W}_\mathsf{r};\mathcal{M}) \\ &= \frac{\mathrm{p}(\theta_t=\omega_j,m_t=m,\boldsymbol{S}|\mathcal{W}_\mathsf{r};\mathcal{M})}{\mathrm{p}(\boldsymbol{S}|\mathcal{W}_\mathsf{r};\mathcal{M})} \\ &= \frac{\sum_{i=2}^{J-1}\alpha_i(t-1)a_{ij}c^{(jm)}b_{jm}(\boldsymbol{s}_t)\beta_j(t)}{\alpha_J(T)}\end{aligned} \tag{2.30}$$

These two quantities are related in that

$$\sum_{m=1}^{M}\gamma_{s,t}^{(jm)} = \gamma_{s,t}^{(j)} = \sum_{j=1}^{J}\zeta_t^{(ij)} \tag{2.31}$$

With their definition, equation (2.28) may be expressed as

$$\begin{aligned}\mathcal{Q}(\mathcal{M};\hat{\mathcal{M}}) &= \sum_{\boldsymbol{\theta}\in\boldsymbol{\Theta}}\sum_{\boldsymbol{m}\in\boldsymbol{M}}\frac{\mathrm{p}(\boldsymbol{S},\boldsymbol{\theta},\boldsymbol{m}|\mathcal{W}_\mathsf{r};\mathcal{M})}{\mathrm{p}(\boldsymbol{S}|\mathcal{W}_\mathsf{r};\mathcal{M})}\left(\log\mathrm{p}(\boldsymbol{S}|\boldsymbol{\theta},\boldsymbol{m},\mathcal{W}_\mathsf{r};\hat{\mathcal{B}})+\log\mathrm{P}(\boldsymbol{\theta}|\mathcal{W}_\mathsf{r};\hat{\mathcal{A}})\right) \\ &= \sum_{t=1}^{T}\left\{\sum_{i=1}^{J}\gamma_{s,t}^{(j)}\log\hat{b}_j(\boldsymbol{s}_t)+\sum_{i=1}^{J}\sum_{j=1}^{J}\zeta_t^{(ij)}\log\hat{a}_{ij}\right\} \\ &= \sum_{t=1}^{T}\left\{\sum_{i=1}^{J}\sum_{m=1}^{M}\gamma_{s,t}^{(jm)}\left[\log\hat{c}^{(jm)}+\log\hat{b}_{jm}(\boldsymbol{s}_t)\right]+\sum_{i=1}^{J}\sum_{j=1}^{J}\zeta_t^{(ij)}\log\hat{a}_{ij}\right\}\end{aligned} \tag{2.32}$$

Equating equation (2.32) to zero and solving gives ML estimates for the transition weights and mean, variance and weights of the acoustic model components

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T}\zeta_t^{(ij)}}{\sum_{t=1}^{T}\sum_{k=1}^{J}\zeta_t^{(ik)}} \tag{2.33}$$

$$\hat{c}^{(jm)} = \frac{\sum_{t=1}^{T}\gamma_{s,t}^{(jm)}}{\sum_{m=1}^{M}\sum_{t=1}^{T}\gamma_{s,t}^{(jm)}} \tag{2.34}$$

$$\hat{\boldsymbol{\mu}}_s^{(jm)} = \frac{\sum_{t=1}^{T}\gamma_{s,t}^{(jm)}\boldsymbol{s}_t}{\sum_{t=1}^{T}\gamma_{s,t}^{(jm)}} \tag{2.35}$$

$$\hat{\boldsymbol{\Sigma}}_{s,\mathsf{full}}^{(jm)} = \frac{\sum_{t=1}^{T}\gamma_{s,t}^{(jm)}\left(\boldsymbol{s}_t-\hat{\boldsymbol{\mu}}_s^{(jm)}\right)\left(\boldsymbol{s}_t-\hat{\boldsymbol{\mu}}_s^{(jm)}\right)^\mathsf{T}}{\sum_{t=1}^{T}\gamma_{s,t}^{(jm)}} \tag{2.36}$$

Derivations for these solutions can be found in Huang et al. [71, 72]. To compute diagonal variances, as discussed on page 2.3.1, the full covariance is diagonalised

$$\hat{\boldsymbol{\Sigma}}_s^{(jm)} = \text{diag}\{\hat{\boldsymbol{\Sigma}}_{s,\text{full}}^{(jm)}\} \tag{2.37}$$

By using a mixture of Gaussians with diagonal covariances, some intra-frame correlations can still be captured, with fewer parameters and without the associated computational cost.

Beyond this section, the state index j for the component parameters will be omitted for simplicity; the summation over all the HMM states and model components within each state will be been reduced to a single summation over all components in the model.

2.3.3 Context Dependent Models and State Clustering

In acoustic modelling, whole-word models may be used for small vocabulary recognition tasks. While effective in limited domains such as isolated word, yes/no or digit recognition, there is difficulty in capturing the co-articulatory effects between words and having sufficient training data to estimate models for every word in a large vocabulary system. Sub-word units such as phones or syllables are alternatives to whole-word models. The acoustic expression of a phone will often be different depending on the neighbouring phones, i.e. the context. This co-articulatory effect motivates the use of context-dependent phone models [7]. Left and right biphone models depend on the phone that precedes or follows. However, the most popular choice is the triphone, where different models exist based on both the immediate left and right contexts. More data are required to train triphone acoustic models compared to biphone due to the increased context. An example of the triphones, using HTK naming convention, that constitute the word 'lexicon' is shown in figure 2.7. This example shows word-internal models. In contrast, cross-word models are used in this work, where the start and end models, i.e. 'l+eh' and 'a-n', are further contextualised by the preceding and subsequent words respectively. Larger contexts may also be used such as quinphones [32, 121], with two phones to the left and two to the right, and even septaphones where this is increased to three phones to the left and right [18].

```
lexicon → l+eh l-eh+k eh-k+s k-s+ih s-ih+k ih-k+a k-a+n a-n
```

Figure 2.7: Word-internal triphone representation of the word "lexicon".

Increasing the size of the phone context exponentially raises the number of models and parameters to train. For instance, with the reduced standard TIMIT [54] phone set of 39, a triphone system would require almost sixty thousand models and a quinphone system nine million models. Even with copious amounts of data, some models may be "unseen" and not represented in the data. To address this, and deal with models with little training data, HMM states with similar output distributions may be tied or clustered together to share training data. Which states are tied together may be determined using data-driven clustering, however unseen contexts cannot be clustered. Alternatively, state clustering using decision trees [7, 111, 154] built from expert phonetic knowledge avoids this issue. An example decision tree is shown in figure 2.8.

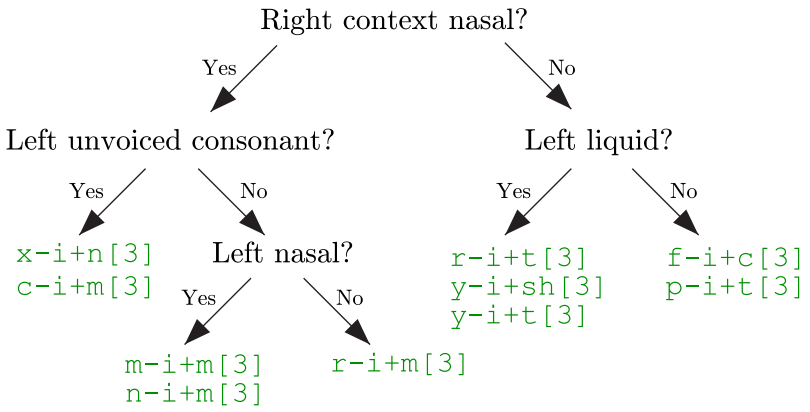


Figure 2.8: Decision tree for triphone state clustering. Example triphone models are shown in green, with their middle state being clustered.

2.3.4 Covariance Modelling

The DCT does not fully decorrelate the static cepstral features [100] and the addition of dynamic coefficients introduces further intra-frame correlations; recognisers that model these correlations achieve greater accuracy [114]. However, using full or block-diagonal covariance matrices greatly increases the number of parameters that need to be estimated [41]; there may be insufficient data to do so. Moreover, there is a significant increase in computational cost to perform a full or block-diagonal Gaussian evaluation compared to a diagonal form. This has motived a form of covariance modelling called semi-tied covariance (STC) matrices [41]. In STC, each component covariance matrix is divided into two parts: a component-specific diagonal variance $\boldsymbol{\Sigma}^{(m)}_{s,\mathsf{diag}}$ and a shared, semi-tied non-diagonal matrix $\boldsymbol{H}$. The term $\boldsymbol{\Sigma}^{(m)}_{s,\mathsf{diag}}$ is equivalent to that given in equation (2.37), but with the state index j omitted for brevity, and the subscript diag added to emphasise the diagonal matrix structure. The covariance matrix may be expressed as

$$\boldsymbol{W}^{(m)}_{s} = \boldsymbol{H}\boldsymbol{\Sigma}^{(m)}_{s,\mathsf{diag}}\boldsymbol{H}^{\mathsf{T}} \tag{2.38}$$

It is easier to deal with the inverse of the matrix, i.e. $\boldsymbol{A}_{\mathsf{stc}} = \boldsymbol{H}^{-1}$. This matrix $\boldsymbol{A}_{\mathsf{stc}}$ is called the semi-tied transform. Here it will be discussed as a global transform, however it may be class dependent. The semi-tied transforms may be estimated iteratively using the following auxiliary function [41]

$$\mathcal{Q}(\mathcal{M};\hat{\mathcal{M}}) = \sum_{m=1}^{M}\sum_{t=1}^{T}\gamma^{(m)}_{s,t}\left\{\log\left(\left(\boldsymbol{a}_{\mathsf{stc},\bar{i}}\boldsymbol{p}^{\mathsf{T}}_{\bar{i}}\right)^2\right) - \log|\boldsymbol{\Sigma}^{(m)}_{s,\mathsf{diag}}| - \sum_{d=1}^{D}\frac{\left(\boldsymbol{a}_{\mathsf{stc},\bar{d}}(\boldsymbol{s}_t - \boldsymbol{\mu}^{(m)}_{s})\right)^2}{\sigma^{(m)2}_{s,d}}\right\} \tag{2.39}$$

The row vector $\boldsymbol{a}_{\mathsf{stc},\bar{i}}$ is the ith row of $\boldsymbol{A}_{\mathsf{stc}}$; the bar over the row index term, i or d, denotes the index gives a row vector for a matrix. The starting acoustic model $\mathcal{M}$, and updated acoustic model $\hat{\mathcal{M}}$, include the HMM parameters and the semi-tied transform. The semi-tied transform may be initialised with an identity matrix. The scalar variance $\sigma^{(m)2}_{s,d}$ is the dth element of the leading diagonal of $\boldsymbol{\Sigma}^{(m)}_{s,\mathsf{diag}}$. The matrix $\boldsymbol{G}_i$ is given by

$$\boldsymbol{G}_i = \sum_{m=1}^{M}\frac{\gamma^{(m)}_{s}}{\sigma^{(m)2}_{s,i}}\boldsymbol{\Sigma}^{(m)}_{s,\mathsf{full}} \tag{2.40}$$

An estimate of the full covariance matrix $\boldsymbol{\Sigma}^{(m)}_{s,\mathsf{full}}$ is given by equation (2.36). The rows of the transformation matrix are given by

$$\boldsymbol{a}_{\mathsf{stc},\bar{i}} = \boldsymbol{p}_{\bar{i}}\boldsymbol{G}_i^{-1}\sqrt{\frac{T}{\boldsymbol{p}_{\bar{i}}\boldsymbol{G}_i^{-1}\boldsymbol{p}_{\bar{i}}^{\mathsf{T}}}} \tag{2.41}$$

and $\boldsymbol{p}_{\bar{i}}$ is the ith co-factor row, $[\chi_{i1} \;\; \chi_{i2} \;\; \cdots \;\; \chi_{iD}]$, of $\boldsymbol{A}_{\mathsf{stc}}$. A co-factor i,j is defined as $\chi_{ij} = (\text{-}1)^{i+j}m_{ij}$ where m_{ij} is the minor i,j defined as the reduced determinant of $\boldsymbol{A}_{\mathsf{stc}}$ computed without row i and column j. The estimation of the transform matrix is iterative and row-by-row. Each row of $\boldsymbol{A}_{\mathsf{stc}}$ is optimised with all other rows fixed; rows are related to each other by the co-factors. Once the semi-tied transform is estimated, the model parameters may be re-estimated. Though equations (2.34) and (2.36) are unchanged, the estimate of the diagonal model variance in equation (2.37) now becomes

$$\hat{\boldsymbol{\Sigma}}^{(m)}_{s,\mathsf{diag}} = \text{diag}\left\{\boldsymbol{A}_{\mathsf{stc}}\hat{\boldsymbol{\Sigma}}^{(m)}_{s,\mathsf{full}}\boldsymbol{A}_{\mathsf{stc}}^{\mathsf{T}}\right\} \tag{2.42}$$

The output probability given in equation (2.23) is modified as follows

$$b(\boldsymbol{s}_t) = \sum_{m=1}^{M} c^{(m)}\mathcal{N}\left(\boldsymbol{A}_{\mathsf{stc}}\boldsymbol{s}_t; \boldsymbol{A}_{\mathsf{stc}}\boldsymbol{\mu}_s^{(m)}, \boldsymbol{\Sigma}_s^{(m)}\right) \tag{2.43}$$

Since the variance remains diagonal, the output probabilities can still be efficiently computed. The main cost is transforming M component means, although once $\boldsymbol{\mu}_s^{(m)}$ is updated the result may be cached.

2.3.5 Discriminative Training

Traditionally, acoustic model parameters are trained to maximise the likelihood of the training data. However, maximising the likelihood of the training data is not closely related to the typical evaluation criteria: error rate. Also during the maximisation step in EM, each set of model parameters is estimated independently from all other models. Furthermore for the ML training to be optimal, it is assumed that HMMs are the optimal representation for speech which is not the case. These issues with ML estimation motivate an alternative form of estimating model parameters called discriminative training.

Discriminative training focuses on estimating model parameters that minimise the error rate. An early form of discriminative training used a maximum mutual information (MMI) [141] criterion. MMI aims to optimise the posterior probability that a model generated a portion of the training utterance—this maximises the mutual information between the training data and the models. While this addresses the independent model parameter estimation issue, other forms of discriminative training may more closely link parameter estimation to recognition errors. Various forms of discriminative training have been described in a unifying minimum Bayes' risk (MBR) training framework. The acoustic model parameters $\mathcal{M}$ are trained to minimise the expected loss during decoding [16]. Thus the following objective function is minimised

$$\mathcal{F}_{\mathsf{mbr}}(\mathcal{M}) = \sum_{u\in\mathcal{U}}\sum_{\mathcal{W}_{\mathsf{h}}^{(u)}} \text{P}(\mathcal{W}_{\mathsf{h}}^{(u)}|\boldsymbol{S},\mathcal{M})\,\mathcal{K}(\mathcal{W}_{\mathsf{h}}^{(u)},\mathcal{W}_{\mathsf{r}}^{(u)}) \tag{2.44}$$

where $\mathrm{P}(\mathcal{W}_{\mathsf{h}}^{(u)}|\boldsymbol{S},\mathcal{M})$ is posterior probability of a hypothesis and $\mathcal{K}(\mathcal{W}_{\mathsf{h}}^{(u)},\mathcal{W}_{\mathsf{r}}^{(u)})$ is the loss function. The loss is computed over all training utterances $\mathcal{U}$ and all possible hypothesised word sequences for an utterance u. It is a function between an utterance hypothesis $\mathcal{W}_{\mathsf{h}}^{(u)}$ and the reference $\mathcal{W}_{\mathsf{r}}^{(u)}$. Actually using all competing hypothesis is intractable, so lattices or N-best lists represent a subset of word sequences. In this way, MBR training takes into account competing hypotheses, minimising their posterior probability, during parameter estimation whereas ML training only uses the reference hypothesis.

One form of MBR training, which may be called minimum word error (MWE), uses the Levenshtein string error to compute the loss function making it close to evaluation using word error rate [16, 113] discussed later in section 2.4.3. Minimum classification error (MCE) [75] may be viewed as MBR training where the loss function is zero when the word sequence matches the reference and one otherwise. Hence MCE minimises the string-level error rate by reducing the posterior probability of competing hypotheses and not the reference. MPE instead computes the loss over phone sequences—the loss is a function of the number of phone errors. To improve generalisation acoustic de-weighting, language model simplification during training and various smoothing techniques are often used [45]. Minimum phone error (MPE) was found to consistently give better results than MMI [113]. A modified MCE form that optimises a discriminative "margin" between the correct class and the next best class has given the best reported performance on the TIDigits task so far [155]. On the large vocabulary Wall Street Journal (WSJ) task, MCE gives performance [102] similar to the MPE results in Povey [113], although direct comparisons are difficult to find.

2.4 Speech Recognition

This section discusses some aspects related to using HMMs for speech recognition. Language modelling to provide the language score in equation (2.2) is introduced. Common methods of decoding with HMMs are presented. Lastly, how recognition performance may be evaluated is discussed.

2.4.1 Language Modelling

In addition to the acoustic score, the recogniser requires a language score $\mathrm{P}(\mathcal{W};\dot{\mathcal{M}})$ to provide the posterior probability of a word sequence. It may be assumed that the probability of an individual word depends only on the words that preceded it, i.e. the word history. Thus, the probability of a sequence of N words, $\mathcal{W} = \{w_1,...,w_N\}$, may be written as the product of the conditional probability of each word in the sequence given the word's history

$$\mathrm{P}(\mathcal{W};\dot{\mathcal{M}}) = \mathrm{P}(w_1;\dot{\mathcal{M}})\prod_{i=2}^{N}\mathrm{P}(w_i|\mathcal{W}_1^{i-1};\dot{\mathcal{M}}) \tag{2.45}$$

where $\mathcal{W}_1^{i-1}$ is the partial sequence of words up to word $i-1$.

A popular form of language model (LM) is based on the n-gram. N-gram models are trained on a large amount[1] of data and exhibit good coverage. N-grams assume that the likelihood of a word is only dependent on the $n-1$ words that precede it

$$\mathrm{P}(w_i|\mathcal{W}_1^{i-1};\dot{\mathcal{M}}) \approx \mathrm{P}(w_i|w_{i-1},w_{i-2},\ldots,w_{i-n+1};\dot{\mathcal{M}}) \tag{2.46}$$

[1]For example, Google has publicly released a trillion-word corpus with 5-gram counts in 2006.

N-gram LMs tend to be domain specific and require back-off and smoothing techniques to handle data sparseness [72]. Their complexity may also lead to unacceptable latency when implemented in a dialogue system [34]. Simple systems use unigram (n=1) or bigram (n=2) probabilities, whereas more complex ones will involve trigram (n=3), 4-gram (n=4) or even 5-gram (n=5) models. The IBM TC-STAR LVCSR system [121] directly decodes using a 4-gram LM for European parliamentary speech transcription: a 5.5M n-gram LM is used in the static decoding graph, while a much larger 130M n-gram LM is used to re-score and improve results. An alternative strategy is to use a trigram LM to produce N-best lists re-scored with a 5-gram LM in Mandarin broadcast news transcription [74]. LMs may be compared by computing their perplexity [72] for a text corpus.

2.4.2 Decoding

As discussed at the beginning of this chapter, in section 2.1, decoding refers to the process of searching for the word sequence that may have generated an observation sequence. Often this is approximated by finding the most likely state sequence $\hat{\boldsymbol{\theta}}$, given acoustic and language models $\mathcal{M}$ and $\dot{\mathcal{M}}$. The search is then conducted on all possible state sequences to find the best, which in an ML framework is the maximum likelihood sequence

$$\begin{aligned}\hat{\boldsymbol{\theta}} &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, \mathrm{p}(\boldsymbol{S}, \boldsymbol{\theta}; \mathcal{M}, \dot{\mathcal{M}}) \\ &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, \mathrm{p}(\boldsymbol{S}|\boldsymbol{\theta}; \mathcal{M})\, \mathrm{P}(\boldsymbol{\theta}; \dot{\mathcal{M}})\end{aligned} \tag{2.47}$$

The word sequence is then recovered from the state sequence. To simplify the discussion of decoding, aspects such as multiple pronunciations and model tying are not discussed. Exhaustively searching for the optimal state sequence is $\mathcal{O}\left\{J^T T\right\}$, where J is the number of states and T the number of observation frames—impractical for all but the smallest systems. Fortunately the Viterbi algorithm [72] provides an efficient recursive form to find $bm\theta$. A function based on the probability of the most likely partial state sequence $\boldsymbol{\theta}_1^{t-1}$, ending in state j at time t, is defined as follows, omitting the $\mathcal{M}$ for clarity

$$\begin{aligned}v_j(t) &= \max_{\theta_1\theta_2\cdots\theta_{t-1}} \mathrm{p}(\boldsymbol{s}_1\boldsymbol{s}_2\cdots\boldsymbol{s}_t, \theta_1\theta_2\cdots\theta_t = \omega_j) \\ &= \max_{\boldsymbol{\theta}_1^{t-1}} \mathrm{p}(\boldsymbol{S}_1^t, \boldsymbol{\theta}_1^{t-1}, \theta_t = \omega_j)\end{aligned} \tag{2.48}$$

This function can be recursively computed

$$\begin{aligned}v_j(t) &= \mathrm{p}(\boldsymbol{s}_t|\theta_t = \omega_j) \max_{1<i<J} \mathrm{P}(\theta_t = \omega_j|\theta_{t-1} = \omega_i)\, \mathrm{p}(\boldsymbol{S}_1^{t-1}, \boldsymbol{\theta}_1^{t-2}, \theta_{t-1} = \omega_i) \\ &= b_j(\boldsymbol{s}_t) \max_{1<i<J} a_{ij} v_i(t-1)\end{aligned} \tag{2.49}$$

with starting conditions

$$v_1(1) = 1 \tag{2.50}$$

$$v_j(1) = a_{1j} b_j(\boldsymbol{s}_1), \text{ for } 1 < j < J \tag{2.51}$$

and final condition

$$v_J(T) = \max_{1<i<J} a_{iJ} v_i(T) \tag{2.52}$$

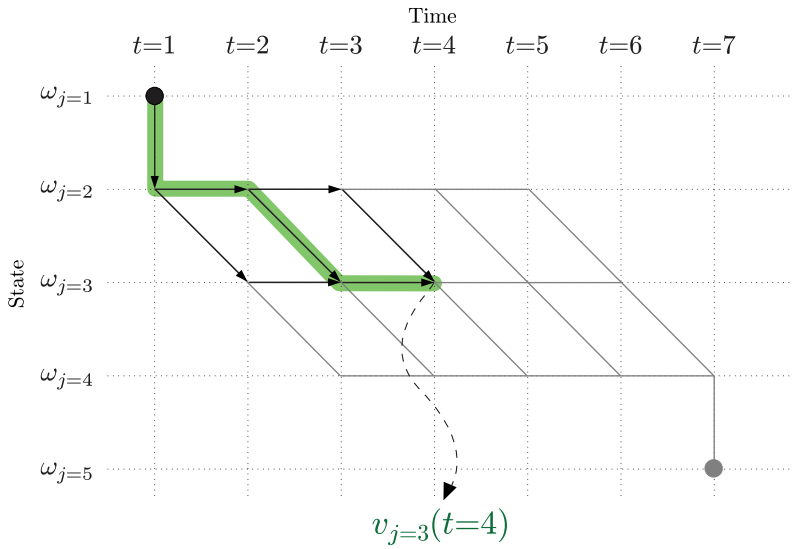


Figure 2.9: Viterbi path highlighted in green, chosen from possible paths for state $\omega_{j=3}$ and time $t = 4$. $v_j(t)$ gives likelihood of this path.

Hence the likelihood of the observation sequence may be approximated as

$$\mathrm{p}(\boldsymbol{S}|\mathcal{W}) \approx v_J(T) \tag{2.53}$$

The Viterbi algorithm is time-synchronous, progressing from left to right, and has complexity $\mathcal{O}\{J^2T\}$ [72]. Unlike the forward probability $\alpha_j(t)$, which gives the likelihood for all paths to ω_j at time t, $v_j(t)$ gives the likelihood for only the maximum. Figure 2.9 can be compared with figure 2.5, for state $\omega_{j=3}$ and time $t = 4$, to contrast the difference between the Viterbi likelihood and the forward probability.

The Viterbi algorithm discussed so far has been in the context of isolated word recognition. However, in LVCSR there is usually insufficient data to robustly estimate an HMM for every word. Hence, sub-word units such as syllables or phones are used. A spoken phrase may be generated by concatenated words, which are composite HMMs formed by sequences of phone models as dictated by a dictionary as shown in figure 2.10. Now uncovering the most

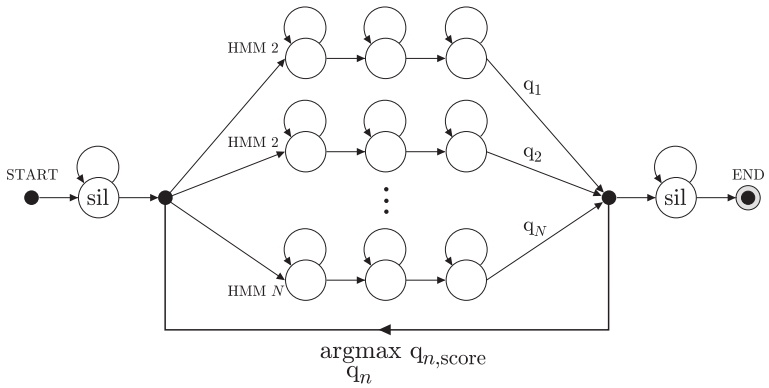


Figure 2.10: Connecting HMMs for continuous speech recognition or when sub-word units are used. An optional silence/pause model may be used between words.

likely hidden state sequence is no longer trivial. A specific implementation of the Viterbi algorithm, token passing [153], provides a method of recovering the most likely state sequence for an observation sequence when connected HMMs are used. Every model state contains one token. A token holds a score $\mathrm{q}_{n,\mathrm{score}}$, which is the likelihood of the partial path up to the current time frame $v_j(t)$, and frame index $\mathrm{q}_{n,\mathrm{start}}$, which is the time frame when the token first entered the HMM. At each time step, these tokens are propagated simultaneously in parallel for all models and states. As shown in the diagram, the token q_n with the highest score $\mathrm{q}_{n,\mathrm{score}}$, i.e. likelihood, is propagated to the start state of all models. At this point, a model insertion penalty or word insertion penalty may be added to $\mathrm{q}_{n,\mathrm{score}}$. At the end of the utterance, the highest likelihood token in all of the exit states of all the models is used to begin a traceback of the token history to yield the most likely word sequence. The traceback requires an array which for each time frame stores two values: the identity of the model generating the best token for the time frame, and that token's start frame index. Tracing back through the array gives the sequence of models that most likely produces the observation sequence. The word sequence can be ascertained from the model sequence.

There are some practical considerations when using the approach outlined here. The probabilities may become very small and hence should be stored as log probabilities. Also there is often a large difference in the dynamic range between the acoustic score and the language score. The acoustic score is usually underestimated due to the independence assumptions. Applying a language or grammar scale factor, which is empirically determined, can address this issue [72]. Also, a word insertion penalty is frequently applied to counteract the tendency of short words being inserted [72]. Hence the most likely word sequence is given by

$$\hat{\mathcal{W}} = \underset{\mathcal{W}}{\operatorname{argmax}} \left\{ \log \mathrm{p}(\boldsymbol{S}|\mathcal{W}) + p_{\mathsf{lm}} \log \mathrm{P}(\mathcal{W}) + p_{\mathsf{ip}} \mathrm{length}(\mathcal{W}) \right\} \tag{2.54}$$

where p_{lm} is the language model scale factor, p_{ip} is the word insertion penalty, and $\mathrm{length}(\mathcal{W})$ gives the number of words in $\mathcal{W}$.

Increased complexity of the acoustic models, e.g. adding states or context, and language models, e.g. accounting for word histories, can greatly increase the decoding search space. This increased cost of searching the state space may be alleviated by *pruning* out low likelihood tokens. This is usually implementing by setting a pruning threshold; those tokens whose log probability fall below this *beam-width* from the best token at the time step are deactivated [152]. Decreasing the beam-width improves search speed, by reducing the number of likelihood calculations. This comes at the expense of possibly introducing *search errors* when a token containing the most likely partial path is pruned out before reaching the end of the utterance.

Token passing is a breadth first, time-synchronous search since paths are extended at each time frame in parallel. Since the incorporation of contexts such as phone and word history can greatly increase the network size, such that memory issues may arise, the search space may be dynamically expanded as contexts are encountered during decoding. Nevertheless, creating a static graph of the entire search space for LVCSR with context dependent models and n-gram language models has proven possible [103]. An alternative to token passing is stack decoding. Stack decoding grows a tree of hypotheses word-by-word and expands the most promising paths first making it a best-first search. Since paths may vary in length, it is asynchronous, which can complicate pruning when comparing acoustic scores [6]. Stack decoding does ease the application of language models since the word history is readily available for each path.

As the complexity of acoustic and language models increases, the time required to decode may increase dramatically. One strategy to address this is to perform multiple decoding passes over the test data with successively more complex systems. Following the initial pass, a second adaptation pass over the data with a simple bigram or trigram language model may be used to generate a word lattice to represent a reduced search space of possible hypotheses. This can then be re-scored by a more powerful language model using 4- or 5-gram probabilities to yield the final transcription [121] or lead into a third pass using even more powerful cross–adaptation [53] and system combination methods such as ROVER [33] or CNC [31]. Efficient LVCSR decoding techniques with complex acoustic and language models continues to be an area of active research. A brief overview of decoding techniques is given in Aubert [6].

2.4.3 Evaluation

The recognition accuracy of a speech recognition system may be determined by comparing the hypothesised transcription with a reference transcription. One metric is the sentence or string error rate which is the number of correctly recognised sentences over the total number of sentences. Another common metric, which is predominantly quoted in this work, is the word error rate (WER). This is computed using the Levenshtein string edit distance. The word sequences are compared using a dynamic programming-based string alignment algorithm [72] to determine the number of deletion errors D, substitution errors S, and insertion errors I. The percentage WER is then

$$\%\text{WER} = \frac{D + S + I}{N} \times 100\% \tag{2.55}$$

where N is the number of words in the reference transcription. An alternative measure is sentence-level accuracy, which is the number of exactly transcribed sentences over the total number of sentences [152]. While WER or sentence accuracy are sensible evaluation criteria for transcription tasks, for other ASR systems it may not be an optimal guide to performance. For example, dialogue system evaluations may quote concept accuracy [12] or task completion rate.

2.5 Adaptation and Normalisation

Despite the amount of data used to train the acoustic models and efforts to produce speaker independent systems, there is still degradation when *factors* or conditions in testing are not represented in the training data. For example, these may be new speakers, different accents, unseen microphones, or environmental noise. In this work, parameters associated with the training conditions are typically denoted by a subscript s as in equations (2.34) to (2.36). Those associated with the mismatched, observed, test conditions are denoted by o. To distinguish between observations that match the training conditions, $\boldsymbol{s}_t$, the vector $\boldsymbol{o}_t$, of the same structure and dimensionality, will represent mismatched, test condition observations.

Two main categories for approaches that address this mismatch between training and test conditions are

- **Adaptation**: given some data from the test condition, the acoustic model parameters, mainly $\mathcal{B}$, are updated to better match the condition, and

- **Normalisation**: factors are removed from the training data and test, before model parameter estimation and decoding, such that they conform to a standard measure.

The first updates the output distribution parameters $\mathcal{B}$ while the second normalises the features. These techniques may require a word-level transcription of the data. Normalisation schemes transform features to conform to a target norm, such as zero mean in cepstral mean normalisation. Some conventional approaches that fit these schemes are discussed in more detail in the following sections. The first two may be classified as adaptation techniques, the second two normalisation schemes.

2.5.1 Maximum Likelihood Linear Regression

ML linear regression (MLLR) adaptation [46, 89] estimates an affine transformation of the acoustic model parameters. The transformation maximises the likelihood of the available adaptation data. Since the amount of adaptation data is usually limited compared to the amount of data available for training the acoustic models, it is useful to share the data such that a single transform is estimated from observations associated with many components. MLLR transforms have the following form

$$\boldsymbol{\mu}_o^{(m)} = \boldsymbol{A}^{(r_m)}\boldsymbol{\mu}_s^{(m)} + \boldsymbol{b}^{(r_m)} \tag{2.56}$$

This transforms the component mean $\boldsymbol{\mu}_s^{(m)}$, estimated in training conditions, to an adapted mean $\boldsymbol{\mu}_o^{(m)}$, such that it matches the test adaptation conditions designated by the subscript o. The superscript r_m indicates that the transform applied to acoustic model component m is based on the regression class r that component m belongs to. The total number of classes R is usually small, especially compared to the number of model components M. Components may be clustered together in a regression class tree, an example of which is shown in figure 2.11. If

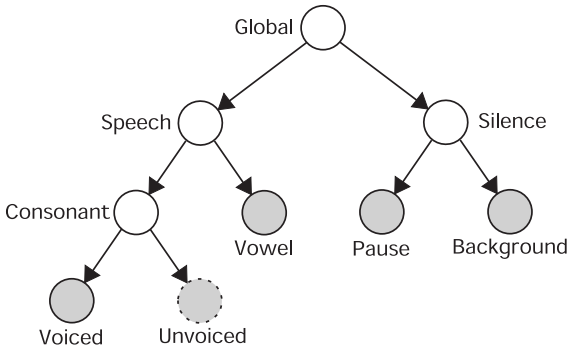


Figure 2.11: Regression class tree for adaptation.

sufficient data are available, then all the base classes, the leaf nodes in the tree, may each have their own transform. However if there are insufficient data to reliably estimate a transform, as indicated by the dotted node, the class may regress to using the transform of its parent. The adaptation data for estimating this transform is aggregated from its children. How much data is sufficient for a transform to be estimated is empirically determined by a "split" threshold; this depends on the complexity of transform, e.g. a diagonal matrix transform requires less data than a full matrix.

Hence, using a regression tree gives an elegant means to scale the number of transforms to the available adaptation data. In the example tree, the unvoiced recognition model components would use the consonant transform, which is trained on the combined data from voiced and unvoiced consonant observations. In practice adaptation schemes use a data-driven clustering approach to generate regression class trees [40, 129]. This may be achieved through k-means clustering and a Kullback-Leibler distance measure [129, 146] or simpler centroid-splitting with a Euclidean distance measure [152].

For a solution to estimate diagonal MLLR transforms see Leggetter and Woodland [89], and for block-diagonal and full see Gales [46]. The latter forms give improved performance since correlations between dimensions are captured, but require more adaptation data to be robustly estimated. The estimation of MLLR transforms also requires a transcription of the adaptation data. If the transcription is known, eg in speaker enrollment scenarios, then the adaptation is *supervised*; otherwise, in *unsupervised* training an initial recognition pass over the data gives a hypothesised transcription. In Sankar et al. [123] it was found that adaptation data transcription word error rates of 20% had a minimal effect on the MLLR transforms estimated in comparison to transforms estimated in a supervised fashion. Hence MLLR transforms can be estimated in an unsupervised fashion if the word error rate of the hypothesised transcription are around 20% or less.

MLLR is often compared to maximum a posteriori (MAP) adaptation [55]. MAP produces an adapted model set that may be considered a weighted combination of well-trained, but mismatched, prior models and ML parameter estimates from limited matched test adaptation data. It was shown that MLLR is more effective with less adaptation data than MAP for speaker adaptation, however with an adequate amount of data MAP outperforms MLLR [72]. This is because MAP has greater flexibility to individually update each acoustic model component [72, 152].

An ML model variance transformation [49] of this form may be estimated

$$\boldsymbol{\Sigma}_o^{(m)} = \boldsymbol{B}^{(m)-\mathsf{T}} \boldsymbol{H}^{(r_m)} \boldsymbol{B}^{(m)-1} \tag{2.57}$$

where $\boldsymbol{B}^{(m)}$ is the Choleski factor of the inverse of the unadapted model variance $\boldsymbol{\Sigma}_s^{(m)-1}$ so

$$\boldsymbol{\Sigma}_s^{(m)-1} = \boldsymbol{B}^{(m)} \boldsymbol{B}^{(m)\mathsf{T}} \tag{2.58}$$

The MLLR variance transformation $\boldsymbol{H}^{(r_m)}$ adapts diagonal model variances into full covariances if $\boldsymbol{H}^{(m)}$ is full, which can make likelihood calculations expensive.

2.5.2 Constrained Maximum Likelihood Linear Regression

A different form of affine transformation is possible by constraining the transformation of the variances to be the same as the mean transform

$$\boldsymbol{\mu}_o^{(m)} = \boldsymbol{H}^{(r_m)} \boldsymbol{\mu}_s^{(m)} - \boldsymbol{g}^{(r_m)} \tag{2.59}$$

$$\boldsymbol{\Sigma}_o^{(m)} = \boldsymbol{H}^{(r_m)} \boldsymbol{\Sigma}_s^{(m)} \boldsymbol{H}^{(r_m)\mathsf{T}} \tag{2.60}$$

This constrained form is called CMLLR [46] or sometimes FMLLR [124]. In contrast to the MLLR variance transform, CMLLR can be efficiently applied in the feature space

$$\tilde{\boldsymbol{o}}_t^{(r_m)} = \boldsymbol{A}^{(r_m)} \boldsymbol{s}_t + \boldsymbol{b}^{(r_m)} \tag{2.61}$$

where $\boldsymbol{A}^{(r_m)} = [\boldsymbol{H}^{(r_m)}]^{-1}$ and $\boldsymbol{b}^{(r_m)} = [\boldsymbol{A}^{(r_m)}]^{-1}\boldsymbol{g}^{(r_m)}$. The term $\boldsymbol{o}_t^{(r_m)}$ indicates that each class of model components has its own transformed feature vector—this equates to having R parallel feature streams being propagated to the decoder. A normalisation term of $\log|\boldsymbol{A}^{(r_m)}|$ is also required during the likelihood calculation [46] which is

$$\mathrm{p}(\boldsymbol{o}_t|\theta_t;\check{\mathcal{M}}) = \sum_{m\in\theta_t} c^{(m)}|\boldsymbol{A}^{(r_m)}|\mathcal{N}\left(\boldsymbol{A}^{(r_m)}\boldsymbol{o}_t + \boldsymbol{b}^{(r_m)};\boldsymbol{\mu}_s^{(m)},\boldsymbol{\Sigma}_s^{(m)}\right) \tag{2.62}$$

Like in MLLR, the matrix $\boldsymbol{A}^{(r_m)}$ need not be full; it may have a block-diagonal or diagonal structure. CMLLR has been used to provide robustness to varying speakers [3] and noise [83].

The CMLLR transforms are estimated by optimising the extended transform matrix $\boldsymbol{W}^{(r_m)} = [\boldsymbol{b}^{(r_m)}\boldsymbol{A}^{(r_m)}]$ in the following auxiliary function

$$\begin{aligned}\mathcal{Q}(\mathcal{M},\mathcal{T};\mathcal{M},\hat{\mathcal{T}}) = K - \frac{1}{2}\sum_{m=1}^{M}\sum_{t=1}^{T}\gamma_{o,t}^{(m)}\big[\log(|\boldsymbol{\Sigma}_s^{(m)}|) - \log(|\boldsymbol{A}^{(r_m)}|^2) + \\ (\boldsymbol{W}^{(r_m)}\boldsymbol{\xi}_t - \boldsymbol{\mu}_s^{(m)})^{\mathsf{T}}\boldsymbol{\Sigma}_s^{(m)-1}(\boldsymbol{W}^{(r_m)}\boldsymbol{\xi}_t - \boldsymbol{\mu}_s^{(m)})\big]\end{aligned} \tag{2.63}$$

where $\boldsymbol{\xi}_t$ is the extended observation vector $\left[\begin{smallmatrix}1\\ \boldsymbol{o}_t\end{smallmatrix}\right]$ and K is a constant term associated with the transition probabilities and Gaussian normalisation terms. The number of test adaptation frames is given by T and $\boldsymbol{o}_t$ denotes a test observation. Note that the component posterior for time t, $\gamma_{o,t}^{(m)}$, is computed from the test data using either a reference or hypothesised transcription. The parameters of the auxiliary function $\mathcal{Q}$ indicate that the acoustic model parameters, $\mathcal{M}$, are constant, but the set of transforms $\mathcal{T}$ is updated to $\hat{\mathcal{T}}$. The ML estimate of the ith row of $\boldsymbol{W}^{(r)}$, given all other rows, is

$$\boldsymbol{w}_{\bar{i}}^{(r)} = \left(\alpha\boldsymbol{p}_{\bar{i}}^{(r)} + \boldsymbol{k}_i^{(r)\mathsf{T}}\right)\boldsymbol{G}_i^{(r)-1} \tag{2.64}$$

where $\bar{i}$ indexes the row, yielding a row vector, rather than a column and a column vector. A co-factor was defined in section 2.3.4, except here it is of $\boldsymbol{A}^{(r_m)}$. A solution for α was given in Gales [46]. The sufficient statistics for regression class r and row i are given by

$$\boldsymbol{G}_i^{(r)} = \sum_{m\in r}\frac{1}{\sigma_{s,i}^{(m)2}}\sum_{t=1}^{T}\gamma_{o,t}^{(m)}\boldsymbol{\xi}_t\boldsymbol{\xi}_t^{\mathsf{T}} \tag{2.65}$$

$$\boldsymbol{k}_i^{(r)} = \sum_{m\in r}\frac{\mu_{s,i}^{(m)}}{\sigma_{s,i}^{(m)2}}\sum_{t=1}^{T}\gamma_{o,t}^{(m)}\boldsymbol{\xi}_t \tag{2.66}$$

where $m{\in}r$ indicates components m in regression class r. Like with STC, equation (2.64) is applied to estimate each row of $\boldsymbol{W}^{(r)}$ while keeping the others fixed. The estimate is guaranteed to improve the likelihood and provides one step in a stable, row-by-row, iterative process [46].

2.5.3 Cepstral Mean and Variance Normalisation

An effective method to address channel mismatch is cepstral mean normalisation (CMN), also known as cepstral mean subtraction. This technique removes the cepstral bias that results from fixed or slowly changing convolutional noise sources

$$\hat{\boldsymbol{x}}_t = \boldsymbol{x}_t - \boldsymbol{\mu}_{\mathsf{cmn}} \tag{2.67}$$

where $\boldsymbol{x}_t$ was defined as a vector of the static elements of the feature vector $\boldsymbol{s}_t$ in equation (2.6). The dynamic coefficients are not affected. Since MFCC is approximately homomorphic, as discussed in section 2.2, subtracting the cepstral bias can remove the effects of linear filtering. The bias may be estimated per utterance, that is $\boldsymbol{\mu}_{\mathsf{cmn}} = \frac{1}{T}\sum_{t=1}^{T}\boldsymbol{s}_t$, or over an entire conversation side, that is a set of utterances from one speaker. Per utterance CMN is effective, but if the length of utterances varies substantially, the cepstral mean estimate may be unreliable due to the changing proportion of speech to silence. Also for real-time use, there is the issue of how to estimate the bias without incurring a large delay in response time. Reports [96, 151] clearly show how well CMN addresses channel mismatch.

A natural extension to CMN is cepstral variance normalisation (CVN) where the target variance of the observations is set to some constant value such as unity

$$\hat{s}_{t,d} = \frac{1}{\sqrt{\sigma^2_{\mathsf{cvn},d}}} s_{t,d} \tag{2.68}$$

where $\sigma^2_{\mathsf{cvn},d}$ is the variance of the dth dimension of the observed data. Note here that the complete feature vector is being normalised. Normalising the distribution of the feature space to zero mean and unit variance is also known as *sphering* the data. As in CMN, the scale factor may be computed per utterance, or per speaker. The cost of applying CMN and CVN is minimal since it is a simple shift and scale of the feature vector. Hence, it is a widely applied technique in ASR systems.

2.5.4 Gaussianisation

More powerful non-linear transforms can be applied to match the histogram of the test data to the training data. These include histogram equalisation [139, 140] or normalisation [104] and Gaussianisation [17, 99]. In the latter, histograms are modelled using GMMs. The combination of the source cumulative density function (CDF) with the inverse Gaussian CDF gives a function that transforms the source histogram to a Gaussian PDF as shown in figure 2.12. The result is a non-linear function applied to the features

$$\hat{s}_{t,d} = \phi^{-1}\left(\int_{-\infty}^{s_{t,d}} \sum_{k=1}^{K} \check{c}_d^{(k)}\, \mathcal{N}\big(s; \check{\mu}_d^{(k)}, \check{\sigma}_d^{(k)2}\big)\mathrm{d}s\right) \tag{2.69}$$

where $\phi^{-1}(\cdot)$ denotes the inverse Gaussian CDF and $\check{\mu}_d^{(k)}$, $\check{\sigma}_d^{(k)2}$ and $\check{c}_d$ the component mean, variance and weight for each mixture component k. Gaussianisation may be applied on a per utterance, per speaker or a global level, with each entity at a level requiring D single-dimensional GMMs, each with K components. Single component GMMs are equivalent to

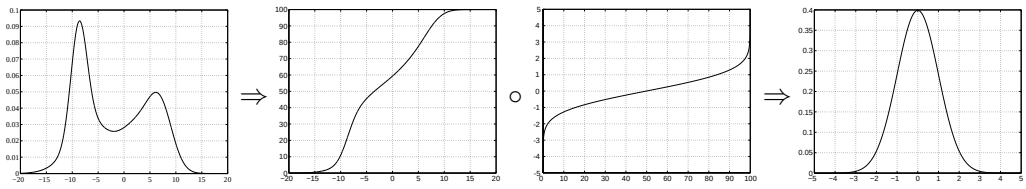


Figure 2.12: Histogram normalisation with Gaussianisation.

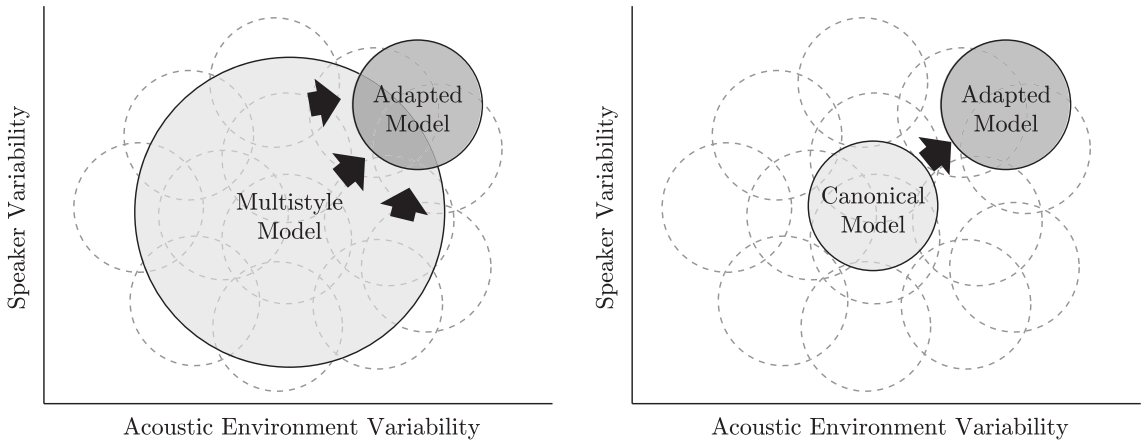


Figure 2.13: Multistyle trained system versus adaptively trained system. Dotted circles represent clusters of homogeneous speaker/environment data.

CMN and CVN. Greater number of components may be viewed as normalising higher-order moments. As in with CMN and CVN, Gaussianisation should be applied to both the training and test features. This scheme provides a compact representation of the histogram compared to some of the other techniques.

2.5.5 Adaptive Training

Adaptive training may be applied to remove unwanted, non-linguistic factors, such as speaker differences or the acoustic environment, from being included in the acoustic models [3, 22, 43, 44]. In multistyle training [98] the acoustic model is forced to represent all these factors; a speaker independent model may be considered a multistyle model. Adaptive training instead uses transforms to model the variation from different factors. There are two main sets of parameters in an adaptive training framework:

- **The Canonical Model**: This captures the "true" acoustic variability of speech. It models the training data given the appropriate factor transform. It is an HMM represented by $\mathcal{M}$.
- **Transforms**: These represent extraneous acoustic variability due to different factors. A different transform is necessary to adapt the canonical model to each specific homogeneous condition. The entire set of transforms is denoted by $\mathcal{T}$.

Adaptive training should produce an canonical acoustic model that is more amenable to being adapted to another speaker or environmental condition compared to clean- or multistyle-trained models [44]. Figure 2.13 provides an illustration why this may be the case. In sub-figure 'a', the multistyle model must capture more variability in the data due to speaker or environmental factors. In sub-figure 'b', during training a set of transforms for each cluster of homogeneous data is estimated. The canonical model parameters are then estimated using these transforms to reduce variability of the speech data due to extraneous factors. Thus adapting the multistyle model for a particular condition may be less precise than with a compact canonical model.

Normalisation techniques like CMN and Gaussianisation may be viewed as adaptive training schemes since factors are being removed from the features before model training. Noise compensation techniques [22, 157] that attempt to remove the noise from the features to give "clean", normalised features for training have been also used in a similar manner. A crucial difference is that while normalisation techniques transform features to some standard, adaptive training in general does not require such a target. Adaptive training schemes have been classified as either model independent, model-dependent feature transformation or model transformation [44]. The key difference between the first scheme and the latter two is whether or not the acoustic models are needed to compute the factor transform. Normalisation techniques typically fall under model-independent schemes, the notable exception being vocal tract length normalisation (VTLN) [88]. VTLN is a speaker normalisation technique that warps the frequency scale to reduce the variability due to differing vocal tract lengths. It is model-dependent because the warping factor is estimated in an ML fashion with the acoustic models [88].

ML forms of adaptive training will necessarily be model-dependent to compute the likelihood of the factor transform—they may be divided into techniques that only update the features, e.g. CMLLR [46] and VTLN, or update the model parameters, e.g. MLLR [3] or cluster adaptive training (CAT) [42]. With the addition of condition dependent transforms, the likelihood of the training data, which was defined in equation (2.12), must be updated for the application of the factor transform

$$\mathrm{p}(\boldsymbol{S}|\mathcal{W}_{\mathsf{r}};\mathcal{M},\mathcal{T}) = \prod_{h=1}^{H} \mathrm{p}(\boldsymbol{S}^{(h)}|\mathcal{W}_{\mathsf{r}}^{(h)};\mathcal{M},\mathcal{T}^{(h)}) \tag{2.70}$$

where h indexes a homogeneous block of training data $\boldsymbol{S}^{(h)}$, e.g. when the speaker and acoustic environment is fixed, which has an associated transcription $\mathcal{W}_{\mathsf{r}}^{(h)}$. H indicates the total number of homogeneous blocks of training data. Thus H sets of transforms are required and the entire set of transforms denoted by $\mathcal{T} = \left\{\mathcal{T}^{(1)},\ldots,\mathcal{T}^{(h)},\ldots,\mathcal{T}^{(H)}\right\}$. Each set of transforms $\mathcal{T}^{(h)}$ may have multiple transforms, one for each regression class. In ML adaptive training, the estimate of the canonical model parameters $\hat{\mathcal{M}}$ maximises equation (2.70)

$$\hat{\mathcal{M}} = \underset{\mathcal{M}}{\operatorname{argmax}}\, \mathrm{p}(\boldsymbol{S}|\mathcal{W}_{\mathsf{r}};\mathcal{M},\mathcal{T}) \tag{2.71}$$

To do so, a set of transforms $\hat{\mathcal{T}}$ must also be estimated for training purposes

$$\hat{\mathcal{T}} = \underset{\mathcal{T}}{\operatorname{argmax}}\, \mathrm{p}(\boldsymbol{S}|\mathcal{W}_{\mathsf{r}};\mathcal{M},\mathcal{T}) \tag{2.72}$$

Due to the difficultly in directly optimising both (2.71) and (2.72), EM is used. The iterative, interleaved training procedure is shown in figure 2.14. The initial parameters may be identity transforms and a well-trained speaker independent, multistyle-trained acoustic model. Convergence occurs when the increase in the auxiliary function, when optimising $\mathcal{M}$ for a fixed $\mathcal{T}$, fails to increase beyond a threshold. While the final acoustic model is used in testing, a new set of transforms must be estimated for the test condition.

Initialise $\mathcal{M}^i$ and $\mathcal{T}^i$, $i = 0$
Do
  E-step: compute $\mathcal{Q}(\mathcal{M}^i, \mathcal{T}^i; \mathcal{M}^i, \mathcal{T}^i)$
  M-step: estimate $\mathcal{T}^{i+1} = \text{argmax}_{\hat{\mathcal{T}}}\, \mathcal{Q}(\mathcal{M}^i, \mathcal{T}^i; \mathcal{M}^i, \hat{\mathcal{T}})$
  E-step: compute $\mathcal{Q}(\mathcal{M}^i, \mathcal{T}^{i+1}; \mathcal{M}^i, \mathcal{T}^{i+1})$
  M-step: estimate $\mathcal{M}^{i+1} = \text{argmax}_{\hat{\mathcal{M}}}\, \mathcal{Q}(\mathcal{M}^i, \mathcal{T}^{i+1}; \hat{\mathcal{M}}, \mathcal{T}^{i+1})$
While $\mathcal{Q}(\mathcal{M}^i, \mathcal{T}^{i+1}; \mathcal{M}^{i+1}, \mathcal{T}^{i+1}) - \mathcal{Q}(\mathcal{M}^i, \mathcal{T}^{i+1}; \mathcal{M}^i, \mathcal{T}^{i+1}) >$ threshold.

Figure 2.14: Adaptive training algorithm.

Adaptive training may be conducted with CMLLR transforms. Transforms are estimated using an auxiliary function similar to equation (2.63)

$$\begin{aligned}\mathcal{Q}(\mathcal{M}, \mathcal{T}; \hat{\mathcal{M}}, \hat{\mathcal{T}}) &= \mathbb{E}_{\mathcal{M},\mathcal{T}}\Big[\log \mathrm{p}(\boldsymbol{S}, \boldsymbol{M}; \hat{\mathcal{M}}, \hat{\mathcal{T}})\Big] \\ &= -\frac{1}{2}\sum_{m=1}^{M}\sum_{h=1}^{H}\sum_{t=1}^{T^{(h)}}\gamma_{s,t}^{(mh)}\big[K^{(m)} + \log(|\boldsymbol{\Sigma}_s^{(m)}|) - \log(|\boldsymbol{A}^{(r_m h)}|^2) + \\ &\qquad (\boldsymbol{W}^{(r_m h)}\boldsymbol{\xi}_t - \boldsymbol{\mu}_s^{(m)})^{\mathsf{T}}\boldsymbol{\Sigma}_s^{(m)^{-1}}(\boldsymbol{W}^{(r_m h)}\boldsymbol{\xi}_t - \boldsymbol{\mu}_s^{(m)})\big]\end{aligned} \tag{2.73}$$

but now the transform set $\mathcal{T}$ is also used to compute $\gamma_{s,t}^{(mh)}$, which is the posterior probability of an observation being generated by component m, homogeneous block h, with transcription $\mathcal{W}_{\mathsf{r}}^{(h)}$ with the set of all possible component/state sequences denoted by $\boldsymbol{M}$. The $K^{(m)}$ terms related to transition and mixture weights. If $\tilde{\boldsymbol{s}}_t^{(r_m h)} = \boldsymbol{W}^{(r_m h)}\boldsymbol{\xi}_t$ then equations (2.35) and (2.37) are now

$$\hat{\boldsymbol{\mu}}_s^{(m)} = \frac{\sum_{h=1}^{H}\sum_{t=1}^{T^{(h)}}\gamma_{s,t}^{(mh)}\tilde{\boldsymbol{s}}_t^{(r_m h)}}{\sum_{h=1}^{H}\sum_{t=1}^{T^{(h)}}\gamma_{s,t}^{(mh)}} \tag{2.74}$$

$$\hat{\boldsymbol{\Sigma}}_s^{(m)} = \text{diag}\left\{\frac{\sum_{h=1}^{H}\sum_{t=1}^{T^{(h)}}\gamma_{s,t}^{(mh)}\big(\tilde{\boldsymbol{s}}_t^{(r_m h)} - \hat{\boldsymbol{\mu}}_s^{(m)}\big)\big(\tilde{\boldsymbol{s}}_t^{(r_m h)} - \hat{\boldsymbol{\mu}}_s^{(m)}\big)^{\mathsf{T}}}{\sum_{h=1}^{H}\sum_{t=1}^{T^{(h)}}\gamma_{s,t}^{(mh)}}\right\} \tag{2.75}$$

The set of transforms $\mathcal{T}^{(h)}$ contains a group of transforms $\mathcal{T}^{(r_m h)}$ for each homogeneous block; the class index r_m associates the transform to the model component as described in section 5.2.4.

While SAT normalises out speaker variability, more than one factor may contribute to non-discriminatory variation in the observed data. Hence, adaptive training may be generalised to account for multiple acoustic factors [43]. An example is CAT combined with CMLLR [156] to give structured transforms. A CAT system specifies speaker dependent models by a set of cluster mean interpolation weights; CMLLR provides environmental adaptation using an affine transform of the feature vector. The CAT-CMLLR system showed better results than a conventional SAT system using only CMLLR on CTS [156].

2.6 Summary

This chapter has described automatic speech recognition using HMMs with continuous output distributions. The various front-end processing steps necessary to generate a standard

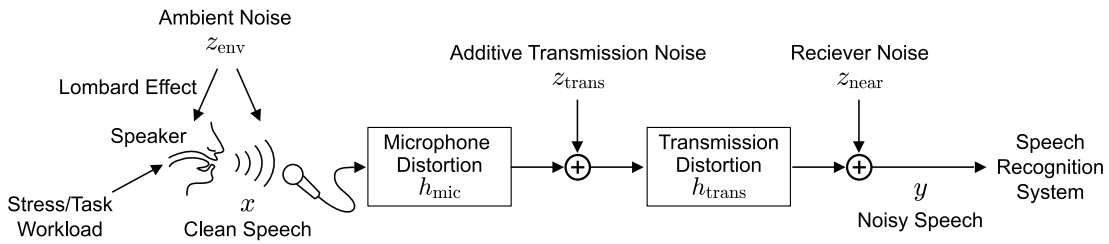
speech parameterisation, MFCC, were given. The use of hidden Markov models for the acoustic modelling of speech was presented. How to estimate the model parameters and performing recognition with such models is discussed. Some conventional LVCSR schemes, such as semi-tied covariances and discriminative training, used to improve acoustic modelling were introduced. Lastly, a few standard adaptation and normalisation techniques to boost performance further are presented. These include MLLR and CMLLR for adaptation, cepstral normalisation and Gaussianisation for normalisation, and adaptive training as a general method of removing extraneous variation during training of the acoustic models.

CHAPTER 3
The Effects of Noise

It is important to understand the difficulties noise presents to current algorithms in order to begin to address the problem of automatic speech recognition in noise. 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.

3.1 Model of the Environment

It is not possible to name and describe all the noises that a speech recogniser could encounter: noise is inherently unpredictable. Fortunately, noise may be approximately characterised by a model of the acoustic environment. The production of the underlying speech signal is influenced by stress, emotion or noise. What is spoken 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 [62] shown in figure 3.1.



$$y(\tau) = \left[\left(\left\{\left[x(\tau)\,\middle|\,\begin{matrix}\text{Task Workload}\\ \text{Stress}\\ \text{Noise}\end{matrix}\right]_{z_{\text{env}}(\tau)} + z_{\text{env}}(\tau)\right\} * h_{\text{mic}}(\tau) + z_{\text{chan}}(\tau)\right) * h_{\text{chan}}(\tau)\right] + z_{\text{near}}(\tau) \quad (3.1)$$

Figure 3.1: Sources of noise and distortion that can effect speech.

This model accounts for changes in speech production due to the task workload, stress or surrounding noise by conditioning $x(\tau)$ on these factors. The last factor, noise, is the cause of the Lombard effect: as the level of noise increases, speakers tend to hyper-articulate, emphasising vowels while consonants become distorted [77]. It is well known that recognition performance degrades significantly for stressed speech, such as Lombard, angry or loud speech compared to neutrally produced speech [15, 76], which recognisers are trained on. Attempts to address these effects have been beneficial [14, 132]; however in this work, these effects on speech production will not be directly addressed.

In the model given in equation (3.1), a major source of corrupted noise is the additive, ambient environmental noise, $z_{\text{env}}(\tau)$, present when the user is speaking. The combined speech and noise signal is then captured and filtered by the microphone impulse response, $h_{\text{mic}}(\tau)$, which can be another large source of distortion. Transmission may also add noise, represented by $z_{\text{trans}}(\tau)$ and $h_{\text{trans}}(\tau)$, although it is expected to be small. The noise at the receiver side $z_{\text{near}}(\tau)$ is also expected to be minimal. Equation (3.1) may be simplified by combining the various additive and convolutional noise sources into single additive noise, $z(\tau)$, and linear channel noise, $h(\tau)$, variables. Doing so gives this standard, oft-used model [1, 39, 106] of the noisy acoustic environment in the time domain show in figure 3.2. The noisy signal is now given by

$$y(\tau) = x(\tau) * h(\tau) + z(\tau) \tag{3.2}$$

where $y(\tau)$ is the noise corrupted speech and $x(\tau)$ the "clean" speech. Note that $z(\tau)$ is a microphone and channel filtered version of the actual ambient noise $z_{\text{env}}(\tau)$ present with the speaker and therefore dependent on $h(\tau)$; still for simplicity, they are assumed independent.

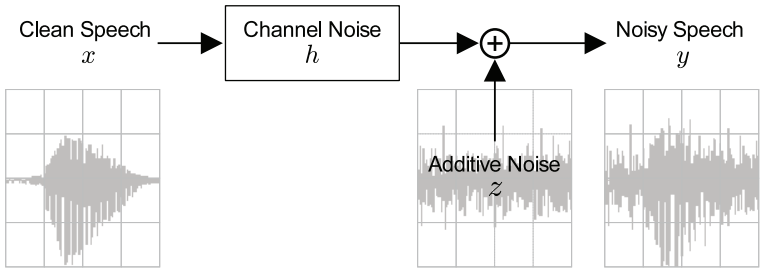


Figure 3.2: Simplified model of the noisy acoustic environment.

Using this model of the noise environment, the front-end processing steps given in section 2.2 may be applied to determine how the noise and speech interact in the cepstral domain. After applying the DFT, the spectrum is warped and smoothed using the mel-spaced filterbank to give a linear-spectral relationship

$$\begin{aligned} y_{f_i,t} &= x_{f_i,t} h_{f_i,t} + z_{f_i,t} \\ &\approx x_{f_i,t} h_{f_i} + z_{f_i,t} \end{aligned} \tag{3.3}$$

where $y_{f_i,t}$, $x_{f_i,t}$, h_{f_i} and $z_{f_i,t}$ are noisy speech, clean speech, channel and additive noise variables expressed in the frequency domain warped by the filterbank. The subscript f_i denotes filterbank bin i and t the signal time frame. It is assumed the channel is time invariant. The spectral phase is usually discarded, hence the power spectrum is obtained

from equation (3.3) as follows

$$\begin{aligned}|y_{f_i,t}|^2 &= y^*_{f_i,t} y_{f_i,t} \\ &= |x_{f_i,t}|^2|h_{f_i}|^2 + |z_{f_i,t}|^2 + 2\cos(\kappa_i)|x_{f_i,t}||h_{f_i}||z_{f_i,t}|\end{aligned} \tag{3.4}$$

where $y^*_{f_i,t}$ is the complex conjugate of $y_{f_i,t}$ and κ_i is the phase difference between the speech and noise, $\angle x_{f_i,t} - \angle n_{f_i,t}$. By assuming there is sufficient smoothing over the filterbank bins, the cross-terms may be ignored; the magnitude spectrum is then approximated by

$$|y_{f_i,t}| \approx |x_{f_i,t}||h_{f_i}| + |z_{f_i,t}| \tag{3.5}$$

Equation (3.5) is the environment model in the linear-spectral domain. To obtain a log-spectral model, the natural logarithm may be applied

$$\begin{aligned}\log|y_{f_i,t}| &\approx \log\big(|x_{f_i,t}||h_{f_i}| + |z_{f_i,t}|\big) \\ &= \log\left(|x_{f_i,t}||h_{f_i}|\left(1 + \frac{|z_{f_i,t}|}{|x_{f_i,t}||h_{f_i}|}\right)\right) \\ &= \log\big(|x_{f_i,t}||h_{f_i}|\big) + \log\left(1 + \exp\left(\log\frac{|z_{f_i,t}|}{|x_{f_i,t}||h_{f_i}|}\right)\right) \\ &= \log|x_{f_i,t}| + \log|h_{f_i}| + \log\big(1 + \exp\big(\log|z_{f_i,t}| - \log|x_{f_i,t}| - \log|h_{f_i}|\big)\big)\end{aligned} \tag{3.6}$$

These log-spectral variables are converted to the cepstral domain as follows

$$\boldsymbol{y}^l_t = \begin{bmatrix}\log|y_{f_1,t}| & \log|y_{f_2,t}| & \cdots & \log|y_{f_N,t}|\end{bmatrix}^\mathsf{T} \qquad \boldsymbol{y}_t = \boldsymbol{C}\boldsymbol{y}^l_t \tag{3.7}$$

$$\boldsymbol{x}^l_t = \begin{bmatrix}\log|x_{f_1,t}| & \log|x_{f_2,t}| & \cdots & \log|x_{f_N,t}|\end{bmatrix}^\mathsf{T} \qquad \boldsymbol{x}_t = \boldsymbol{C}\boldsymbol{x}^l_t \tag{3.8}$$

$$\boldsymbol{h}^l = \begin{bmatrix}\log|h_{f_1}| & \log|h_{f_2}| & \cdots & \log|h_{f_N}|\end{bmatrix}^\mathsf{T} \qquad \boldsymbol{h} = \boldsymbol{C}\boldsymbol{h}^l \tag{3.9}$$

$$\boldsymbol{z}^l_t = \begin{bmatrix}\log|z_{f_1,t}| & \log|z_{f_2,t}| & \cdots & \log|z_{f_N,t}|\end{bmatrix}^\mathsf{T} \qquad \boldsymbol{z}_t = \boldsymbol{C}\boldsymbol{z}^l_t \tag{3.10}$$

where the subscript l indicates a log-spectral variable and N is the number of channels in the filterbank. The terms $\boldsymbol{C}$ and $\boldsymbol{C}^{-1}$ indicate the DCT matrix and its inverse (IDCT). The coefficients of the DCT matrix are given by

$$c_{di} = \sqrt{\frac{2}{N}}\cos\left(\frac{\pi d}{N}(i - 0.5)\right) \tag{3.11}$$

to obtain a $N \times N$ matrix. The IDCT matrix may be obtained by inverting the DCT. As discussed in section 2.2, the number of MFCC may be truncated to 13 or less where N is usually 20 or greater. Hence rows in the IDCT matrix may be truncated such that a smoothed version, $\tilde{\boldsymbol{y}}^l_t$, of the original log-spectral vector, $\boldsymbol{y}^l_t$, may be obtained, i.e. $\tilde{\boldsymbol{y}}^l_t = \boldsymbol{C}^{-1}\boldsymbol{y}_t$.

Finally, equation (3.6) is transformed to the cepstral domain by first considering its vector form and then applying the DCT

$$\begin{aligned}\boldsymbol{y}^l_t &= \boldsymbol{x}^l_t + \boldsymbol{h}^l + \mathbf{log}\big(\mathbf{1} + \mathbf{exp}\big((\boldsymbol{z}^l_t - \boldsymbol{x}^l_t - \boldsymbol{h}^l)\big)\big) \\ \boldsymbol{C}\boldsymbol{y}^l_t &= \boldsymbol{C}\boldsymbol{x}^l_t + \boldsymbol{C}\boldsymbol{h}^l + \boldsymbol{C}\mathbf{log}\big(\mathbf{1} + \mathbf{exp}\big(\boldsymbol{C}^{-1}(\boldsymbol{C}\boldsymbol{z}^l_t - \boldsymbol{C}\boldsymbol{x}^l_t - \boldsymbol{C}\boldsymbol{h}^l)\big)\big) \\ \boldsymbol{y}_t &= \boldsymbol{x}_t + \boldsymbol{h} + \boldsymbol{C}\mathbf{log}\big(\mathbf{1} + \mathbf{exp}\big(\boldsymbol{C}^{-1}(\boldsymbol{z}_t - \boldsymbol{x}_t - \boldsymbol{h})\big)\big)\end{aligned} \tag{3.12}$$

where the bold **log** and **exp** functions indicate element-wise operations that yield a vector of the same dimensionality as the input vector. Equation (3.12) clearly shows that the corrupted speech is a complicated non-linear function of the channel, noise and clean speech.

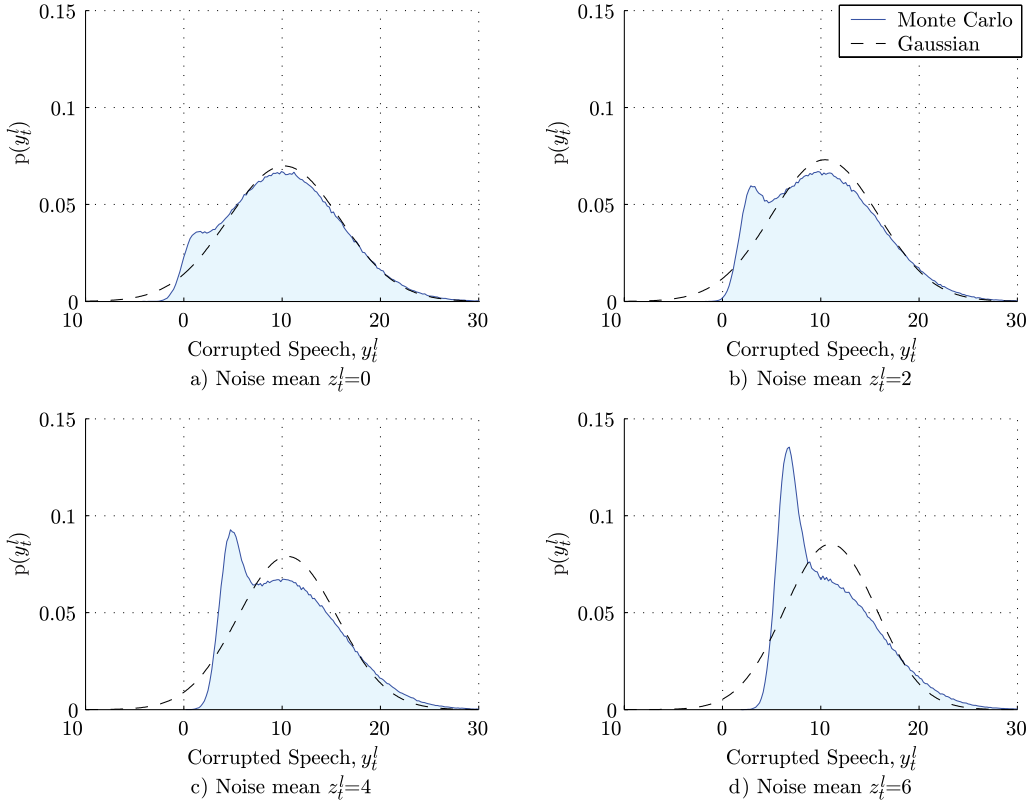


Figure 3.3: Corrupted speech distribution with clean speech of mean 10, variance 5, and ML estimate of Gaussian distribution.

3.2 Effect on Speech Distributions

To more clearly understand the effects of noise, a simulation of its influence on a Gaussian distribution can be conducted. Figure 3.3 shows how a single Gaussian representing the clean speech x_t^l, is affected by additive noise z_t^l at different levels in the log-spectral domain for a single dimension.If noise and clean speech are considered to be Gaussian distributed in the log-spectral domain, the following equation, derived from equation (3.5), can be used to draw random vectors and plot a histogram of the corrupted speech y_t^l

$$y_t^l = \log(\exp(x_t^l) + \exp(z_t^l)) \tag{3.13}$$

The noise is a randomly generated Gaussian variable with the mean rising from 0 to 6 at a constant variance of 1. The clean speech is also a randomly generated Gaussian variable with mean 10 and variance 5. In the log-spectral domain, the magnitude of the variables correspond to the energy level. At first there is a distinct bimodal distribution, but as the noise mean 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 dotted line shows the ML estimate of a Gaussian model of the noisy distribution. It is clear that with the increasing noise, it becomes a poorer fit for the data. As discussed in Gales [39], this may be addressed by increasing the number of components used to model the corrupted speech distribution for each clean speech acoustic model component; this may come with a large computational cost. In practice, most systems use GMMs to model the

state output distributions. Since the effect of noise is non-linear, most components may be either unaffected by the noise or subsumed by it, hence few clean components may actually lead to a bimodal corrupted speech distribution.

The same trend can be seen with actual speech data as shown in figure 3.4. This plots histograms of the 0th cepstral coefficient (C_0) for telephone number utterances in a single vehicle from the Toshiba In-car corpus described in section 9.2. The histograms are normalised to have the same area. It shows how the distribution of C_0 can vary widely. Of the three Car conditions, at idle the SNR[1] is highest, maintaining a roughly bimodal distribution with sharp noise peak at 40 and a spread out speech distribution centred around 60. However, as the SNR decreases to 18 dB, the noise becomes highly intermingled with the speech producing a unimodal distribution. These effects demonstrate how real noise causes changes to speech distributions, producing mismatch problems and reduced ASR performance.

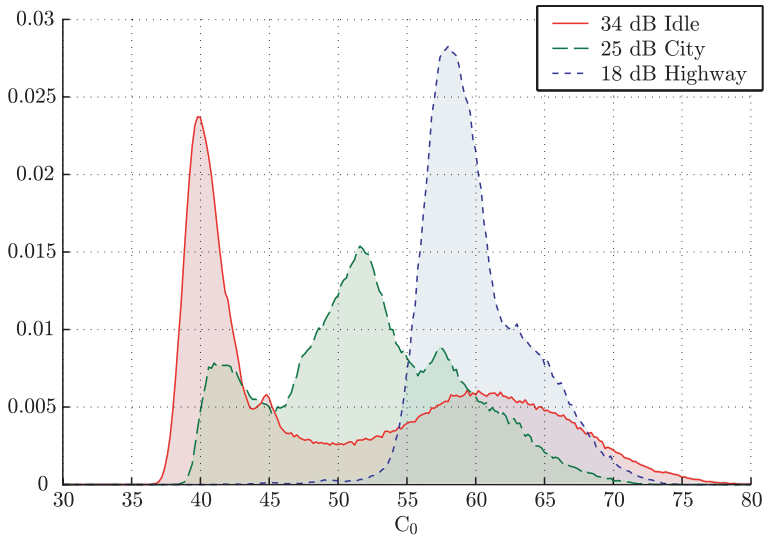


Figure 3.4: Histograms of C_0 for noisy speech recorded in three car conditions: idling and city and highway driving.

3.3 Effect on Intra-frame Correlations

Section 2.3.4 discussed how diagonal model covariances are not optimal for ASR. Recent ASR systems [32, 53, 121, 134] have demonstrated the importance of modelling correlations between dimensions by either using full adaptation transforms, HLDA [87], or STC [41] techniques. Thus it is interesting to examine how environmental noise affects these intra-frame correlations. Figure 3.5 shows contours of equal probability for full bivariate Gaussian distributions modelling C_0 and C_1 for the same Toshiba test conditions from figure 3.4 but with the addition of an Office condition. As in figure 3.3, the variances compress as the SNR decreases—C_1 also follows this trend. Since C_0 correlates strongly with signal energy, there is a greater shift in the mean in this dimension, and more compression in the contour, which causes a slight rotation. For the office data, the major axis of the probability contour is about a 45° rotation from the car data and clearly different. Hence different environmental conditions

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

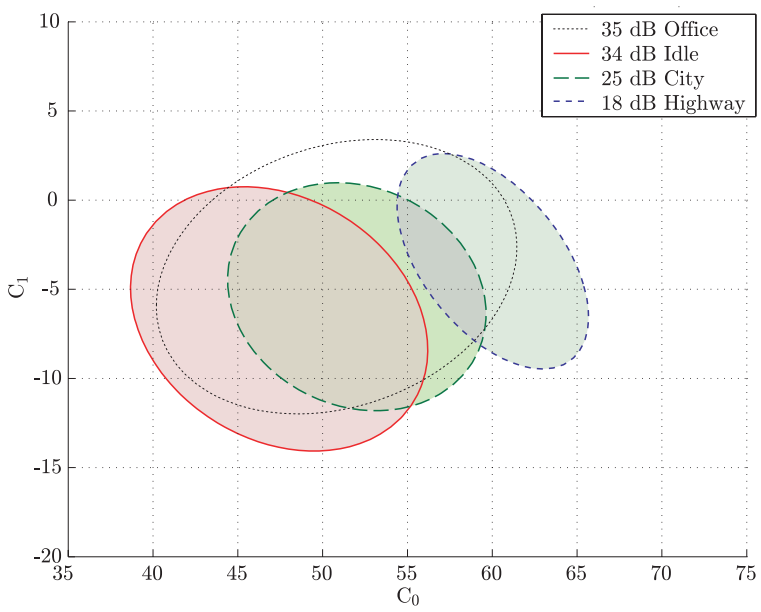


Figure 3.5: Covariance between C_0 and C_1 of noisy speech recorded in an office and three car conditions: idling and city and highway driving.

can give very different dimensional correlations. Systems that model these correlations, for example with HLDA or STC transforms, may exhibit a larger performance degradation than without this additional modelling, due to mismatched correlations between the training data and the noisy test environment. This is obvious in figure 3.6 where the correlations are plotted for Office and Car at highway speed conditions; the correlations are quite distinct for these two conditions. Thus it is also important to consider how noise may change intra-frame correlations when improving the robustness of state-of-the-art ASR systems.

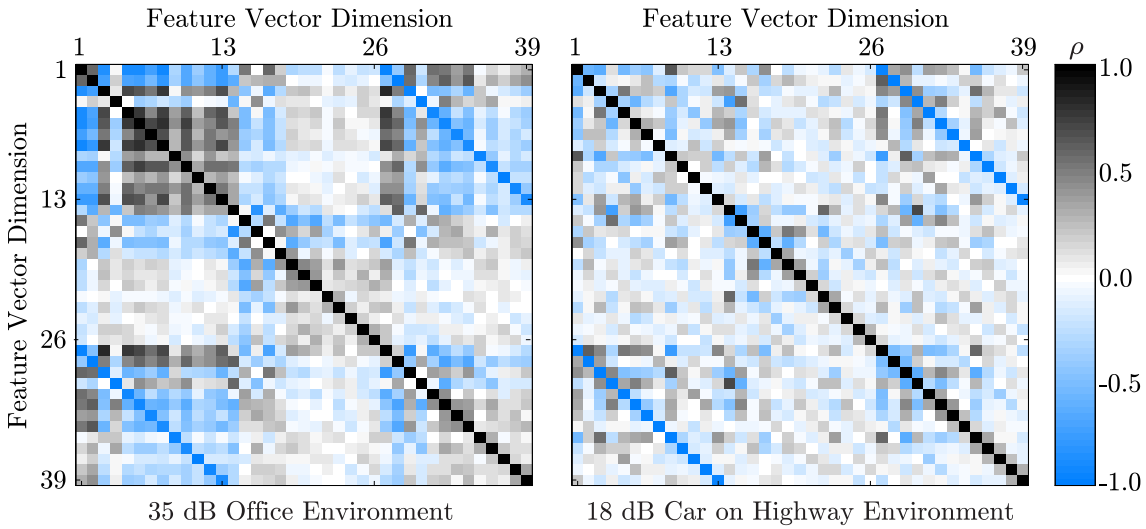


Figure 3.6: Global correlation between dimensions of the full feature vector.

3.4 Summary

This chapter discussed how noise affects speech in the context of ASR. A standard model describing how noisy speech is produced from clean speech corrupted by additive and convolutional noise was presented. The model was discussed for various domains that speech recognisers may operate in including spectral, log-spectral and cepstral. The model was used to simulate how Gaussian distributed noise and speech variables interact. These simulations were then compared with real noisy data. Significant changes occur to speech distributions in the presence of the noise: as the noise level increases, the noisy speech mean increases, variances compress and the correlations between dimensions change. These are all factors that should be kept in mind when considering noise robustness algorithms for ASR.

Chapter 4
Techniques for Noise Robustness

There are many approaches to robustly recognising noise corrupted speech. Ideally, a noise invariant speech parameterisation should be found. This has not proven possible for widely varying levels of noise. Hence this thesis examines techniques that reduce the mismatch between the training and usage conditions. These techniques can be grouped into two distinct approaches as shown in figure 4.1. Front-end noise compensation approaches modify noise corrupted observations to provide an estimate of the feature vector that more closely resembles the clean speech found in training. These estimates can then be decoded using the clean-trained acoustic models. Acoustic model compensation updates the clean-trained acoustic models to a corrupted model set that better matches the noise corrupted observations in the target environment. Many of the techniques in section 2.5 may be used for noise robustness. This chapter discusses techniques that are more specifically targeted for noise robustness.

4.1 A Framework for Noise Robust ASR

The previous chapter described the effects of noise on speech. 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. As in the presentation of adaptation techniques, the mismatched test observation sequence may be denoted $\boldsymbol{O} = \{\boldsymbol{o}_1, \ldots, \boldsymbol{o}_T\}$, where in this chapter the primary cause of mismatch is noise. A single, corrupted speech observation $\boldsymbol{o}_t$ can be thought of as the combination of the outputs from hidden clean speech and noise processes. Thus it may

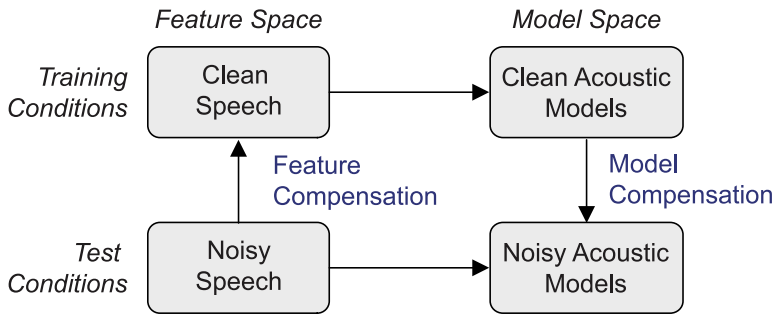


Figure 4.1: Methods of reducing the acoustic mismatch between test and training conditions.

be assumed that observations are conditionally independent given the hidden clean speech $\boldsymbol{s}_t$ and the corrupting noise $\boldsymbol{n}_t$ observations at a time instance. Thus the clean speech $\boldsymbol{S}$ and noise sequence $\boldsymbol{N}$ can be considered independent, each generated by a hidden first-order Markov processes. The likelihood of the observation sequence can then be expressed as

$$\begin{aligned} \mathrm{p}(\boldsymbol{O};\mathcal{M},\mathcal{M}_n) &= \iint_{2\mathcal{R}^{DT}} \mathrm{p}(\boldsymbol{O}|\boldsymbol{S},\boldsymbol{N};\mathcal{M},\mathcal{M}_n)\,\mathrm{p}(\boldsymbol{S};\mathcal{M})\,\mathrm{p}(\boldsymbol{N};\mathcal{M}_n)\mathrm{d}\boldsymbol{S}\mathrm{d}\boldsymbol{N} \\ &\approx \sum_{\boldsymbol{\theta},\boldsymbol{\theta}^n\in\boldsymbol{\Theta}} \mathrm{P}(\boldsymbol{\theta};\mathcal{M})\,\mathrm{P}(\boldsymbol{\theta}^n;\mathcal{M}_n) \prod_{t=1}^{T} \iint_{2\mathcal{R}^{D}} \mathrm{p}(\boldsymbol{o}_t|\boldsymbol{s}_t,\boldsymbol{n}_t)\,\mathrm{p}(\boldsymbol{s}_t|\theta_t)\,\mathrm{p}(\boldsymbol{n}_t|\theta_t^n)\mathrm{d}\boldsymbol{s}_t\mathrm{d}\boldsymbol{n}_t \end{aligned} \quad (4.1)$$

where θ_t and θ_t^n denote the hidden state of the clean speech and noise respectively at time t, $\boldsymbol{\Theta}$ the set of all possible sequences of length T through the state space, $\mathcal{M}$ the clean speech model and $\mathcal{M}_n$ the noise model, and D is the dimensionality of the feature space including both static and dynamic coefficients. The first double integration takes place over the entire D-dimensional Euclidean space and T time, whereas the second just over the $\mathcal{R}^D$ space. The clean speech output probability $\mathrm{p}(\boldsymbol{s}_t|\theta_t)$ is conditioned on the state θ_t in the model set $\mathcal{M}$, hence $\mathcal{M}$ is not explicitly stated; this also applies for $\mathrm{p}(\boldsymbol{n}_t|\theta_t^n)$ and $\mathcal{M}_n$. The conditional independence assumptions are compactly specified in the dynamic Bayesian network shown in figure 4.2. This extends the DBN for clean speech, shown previously in figure 2.3, with a second, parallel, first-order Markov chain representing the noise process.

Hence this framework is an extension of the typical application of HMMs for speech recognition. However, assumptions that are tolerable with clean speech, such as the conditional independence of observations, and the lack of explicit duration modelling may result in increased fragility to noise. Hermansky [66] contends that the fragility of ASR in realistic situations is due to excessive attention to spectral structure and poor modelling of the temporal structure of speech signals. 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 of the noise is not true as demonstrated by the Lombard effect; however, it may be assumed for simplicity. Speech signal production has strong constraints that could be exploited for more robust recognition that are not exploited by the first-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 [25] for speech enhancement.

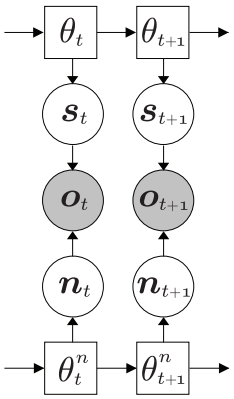


Figure 4.2: Dynamic Bayesian network for noise robust speech recognition. Arrows indicate dependencies, observed variables are shaded, and hidden variables unshaded. Circles represent continuous variables, squares discrete.

Despite the drawbacks in using HMMs for noise robust speech recognition, there have been successful applications. In general, to find the most likely combined state sequence of noise and speech states, for a DBN as shown in figure 4.2, requires a 3-dimensional Viterbi search [143]. The additional computational cost may be avoided if certain assumptions are made. This chapter will discuss various noise robustness techniques and such assumptions within this framework for noise robust speech recognition.

4.2 Inherently Robust Front-ends

A straightforward response to the problem of environmental noise is to build a system that is immune to it. The move from using log-spectral features to MFCC could be considered as shifting towards a more robust parameterisation compared to filterbank parameters. However it is widely known that MFCC and PLP parameters on their own are not immune to noise. In this framework for noise robust speech recognition, an inherently robust front-end would remove the dependency of the observations from the noise process and allow decoding with the noisy observations directly. The acoustic score of $\boldsymbol{O}$ for a word sequence $\mathcal{W}$ may be given by

$$\mathrm{p}(\boldsymbol{O}|\mathcal{W};\mathcal{M}) \approx \sum_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \mathrm{P}(\boldsymbol{\theta};\mathcal{M}) \prod_{t=1}^{T} \mathrm{p}(\boldsymbol{o}_t|\theta_t) \tag{4.2}$$

where $\boldsymbol{\theta}$ is the best state sequences for $\mathcal{W}$. Many other front-end parameterisations have been proposed for their robustness against noise including PMVDR [150] and synchrony-based processing [78] although they give limited gains and can degrade performance in clean environments. Moreover, they can complicate the speech-noise mismatch function, i.e. equation (3.12) for MFCC.

RASTA-PLP [67, 72] is a popularly cited robustness technique. Relative spectral (RASTA) processing is usually applied to PLP coefficients, although not a requirement, to give RASTA-PLP coefficients. The band-pass filtering in RASTA is motivated by observing 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 variable linear-log function applied in the spectral domain gives rise to J-RASTA. To what degree the function is logarithmic or linear depends on the J parameter. It effectively controls between removing convolutional cepstral bias or additive spectral noise. J-RASTA can address modest levels of additive and convolutional noise [82].

4.3 Feature-based Noise Compensation

As shown in figure 4.1, one approach to improve ASR robustness to noise is to remove the noise from the incoming observations $\boldsymbol{O}$. This "cleaning" results in features that better match the original clean speech acoustic model was trained on

$$\hat{\boldsymbol{S}} = \mathcal{F}(\boldsymbol{O}, \mathcal{M}, \mathcal{M}_n) \tag{4.3}$$

where $\hat{\boldsymbol{S}} = \{\hat{\boldsymbol{s}}_1, \ldots, \hat{\boldsymbol{s}}_T\}$ denotes the set of estimated clean speech observations generated by function $\mathcal{F}$ computed from the noise-corrupted observations $\boldsymbol{O}$ 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

$$\int_{\mathcal{R}^D} \mathrm{p}(\boldsymbol{o}_t|\boldsymbol{s}_t, \boldsymbol{n}_t)\, \mathrm{p}(\boldsymbol{n}_t|\theta_t^n)\, \mathrm{d}\boldsymbol{n}_t \approx \mathrm{p}\big(\boldsymbol{o}_t|\boldsymbol{s}_t; \check{\mathcal{M}}\big) \approx \delta(\hat{\boldsymbol{s}}_t - \boldsymbol{s}_t) \tag{4.4}$$

The marginalisation on the left may not be conducted, nor actual forms for the distributions in the integrand specified. As the delta function indicates, standard feature enhancement simply computes a mapping from the noisy speech vector to the clean. It may do so using some compensation parameters denoted by $\check{\mathcal{M}}$. Since in feature compensation, the noise process is considered to be the same for all models, the probability of the noise sequence can be ignored $\mathrm{P}(\boldsymbol{\theta}^n; \mathcal{M}_n)$. Thus substituting equation (4.4) into equation (4.1), yields the following expression where the estimate of the clean speech is directly used for decoding

$$\mathrm{p}(\boldsymbol{O}|\mathcal{W}; \mathcal{M}, \mathcal{M}_n) \approx \sum_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathrm{P}(\boldsymbol{\theta}; \mathcal{M}) \prod_{t=1}^{T} \mathrm{p}(\hat{\boldsymbol{s}}_t|\theta_t) \tag{4.5}$$

There are various methods to compute $\hat{\boldsymbol{s}}_t$. These can be broadly classified into those that enhance the spectral domain, and those that compensate the cepstral parameters. Figure 4.3 outlines the standard feature compensation process.

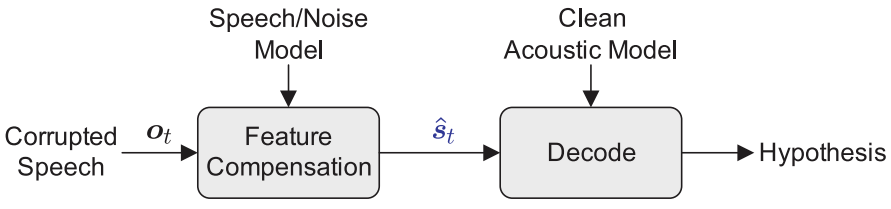


Figure 4.3: The standard feature compensation process.

4.3.1 Speech Enhancement

An early method of addressing additive noise is spectral subtraction (SS) [11]. The noise magnitude spectrum is estimated from frames that 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 assuming the noise is additive and varies slowly in time. A general form for SS is

$$|\hat{x}_{f_i,t}|^\alpha = \max\big(|y_{f_i,t}|^\alpha - \mathcal{E}\{|z_{f_i,t}|^\alpha\}, \epsilon\big) \tag{4.6}$$

where $\mathcal{E}\{|z_{f_i,t}|\}$ is the expected value of the noise spectrum. Power SS results from $\alpha = 2$ and magnitude SS at $\alpha = 1$. These remove the additive noise in the power spectral domain in equation (3.4) or magnitude spectral domain in equation (3.5). This technique is fairly effective although negative spectra that result must be addressed, here with the floor constant ϵ, and a voice activity detector is needed to provide a background noise estimate. Magnitude SS assumes the speech and noise are in phase, which is generally not true. In contrast, power SS assumes the noise and speech are uncorrelated, which should give better results.

The enhancement can also be improved by having a more detailed model of the speech rather than a simple global one. This motivates state-based speech enhancement where improved 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 Wiener 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 Ephraim et al. [30]. 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, by using the state rather than global statistics, for use in the enhancement process. Enhancement with auto-regressive, hidden Markov models of speech is studied in Ephraim [28], Logan and Robinson [101], Seymour and Niranjan [128].

As discussed in [29], speech enhancement can be viewed as minimising the average distortion between an estimator of the clean speech vector $\check{\boldsymbol{s}}_t$ and the hidden, true clean speech vector $\boldsymbol{s}_t$. If the distortion measure is the Euclidean norm $||\cdot||$ then this leads to the following MMSE estimate of the clean speech

$$\hat{\boldsymbol{s}}_t = \underset{\check{\boldsymbol{s}}_t}{\operatorname{argmin}}\, \mathcal{E}\big\{||\boldsymbol{s}_t - \check{\boldsymbol{s}}_t||^2 \,\big|\, \boldsymbol{O}; \check{\mathcal{M}}\big\} \tag{4.7}$$

where and $\check{\mathcal{M}}$ is the set of front-end parameters used for enhancement. Thus the commonly used MMSE clean speech estimator may be derived as follows

$$\begin{aligned}\hat{\boldsymbol{s}}_t &= \underset{\check{\boldsymbol{s}}_t}{\operatorname{argmin}}\, \mathcal{E}\big\{||\boldsymbol{s}_t - \check{\boldsymbol{s}}_t||^2 \,\big|\, \boldsymbol{O}; \check{\mathcal{M}}\big\} \\ &= \mathcal{E}\big\{\boldsymbol{s}_t|\boldsymbol{O}; \check{\mathcal{M}}\big\} \\ &= \int_{\mathcal{R}^D} \boldsymbol{s}_t\, \mathrm{p}(\boldsymbol{s}_t|\boldsymbol{O}; \check{\mathcal{M}})\mathrm{d}\boldsymbol{s}_t \end{aligned} \tag{4.8}$$

for the complete cepstral domain clean speech feature vector $\boldsymbol{s}_t$. Note that the enhanced vector at time t depends on the entire noisy observation sequence $\boldsymbol{O}$ and thus computing it is not causal. This can be addressed by conditioning the estimate only on the current frame as discussed in the following section.

4.3.2 SPLICE

The SPLICE [22] algorithm is a recent technique that has shown good results on a standard noise robustness testing corpus called Aurora2 [68]. SPLICE stands for stereo piece-wise linear enhancement. SPLICE may be considered a special case of the earlier probabilistic optimum filtering (POF) [109] technique; both are MMSE estimators of the form given in equation (4.8). They make use of stereo data to train compensation parameters. Stereo data refers to parallel corpora where there is one channel of noisy speech and one channel of clean speech, but the same speech is being said in both channels. Such a corpus can be made by artificially adding noise to a clean speech database or for example using a close talk and a far talk microphone to record speech in a noisy environment. SPLICE and POF are also both piece-wise transformations since correction vectors are estimated for different regions of the acoustic space. With POF, the transformation of the current frame may include a rotation of the space and surrounding frames, whereas in SPLICE a simple linear bias between the current clean and noisy frame is estimated for each region. In SPLICE, the MMSE estimate of the clean speech from equation (4.8) makes the approximation that the clean speech at time t is dependent only on the observation at that time frame $\boldsymbol{o}_t$

$$\begin{aligned}\hat{\boldsymbol{s}}_t &= \int_{\mathcal{R}^D} \boldsymbol{s}_t \, \mathrm{p}\big(\boldsymbol{s}_t|\boldsymbol{O};\check{\mathcal{M}}\big)\,\mathrm{d}\boldsymbol{s}_t \\ &\approx \int_{\mathcal{R}^D} \boldsymbol{s}_t \, \mathrm{p}\big(\boldsymbol{s}_t|\boldsymbol{o}_t;\check{\mathcal{M}}\big)\,\mathrm{d}\boldsymbol{s}_t \end{aligned} \tag{4.9}$$

The clean speech posterior may be modelled by a GMM such that

$$\begin{aligned}\hat{\boldsymbol{s}}_t &= \int_{\mathcal{R}^D} \boldsymbol{s}_t \sum_{k=1}^{K} \mathrm{P}(k|\boldsymbol{o}_t)\,\mathrm{p}(\boldsymbol{s}_t|\boldsymbol{o}_t,k)\,\mathrm{d}\boldsymbol{s}_t \\ &= \sum_{k=1}^{K} \mathrm{P}(k|\boldsymbol{o}_t) \int_{\mathcal{R}^D} \boldsymbol{s}_t \,\mathrm{p}(\boldsymbol{s}_t|\boldsymbol{o}_t,k)\,\mathrm{d}\boldsymbol{s}_t \end{aligned} \tag{4.10}$$

where k indexes a component in the front-end model set $\check{\mathcal{M}}$. To derive the component clean speech posterior $\mathrm{p}(\boldsymbol{s}_t|\boldsymbol{o}_t,k)$, the noise-corrupted feature space is modelled by a GMM

$$\mathrm{p}\big(\boldsymbol{o}_t;\check{\mathcal{M}}\big) = \sum_{k=1}^{K} \mathrm{P}(k|\boldsymbol{o}_t)\,\mathcal{N}\Big(\boldsymbol{o}_t;\boldsymbol{\mu}_o^{(k)},\boldsymbol{\Sigma}_o^{(k)}\Big) \tag{4.11}$$

where the posterior of component k is given by

$$\mathrm{P}(k|\boldsymbol{o}_t) = \frac{\check{c}^{(k)}\,\mathrm{p}(\boldsymbol{o}_t|k)}{\sum_{i=1}^{K}\check{c}^{(i)}\,\mathrm{p}(\boldsymbol{o}_t|i)} \tag{4.12}$$

and $\check{c}^{(k)}$ or $\check{c}^{(i)}$ are the prior component weight. For each region of the noise-corrupted feature space represented by component k, a bias is estimated to map the noisy speech to clean speech. Thus the component clean speech posterior has the form

$$\mathrm{p}(\boldsymbol{s}_t|\boldsymbol{o}_t,k) = \mathcal{N}\big(\boldsymbol{s}_t;\boldsymbol{o}_t+\check{\boldsymbol{\mu}}^{(k)},\check{\boldsymbol{\Sigma}}^{(k)}\big) \tag{4.13}$$

The noisy to clean speech mapping, or bias, vectors are estimated using stereo data as follows

$$\check{\boldsymbol{\mu}}^{(k)} = \mathcal{E}\left\{\boldsymbol{s}_t-\boldsymbol{o}_t|k\right\} \tag{4.14}$$

$$\check{\boldsymbol{\Sigma}}^{(k)} = \mathcal{E}\left\{(\boldsymbol{s}_t-\boldsymbol{o}_t)(\boldsymbol{s}_t-\boldsymbol{o}_t)^{\mathsf{T}}|k\right\} - \check{\boldsymbol{\mu}}^{(k)}\check{\boldsymbol{\mu}}^{(k)\mathsf{T}} \tag{4.15}$$

The term $\check{\boldsymbol{\Sigma}}^{(k)}$ can be interpreted as the expected square error for component k. Substituting equation (4.13) into equation (4.10) gives

$$\begin{aligned}\hat{\boldsymbol{s}}_t &= \sum_{k=1}^{K} \mathrm{P}(k|\boldsymbol{o}_t) \int_{\mathcal{R}^D} \boldsymbol{s}_t \mathcal{N}\big(\boldsymbol{s}_t; \boldsymbol{o}_t + \check{\boldsymbol{\mu}}^{(k)}, \check{\boldsymbol{\Sigma}}^{(k)}\big) \\ &= \sum_{k=1}^{K} \mathrm{P}(k|\boldsymbol{o}_t) \left(\boldsymbol{o}_t + \check{\boldsymbol{\mu}}^{(k)}\right) \\ &= \boldsymbol{o}_t + \sum_{k=1}^{K} \mathrm{P}(k|\boldsymbol{o}_t)\, \check{\boldsymbol{\mu}}^{(k)} \end{aligned} \tag{4.16}$$

This form may be called *soft* SPLICE enhancement since the clean speech estimate is updated by a weighted sum of the bias vectors. Alternatively, instead of a soft weighted estimate, the component with the highest posterior probability, denoted by k^*, can be used

$$k^* = \underset{k}{\operatorname{argmax}}\, \mathrm{P}(k|\boldsymbol{o}_t) \tag{4.17}$$

This *hard* selection yields a more efficient version of SPLICE

$$\hat{\boldsymbol{s}}_t = \boldsymbol{o}_t + \check{\boldsymbol{\mu}}^{(k^*)} \tag{4.18}$$

SPLICE is not intrinsically tied to stereo data; with a prior clean speech GMM, a corrupted speech GMM may be estimated using VTS compensation [2, 106] and the biases computed from the two GMM. Limiting the update of the feature vector to only a bias form is efficient, however a MLLR-like affine transform would be more accurate as suggested in Deng et al. [22]. Examples of feature compensation techniques that use affine transformations are the front-end GMM form of CMLLR [93] and MBFE, which is discussed in the next section.

4.3.3 MBFE

An alternative to using a front-end GMM is to have a front-end HMM which is theoretically a better model of the temporal aspects of speech. This requires decoding with a simplified HMM in the front-end to determine the state sequence used for compensation. In the model-based feature enhancement (MBFE) [137] technique, an ergodic HMM is used and only the static features are compensated. However, a version which enhances the complete feature vector may be devised. The MMSE estimate of the clean speech for this form is given by

$$\begin{aligned}\hat{\boldsymbol{s}}_t &= \int_{\mathcal{R}^D} \boldsymbol{s}_t\, \mathrm{p}\big(\boldsymbol{s}_t|\boldsymbol{O}; \check{\mathcal{M}}\big)\, \mathrm{d}\boldsymbol{s}_t \\ &\approx \sum_{k=1}^{K} \gamma_{o,t}^{(k)} \int_{\mathcal{R}^D} \boldsymbol{s}_t\, \mathrm{p}(\boldsymbol{s}_t|\boldsymbol{o}_t, k) \mathrm{d}\boldsymbol{s}_t \\ &= \sum_{k=1}^{K} \gamma_{o,t}^{(k)} \Big\{ \boldsymbol{\mu}_s^{(k)} + \boldsymbol{\Sigma}_{os}^{(k)} \boldsymbol{\Sigma}_o^{(k)\,-1} \big(\boldsymbol{o}_t - \boldsymbol{\mu}_o^{(k)}\big) \Big\} \end{aligned} \tag{4.19}$$

where k now indexes the front-end HMM state in the front-end model set $\check{\mathcal{M}}$. The probability of state k at time t given the noisy observation sequence $\boldsymbol{O}$, i.e. the state posterior $\gamma_{o,t}^{(k)}$, is

calculated using the forward-backward algorithm. It was found that using more detailed phoneme-based models would provide more accurate state statistics, however higher error rates in determining the front-end state negatively impacted overall performance [138] .

A GMM version may be derived where no decoding in the front-end is required. This would be similar to the SPLICE form, except the posterior probability of the clean speech for each region of acoustic space is predicted from the joint distribution. The MMSE estimate from equation (4.10) becomes

$$\begin{aligned}\hat{\boldsymbol{s}}_t &\approx \sum_{k=1}^{K} \mathrm{P}(k|\boldsymbol{o}_t) \int_{\mathcal{R}^D} \boldsymbol{s}_t \, \mathrm{p}(\boldsymbol{s}_t|\boldsymbol{o}_t, k) \, \mathrm{d}\boldsymbol{s}_t \\ &= \sum_{k=1}^{K} \mathrm{P}(k|\boldsymbol{o}_t) \Big\{ \boldsymbol{\mu}_s^{(k)} + \boldsymbol{\Sigma}_{os}^{(k)} \boldsymbol{\Sigma}_o^{(k)\,-1} \big(\boldsymbol{o}_t - \boldsymbol{\mu}_o^{(k)}\big) \Big\} \end{aligned} \tag{4.20}$$

and as in SPLICE, a single "max" component k^* may be selected for each time frame

$$\hat{\boldsymbol{s}}_t \approx \boldsymbol{\mu}_s^{(k^*)} + \boldsymbol{\Sigma}_{os}^{(k^*)} \boldsymbol{\Sigma}_o^{(k^*)\,-1} \big(\boldsymbol{o}_t - \boldsymbol{\mu}_o^{(k^*)}\big) \tag{4.21}$$

The joint distribution may be predicted from a model of the clean speech and noise using a model compensation scheme such as VTS or PMC. This gives an affine transform of the feature vector of the form suggested in Deng et al. [22] which is more powerful than the simple SPLICE bias.

4.4 Acoustic Model Compensation

Rather than updating the features, the acoustic model parameters can be compensated to match the noisy test conditions. This is the other main noise robustness approach illustrated in figure 4.1. An obvious example of updating the models is to re-train them with data from the new environment. This may be referred to as *matched* or multipass training. While matched training usually yields the best results in a variety of papers surveyed [47, 58, 151], it is not very practical since large amounts of noisy training data are required and the noise condition may vary. Artificial methods of corrupting the training data have been explored which also yield good results. Samples of noise, such as those from the NOISEX-92 database [144], can be added to the clean training data to generate noise-corrupted training data. This provides good results for levels of noise down to 6-10dB. However, matched training cannot easily address changing noise conditions. Adding a variety of noise samples to clean training data is known as *multistyle* or multicondition training [22, 98], which has shown to improve noise robustness [68].

Due to the unpredictable nature of noise, it is not possible to account for all noise conditions that may be encountered by including them in the training data. Thus other acoustic model compensation methods that update the model parameters may be categorised as either: *adaptive*, where sufficient corrupted speech data are available to update the acoustic models to match the noisy speech observations; or *predictive*, where a noise model is combined with the clean speech models to provide a corrupted speech acoustic model using some model of the acoustic environment. MAP [55] and MLLR-style transforms can be considered adaptive forms, whilst PMC and VTS are predictive techniques. The noisy speech acoustic models can

be predicted from the clean acoustic models by combining them with a model of the noise $\mathcal{M}_n$ using some function

$$\hat{\mathcal{M}} = \mathcal{G}(\mathcal{M}, \mathcal{M}_n) \tag{4.22}$$

where $\hat{\mathcal{M}}$ is the compensated noisy acoustic model. Hence decoding is performed using unaltered noisy observations

$$\mathrm{p}(\boldsymbol{O}; \hat{\mathcal{M}}) = \sum_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathrm{P}(\boldsymbol{\theta}; \hat{\mathcal{M}}) \prod_{t=1}^{T} \mathrm{p}(\boldsymbol{o}_t | \theta_t) \tag{4.23}$$

where θ_t indicates a state in $\hat{\mathcal{M}}$ and thus the dependency on $\hat{\mathcal{M}}$ omitted for concision. The next few subsections will discussed various methods of deriving $\hat{\mathcal{M}}$. The first is SPR, which is a form of re-training all the model parameters. Adaptive model-based compensation schemes like MLLR and CMLLR have been previously discussed in chapter 2; hence after the subsection on SPR, predictive forms are discussed.

4.4.1 Single-pass Re-training

When re-training acoustic models directly on corrupted speech training data, the state posteriors may be poor due to noise—this will reduce the variation between states and blur the boundaries between distinct regions of speech [39]. Single-pass re-training (SPR) [39] is a method of re-estimating the acoustic models that avoids this issue. If a stereo database is available then the state posteriors can be estimated on clean speech and the distribution parameters on noise-corrupted data. For example the noise compensated model mean may be estimated as follows

$$\boldsymbol{\mu}_o^{(m)} = \frac{\sum_{t=1}^{T} \gamma_{s,t}^{(m)} \boldsymbol{o}_t}{\sum_{t=1}^{T} \gamma_{s,t}^{(m)}} \tag{4.24}$$

where $\gamma_{s,t}^{(m)}$ is the component posterior obtained from *clean* observation data. This represents an ideal form of model compensation since the state posteriors and component weights are estimated from clean data, but the distribution parameters are the ML estimates for noisy data.

With SPR though, the corrupted speech distributions may still be badly modelled since each Gaussian distribution is only shifted and scaled, whereas figure 3.3 clearly shows that corrupting noise may yield a bi-modal distribution. This is a general problem for all model compensation techniques that yield a Gaussian distribution as the compensated distribution for each Gaussian in the uncompensated acoustic model. In reality, a stereo database is not usually available, and SPR is a limited offline compensation technique not suitable for varying acoustic environments. It is unfeasible to have the entire training database online and corrupt it using samples of the current noise to re-train the model parameters. Nevertheless SPR, when possible, is a useful method for evaluating model compensation schemes since it provides a reasonable upper limit baseline.

4.4.2 Parallel Model Combination

Parallel model combination (PMC) combines separate noise and speech models to form a corrupted speech model directly for use in the recognition process. It assumes the component posteriors remain unchanged in noise [51]. Therefore only the model component distributions need updating. In non-iterative forms of PMC, each clean speech model component is combined with the noise model via a mismatch function to yield an updated component. Specific additive, convolutional, additive and convolutional, and bandwidth limited channel mismatch functions can be found in Gales [47]. The log-normal approximation is a popular and efficient choice that assumes the sum of two log-normal distributions is approximately log-normal, however it cannot be applied with delta and delta-delta parameters due to the resulting complexity of the forms [50]. Another approximation is the log-add, which may be used to update the component means of the static dimensions

$$\begin{aligned}\mu_{y,i}^{l(m)} &= \log\big(\exp(\mu_{x,i}^{l(m)}) + \exp(\mu_{z,i}^{l})\big) \\ &= \mu_{x,i}^{l(m)} + \log\big(1 + \exp(\mu_{z,i}^{l} - \mu_{x,i}^{l(m)})\big)\end{aligned} \tag{4.25}$$

where the superscript l indicates the parameter belongs to the log-spectral domain rather than an exponent. This is derived from equation (3.13) by assuming the variances of the variables are small.

As discussed with SPR, the transform of each Gaussian component in the clean model, to reflect the noise, does not give a good model of the overall corrupted speech distribution as seen in figure 3.3. Iterative PMC (IPMC) addresses this issue by representing each component with multiple components, iteratively re-estimating the GMM modelling the corrupted speech, still based on state alignments from the clean speech model [39]. This increases the number of components in the overall system. Alternatively, data-driven iterative PMC [39] 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 [47]. DPMC gave results equivalent to matched systems at levels below 20 dB SNR [52]. However, this iterative estimation is computationally expensive.

4.4.3 Vector Taylor Series Model Compensation

As the discussion of PMC shows, deriving a corrupted speech output distribution, given the clean acoustic model and a noise model, is not straightforward. Directly determining the expected value of equation (3.12) is problematic due to the non-linear effect of noise on cepstral speech features. For convenience it is repeated here without the time subscripts for brevity

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{h} + \boldsymbol{C}\mathbf{log}\big(\mathbf{1} + \mathbf{exp}\big(\boldsymbol{C}^{-1}(\boldsymbol{z} - \boldsymbol{x} - \boldsymbol{h})\big)\big) \tag{4.26}$$

Hence many approximations to this function have been proposed, such as selecting the maximum of either the noise or speech, i.e. noise masking [145] or PMC as discussed in the previous section. Another approach is to linearise it with a truncated vector Taylor series

(VTS) [2, 80, 106] to individually update each model component. The first-order VTS approximation of the static corrupted speech may be expressed as

$$\boldsymbol{y}_{\text{vts}} = \boldsymbol{y}\big|_{\mu_0^{(m)}} + \boldsymbol{J}_x^{(m)}\big(\boldsymbol{x} - \boldsymbol{\mu}_x^{(m)}\big) + \boldsymbol{J}_z^{(m)}\big(\boldsymbol{z} - \boldsymbol{\mu}_z\big) + \boldsymbol{J}_h^{(m)}\big(\boldsymbol{h} - \boldsymbol{\mu}_h\big) \tag{4.27}$$

where $|_{\mu_0^{(m)}}$ indicates evaluation at the Taylor series expansion point of the clean speech component mean $\boldsymbol{\mu}_x^{(m)}$, and the additive noise mean $\boldsymbol{\mu}_z$ and channel noise $\boldsymbol{\mu}_h$. The Jacobian matrices are defined as follows

$$\begin{aligned}\boldsymbol{J}_x^{(m)} &= \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}}\Big|_{\mu_0^{(m)}} = \Big[\boldsymbol{\nabla}_{\boldsymbol{x}}\, y_1\big|_{\mu_0^{(m)}} \cdots \boldsymbol{\nabla}_{\boldsymbol{x}}\, y_i\big|_{\mu_0^{(m)}} \cdots \boldsymbol{\nabla}_{\boldsymbol{x}}\, y_{D_s}\big|_{\mu_0^{(m)}}\Big]^{\mathsf{T}} \\ &= \begin{bmatrix} \frac{\partial y_1}{\partial x_1}\big|_{\mu_0^{(m)}} & \cdots & \frac{\partial y_1}{\partial x_{D_s}}\big|_{\mu_0^{(m)}} \\ \vdots & & \vdots \\ \frac{\partial y_{D_s}}{\partial x_1}\big|_{\mu_0^{(m)}} & \cdots & \frac{\partial y_{D_s}}{\partial x_{D_s}}\big|_{\mu_0^{(m)}} \end{bmatrix} = \boldsymbol{I} - \boldsymbol{C}\boldsymbol{F}\boldsymbol{C}^{-1}\end{aligned} \tag{4.28}$$

$$\boldsymbol{J}_h^{(m)} = \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{h}} = \boldsymbol{J}_x^{(m)} \tag{4.29}$$

$$\boldsymbol{J}_z^{(m)} = \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{z}} = \boldsymbol{C}\boldsymbol{F}\boldsymbol{C}^{-1} \tag{4.30}$$

where D_s is the number of static features and the elements of the diagonal matrix $\boldsymbol{F}$ are

$$\begin{aligned} f_{ii} &= \frac{\exp(\boldsymbol{c}_i^{-1}(\boldsymbol{z} - \boldsymbol{x} - \boldsymbol{h}))}{1 + \exp(\boldsymbol{c}_i^{-1}(\boldsymbol{z} - \boldsymbol{x} - \boldsymbol{h}))}\Big|_{\mu_0^{(m)}} \\ &= \frac{\exp(\boldsymbol{c}_i^{-1}(\boldsymbol{\mu}_z - \boldsymbol{\mu}_x - \boldsymbol{\mu}_h))}{1 + \exp(\boldsymbol{c}_i^{-1}(\boldsymbol{\mu}_z - \boldsymbol{\mu}_x - \boldsymbol{\mu}_h))} \end{aligned} \tag{4.31}$$

The term $\boldsymbol{c}_i$ is a row vector that is the ith row of the DCT matrix $\boldsymbol{C}$. The terms f_{ii} vary from 0 to 1 depending on the ratio of the speech to the noise. If the noise level $\boldsymbol{\mu}_n$ is greater than the speech $\boldsymbol{\mu}_x^{(m)}$ in the log-spectral domain, then $f_{ii} \to 1$ and $\boldsymbol{J}_x^{(m)}$ tends to zero; otherwise if little noise is present, $f_{ii} \to 0$ and $\boldsymbol{J}_x^{(m)}$ tends to identity. The term $\boldsymbol{J}_z^{(m)}$ behaves in the opposite manner to $\boldsymbol{J}_x^{(m)}$.

Taking the expected value of equation (4.27), given a specific component m, is straightforward since it is a linear function of three vectors: the additive noise, the clean speech, and the channel noise. This may be expressed as

$$\begin{aligned}\boldsymbol{\mu}_y^{(m)} = \mathcal{E}\{\boldsymbol{y}|m\} &\approx \mathcal{E}\{\boldsymbol{y}_{\text{vts}}|m\} \\ &= \boldsymbol{y}\big|_{\mu_0^{(m)}} \\ &= \boldsymbol{\mu}_x^{(m)} + \boldsymbol{\mu}_h + \boldsymbol{C}\mathbf{log}(\mathbf{1} + \mathbf{exp}(\boldsymbol{C}^{-1}(\boldsymbol{\mu}_z - \boldsymbol{\mu}_x^{(m)} - \boldsymbol{\mu}_h)))\end{aligned} \tag{4.32}$$

Without considering the channel noise, this is equivalent to the log-add approximation result given in equation (4.25) transformed to the cepstral domain. Unlike using the log-add approximation, VTS model compensation provides a covariance matrix update. The covariance of the linear corrupted speech function is simply the sum of the transformed covariances of the clean speech, additive noise and channel

$$\begin{aligned}\boldsymbol{\Sigma}_{y,\text{full}}^{(m)} = \mathcal{E}\big\{\boldsymbol{y}\boldsymbol{y}^{\mathsf{T}}\big|m\big\} - \boldsymbol{\mu}_y^{(m)}\boldsymbol{\mu}_y^{(m)\mathsf{T}} &\approx \mathcal{E}\big\{\boldsymbol{y}_{\text{vts}}\boldsymbol{y}_{\text{vts}}^{\mathsf{T}}\big|m\big\} - \boldsymbol{\mu}_y^{(m)}\boldsymbol{\mu}_y^{(m)\mathsf{T}} \\ &\approx \boldsymbol{J}_x^{(m)}\boldsymbol{\Sigma}_x^{(m)}\boldsymbol{J}_x^{(m)\mathsf{T}} + \boldsymbol{J}_z^{(m)}\boldsymbol{\Sigma}_z\boldsymbol{J}_z^{(m)\mathsf{T}} + \boldsymbol{J}_h^{(m)}\boldsymbol{\Sigma}_h\boldsymbol{J}_h^{(m)\mathsf{T}}\end{aligned} \tag{4.33}$$

assuming the clean speech, additive noise and channel noise are independent of each other. Equation (A.24) in section A.3 gives the mean and covariance of a linear function of Gaussian variables. The term $\boldsymbol{\Sigma}_z$ denotes the variance of the static additive noise and $\boldsymbol{\Sigma}_h$ the variance of the static channel variance. Since the Jacobian matrices $\boldsymbol{J}_x^{(m)}$, $\boldsymbol{J}_z^{(m)}$ and $\boldsymbol{J}_h^{(m)}$ are full, the corrupted speech covariance matrix will also be full and hence diagonalised for standard decoders. Also, it is often assumed that the channel noise does not vary, that is $\boldsymbol{\Sigma}_h = 0$. Thus the static corrupted speech variance may be given by

$$\boldsymbol{\Sigma}_y^{(m)} \approx \text{diag}\left\{\boldsymbol{J}_x^{(m)}\boldsymbol{\Sigma}_x^{(m)}\boldsymbol{J}_x^{(m)\mathsf{T}} + \boldsymbol{J}_z^{(m)}\boldsymbol{\Sigma}_z\boldsymbol{J}_z^{(m)\mathsf{T}}\right\} \tag{4.34}$$

Similar to PMC, using these update formula assumes that a clean speech Gaussian component corrupted by noise may be approximated by another Gaussian distribution; this is clearly not optimal since it was shown in figure 3.3 that the corrupted speech distribution can be bimodal. Nevertheless, for efficiency this approximation is often maintained.

From equations (4.28) and (4.30) it can be seen that

$$\boldsymbol{J}_x^{(m)} + \boldsymbol{J}_z^{(m)} = \boldsymbol{I} \tag{4.35}$$

Hence it can be observed that the compensated variance in equation (4.34) scales between the clean speech variance in low noise to the additive noise variance in high noise. The shifting correlations between dimensions are captured in figure 3.5. Thus the diagonal approximation of the corrupted speech variance in equation (4.34) may be less appropriate if the noise has correlations between dimensions that differ from the speech and when the SNR becomes low enough that this difference emerges.

Standard acoustic models use simple differences or linear regression to compute delta parameters to model the dynamic features of speech as discussed in section 2.2.1. This complicates the compensation of these features for noisy conditions, for example making it difficult to apply the log-normal approximation to compensate the dynamic covariance matrices. In this thesis, a Continuous-Time approximation [39] is used to derive the compensated dynamic parameters. Full derivations for the dynamic features are given in appendix B.1, with the final compensation formulae summarised here. Assuming that the additive noise is stationary, $\mathcal{E}\{\Delta\boldsymbol{z}\} = 0$, and the convolutional noise is constant, i.e. $\Delta\boldsymbol{h} = 0$, the delta noisy speech mean may be approximated by

$$\begin{aligned}\boldsymbol{\mu}_{\Delta y}^{(m)} &\approx \mathcal{E}\left\{\frac{\partial \boldsymbol{y}_{\text{vts}}}{\partial t}\middle| m\right\} = \mathcal{E}\left\{\frac{\partial \boldsymbol{y}_{\text{vts}}}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{x}}{\partial t} + \frac{\partial \boldsymbol{y}_{\text{vts}}}{\partial \boldsymbol{z}}\frac{\partial \boldsymbol{z}}{\partial t} + \frac{\partial \boldsymbol{y}_{\text{vts}}}{\partial \boldsymbol{h}}\frac{\partial \boldsymbol{h}}{\partial t}\middle| m\right\} \\ &\approx \boldsymbol{J}_x^{(m)}\boldsymbol{\mu}_{\Delta x}^{(m)}\end{aligned} \tag{4.36}$$

Similarly for the dynamic noisy speech variance

$$\begin{aligned}\boldsymbol{\Sigma}_{\Delta y}^{(m)} &\approx \mathcal{E}\left\{\frac{\partial \boldsymbol{y}_{\text{vts}}}{\partial t}\frac{\partial \boldsymbol{y}_{\text{vts}}}{\partial t}^{\mathsf{T}}\middle| m\right\} - \boldsymbol{\mu}_{\Delta y}^{(m)}\boldsymbol{\mu}_{\Delta y}^{(m)\mathsf{T}} \\ &\approx \text{diag}\left\{\boldsymbol{J}_x^{(m)}\boldsymbol{\Sigma}_{\Delta x}^{(m)}\boldsymbol{J}_x^{(m)\mathsf{T}} + \boldsymbol{J}_z^{(m)}\boldsymbol{\Sigma}_{\Delta z}\boldsymbol{J}_z^{(m)\mathsf{T}}\right\}\end{aligned} \tag{4.37}$$

Following the same reasoning, the delta-delta parameters are then

$$\boldsymbol{\mu}_{\Delta^2 y}^{(m)} \approx \boldsymbol{J}_x^{(m)}\boldsymbol{\mu}_{\Delta^2 x}^{(m)} \tag{4.38}$$

$$\boldsymbol{\Sigma}_{\Delta^2 y}^{(m)} \approx \text{diag}\left\{\boldsymbol{J}_x^{(m)}\boldsymbol{\Sigma}_{\Delta^2 x}^{(m)}\boldsymbol{J}_x^{(m)\mathsf{T}} + \boldsymbol{J}_z^{(m)}\boldsymbol{\Sigma}_{\Delta^2 z}\boldsymbol{J}_z^{(m)\mathsf{T}}\right\} \tag{4.39}$$

Derivations for these delta-delta parameters are presented in section B.2. To summarise, the noisy speech mean and variance are approximated by

$$\boldsymbol{\mu}_o^{(m)} = \begin{bmatrix} \boldsymbol{\mu}_y^{(m)} \\ \boldsymbol{\mu}_{\Delta y}^{(m)} \\ \boldsymbol{\mu}_{\Delta^2 y}^{(m)} \end{bmatrix} \approx \begin{bmatrix} \boldsymbol{\mu}_x^{(m)} + \boldsymbol{\mu}_h + \boldsymbol{C}\mathbf{log}(\mathbf{1} + \mathbf{exp}(\boldsymbol{C}^{-1}(\boldsymbol{\mu}_z - \boldsymbol{\mu}_x^{(m)} - \boldsymbol{\mu}_h))) \\ \boldsymbol{J}_x^{(m)} \boldsymbol{\mu}_{\Delta x}^{(m)} \\ \boldsymbol{J}_x^{(m)} \boldsymbol{\mu}_{\Delta^2 x}^{(m)} \end{bmatrix} \tag{4.40}$$

$$\begin{aligned} \boldsymbol{\Sigma}_o^{(m)} &= \begin{bmatrix} \boldsymbol{\Sigma}_y^{(m)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{\Delta y}^{(m)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Sigma}_{\Delta^2 y}^{(m)} \end{bmatrix} \\ &\approx \begin{bmatrix} \boldsymbol{J}_x^{(m)} \boldsymbol{\Sigma}_x^{(m)} \boldsymbol{J}_x^{(m)\mathsf{T}} + \boldsymbol{J}_z^{(m)} \boldsymbol{\Sigma}_z \boldsymbol{J}_z^{(m)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{J}_x^{(m)} \boldsymbol{\Sigma}_{\Delta x}^{(m)} \boldsymbol{J}_x^{(m)\mathsf{T}} + \boldsymbol{J}_z^{(m)} \boldsymbol{\Sigma}_{\Delta z} \boldsymbol{J}_z^{(m)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{J}_x^{(m)} \boldsymbol{\Sigma}_{\Delta^2 x}^{(m)} \boldsymbol{J}_x^{(m)\mathsf{T}} + \boldsymbol{J}_z^{(m)} \boldsymbol{\Sigma}_{\Delta^2 z} \boldsymbol{J}_z^{(m)} \end{bmatrix} \end{aligned} \tag{4.41}$$

where the block-diagonal matrix $\boldsymbol{\Sigma}_o^{(m)}$ is diagonalised. Taking first- and second-order derivatives of the linearised version of the noisy speech vector to compute the expected values of the dynamic features is not optimal. It would be more effective to first take the derivatives of the noisy speech vector, given in equation (3.12), and then linearise them to obtain the expected values.

4.4.4 Algonquin

Algonquin was derived as a MMSE feature enhancement scheme [35] which addresses the phase component in equation (3.4). For this thesis, the model adaptation version of Algonquin is of interest since uncertainty of observations due to noise is discussed. This form of model adaptation directly computes state conditional likelihoods using variational estimation [84]. The posterior distribution of the clean speech, noise, channel and state variables is approximated by a simpler parameterised distribution

$$\mathrm{p}(\boldsymbol{x}_t, \boldsymbol{z}_t, \theta, \theta_z | \boldsymbol{y}_t) \approx \mathrm{q}(\boldsymbol{x}_t, \boldsymbol{z}_t, \theta, \theta_z) \tag{4.42}$$

where θ and θ_z indicate the hidden clean speech and noise states. The variational parameters of $\mathrm{q}_{\boldsymbol{y}}$ are optimised per frame for every model component in an iterative fashion. This approximation to the posterior distribution is used for estimating the clean speech in the enhancement version of Algonquin, and for computing the soft information score in the model adaptation form. The output calculations in the recogniser are then approximated: $\mathrm{p}(\boldsymbol{y}_t|\theta) \approx \mathrm{q}(\theta)/\,\mathrm{p}(\theta)$. While the log-spectral domain results are promising, Algonquin requires the approximation of every state in the acoustic model using variational inference to obtain a Gaussian approximation of each state posterior. This makes it rather computationally expensive.

4.5 Uncertainty-based Schemes

The term "uncertainty" has been loosely applied in a variety of contexts to describe various robustness techniques for ASR. In this work, the concept of uncertainty decoding is distinct from uncertain observation decoding [4, 5] and uncertain model parameters [73]. The so called

"soft-information" paradigm presented in the Algonquin framework [84, 85] is viewed in this work as a model-based compensation approach for the reasons outlined in section 4.4.4. For missing feature theory [19, 119], data imputation with soft data is an observation uncertainty approach. In contrast, data marginalisation can be construed as a limited form of front-end uncertainty decoding, restricted to the spectral domain, and where features are either completely certain or uncertain. These different uncertainty-based techniques are elaborated in the following subsections.

4.5.1 Observation Uncertainty

Feature compensation schemes, such as speech enhancement, provide an estimate of the clean speech to the decoder. This assumes the enhancement is exact and the estimate is the true value. However, it may be reasonable to consider that the de-noising process is not exact and there is some residual uncertainty that may be passed to the decoder. Hence in the observation uncertainty approach[1], instead of using a point estimate of the features as shown in figure 4.3, the clean speech posterior is passed to the decoder as shown in figure 4.4.

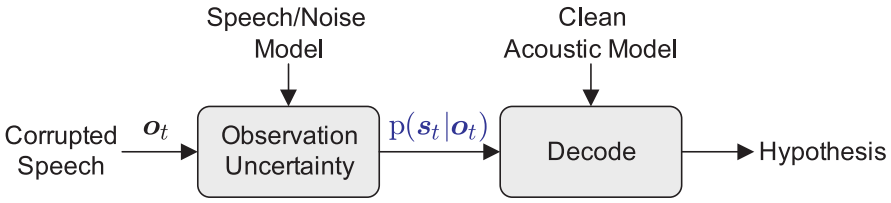


Figure 4.4: Feature compensation with uncertain observations.

Hence, if the clean speech estimate is now considered a multivariate Gaussian distribution $\boldsymbol{s}_t \sim \mathcal{N}(\hat{\boldsymbol{s}}_t, \boldsymbol{\Sigma}_{\hat{s}})$, then the decoding likelihood requires integration over the true clean speech

$$\begin{aligned}\mathrm{p}\big(\boldsymbol{o}_t|m;\check{\mathcal{M}}\big) &\approx \int_{\mathcal{R}^D} \mathrm{p}(\boldsymbol{s}_t|\boldsymbol{o}_t;\check{\mathcal{M}})\,\mathrm{p}(\boldsymbol{s}_t|m)\mathrm{d}\boldsymbol{s}_t \\ &\approx \int_{\mathcal{R}^D} \mathcal{N}\big(\boldsymbol{s}_t;\hat{\boldsymbol{s}}_t,\boldsymbol{\Sigma}_{\hat{s}}\big)\mathcal{N}\big(\boldsymbol{s}_t;\boldsymbol{\mu}_s^{(m)},\boldsymbol{\Sigma}_s^{(m)}\big)\mathrm{d}\boldsymbol{s}_t \\ &= \mathcal{N}\big(\hat{\boldsymbol{s}}_t;\boldsymbol{\mu}_s^{(m)},\boldsymbol{\Sigma}_s^{(m)}+\boldsymbol{\Sigma}_{\hat{s}}\big) \end{aligned} \tag{4.43}$$

where $\hat{\boldsymbol{s}}_t$ is the clean speech estimate, and $\boldsymbol{\Sigma}_{\hat{s}}$ is the expected error of the enhancement process. In SPLICE this is

$$\boldsymbol{\Sigma}_{\hat{s}} = \check{\boldsymbol{\Sigma}}^{(k^*)} \tag{4.44}$$

and defined in equation (4.15). In MBFE, this is

$$\boldsymbol{\Sigma}_{\hat{s}}^{(k)} = \boldsymbol{\Sigma}_s^{(k)} - \boldsymbol{\Sigma}_{so}^{(k)}\boldsymbol{\Sigma}_o^{(k)\text{-}1}\boldsymbol{\Sigma}_{os}^{(k)} \tag{4.45}$$

Other enhancement schemes have been extended to provide this uncertainty, for example computed from the formants [70], a polynomial function of the SNR [4], a parametric model of the clean speech [23, 24], Weiner filtering [10] or a particle filter [147]. Although this approach is widely used, adding the variance of enhancement process has not been well motivated in the

[1] Observation uncertainty has also been called uncertain observation decoding [4], soft data [108] and more confusingly uncertainty decoding [23, 60, 133].

literature. Perhaps this is why the variances need to be scaled before being added in MBFE with uncertainty [137], the variances are considered too large [23], the adding the variance of the delta-delta features did not improve results [23, 24] or there is degradation compared to the non-uncertainty form in high SNR [10].

4.5.2 Uncertainty Decoding

Uncertainty decoding first appeared in the context of SPLICE [26] and Algonquin [84], although as previously discussed, the latter may be viewed as an entirely model-based approach. The integration over the hidden noise variable in equation (4.1) may be performed independently of the clean speech prior

$$\int_{\mathcal{R}^D} \mathrm{p}\big(\boldsymbol{o}_t|\boldsymbol{s}_t,\boldsymbol{n}_t\big)\,\mathrm{p}\big(\boldsymbol{n}_t|\theta_t^n\big)\mathrm{d}\boldsymbol{n}_t \approx \mathrm{p}\big(\boldsymbol{o}_t|\boldsymbol{s}_t;\check{\mathcal{M}}\big) \tag{4.46}$$

The form and parameters of the distribution on the left hand side are not defined since this integration is completely approximated by the corrupted speech conditional distribution on the right-hand side. This latter distribution has parameters $\check{\mathcal{M}}$. To avoid 3-D decoding, it may be assumed that the noise is stationary, obviating terms related to the noise state, and implying $\check{\mathcal{M}}$ only captures a single noise condition. This simplifies equation (4.1) as follows

$$\begin{aligned}\mathrm{p}(\boldsymbol{O};\mathcal{M},\mathcal{M}_n) &\approx \sum_{\boldsymbol{\theta},\boldsymbol{\theta}^n\in\boldsymbol{\Theta}} \mathrm{P}(\boldsymbol{\theta};\mathcal{M})\,\mathrm{P}(\boldsymbol{\theta}^n;\mathcal{M}_n)\prod_{t=1}^{T}\iint_{2\mathcal{R}^D}\mathrm{p}(\boldsymbol{o}_t|\boldsymbol{s}_t,\boldsymbol{n}_t)\,\mathrm{p}(\boldsymbol{s}_t|\theta_t)\,\mathrm{p}(\boldsymbol{n}_t|\theta_t^n)\mathrm{d}\boldsymbol{s}_t\mathrm{d}\boldsymbol{n}_t \\ &\approx \sum_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \mathrm{P}(\boldsymbol{\theta};\mathcal{M})\prod_{t=1}^{T}\int_{\mathcal{R}^D}\mathrm{p}\big(\boldsymbol{o}_t|\boldsymbol{s}_t;\check{\mathcal{M}}\big)\,\mathrm{p}(\boldsymbol{s}_t|\theta_t)\,\mathrm{d}\boldsymbol{s}_t\end{aligned} \tag{4.47}$$

The solution to the integral in equation (4.47) is of course highly dependent on the form of the two parts of the integrand. Ideally, the form of the corrupted speech conditional distribution $\mathrm{p}\big(\boldsymbol{o}_t|\boldsymbol{s}_t;\check{\mathcal{M}}\big)$ should be independent of the acoustic model complexity and make the marginalisation with the clean speech models tractable. If the conditional distribution 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. Hence uncertainty decoding may be viewed as passing the corrupted conditional density function to the decoder as shown in figure 4.5. Examining this distribution in more detail may yield insight into what approximations are appropriate to best model it, with parameters that are efficient to computer, yet minimise the cost of updating the acoustic model.

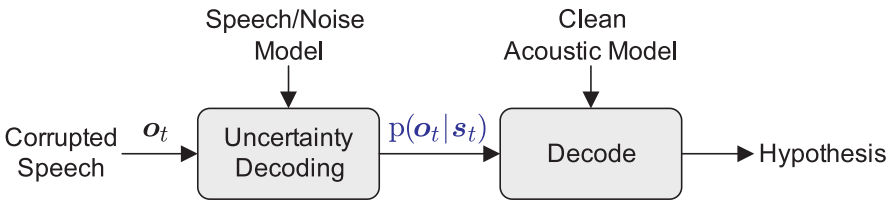


Figure 4.5: Uncertainty decoding.

There are several approaches to modeling the corrupted speech conditional distribution. Using a joint distribution of the clean and corrupted speech to derive it leads to joint uncertainty decoding [93], which is discussed in detail in the next chapter. Approximating it

through an application of Bayes' rule and using the SPLICE form of the clean speech posterior gives the SPLICE with uncertainty form.

4.5.2.1 SPLICE with Uncertainty

SPLICE with uncertainty (SPLICEU) [26] computes the corrupted speech conditional in the front-end, once per frame of speech for efficiency, representing it with a single Gaussian for ease of marginalisation. The conditional corrupted speech distribution can be transformed to the clean speech posterior, through the application of Bayes' rule. This yields the following form of the conditional corrupted speech posterior

$$\mathrm{p}\big(\boldsymbol{o}_t|\boldsymbol{s}_t;\check{\mathcal{M}}\big) = \frac{\sum_{k=1}^{K}\check{c}^{(k)}\,\mathrm{p}(\boldsymbol{s}_t|\boldsymbol{o}_t,k)\,\mathrm{p}(\boldsymbol{o}_t|k)}{\mathrm{p}\big(\boldsymbol{s}_t;\check{\mathcal{M}}\big)} \tag{4.48}$$

where the clean speech posterior and corrupted speech model are Gaussian mixture models of K components, and weighted by $\check{c}^{(k)}$.

Modelling the denominator in equation (4.48) with a GMM makes the marginalisation in equation (4.47) intractable, thus a simple, single Gaussian approximation is used instead. This is a rather crude assumption as a single Gaussian does not represent the clean speech distribution well. Nevertheless

$$\mathrm{p}\big(\boldsymbol{s}_t;\check{\mathcal{M}}\big) \approx \mathcal{N}\big(\boldsymbol{s}_t;\bar{\boldsymbol{\mu}}_s,\bar{\boldsymbol{\Sigma}}_s\big) \tag{4.49}$$

where the parameters are estimated from the corrupted speech GMM, compensated using the SPLICE parameters. The individual components are combined as follows

$$\bar{\boldsymbol{\mu}}_s = \sum_{k=1}^{K}\check{c}^{(k)}\big(\boldsymbol{\mu}_o^{(k)} + \check{\boldsymbol{\mu}}^{(k)}\big) \tag{4.50}$$

$$\bar{\boldsymbol{\Sigma}}_s = \sum_{k=1}^{K}\check{c}^{(k)}\big(\boldsymbol{\mu}_o^{(k)}\boldsymbol{\mu}_o^{(k)\mathsf{T}} + \check{\boldsymbol{\mu}}^{(k)}\check{\boldsymbol{\mu}}^{(k)\mathsf{T}} + \boldsymbol{\Sigma}_o^{(k)} + \check{\boldsymbol{\Sigma}}^{(k)}\big) - \bar{\boldsymbol{\mu}}_s\bar{\boldsymbol{\mu}}_s^{\mathsf{T}} \tag{4.51}$$

The parameters $\check{\mu}_i^{(k)}$, which is the standard SPLICE enhancement bias, and $\check{\sigma}_i^{(k)2}$ are defined in equations (4.14) and (4.15).

Given the SPLICE form of the clean speech posterior, a front-end GMM modelling the corrupted speech and the simplified denominator, and diagonal covariance matrices as in [26], the conditional takes the form

$$\begin{aligned}
\mathrm{p}\big(\boldsymbol{o}_t|\boldsymbol{s}_t;\check{\mathcal{M}}\big) &= \frac{\sum_{k=1}^{K}\check{c}^{(k)}\,\mathrm{p}(\boldsymbol{s}_t|\boldsymbol{o}_t,k)\,\mathrm{p}(\boldsymbol{o}_t|k)}{\mathrm{p}\big(\boldsymbol{s}_t;\check{\mathcal{M}}\big)} \\
&\approx \frac{\sum_{k=1}^{K}\check{c}^{(k)}\mathcal{N}\big(\boldsymbol{s}_t;\boldsymbol{o}_t+\check{\boldsymbol{\mu}}^{(k)},\check{\boldsymbol{\Sigma}}^{(k)}\big)\,\mathrm{p}\big(\boldsymbol{o}_t|k\big)}{\mathcal{N}\big(\boldsymbol{s}_t;\bar{\boldsymbol{\mu}}_s,\bar{\boldsymbol{\Sigma}}_s\big)} \\
&= \sum_{k=1}^{K}\check{c}^{(k)}\,\mathrm{p}(\boldsymbol{o}_t|k)\,|\boldsymbol{A}^{(k)}|\mathcal{N}\Big(\boldsymbol{A}^{(k)}\boldsymbol{o}_t+\boldsymbol{b}^{(k)};\boldsymbol{s}_t,\boldsymbol{\Sigma}_{\mathsf{b}}^{(k)}\Big)
\end{aligned} \tag{4.52}$$

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

$$a_{ii}^{(k)} = \frac{\bar{\sigma}_{s,i}^2}{\bar{\sigma}_{s,i}^2 - \check{\sigma}_i^{(k)2}} \tag{4.53}$$

$$b_i^{(k)} = a_{ii}^{(k)} \left(\check{\mu}_i^{(k)} - \frac{\check{\sigma}_i^{(k)2}}{\bar{\sigma}_{s,i}^{(k)2}} \bar{\mu}_{s,i} \right) \tag{4.54}$$

$$\sigma_{\mathsf{b},i}^{(k)2} = a_{ii}^{(k)} \check{\sigma}_i^{(k)2} \tag{4.55}$$

Since the corrupted speech conditional PDF and the clean speech PDF are both mixtures of Gaussians, the integral from equation (4.47) can be simplified as follows

$$\begin{aligned} \mathrm{p}\big(\boldsymbol{o}_t|\theta_t;\check{\mathcal{M}}\big) &= \int_{\mathcal{R}^D} \mathrm{p}\big(\boldsymbol{o}_t|\boldsymbol{s}_t;\check{\mathcal{M}}\big)\,\mathrm{p}\big(\boldsymbol{s}_t|\theta_t\big)\mathrm{d}\boldsymbol{s}_t \\ &= \sum_{k=1}^{K} \sum_{m\in\theta_t} \check{c}^{(k)} c^{(m)}\,\mathrm{p}\big(\boldsymbol{o}_t|k\big)|\boldsymbol{A}^{(k)}|\mathcal{N}\Big(\boldsymbol{A}^{(k)}\boldsymbol{o}_t+\boldsymbol{b}^{(k)};\boldsymbol{\mu}_s^{(m)},\boldsymbol{\Sigma}_s^{(m)}+\boldsymbol{\Sigma}_{\mathsf{b}}^{(k)}\Big) \end{aligned} \tag{4.56}$$

See appendix A.2 for derivation.

To avoid a negative variance when the variance bias is added to the model variance in equation (4.56), the denominator of the $a_{ii}^{(k)}$ term in equation (4.55) should be forced to remain positive. In [26] this is achieved by constraining the difference of the two denominator terms to be greater than some factor

$$\bar{\sigma}_{s,i}^2 - \check{\sigma}_i^{(k)2} \geq \alpha \bar{\sigma}_{s,i}^2 \tag{4.57}$$

where α is a fraction of the global speech variance $\bar{\sigma}_{s,i}^2$. This floor effectively places a maximum value on $a_{ii}^{(k)}$ where

$$a_{ii}^{(k)} = \min\left(\frac{1}{\alpha} \;,\; \frac{\bar{\sigma}_{s,i}^2}{\bar{\sigma}_{s,i}^2 - \check{\sigma}_i^{(k)2}} \right) \tag{4.58}$$

In the limit, when α is very large, the uncertainty aspect is ignored, returning processing to the standard SPLICE enhancement scheme.

It is clear in equation (4.56) that the overall number of Gaussian evaluations is a product of the number of components in the front-end GMM and the number of components in the acoustic model state. This can be avoided by using the same *hard* approximation from the standard SPLICE form, where only the most probable component k^* is used as decided by equation (4.17). The effective number of components per state in the acoustic model remains unchanged with this approximation, resulting in a more computationally efficient form

$$\mathrm{p}\big(\boldsymbol{o}_t|\theta_t;\check{\mathcal{M}}\big) \propto \sum_{m\in\theta_t} c^{(m)}|\boldsymbol{A}^{(k^*)}|\mathcal{N}\Big(\boldsymbol{A}^{(k^*)}\boldsymbol{o}_t+\boldsymbol{b}^{(k^*)};\boldsymbol{\mu}_s^{(m)},\boldsymbol{\Sigma}_s^{(m)}+\boldsymbol{\Sigma}_{\mathsf{b}}^{(k^*)}\Big) \tag{4.59}$$

The terms $\check{c}^{(k)}$, $\mathrm{p}\big(\boldsymbol{o}_t|k;\check{\mathcal{M}}\big)$ and $|\boldsymbol{A}^{(k)}|$ may be omitted since they are constant for a given frame.

4.5.3 Missing Feature Theory

Missing feature theory (MFT), treats heavily noise-corrupted elements of a spectral domain feature vector as unreliable/missing and those less distorted as reliable/present [19, 119]. This is motivated from studies indicating humans can recognise speech from a "very small proportion of clean frequency channels at any one point in time" [108]. Detecting missing areas of speech is a key aspect of MFT. It is done using a variety of possible measures including SNR-based ones [19, 108], "harmonicity", a combination [8] or a Bayesian classifier using a variety of features [120]. It is conducted at a spectral level because decorrelating transforms such as the DCT spread single unreliable spectral channels to all dimensions in the cepstral space. Once parameters have been labelled as missing or present, the missing ones can be restored [120] or marginalised over [8, 19]. Thus missing feature techniques fall under two approaches: imputation and marginalisation. Both identify regions of the noisy spectral feature vector $\boldsymbol{y}_t^l$ that are missing where the superscript l indicates a log-spectral domain variable[1]. For example

$$\boldsymbol{y}_t^l = \begin{bmatrix} \boldsymbol{y}_{\mathsf{p},t}^l \\ \boldsymbol{y}_{\mathsf{m},t}^l \end{bmatrix} \tag{4.60}$$

where $\boldsymbol{y}_{\mathsf{p},t}^l$ are the components that are considered present, and $\boldsymbol{y}_{\mathsf{m},t}^l$ the missing values. The total number of elements on both the left and right side of the equation are the same.

The two approaches differ in how to handle missing areas. Imputation replaces missing values with estimated values. The reconstructed feature vector is then used as if it was a clean speech vector, and is thus similar to enhancement schemes. Marginalisation classifies solely on $\boldsymbol{y}_{\mathsf{p},t}^l$ by marginalising out the missing components

$$\mathrm{p}(\boldsymbol{Y}^l; \mathcal{M}, \mathcal{M}_n) \approx \sum_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathrm{P}(\boldsymbol{\theta}; \mathcal{M}) \prod_{t=1}^{T} \mathrm{p}(\boldsymbol{y}_{\mathsf{p},t}^l | \theta_t) \int_{\mathcal{R}^{D_\mathsf{m}}} \mathrm{p}(\boldsymbol{y}_{\mathsf{m},t}^l | \theta_t) \mathrm{d}\boldsymbol{y}_{\mathsf{m},t}^l \tag{4.61}$$

where the integral becomes unity and D_m is the number of missing feature elements. This is a rather poor approximation since the feature elements are highly correlated, especially in the spectral domain. Nevertheless, decoding proceeds only with the present features

$$\mathrm{p}(\boldsymbol{Y}^l; \mathcal{M}, \mathcal{M}_n) \approx \sum_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathrm{P}(\boldsymbol{\theta}; \mathcal{M}) \prod_{t=1}^{T} \mathrm{p}(\boldsymbol{y}_{\mathsf{p},t}^l | \theta_t) \tag{4.62}$$

While this form of decoding with missing features is efficient, a form of bounded marginalisation gives much improved results by giving a bound on the integration. It was concluded that marginalisation gave superior accuracy to imputation in the spectral domain [19, 120]. However, marginalisation requires changes to the recogniser and is limited to using only spectral features whereas imputation can be used as a general front-end enhancement system by transforming the restored features into the cepstral domain [120].

In the uncertainty decoding framework, MFT can be viewed as an approximation to the corrupted speech conditional in the spectral domain for only the static features

$$\mathrm{p}(y_{t,i}^l | x_{t,i}^l; \check{\mathcal{M}}) = \begin{cases} \delta(y_{t,i}^l - x_{t,i}^l), & \text{if } y_{t,i}^l \text{ is present} \\ 1, & \text{if } y_{t,i}^l \text{ is missing} \end{cases} \tag{4.63}$$

[1]The cube root has also been used for energy compression in MFT [8].

which when performing the integration in a spectral domain version of equation (4.47)

$$\mathrm{p}(\boldsymbol{Y}^l;\mathcal{M},\mathcal{M}_n) \approx \sum_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \mathrm{P}(\boldsymbol{\theta}|\mathcal{M}) \prod_{t=1}^{T} \int_{\mathcal{R}^{D_s}} \mathrm{p}(\boldsymbol{y}_t^l|\boldsymbol{x}_t^l;\check{\mathcal{M}})\, \mathrm{p}(\boldsymbol{x}_t^l|\theta_t)\mathrm{d}\boldsymbol{x}_t^l \tag{4.64}$$

gives the same form as equation (4.62). Strictly in uncertainty decoding, for missing features $\mathrm{p}(y_{t,i}^l|x_{t,i}^l;\check{\mathcal{M}}) = \mathrm{p}(y_{t,i}^l|\mu_{z,i}^l,\sigma_{z,i}^{l2})$ since in areas where noise subsumes speech, the corrupted speech conditional distribution becomes the noise distribution [94]. Either value does not affect decoding since the same value is used for all output likelihood calculations at each frame. The difficulty for marginalisation in MFT is that it is carried out in the spectral domain, while most state-of-the-art systems operate in the cepstral domain. It also unnecessarily applies a "hard" decision on the reliability of the features, whereas uncertainty decoding is domain-agnostic avoiding this hard decision.

MFT imputation has been modified to use a "soft" mask [8, 108]. Instead of applying a hard decision to each channel, the decision is a weighted sum of the present and missing outcomes. It has also been extended by considered the features as "soft" data [108]. This applies to unreliable, missing features and is similar to observation uncertainty methods described here. An evidence pdf, $\mathrm{s}(\boldsymbol{y}_t^l;\check{\mathcal{M}})$, plays the same role as the posterior distribution in observation uncertainty

$$\mathrm{p}(\boldsymbol{y}_t^l|m;\check{\mathcal{M}}) \approx \int_{\mathcal{R}^{D_s}} \mathrm{s}(\boldsymbol{y}_t^l;\check{\mathcal{M}})\, \mathrm{p}(\boldsymbol{y}_t^l|m)\mathrm{d}\boldsymbol{y}_t^l \tag{4.65}$$

However, delta, uniform and bounded Gaussian distributions are evaluated as forms for the evidence pdf rather than a standard Gaussian. The delta form is equivalent to data imputation; the uniform distribution was found to be better than the bounded Gaussian.

Overall, this survey paper [119] has found that recognition with data imputation results transformed to the cepstral domain are superior to spectral domain marginalisation. It also concludes that marginalisation approaches in the cepstral domain are generally ineffective as shown in Van hamme [142].

4.6 Noise Model Estimation

For many noise compensation techniques a model of the noise is necessary. Frequently, an additive noise model is estimated from background, non-speech areas, such as the first and last 10-30 frames of each utterance—this has worked well for Algonquin enhancement [35], VTS feature compensation [127], MBFE [136], and Weiner filtering [10]. However, a robust voice activity detector is required and generally detecting speech becomes more difficult as the noise level increases. Furthermore, while this approach may provide a good model for short utterances, however some sentences may be sufficiently long that the noise environment changes while speech continues to be spoken. Even on the Aurora2 [68], which is a short artificially corrupted digit string recognition task, some gains are obtained by updating the model during the speech; for example, in Stouten et al. [138] the noise model is updated every 100 frames.

It is not straightforward to estimate a convolutional noise model using just the background segments of an utterance. In conjunction with a background estimated additive noise model, the channel noise may be estimated over the entire utterance using EM [135]. In contrast,

Moreno [106] provides an EM-based framework to estimate both the means of the additive and convolutional noise in a ML fashion in the log-spectral domain for only the static features. This allows for *unsupervised* noise estimation of the full noise model whilst the speaker is still speaking. The form of maximisation of the static additive and convolutional noise means described here is based on the ML formulation introduced in Moreno [106], but in the cepstral domain. The first-order Taylor series approximation in equation (4.27) may be used to express the static corrupted speech mean as a function of initial and new additive and convolutional noise means

$$\begin{aligned}\hat{\boldsymbol{\mu}}_y^{(m)} &\approx \mathcal{E}\Big\{\boldsymbol{y}\big|_{\mu_0^{(m)}} + \boldsymbol{J}_x^{(m)}\big(\boldsymbol{x}-\boldsymbol{\mu}_x^{(m)}\big) + \boldsymbol{J}_z^{(m)}\left(\boldsymbol{z}-\boldsymbol{\mu}_z\right) + \boldsymbol{J}_h^{(m)}\left(\boldsymbol{h}-\boldsymbol{\mu}_h\right)\Big\} \\ &= \boldsymbol{\mu}_y^{(m)} + \boldsymbol{J}_z^{(m)}(\hat{\boldsymbol{\mu}}_z - \boldsymbol{\mu}_z) + \boldsymbol{J}_h^{(m)}(\hat{\boldsymbol{\mu}}_h - \boldsymbol{\mu}_h) \end{aligned} \tag{4.66}$$

assuming that the speech and noise are independent. The terms with the Jacobian matrices will vanish when the estimated value and the current values of the noise means converge. The noise means are estimated in an ML fashion such that when they are combined with the clean speech acoustic model, they maximise the likelihood of some corrupted speech data $\boldsymbol{Y}$ from the mismatched test condition. The auxiliary function is as follows

$$\begin{aligned}\mathcal{Q}(\boldsymbol{\mu}_z, \boldsymbol{\mu}_h; \hat{\boldsymbol{\mu}}_z, \hat{\boldsymbol{\mu}}_h) &= \mathbb{E}_{\hat{\mathcal{M}}}\Big[\log \mathrm{p}(\boldsymbol{Y}, \boldsymbol{M}; \mathcal{M}, \hat{\mathcal{M}}_n)\Big] \\ &= \sum_{t=1}^{T}\sum_{m=1}^{M} \gamma_{y,t}^{(m)} \log \mathrm{p}(\boldsymbol{y}_t | m; \hat{\boldsymbol{\mu}}_z, \boldsymbol{\Sigma}_z, \hat{\boldsymbol{\mu}}_h) \\ &= \sum_{t=1}^{T}\sum_{m=1}^{M} \gamma_{y,t}^{(m)} \left\{-\frac{1}{2}\log|\boldsymbol{\Sigma}_y^{(m)}| - \frac{1}{2}\big(\boldsymbol{y}_t - \hat{\boldsymbol{\mu}}_y^{(m)}\big)^{\mathsf{T}} \boldsymbol{\Sigma}_y^{(m)\,-1}\big(\boldsymbol{y}_t - \hat{\boldsymbol{\mu}}_y^{(m)}\big)\right\} \end{aligned} \tag{4.67}$$

where $\mathcal{M}_n = \{\boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z, \boldsymbol{\mu}_h\}$ and only the terms dependent on the noise model are shown. The clean acoustic model parameters and the static additive noise variance $\boldsymbol{\Sigma}_z$ are unchanged throughout this noise mean estimation process. The component posterior $\gamma_{y,t}^{(m)} = \mathrm{P}(m_t = m|\boldsymbol{Y}, \mathcal{W}_{\mathsf{h}}; \mathcal{M}, \hat{\mathcal{M}}_n)$ is computed from the complete data set $\{\boldsymbol{Y}, \boldsymbol{M}\}$ which requires a hypothesis $\mathcal{W}_{\mathsf{h}}$ from an initial decoding run. The noisy speech acoustic model $\hat{\mathcal{M}}$ used to compute $\gamma_{y,t}^{(m)}$ is predicted by combining $\mathcal{M}$ and $\mathcal{M}_n$ using VTS compensation, but only for the static cepstral dimensions and with the zero-order form of the corrupted speech mean given in equation (4.32). The maximisation step differs by using the form given in equation (4.66). To find updated estimates of the additive and convolutional noise, the auxiliary function is differentiated with respect to the parameters sought and equated to zero to solve. A key simplifying factor is that the Jacobian matrices are considered constant although they are functions of the noise. Thus, the partial derivative of the auxiliary w.r.t. the additive

noise mean is

$$\begin{aligned}
\frac{\partial}{\partial\hat{\boldsymbol{\mu}}_z}\mathcal{Q}(\boldsymbol{\mu}_z,\boldsymbol{\mu}_h;\hat{\boldsymbol{\mu}}_z,\hat{\boldsymbol{\mu}}_h) &= \sum_{t=1}^{T}\sum_{m=1}^{M}\gamma_{y,t}^{(m)}\frac{\partial}{\partial\hat{\boldsymbol{\mu}}_z}\log \mathrm{p}(\boldsymbol{y}_t|m;\hat{\boldsymbol{\mu}}_z,\boldsymbol{\Sigma}_z,\hat{\boldsymbol{\mu}}_h) \\
&= \sum_{t=1}^{T}\sum_{m=1}^{M}\gamma_{y,t}^{(m)}\frac{\partial}{\partial\hat{\boldsymbol{\mu}}_z}\left\{-\frac{1}{2}\log|\boldsymbol{\Sigma}_y^{(m)}| - \frac{1}{2}\big(\boldsymbol{y}_t-\hat{\boldsymbol{\mu}}_y^{(m)}\big)^{\mathsf{T}}\boldsymbol{\Sigma}_y^{(m)^{-1}}\big(\boldsymbol{y}_t-\hat{\boldsymbol{\mu}}_y^{(m)}\big)\right\} \\
&= \sum_{t=1}^{T}\sum_{m=1}^{M}\gamma_{y,t}^{(m)}\left\{0-\frac{1}{2}\frac{\partial}{\partial\hat{\boldsymbol{\mu}}_z}\big(\boldsymbol{y}_t-\hat{\boldsymbol{\mu}}_y^{(m)}\big)^{\mathsf{T}}\boldsymbol{\Sigma}_y^{(m)^{-1}}\big(\boldsymbol{y}_t-\hat{\boldsymbol{\mu}}_y^{(m)}\big)\right\} \\
&= \sum_{t=1}^{T}\sum_{m=1}^{M}\gamma_{y,t}^{(m)}\boldsymbol{J}_z^{(m)\mathsf{T}}\boldsymbol{\Sigma}_y^{(m)^{-1}}\times \\
&\qquad \Big(\boldsymbol{y}_t-\boldsymbol{\mu}_y^{(m)}+\boldsymbol{J}_z^{(m)}\boldsymbol{\mu}_z+\boldsymbol{J}_h^{(m)}\boldsymbol{\mu}_h-\boldsymbol{J}_z^{(m)}\hat{\boldsymbol{\mu}}_z-\boldsymbol{J}_h^{(m)}\hat{\boldsymbol{\mu}}_h\Big) \\
&= \boldsymbol{d}-\boldsymbol{E}\hat{\boldsymbol{\mu}}_z-\boldsymbol{F}\hat{\boldsymbol{\mu}}_h
\end{aligned} \tag{4.68}$$

where

$$\boldsymbol{d} = \sum_{m=1}^{M}\boldsymbol{J}_z^{(m)\mathsf{T}}\boldsymbol{\Sigma}_y^{(m)^{-1}}\sum_{t=1}^{T}\gamma_{y,t}^{(m)}\Big(\boldsymbol{y}_t-\boldsymbol{\mu}_y^{(m)}+\boldsymbol{J}_z^{(m)}\boldsymbol{\mu}_z+\boldsymbol{J}_h^{(m)}\boldsymbol{\mu}_h\Big) \tag{4.69}$$

$$\boldsymbol{E} = \sum_{m=1}^{M}\gamma_y^{(m)}\boldsymbol{J}_z^{(m)\mathsf{T}}\boldsymbol{\Sigma}_y^{(m)^{-1}}\boldsymbol{J}_z^{(m)} \qquad \boldsymbol{F} = \sum_{m=1}^{M}\gamma_y^{(m)}\boldsymbol{J}_z^{(m)\mathsf{T}}\boldsymbol{\Sigma}_y^{(m)^{-1}}\boldsymbol{J}_h^{(m)} \tag{4.70}$$

and $\gamma_y^{(m)} = \sum_{t=1}^{T}\gamma_{y,t}^{(m)}$. Similarly, for the convolutional noise the partial derivative is

$$\begin{aligned}
\frac{\partial}{\partial\hat{\boldsymbol{\mu}}_h}\mathcal{Q}(\boldsymbol{\mu}_z,\boldsymbol{\mu}_h;\hat{\boldsymbol{\mu}}_z,\hat{\boldsymbol{\mu}}_h) &= \sum_{t=1}^{T}\sum_{m=1}^{M}\gamma_{y,t}^{(m)}\frac{\partial}{\partial\hat{\boldsymbol{\mu}}_h}\log \mathrm{p}(\boldsymbol{y}_t|m;\hat{\boldsymbol{\mu}}_z,\boldsymbol{\Sigma}_z,\hat{\boldsymbol{\mu}}_h) \\
&= \sum_{t=1}^{T}\sum_{m=1}^{M}\gamma_{y,t}^{(m)}\left\{0-\frac{1}{2}\frac{\partial}{\partial\hat{\boldsymbol{\mu}}_h}\big(\boldsymbol{y}_t-\hat{\boldsymbol{\mu}}_y^{(m)}\big)^{\mathsf{T}}\boldsymbol{\Sigma}_y^{(m)^{-1}}\big(\boldsymbol{y}_t-\hat{\boldsymbol{\mu}}_y^{(m)}\big)\right\} \\
&= \sum_{t=1}^{T}\sum_{m=1}^{M}\gamma_{y,t}^{(m)}\boldsymbol{J}_h^{(m)\mathsf{T}}\boldsymbol{\Sigma}_y^{(m)^{-1}}\times \\
&\qquad \Big(\boldsymbol{y}_t-\boldsymbol{\mu}_y^{(m)}+\boldsymbol{J}_z^{(m)}\boldsymbol{\mu}_z+\boldsymbol{J}_h^{(m)}\boldsymbol{\mu}_h-\boldsymbol{J}_z^{(m)}\hat{\boldsymbol{\mu}}_z-\boldsymbol{J}_h^{(m)}\hat{\boldsymbol{\mu}}_h\Big) \\
&= \boldsymbol{u}-\boldsymbol{V}\hat{\boldsymbol{\mu}}_z-\boldsymbol{W}\hat{\boldsymbol{\mu}}_h
\end{aligned} \tag{4.71}$$

where

$$\boldsymbol{u} = \sum_{m=1}^{M}\boldsymbol{J}_h^{(m)\mathsf{T}}\boldsymbol{\Sigma}_y^{(m)^{-1}}\sum_{t=1}^{T}\gamma_{y,t}^{(m)}\Big(\boldsymbol{y}_t-\boldsymbol{\mu}_y^{(m)}+\boldsymbol{J}_z^{(m)}\boldsymbol{\mu}_z+\boldsymbol{J}_h^{(m)}\boldsymbol{\mu}_h\Big) \tag{4.72}$$

$$\boldsymbol{V} = \sum_{m=1}^{M}\gamma_y^{(m)}\boldsymbol{J}_h^{(m)\mathsf{T}}\boldsymbol{\Sigma}_y^{(m)^{-1}}\boldsymbol{J}_z^{(m)} \qquad \boldsymbol{W} = \sum_{m=1}^{M}\gamma_y^{(m)}\boldsymbol{J}_h^{(m)\mathsf{T}}\boldsymbol{\Sigma}_y^{(m)^{-1}}\boldsymbol{J}_h^{(m)} \tag{4.73}$$

These derivatives given in equations (4.68) and (4.71) can be equated with zero to find the optimal points of the auxiliary function

$$\boldsymbol{d}-\boldsymbol{E}\hat{\boldsymbol{\mu}}_z-\boldsymbol{F}\hat{\boldsymbol{\mu}}_h = 0 \tag{4.74}$$

$$\boldsymbol{u}-\boldsymbol{V}\hat{\boldsymbol{\mu}}_z-\boldsymbol{W}\hat{\boldsymbol{\mu}}_h = 0 \tag{4.75}$$

which can be written in matrix form as

$$\begin{bmatrix} \boldsymbol{E} & \boldsymbol{F} \\ \boldsymbol{V} & \boldsymbol{W} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\mu}}_z \\ \hat{\boldsymbol{\mu}}_h \end{bmatrix} = \begin{bmatrix} \boldsymbol{d} \\ \boldsymbol{u} \end{bmatrix} \tag{4.76}$$

Note that $\boldsymbol{V} = \boldsymbol{F}^{\mathsf{T}}$. Solving the linear system in equation (4.76) gives the following formulae for the parameters to be estimated

$$\hat{\boldsymbol{\mu}}_h = (\boldsymbol{W} - \boldsymbol{V}\boldsymbol{E}^{-1}\boldsymbol{F})^{-1}(\boldsymbol{u} - \boldsymbol{V}\boldsymbol{E}^{-1}\boldsymbol{d}) \tag{4.77}$$

$$\begin{aligned} \hat{\boldsymbol{\mu}}_z &= (\boldsymbol{V} - \boldsymbol{W}\boldsymbol{F}^{-1}\boldsymbol{E})^{-1}(\boldsymbol{u} - \boldsymbol{W}\boldsymbol{F}^{-1}\boldsymbol{d}) \\ &= (\boldsymbol{E} - \boldsymbol{F}\boldsymbol{W}^{-1}\boldsymbol{V})^{-1}(\boldsymbol{d} - \boldsymbol{F}\boldsymbol{W}^{-1}\boldsymbol{u}) \end{aligned} \tag{4.78}$$

This provides ML estimates of the channel and additive noise means, for compensating static parameters since the auxiliary function in equation (4.67) only includes the static elements of the observations. Chapter 6 discusses methods of estimating an ML noise estimate for systems that also include dynamic coefficients. An estimate of the additive noise variance is discussed as well.

4.7 Summary

There has been much research on improving ASR performance in noisy environments. Front-end feature-based techniques tend to be computationally efficient, and responsive to changing conditions, but tend to fail in noisier conditions. Model compensation is more powerful, but requires a considerable number of acoustic model component updates and hence comes with a heavy computational cost. While adaptive forms such as MAP or CMLLR may be used to compensate for noise, the predictive forms discussed in this chapter, such as PMC and VTS compensation, have the advantage that only a noise model for an environment is necessary to compensate the system. Adaptive forms need sufficient noise-corrupted speech data to robustly estimate transforms for example. In contrast, predictive forms only require enough test data, which does not need to contain speech, to estimate the noise model. Hence, although noise may be inherently unpredictable, it may be characterised into additive and convolutional components, and estimated, leading to noise compensation forms that can help ASR handle unseen, adverse environments.

Recently, there has been interest in uncertainty-based noise compensation techniques. Uncertainty forms may be considered a hybrid between feature compensation and model-based approaches since the features are updated, but the model update is simply a bias added to the model variances. There are important differences between *observation uncertainty*, which adds the enhancement variance to the model variances, and *uncertainty decoding*, which is a form that results when using the framework for noise robust speech recognition presented in this chapter. The limitations and issues of observation uncertainty were discussed. Uncertainty decoding, and in particular joint uncertainty decoding, is discussed in detail in the next chapter.

CHAPTER 5
Joint Uncertainty Decoding

Joint uncertainty decoding can be viewed as a set of model-based compensation approaches that are characterised by a feature transformation and an "uncertainty" bias on the model variances. This form of compensation is much more efficient than pure model-based forms like VTS model compensation, but can be just as powerful—both these attributes are due to this simple uncertainty bias. The transformation is derived from the joint distribution between the clean and noisy speech, or more generally the training and test speech. While the joint distribution may be estimated from so-called stereo data, it can also be predicted by using noise mismatch functions and models of the clean speech and noise. This chapter presents joint uncertainty decoding (JUD), which falls under this latter approach and has front-end and model-based flavours. A comparison of these two forms of JUD gives important insights into the limitations of all front-end uncertainty decoding approaches.

5.1 The Corrupted Speech Conditional Distribution

It can be seen, from the likelihood of the complete corrupted speech observation sequence given in equation (4.47), that the likelihood of a corrupted speech observation $\boldsymbol{o}_t$ may be expressed as the following integral

$$\mathrm{p}(\boldsymbol{o}_t|\theta_t;\check{\mathcal{M}}) = \int_{\mathcal{R}^D} \mathrm{p}\big(\boldsymbol{o}_t|\boldsymbol{s}_t;\check{\mathcal{M}}\big)\,\mathrm{p}\big(\boldsymbol{s}_t|\theta_t\big)\mathrm{d}\boldsymbol{s}_t \tag{5.1}$$

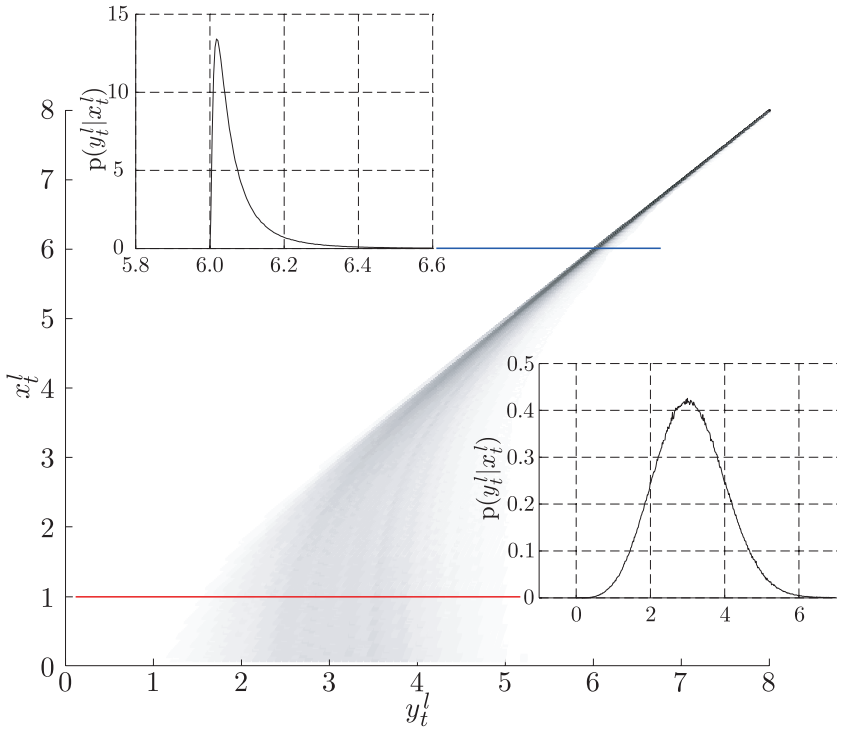


Figure 5.1: Joint distribution of clean x_t^l and corrupted speech y_t^l with an additive noise source $\mathcal{N}(3, 1)$ in log spectral domain.

where the first distribution in the integrand is the corrupted speech conditional distribution and the second is the prior distribution of the clean speech $\boldsymbol{s}_t$. Recall that θ_t is the hidden clean speech state of acoustic model set $\mathcal{M}$, $\check{\mathcal{M}}$ the front-end compensation parameter set and D is the number of dimensions of the feature vector. Only the corrupted speech conditional distribution is dependent on the noise; the prior is the state output distribution of the clean speech acoustic model. Hence, an important issue in uncertainty decoding is finding an efficient yet accurate representation of the corrupted speech conditional distribution that is also amenable to marginalisation with a Gaussian distribution. The main difficulty is that $\mathrm{p}(\boldsymbol{o}_t|\boldsymbol{s}_t;\check{\mathcal{M}})$ is complex. This is demonstrated by a numerical simulation of the joint clean and corrupted speech distribution in figure 5.1, again using equation (3.13) from section 3.2

$$y_t^l = \log(\exp(x_t^l) + \exp(z_t^l))$$

where recall y_t^l is the corrupted speech and the subscript l indicates a log-spectral domain variable[1]. The additive noise z_t^l again is generated from a single Gaussian distribution. The clean speech x_t^l is uniform over the interval $[0, 8]$ to demonstrate how the joint distribution changes as the clean speech does with a fixed noise source. The joint distribution is highly non-linear and difficult to characterise parametrically.

The corrupted speech conditional distribution varies greatly over the range of values shown. When the speech value is much larger than the noise mean, i.e. when $x_t^l = 6$ in

[1]The apparent change of variables occurs because in this simulation, a single static dimension is being examined. The complete vectors $\boldsymbol{o}_t$, $\boldsymbol{s}_t$ and $\boldsymbol{n}_t$ are the aggregation of static, delta and delta-delta coefficients, i.e. $\boldsymbol{o}_t = [\boldsymbol{y}_t^\mathsf{T} \;\; \Delta\boldsymbol{y}_t^\mathsf{T} \;\; \Delta^2\boldsymbol{y}_t^\mathsf{T}]^\mathsf{T}$, $\boldsymbol{s}_t = [\boldsymbol{x}_t^\mathsf{T} \;\; \Delta\boldsymbol{x}_t^\mathsf{T} \;\; \Delta^2\boldsymbol{z}_t^\mathsf{T}]^\mathsf{T}$ and $\boldsymbol{n}_t = [\boldsymbol{z}_t^\mathsf{T} \;\; \Delta\boldsymbol{z}_t^\mathsf{T} \;\; \Delta^2\boldsymbol{z}_t^\mathsf{T}]^\mathsf{T}$

figure 5.1, the conditional distribution is relatively deterministic—the speech is unaffected by the noise. However, when the SNR drops the variance of the conditional distribution increases until the noise subsumes the speech. For example when $x_t^l = 1$ in figure 5.1, the conditional distribution becomes the additive noise distribution. Increasing the noise mean would shift the distribution up and to the right such that when the SNR is low, the corrupted speech conditional distribution continues to converge to the noise distribution. Thus the effective form of the corrupted speech conditional distribution strongly depends on the difference between the clean speech and the noise and the noise variance. Approximating the conditional distribution with a constant density function independent of the clean speech would be poor.

5.2 Gaussian Approximations

As most HMM-based recognisers use a GMM form of the clean speech state distribution, if the corrupted speech conditional distribution is also Gaussian, then deriving an analytical form is trivial since the Gaussian family of distributions are self-conjugate—the convolution of two Gaussians also yields a Gaussian. This section discusses approaches to modelling the corrupted speech conditional distribution such that each acoustic model component is convolved with a single Gaussian distribution. The first form, front-end Joint, selects an appropriate Gaussian conditional distribution at each time frame using a front-end GMM. The second, chooses the representative Gaussian for the conditional distribution based on which regression class the acoustic model component belongs to.

5.2.1 Front-end JUD

Similar to how the clean speech posterior was approximated by a front-end GMM in SPLICE and SPLICEU, the corrupted speech conditional distribution in equation (5.1) may also be represented by a GMM

$$\mathrm{p}\big(\boldsymbol{o}_t|\boldsymbol{s}_t;\check{\mathcal{M}}\big) \approx \sum_{k=1}^{K} \mathrm{P}\big(k|\boldsymbol{s}_t;\check{\mathcal{M}}\big)\,\mathcal{N}(\boldsymbol{o}_t;f_\mu(\boldsymbol{s}_t,k),f_\Sigma(\boldsymbol{s}_t,k)) \tag{5.2}$$

where the mean and variance of the kth component output distribution is dependent on the clean speech $\boldsymbol{s}_t$ as denoted by $f_\mu(\boldsymbol{s}_t,k)$ and $f_\Sigma(\boldsymbol{s}_t,k)$. With this approximation to the corrupted speech conditional distribution, two issues to address are:

- the component posterior $\mathrm{P}(k|\boldsymbol{s}_t;\check{\mathcal{M}})$ is conditioned on the clean speech;
- the number of components may influence the total number of effective components evaluated.

The component posterior is conditional on the hidden "clean speech" variable which depends on the state of the clean speech model. However for efficiency, the front-end compensation should be as independent of the acoustic models as much as possible. Furthermore, 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 the following approximations.

The first issue may be resolved by making the component posterior conditional on the observed corrupted speech rather than the hidden clean speech

$$\mathrm{P}(k|\boldsymbol{s}_t) \approx \mathrm{P}(k|\boldsymbol{o}_t) \tag{5.3}$$

This has a crude effect of passing the same component, and thus conditional distribution, to the decoder, regardless of the state in the clean speech acoustic model. As seen from 5.1, the conditional distribution should have a smaller variance in high SNR, and larger variance when the SNR is low. However by using this approximation, the computation of the conditional distribution is completely independent of the clean speech acoustic model state.

The component output distribution parameters need to be derived. By assuming the joint distribution of the clean and corrupted for a front-end component k is Gaussian

$$\begin{bmatrix}\boldsymbol{s}_t \\ \boldsymbol{o}_t\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_s^{(k)} \\ \boldsymbol{\mu}_o^{(k)}\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}_s^{(k)} & \boldsymbol{\Sigma}_{so}^{(k)} \\ \boldsymbol{\Sigma}_{os}^{(k)} & \boldsymbol{\Sigma}_o^{(k)}\end{bmatrix}\right) \tag{5.4}$$

a Gaussian conditional distribution can be derived from the joint, as shown in appendix A.1

$$\begin{aligned}&\mathcal{N}\big(\boldsymbol{o}_t; f_\mu(\boldsymbol{s}_t, k), f_\Sigma(\boldsymbol{s}_t, k)\big) \\ &\quad = \mathcal{N}\Big(\boldsymbol{o}_t; \boldsymbol{\mu}_o^{(k)} + \boldsymbol{\Sigma}_{os}^{(k)}\boldsymbol{\Sigma}_s^{(k)\text{-}1}\big(\boldsymbol{s}_t - \boldsymbol{\mu}_s^{(k)}\big), \boldsymbol{\Sigma}_o^{(k)} - \boldsymbol{\Sigma}_{os}^{(k)}\boldsymbol{\Sigma}_s^{(k)\text{-}1}\boldsymbol{\Sigma}_{so}^{(k)}\Big)\end{aligned} \tag{5.5}$$

It is a property of multivariate Gaussian distributions, that if the joint distribution is Gaussian distributed, then the conditional distribution is as well. In equation (5.5), the clean speech variable $\boldsymbol{s}_t$ is transformed. However, by applying the transformation on the features rather than on $\boldsymbol{s}_t$, the resulting compensation of the acoustic model component will be simplified later. Thus, first the feature space will be transformed by $\boldsymbol{\Sigma}_s^{(k)}\boldsymbol{\Sigma}_{os}^{(k)\text{-}1}$. This updates the variance in equation (5.5), now defined as $\boldsymbol{\Sigma}_{\mathsf{b}}^{(k)}$, as follows

$$\begin{aligned}\boldsymbol{\Sigma}_{\mathsf{b}}^{(k)} &= \boldsymbol{\Sigma}_s^{(k)}\boldsymbol{\Sigma}_{os}^{(k)\text{-}1}\Big(\boldsymbol{\Sigma}_o^{(k)} - \boldsymbol{\Sigma}_{os}^{(k)}\boldsymbol{\Sigma}_s^{(k)\text{-}1}\boldsymbol{\Sigma}_{so}^{(k)}\Big)\big(\boldsymbol{\Sigma}_s^{(k)}\boldsymbol{\Sigma}_{os}^{(k)\text{-}1}\big)^{\mathsf{T}} \\ &= \boldsymbol{\Sigma}_s^{(k)}\boldsymbol{\Sigma}_{os}^{(k)\text{-}1}\boldsymbol{\Sigma}_o^{(k)}\boldsymbol{\Sigma}_{os}^{(k)\text{-}\mathsf{T}}\boldsymbol{\Sigma}_s^{(k)} - \boldsymbol{\Sigma}_s^{(k)}\end{aligned} \tag{5.6}$$

Equation (5.5) may then be re-expressed in the following manner

$$\begin{aligned}&\mathcal{N}\big(\boldsymbol{o}_t; f_\mu(\boldsymbol{s}_t, k), f_\Sigma(\boldsymbol{s}_t, k)\big) \\ &\quad = |\boldsymbol{\Sigma}_s^{(k)}\boldsymbol{\Sigma}_{os}^{(k)\text{-}1}|\mathcal{N}\Big(\boldsymbol{\Sigma}_s^{(k)}\boldsymbol{\Sigma}_{os}^{(k)\text{-}1}\boldsymbol{o}_t; \boldsymbol{\Sigma}_s^{(k)}\boldsymbol{\Sigma}_{os}^{(k)\text{-}1}\boldsymbol{\mu}_o^{(k)} + \boldsymbol{s}_t - \boldsymbol{\mu}_s^{(k)}, \boldsymbol{\Sigma}_{\mathsf{b}}^{(k)}\Big) \\ &\quad = |\boldsymbol{\Sigma}_s^{(k)}\boldsymbol{\Sigma}_{os}^{(k)\text{-}1}|\mathcal{N}\Big(\boldsymbol{\Sigma}_s^{(k)}\boldsymbol{\Sigma}_{os}^{(k)\text{-}1}\big(\boldsymbol{o}_t - \boldsymbol{\mu}_o^{(k)}\big) + \boldsymbol{\mu}_s^{(k)}; \boldsymbol{s}_t, \boldsymbol{\Sigma}_{\mathsf{b}}^{(k)}\Big) \\ &\quad = |\boldsymbol{A}^{(k)}|\mathcal{N}\Big(\boldsymbol{A}^{(k)}\boldsymbol{o}_t + \boldsymbol{b}^{(k)}; \boldsymbol{s}_t, \boldsymbol{\Sigma}_{\mathsf{b}}^{(k)}\Big)\end{aligned} \tag{5.7}$$

where

$$\boldsymbol{A}^{(k)} = \boldsymbol{\Sigma}_s^{(k)}\boldsymbol{\Sigma}_{os}^{(k)\text{-}1} \tag{5.8}$$

$$\boldsymbol{b}^{(k)} = \boldsymbol{\mu}_s^{(k)} - \boldsymbol{A}^{(k)}\boldsymbol{\mu}_o^{(k)} \tag{5.9}$$

$$\boldsymbol{\Sigma}_{\mathsf{b}}^{(k)} = \boldsymbol{A}^{(k)}\boldsymbol{\Sigma}_o^{(k)}\boldsymbol{A}^{(k)\mathsf{T}} - \boldsymbol{\Sigma}_s^{(k)} \tag{5.10}$$

With the approximations for the component posterior and this Gaussian form of the component corrupted speech conditional distribution, equation (5.1) may be expressed as follows

$$\begin{aligned}
\mathrm{p}\big(\boldsymbol{o}_t|\theta_t;\check{\mathcal{M}}\big) &= \int_{\mathcal{R}^D} \mathrm{p}\big(\boldsymbol{o}_t|\boldsymbol{s}_t;\check{\mathcal{M}}\big)\,\mathrm{p}(\boldsymbol{s}_t|\theta_t)\,\mathrm{d}\boldsymbol{s}_t \\
&\approx \int_{\mathcal{R}^D} \sum_{k=1}^{K} \mathrm{P}\big(k|\boldsymbol{o}_t\big)|\boldsymbol{A}^{(k)}|\mathcal{N}\big(\boldsymbol{A}^{(k)}\boldsymbol{o}_t+\boldsymbol{b}^{(k)};\boldsymbol{s}_t,\boldsymbol{\Sigma}_{\mathsf{b}}^{(k)}\big)\sum_{m\in\theta_t} c^{(m)}\mathcal{N}\big(\boldsymbol{s}_t;\boldsymbol{\mu}_s^{(m)},\boldsymbol{\Sigma}_s^{(m)}\big)\mathrm{d}\boldsymbol{s}_t \\
&= \sum_{k=1}^{K}\sum_{m\in\theta_t} c^{(m)}\,\mathrm{P}\big(k|\boldsymbol{o}_t\big)|\boldsymbol{A}^{(k)}|\int_{\mathcal{R}^D}\mathcal{N}\big(\boldsymbol{A}^{(k)}\boldsymbol{o}_t+\boldsymbol{b}^{(k)};\boldsymbol{s}_t,\boldsymbol{\Sigma}_{\mathsf{b}}^{(k)}\big)\mathcal{N}\big(\boldsymbol{s}_t;\boldsymbol{\mu}_s^{(m)},\boldsymbol{\Sigma}_s^{(m)}\big)\mathrm{d}\boldsymbol{s}_t
\end{aligned} \tag{5.11}$$

where the notation $m\in\theta_t$ indicates all components in state θ_t of acoustic model $\mathcal{M}$. The integral with two Gaussian distributions may be treated as a convolution, which yields a single Gaussian

$$\mathrm{p}\big(\boldsymbol{o}_t|\theta_t;\check{\mathcal{M}}\big) \approx \sum_{k=1}^{K}\sum_{m\in\theta_t} c^{(m)}\,\mathrm{P}\big(k|\boldsymbol{o}_t\big)|\boldsymbol{A}^{(k)}|\mathcal{N}\Big(\boldsymbol{A}^{(k)}\boldsymbol{o}_t+\boldsymbol{b}^{(k)};\boldsymbol{\mu}_s^{(m)},\boldsymbol{\Sigma}_s^{(m)}+\boldsymbol{\Sigma}_{\mathsf{b}}^{(k)}\Big) \tag{5.12}$$

as shown in appendix A.2.

The second issue is now clear: directly decoding with equation (5.12) may be computationally expensive since the overall number of Gaussian evaluations is $K\times M$. It would be more efficient to pass a single Gaussian as in SPLICE. This entails selecting the most appropriate front-end component k^*

$$k^* = \underset{k}{\mathrm{argmax}}\,\mathrm{P}(k|\boldsymbol{o}_t) \tag{5.13}$$

where the component posterior was defined previously in equation (4.12). With this approximation, equation (5.12) becomes

$$\mathrm{p}\big(\boldsymbol{o}_t|\theta_t;\check{\mathcal{M}}\big) \propto \sum_{m\in\theta_t} c^{(m)}|\boldsymbol{A}^{(k^*)}|\mathcal{N}\Big(\boldsymbol{A}^{(k^*)}\boldsymbol{o}_t+\boldsymbol{b}^{(k^*)};\boldsymbol{\mu}_s^{(m)},\boldsymbol{\Sigma}_s^{(m)}+\boldsymbol{\Sigma}_{\mathsf{b}}^{(k^*)}\Big) \tag{5.14}$$

which is the same form of SPLICEU likelihood calculation given in equation (4.59), but with the JUD transform parameters given by

$$\boldsymbol{A}^{(k^*)} = \boldsymbol{\Sigma}_s^{(k^*)}\boldsymbol{\Sigma}_{os}^{(k^*)\text{-}1} \tag{5.15}$$

$$\boldsymbol{b}^{(k^*)} = \boldsymbol{\mu}_s^{(k^*)} - \boldsymbol{A}^{(k^*)}\boldsymbol{\mu}_o^{(k^*)} \tag{5.16}$$

$$\boldsymbol{\Sigma}_{\mathsf{b}}^{(k^*)} = \boldsymbol{A}^{(k^*)}\boldsymbol{\Sigma}_o^{(k^*)}\boldsymbol{A}^{(k^*)\mathsf{T}} - \boldsymbol{\Sigma}_s^{(k^*)} \tag{5.17}$$

This form of uncertainty decoding is called FE-Joint. Although the forms are similar, FE-Joint and SPLICEU are derived in a different way with different approximations. Compared to the SPLICEU parameters given in equations (4.53) to (4.55), no explicit flooring is required and the matrix parameters $\boldsymbol{A}^{(k^*)}$ and $\boldsymbol{\Sigma}_{\mathsf{b}}^{(k^*)}$ and may be full.

Figure 5.2 demonstrates the operation of the SPLICEU and FE-Joint. It shows how the compensation parameters are selected in the front-end and then applied during decoding. While similar to feature-based forms like POF [109], MBFE or front-end CMLLR [93], where the affine feature transform is selected by a front-end GMM, in uncertainty decoding there is the addition of a "uncertainty" variance bias added to the model variances.

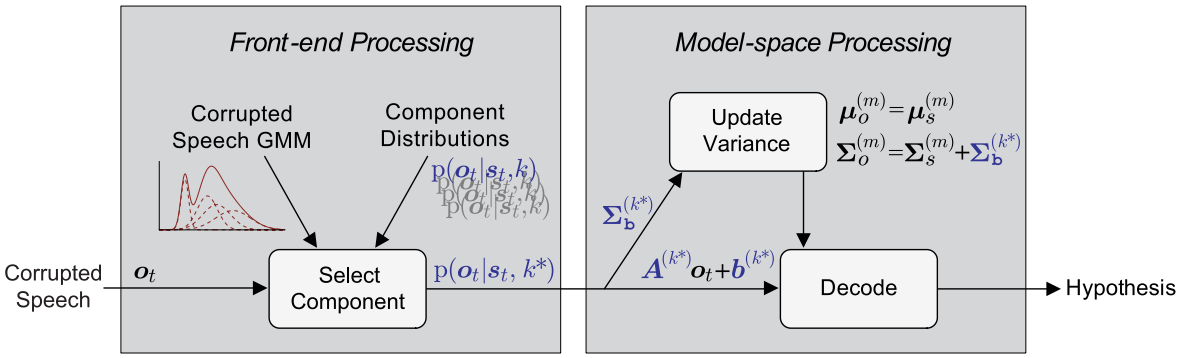


Figure 5.2: Front-end uncertainty decoding.

5.2.2 Issues with Front-end Uncertainty Decoding Schemes

One serious drawback of front-end uncertainty schemes is that the model variances must be updated every time the variance bias changes. The variance bias changes as the front-end component k^* changes. Although, the update is simple compared to a technique such as model-based VTS compensation, the update and re-computation of the normalisation term must be executed for every acoustic model component. However, there is an even larger concern for *front-end* uncertainty decoding forms that choose a single transform of the features and model variances at each time frame.

Consider the joint distribution of the clean speech and noise shown previously in figure 5.1. Two corrupted speech conditional distributions, $\mathrm{p}(o_t|s_t)$, are marked. The upper one results when the SNR is relatively high, with the clean speech $x_t = 6$ compared to the additive noise mean of 3. This yields a highly skewed distribution that peaks sharply and is highly non-Gaussian, yet modelled with a Gaussian distribution. As the SNR increases this becomes more pronounced until it becomes a delta function yielding the clean speech distribution when substituted in equation (5.1). This is expected, since when the SNR is high, the noise should have no influence on compensating the acoustic models.

The corrupted speech conditional distribution looks very different when the SNR is low. For example, in figure 5.1 consider when $x_t = 1$. At this point, the distribution is Gaussian, matching the corrupting additive noise distribution, with a mean of 3 and variance of 1. Thus in low SNR, the conditional distribution degenerates to the distribution of the additive noise

$$\mathrm{p}(\boldsymbol{o}_t|\boldsymbol{s}_t;\check{\mathcal{M}}) \approx \mathcal{N}(\boldsymbol{o}_t;\boldsymbol{\mu}_n,\boldsymbol{\Sigma}_n) \tag{5.18}$$

where $\boldsymbol{\mu}_n$ and $\boldsymbol{\Sigma}_n$ are the additive noise mean and variance respectively. Intuitively this makes sense, since the noise masks the speech. This result has also been independently documented in Benítez et al. [10], however the consequences for uncertainty decoding forms, such as SPLICEU and JUD, were not examined. If equation (5.18) is substituted into equation (5.1), the distribution of the corrupted speech becomes the additive noise distribution

$$\begin{aligned}\mathrm{p}(\boldsymbol{o}_t|\theta_t;\check{\mathcal{M}}) &\approx \int_{\mathcal{R}^D} \mathcal{N}\big(\boldsymbol{o}_t;\boldsymbol{\mu}_n,\boldsymbol{\Sigma}_n\big)\,\mathrm{p}(\boldsymbol{s}_t|\theta_t)\mathrm{d}\boldsymbol{s}_t \\ &= \mathcal{N}\big(\boldsymbol{o}_t;\boldsymbol{\mu}_n,\boldsymbol{\Sigma}_n\big)\int_{\mathcal{R}^D}\mathrm{p}(\boldsymbol{s}_t|\theta_t)\mathrm{d}\boldsymbol{s}_t \\ &= \mathcal{N}(\boldsymbol{o}_t;\boldsymbol{\mu}_n,\boldsymbol{\Sigma}_n)\end{aligned} \tag{5.19}$$

since the conditional distribution is no longer a function of the clean speech.

Thus regardless of the original recognition model component, the compensated distribution used during decoding will always be identical to the noise distribution. When a single conditional distribution is estimated and used for all components, in low SNR conditions a frame, or sequence of frames, will have no discriminatory power between classes: every distribution will look the same. If the recognition task has additional constraints beyond the acoustic models, such as a language model, then it may be possible to distinguish between different models during these non-discriminatory regions if these are applied. However, when there is no language model or other restrictions, for example with a continuous digit recognition task such as Aurora2, then these areas where no discriminatory acoustic information is available will be very susceptible to errors. These errors will probably be insertions since these areas are likely to be background regions, although low-energy speech may be substituted by other models if the noise is significant enough to mask the speech.

Insight into this aspect of the conditional distribution may be gained by examining the nature of the joint distribution, as given in equation (5.4), in low energy speech regions. For regions with low SNR, the corrupted speech distribution is dominated by the noise; in other words, the noise masks the speech. The cross-covariance term $\boldsymbol{\Sigma}_{so}^{(k^*)}$ for a front-end component associated with these regions of low speech energy will be approximately zero since the clean speech and noise are independent

$$\boldsymbol{\Sigma}_{so}^{(k^*)} \approx \mathbf{0} \tag{5.20}$$

This lack of correlation drives $\boldsymbol{A}^{(k^*)}$, defined in equation (5.15), to infinity along with the uncertainty bias. In front-end uncertainty decoding, this is the expected behaviour because the front-end has determined that in these areas, the uncertainty is high, since the SNR is low. The relationship to equation (5.19) becomes clearer by examining equation (5.14) for a single model component m

$$\begin{aligned}
\mathrm{p}\big(\boldsymbol{o}_t|m;\check{\mathcal{M}}\big) &= |\boldsymbol{A}^{(k^*)}|\mathcal{N}\Big(\boldsymbol{A}^{(k^*)}\boldsymbol{o}_t+\boldsymbol{b}^{(k^*)};\boldsymbol{\mu}_s^{(m)},\boldsymbol{\Sigma}_s^{(m)}+\boldsymbol{\Sigma}_{\mathsf{b}}^{(k^*)}\Big) \\
&= \mathcal{N}\Big(\boldsymbol{o}_t+\boldsymbol{A}^{(k^*)\text{-}1}\boldsymbol{b}^{(k^*)};\boldsymbol{A}^{(k^*)\text{-}1}\boldsymbol{\mu}_s^{(m)},\boldsymbol{A}^{(k^*)\text{-}1}(\boldsymbol{\Sigma}_s^{(m)}+\boldsymbol{\Sigma}_{\mathsf{b}}^{(k^*)})\boldsymbol{A}^{(k^*)\text{-}\mathsf{T}}\Big) \\
&= \mathcal{N}\Big(\boldsymbol{o}_t;\boldsymbol{A}^{(k^*)\text{-}1}(\boldsymbol{\mu}_s^{(m)}-\boldsymbol{b}^{(k^*)}),\boldsymbol{A}^{(k^*)\text{-}1}(\boldsymbol{\Sigma}_s^{(m)}+\boldsymbol{\Sigma}_{\mathsf{b}}^{(k^*)})\boldsymbol{A}^{(k^*)\text{-}\mathsf{T}}\Big)
\end{aligned} \tag{5.21}$$

simplifying the mean and variance, given $\boldsymbol{A}^{(k^*)\text{-}1}=\boldsymbol{\Sigma}_{os}^{(k^*)}\boldsymbol{\Sigma}_s^{(k^*)\text{-}1}\approx\mathbf{0}$ from equation (5.20)

$$\begin{aligned}
\boldsymbol{A}^{(k^*)\text{-}1}(\boldsymbol{\mu}_s^{(m)}-\boldsymbol{b}^{(k^*)}) &= \boldsymbol{A}^{(k^*)\text{-}1}(\boldsymbol{\mu}_s^{(m)}-\boldsymbol{\mu}_s^{(k^*)}-\boldsymbol{A}^{(k^*)}\boldsymbol{\mu}_o^{(k^*)}) \\
&= \boldsymbol{A}^{(k^*)\text{-}1}(\boldsymbol{\mu}_s^{(m)}-\boldsymbol{\mu}_s^{(k^*)})+\boldsymbol{\mu}_o^{(k^*)} \\
&\approx \boldsymbol{\mu}_o^{(k^*)}
\end{aligned} \tag{5.22}$$

$$\begin{aligned}
&\boldsymbol{A}^{(k^*)\text{-}1}\big(\boldsymbol{\Sigma}_s^{(m)}+\boldsymbol{\Sigma}_{\mathsf{b}}^{(k^*)}\big)\boldsymbol{A}^{(k^*)\text{-}\mathsf{T}} \\
&\quad= \boldsymbol{A}^{(k^*)\text{-}1}\boldsymbol{\Sigma}_s^{(m)}\boldsymbol{A}^{(k^*)\text{-}\mathsf{T}}+\boldsymbol{A}^{(k^*)\text{-}1}\big(\boldsymbol{A}^{(k^*)}\boldsymbol{\Sigma}_o^{(k^*)}\boldsymbol{A}^{(k^*)\mathsf{T}}-\boldsymbol{\Sigma}_s^{(k^*)}\big)\boldsymbol{A}^{(k^*)\text{-}\mathsf{T}} \\
&\quad= \boldsymbol{A}^{(k^*)\text{-}1}\boldsymbol{\Sigma}_s^{(m)}\boldsymbol{A}^{(k^*)\text{-}\mathsf{T}}+\boldsymbol{\Sigma}_o^{(k^*)}-\boldsymbol{A}^{(k^*)\text{-}1}\boldsymbol{\Sigma}_s^{(k^*)}\boldsymbol{A}^{(k^*)\text{-}\mathsf{T}} \\
&\quad\approx \boldsymbol{\Sigma}_o^{(k^*)}
\end{aligned} \tag{5.23}$$

hence

$$\mathrm{p}\big(\boldsymbol{o}_t|m;\check{\mathcal{M}}\big) \approx \mathcal{N}\Big(\boldsymbol{o}_t;\boldsymbol{\mu}_o^{(k^*)},\boldsymbol{\Sigma}_o^{(k^*)}\Big) \tag{5.24}$$

which is simply the noise distribution when component k^* represents a background acoustic region. This occurs regardless of what component m or state θ_t is. If all acoustic model components are mapped to the noise distribution for several frames, then search errors may result in these regions. These will likely be insertions because the uncertainty will be highest in low energy, non-speech regions.

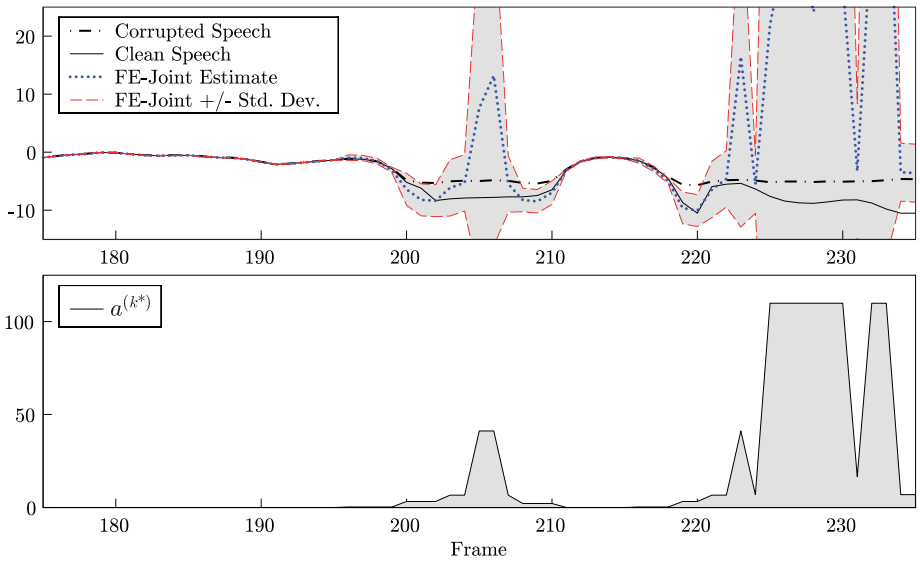


Figure 5.3: Plot of log energy dimension from Aurora2 digit string 8-6-zero-1-1-6-2, showing 16-component GMM FE-Joint estimate $a^{(k^*)}o_t + b^{(k^*)}$, uncertainty bias $\sigma_{\mathsf{b}}^{(k^*)}$, and $a^{(k^*)}$.

A clear illustration of this issue with FE-Joint is presented in figure 5.3. This figure shows the clean speech, corrupted speech, FE-Joint estimate, given by $a^{(k^*)}o_t + b^{(k^*)}$, and the uncertainty bias $\sigma_{\mathsf{b}}^{(k^*)}$ for a simple system with a 16-component front-end GMM. For those regions of higher energy speech, for example frames 210 to 220 where the vowel 'i' is articulated, the variance bias is small. On the other hand, in the lower energy regions around this vowel, for example frames 225 to 230, the variance becomes too large to be measured on this scale, as is the FE-Joint estimate of the value. These large variances are associated with large values of the scale factor $a^{(k^*)}$ as shown in figure 5.3 due to very small correlations between the clean and corrupted speech as discussed earlier. In this example, from frames 225 to 230 the value of $a^{(k^*)}$ is around 100. With greater numbers of front-end components, these effects are amplified as parameters are no longer smoothed.

5.2.3 Front-end JUD with Flooring

The behaviour of FE-Joint compensation in low SNR, where the feature scaling and uncertainty variance bias both become very large, is a straightforward result of the assumptions used to make this form of compensation efficient. However, as discussed in the previous section, the approximations applied may lead to excess insertions in noisier areas. Consequently, the compensation parameters may be limited such that all the acoustic model components are not transformed to the noise model. An obvious approach is to manipulate the correlation

coefficients for each of the dimensions, defined as

$$\rho_{so,i}^{(k^*)} = \frac{\sigma_{so,i}^{(k^*)}}{\sqrt{\sigma_{s,i}^{(k^*)2}\sigma_{o,i}^{(k^*)2}}} \tag{5.25}$$

where $\sigma_{so,i}^{(k^*)}$ is defined as the covariance of dimension i between the clean speech and noisy speech for component k^*.

The compensation parameter estimates given in equations (5.15) to (5.17) can then be re-expressed in terms of the correlation coefficient as

$$a_{ii}^{(k^*)} = \frac{\sigma_{s,i}^{(k^*)}}{\rho_{so,i}^{(k^*)}\sigma_{o,i}^{(k^*)}} \tag{5.26}$$

$$b_i^{(k^*)} = \mu_{s,i}^{(k^*)} - a_{ii}^{(k^*)}\mu_{o,i}^{(k^*)} \tag{5.27}$$

$$\sigma_{\mathsf{b},i}^{(k^*)2} = \frac{\sigma_{s,i}^{(k^*)2}}{\rho_{so,i}^{(k^*)2}} - \sigma_{s,i}^{(k^*)2} \tag{5.28}$$

for the diagonal form of FE-Joint. To restrict extreme values of $a_{ii}^{(k^*)}$ and $\sigma_{\mathsf{b},i}^{(k^*)2}$, a minimum value on the correlation coefficient can be enforced. Accordingly, the correlation $\rho_{so,i}^{(k^*)}$ in equations (5.26) to (5.28) is set to

$$\hat{\rho}_{so,i}^{(k^*)} = \max\left(\rho_{so,i}^{(k^*)}, \rho\right) \tag{5.29}$$

where ρ is an empirically determined constant. Increasing the value of ρ raises the minimum acceptable correlation, decreasing the maximum variance bias. This can be viewed as enforcing a SNR floor; although the actual local SNR fall below this, the compensation scheme acts as though the floor is the actual level. The effects of this flooring on the same snippet of artificially corrupted speech from figure 5.3 is shown in figure 5.4. As anticipated, the extremes in the variance bias observed before have been reduced.

In the limit, it is possible to set $\rho = 1$, which can be interpreted as assuming there is no noise in the environment, resulting in $\sigma_{\mathsf{b},i}^{(k^*)2} = 0$, from equation (5.28); this leads to an enhancement form with this estimate of the clean speech

$$\hat{s}_{t,i} = \mu_{s,i}^{(k^*)} + \frac{\sigma_{s,i}^{(k^*)}}{\sigma_{o,i}^{(k^*)}}\left(o_{t,i} - \mu_{o,i}^{(k^*)}\right) \tag{5.30}$$

This can be compared to the MBFE clean speech estimate, given in equation (4.21), with diagonal covariances

$$\hat{s}_{t,i} = \mu_{s,i}^{(k^*)} + \frac{\sigma_{os,i}^{(k^*)}}{\sigma_{o,i}^{(k^*)2}}\left(o_{t,i} - \mu_{o,i}^{(k^*)}\right) \tag{5.31}$$

where a single max front-end component chosen at each time frame. If the clean and noisy speech are fully correlated, then $\sigma_{os,i}^{(k^*)} = \sigma_{o,i}^{(k^*)}\sigma_{s,i}^{(k^*)}$ and equations (5.30) and (5.31) simplify to the same form.

Returning to the fundamental issue with front-end uncertainty forms, it was shown that FE-Joint suffers from a problem where all output distributions become the same in low SNR.

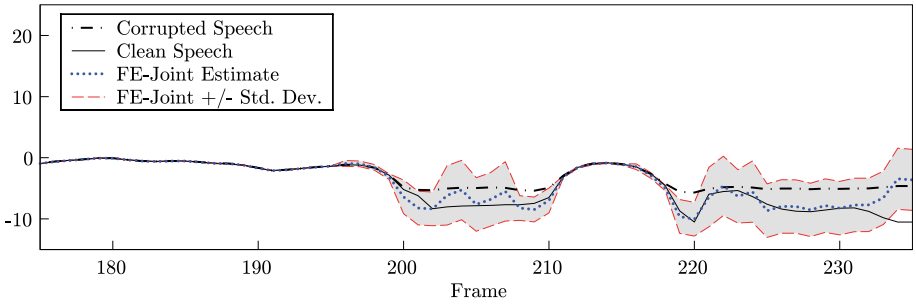


Figure 5.4: Plot of log energy dimension from Aurora2 digit string 8-6-zero-1-1-6-2, showing 16-component GMM FE-Joint estimate $a^{(k^*)}o_t + b^{(k^*)}$, and uncertainty bias $\sigma_{\mathsf{b}}^{(k^*)}$, with correlation flooring $\rho = 0.1$.

Since this is a problem for all front-end uncertainty forms then SPLICEU should also suffer from it. However this issue has not been observed, for example, on the Aurora2 results presented in [26]. This is because SPLICEU applies a limit on the maximum value of the variance bias scaling factor $a_{ii}^{(k^*)}$ to $1/\alpha$ in equation (4.58). Here α is also an empirically determined parameter. In addition to this explicit flooring, there is also an under-estimate of the value of $a_{ii}^{(k^*)}$. In order to make the calculation of the SPLICEU uncertainty efficient, a global variance is used in the denominator of equation (4.55). Since this will be larger than any of the individual front-end components that should be used, the scaling estimate will be lower than expected as can be discerned from this equation. This under-estimation will become larger as the number of front-end components increases, therefore the variance of the individual model components will become smaller and smaller compared to the global variance and exactly the situation when a component might expected to be associated only with a noise region.

5.2.4 Model-based JUD Transforms

As opposed to associating a corrupted speech conditional distribution with a region of the feature space, the conditional distribution may be linked with group of acoustic model components. Model components may be clustered using a regression tree as described in section 2.5.1. For each class, a joint distribution of the clean and corrupted speech can be estimated, and therefore a corrupted speech conditional distribution determined. Hence, the conditional distribution is a function of which regression class, r, the acoustic model component, m, belongs to and may again be approximated by a Gaussian distribution

$$\mathrm{p}(\boldsymbol{o}_t|\boldsymbol{s}_t;\check{\mathcal{M}}) \approx \mathcal{N}(\boldsymbol{o}_t; f_\mu(\boldsymbol{s}_t, r), f_\Sigma(\boldsymbol{s}_t, r)) \tag{5.32}$$

The mean and variance of the distribution are now a function of the class and the clean speech as denoted by $f_\mu(\boldsymbol{s}_t, r)$ and $f_\Sigma(\boldsymbol{s}_t, r)$. This conditional distribution can be derived from a joint distribution of the clean and corrupted speech for the class r much like in the front-end case

$$\begin{bmatrix}\boldsymbol{s}_t \\ \boldsymbol{o}_t\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_s^{(r)} \\ \boldsymbol{\mu}_o^{(r)}\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}_s^{(r)} & \boldsymbol{\Sigma}_{so}^{(r)} \\ \boldsymbol{\Sigma}_{os}^{(r)} & \boldsymbol{\Sigma}_o^{(r)}\end{bmatrix}\right) \tag{5.33}$$

where a Gaussian approximation of the joint distribution also gives a Gaussian form for the corrupted speech conditional distribution. When this form of the corrupted speech conditional

distribution is substituted into equation (5.1), the corrupted speech observation likelihood for a state θ_t becomes

$$\mathrm{p}(\boldsymbol{o}_t|\theta_t;\check{\mathcal{M}}) \approx \sum_{m\in\theta_t} c^{(m)}|\boldsymbol{A}^{(r_m)}|\mathcal{N}\Big(\boldsymbol{A}^{(r_m)}\boldsymbol{o}_t + \boldsymbol{b}^{(r_m)}; \boldsymbol{\mu}_s^{(m)}, \boldsymbol{\Sigma}_s^{(m)} + \boldsymbol{\Sigma}_{\mathsf{b}}^{(r_m)}\Big) \tag{5.34}$$

where $\check{\mathcal{M}} = \left\{\boldsymbol{A}^{(1)}, \boldsymbol{b}^{(1)}, \boldsymbol{\Sigma}_{\mathsf{b}}^{(1)}, \ldots, \boldsymbol{A}^{(R)}, \boldsymbol{b}^{(R)}, \boldsymbol{\Sigma}_{\mathsf{b}}^{(R)}\right\}$, r_m denotes the regression class r that component m belongs to, and R the total number of regression classes. As in FE-Joint, the transform parameters are a function of the joint distribution parameters

$$\boldsymbol{A}^{(r_m)} = \boldsymbol{\Sigma}_s^{(r_m)}\boldsymbol{\Sigma}_{os}^{(r_m)\text{-}1} \tag{5.35}$$

$$\boldsymbol{b}^{(r_m)} = \boldsymbol{\mu}_s^{(r_m)} - \boldsymbol{A}^{(r_m)}\boldsymbol{\mu}_o^{(r_m)} \tag{5.36}$$

$$\boldsymbol{\Sigma}_{\mathsf{b}}^{(r_m)} = \boldsymbol{A}^{(r_m)}\boldsymbol{\Sigma}_o^{(r_m)}\boldsymbol{A}^{(r_m)\mathsf{T}} - \boldsymbol{\Sigma}_s^{(r_m)} \tag{5.37}$$

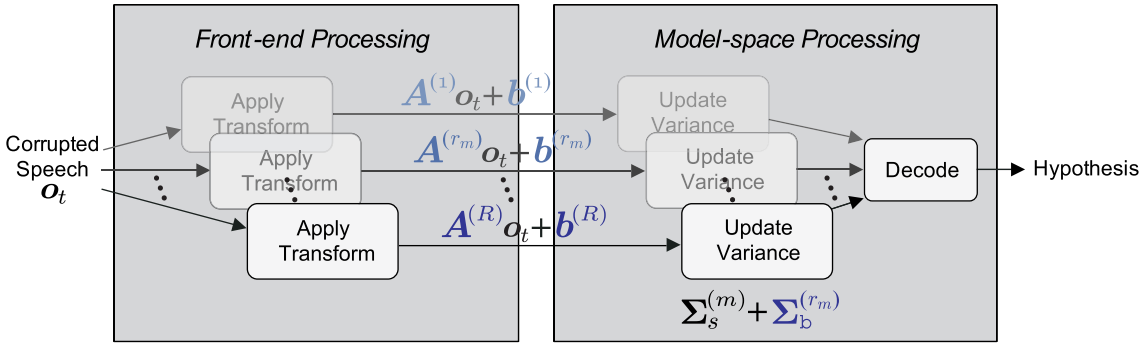


Figure 5.5: Model-based joint uncertainty decoding.

The operation of this model-based JUD form, which will be referred to as M-Joint compensation, is shown in figure 5.5. Compared to the FE-Joint form depicted in figure 5.2, the observation is transformed by multiple transforms, much like in CMLLR, such that there are R parallel versions of the observation passed to the decoder. Also each class has a different variance bias associated with it. However, compared to FE-Joint, this variance does not change over time and may be cached; it need only be updated if the noise condition itself changes. Since for any given time frame, model components are being compensated by different transforms, M-Joint compensation will not be affected by the issues discussed previously in section 5.2.2. Moreover, for FE-Joint, the cost of selecting a single maximum component from K components in the front-end, is of similar order to applying R transforms for multiple features in M-Joint. Thus for equivalent numbers of K and R, FE-Joint and M-Joint are of similar computational complexity.

5.3 Approximating the Joint Distribution

The structure of the covariance matrices of the joint distribution in equation (5.4) or (5.33) will affect the overall computational cost. For example, a block-diagonal form may be used

$$\mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_s^{(r)}\\ \boldsymbol{\mu}_o^{(r)}\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}_s^{(r)} & \boldsymbol{\Sigma}_{so}^{(r)}\\ \boldsymbol{\Sigma}_{os}^{(r)} & \boldsymbol{\Sigma}_o^{(r)}\end{bmatrix}\right) \approx \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_x^{(r)}\\ \boldsymbol{\mu}_{\Delta x}^{(r)}\\ \boldsymbol{\mu}_{\Delta^2 x}^{(r)}\\ \boldsymbol{\mu}_y^{(r)}\\ \boldsymbol{\mu}_{\Delta y}^{(r)}\\ \boldsymbol{\mu}_{\Delta^2 y}^{(r)}\end{bmatrix}, \begin{bmatrix}\begin{bmatrix}\boldsymbol{\Sigma}_x^{(r)} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \boldsymbol{\Sigma}_{\Delta x}^{(r)} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Sigma}_{\Delta^2 x}^{(r)}\end{bmatrix} & \begin{bmatrix}\boldsymbol{\Sigma}_{xy}^{(r)} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \boldsymbol{\Sigma}_{\Delta x\Delta y}^{(r)} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Sigma}_{\Delta^2 x\Delta^2 y}^{(r)}\end{bmatrix}\\ \begin{bmatrix}\boldsymbol{\Sigma}_{xy}^{(r)} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \boldsymbol{\Sigma}_{\Delta x\Delta y}^{(r)} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Sigma}_{\Delta^2 x\Delta^2 y}^{(r)}\end{bmatrix}^{\mathsf{T}} & \begin{bmatrix}\boldsymbol{\Sigma}_y^{(r)} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \boldsymbol{\Sigma}_{\Delta y}^{(r)} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Sigma}_{\Delta^2 y}^{(r)}\end{bmatrix}\end{bmatrix}\right) \tag{5.38}$$

where it may be assumed that the covariance between the static, delta and acceleration dimensions is zero. Furthermore, if each static, delta and acceleration covariance matrix is diagonal, then $\boldsymbol{\Sigma}_{\mathsf{b}}^{(r_m)}$ will also be diagonal. This diagonal approximation was shown to provide good robustness, and yields transforms that may each be applied in linear time to the feature vector and a model component. However, as discussed in section 3.3, noise may introduce changes in intra-frame correlations hence block-diagonal and full covariance forms may give improved results. Using these more precise covariance matrices though will require block-diagonal or full covariance decoding though since $\boldsymbol{\Sigma}_{\mathsf{b}}^{(r_m)}$ will become block-diagonal or full. Adding this term to the model variances, i.e. $\boldsymbol{\Sigma}_s^{(m)}+\boldsymbol{\Sigma}_{\mathsf{b}}^{(r_m)}$, results in an acoustic model covariance of that is no longer diagonal which makes the Gaussian likelihood calculation rather inefficient. Keeping the linear feature transform full, and diagonalising the variance bias term produced poor results [93][1].

5.4 Estimating JUD Compensation Parameters

Estimation of the JUD compensation parameters has so far not been discussed. The parameters are derived from the joint distribution of the clean and corrupted speech as given by equations (5.15) to (5.17) for FE-Joint or equations (5.35) to (5.37) for M-Joint. The joint distribution may be estimated with stereo data, although in practice stereo data are not typically available. The joint distribution may also be predicted using a clean speech model, a noise model, and a mismatch function describing how the two combine to form noisy speech. Figure 5.6 shows the general method for estimating the compensation parameters for a regression class. The term "predicted" is used since the joint distribution is not directly estimated from adaptation data from the test environment. Instead only a low-dimensional set of noise model parameters are estimated from the adaptation data, unlike MLLR transforms that are directly estimated from such data. In predictive forms of compensation, the noise model is then combined with *a priori* speech models to derive compensated parameters [47]. This is achieved through so-called mismatch function approximations that relate noise, clean speech

[1] This may be overcome by training a semi-tied transform, in effect a feature-space transform, that maintains the diagonal uncertainty bias term, but provides a rotation of the feature space [48].

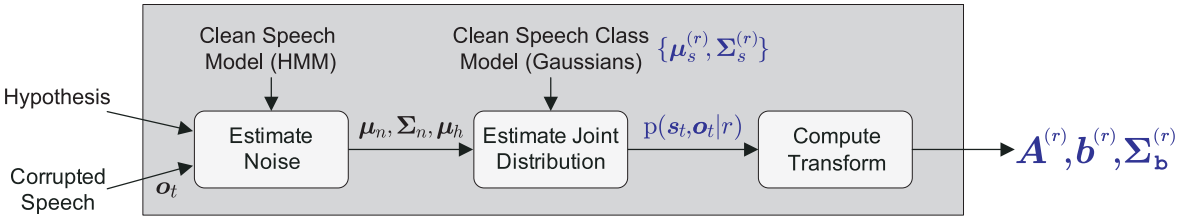


Figure 5.6: Estimating model-based joint uncertainty decoding transforms.

and noisy speech, such as log-normal or log-add as discussed in section 4.4.2, or the VTS approximation given in equation (4.27). These give direct, analytic means of predicting the joint distribution; alternatively, DPMC, as discussed in section 4.4.2, can be used to iteratively predict the joint distribution. The noise model necessary for these schemes may include estimates of the static additive noise mean $\boldsymbol{\mu}_n$, static channel mean $\boldsymbol{\mu}_h$ and additive noise covariance

$$\boldsymbol{\Sigma}_n = \begin{bmatrix} \boldsymbol{\Sigma}_z & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{\Delta z} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Sigma}_{\Delta^2 z} \end{bmatrix} \tag{5.39}$$

The additive noise variance need not be block-diagonal; it may be full or diagonal with the latter being the actual form examined in this work. Methods to estimate a noise model are discussed in the next chapter. In addition, a single Gaussian $\mathcal{N}(\boldsymbol{\mu}_s^{(r)}, \boldsymbol{\Sigma}_s^{(r)})$ models each clean speech regression class; this is discussed further in section 5.4.1. Once the joint distribution is estimated, the FE-Joint or M-Joint transform parameters are easily computed. The same approach is used in Xu et al. [149] for front-end joint uncertainty decoding comparing VTS and DPMC techniques to generate the joint distribution and in the MBFE technique, discussed in section 4.3.3, to derive a joint distribution for each front-end state [135].

In section 4.4.3 it was shown how given a distribution of the clean speech and a noise model, the corrupted speech distribution can be estimated using a first-order VTS approximation of the acoustic environment model. For joint uncertainty decoding, the cross-covariance $\boldsymbol{\Sigma}_{os}^{(r)}$ is also necessary. Similar to the derivation of the corrupted speech covariance, the cross-covariance will be broken down into three blocks on the main diagonal as in equation (5.38). The static cross-covariance is derived by first determining the covariance between the static clean speech and the first-order VTS approximation of the static corrupted speech

$$\begin{aligned} \boldsymbol{\Sigma}_{yx}^{(r)} &= \mathcal{E}\{\boldsymbol{y}\boldsymbol{x}^{\mathsf{T}}|r\} - \boldsymbol{\mu}_y^{(r)}\boldsymbol{\mu}_x^{(r)\mathsf{T}} \\ &\approx \mathcal{E}\{\boldsymbol{y}_{\mathsf{vts}}\boldsymbol{x}^{\mathsf{T}}|r\} - \boldsymbol{\mu}_y^{(r)}\boldsymbol{\mu}_x^{(r)\mathsf{T}} \end{aligned} \tag{5.40}$$

The first-order VTS approximation was given by equation (4.27), and repeated here

$$\boldsymbol{y}_{\mathsf{vts}} = \boldsymbol{y}|_{\mu_0^{(r)}} + \boldsymbol{J}_x^{(r)}\left(\boldsymbol{x} - \boldsymbol{\mu}_x^{(r)}\right) + \boldsymbol{J}_z^{(r)}\left(\boldsymbol{z} - \boldsymbol{\mu}_z\right) + \boldsymbol{J}_h^{(r)}\left(\boldsymbol{h} - \boldsymbol{\mu}_h\right)$$

Recall that $|_{\mu_0^{(r)}}$ indicates evaluation at the VTS expansion point $\mu_0^{(r)}$; however, now the clean speech mean for a regression class r, $\boldsymbol{\mu}_x^{(r)}$, is used rather than the component mean $\boldsymbol{\mu}_x^{(m)}$. The Jacobian matrix $\boldsymbol{J}_x^{(r)}$ is also evaluated at the same expansion point. Each row of the Jacobian matrices gives the gradient of a dimension of the corrupted speech with respect to the clean

speech, additive noise or convolutional noise vectors; the Jacobian matrices were defined in equations (4.28) to (4.30).

Assuming independence between the clean speech and noise allows only the terms from the truncated VTS approximation of the corrupted speech that affect the cross-covariance, i.e. are a function of the clean speech, to be considered

$$\begin{aligned}\boldsymbol{\Sigma}_{yx}^{(r)} &\approx \mathcal{E}\left\{\left[\boldsymbol{J}_x^{(r)}(\boldsymbol{x}-\boldsymbol{\mu}_x^{(r)})\right]\boldsymbol{x}^{\mathsf{T}}\middle|r\right\} - \mathcal{E}\left\{\boldsymbol{J}_x^{(r)}(\boldsymbol{x}-\boldsymbol{\mu}_x^{(r)})\middle|r\right\}\boldsymbol{\mu}_x^{(r)\mathsf{T}} \\ &= \boldsymbol{J}_x^{(r)}\boldsymbol{\Sigma}_x^{(r)}\end{aligned} \tag{5.41}$$

since the covariance between a random vector and its transformed version is simply a linear transform of the random vector covariance as shown in appendix A.3. Thus the static cross-covariance for a regression class is given by

$$\boldsymbol{\Sigma}_{yx}^{(r)} \approx \boldsymbol{J}_x^{(r)}\boldsymbol{\Sigma}_x^{(r)} \tag{5.42}$$

To derive the delta and delta-delta cross-covariances, the Continuous-Time approximation is applied as it was in section 4.4.3. This gives

$$\boldsymbol{\Sigma}_{\Delta y\Delta x}^{(r)} \approx \boldsymbol{J}_x^{(r)}\boldsymbol{\Sigma}_{\Delta x}^{(r)} \tag{5.43}$$

$$\boldsymbol{\Sigma}_{\Delta^2 y\Delta^2 x}^{(r)} \approx \boldsymbol{J}_x^{(r)}\boldsymbol{\Sigma}_{\Delta^2 x}^{(r)} \tag{5.44}$$

A full derivation may be found in appendix section B.3. This approach does not consider the static–delta, delta–delta-delta and static–delta-delta cross covariances; a matrix approach to deriving dynamic feature coefficients [39] may give analytic forms of these terms. Equations (5.42) to (5.44) provide the cross-covariance terms for the joint distribution in equation (5.38). Hence an approximation to the joint distribution may be predicted from the clean speech parameters $\boldsymbol{\mu}_s^{(r)}$ and $\boldsymbol{\Sigma}_s^{(r)}$ and a noise model, using the derivations for the corrupted speech parameters from equations (4.40) and (4.41), and these cross-covariance terms. For example, the transform matrix in equation (5.35), using this approximate joint distribution, is given by

$$\begin{aligned}\boldsymbol{A}^{(r)} &= \boldsymbol{\Sigma}_s^{(r)}\boldsymbol{\Sigma}_{os}^{(r)\text{-}1} \\ &= \begin{bmatrix}\boldsymbol{\Sigma}_x^{(r)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{\Delta x}^{(r)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Sigma}_{\Delta^2 x}^{(r)}\end{bmatrix}\begin{bmatrix}\boldsymbol{\Sigma}_{yx}^{(r)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{\Delta y\Delta x}^{(r)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Sigma}_{\Delta^2 y\Delta^2 x}^{(r)}\end{bmatrix}^{\text{-}1} \\ &\approx \begin{bmatrix}\boldsymbol{\Sigma}_x^{(r)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{\Delta x}^{(r)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Sigma}_{\Delta^2 x}^{(r)}\end{bmatrix}\begin{bmatrix}\left[\boldsymbol{J}_x^{(r)}\boldsymbol{\Sigma}_x^{(r)}\right]^{\text{-}1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \left[\boldsymbol{J}_x^{(r)}\boldsymbol{\Sigma}_{\Delta x}^{(r)}\right]^{\text{-}1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \left[\boldsymbol{J}_x^{(r)}\boldsymbol{\Sigma}_{\Delta^2 x}^{(r)}\right]^{\text{-}1}\end{bmatrix}\end{aligned} \tag{5.45}$$

Noting that $(\boldsymbol{AB})^{\text{-}1} = \boldsymbol{B}^{\text{-}1}\boldsymbol{A}^{\text{-}1}$, this simplifies to

$$\boldsymbol{A}^{(r)} \approx \begin{bmatrix}\boldsymbol{J}_x^{(r)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{J}_x^{(r)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{J}_x^{(r)}\end{bmatrix}^{\text{-}1} \tag{5.46}$$

For this work, when using noise models to predict the joint distribution, the joint distribution terms $\boldsymbol{\Sigma}_s^{(r)}$, $\boldsymbol{\Sigma}_o^{(r)}$ and $\boldsymbol{\Sigma}_{so}^{(r)}$ are diagonalised—this actually results in a diagonal $\boldsymbol{A}^{(r)}$ to also

give a diagonal uncertainty variance bias. Recall that $\boldsymbol{J}_x^{(r)} \rightarrow \boldsymbol{I}$ when the noise level is low; in this case, the transform is also identity indicating that no compensation is required. When the noise subsumes the speech $\boldsymbol{J}_x^{(r)} \rightarrow \mathbf{0}$, which causes $\boldsymbol{A}^{(r)} \rightarrow \infty$. The implications of this were discussed in section 5.2.2. For M-Joint compensation, this is not an issue so long as only a subset of the transforms are affected in this manner, since different model regression classes are transformed by different transforms. If they are all affected, then the noise is strong enough to mask all speech, and no speech information is available in the signal to transcribe. In this work, when predicting the joint distribution blocks—$\boldsymbol{\Sigma}_o^{(r)}$, $\boldsymbol{\Sigma}_{os}^{(r)}$, and $\boldsymbol{\Sigma}_s^{(r)}$—are all diagonalised resulting in a diagonal $\boldsymbol{A}^{(r)}$ and variance bias $\boldsymbol{\Sigma}_{\mathsf{b}}^{(r)}$. Diagonalising these matrices is not optimal though since it has been shown that full matrix forms are superior [93].

The feature bias from equation (5.36) may also be simplified

$$\begin{aligned}
\boldsymbol{b}^{(r)} &= \boldsymbol{\mu}_s^{(r)} - \boldsymbol{A}^{(r)}\boldsymbol{\mu}_o^{(r)} \\
&\approx \begin{bmatrix} \boldsymbol{\mu}_x^{(r)} \\ \boldsymbol{\mu}_{\Delta x}^{(r)} \\ \boldsymbol{\mu}_{\Delta^2 x}^{(r)} \end{bmatrix} - \begin{bmatrix} \boldsymbol{J}_x^{(r)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{J}_x^{(r)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{J}_x^{(r)} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{y}_t|_{\mu_0^{(r)}} \\ \boldsymbol{J}_x^{(r)}\boldsymbol{\mu}_{\Delta x}^{(r)} \\ \boldsymbol{J}_x^{(r)}\boldsymbol{\mu}_{\Delta^2 x}^{(r)} \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{\mu}_x^{(r)} - \left[\boldsymbol{J}_x^{(r)}\right]^{-1}\boldsymbol{y}|_{\mu_0^{(r)}} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}
\end{aligned} \tag{5.47}$$

The uncertainty variance bias, from equation (5.37), has a block diagonal form

$$\begin{aligned}
\boldsymbol{\Sigma}_{\mathsf{b}}^{(r)} &= \boldsymbol{A}^{(r)}\boldsymbol{\Sigma}_o^{(r)}\boldsymbol{A}^{(r)\mathsf{T}} - \boldsymbol{\Sigma}_s^{(r)} \\
&= \begin{bmatrix} \boldsymbol{J}_x^{(r)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{J}_x^{(r)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{J}_x^{(r)} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{\Sigma}_y^{(r)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{\Delta y}^{(r)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Sigma}_{\Delta^2 y}^{(r)} \end{bmatrix} \begin{bmatrix} \boldsymbol{J}_x^{(r)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{J}_x^{(r)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{J}_x^{(r)} \end{bmatrix}^{-\mathsf{T}} - \begin{bmatrix} \boldsymbol{\Sigma}_x^{(r)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{\Delta x}^{(r)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Sigma}_{\Delta^2 x}^{(r)} \end{bmatrix}
\end{aligned} \tag{5.48}$$

The static block may be simplified as follows

$$\begin{aligned}
\left[\boldsymbol{\Sigma}_{\mathsf{b}}^{(r)}\right]_{11} &= \left[\boldsymbol{J}_x^{(r)}\right]^{-1}\boldsymbol{\Sigma}_y^{(r)}\left[\boldsymbol{J}_x^{(r)}\right]^{-\mathsf{T}} - \boldsymbol{\Sigma}_x^{(r)} \\
&= \left[\boldsymbol{J}_x^{(r)}\right]^{-1}\left[\boldsymbol{J}_x^{(r)}\boldsymbol{\Sigma}_x^{(r)}\boldsymbol{J}_x^{(r)\mathsf{T}} + \boldsymbol{J}_z^{(r)}\boldsymbol{\Sigma}_z\boldsymbol{J}_z^{(r)\mathsf{T}}\right]\left[\boldsymbol{J}_x^{(r)}\right]^{-\mathsf{T}} - \boldsymbol{\Sigma}_x^{(r)} \\
&= \left[\boldsymbol{J}_x^{(r)}\right]^{-1}\boldsymbol{J}_z^{(r)}\boldsymbol{\Sigma}_z\boldsymbol{J}_z^{(r)\mathsf{T}}\left[\boldsymbol{J}_x^{(r)}\right]^{-\mathsf{T}}
\end{aligned} \tag{5.49}$$

Similarly, the delta and delta-delta blocks are

$$\left[\boldsymbol{\Sigma}_{\mathsf{b}}^{(r)}\right]_{22} = \left[\boldsymbol{J}_x^{(r)}\right]^{-1}\boldsymbol{J}_z^{(r)}\boldsymbol{\Sigma}_{\Delta z}\boldsymbol{J}_z^{(r)\mathsf{T}}\left[\boldsymbol{J}_x^{(r)}\right]^{-\mathsf{T}} \tag{5.50}$$

$$\left[\boldsymbol{\Sigma}_{\mathsf{b}}^{(r)}\right]_{33} = \left[\boldsymbol{J}_x^{(r)}\right]^{-1}\boldsymbol{J}_z^{(r)}\boldsymbol{\Sigma}_{\Delta^2 z}\boldsymbol{J}_z^{(r)\mathsf{T}}\left[\boldsymbol{J}_x^{(r)}\right]^{-\mathsf{T}} \tag{5.51}$$

Notice that if the noise level is high, then $\boldsymbol{J}_x^{(r)} \rightarrow 0$, which causes the uncertainty bias to become very large since the Jacobian matrix for clean speech is inverted. If the noise level is low then $\boldsymbol{J}_x^{(r)} \rightarrow 0$ causes the bias to become small.

The quality of using a VTS-based approximation for the corrupted speech may be investigated through a simulation in the log-spectral domain. Figure 5.7 shows how the first-order

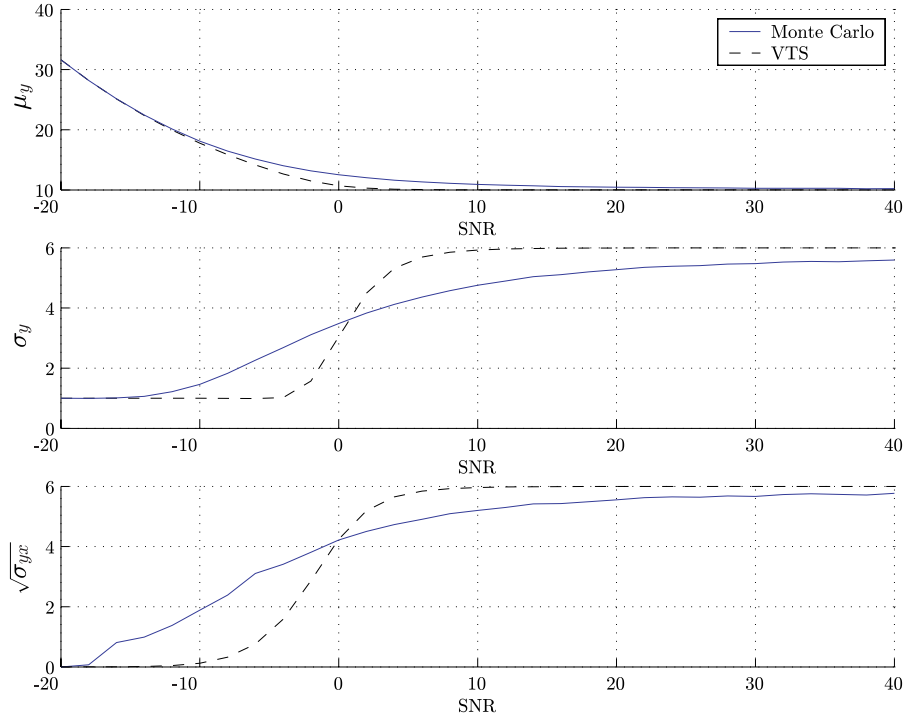


Figure 5.7: Comparing Monte Carlo and VTS generated corrupted speech y_t^l distributions and cross-covariance between and clean and corrupted speech in log-spectral domain.

VTS approximation of the compensated mean and variances compare to ML estimates trained on the actual Monte Carlo corrupted speech data where y_t^l was given in equation (3.13). The mean and standard deviation of the clean speech were 10 and 6 respectively, the variance of the noise 1, and the mean of the noise adjusted to achieve the SNR indicated. The cross covariance between the clean and corrupted speech is also shown along with the first-order VTS estimated values. The VTS compensated mean appears quite accurate compared to the numerical result. The variance however is not as well approximated; this is similar to previous results [2, 106]. The cross-covariance between the clean and corrupted speech is also only roughly approximated especially in the lower SNRs. This demonstrates the limitations of the VTS approximation of the corrupted speech especially in lower SNR.

5.4.1 The Clean Speech Class Model

The previous section demonstrated how the complete joint distribution for a model class r may be derived given a noise model and clean speech model; this clean speech model has not been discussed in detail. For the joint distribution given in equation (5.33), a model of the clean speech $\mathcal{N}\left(\boldsymbol{\mu}_s^{(r)}, \boldsymbol{\Sigma}_s^{(r)}\right)$ is necessary to compute the rest of the distribution parameters. There are three different approaches to deriving this Gaussian model of the clean speech. As shown in equation (5.45), $\boldsymbol{\Sigma}_s^{(r)}$ may take a block-diagonal form. Alternatively, it may be assumed to be diagonal, however this may be a poor approximation because

$$\text{diag}\left\{\boldsymbol{J}_x^{(r)}\boldsymbol{\Sigma}_{x,\mathsf{diag}}^{(r)}\boldsymbol{J}_x^{(r)\mathsf{T}}\right\} \neq \text{diag}\left\{\boldsymbol{J}_x^{(r)}\boldsymbol{\Sigma}_{x,\mathsf{full}}^{(r)}\boldsymbol{J}_x^{(r)\mathsf{T}}\right\} \tag{5.52}$$

where $\boldsymbol{\Sigma}_{x,\text{diag}}^{(r)}$ is the diagonal variance of the static clean speech and $\boldsymbol{\Sigma}_{x,\text{full}}^{(r)}$ the full covariance version. Since the Jacobian matrix $\boldsymbol{J}_x$ is full, it may be helpful to use a full covariance matrix for each block $\boldsymbol{\Sigma}_x^{(r)}$, $\boldsymbol{\Sigma}_{\Delta x}^{(r)}$ and $\boldsymbol{\Sigma}_{\Delta^2 x}^{(r)}$ that form $\boldsymbol{\Sigma}_s^{(r)}$. There are two ways to estimate these matrices. The first is to estimate a full covariance version of the acoustic model using the same alignments as the diagonal, e.g. through SPR, so that each component has a full $\boldsymbol{\Sigma}_{s,\text{full}}^{(m)}$ (although only block-diagonal matrices are needed). The mean $\boldsymbol{\mu}_s^{(r)}$ and full covariance matrix $\boldsymbol{\Sigma}_s^{(r)}$ for class r may then be given by

$$\boldsymbol{\mu}_s^{(r)} = \frac{1}{\gamma_s^{(r)}} \sum_{m \in r} \gamma_s^{(m)} \boldsymbol{\mu}_s^{(m)} \tag{5.53}$$

$$\boldsymbol{\Sigma}_s^{(r)} = \frac{1}{\gamma_s^{(r)}} \sum_{m \in r} \gamma_s^{(m)} \left(\boldsymbol{\Sigma}_{s,\text{full}}^{(m)} + \boldsymbol{\mu}_s^{(m)} \boldsymbol{\mu}_s^{(m)\mathsf{T}} \right) - \boldsymbol{\mu}_s^{(r)} \boldsymbol{\mu}_s^{(r)\mathsf{T}} \tag{5.54}$$

$$\gamma_s^{(m)} = \sum_{t=1}^{T} \gamma_{s,t}^{(m)} \tag{5.55}$$

$$\gamma_s^{(r)} = \sum_{m \in r} \gamma_s^{(m)} \tag{5.56}$$

where the component posterior $\gamma_{s,t}^{(m)}$ may be computed from the state $\gamma_{s,t}^{(j)}$, $\gamma_{s,t}^{(m)} = \gamma_{s,t}^{(j)} c^{(m)}$, and $c^{(m)}$ denotes the component prior. Since most HMM-based recognisers use diagonal model covariances, it is cumbersome to require a full covariance model. An approximation is to use the standard diagonal model variances, i.e. replace $\boldsymbol{\Sigma}_{s,\text{full}}^{(m)}$ with $\boldsymbol{\Sigma}_s^{(m)}$ in equation (5.54). For low numbers of classes R compared to the number of model components M in the acoustic model, such that there are many model components per class, this should be a good approximation since the between component variance should dominate over the component variance—this is approach taken in this work. Hence a full matrix clean speech class variance $\boldsymbol{\Sigma}_s^{(r)}$ may be estimated from a standard diagonal variance clean speech acoustic model, provided the frame/state alignments are available.

5.5 Comparing JUD with VTS compensation

Increasing the number of classes R to equal the number of model components M, using a diagonal acoustic model variance approximation, is equivalent to VTS model compensation of each individual acoustic model component. This is clear when the component corrupted speech likelihood in equation (5.34) is re-expressed as

$$\begin{aligned} \mathrm{p}(\boldsymbol{o}_t|m;\check{\mathcal{M}}) &= |\boldsymbol{A}^{(r_m)}| \mathcal{N}\left(\boldsymbol{A}^{(r_m)} \boldsymbol{o}_t + \boldsymbol{b}^{(r_m)}; \boldsymbol{\mu}_s^{(m)}, \boldsymbol{\Sigma}_s^{(m)} + \boldsymbol{\Sigma}_{\mathsf{b}}^{(r_m)} \right) \\ &= \mathcal{N}\left(\boldsymbol{o}_t; \boldsymbol{A}^{(r_m)\text{-}1} \left(\boldsymbol{\mu}_s^{(m)} - \boldsymbol{\mu}_s^{(r_m)} \right) + \boldsymbol{\mu}_o^{(r_m)}, \boldsymbol{A}^{(r_m)\text{-}1} \left(\boldsymbol{\Sigma}_s^{(m)} - \boldsymbol{\Sigma}_s^{(r_m)} \right) \boldsymbol{A}^{(r_m)\text{-}\mathsf{T}} + \boldsymbol{\Sigma}_o^{(r_m)} \right) \end{aligned} \tag{5.57}$$

If $M = R$ then there is a one-to-one mapping of r_m to m, hence $\boldsymbol{\mu}_s^{(m)} = \boldsymbol{\mu}_s^{(r_m)}$, and $\boldsymbol{\Sigma}_s^{(m)} = \boldsymbol{\Sigma}_s^{(r_m)}$, making the differences between these terms zero. Therefore from equation (5.57)

$$\mathrm{p}(\boldsymbol{o}_t|m;\check{\mathcal{M}}) = \mathcal{N}\left(\boldsymbol{o}_t; \boldsymbol{\mu}_o^{(r_m)}, \boldsymbol{\Sigma}_o^{(r_m)} \right) = \mathcal{N}\left(\boldsymbol{o}_t; \boldsymbol{\mu}_o^{(m)}, \boldsymbol{\Sigma}_o^{(m)} \right) \tag{5.58}$$

where $\boldsymbol{\mu}_o^{(r_m)}$ and $\boldsymbol{\Sigma}_o^{(r_m)}$ are from the joint distribution derived using VTS model compensation. In general, when $R = M$, M-Joint converges to whatever model compensation technique was

used to derive the joint distribution needed to the compute the transformation parameters. For example if SPR was used to derive the joint distribution for each M-Joint tranform, then when $R = M$ M-Joint is equivalent to directly using SPR to estimate each compensated model component mean and variance; the same applies if instead VTS or PMC are used to predict the joint distribution. In comparison, front-end uncertainty decoding forms like SPLICEU and FE-Joint do not have this characteristic since all model components are affected by the same uncertainty bias chosen by the front-end.

This convergence of M-Joint, when $R = M$, to the form of model-based compensation that was used to derive the joint distribtuion is a very useful property. It allows a flexibility in controlling the computational cost of the M-Joint scheme by adjusting the number of model classes R. Using a VTS approximation to compensate each acoustic model component, is equivalent to deriving a corrupted speech conditional distribution for each acoustic model component of the clean speech models in equation (4.47); this involves computing and applying two Jacobian matrices, $\boldsymbol{J}_x^{(m)}$ and $\boldsymbol{J}_z^{(m)}$, for *each component*, with both operations costing $\mathcal{O}(MD_s^2)$ where D_s is the number of static features. In contrast, M-Joint transforms are estimated *per class* r such that that $\boldsymbol{J}_x^{(r)}$ and $\boldsymbol{J}_z^{(r)}$ are shared over similar components. This sharing of the cost of computing the joint distribution is far cheaper at $\mathcal{O}(RD_s^2)$ if the number of classes R is much smaller than the number of components M. The compensation itself is also more efficient where only R feature updates are computed rather than updating all M component means. The model variance update is simpler, with a single vector addition, rather than several matrix multiplications and an add necessary for VTS compensation.

By using a VTS approximation to generate the joint distribution from models of the clean speech and noise, the associated JUD transform compensates precisely for noise. However, the joint distribution may thought of as a general statistical model of the relationship between the speech seen in training $\boldsymbol{S}$ and the observed speech $\boldsymbol{O}$ in testing. Hence, the joint distribution can model other factors in addition to noise if this is taken into account during its generation. For example, vocal tract length or a feature decorrelating transform could be incorporated in the mismatch function. Furthermore, M-Joint transforms may also compensate multistyle systems for environmental mismatch. In this case, the noise model no longer represents additive and convolutional noise but are simply parameters that generate transforms which reduce the mismatch between the multistyle-trained models and the test conditions.

5.6 Comparing JUD with CMLLR

Like CMLLR, JUD transforms compensate trained systems to more closely match the test environment. The M-Joint likelihood in equation (5.34)

$$\mathrm{p}(\boldsymbol{o}_t|\theta_t;\check{\mathcal{M}}) = \sum_{m\in\theta_t} c^{(m)}|\boldsymbol{A}^{(r_m)}|\mathcal{N}\Big(\boldsymbol{A}^{(r_m)}\boldsymbol{o}_t + \boldsymbol{b}^{(r_m)};\boldsymbol{\mu}_s^{(m)},\boldsymbol{\Sigma}_s^{(m)} + \boldsymbol{\Sigma}_{\mathsf{b}}^{(r_m)}\Big) \tag{5.59}$$

is similar to CMLLR, the likelihood of which may be expressed as

$$\mathrm{p}(\boldsymbol{o}_t|\theta_t;\check{\mathcal{M}}) = \sum_{m\in\theta_t} c^{(m)}|\boldsymbol{A}^{(r_m)}|\mathcal{N}\left(\boldsymbol{A}^{(r_m)}\boldsymbol{o}_t + \boldsymbol{b}^{(r_m)};\boldsymbol{\mu}_s^{(m)},\boldsymbol{\Sigma}_s^{(m)}\right) \tag{5.60}$$

as given previously in equation (2.62). Both have an affine transform of the features, however with M-Joint compensation there is an additional variance bias $\boldsymbol{\Sigma}_{\mathsf{b}}^{(r_m)}$. An efficient implementation of CMLLR uses multiple parallel features for each regression class to avoid changing

the model parameters; however with M-Joint, the model variances and cached normalisation terms must be updated. This makes M-Joint less efficient than CMLLR. The key differences between standard CMLLR and model-based JUD schemes may be summarised as

- in the Gaussian evaluation, the M-Joint has a variance offset, the "uncertainty", that CMLLR does not have,
- M-Joint is highly restricted to only compensating for what the mismatch function allows, e.g. environmental noise, whereas CMLLR transforms have a much larger number of free parameters, hence,
- CMLLR specifically requires corrupted speech adaptation data, whilst M-Joint only needs a noise model which may be estimated from the background.

The noise model itself has relatively few parameters and may be estimated on very little data; in Kim et al. [80] only a few frames of noisy data are necessary to train an accurate model. With CMLLR the amount of training data required scales with the number and complexity of transforms used. In comparison, the number of M-Joint transforms that may be used is independent of the amount of adaptation data available and is more of a function of available computational resources. Furthermore, if the noise is stationary it may be estimated in advance, in the background before the onset of speech, whereas CMLLR requires actual corrupted speech data.

Scheme	Type	# Free Parameters
CMLLR	Adaptive	$R(\underbrace{D}_{\boldsymbol{A}^{(r)}} + \underbrace{D}_{\boldsymbol{b}^{(r)}})$
SPLICE	Feature-based	$K(\underbrace{2D+1}_{\mathrm{p}(\boldsymbol{o}_t\|\mathcal{M})} + \underbrace{D}_{\check{\boldsymbol{\mu}}^{(k^*)}})$
SPLICEU	Feature-based	$K(\underbrace{2D+1}_{\mathrm{p}(\boldsymbol{o}_t\|\check{\mathcal{M}})} + \underbrace{D}_{\check{\boldsymbol{\mu}}^{(k^*)}} + \underbrace{D}_{\check{\boldsymbol{\Sigma}}^{(k^*)}})$
M-Joint VTS	Predictive, Model-based	$\underbrace{D_s}_{\boldsymbol{\mu}_z} + \underbrace{D}_{\boldsymbol{\Sigma}_n} + \underbrace{D_s}_{\boldsymbol{\mu}_h}$

Table 5.1: Number of free parameters to estimate for diagonal forms of various noise compensation schemes.

Table 5.1 compares the number of free parameters that need to be estimated for a variety of noise compensation schemes. The advantage of predictive schemes is clear: the number of free parameters is low and fixed to the model of the noise, while for adaptive schemes it typically varies with the number of transforms R. Hence predictive compensation can be more effective with less training data than adaptive forms. The SPLICE schemes described in sections 4.3.2 and 4.5.2.1 are estimated using stereo data[1], hence the number of free parameters varies according to the number of front-end components K. The model-based JUD form, being a predictive technique, has a low number of free parameters like VTS compensation and PMC. While this restricts the modelling power of M-Joint transforms compared to CMLLR, M-Joint transforms may be estimated on less data than adaptive forms.

[1]This is not an inherent limitation of SPLICE; a predictive form can be devised where a joint distribution may be predicted in the same manner as JUD to derive the clean speech posterior distribution necessary to obtain the SPLICE parameters.

5.7 Computational Cost

An important consideration for noise robustness techniques is the computational cost in terms of parameter storage and additional operations during front-end processing or decoding. The cost of storing parameters may simply be the number of free parameters shown in table 5.1, although in practice M-Joint transforms may be pre-computed and stored. The additional processing overhead incurred by various noise robustness techniques is summarised in table 5.2. Front-end enhancement schemes like SPLICE only incur a front-end processing cost

Compensation Scheme	Front-end Cost	Compensation Cost
Feature Enhancement	$\mathcal{O}(DTK)$	None
Front-end Uncertainty	$\mathcal{O}(DTK)$	$\mathcal{O}(MDT)$
Model-based Uncertainty	$\mathcal{O}(DTR)$	$\mathcal{O}(MD)$
Model-based Forms	None	$\mathcal{O}(MD_s^3)$

D–# of feature dimensions R–# of acoustic model classes
T–# of frames M–# of acoustic model components
K–# of front-end GMM components

Table 5.2: Computational cost for diagonal forms of different noise compensation schemes.

that scales with the complexity of the front-end model. Front-end uncertainty schemes, like observation uncertainty forms, SPLICEU or FE-Joint, in addition to the front-end processing cost, expand the model variances with the uncertainty variance bias. Furthermore, this requires re-computation of the Gaussian normalisation term that is normally cached. Thus each Gaussian evaluation now requires $\mathcal{O}\{5D\}$ operations compared to $\mathcal{O}\{3D\}$ for standard decoding. Assuming that likelihood calculations typically account for 50% of the processing load, this corresponds to an overall system load increase of 33% to apply an uncertainty bias on the model variances—this was confirmed experimentally [4].

Since using M-Joint transforms has the same form as the other uncertainty schemes, it also has the same cost. Unlike the front-end forms however, the uncertainty variance may be cached if the noise environment is stationary, greatly improving speed since now only front-end processing is required. This makes it similar to pure model-based forms like PMC and VTS compensation, although the acoustic model update cost is much cheaper. For example, VTS compensation has a cost of $\mathcal{O}\{MD_s^3\}$ due to the block matrix-block matrix multiplication necessary for the variance update. In contrast, M-Joint, with diagonal transforms, has a processing cost linear with the number of dimensions. Therefore M-Joint transforms are effective for changing environments when the acoustic models need frequent updating compared to PMC or VTS compensation; moreover, M-Joint can be executed without model update caching at a similar cost to some front-end uncertainty schemes compared to pure model-based schemes where it would be prohibitively expensive to do so.

5.8 Predictive CMLLR

A significant cost in M-Joint compensation is applying this uncertainty variance bias; when the joint distribution is full, this results in full transforms, and expensive full covariance

decoding is required. A "predictive" form of CMLLR (PCMLLR) can be derived such that an affine transform of the features may be computed from models of the clean speech and noise, without large amounts of adaptation data [48]. This gives an approximate uncertainty decoding form where no model variance bias is applied during decoding and hence a more complex structure to the joint distribution may be used with the only additional cost of being in the front-end.

PCMLLR transforms are estimated using "predicted" statistics gathered in a feature-space transformed by the M-Joint feature transform

$$\tilde{\boldsymbol{o}}_t = \boldsymbol{A}^{(r)}\boldsymbol{o}_t + \boldsymbol{b}^{(r)} \tag{5.61}$$

such that

$$\mathcal{E}\{\tilde{\boldsymbol{o}}_t | m \in r\} = \boldsymbol{\mu}_s^{(m)} \tag{5.62}$$

$$\mathcal{E}\left\{\tilde{\boldsymbol{o}}_t \tilde{\boldsymbol{o}}_t^{\mathsf{T}} | m \in r\right\} = \boldsymbol{\Sigma}_s^{(m)} + \boldsymbol{\Sigma}_{\mathsf{b}}^{(r_m)} + \boldsymbol{\mu}_s^{(m)} \boldsymbol{\mu}_s^{(m)\mathsf{T}} \tag{5.63}$$

from equation (5.34). In equations (2.65) and (2.66), which are necessary for estimating the CMLLR transform, the statistics can then be replaced with their predicted values

$$\begin{aligned}
\boldsymbol{G}_i^{(r)} &\approx \sum_{m \in r} \frac{1}{\sigma_{s,i}^{(m)2}} \gamma_s^{(m)} \begin{bmatrix} 1 & \mathcal{E}\{\tilde{\boldsymbol{o}}_t | r\}^{\mathsf{T}} \\ \mathcal{E}\{\tilde{\boldsymbol{o}}_t | r\} & \mathcal{E}\left\{\tilde{\boldsymbol{o}}_t \tilde{\boldsymbol{o}}_t^{\mathsf{T}} | r\right\} \end{bmatrix} \\
&= \sum_{m \in r} \frac{1}{\sigma_{s,i}^{(m)2}} \gamma_s^{(m)} \begin{bmatrix} 1 & \boldsymbol{\mu}_s^{(m)\mathsf{T}} \\ \boldsymbol{\mu}_s^{(m)} & \boldsymbol{\Sigma}_s^{(m)} + \boldsymbol{\Sigma}_{\mathsf{b}}^{(r)} + \boldsymbol{\mu}_s^{(m)} \boldsymbol{\mu}_s^{(m)\mathsf{T}} \end{bmatrix}
\end{aligned} \tag{5.64}$$

$$\boldsymbol{k}_i^{(r)} \approx \sum_{m \in r} \frac{1}{\sigma_{s,i}^{(m)2}} \gamma_s^{(m)} \begin{bmatrix} 1 \\ \mathcal{E}\{\tilde{\boldsymbol{o}}_t | r\} \end{bmatrix} = \sum_{m \in r} \frac{1}{\sigma_{s,i}^{(m)2}} \gamma_s^{(m)} \begin{bmatrix} 1 \\ \boldsymbol{\mu}_s^{(m)} \end{bmatrix} \tag{5.65}$$

Estimating a PCMLLR transform, $\boldsymbol{A}_{\mathsf{pcmllr}}^{(r)}$ and $\boldsymbol{b}_{\mathsf{pcmllr}}^{(r)}$, can be made very efficient since

$$\begin{aligned}
\boldsymbol{G}_i^{(r)} &= \sum_{m \in r} \frac{1}{\sigma_{s,i}^{(m)2}} \gamma_s^{(m)} \begin{bmatrix} 1 & \boldsymbol{\mu}_s^{(m)\mathsf{T}} \\ \boldsymbol{\mu}_s^{(m)} & \boldsymbol{\Sigma}_s^{(m)} + \boldsymbol{\mu}_s^{(m)} \boldsymbol{\mu}_s^{(m)\mathsf{T}} \end{bmatrix} + \sum_{m \in r} \frac{1}{\sigma_{s,i}^{(m)2}} \gamma_s^{(m)} \begin{bmatrix} 0 & 0 \\ 0 & \boldsymbol{\Sigma}_{\mathsf{b}}^{(r)} \end{bmatrix} \\
&= \underbrace{\sum_{m \in r} \frac{1}{\sigma_{s,i}^{(m)2}} \gamma_s^{(m)} \begin{bmatrix} 1 & \boldsymbol{\mu}_s^{(m)\mathsf{T}} \\ \boldsymbol{\mu}_s^{(m)} & \boldsymbol{\Sigma}_s^{(m)} + \boldsymbol{\mu}_s^{(m)} \boldsymbol{\mu}_s^{(m)\mathsf{T}} \end{bmatrix}}_{\text{cached}} + \begin{bmatrix} 0 & 0 \\ 0 & \boldsymbol{\Sigma}_{\mathsf{b}}^{(r)} \end{bmatrix} \underbrace{\sum_{m \in r} \frac{\gamma_s^{(m)}}{\sigma_{s,i}^{(m)2}}}_{\text{cached}}
\end{aligned} \tag{5.66}$$

The first summation in the right-hand side of this equation is comprised entirely of clean speech statistics and can be cached as well as the mean update. Also, since the uncertainty variance bias is fixed for a class, it may be taken out of the second summation over all components; this summation can also be cached. Hence only the uncertainty variance bias term is dependent on the noise environment whilst the rest of the statistics may be pre-computed and stored per regression class as noted in equation (5.66). Because it is out of the summation, matrix $\boldsymbol{G}_i^{(r)}$ can be rapidly computed for changing noise since the number of classes, R, is usually small compared to number of components, M.

The overall decoding likelihood for PCMLLR decoding is then given by

$$\mathrm{p}(\boldsymbol{o}_t | \theta_t; \check{\mathcal{M}}) = \sum_{m \in \theta_t} |\boldsymbol{A}_{\mathsf{pcmllr}}^{(r_m)}| |\boldsymbol{A}^{(r_m)}| \mathcal{N}\left(\boldsymbol{A}_{\mathsf{pcmllr}}^{(r_m)} \tilde{\boldsymbol{o}}_t + \boldsymbol{b}_{\mathsf{pcmllr}}^{(r_m)}; \boldsymbol{\mu}_s^{(m)}, \boldsymbol{\Sigma}_s^{(m)}\right) \tag{5.67}$$

where r_m gives the regression class r that component m belongs to and $\tilde{\boldsymbol{o}}_t$ is defined by equation (5.61). The two affine transformations may be combined into a single one. The structure of the transformation depends on the form of the uncertainty variance bias $\boldsymbol{\Sigma}_{\mathsf{b}}^{(r)}$, e.g. if it is block-diagonal then the PCMLLR transform will also be block-diagonal. In contrast to standard CMLLR, PCMLLR is a predictive form where the affine transformation is efficiently derived from combining models of the speech and noise through a M-Joint transform. Thus like M-Joint, the PCMLLR transform may be estimated with little data, but unlike M-Joint there is no variance bias offset. Although decoding is more efficient with PCMLLR, without the variance bias term it is less effective. Hence PCMLLR combines the efficient compensation of CMLLR, due to a simple affine transformation of the features, with the predictive nature of M-Joint compensation. The extra computation efficiency is gained at the expense of the effectiveness of the uncertainty variance bias.

5.9 Non-Gaussian Approximations

Using a single Gaussian, due to equation (5.13) or (5.32), to model the corrupted speech conditional distribution may not be optimal. Alternate forms such as the Weibull or gamma distribution appear more representative than the Gaussian distribution—the mode is more accurate and the asymmetry better models the long right tail of the conditional distribution.

The formula for the gamma distribution is

$$\mathrm{p}(y_t^l|x_t^l) = \frac{\left(\frac{y_t^l - x_t^l}{b}\right)^{(a-1)} e^{\left(-\frac{y_t^l - x_t^l}{b}\right)}}{b\Gamma(a)} \tag{5.68}$$

where

$$\Gamma(a) = \int_0^\infty t^{(a-1)} e^{-t} \mathrm{d}t \tag{5.69}$$

and a and b are parameters of the distribution and y_t^l and x_t^l are log-spectral domain noisy and clean speech variables.

The formula for the Weibull distribution is

$$\mathrm{p}(y_t^l|x_t^l) = cd(y_t^l - x_t^l)^{(d-1)} e^{(-c(y_t^l - x_t^l)^d)} \tag{5.70}$$

where c and d are parameters of the distribution. Figure 5.8 compares different parametric distributions for the corrupted speech conditional distribution generated for different values of the clean speech mean and a fixed Gaussian noise source. This uses the same log-spectral model of the environment given in equation (3.13). When the speech energy is high compared to the noise, the corrupted speech conditional distribution is deterministic, and when the energy is low relative to the noise, it is Gaussian. When the speech is not completely subsumed by the noise, as in sub-figures b, c and d, the conditional distribution is distinctly non-Gaussian. It is clear the skewed, non-symmetrical Weibull and Gamma distributions are better forms for representing the corrupted speech conditional distribution than the Gaussian.

Although 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 corrupted speech conditional distribution, analytic solutions of the integral in equation (4.47) are difficult to derive if the conditional distribution takes these forms. This

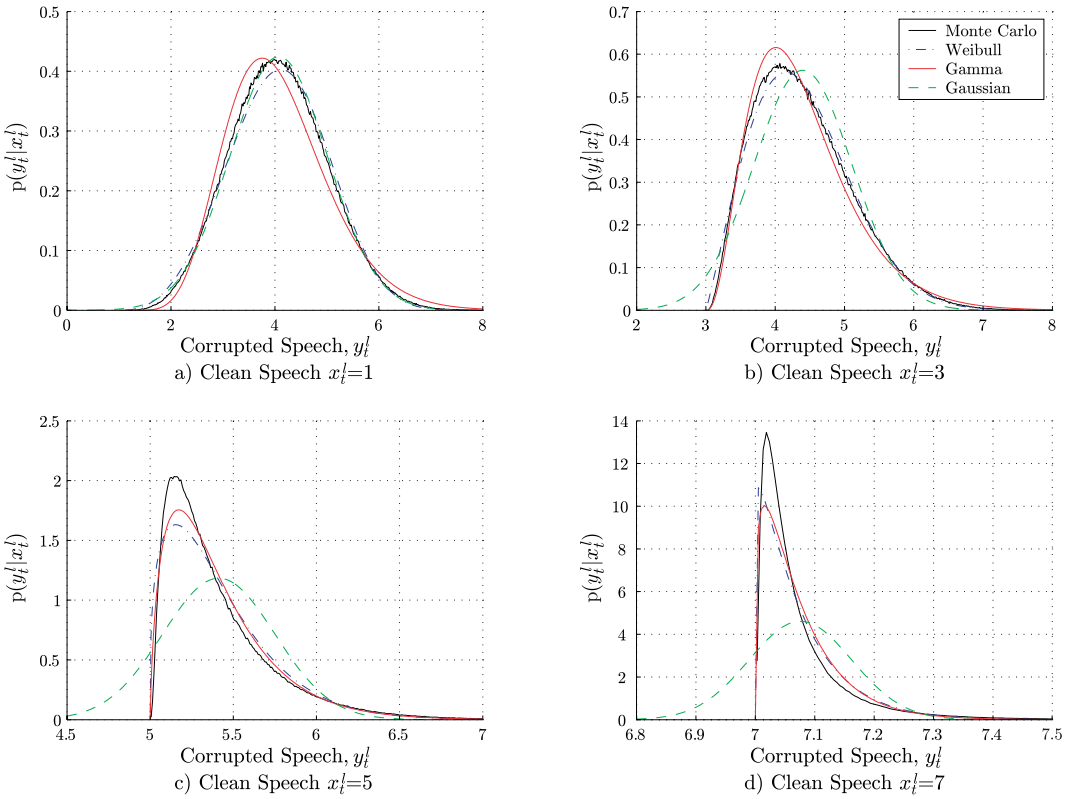


Figure 5.8: Corrupted speech conditional distribution with additive noise $z_t^l \sim \mathcal{N}(4, 1)$ in log-spectral domain. Various distributions are fitted to the simulated data.

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. The natural conjugate distribution for a Gaussian is another Gaussian.

Instead of using only a single Gaussian to represent the corrupted speech conditional distribution, a GMM may be used. For example, to improve the model-based JUD form from equation (5.32) such that each class r now has K_r components instead of just one

$$\mathrm{p}(\boldsymbol{o}_t|\boldsymbol{s}_t, r; \check{\mathcal{M}}) \approx \sum_{k=1}^{K_r} c^{(k)} \mathcal{N}(\boldsymbol{o}_t; f_\mu(\boldsymbol{s}_t, r, k), f_\Sigma(\boldsymbol{s}_t, r, k)) \tag{5.71}$$

This however would multiply the overall number of recognition Gaussians by K_r. Nevertheless, a small number of components may sufficiently improve upon a single Gaussian. Figure 5.9 compares a 2-component GMM to a single Gaussian model of the corrupted speech condition in sub-plot 'd' from figure 5.8—clearly, the GMM is a better fit than a single Gaussian in this situation. For acoustic model components that represent low energy speech and are more affected by additive noise, a single Gaussian should be sufficient, so long that the noise is well modelled by a single Gaussian. Alternatively, lower weighted components of the mixture may be pruned; in figure 5.9 the left component represents the majority of the probability mass. Hence modelling the conditional distribution for each class using a GMM, with a variable number of Gaussians, could possibly give improved modelling accuracy with a limited increase in the overall number of recognition components.

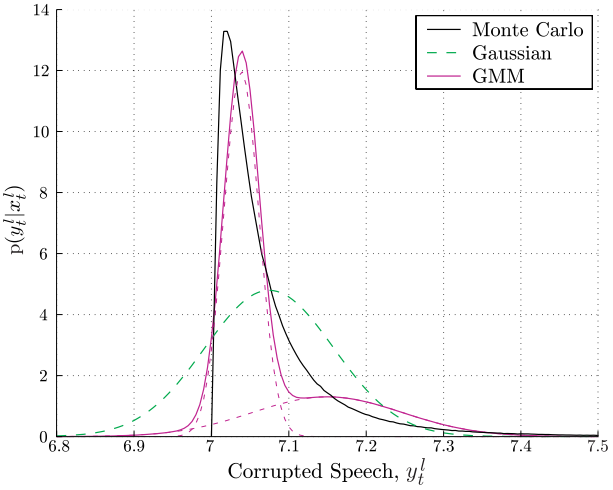


Figure 5.9: Corrupted speech conditional distribution with clean speech $x_t = 7$, additive noise $z_t^l \sim \mathcal{N}(4, 1)$. Single Gaussian and 2-component GMM fitted (components dotted).

5.10 Summary

In this chapter, uncertainty decoding was formally introduced in the context of the noise robust ASR framework described in section 4.1. Given that the form of the speech prior in recognition systems is typically Gaussian, the main research focus for uncertainty decoding is to find tractable and accurate forms of the corrupted speech conditional distribution. Joint uncertainty decoding (JUD) approximates the joint distribution between the clean training conditions and the noisy test conditions with a Gaussian leading to a Gaussian corrupted speech conditional distribution. While this latter distribution can be highly skewed, modelling it with a Gaussian provides a Gaussian result when convolved with the Gaussian speech prior. Two forms of JUD were presented: front-end and model-based. The front-end JUD form has less restrictive approximations than SPLICEU that naturally allow JUD to have more powerful block-diagonal or full matrix transforms. It was demonstrated that front-end uncertainty decoding forms can potentially have problems in low SNR, since all acoustic model components may be transformed to the noise model. Model-based forms, like model-based JUD (M-Joint) and VTS, avoid this since different transforms for different components reflect that noise affects each recognition component differently. The M-Joint form is similar to CMLLR, but with the addition of a model variance bias term. Another difference is that the JUD transform may be predicted using a low number of parameters representing the speech and noise models. M-Joint transforms, for any number of classes, may be estimated on a limited amount of adaptation data, which need not include speech, sufficient to estimate the noise model; in contrast, CMLLR requires adaptation data, containing corrupted speech, in amounts proportional to the number and form, e.g. diagonal or full, of the transformations, to provide robust estimates. Thus the number of M-Joint transforms may be chosen to achieve the desired computational profile; at the limit, when the number of transforms equals the number of model components, it is the equivalent of model-based VTS compensation. The M-Joint transform though is more efficient to apply at run-time than the VTS form. These aspects make JUD an attractive, noise robustness technique.

CHAPTER 6
Noise Model Estimation

Predictive forms of noise compensation, such as M-Joint transforms and VTS acoustic model compensation, require a model of the corrupting noise. Often, an acoustic model of the additive noise is estimated from audio segments of the test environment that do not contain speech. However, combining this model with a ML-trained acoustic speech model does not give a corrupted speech model that maximises the likelihood of the test data. Nor is the model appropriate for multistyle-trained acoustic models. Hence this chapter discusses ML methods for jointly estimating a model of the additive and convolutional noise. Different methods are outlined, tailored for either VTS or M-Joint compensation. While some gains were obtained with multi-component noise models for both feature compensation [36, 135] and acoustic model compensation [86], here only a single Gaussian model of the additive noise is considered.

6.1 Maximum Likelihood Noise Model Estimation

An ML form of noise model estimation seeks to find a noise model, which when used to compensate the acoustic model, gives a noisy speech model that maximises the likelihood of data from the noise-corrupted environment. The noise model is denoted by $\mathcal{M}_n = \{\boldsymbol{\mu}_n, \boldsymbol{\mu}_h, \boldsymbol{\Sigma}_n\}$, where $\boldsymbol{\mu}_h$ represents the channel noise and $\boldsymbol{\mu}_n$ and $\boldsymbol{\Sigma}_n$ the additive noise mean and variance. The noise model should maximise the likelihood of observed noisy data

$$\hat{\mathcal{M}}_n = \underset{\mathcal{M}_n}{\operatorname{argmax}}\, \mathrm{p}(\boldsymbol{O}|\mathcal{W}_{\mathsf{h}}; \mathcal{M}, \mathcal{M}_n) \tag{6.1}$$

given a hypothesis $\mathcal{W}_{\mathsf{h}}$ and the acoustic model $\mathcal{M}$, which is compensated using VTS or M-Joint compensation using $\mathcal{M}_n$. Directly finding the noise model that optimises equation (6.1)

is difficult for either VTS or M-Joint compensation because of the hidden state sequence. Hence an iterative EM approach is used. The auxiliary function is

$$\begin{aligned}\mathcal{Q}(\mathcal{M}_n;\hat{\mathcal{M}}_n) &= \mathbb{E}_{\hat{\mathcal{M}}}\Big[\log \mathrm{p}(\boldsymbol{O},\boldsymbol{M};\mathcal{M},\hat{\mathcal{M}}_n)\Big] \\ &= \sum_{t=1}^{T}\sum_{m=1}^{M}\gamma_{o,t}^{(m)}\log \mathrm{p}(\boldsymbol{o}_t|m;\mathcal{M},\hat{\mathcal{M}}_n) \end{aligned} \tag{6.2}$$

where $\gamma_{o,t}^{(m)}$ is the posterior probability of component m at time t given the complete data set, i.e. $\mathrm{P}(m_t = m|\boldsymbol{O},\mathcal{W}_{\mathsf{h}};\mathcal{M},\hat{\mathcal{M}}_n)$. Recall that m uniquely indexes any component in the HMM for all states, thus the state index is unnecessary. The set of all possible hidden component/state sequences $\boldsymbol{M}$ for the complete data set may be computed from a hypothesised transcription $\mathcal{W}_{\mathsf{h}}$ produced from a decoding run using a compensated model $\mathcal{M}$ with the initial noise estimate $\mathcal{M}_n$.

Figure 6.1 depicts the general ML noise model estimation procedure. An initial noise model is necessary to begin: this may be a "quiet" noise model, where the convolutional noise and additive noise are effectively zero; or, it may be estimated from the first few frames of speech. In the expectation step, this model is used to compensate the acoustic model to align the noisy observation data and compute the complete data set. From the complete data set, statistics necessary to perform the maximisation step are gathered. The required statistics and maximisation steps 4 and 5 will differ depending on whether an ML VTS or M-Joint noise

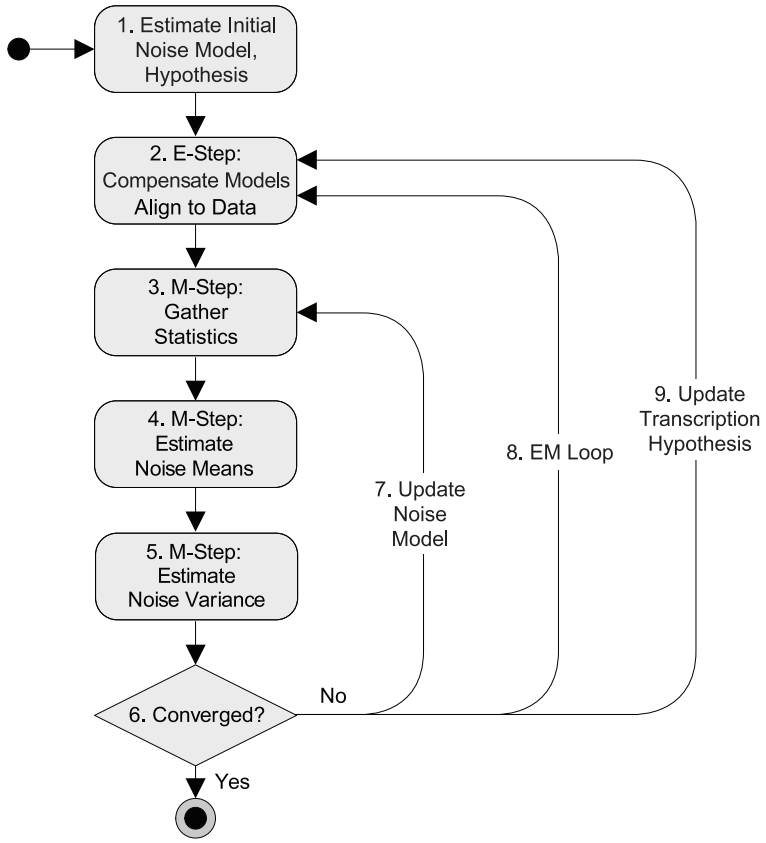


Figure 6.1: EM-based ML noise model estimation procedure.

model is being estimated. There are many points in the process where iterations may take place. The statistics gathered may require an expansion point for VTS approximation and thus benefit from an updated noise model shown in loop 7; this may be repeated without conducting addition passes over the observation data. Loop 8 begins another EM iteration where the updated noise model is used to recompute the complete data set for the initial transcription hypothesis. Compared to loop 7, loop 8 requires a pass over the observation data. Alternatively, as indicated by loop 9, this hypothesis may be improved by performing another decoding pass over the test data using acoustic models compensated with the updated noise model; this may be important in noisier environments where the initial hypothesis is poor. Convergence occurs when the auxiliary function fails to increase by a certain threshold.

There are many aspects that make ML noise model estimation an attractive method to specify the noise environment. Apart from the consistency of producing ML-compensated models from ML-trained clean models, a noise model can also be estimated for multistyle-trained acoustic models. In this case, the estimated noise model is especially no longer a model of acoustic noise, but rather a low-dimensional set of parameters that best reduce the mismatch between the acoustic models and test conditions in a ML sense. Moreover, any deficiencies in the mismatch function, for example with the VTS approximation as was shown in figure 5.7, may be partially alleviated by the noise estimation process. Another benefit is that the noise model may be re-estimated while speech is spoken, not only during non-speech regions. This allows for noise model adaptation during long speech segments where the environment may evolve. In this scenario, JUD is particularly relevant since it can rapidly compensate models compared to other techniques such as VTS. Also the system does not require a robust speech detector, whereas one is necessary if the noise model is estimated from background regions. Furthermore, separating speech from noise becomes more difficult as the SNR decreases. For these reasons, the extra complexity of an ML noise estimation procedure is explored for both VTS and M-Joint compensation forms.

6.2 VTS Noise Model Estimation

An iterative solution for updating the static means of the additive and convolutional noise was given in Moreno [106]—this is modified to operate in the same domain many speech recognisers operate: the cepstral domain, as described in section 4.6. This form of noise estimation scheme is designed to give an ML noise model for a system involving only static feature coefficients. The majority of recognisers will use delta and acceleration coefficients. Hence, the component posteriors should be computed using these coefficients and a complete speech acoustic model compensated using VTS. The complete auxiliary function should always be checked for improvement at every iteration of the noise model, as opposed to only optimising with the static auxiliary function given in equation (4.67). From equation (6.2), the auxiliary

function for noise model estimation with VTS compensation is

$$\begin{aligned}\mathcal{Q}_{\text{vts}}(\mathcal{M}_n;\hat{\mathcal{M}}_n) &= \sum_{t=1}^{T}\sum_{m=1}^{M}\gamma_{o,t}^{(m)}\log \mathrm{p}(\boldsymbol{o}_t|m;\mathcal{M},\hat{\mathcal{M}}_n)\\ &= -\frac{1}{2}\sum_{t=1}^{T}\sum_{m=1}^{M}\gamma_{o,t}^{(m)}\Big[\log|\boldsymbol{\Sigma}_{y}^{(m)}| + \big(\boldsymbol{y}_t-\boldsymbol{\mu}_{y}^{(m)}\big)^{\mathsf{T}}\boldsymbol{\Sigma}_{y}^{(m)\text{-}1}\big(\boldsymbol{y}_t-\boldsymbol{\mu}_{y}^{(m)}\big)\\ &\qquad\qquad \log|\boldsymbol{\Sigma}_{\Delta y}^{(m)}| + \big(\Delta\boldsymbol{y}_t-\boldsymbol{\mu}_{\Delta y}^{(m)}\big)^{\mathsf{T}}\boldsymbol{\Sigma}_{\Delta y}^{(m)\text{-}1}\big(\Delta\boldsymbol{y}_t-\boldsymbol{\mu}_{\Delta y}^{(m)}\big)\\ &\qquad\qquad \log|\boldsymbol{\Sigma}_{\Delta^2 y}^{(m)}| + \big(\Delta^2\boldsymbol{y}_t-\boldsymbol{\mu}_{\Delta^2 y}^{(m)}\big)^{\mathsf{T}}\boldsymbol{\Sigma}_{\Delta^2 y}^{(m)\text{-}1}\big(\Delta^2\boldsymbol{y}_t-\boldsymbol{\mu}_{\Delta^2 y}^{(m)}\big)\Big]\end{aligned} \tag{6.3}$$

where terms independent of the noise are ignored and D is the dimensionality of the feature vector and here equal to $3D_s$—the sum of the number of static, delta and acceleration coefficients. Terms independent of the noise model parameters are omitted. The corrupted speech model parameters are predicted from the clean using VTS compensation outlined previously in section 4.4.3.

Hence, the goal is to estimate the set of noise parameters $\mathcal{M}_n$ that maximise the auxiliary function

$$\mathcal{M}_n^{i+1} = \underset{\hat{\mathcal{M}}_n}{\operatorname{argmax}}\,\mathcal{Q}_{\text{vts}}\big(\mathcal{M}_n^{i};\hat{\mathcal{M}}_n\big) \tag{6.4}$$

The noise means, $\hat{\boldsymbol{\mu}}_n$ and $\hat{\boldsymbol{\mu}}_h$, affect all the terms in the square brackets of equation (6.3) since they are part of the evaluation point of the Jacobian matrices used to derive the corrupted speech parameters. As the noise means and the additive noise variance affect the same terms, these should be maximised serially. However, because the static, delta, and delta-delta additive noise variance parameters affect different terms in equation (6.3), they may be maximised in parallel. Hence, for the maximisation step, first re-estimate the noise model means then use these updated noise means to re-estimate the variances. The means can then be updated again, and so on.

6.2.1 Estimating the Static Noise Means

The static additive noise and channel means may be estimated using the statics fixed point techniques as described in section 4.6. However, this procedure needs to be updated for use with dynamic features. First, the component posteriors are computed using the complete features rather than just the statics. Thus the auxiliary function for estimating the static noise means is

$$\begin{aligned}\mathcal{Q}(\boldsymbol{\mu}_n,\boldsymbol{\mu}_h;\hat{\boldsymbol{\mu}}_z,\hat{\boldsymbol{\mu}}_h) &= \mathbb{E}_{\hat{\mathcal{M}}}\Big[\log \mathrm{p}(\boldsymbol{O},\boldsymbol{M};\mathcal{M},\hat{\mathcal{M}}_n)\Big]\\ &= \sum_{t=1}^{T}\sum_{m=1}^{M}\gamma_{o,t}^{(m)}\log \mathrm{p}(\boldsymbol{y}_t|m;\hat{\boldsymbol{\mu}}_z,\boldsymbol{\Sigma}_z,\hat{\boldsymbol{\mu}}_h)\\ &= \sum_{t=1}^{T}\sum_{m=1}^{M}\gamma_{o,t}^{(m)}\Big\{-\frac{1}{2}\log|\boldsymbol{\Sigma}_{y}^{(m)}| - \frac{1}{2}\big(\boldsymbol{y}_t-\hat{\boldsymbol{\mu}}_{y}^{(m)}\big)^{\mathsf{T}}\boldsymbol{\Sigma}_{y}^{(m)\text{-}1}\big(\boldsymbol{y}_t-\hat{\boldsymbol{\mu}}_{y}^{(m)}\big)\Big\}\end{aligned} \tag{6.5}$$

where recall that $\hat{\boldsymbol{\mu}}_{y}^{(m)}$ is derived using a first-order VTS approximation. Other than the component posterior, the statistics and noise mean updates are otherwise unchanged for this

Initialise $\hat{\eta} = 1$, $\bar{\boldsymbol{\mu}}_n = \boldsymbol{\mu}_n^{i+1}$, $\bar{\boldsymbol{\mu}}_h = \boldsymbol{\mu}_h^{i+1}$
Do
$\eta = \hat{\eta}\alpha$
$\boldsymbol{\mu}_n^{i+1} = \eta\bar{\boldsymbol{\mu}}_n + (1-\eta)\boldsymbol{\mu}_n^i$
$\boldsymbol{\mu}_h^{i+1} = \eta\bar{\boldsymbol{\mu}}_h + (1-\eta)\boldsymbol{\mu}_h^i$
$\hat{\eta} = \eta$
While $\mathcal{Q}_{\mathsf{vts}}(\mathcal{M}_n; \boldsymbol{\mu}_n^i, \boldsymbol{\mu}_h^i) > \mathcal{Q}_{\mathsf{vts}}(\mathcal{M}_n; \boldsymbol{\mu}_n^{i+1}, \boldsymbol{\mu}_h^{i+1})$

Figure 6.2: Noise model estimation back-off procedure.

process compared to the statics fixed point estimation form. The auxiliary function is different from $\mathcal{Q}_{\mathsf{vts}}$, which uses a zero-order VTS approximation during maximisation. Maximising the auxiliary in equation (6.5) does not account for dynamic coefficients and therefore while it may improve $\mathcal{Q}_{\mathsf{vts}}$, may not give the best ML estimates. Nevertheless, this is the procedure that will be used to estimate the noise model means. Also since maximising equation (6.5) only accounts for the static features, the estimated noise means must be checked that they improve the complete auxiliary given by equation (6.3). That is ensure

$$\mathcal{Q}_{\mathsf{vts}}(\mathcal{M}_n; \boldsymbol{\mu}_n^i, \boldsymbol{\mu}_h^i) < \mathcal{Q}_{\mathsf{vts}}(\mathcal{M}_n; \boldsymbol{\mu}_n^{i+1}, \boldsymbol{\mu}_h^{i+1}) \tag{6.6}$$

for every iteration i.

If for an iteration i the auxiliary function $\mathcal{Q}_{\mathsf{vts}}$ does not increase, then the back-off strategy shown in figure 6.2 is applied. This interpolates the update $\{\boldsymbol{\mu}_n^{i+1}, \boldsymbol{\mu}_n^{i+1}\}$ from the overshooting estimate $\{\bar{\boldsymbol{\mu}}_n, \bar{\boldsymbol{\mu}}_h\}$ and the previous estimate $\{\boldsymbol{\mu}_h, \boldsymbol{\mu}_h\}$. As $\hat{\eta}$ becomes smaller, the update becomes closer to the previous estimate until there is no difference, which for this strategy indicates convergence. Setting α to a half produced reasonable results. An example of this back-off process is depicted in figure 6.3 where only a scalar additive noise mean is considered. The first derivative step decreases the auxiliary function value—only after two back-off iterations does the step improve the auxiliary function value. This is where the process terminates.

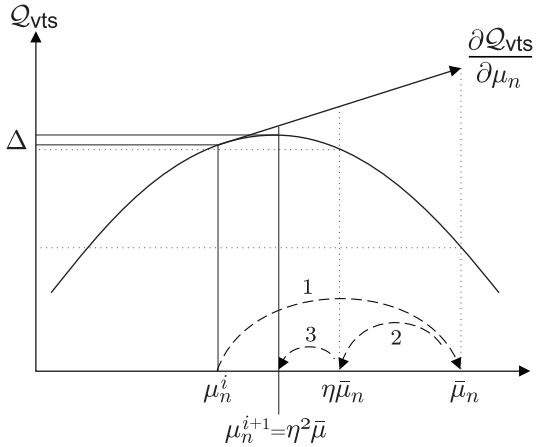


Figure 6.3: Noise model estimation back-off example. The estimate $\bar{\mu}_n$ overshoots the maximum value (1), and thus is backed off twice (2,3). The final update is $\mu_n^{i+1} = \eta^2\bar{\mu}_n$ which improves the auxiliary function by Δ.

6.2.2 Estimating the Additive Noise Variance

It is difficult to obtain a closed form solution for the static additive noise variance that maximises the auxiliary function since the additive noise variance affects all model variances. Hence a simple, iterative, first-order gradient-based optimisation scheme is used to estimate the full complete additive noise variance

$$\hat{\boldsymbol{\Sigma}}_n = \begin{bmatrix} \hat{\boldsymbol{\Sigma}}_z & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \hat{\boldsymbol{\Sigma}}_{\Delta z} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \hat{\boldsymbol{\Sigma}}_{\Delta^2 z} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Sigma}_z & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{\Delta z} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Sigma}_{\Delta^2 z} \end{bmatrix} + \nu \begin{bmatrix} \frac{\partial \mathcal{Q}_{\text{vts}}}{\partial \boldsymbol{\Sigma}_z} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \frac{\partial \mathcal{Q}_{\text{vts}}}{\partial \boldsymbol{\Sigma}_{\Delta z}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \frac{\partial \mathcal{Q}_{\text{vts}}}{\partial \boldsymbol{\Sigma}_{\Delta^2 z}} \end{bmatrix} \tag{6.7}$$

where the additive noise covariance matrix is approximated by a diagonal structure and ν is a scalar learning rate and in this work set to unity. There is no guarantee that the step taken will improve the auxiliary—the step may be too large and significantly overshoot it. Hence, it is important to also back-off the new variance estimate towards the old, as with the means in figure 6.2. The optimisation may end when the auxiliary function fails to increase beyond a threshold or until a certain number of iterations has passed. It was found that a maximum number of iterations, set at 10, was effective. Compared to the statics fixed point estimation of the noise means where iteration occurs by successively improving the VTS expansion point, the first-order approach slowly steps toward the maximum.

The derivative of the auxiliary function, given in equation (6.3), w.r.t. the static additive noise variance is required. This may be expressed as

$$\begin{aligned} \frac{\partial \mathcal{Q}_{\text{vts}}}{\partial \boldsymbol{\Sigma}_z} &= \sum_{t=1}^{T} \sum_{m=1}^{M} \gamma_{o,t}^{(m)} \frac{\partial}{\partial \boldsymbol{\Sigma}_z} \left[\log \mathrm{p}\left(\boldsymbol{o}_t \middle| m; \mathcal{M}, \hat{\mathcal{M}}_n \right) \right] \\ &= \sum_{t=1}^{T} \sum_{m=1}^{M} \gamma_{o,t}^{(m)} \frac{\partial}{\partial \boldsymbol{\Sigma}_z} \left[-\frac{1}{2} \log |\boldsymbol{\Sigma}_y^{(m)}| - \frac{1}{2} \left(\boldsymbol{y}_t - \boldsymbol{\mu}_y^{(m)} \right)^{\mathsf{T}} \boldsymbol{\Sigma}_y^{(m)^{-1}} \left(\boldsymbol{y}_t - \boldsymbol{\mu}_y^{(m)} \right) \right] \end{aligned} \tag{6.8}$$

where terms independent of the static additive noise variance are ignored. As shown in appendix C, by ignoring cross-terms between dimensions, equation (6.8), for dimension i, may be approximated by

$$\frac{\partial \mathcal{Q}_{\text{vts}}}{\partial \sigma_{z,i}^2} \approx -\frac{1}{2} \sum_{m=1}^{M} \sum_{d=1}^{D_s} \left[\boldsymbol{J}_z^{(m)} \right]_{di}^2 \frac{1}{\sigma_{y,d}^{(m)2}} \left\{ \left(1 - \frac{\mu_{y,d}^{(m)2}}{\sigma_{y,d}^{(m)2}} \right) \gamma_o^{(m)} - \frac{p_d^{(m)} - 2 q_d^{(m)} \mu_{y,d}^{(m)}}{\sigma_{y,d}^{(m)2}} \right\} \tag{6.9}$$

where $\left[\boldsymbol{J}_z^{(m)} \right]_{di}^2$ gives the square of the element in row d, column i, of the Jacobian matrix and the sufficient statistics $\boldsymbol{p}^{(m)}$ and $\boldsymbol{q}^{(m)}$ are defined as

$$p_d^{(m)} = \sum_{t=1}^{T} \gamma_{o,t}^{(m)} y_{t,d}^2 \qquad\qquad q_d^{(m)} = \sum_{t=1}^{T} \gamma_{o,t}^{(m)} y_{t,d} \tag{6.10}$$

Since the derivatives of the corrupted speech variances w.r.t. the additive noise variance are all the same for the different blocks, i.e.

$$\begin{aligned} \frac{\partial \boldsymbol{\Sigma}_y^{(m)}}{\partial \sigma_{z,i}^2} &= \frac{\partial \boldsymbol{\Sigma}_{\Delta y}^{(m)}}{\partial \sigma_{\Delta z,i}^2} = \frac{\partial \boldsymbol{\Sigma}_{\Delta^2 y}^{(m)}}{\partial \sigma_{\Delta^2 z,i}^2} \\ &\approx \left[\boldsymbol{J}_z^{(m)} \right]_i \left[\boldsymbol{J}_z^{(m)} \right]_i^{\mathsf{T}} \end{aligned} \tag{6.11}$$

the derivatives of the dynamic corrupted speech variances may be computed in the same manner as the static dimensions. This can be plainly seen in the derivation of equation (C.5) found in appendix C. Thus the auxiliary derivatives for dynamic coefficients of the additive noise variance are given by

$$\begin{aligned}\frac{\partial \mathcal{Q}_{\mathsf{vts}}}{\partial \sigma^2_{\Delta z,i}} &= -\frac{1}{2}\sum_{t=1}^{T}\sum_{m=1}^{M}\gamma_{o,t}^{(m)}\frac{\partial}{\partial \sigma^2_{\Delta z,i}}\Big[\log|\boldsymbol{\Sigma}_{\Delta y}^{(m)}| + \big(\Delta \boldsymbol{y}_t - \boldsymbol{\mu}_{\Delta y}^{(m)}\big)^{\mathsf{T}}\boldsymbol{\Sigma}_{\Delta y}^{(m)\text{-}1}\big(\Delta \boldsymbol{y}_t - \boldsymbol{\mu}_{\Delta y}^{(m)}\big)\Big] \\ &\approx -\frac{1}{2}\sum_{m=1}^{M}\sum_{d=1}^{D_s}\big[\boldsymbol{J}_z^{(m)}\big]_{di}^2\frac{1}{\sigma^{(m)2}_{\Delta y,d}}\left\{\left(1-\frac{\mu^{(m)2}_{\Delta y,d}}{\sigma^{(m)2}_{\Delta y,d}}\right)\gamma_o^{(m)} - \frac{p^{(m)}_{\Delta d} - 2q^{(m)}_{\Delta d}\mu^{(m)}_{\Delta y,d}}{\sigma^{(m)2}_{\Delta y,d}}\right\} \qquad (6.12)\end{aligned}$$

$$\begin{aligned}\frac{\partial \mathcal{Q}_{\mathsf{vts}}}{\partial \sigma^2_{\Delta^2 z,i}} &= -\frac{1}{2}\sum_{t=1}^{T}\sum_{m=1}^{M}\gamma_{o,t}^{(m)}\frac{\partial}{\partial \sigma^2_{\Delta^2 z,i}}\Big[\log|\boldsymbol{\Sigma}_{\Delta^2 y}^{(m)}| + \big(\Delta^2 \boldsymbol{y}_t - \boldsymbol{\mu}_{\Delta^2 y}^{(m)}\big)^{\mathsf{T}}\boldsymbol{\Sigma}_{\Delta^2 y}^{(m)\text{-}1}\big(\Delta^2 \boldsymbol{y}_t - \boldsymbol{\mu}_{\Delta^2 y}^{(m)}\big)\Big] \\ &\approx -\frac{1}{2}\sum_{m=1}^{M}\sum_{d=1}^{D_s}\big[\boldsymbol{J}_z^{(m)}\big]_{di}^2\frac{1}{\sigma^{(m)2}_{\Delta^2 y,d}}\left\{\left(1-\frac{\mu^{(m)2}_{\Delta^2 y,d}}{\sigma^{(m)2}_{\Delta^2 y,d}}\right)\gamma_o^{(m)} - \frac{p^{(m)}_{\Delta^2 d} - 2q^{(m)}_{\Delta^2 d}\mu^{(m)}_{\Delta^2 y,d}}{\sigma^{(m)2}_{\Delta^2 y,d}}\right\} \qquad (6.13)\end{aligned}$$

6.3 M-Joint Noise Model Estimation

The previous section described a noise estimation procedure directed at obtaining an ML noise model presuming a VTS compensation scheme. While the M-Joint form may perform sufficiently well with such noise parameters, they are not optimal in an ML sense. Although M-Joint converges to VTS compensation when the number of classes equals the number of model components, if this is not the case, then the auxiliary functions are different. Hence, for noise models to be used for M-Joint compensation, an alternate auxiliary function during estimation should be used. This section outlines differences between the VTS compensation described previously, and noise model estimation for the generation of M-Joint transforms.

The same form of auxiliary from equation (6.2) is used for ML M-Joint noise model estimation

$$\begin{aligned}\mathcal{Q}_{\mathsf{jnt}}(\mathcal{M}_n;\hat{\mathcal{M}}_n) &= \sum_{t=1}^{T}\sum_{m=1}^{M}\gamma_{o,t}^{(m)}\log \mathrm{p}(\boldsymbol{o}_t|m;\mathcal{M},\hat{\mathcal{M}}_n) \\ &= \sum_{t=1}^{T}\sum_{m=1}^{M}\gamma_{o,t}^{(m)}\log\Big[|\boldsymbol{A}^{(r_m)}|\mathcal{N}\Big(\boldsymbol{A}^{(r_m)}\boldsymbol{o}_t + \boldsymbol{b}^{(r_m)};\boldsymbol{\mu}_s^{(m)},\boldsymbol{\Sigma}_s^{(m)} + \boldsymbol{\Sigma}_{\mathsf{b}}^{(r_m)}\Big)\Big] \qquad (6.14)\end{aligned}$$

where the component posteriors $\gamma_{o,t}^{(m)}$ and the log Gaussian probability are now computed using an M-Joint compensated system. Terms independent of the noise model parameters are omitted. The predicted linear transform matrix in this work is block-diagonal

$$\boldsymbol{A}^{(r_m)} \approx \begin{bmatrix}\boldsymbol{J}_x^{(r_m)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{J}_x^{(r_m)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{J}_x^{(r_m)}\end{bmatrix}^{-1} \qquad (6.15)$$

as was given previously in by equation (5.46). Equation (5.47) expressed the feature bias as

$$\boldsymbol{b}^{(r_m)} \approx \begin{bmatrix} \boldsymbol{\mu}_x^{(r_m)} - \left[\boldsymbol{J}_x^{(r_m)}\right]^{-1} \boldsymbol{y}_t|_{\mu_0^{(r_m)}} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tag{6.16}$$

This allows equation (6.14) to be re-expressed as

$$\begin{aligned} \mathcal{Q}_{\mathsf{jnt}} = -\frac{1}{2} \sum_{t=1}^{T} \sum_{m=1}^{M} \gamma_{o,t}^{(m)} \Big[& \log\left|\boldsymbol{\Sigma}_x^{(m)} + \left[\boldsymbol{\Sigma}_{\mathsf{b}}^{(r_m)}\right]_{11}\right| + \log\left|\boldsymbol{\Sigma}_{\Delta x}^{(m)} + \left[\boldsymbol{\Sigma}_{\mathsf{b}}^{(r_m)}\right]_{22}\right| + \log\left|\boldsymbol{\Sigma}_{\Delta^2 x}^{(m)} + \left[\boldsymbol{\Sigma}_{\mathsf{b}}^{(r_m)}\right]_{33}\right| - \\ & 3\log|\boldsymbol{J}_x^{(r)}| \quad + \quad \tilde{\boldsymbol{y}}^{\mathsf{T}} \left(\boldsymbol{\Sigma}_x^{(m)} + \left[\boldsymbol{\Sigma}_{\mathsf{b}}^{(r_m)}\right]_{11}\right)^{-1} \tilde{\boldsymbol{y}} \quad + \\ & \left(\left[\boldsymbol{J}_x^{(r_m)}\right]^{-1} \Delta \boldsymbol{y}_t - \boldsymbol{\mu}_{\Delta x}^{(m)}\right)^{\mathsf{T}} \left(\boldsymbol{\Sigma}_{\Delta x}^{(m)} + \left[\boldsymbol{\Sigma}_{\mathsf{b}}^{(r_m)}\right]_{22}\right)^{-1} \left(\left[\boldsymbol{J}_x^{(r_m)}\right]^{-1} \Delta \boldsymbol{y}_t - \boldsymbol{\mu}_{\Delta x}^{(m)}\right) + \\ & \left(\left[\boldsymbol{J}_x^{(r_m)}\right]^{-1} \Delta^2 \boldsymbol{y}_t - \boldsymbol{\mu}_{\Delta^2 x}^{(m)}\right)^{\mathsf{T}} \left(\boldsymbol{\Sigma}_{\Delta^2 x}^{(m)} + \left[\boldsymbol{\Sigma}_{\mathsf{b}}^{(r_m)}\right]_{33}\right)^{-1} \left(\left[\boldsymbol{J}_x^{(r_m)}\right]^{-1} \Delta^2 \boldsymbol{y}_t - \boldsymbol{\mu}_{\Delta^2 x}^{(m)}\right) \Big] \end{aligned} \tag{6.17}$$

where only the terms that depend on the noise model are shown and

$$\tilde{\boldsymbol{y}} = \left(\left[\boldsymbol{J}_x^{(r_m)}\right]^{-1} \boldsymbol{y}_t - \boldsymbol{\mu}_x^{(m)} + \boldsymbol{\mu}_x^{(r_m)} - \left[\boldsymbol{J}_x^{(r_m)}\right]^{-1} \boldsymbol{y}|_{\mu_0^{(r_m)}}\right) \tag{6.18}$$

Each M-Joint transform $\mathcal{T}^{(r)} = \left\{\boldsymbol{A}^{(r)}, \boldsymbol{b}^{(r)}, \boldsymbol{\Sigma}_{\mathsf{b}}^{(r)}\right\}$, is computed from the joint distribution predicted from the clean speech class model, derived from $\mathcal{M}$ as described in section 5.4, and the estimated noise parameters $\hat{\mathcal{M}}_n$. Thus for noise model estimation, the M-Joint compensation matrices in equation (6.14) are diagonal, but may be easily extended to be block-diagonal by not diagonalising the result from the clean speech covariance and Jacobian matrix multiplications when computing the joint distribution.

The ML M-Joint noise model may be iteratively updated using this second-order gradient-based optimisation scheme

$$\hat{\mu}_{z,i} = \mu_{z,i} - \zeta \frac{\frac{\partial \mathcal{Q}_{\mathsf{jnt}}}{\partial \mu_{z,i}}}{\frac{\partial^2 \mathcal{Q}_{\mathsf{jnt}}}{\partial \mu_{z,i}^2}} \tag{6.19}$$

$$\hat{\sigma}_{z,i}^2 = \sigma_{z,i}^2 - \zeta \frac{\frac{\partial \mathcal{Q}_{\mathsf{jnt}}}{\partial \sigma_{z,i}^2}}{\frac{\partial^2 \mathcal{Q}_{\mathsf{jnt}}}{\partial (\sigma_{z,i}^2)^2}} \tag{6.20}$$

$$\hat{\mu}_{h,i} = \mu_{h,i} - \zeta \frac{\frac{\partial \mathcal{Q}_{\mathsf{jnt}}}{\partial \mu_{h,i}}}{\frac{\partial^2 \mathcal{Q}_{\mathsf{jnt}}}{\partial \mu_{h,i}^2}} \tag{6.21}$$

where ζ is the learning rate. This should be faster than the first-order approach taken to estimate the VTS static additive noise variance.

The second-order derivatives need to be conditioned such that they remain negative to ensure the updates converge to a local maximum. In case they are not negative, a simple back-off strategy is to use a fixed step size with only the first-order gradient. It would be advantageous to optimise the log of the noise variance instead, however this was not

implemented. It is also important to ensure that each step in the iteration improves the auxiliary function and hence a multi-tiered back-off of the estimates generated is used similar to the VTS noise model mean back-off strategy given in figure 6.2. The M-Joint auxiliary function can be expressed as

$$\mathcal{Q}_{\mathsf{jnt}} = \mathcal{Q}_{\mathsf{jnt},y} + \mathcal{Q}_{\mathsf{jnt},\Delta y} + \mathcal{Q}_{\mathsf{jnt},\Delta^2 y} \tag{6.22}$$

where

$$\mathcal{Q}_{\mathsf{jnt},y} = -\frac{1}{2}\sum_{t=1}^{T}\sum_{m=1}^{M}\gamma_{o,t}^{(m)}\left[-\log|\boldsymbol{J}_x^{(r_m)}| + \log\left|\boldsymbol{\Sigma}_x^{(m)} + \left[\boldsymbol{\Sigma}_{\mathsf{b}}^{(r_m)}\right]_{11}\right| \quad + \tilde{\boldsymbol{y}}^{\mathsf{T}}\left(\boldsymbol{\Sigma}_x^{(m)} + \left[\boldsymbol{\Sigma}_{\mathsf{b}}^{(r_m)}\right]_{11}\right)^{-1}\tilde{\boldsymbol{y}}\right] \tag{6.23}$$

$$\mathcal{Q}_{\mathsf{jnt},\Delta y} = -\frac{1}{2}\sum_{t=1}^{T}\sum_{m=1}^{M}\gamma_{o,t}^{(m)}\left[-\log|\boldsymbol{J}_x^{(r_m)}| + \log\left|\boldsymbol{\Sigma}_{\Delta x}^{(m)} + \left[\boldsymbol{\Sigma}_{\mathsf{b}}^{(r_m)}\right]_{22}\right| \; + \right.$$
$$\left.\left(\left[\boldsymbol{J}_x^{(r_m)}\right]^{-1}\Delta\boldsymbol{y}_t - \boldsymbol{\mu}_{\Delta x}^{(m)}\right)^{\mathsf{T}}\left(\boldsymbol{\Sigma}_{\Delta x}^{(m)} + \left[\boldsymbol{\Sigma}_{\mathsf{b}}^{(r_m)}\right]_{22}\right)^{-1}\left(\left[\boldsymbol{J}_x^{(r_m)}\right]^{-1}\Delta\boldsymbol{y}_t - \boldsymbol{\mu}_{\Delta x}^{(m)}\right)\right] \tag{6.24}$$

$$\mathcal{Q}_{\mathsf{jnt},\Delta^2 y} = -\frac{1}{2}\sum_{t=1}^{T}\sum_{m=1}^{M}\gamma_{o,t}^{(m)}\left[-\log|\boldsymbol{J}_x^{(r_m)}| + \log\left|\boldsymbol{\Sigma}_{\Delta^2 x}^{(m)} + \left[\boldsymbol{\Sigma}_{\mathsf{b}}^{(r_m)}\right]_{33}\right| \; + \right.$$
$$\left.\left(\left[\boldsymbol{J}_x^{(r_m)}\right]^{-1}\Delta^2\boldsymbol{y}_t - \boldsymbol{\mu}_{\Delta^2 x}^{(m)}\right)^{\mathsf{T}}\left(\boldsymbol{\Sigma}_{\Delta^2 x}^{(m)} + \left[\boldsymbol{\Sigma}_{\mathsf{b}}^{(r_m)}\right]_{33}\right)^{-1}\left(\left[\boldsymbol{J}_x^{(r_m)}\right]^{-1}\Delta^2\boldsymbol{y}_t - \boldsymbol{\mu}_{\Delta^2 x}^{(m)}\right)\right] \tag{6.25}$$

The back-off then proceeds as follows for the noise mean estimates. First ensure the static M-Joint auxiliary function improves

$$\mathcal{Q}_{\mathsf{jnt},y}(\mathcal{M}_n; \boldsymbol{\mu}_n^i, \boldsymbol{\mu}_h^i) < \mathcal{Q}_{\mathsf{jnt},y}(\mathcal{M}_n; \boldsymbol{\mu}_n^{i+1}, \boldsymbol{\mu}_h^{i+1}) \tag{6.26}$$

then the static and the deltas

$$\mathcal{Q}_{\mathsf{jnt},y}(\mathcal{M}_n; \boldsymbol{\mu}_n^i, \boldsymbol{\mu}_h^i) + \mathcal{Q}_{\mathsf{jnt},\Delta y}(\mathcal{M}_n; \boldsymbol{\mu}_n^i, \boldsymbol{\mu}_h^i) <$$
$$\mathcal{Q}_{\mathsf{jnt},y}(\mathcal{M}_n; \boldsymbol{\mu}_n^{i+1}, \boldsymbol{\mu}_h^{i+1}) + \mathcal{Q}_{\mathsf{jnt},\Delta y}(\mathcal{M}_n; \boldsymbol{\mu}_n^{i+1}, \boldsymbol{\mu}_h^{i+1}) \tag{6.27}$$

and finally the complete auxiliary function

$$\mathcal{Q}_{\mathsf{jnt},y}(\mathcal{M}_n; \boldsymbol{\mu}_n^i, \boldsymbol{\mu}_h^i) + \mathcal{Q}_{\mathsf{jnt},\Delta y}(\mathcal{M}_n; \boldsymbol{\mu}_n^i, \boldsymbol{\mu}_h^i) + \mathcal{Q}_{\mathsf{jnt},\Delta^2 y}(\mathcal{M}_n; \boldsymbol{\mu}_n^i, \boldsymbol{\mu}_h^i) <$$
$$\mathcal{Q}_{\mathsf{jnt},y}(\mathcal{M}_n; \boldsymbol{\mu}_n^{i+1}, \boldsymbol{\mu}_h^{i+1}) + \mathcal{Q}_{\mathsf{jnt},\Delta y}(\mathcal{M}_n; \boldsymbol{\mu}_n^{i+1}, \boldsymbol{\mu}_h^{i+1}) + \mathcal{Q}_{\mathsf{jnt},\Delta^2 y}(\mathcal{M}_n; \boldsymbol{\mu}_n^{i+1}, \boldsymbol{\mu}_h^{i+1}) \tag{6.28}$$

which is the same as ensuring

$$\mathcal{Q}_{\mathsf{jnt}}(\mathcal{M}_n; \boldsymbol{\mu}_n^i, \boldsymbol{\mu}_h^i) < \mathcal{Q}_{\mathsf{jnt},y}(\mathcal{M}_n; \boldsymbol{\mu}_n^{i+1}, \boldsymbol{\mu}_h^{i+1}) \tag{6.29}$$

The static coefficients are interpolated between the new and old until the auxiliary function has increased, then this is conducted with the delta coefficients and finally delta-deltas. Here the noise means have been checked; the estimation and checking of the additive noise variances should follow with these updated noise mean estimates. Although analytical gradients would be much faster, in this work numerically computed derivatives of $\mathcal{Q}_{\mathsf{jnt}}$ were used for all the M-Joint noise model results.

6.4 Initialising the Noise Model

Before the noise estimation process may begin, the noise model must be initialised. Estimates for the initial additive and convolutional noise means may be the minimum energy frame of a noisy utterance observation sequence $\boldsymbol{Y}$ and the expected difference between the noisy speech and the clean

$$\boldsymbol{\mu}_z = \min\{\boldsymbol{Y}\} \tag{6.30}$$

$$\boldsymbol{\mu}_h = \mathcal{E}\{\boldsymbol{Y}\} - \bar{\boldsymbol{\mu}}_x \tag{6.31}$$

as used by Moreno [106], where $\bar{\boldsymbol{\mu}}_x$ is the static global speech mean. For this work, the convolutional noise is instead initialised to zero so that a full pass over the data is not required.

The additive noise variance can be estimated from using the background noise frames [35, 136]. The initial value here of $\boldsymbol{\Sigma}_n$ is set to the variance of the first five frames of the observed speech. This is similar to the noise initialisation scheme in Kim et al. [79], where the first 3–4 frames are considered silence and used to initialise the noise model parameters. This should provide a much better estimate than using the global clean speech variance, which should be considered an upper bound, but may be worse than initialising it to a small value if there is little environmental noise. In this work, if this first initialisation fails to provide a noise estimate that improves the auxiliary function, then the additive noise is set to $\boldsymbol{C}\boldsymbol{f}_0$ and a small initial variance, where $\boldsymbol{f}_0$ is the log zero vector, to represent a "quiet" noise condition.

6.5 Improving Estimation Speed

If the speech acoustic models are HMMs, then a hypothesis of the test data is required to compute the component posteriors. The hypothesis may be obtained by running an initial decoding pass over the test data. It may be quite poor though if the initial noise model is inaccurate. Alternatively, a GMM speech model may be used, which does not require a hypothesis, but is a weaker model of speech. A GMM speech model will also generally be faster than an HMM if there are fewer components; in either form, efficiency is improved by only including components with a minimum occupancy, i.e. number of associated observations. Once the noise model is updated, it may be used again to compensate the speech model, to begin another EM iteration. The hypothesis at this point may also be updated by conducting another pass over the test data with speech models compensated by an improved noise model.

The first iteration to estimate the noise models may be slightly modified to improve the speed of estimating ML M-Joint noise models and provide a general approach to estimating models for both clean- and multistyle-trained models. First, for the expectation stage, when computing the complete data set and auxiliary function, the acoustic models are not compensated. This prevents poor initial noise estimates from degrading noise model estimation for multistyle acoustic models, which already represent varying levels of noise and speech. However, noise estimation may suffer for clean acoustic models since the state alignments may be quite poor when the SNR is low. For the maximisation step, first the ML VTS noise model estimation process outlined in section 6.2 is followed. The initial noise model used is described in the previous section 6.4. For a M-Joint noise model, this VTS-tuned noise model will be further refined, in the same maximisation step, using the M-Joint noise model estimation procedure given in section 6.3. This provides a well-trained ML M-Joint noise model with only one pass over the test data, not including obtaining the hypothesis, in the

same manner for both clean and multistyle acoustic models. This approach was taken since M-Joint noise models needed to be initialised with VTS noise models to obtain good results. Experiments show that this procedure yielded reasonable results for compensating both clean and multistyle acoustic models.

6.6 Noise Model Estimation with a Transformed Feature-Space

Many state-of-the-art speech recognition techniques to improve performance use a transformation of the features making them efficient. Examples of these included CMLLR for adaptation and STC for covariance modelling. In contrast to the description of STC in section 2.3.4, here a $D \times D$ matrix transforms the *noisy* feature space

$$\tilde{\boldsymbol{o}} = \boldsymbol{A}\boldsymbol{o} = \boldsymbol{A}\begin{bmatrix} \boldsymbol{y} \\ \Delta\boldsymbol{y} \\ \Delta^2\boldsymbol{y} \end{bmatrix} \tag{6.32}$$

where $\tilde{\boldsymbol{o}}$ are the noisy features in the transformed space and the time subscript omitted for simplicity. The inverse of the STC matrix, which will be defined as $\tilde{\boldsymbol{A}} = \boldsymbol{A}^{-1}$, may be used to derive the original cepstra from the transformed features

$$\begin{aligned} \boldsymbol{o} &= \tilde{\boldsymbol{A}}\tilde{\boldsymbol{o}} \\ \begin{bmatrix} \boldsymbol{y} \\ \Delta\boldsymbol{y} \\ \Delta^2\boldsymbol{y} \end{bmatrix} &= \begin{bmatrix} \tilde{\boldsymbol{A}}_{11} & \tilde{\boldsymbol{A}}_{12} & \tilde{\boldsymbol{A}}_{13} \\ \tilde{\boldsymbol{A}}_{21} & \tilde{\boldsymbol{A}}_{22} & \tilde{\boldsymbol{A}}_{23} \\ \tilde{\boldsymbol{A}}_{31} & \tilde{\boldsymbol{A}}_{32} & \tilde{\boldsymbol{A}}_{33} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{y}} \\ \Delta\tilde{\boldsymbol{y}} \\ \Delta^2\tilde{\boldsymbol{y}} \end{bmatrix} \end{aligned} \tag{6.33}$$

where the subscript indices denote the blocks in the matrix rather than specific elements. Hence the following static cepstral vectors may be obtained from their decorrelated versions in the following manner

$$\boldsymbol{y} = \tilde{\boldsymbol{A}}_{11}\tilde{\boldsymbol{y}} + \tilde{\boldsymbol{A}}_{12}\Delta\tilde{\boldsymbol{y}} + \tilde{\boldsymbol{A}}_{13}\Delta^2\tilde{\boldsymbol{y}} \tag{6.34}$$

$$\boldsymbol{x} = \tilde{\boldsymbol{A}}_{11}\tilde{\boldsymbol{x}} + \tilde{\boldsymbol{A}}_{12}\Delta\tilde{\boldsymbol{x}} + \tilde{\boldsymbol{A}}_{13}\Delta^2\tilde{\boldsymbol{x}} \tag{6.35}$$

$$\boldsymbol{h} = \tilde{\boldsymbol{A}}_{11}\tilde{\boldsymbol{h}} + \tilde{\boldsymbol{A}}_{12}\Delta\tilde{\boldsymbol{h}} + \tilde{\boldsymbol{A}}_{13}\Delta^2\tilde{\boldsymbol{h}} \tag{6.36}$$

$$\boldsymbol{z} = \tilde{\boldsymbol{A}}_{11}\tilde{\boldsymbol{z}} + \tilde{\boldsymbol{A}}_{12}\Delta\tilde{\boldsymbol{z}} + \tilde{\boldsymbol{A}}_{13}\Delta^2\tilde{\boldsymbol{z}} \tag{6.37}$$

Again, ignoring the time subscripts, the cepstral model of the noisy acoustic environment was previously given as

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{h} + \boldsymbol{C}\log\left(\mathbf{1} + \exp\left(\boldsymbol{C}^{-1}(\boldsymbol{z} - \boldsymbol{x} - \boldsymbol{h})\right)\right)$$

by equation (3.12). Substituting in equations (6.34) to (6.37) gives

$$\begin{aligned} \tilde{\boldsymbol{A}}_{11}\tilde{\boldsymbol{y}} + \tilde{\boldsymbol{A}}_{12}\Delta\tilde{\boldsymbol{y}} + \tilde{\boldsymbol{A}}_{13}\Delta^2\tilde{\boldsymbol{y}} = {} & \tilde{\boldsymbol{A}}_{11}\tilde{\boldsymbol{x}} + \tilde{\boldsymbol{A}}_{12}\Delta\tilde{\boldsymbol{x}} + \tilde{\boldsymbol{A}}_{13}\Delta^2\tilde{\boldsymbol{x}} + \\ & \tilde{\boldsymbol{A}}_{11}\tilde{\boldsymbol{h}} + \tilde{\boldsymbol{A}}_{12}\Delta\tilde{\boldsymbol{h}} + \tilde{\boldsymbol{A}}_{13}\Delta^2\tilde{\boldsymbol{h}} + \\ & \boldsymbol{C}\log\Big\{\mathbf{1} + \exp\Big(\boldsymbol{C}^{-1}\{\tilde{\boldsymbol{A}}_{11}\tilde{\boldsymbol{z}} + \tilde{\boldsymbol{A}}_{12}\Delta\tilde{\boldsymbol{z}} + \tilde{\boldsymbol{A}}_{13}\Delta^2\tilde{\boldsymbol{z}} - \\ & \qquad \tilde{\boldsymbol{A}}_{11}\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{A}}_{12}\Delta\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{A}}_{13}\Delta^2\tilde{\boldsymbol{x}} - \\ & \qquad \tilde{\boldsymbol{A}}_{11}\tilde{\boldsymbol{h}} - \tilde{\boldsymbol{A}}_{12}\Delta\tilde{\boldsymbol{h}} - \tilde{\boldsymbol{A}}_{13}\Delta^2\tilde{\boldsymbol{h}}\}\Big)\Big\} \end{aligned} \tag{6.38}$$

The expected value of the static transformed features, $\tilde{\boldsymbol{y}}$, will also be a function of the dynamic features $\Delta\tilde{\boldsymbol{y}}$ and $\Delta^2\tilde{\boldsymbol{y}}$. This complicates the derivation of the compensated acoustic model parameters for the transformed corrupted speech parameters.

6.6.1 Block-diagonal Feature Transformation

To simplify equation (6.38), only the main diagonal blocks of $\tilde{\boldsymbol{A}}$ may be considered non-zero. Thus if $\tilde{\boldsymbol{A}}_{12} = \tilde{\boldsymbol{A}}_{13} = \boldsymbol{0}$ then this reduces equation (6.38) to

$$\begin{aligned}\tilde{\boldsymbol{A}}_{11}\tilde{\boldsymbol{y}} &= \tilde{\boldsymbol{A}}_{11}\tilde{\boldsymbol{x}} + \tilde{\boldsymbol{A}}_{11}\tilde{\boldsymbol{h}} + \quad \boldsymbol{C}\mathbf{log}\big(\boldsymbol{1} + \mathbf{exp}(\boldsymbol{C}^{-1}\tilde{\boldsymbol{A}}_{11}(\tilde{\boldsymbol{z}} - \tilde{\boldsymbol{x}} - \tilde{\boldsymbol{h}}))\big) \\ \tilde{\boldsymbol{y}} &= \quad \tilde{\boldsymbol{x}} + \quad \tilde{\boldsymbol{h}} + \boldsymbol{A}_{11}\boldsymbol{C}\mathbf{log}(\boldsymbol{1} + \mathbf{exp}(\boldsymbol{C}^{-1}\tilde{\boldsymbol{A}}_{11}(\tilde{\boldsymbol{z}} - \tilde{\boldsymbol{x}} - \tilde{\boldsymbol{h}})))\end{aligned} \tag{6.39}$$

where the inverse of $\tilde{\boldsymbol{A}}_{11}$ is now simply the block $\boldsymbol{A}_{11}$ from a block-diagonal $\boldsymbol{A}$. The first-order VTS approximation of equation (6.39) is very similar to the form without the feature transformation

$$\tilde{\boldsymbol{y}}_{\mathsf{vts}} = \tilde{\boldsymbol{y}}\big|_{\tilde{\mu}_0} + \tilde{\boldsymbol{J}}_x(\tilde{\boldsymbol{x}} - \boldsymbol{\mu}_{\tilde{x}}) + \tilde{\boldsymbol{J}}_z(\tilde{\boldsymbol{z}} - \boldsymbol{\mu}_{\tilde{z}}) + \tilde{\boldsymbol{J}}_h(\tilde{\boldsymbol{h}} - \boldsymbol{\mu}_{\tilde{h}}) \tag{6.40}$$

with the Jacobian matrices now given by

$$\tilde{\boldsymbol{J}}_x = \Big[\boldsymbol{\nabla}_{\tilde{\boldsymbol{x}}}\,\tilde{y}_1\big|_{\tilde{\mu}_0} \quad \cdots \quad \boldsymbol{\nabla}_{\tilde{\boldsymbol{x}}}\,\tilde{y}_{D_s}\big|_{\tilde{\mu}_0}\Big]^{\mathsf{T}} = \boldsymbol{I} - \boldsymbol{A}_{11}\boldsymbol{C}\tilde{\boldsymbol{F}}\boldsymbol{C}^{-1}\tilde{\boldsymbol{A}}_{11} \tag{6.41}$$

$$\tilde{\boldsymbol{J}}_h = \tilde{\boldsymbol{J}}_x, \qquad \tilde{\boldsymbol{J}}_z = \boldsymbol{A}_{11}\boldsymbol{C}\tilde{\boldsymbol{F}}\boldsymbol{C}^{-1}\tilde{\boldsymbol{A}}_{11} \tag{6.42}$$

and D_s denoting the number of static coefficients. The expansion point $\tilde{\boldsymbol{\mu}}_0$ is about the mean of the transformed additive noise $\boldsymbol{\mu}_{\tilde{z}}$, speech $\boldsymbol{\mu}_{\tilde{x}}$, and channel $\boldsymbol{\mu}_{\tilde{h}}$. The elements of the diagonal matrix $\tilde{\boldsymbol{F}}$ are

$$\tilde{f}_{ii} = \frac{\exp\Big(\boldsymbol{c}_i^{-1}\tilde{\boldsymbol{A}}_{11}\big(\boldsymbol{\mu}_{\tilde{z}} - \boldsymbol{\mu}_{\tilde{x}} - \boldsymbol{\mu}_{\tilde{h}}\big)\Big)}{1 + \exp\Big(\boldsymbol{c}_i^{-1}\tilde{\boldsymbol{A}}_{11}\big(\boldsymbol{\mu}_{\tilde{z}} - \boldsymbol{\mu}_{\tilde{x}} - \boldsymbol{\mu}_{\tilde{h}}\big)\Big)} \tag{6.43}$$

where recall the term $\boldsymbol{c}_i^{-1}$ gives a row vector that is the ith row of the inverse DCT matrix $\boldsymbol{C}^{-1}$. The expected values and dynamic feature compensation are similar to the untransformed features

$$\begin{aligned}\boldsymbol{\mu}_{\tilde{y}} &= \mathcal{E}\{\tilde{\boldsymbol{y}}\} \approx \mathcal{E}\{\tilde{\boldsymbol{y}}_{\mathsf{vts}}\} = \tilde{\boldsymbol{y}}\big|_{\tilde{\mu}_0} \\ &= \boldsymbol{\mu}_{\tilde{x}} + \boldsymbol{\mu}_{\tilde{h}} + \boldsymbol{A}_{11}\boldsymbol{C}\mathbf{log}\Big(\boldsymbol{1} + \mathbf{exp}\Big(\boldsymbol{C}^{-1}\tilde{\boldsymbol{A}}_{11}\big(\boldsymbol{\mu}_{\tilde{z}} - \boldsymbol{\mu}_{\tilde{x}} - \boldsymbol{\mu}_{\tilde{h}}\big)\Big)\Big)\end{aligned} \tag{6.44}$$

and the variance, after assuming no channel variation, is

$$\begin{aligned}\boldsymbol{\Sigma}_{\tilde{y}} &= \mathtt{Var}\{\tilde{\boldsymbol{y}}\} \approx \mathtt{Var}\{\tilde{\boldsymbol{y}}_{\mathsf{vts}}\} \\ &= \tilde{\boldsymbol{J}}_x\boldsymbol{\Sigma}_{\tilde{x}}\tilde{\boldsymbol{J}}_x^{\mathsf{T}} + \tilde{\boldsymbol{J}}_z\boldsymbol{\Sigma}_{\tilde{z}}\tilde{\boldsymbol{J}}_z^{\mathsf{T}}\end{aligned} \tag{6.45}$$

which is typically diagonalised. The derivations for the dynamic coefficients, using the Continuous-Time approximation, remain unchanged by the feature transformation other than the modified Jacobian matrices

$$\boldsymbol{\mu}_{\Delta\tilde{y}} \approx \tilde{\boldsymbol{J}}_x\boldsymbol{\mu}_{\Delta\tilde{x}} \tag{6.46}$$

$$\boldsymbol{\Sigma}_{\Delta\tilde{y}} \approx \tilde{\boldsymbol{J}}_x\boldsymbol{\Sigma}_{\Delta\tilde{x}}\tilde{\boldsymbol{J}}_x^{\mathsf{T}} + \tilde{\boldsymbol{J}}_z\boldsymbol{\Sigma}_{\Delta\tilde{z}}\tilde{\boldsymbol{J}}_z^{\mathsf{T}} \tag{6.47}$$

Note that this block form of feature transformation may be easily implemented in an existing system by applying the feature transform to the DCT and IDCT matrices

$$\tilde{\boldsymbol{C}} = \boldsymbol{A}_{11}\boldsymbol{C} \qquad \text{and} \qquad \tilde{\boldsymbol{C}}^{-1} = \boldsymbol{C}^{-1}\tilde{\boldsymbol{A}}_{11} \tag{6.48}$$

The VTS and M-Joint noise estimation and compensation processes can then operate without any further regard to the feature transform with these updated matrices.

6.7 Summary

This chapter discussed the maximum likelihood estimation of noise models consisting of the channel noise mean and additive noise mean and variance. The noise estimation procedure should be consistent with the target noise compensation scheme: both VTS and M-Joint ML noise model estimation procedures were presented. Although the additive noise may be estimated from the background, non-speech audio segments, this requires a voice activity detector which has its own issues when the noise level rises. A benefit of this ML noise model estimation procedure it it allows noise to be estimated during speech. Thus if there are long speech segments where the noise changes, then the noise model may be updated within the utterance. For a frequently changing environment, a fast compensation form like JUD is particularly important. Also, a noise model may be estimated for with multistyle acoustic models. In this case, the model is no longer an acoustic noise model, but a set of parameters that maximise the test data likelihood when the multistyle models are compensated with predictive forms that assume a certain model of the acoustic environment. Hence an EM-based approach to noise model estimation gives a consistent and comprehensive method of generating noise models for model-based noise compensation.

Chapter 7
Joint Adaptive Training

Adaptation has been shown to be a powerful technique to reduce the acoustic mismatch between training and test conditions [46, 89, 110]. When there is insufficient data to re-train models to match the testing condition, adaptation provides an efficient way to include data from the test condition. Adaptative training extends this by removing testing condition variablity from the acoustic model itself. It was first shown to be succesful for speaker normalisation [3]. However it can be more generally applied to reduce the acoustic mismatch from many factors such as the speaker, channel and environmental variability [43]. Using adaptive training yields a "purer" acoustic model than multistyle techniques that need to incorporate all the extraneous variability due to non-speech factors in the models. Moreover, the resulting canonical model may be a better "clean" acoustic model for transformation that all predictive noise compensation techniques require.

Linear transforms like MLLR [3] and CMLLR [46] have been successfully used in adaptive training. In this work, the use of JUD transforms for adaptive training framework is explored as a method of handling training data with varying noise levels. This form of noise adaptive training is referred to as joint adaptive training (JAT). Rather than separately modelling the speaker and noise condition with a MLLR transform and cluster adaptive training respectively as in Gales [43], an M-Joint transform will model both for each speaker/noise condition.

In CMLLR and feature normalisation techniques, observations are compensated and treated as if they were done so exactly and perfectly when canonical model parameters are estimated. As this chapter shows, the uncertainty variance bias term de-weights noisy observations. When the noise is high, the uncertainty is large, and these observations contribute less to estimating the canonical model parameters. This allows JAT to train a "cleaner" acoustic model of speech.

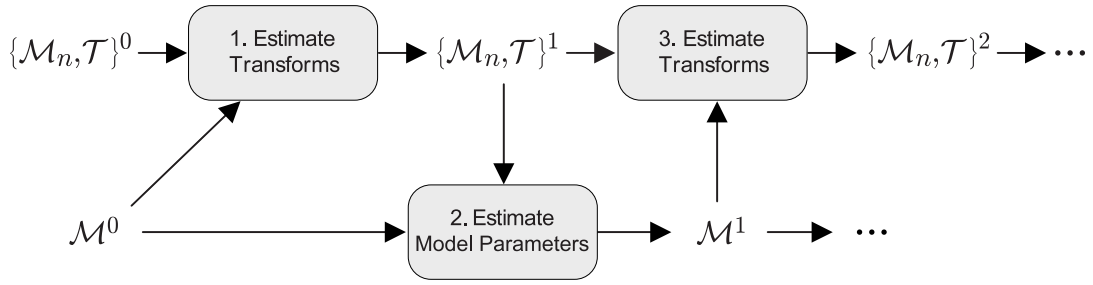


Figure 7.1: Joint adaptive training.

7.1 An Adaptive Training Framework

JAT follows the adaptive training framework outlined in section 2.5.5. The canonical acoustic model parameters $\mathcal{M}$ and set of transforms $\mathcal{T}$ are estimated such that they maximise the likelihood of the heterogeneous training data comprised of H homogeneous blocks. From equation (2.70), this may be expressed as

$$\mathrm{p}(\boldsymbol{S}|\mathcal{W}_{\mathsf{r}};\mathcal{M},\mathcal{T}) = \prod_{h=1}^{H} \mathrm{p}(\boldsymbol{S}^{(h)}|\mathcal{W}_{\mathsf{r}}^{(h)};\mathcal{M},\mathcal{T}^{(h)}) \tag{7.1}$$

$$= \prod_{h=1}^{H} \sum_{\boldsymbol{m}\in\boldsymbol{M}^{(h)}} \mathrm{P}(\boldsymbol{m};\mathcal{M},\mathcal{T}^{(h)}) \prod_{t=1}^{T^{(h)}} \mathrm{p}(\boldsymbol{s}_t|m_t;\mathcal{M},\mathcal{T}^{(h)}) \tag{7.2}$$

To review the notation, the heterogeneous training data $\boldsymbol{S}$ has a transcription $\mathcal{W}_{\mathsf{r}}$ and is comprised of H blocks of homogeneous data denoted by $\boldsymbol{S}^{(h)}$ which is of length $T^{(h)}$ and has transcription $\mathcal{W}^{(h)}$. The complete data for a homogeneous block h is $\left\{\boldsymbol{M}^{(h)},\boldsymbol{S}^{(h)}\right\}$ where $\boldsymbol{M}^{(h)}$ represents all possible hidden component/state sequences for $\boldsymbol{S}^{(h)}$ and a given transcription. A component sequence $\boldsymbol{m}$ is of length $T^{(h)}$. While the entire training data set $\boldsymbol{S}$ may have many speakers and come from many different noise environments with varying SNR the homogeneous block of data should be from a single speaker in a stationary noise environment. Again, EM is used to iteratively find suitable canonical model parameters and the noise model parameters to generate the JUD transforms. The M-Joint auxiliary function, from equation (6.14) and used to give ML estimates of the noise for M-Joint compensation, may be extended to

$$\begin{aligned}\mathcal{Q}_{\mathsf{jnt}}\big(\mathcal{M},\mathcal{T};\hat{\mathcal{M}},\hat{\mathcal{T}}\big) &= \mathbb{E}_{\mathcal{M},\mathcal{T}}\Big[\log \mathrm{p}(\boldsymbol{S},\boldsymbol{M};\hat{\mathcal{M}},\hat{\mathcal{T}})\Big] \\ &= \sum_{h=1}^{H}\sum_{t=1}^{T^{(h)}}\sum_{m=1}^{M}\gamma_{s,t}^{(mh)}\log\Big[|\hat{\boldsymbol{A}}^{(r_m h)}|\mathcal{N}\Big(\hat{\boldsymbol{A}}^{(r_m h)}\boldsymbol{s}_t+\hat{\boldsymbol{b}}^{(r_m h)};\hat{\boldsymbol{\mu}}_s^{(m)},\hat{\boldsymbol{\Sigma}}_s^{(m)}+\hat{\boldsymbol{\Sigma}}_{\mathsf{b}}^{(r_m h)}\Big)\Big]\end{aligned} \tag{7.3}$$

where $\gamma_{s,t}^{(mh)}$ is the posterior probability of component m given the observation sequence $\boldsymbol{S}^{(h)}$, transform set $\mathcal{T}^{(h)}$, and model set $\mathcal{M}$. The subscript s indicates parameters are associated with the training data rather than clean conditions. The term r_m, which gives the class r, and thus transform $\mathcal{T}^{(r_m h)} = \left\{\boldsymbol{A}^{(r_m h)},\boldsymbol{b}^{(r_m h)},\boldsymbol{\Sigma}_{\mathsf{b}}^{(r_m h)}\right\}$, associated with component m, was described previously in section 5.2.4.

The overall training regime is as the standard adaptive training algorithm given in figure 2.14 where transforms are model parameters are iteratively estimated. However, direct maximisation of the canonical model parameters is difficult, hence a gradient based approach will be taken in a generalised EM fashion; this is discussed in section 7.3. Figure 7.1 illustrates one and a half iterations of interleaved JAT. The symbol $\mathcal{M}_n$, for example, represents the noise parameters for iteration 1 that are associated with transform $\mathcal{T}$. First, given the current acoustic models $\mathcal{M}$ a new set of transform $\mathcal{T}$ is estimated. Subsequently, the canonical model parameters are updated to $\hat{\mathcal{M}}$ given this new set of transforms. Multiple iterations of this interleaved training may be performed to optimise the auxiliary function.

7.2 Estimating M-Joint Transforms

Section 6.3 discussed noise model estimation using a M-Joint auxiliary function. In JAT, transforms are generated from a noise model, estimated using the same procedures outlined in the previous chapter, and the clean speech model. This clean speech class model, described in section 5.4.1, needs to be re-computed every time the canonical model is updated. A disconnect may arise when during the estimation of a new set of transforms, the initial ML noise parameters may have been estimated using a different clean speech class model. This problem can be clearly understood by following the adaptive training process in figure 7.1. In step 1, the set of transforms $\hat{\mathcal{T}}$ is generated from $\hat{\mathcal{M}}_n$ and the clean speech class model derived from $\mathcal{M}$ and then used in step 2, where a new set of canonical model parameters $\hat{\mathcal{M}}$ are estimated. But when step 3 starts, during the expectation step, the set of transforms generated from $\hat{\mathcal{M}}_n$ and clean speech class model from $\hat{\mathcal{M}}$ is not the same as $\hat{\mathcal{T}}$, which is the set of transforms that EM requires to be the initial starting point.

Nevertheless, it may be possible to begin with the M-Joint transform produced from $\hat{\mathcal{M}}_n$ and $\hat{\mathcal{M}}$. However, not only is it now necessary to verify the newly estimated M-Joint transform yields a higher auxiliary function value than the initial input parameters, but that it also exceeds the auxiliary function value using the input joint transforms, in this example $\hat{\mathcal{T}}$, which were computed from the previous clean speech class model and the initial noise parameters. It may be the case that due to the change in clean speech class model, that the newly estimated parameters may not improve the auxiliary function over the input transform, which was computed from a different clean speech class model. If this is the case, the transform is not updated and the input transform remains the current "best transform".

7.3 Estimating Canonical Model Parameters

After a new set of transforms is estimated, the model parameters must be re-trained. The auxiliary function where only terms dependent on the model parameters are shown is

$$\mathcal{Q}_{\mathsf{jnt}}\Big(\mathcal{M},\mathcal{T};\hat{\mathcal{M}},\mathcal{T}\Big) = \\ -\frac{1}{2}\sum_{h=1}^{H}\sum_{t=1}^{T^{(h)}}\sum_{m=1}^{M}\gamma_{s,t}^{(mh)}\sum_{i=1}^{D}\left(\log\big(\sigma_{s,i}^{(m)2}+\sigma_{\mathsf{b},i}^{(r_m h)2}\big)+\frac{\big(\boldsymbol{a}_i^{(r_m h)}\boldsymbol{s}_t+b_i^{(r_m h)}-\mu_{s,i}^{(m)}\big)^2}{\sigma_{s,i}^{(m)2}+\sigma_{\mathsf{b},i}^{(r_m h)2}}\right) \tag{7.4}$$

where diagonal covariance matrices assumed and $\boldsymbol{a}_{\tilde{i}}^{(r_m h)}$ gives the ith row of $\boldsymbol{A}^{(r_m h)}$. Because the M-Joint transform parameters affect the model parameters and are shared over many homogeneous blocks, there is no closed form solution for the model parameters that maximise this auxiliary function. Hence a generalised EM approach is taken, where Newton's method is applied to optimise the model parameters in the maximisation step

$$\begin{bmatrix} \hat{\mu}_{s,i}^{(m)} \\ \hat{\sigma}_{s,i}^{(m)2} \end{bmatrix} = \begin{bmatrix} \mu_{s,i}^{(m)} \\ \sigma_{s,i}^{(m)2} \end{bmatrix} - \zeta \begin{bmatrix} \frac{\partial^2 \mathcal{Q}_{\mathsf{jnt}}}{\partial \left(\mu_{s,i}^{(m)}\right)^2} & \frac{\partial^2 \mathcal{Q}_{\mathsf{jnt}}}{\partial \mu_{s,i}^{(m)} \partial \sigma_{s,i}^{(m)2}} \\ \frac{\partial^2 \mathcal{Q}_{\mathsf{jnt}}}{\partial \sigma_{s,i}^{(m)2} \partial \mu_{s,i}^{(m)}} & \frac{\partial^2 \mathcal{Q}_{\mathsf{jnt}}}{\partial \left(\sigma_{s,i}^{(m)2}\right)^2} \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial \mathcal{Q}_{\mathsf{jnt}}}{\partial \mu_{s,i}^{(m)}} \\ \frac{\partial \mathcal{Q}_{\mathsf{jnt}}}{\partial \sigma_{s,i}^{(m)2}} \end{bmatrix} \tag{7.5}$$

This requires both first- and second-order derivatives of the auxiliary function with respect to the model mean and variance. Using a second-order Newton approach allows faster convergence than the first-order gradient method used to estimate the additive noise variance in section 6.2.2 of the previous chapter.

The first derivative of the auxiliary in equation (7.4) w.r.t. the mean of component m, dimension i is

$$\frac{\partial \mathcal{Q}_{\mathsf{jnt}}}{\partial \mu_{s,i}^{(m)}} = \sum_{h=1}^{H} \sum_{t=1}^{T^{(h)}} \frac{\gamma_{s,t}^{(mh)}}{\sigma_{s,i}^{(m)2} + \sigma_{\mathsf{b},i}^{(r_m h)2}} \left(\boldsymbol{a}_{\tilde{i}}^{(r_m h)} \boldsymbol{s}_t + b_i^{(r_m h)} - \mu_{s,i}^{(m)} \right) \tag{7.6}$$

and with respect to the model variance

$$\frac{\partial \mathcal{Q}_{\mathsf{jnt}}}{\partial \sigma_{s,i}^{(m)2}} = \sum_{h=1}^{H} \sum_{t=1}^{T^{(h)}} \frac{\gamma_{s,t}^{(mh)}}{2\left(\sigma_{s,i}^{(m)2} + \sigma_{\mathsf{b},i}^{(r_m h)2}\right)} \left(\frac{(\boldsymbol{a}_{\tilde{i}}^{(r_m h)} \boldsymbol{s}_t + b_i^{(r_m h)} - \mu_{s,i}^{(m)})^2}{\sigma_{s,i}^{(m)2} + \sigma_{\mathsf{b},i}^{(r_m h)2}} - 1 \right) \tag{7.7}$$

The uncertainty bias $\sigma_{\mathsf{b},i}^{(r_m h)2}$ adjusts the component posterior $\gamma_{s,t}^{(mh)}$ for both these derivatives. If the SNR is high, then there is no uncertainty and the posterior is not affected. When the SNR is low, the uncertainty will be large, reducing the contribution of noisy observations by de-weighting the component posterior. In areas where the noise completely subsumes the speech, the uncertainty will ensure that these observations do not contribute to the estimate of the model parameters at all—the model parameters will not be updated since the first derivatives of the auxiliary function w.r.t. the model means and variance will be naught. This allows the model parameters to be a better representation of "clean" speech. With normalisation schemes or MLLR-based adaptation, once observations are compensated for noise, they are all treated equally. In contrast, with the uncertainty term, JAT will give greater importance to observations that are less "noisy".

The Hessian matrix is also required and is comprised of the following terms

$$\frac{\partial^2 \mathcal{Q}_{\mathsf{jnt}}}{\partial \left(\mu_{s,i}^{(m)}\right)^2} = \sum_{h=1}^{H} \frac{-1}{\sigma_{s,i}^{(m)2} + \sigma_{\mathsf{b},i}^{(r_m h)2}} \sum_{t=1}^{T^{(h)}} \gamma_{s,t}^{(mh)} \tag{7.8}$$

$$\frac{\partial^2 \mathcal{Q}_{\mathsf{jnt}}}{\partial \left(\sigma_{s,i}^{(m)2}\right)^2} = \sum_{h=1}^{H} \frac{1}{\sigma_{s,i}^{(m)2} + \sigma_{\mathsf{b},i}^{(r_m h)2}} \sum_{t=1}^{T^{(h)}} \gamma_{s,t}^{(mh)} \left(\frac{1}{2} - \frac{\left(\boldsymbol{a}_{\tilde{i}}^{(r_m h)} \boldsymbol{s}_t + b_i^{(r_m h)} - \mu_{s,i}^{(m)}\right)^2}{\sigma_{s,i}^{(m)2} + \sigma_{\mathsf{b},i}^{(r_m h)2}} \right) \tag{7.9}$$

$$\frac{\partial^2 \mathcal{Q}_{\mathsf{jnt}}}{\partial \mu_{s,i}^{(m)} \partial \sigma_{s,i}^{(m)2}} = \frac{\partial^2 \mathcal{Q}_{\mathsf{jnt}}}{\partial \sigma_{s,i}^{(m)2} \partial \mu_{s,i}^{(m)}} = \sum_{h=1}^{H} \frac{-1}{\sigma_{s,i}^{(m)2} + \sigma_{\mathsf{b},i}^{(r_m h)2}} \sum_{t=1}^{T^{(h)}} \gamma_{s,t}^{(mh)} \frac{\boldsymbol{a}_{\tilde{i}}^{(r_m h)} \boldsymbol{s}_t + b_i^{(r_m h)} - \mu_{s,i}^{(m)}}{\sigma_{s,i}^{(m)2} + \sigma_{\mathsf{b},i}^{(r_m h)2}} \tag{7.10}$$

To compute the terms of the Hessian matrix, the following statistics may be gathered per recognition component

$$w_{1,i}^{(m)} = \frac{\partial^2 \mathcal{Q}_{\mathsf{jnt}}}{\partial \big(\mu_{s,i}^{(m)}\big)^2} \tag{7.11}$$

$$w_{2,i}^{(m)} = \sum_{h=1}^{H} \sum_{t=1}^{T^{(h)}} \gamma_{s,t}^{(mh)} \frac{1}{\big(\sigma_{s,i}^{(m)2} + \sigma_{\mathsf{b},i}^{(r_m h)2}\big)^2} \tag{7.12}$$

$$w_{3,i}^{(m)} = \frac{\partial^2 \mathcal{Q}_{\mathsf{jnt}}}{\partial \sigma_{s,i}^{(m)2} \partial \mu_i^{(m)}} \tag{7.13}$$

$$w_{4,i}^{(m)} = -\sum_{h=1}^{H} \sum_{t=1}^{T^{(h)}} \gamma_{s,t}^{(mh)} \frac{\big(\boldsymbol{a}_{\tilde{i}}^{(r_m h)} \boldsymbol{s}_t + b_i^{(r_m h)} - \mu_{s,i}^{(m)}\big)^2}{\big(\sigma_{s,i}^{(m)2} + \sigma_{\mathsf{b},i}^{(r_m h)2}\big)^3} \tag{7.14}$$

while the first-order partial derivatives may be directly accumulated. The accumulates given by equations (7.11) and (7.13) yield equations (7.8) and equations (7.10) respectively. The remaining equation (7.9), the second-order derivative w.r.t. the model variance, may then be re-expressed as

$$\frac{\partial^2 \mathcal{Q}_{\mathsf{jnt}}}{\partial \big(\sigma_{s,i}^{(m)2}\big)^2} = w_{4,i}^{(m)} + \frac{1}{2} w_{2,i}^{(m)} \tag{7.15}$$

7.3.1 Stabilising the Estimation Process

When using this form of optimisation for maximising the auxiliary function, it is important to ensure that the iterations are approaching a global maximum rather than the minimum. This implies that the Hessian matrix must be negative-definite and necessitates checking the second derivatives are negative. Upon inspection, equation (7.8) is guaranteed to be always negative, while equation (7.15) is not. Hence to ensure $\frac{\partial^2 \mathcal{Q}_{\mathsf{jnt}}}{\partial (\sigma_{s,i}^{(m)2})^2}$ is negative, equation (7.15) may be re-written as

$$\begin{aligned} \frac{\partial^2 \mathcal{Q}_{\mathsf{jnt}}}{\partial \big(\sigma_{s,i}^{(m)2}\big)^2} &= w_{2,i}^{(m)} \left(\frac{w_{4,i}^{(m)}}{w_{2,i}^{(m)}} + \frac{1}{2} \right) \\ &\approx w_{2,i}^{(m)} \left(-\hat{\vartheta} + \frac{1}{2} \right) \end{aligned} \tag{7.16}$$

where

$$\hat{\vartheta} = \max \left(\vartheta, -\frac{w_{4,i}^{(m)}}{w_{2,i}^{(m)}} \right) \tag{7.17}$$

This parameter ϑ should remain greater than a half to ensure stability of the optimisation. As the model parameters become better trained, the updated variance should be equal to the expected square of the deviation of the transformed speech from the mean, hence

$$\sum_{h=1}^{H} \sum_{t=1}^{T^{(h)}} \gamma_{s,t}^{(mh)} \frac{\big(\boldsymbol{a}_{\tilde{i}}^{(r_m h)} \boldsymbol{s}_t + b_i^{(r_m h)} - \mu_{s,i}^{(m)}\big)^2}{\sigma_{s,i}^{(m)2} + \sigma_{\mathsf{b},i}^{(r_m h)2}} \rightarrow 1 \tag{7.18}$$

Thus by referring back to equations (7.12) and (7.14), it may be observed that the ratio of $w_{4,i}^{(m)}$ to $w_{2,i}^{(m)}$ should converge to unity as the model parameters better approximate the training data. That is

$$\frac{w_{4,i}^{(m)}}{w_{2,i}^{(m)}} \to 1 \tag{7.19}$$

Hence following how this ratio changes during the training process is useful for gaging convergence. Note from equation (7.16) it follows that if this thresholding is applied, the second derivative becomes a scaling of $w_{2,i}^{(m)}$ by ϑ. Hence to improve estimation speed, ϑ should be large, however as the parameters become better trained, ϑ should diminish.

Also, instead of directly optimising the variance, the log of the variance is estimated to ensure that the converged value remains positive in this work. Thus, make the change of variable

$$\boldsymbol{\varsigma}_s^{(m)} = \mathbf{log}\, \boldsymbol{\Sigma}_s^{(m)} \tag{7.20}$$

where **log** is an element-wise log function. Thus the parameter update formula, stated in equation (7.5), becomes

$$\begin{bmatrix} \hat{\mu}_{s,i}^{(m)} \\ \hat{\varsigma}_{s,i}^{(m)} \end{bmatrix} = \begin{bmatrix} \mu_{s,i}^{(m)} \\ \varsigma_{s,i}^{(m)} \end{bmatrix} - \zeta \begin{bmatrix} \frac{\partial^2 \mathcal{Q}_{\mathsf{jnt}}}{\partial \left(\mu_{s,i}^{(m)}\right)^2} & \frac{\partial^2 \mathcal{Q}_{\mathsf{jnt}}}{\partial \mu_{s,i}^{(m)} \partial \varsigma_{s,i}^{(m)}} \\ \frac{\partial^2 \mathcal{Q}_{\mathsf{jnt}}}{\partial \varsigma_{s,i}^{(m)} \partial \mu_{s,i}^{(m)}} & \frac{\partial^2 \mathcal{Q}_{\mathsf{jnt}}}{\partial \left(\varsigma_{s,i}^{(m)}\right)^2} \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial \mathcal{Q}_{\mathsf{jnt}}}{\partial \mu_{s,i}^{(m)}} \\ \frac{\partial \mathcal{Q}_{\mathsf{jnt}}}{\partial \varsigma_{s,i}^{(m)}} \end{bmatrix} \tag{7.21}$$

The partial derivatives w.r.t. $\varsigma_{s,i}^{(m)}$ may be expressed as a function of the previously given partial derivatives

$$\begin{aligned} \frac{\partial \mathcal{Q}_{\mathsf{jnt}}}{\partial \varsigma_{s,i}^{(m)}} &= \frac{\partial \sigma_{s,i}^{(m)2}}{\partial \varsigma_{s,i}^{(m)}} \frac{\partial \mathcal{Q}_{\mathsf{jnt}}}{\partial \sigma_{s,i}^{(m)2}} \\ &= \sigma_{s,i}^{(m)2} \frac{\partial \mathcal{Q}_{\mathsf{jnt}}}{\partial \sigma_{s,i}^{(m)2}} \end{aligned} \tag{7.22}$$

$$\begin{aligned} \frac{\partial^2 \mathcal{Q}_{\mathsf{jnt}}}{\partial \mu_{s,i}^{(m)} \partial \varsigma_{s,i}^{(m)}} = \frac{\partial^2 \mathcal{Q}_{\mathsf{jnt}}}{\partial \varsigma_{s,i}^{(m)} \partial \mu_{s,i}^{(m)}} &= \frac{\partial \sigma_{s,i}^{(m)2}}{\partial \varsigma_{s,i}^{(m)}} \frac{\partial^2 \mathcal{Q}_{\mathsf{jnt}}}{\partial \sigma_{s,i}^{(m)2} \partial \mu_{s,i}^{(m)}} \\ &= \sigma_{s,i}^{(m)2} \frac{\partial^2 \mathcal{Q}_{\mathsf{jnt}}}{\partial \sigma_{s,i}^{(m)2} \partial \mu_{s,i}^{(m)}} \end{aligned} \tag{7.23}$$

since $\frac{\partial \sigma_{s,i}^{(m)2}}{\partial \varsigma_{s,i}^{(m)}} = \exp \varsigma_{s,i}^{(m)} = \sigma_{s,i}^{(m)2}$, and lastly differentiating equation (7.22)

$$\begin{aligned} \frac{\partial^2 \mathcal{Q}_{\mathsf{jnt}}}{\partial \left(\varsigma_{s,i}^{(m)}\right)^2} &= \frac{\partial}{\partial \varsigma_{s,i}^{(m)}} \left\{ \sigma_{s,i}^{(m)2} \frac{\partial \mathcal{Q}_{\mathsf{jnt}}}{\partial \sigma_{s,i}^{(m)2}} \right\} \\ &= \frac{\partial \sigma_{s,i}^{(m)2}}{\partial \varsigma_{s,i}^{(m)}} \frac{\partial \mathcal{Q}_{\mathsf{jnt}}}{\partial \sigma_{s,i}^{(m)2}} + \sigma_{s,i}^{(m)2} \left\{ \frac{\partial \sigma_{s,i}^{(m)2}}{\partial \varsigma_{s,i}^{(m)}} \frac{\partial \mathcal{Q}_{\mathsf{jnt}}}{\partial \left(\sigma_{s,i}^{(m)2}\right)^2} \right\} \\ &= \sigma_{s,i}^{(m)2} \frac{\partial \mathcal{Q}_{\mathsf{jnt}}}{\partial \sigma_{s,i}^{(m)2}} + \left(\sigma_{s,i}^{(m)2}\right)^2 \frac{\partial \mathcal{Q}_{\mathsf{jnt}}}{\partial \left(\sigma_{s,i}^{(m)2}\right)^2} \end{aligned} \tag{7.24}$$

Finally, the stabilising learning rate ζ in equation (7.5) may be less than one, but in this work a value of unity is used. It was found that during this optimisation process, the variance is sometimes driven to infinity. Hence a measure was introduced to stabilise the variance estimation—the variance was limited to only increase or decrease by a factor υ. Specifically the conditioned model variance is given by

$$\breve{\sigma}_{s,i}^{(m)2} = \min\left(\max\left(\hat{\sigma}_{s,i}^{(m)2}, \frac{1}{\upsilon}\sigma_{s,i}^{(m)2}\right), \upsilon\sigma_{s,i}^{(m)2}\right) \tag{7.25}$$

In this work υ was set at 2.

7.4 Summary

This chapter has described how model-based JUD transforms can be used in an adaptive training framework. Compared to speaker adaptive training, this form of noise compensation may be referred to as noise adaptive training. The end result of using JUD transforms to represent environmental noise during acoustic model training is a "purer" acoustic model of speech. A closed form solution for updating the acoustic model parameters was unavailable, thus a gradient descent approach was taken. Various methods were described to improve the stability of the parameter estimation. A key difference between JAT and SAT using MLLR-type transforms is that during acoustic model training the uncertainty variance bias term proportionally de-weights noisier observations. In normalisation and MLLR-based adaptive training techniques, although the noise may be removed from observations or the model variances updated, the uncertainty due to the noise on observations is not accounted for, but is automatically in JAT. This gives a powerful method to factor out the effects of noise from the acoustic models.

CHAPTER 8

Experimental Results on Artificially Corrupted Speech

This chapter presents results from experiments designed to examine the effectiveness of uncertainty decoding. Evaluation is based on the small vocabulary Aurora2 corpus and noise-corrupted medium vocabulary Resource Management (RM) task. Noise is artificially added in these corpora to carefuly control the experiments and the level of noise in tests. Techniques can be assessed without considering complications arising from noise model estimation, appoximations in the mismatch function or the Lombard effect. Various other noise robustness algorithms are evaluated to provide a contrast with uncertainty decoding. Two forms of front-end uncertainty decoding are presented: SPLICEU and FE-Joint distribution uncertainty decoding. These are compared to a fast feature enhancement form, SPLICE, and the efficient and effective model adaptation scheme CMLLR. Single-pass re-training (SPR) was used to generate matched systems to represent idealised model compensation when stereo data are available. Unless noted, recognition parameters such as grammar scale factor, pruning threshold and model insertion penalty were tuned once for the clean system on clean data and kept constant for the other experiments.

8.1 The Aurora2 Corpus

The Aurora2 [68] small vocabulary digit string recognition task is an artificially noise-corrupted TIDigits corpus. The task involves transcribing utterances that are one to seven digits long. The standard Aurora2 system uses a 39-dimensional feature vector consisting of 12 MFCC appended with unnormalised log energy, delta and delta-delta coefficients. The acoustic models are comprised of 11 whole-word digit models, each with 16 emitting states, 3 Gaussians per state; a 3-state silence model with 6 Gaussians per state; and single-state inter-word pause model tied to the middle state of the silence model. This gives a total of 546 acoustic model components. In this work, an internal version of HTK 3.3 along with its native front-end processing code, as opposed to the reference 2.2 version was used; only very minor differences occurred in the baseline performance.

To train the acoustic models, there are 8440 training utterances available, about 4 hours total, from 110 speakers, evenly split between gender. For matched training, 422 sentences, or about 720 seconds of audio, are provided for each of the 16 conditions: 4 different SNRs ranging from 20 to 5 dB, and with the 4 different additive noise sources N1 to N4: subway, babble, car and exhibition hall. This provides sufficient data to use SPR, reviewed in section 4.4.1, to generate matched models for this small acoustic model of 546 diagonal variance Gaussians. Each of the 16 matched conditions also has a test set of a 1001 sentences with 52 male and 52 female speakers—this is test set A. Test set B is another set of additive noise conditions for conducting unseen noise tests, and test set C for convolutional noise robustness experiments. Since for these Aurora2 experiments only clean acoustic models were tested, test set B was not used. Due to the extremely high error rates at SNRs below 5 dB, these conditions were also not tested. A convention on Aurora2 has been to average WER over the different SNRs. The large difference in magnitudes between error rates means large gains in noisier data can mask losses in cleaner conditions, hence results are reported for each SNR level and only averaged across the four different additive noise conditions in test set A. Lastly no results were reported on test set C since there is no degradation from the channel difference on the clean baseline tests.

Since the task is well defined, comparisons can be made against other compensation algorithms evaluated on this corpus. One criticism of Aurora2 is the average clean on clean baseline word error rate of 0.98% is not close to state-of-the-art performance. To address this, in 2002 a more complex Aurora2 recogniser back-end was specified [59], which increased the number of components per GMM to 20 from the standard 3; this improved what was a considered a weak baseline by 67.8% relative on clean-trained models and 19.2% on multistyle-trained models. However despite these gains, most results are still reported on the standard back-end models, thus this is also the case for this work.

8.1.1 Compensation Parameter Estimation

Compensation parameters were estimated using stereo data. The front-end uncertainty schemes all used diagonal transformations. The front-end GMM parameters required for some forms of noise estimation and 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. For the model-based forms, regression classes were produced using the HTK tool `HHEd` to perform

top-down clustering of model components with a Euclidean distance measure. Either SPR or the noise model estimation described in section 6 was used to estimate M-Joint transforms based on these classes.

8.1.2 Front-end Compensation

In 2001, it was found that SPLICE had the best noise removal performance at a special Aurora2 session in Eurospeech [26]; SPLICEU improved on this result. Given the similarity of FE-Joint to SPLICEU, these two algorithms are compared along with standard SPLICE. The original SPLICE and SPLICEU forms trained a *noisy* speech GMM to partition the acoustic space for each condition. However, it is unrealistic to assume that such a model is available *a priori* for any environment. An alternative, is to start with a *clean* speech GMM, and compensate it, for example using SPR, to each test condition given an estimate of the noise. These two forms of deriving the front-end GMM for partitioning the acoustic space are also explored.

	SNR(dB)			
System	20	15	10	5
Clean	4.6	12.2	31.1	59.2
Matched	1.8	2.8	5.0	11.4
Noisy Speech GMM				
SPLICE	2.0	3.1	6.1	16.5
+Uncertainty, $\alpha = 0.1$	2.2	3.2	6.0	14.5
FE-Joint, $\rho = 0.9$	1.8	2.9	5.7	14.6
Clean Speech GMM				
SPLICE	2.0	3.0	6.2	15.7
+Uncertainty, $\alpha = 0.1$	2.5	4.1	8.9	23.1
FE-Joint, $\rho = 0.9$	1.9	3.0	6.1	16.4

Table 8.1: WER (%) for 256-component front-end GMM schemes compensating clean models on Aurora2 test set A averaged across N1-N4.

Results are presented in table 8.1. As expected, the clean models show significant degradation as the noise level increases. Matched training, using SPR, gave substantial improvements by more than halving the error rate for all SNR levels evaluated. The results for SPLICE and SPLICEU using the directly estimated noisy speech GMM are comparable to those reported in the literature. While the SPLICEU only gives gains in higher noise, the FE-Joint technique exceeds standard SPLICE over all the SNRs evaluated. When the front-end GMM is derived from a clean speech GMM, the standard SPLICE scheme performance changes little, whereas SPLICEU in contrast is quite sensitive to the front-end GMM. There is significant degradation when using SPLICEU with clean speech derived GMM compared to the noisy. It is unclear why this is the case. Adjusting the α parameter failed to improve results. The FE-Joint form demonstrated less sensitivity but still some degradation.

8.1.3 Issue with Front-end Uncertainty Decoding

In section 5.2.2, a fundamental issue for all front-end uncertainty decoding schemes was discussed. If several frames have high uncertainty, the decoder may not have any acoustic information to discriminate between models. In table 8.2, this is clearly shown in the FE-

System		SNR(dB) 20	15	10	5
Clean		4.6	12.2	31.1	59.2
Matched		1.8	2.8	5.0	11.4
SPLICEU	$\alpha = 0.1$	2.2	3.2	6.0	14.5
	$\alpha = 0.95$	2.0	3.2	5.6	12.3
FE-Joint	—	22.7	25.8	28.4	34.4
	$\rho = 0.9$	1.8	2.9	5.7	14.6

Table 8.2: WER (%) for 256-component front-end UD schemes using noisy GMM and compensating clean models, varying parameter flooring, on Aurora2 test set A averaged across N1-N4.

Joint form results when no correlation flooring is applied—the error rates are all greatly increased, with the majority being insertions. Flooring the correlation to a minimum of 0.9 was effective in addressing this problem. The SPLICEU form also benefited from adjusting the α parameter which affects the magnitude of the uncertainty propagated to the decoder as discussed in section 5.2.2. With $\alpha = 0.95$ the SPLICE form still exhibited slightly more insertion errors than the floored FE-Joint form. Table 8.3 gives a more detailed view of how the flooring in FE-Joint compensation affects the errors generated. Without flooring, insertions make up the majority of the errors. With the correlations floored, they are reduced to a minority and below the number of substitution errors.

System		SNR(dB) 20	5
FE-Joint	—	420 (80%)	971 (70%)
	$\rho = 0.9$	19 (20%)	136 (30%)

Table 8.3: Number of insertions, % of total errors in parentheses, for 256-component FE-Joint compensation, varying ρ flooring, on Aurora2 N1 subway noise.

8.1.4 Model-based Compensation

Although front-end uncertainty decoding forms can give good performance, as the previous section demonstrated, the flooring parameters α and ρ conceal a fundamental problem: the uncertainty should be dependent on the clean speech model rather than solely on the feature vector. In contrast, the M-Joint form discussed in section 5.2.4 closely ties the corrupted speech conditional distribution to clean speech model component classes. The classes are generated as discussed in section 2.5.1. The first split is between speech and non-speech components and no minimum split threshold was applied.

Table 8.4 shows how the model-based JUD form compares to the front-end form. Here,

System	Number of Transforms	SNR(dB) 20	15	10	5
Clean	—	4.6	12.2	31.1	59.2
Matched	—	1.8	2.8	5.0	11.4
FE-Joint, $\rho = 0.9$	256	1.9	3.0	6.1	16.4
Diagonal Transformations					
M-Joint	1	3.3	5.9	13.4	32.0
	16	2.5	3.8	7.2	16.6
	256	1.9	2.7	5.2	12.0
Full Transformations					
M-Joint	1	2.4	3.8	7.0	17.1
	16	2.0	2.8	4.2	9.9

Table 8.4: WER (%) for diagonal and full matrix JUD compensation of clean models on Aurora2 test set A averaged across N1-N4.

transforms are either associated with regions of the clean acoustic space, for FE-Joint using the "clean GMM", or clusters of similar acoustic components in M-Joint. With only 16 transforms, the model-based form performs almost as well as the 256-transform front-end version. With the same number of transforms, and therefore compensation parameters, the M-Joint form is superior across all SNR to the FE-Joint form, as well as SPLICE and SPLICEU. Moreover, it is very close to the matched baseline theoretical upper limit illustrating the effectiveness of this technique.

As discussed in section 5.3, the M-Joint transform need not be diagonal—for example, block-diagonal and full matrix transforms are also possible. By modelling the correlations between features, results are greatly improved. A single, full transform performs as well as 16 diagonal ones, and 16 full transforms gives results that exceed matched system performance in higher noise, for example 9.9% WER compared to matched at 11.4% at 5 dB. A full M-Joint transform allows the system to model correlations between features introduced by the noise, which is more pronounced as the noise is greater. The full covariance decoding required to use such transforms is unfortunately computationally rather expensive. However, there are some approaches that are effective in addressing this [48].

8.1.5 Comparison with Other Techniques

Aurora2 allows some comparisons between techniques to be made although there are usually differences in the parameterisations. Still it is useful to get a rough idea how the forms investigated in this thesis compare to others in the literature. Unless noted, recognition is on MFCC parameters with delta and acceleration coefficients.

Table 8.5 compares a variety of different compensation schemes in the literature. The first set are the baselines previously given. The second set represent techniques that make few assumptions of the interfering noise. Histogram equalisation (HE) [139] demonstrates the performance of a normalisation scheme. It performs better than the data imputation with soft missing data approach [108], but does not do as well as the data imputation [142] scheme where the restored spectral features are transformed to the cepstral domain for recognition.As

System	SNR(dB)			
	20	15	10	5
Clean	4.6	12.2	31.1	59.2
Matched	1.8	2.8	5.0	11.4
Histogram equalisation	3.7	6.4	12.8	25.3
SMD, Imputation	3.9	8.2	12.5	43.5
SMD, Soft data marginalisation	3.0	7.3	11.5	26.5
Wiener filtering	2.1	3.7	8.3	20.4
Wiener filtering with obs. unc.	4.5	3.6	7.0	15.2
FE-Joint, $\rho = 0.9$	1.9	3.0	6.1	16.4
M-Joint	1.9	2.7	5.2	12.0

Table 8.5: WER (%) for various noise robustness techniques compensating clean models on Aurora2 test set A averaged across N1-N4. Non-JUD compensation forms quoted from various sources. Soft missing data (SMD) operated in spectral domain only.

expected the soft data marginalisation scheme [108], using a uniform evidence pdf, performed better than the imputation version, but not as well as the imputation scheme in the cepstral domain. This supports the findings in Raj and Stern [119]. Wiener filtering [10] assumes that the noise is additive. Strangely, using the Wiener filter in an observation uncertainty form reduced performance in higher SNR, but gave gains in lower SNR; this indicates a fundamental problem with observation uncertainty. Obviously, the JUD forms gave the best results—a large part of this is because the parameters are trained on stereo data. The HE, SMD and Wiener filtering techniques do not make the use of such data. A fairer comparison would be with M-Joint transforms estimated using a noise model estimated in an unsupervised manner from test data. Such a noise estimation technique will be discussed in the next section.

8.2 The Resource Management Corpus

Due to the simple acoustic models and task, it is questionable whether conclusions drawn from Aurora2 will carry over to more difficult tasks and systems with advanced acoustic modelling. Hence, more extensive exploration of these robustness techniques was conducted on the 1000-word naval ARPA Resource Management (RM) command and control task [117]. For this work, noise is artificially added at the waveform level from the NATO NOISEX-92 database [144]. The clean RM data was recorded in a sound-isolated room using a head mounted Sennheiser HMD414 noise-cancelling 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 16 kHz. 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 totalling 3.8 hours of data.

The 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 is simulated from

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

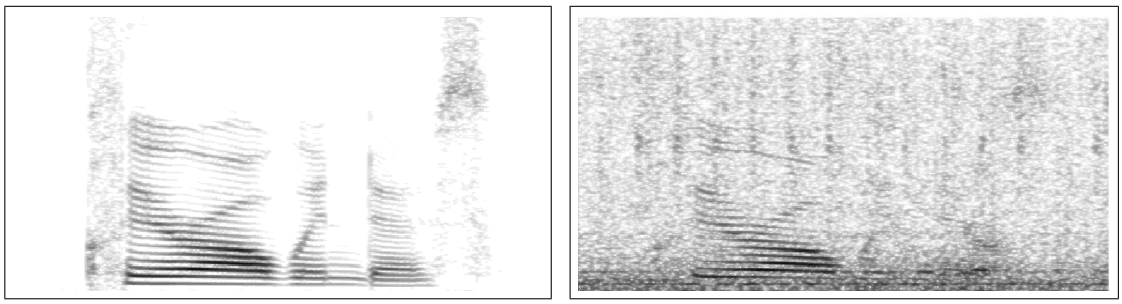


Figure 8.1: Clean spectrum (left) compared with corrupting Operations Room noise at 8 dB SNR (right) for the utterance"Clear all windows".

32 dB to 8 dB SNR. Figure 8.1 shows the how the noise affects one of the RM sentences, "Clear all windows". The noise itself has a dominant low frequency background hum, an unknown repetitive 6 Hz broadband noise of a machine, and intermittent speech. For unseen noise conditions, car noise was used from the interior of a Volvo 340 recorded while driving at 120 km/h, in 4th gear, on an asphalt road, in rainy conditions. The car noise is much more stationary than the Operations Room noise.

The baseline continuous speech recogniser was built using the RM recipe distributed with HTK [152]. The cross-word, gender-independent, acoustic models have 1582 decision tree clustered states, with six components per state totalling 9492 Gaussians. A simple bigram language model was used. All results are quoted as an average of three of the four available test sets, `Feb'89`, `Oct'89` and `Feb'91`; the `Sep'92` test data was not used. This gave a total of 30 test speakers and 900 utterances. A 39-dimensional MFCC feature vector was appended with delta and delta-delta coefficients. Unless otherwise noted all experiments were conducted with a pruning threshold of 300, grammar scale factor of 7 and inter-model transition penalty of 0. For initial experiments where SPR was used to obtain compensation parameters, a normalised log energy feature replaced C_0 (EDA parameterisation); this allows comparisons with SPLICE and to remove noise estimation from being a factor under consideration. The next set of experiments then examine systems where noise estimation was used to generate compensation parameters; instead of normalised log energy, these necessarily used C_0 (0DA parameterisation). The clean-trained acoustic model used the original RM training data base.

For multistyle training, a multi-condition database was artificially created by adding Operations Room noise on a per speaker level to the clean database at 8, 14, 20, 26 and 32 dB in equal proportion. Rather than creating the acoustic model using the full RM recipe with the multistyle data set, an initial multistyle model was obtained from the clean using SPR and stereo data. Four iterations of BW re-estimation were conducted from this initial model using the multi-condition training data to produce the final multistyle model. It should be noted that when matched SPR baselines are reported, these are always using acoustic models derived from the clean and never the multistyle.

8.2.1 Stereo Data Parameter Estimation

It is useful to compare the performance of uncertainty decoding with other compensation techniques. In table 8.6 a variety of compensation forms were evaluated against Operations Room noise at 20 dB on clean-trained acoustic models. A single set of parameters were estimated for this noise level using SPR—not at the speaker level; hence with about an

Compensation		20 dB SNR
None		33.2
Normalisation	CMN	26.6
	CMN+CVN	18.5
	Gaussianisation, 4 comp.	14.6
Feature-based	SPLICE	12.3
	SPLICEU	9.9
	FE-Joint	8.2
Model-based	CMLLR, full transforms	8.9
	M-Joint, diagonal transforms	8.2
	M-Joint, full transforms	7.4
Matched		7.2

Table 8.6: WER (%) for a variety of techniques compensating clean models on Operations Room corrupted RM task at 20 dB SNR (EDA). Compensation parameters at a global level. SPLICE and FE-Joint use "clean", 256-component front-end GMMs. Model-based forms use 16 transforms.

hour of adaptation data to train one set of transforms, parameters for these compensation schemes should be robustly estimated. A simple cepstral bias will reduce the error rate by about a quarter, while also normalising the cepstral variance slightly less than halves the error rate from 33.2% to 18.5%. With SPLICE, applying a cepstral bias based on one of 256 feature regions is better than cepstral normalisation or Gaussianisation. The two uncertainty forms, SPLICEU and FE-Joint using a "clean" front-end GMM, gave results better than the strictly feature compensating SPLICE form by 20% and 33% relative. The difference between SPLICEU and FE-Joint may be attributed to the different approaches in approximating the corrupted speech conditional distribution. Also, while correlation flooring for FE-Joint was necessary on the Aurora2 task, it was not for RM; although the same large feature scaling and uncertainty variance bias effects were observed, the bigram language model limited the number of insertion errors.

The FE-Joint compensation surpassed model-based CMLLR compensation. With 256 diagonal transforms the FE-Joint system had more free parameters available to estimate, at $(39 \times 3) \times 256 = 29952$ for 256 diagonal joint distributions, compared to 16 full CMLLR transforms with $(39 \times 39 + 39) \times 16 = 24960$ parameters. Each joint distribution has the noise speech mean, variance and the cross-covariance to estimate which is $39 \times 3 = 117$ parameters. However, the 16-diagonal transform M-Joint system has far less than either, at $(39 \times 3) \times 16 = 1872$, yet gave comparable performance to the more complex FE-Joint system and was better than CMLLR. This demonstrates the effectiveness of having an uncertainty bias on the model variances at this level of noise. Increasing the number of M-Joint transforms only gave a small improvement to 8.0% WER, whilst decreasing the number of GMM components in the FE-Joint system to 16 increased the error rate to 9.8%. The divide between FE-Joint and M-Joint performance is greater at 14 dB SNR; for 256 transforms, their error rates are 19.0% and 15.4% respectively and 17.1% for 16 diagonal M-Joint transforms. 16 full M-Joint transforms gave the best results, reaching matched performance although with the high cost of full covariance decoding. Unlike on Aurora2, a full transform M-Joint system did not exceed the matched

baseline. Clearly though, model-based uncertainty decoding is superior in terms of accuracy than the other techniques evaluated on this system at this level of noise.

8.2.1.1 Varying the Number of Front-end Components

Table 8.7 presents more detailed SPLICE, SPLICEU and FE-Joint compensation results where a each set of compensation parameters are trained using SPR for this specific noise level. SPLICE and SPLICEU provide an obvious comparison of enhancement with and without adding uncertainty to the model variances. Similar to SPLICE, FE-CMLLR choose a transform of the features in the front-end. However it has a more powerful ML affine transform rather than the simple bias used in SPLICE. Moreover, this makes it similar in form to the SPLICEU and FE-Joint schemes, but without the uncertainty bias. As expected, for all the schemes, with an increase in the number components in the front-end GMM, the error rate is reduced. Unlike with the Aurora2 results, results are only presented on these techniques while using a clean speech based front-end GMM. This is a more reasonable configuration than using the noisy-based front-end GMM since it is unlikely in reality that data will be available to directly estimate one for unseen test conditions.

System	With Uncertainty	# GMM Components			
		1	4	16	256
Clean	—	33.2			
SPLICE	No	24.6	20.7	17.0	12.3
FE-CMLLR		16.3	15.3	12.8	13.5
SPLICE	Yes	11.4	12.4	12.2	9.9
FE-Joint		10.7	9.2	9.8	8.2
Matched	—	7.2			

Table 8.7: WER (%) for feature-based techniques compensating clean models on Operations Room corrupted RM task at 20 dB SNR (EDA).

The single component front-end GMM with SPLICE should be equivalent to CMN applied at a global level, as reported in table 8.6. Both are static biases applied to the feature vector. There is a slight difference between the two results here, 24.6% compared to 26.6%, which is attributable to their different implementations. The hard, max approximation, given in equation (4.17), was found to be effective; in unreported experiments, 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, substantially better at 12.3% than global-level CMN plus CVN at 18.5%, but still far from the matched performance of 7.2%.

For all noise conditions tested from 32 to 8 dB, SPLICEU performs better than SPLICE [90], although only the 20 dB results are reported here. The uncertainty forms worked surprisingly well with few components in the front-end. Only a small gain is obtained from one component at 11.4%, to 9.9% with 256, for SPLICEU. With a single component, a constant global variance is propagated to the decoding process and the features updated with a fixed transform. This out-performed all the other normalisation techniques and also 256-component SPLICE, for example the single component FE-Joint had a WER of 10.7% compared to 12.3% for the best SPLICE form. Notice the large gain of 7.8% absolute by using a global affine transform

and fixed variance bias over a simple global CMN transform. The FE-Joint distribution uncertainty decoding algorithm generally gave better than the SPLICEU form across all tested noise conditions. Here at 20 dB, for the best configurations this was 9.9 % compared to 8.2%. These are still worse than the matched result of 7.2%. Overall the addition "uncertainty" to the decoding process proves beneficial.

8.2.1.2 Varying the Model-based System Complexity

Two forms of model adaptation were compared: CMLLR and M-Joint transforms. As FE-CMLLR and FE-Joint compensation provide a good contrast for transforms selected in the front-end, CMLLR and M-Joint compensation are a good contrast for transforms associated with model regression classes. CMLLR provides a useful baseline as an affine transformation of the features without the model variance bias. With only a single regression class, it represents a global linear transform of the feature vector trained for the noise condition and is equivalent to compensation with a single FE-CMLLR transform. With multiple regression classes, it functions conceptually as parallel front-ends, one for each regression class and associated transform. The M-Joint transforms derived are similar in form to CMLLR with diagonal matrices, but has an added variance offset to the models. While each CMLLR transform computes an ML affine transform between training and test for each model class, in M-Joint compensation a single ML Gaussian joint distribution between the clean and noisy speech is estimated from stereo data. It is from this joint distribution that the M-Joint transform is generated. The effect of the number of classes on accuracy is explored. M-Joint transforms also provide an interesting contrast to the FE-Joint scheme as it does not have this front-end component selection problem. Table 8.8 presents the M-Joint and CMLLR transform performance as a function of the number of transforms and their complexity at 20 dB SNR. Similar to the previous section, transforms are estimated using SPR on all data available for the noise condition.

System	Transform Structure	# of Transforms		
		1	4	16
Clean	—	33.2		
CMLLR	Diagonal	16.3	14.6	10.3
	Full	17.8	14.9	9.2
M-Joint	Diagonal	10.7	9.6	8.2
	Full	10.1	8.0	7.4
Matched	—	7.2		

Table 8.8: WER (%) for model-based techniques compensating clean models on Operations Room corrupted RM task at 20 dB SNR (EDA).

It is clear that for both CMLLR and M-Joint, performance increases proportionally with the number of transforms used. Diagonal M-Joint forms exceed either diagonal or full CMLLR transforms for the same number of transforms. This reflects the benefit of the variance bias in the M-Joint form for noise compensation. Simpler model-based CMLLR and M-Joint forms at 16 transforms perform better or as well as their respective, more complex, 256-component front-end forms, FE-CMLLR and FE-Joint indicating the benefits of a model-based approach. Comparing the results in table 8.7 with table 8.8, global diagonal CMLLR and FE-CMLLR

versions give the same WER of 16.3% which are equivalent just as a global diagonal M-Joint transform is equivalent to FE-Joint with one front-end GMM component, sharing the same unexpectedly robust performance of 10.7%. From there, there is a small incremental improvement, with 128 transforms giving a WER of 8.1%; with CMLLR, at 128 transforms the WER is 8.9%. In either case, when the number of transforms equals the number of acoustic model components, the performance should be the same as SPR matched baseline—this is not the case for their front-end forms as discussed in section 5.5. Despite the same upper limit, with low numbers of diagonal transforms M-Joint compensation is superior to CMLLR indicating the benefit of the model variance bias. The full M-Joint transform yields the best results in almost reaching the matched SPR baseline with a WER of 7.4%. As discussed in the Aurora2 results, this requires computationally expensive full covariance decoding to operate; in contrast, full matrix CMLLR transforms do not.

8.2.1.3 Computational Load

For a fixed beam width, the increase in model variances with uncertainty decoding may cause the number of active models during search to rise. This may reduce the number of search errors, but will increase the computational load. Hence, it is worth investigating the sensitivity of the results to the number of active models.

System	With Uncertainty?	Pruning Threshold	`Feb'89` WER	# Active Models
Clean	—	300	37.6	10153
		150	33.9*	1632
SPLICE, 256 transforms	No	300	14.4	4306
FE-Joint, 1 transform	Yes	300	11.3	19680
		150	11.4	4096
		100	11.7	1144
FE-Joint, 256 transforms	Yes	300	8.0	16417
		150	8.1	3320
		100	9.3	1005
M-Joint, 256 transforms	Yes	300	7.6	8938
		150	7.7	1447
		100	10.7	447
Matched	—	300	6.5	5535
		150	7.1	865

Table 8.9: WER (%) and average number of active models when compensating clean acoustic models on Operations Room corrupted RM `Feb'89` test set only at 20 dB SNR (EDA). *Not all sentences yielded a hypothesis.

Table 8.9 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 front-end GMM and shows how the refined clean speech estimate allows the recogniser to more efficiently perform model pruning during the search. The single component FE-Joint configuration was of interest because of its unexpectedly robust performance. At the standard pruning threshold of 300, FE-Joint

compensation causes a large increase in the number of models active in the recogniser. This is attributable to the selection of the corrupted speech conditional distribution in the front-end causing transformations of the acoustic model components to become similar to each other. But it can be seen that despite a two fold reduction in the pruning threshold and a sizable drop in the number of models evaluated, the WER is only slightly affected. The same behaviour is exhibited by the 256-diagonal transform M-Joint system. However the M-Joint system performs better, in terms of WER, than FE-Joint with fewer active models for the same pruning threshold. The gains found using joint uncertainty decoding are genuine and not due to the increase in models evaluated with the expanded model variances.

8.2.2 Noise Model Estimation

In chapter 6, unsupervised ML noise model estimation schemes were discussed for VTS and M-Joint compensation. These could be applied to both clean- and multistyle-trained acoustic models. It is expected that these ML models should perform better than acoustic models estimated from the additive background noise. Furthermore a noise model estimated from the background is only applicable for compensating clean models; on multistyle-trained models, noise is already present in the training and thus using an acoustic noise model not sensible. As discussed, it is also important to tailor the noise model estimation to the compensation scheme, e.g. VTS or M-Joint compensation. The specific procedure for estimating the noise models was discussed in section 6.5. These aspects of the noise model estimation process, along with varying the number iterations, supervision hypothesis, estimation data and the complexity of speech model used will be reviewed in this section. Necessarily, these results use a different form of feature vector where normalised log energy is replaced by C_0. Normalising the log energy can aid in robustness since the energy level will grow with increased additive noise. Hence this 0DA parameterisation will give a weaker baseline than EDA. However 0DA allows for compensation using the predictive techniques discussed in this work. Unless otherwise noted, the noise model will consist of single vectors of static coefficients for the channel and additive noise means and a diagonal additive noise variance. The dynamic coefficients for the noise means are set to zero. Thus a noise model has a total of 65 parameters.

8.2.2.1 ML Noise Model Estimation

The additive noise mean and variance may be estimated from the entire NOISEX-92 Operating Room noise waveform used to create the noise-corrupted speech test sets. In table 8.10, this approach is compared to the ML VTS noise model estimation process discussed in section 6.2, where a single global VTS noise model, without a convolutional noise estimate, is estimated for the entire test using the reference transcription in a supervised fashion. Clearly, the estimation scheme performs as well as using the "known" acoustic noise. The estimated additive noise mean is similar to the known value. For the known noise, there is little improvement from updating the delta-delta coefficients in addition to the static and delta coefficients. While the ML estimates of the static and delta noise variances are poorer than the known noise model, with the delta-delta variances compensated the ML approach appears better producing fewer substitutions and insertions. This would indicate that the ML estimate is countering some of the effects of the VTS approximation to the mismatch function especially for computing the delta-delta variances. The log-likelihood of the test data, using the reference hypothesis, for the known and ML noise model only differs slightly. Nevertheless, there

Coefficients Compensated		WER			Log-likelihood		
		VTS		SPR	VTS		SPR
Means	Variances	Known	ML Est		Known	ML Est	
—		38.0			-74.4		
$\boldsymbol{y}$		15.2	15.1	14.7	-67.6	-67.6	-67.5
$\boldsymbol{y}+\Delta\boldsymbol{y}$		12.6	12.2	11.8	-66.7	-66.7	-66.5
$\boldsymbol{y}+\Delta\boldsymbol{y}+\Delta^2\boldsymbol{y}$		9.8	10.2	9.0	-66.0	-66.0	-65.4
$\boldsymbol{y}$	$\boldsymbol{y}$	11.9	13.7	12.0	-67.3	-67.6	-66.8
$\boldsymbol{y}+\Delta\boldsymbol{y}$	$\boldsymbol{y}+\Delta\boldsymbol{y}$	8.7	10.8	8.3	-66.6	-66.9	-65.2
$\boldsymbol{y}+\Delta\boldsymbol{y}+\Delta^2\boldsymbol{y}$	$\boldsymbol{y}+\Delta\boldsymbol{y}+\Delta^2\boldsymbol{y}$	8.6	8.0	7.4	-66.4	-66.3	-63.6

Table 8.10: WER (%) and log-likelihood for VTS compensation of clean models on Operations Room corrupted RM task at 20 dB SNR (0DA) varying dimensions compensated and noise model estimation. Reference hypothesis used for global-level ML noise model estimation. No convolutional noise estimated.

is a 0.6% absolute WER difference between VTS compensation of all dimensions using the ML model and the matched baseline. This indicates some deficiency in VTS compensation such that it is unable to capture all the effects of the corrupting noise.

The results in table 8.10 obtained ML noise models in a supervised manner. In practice the reference supervision hypothesis is not available. For unsupervised ML noise model estimation, a recognition hypothesis is produced from an initial decoding pass over the test data. In addition the noise model may be estimated at a speaker level for a more precise estimate. Table 8.11 shows how the ML noise estimation procedure performs when using the recognition hypothesis to align the test data. With a 38% uncompensated error rate, the hypothesis degrades VTS compensation with the ML noise model by 1.6% absolute to 9.6%. There is little gain from estimating a purely additive noise model at the speaker level, indicating that the corrupting noise is relatively stationary. Although including a channel estimate does not give improvements at a global level, at a speaker-level it performs some basic speaker adaptation and improves VTS compensation with this model to 8.4% for noise estimation with a recognition hypothesis.

It was discussed in section 6.3 that the compensation used during the ML noise estimation procedure should be consistent with the compensation used during decoding. Table 8.12 compares ML VTS noise models with ML M-Joint noise models for M-Joint compensation.

Noise Model	Noise Est. Hypothesis	Estimation Level	
		Global	Speaker
None	—	38.0	
$\boldsymbol{\mu}_n$, $\boldsymbol{\Sigma}_n$	Reference	8.0	8.0
	Recognised	9.6	9.5
$\boldsymbol{\mu}_n$, $\boldsymbol{\Sigma}_n$, $\boldsymbol{\mu}_h$	Reference	7.9	6.8
	Recognised	9.6	8.4
Matched	—	7.4	

Table 8.11: WER (%) for VTS compensation of clean models on Operations Room corrupted RM task at 20 dB SNR (0DA) varying estimation level, noise model and hypothesis.

Acoustic Model	Compensation	Noise Est. Type	Test Set SNR		
			Clean	20 dB	14 dB
Clean	—		3.1	38.0	83.7
	M-Joint	VTS	3.1	10.1	35.3
		M-Joint	3.1	9.2	22.6
Multistyle	—		11.7	7.0	15.5
	M-Joint	VTS	9.0	8.6	15.9
		M-Joint	8.6	6.7	12.3
Matched	—		3.1	7.4	14.3

Table 8.12: WER (%) for 16-diagonal M-Joint compensation of clean and multistyle models, comparing noise estimation type, on Operations Room corrupted RM task at 20 dB SNR (0DA). Recognition hypothesis used for speaker-level ML noise model estimation.

As one would expect, there is no degradation in applying M-Joint compensation for clean conditions on clean acoustic models. Using either VTS or M-Joint noise models give large improvements on the noisy test sets. This difference is far more pronounced at 14 dB: there is a 50% relative increase in WER from 22.6 to 35.3%. Also note that despite a recognition hypothesis WER of 83.7%, the noise estimation procedure was able to produce effective models for compensation.

As discussed earlier, estimating a noise model from the background is not suitable for multistyle acoustic models. However, using this ML noise model estimation procedure can generate a model that allow predictive techniques to compensate multistyle acoustic models. This model no longer represents acoustic noise, but rather is a set of parameters that reduce the mismatch between training and test conditions for a given compensation form. As shown in table 8.12, on multistyle acoustic models there is substantial degradation from 7.0% to 8.6% by using the VTS noise model over no compensation. This highlights the need to match the noise model estimation procedure to the compensation that will be used in testing. The multistyle model seems to perform best at 20 dB, which is the average SNR of the multistyle training data; moreover, VTS and M-Joint compensation, with appropriate ML noise models, only provide modest gains of 0.5% (not shown in table 8.12) and 0.3% respectively over the uncompensated multistyle system. This indicates that there is minimal mismatch between the multistyle training and 20 dB test condition. Also at this noise level, the uncompensated multistyle acoustic model slightly outperforms the matched baseline of 7.4%, which is not the case for the other SNR levels. This may be due to increased variances of the background models since the multistyle acoustic models are trained using data with SNR in the range of 8 to 32 dB. For all conditions though, and either clean or multistyle acoustic models, M-Joint compensation performed better with an M-Joint noise model rather than a VTS noise model.

Overall, compensating multistyle acoustic models gave better results than compensating the clean models. This may be due to a more accurate recognition hypothesis—the WER of uncompensated clean models is over five times that of uncompensated multistyle. In section 6.1 it was discussed how the hypothesis, or the noise model, used for generating the complete data set may be updated in further iterations of the noise model estimation process. Table 8.13 shows how increasing the number of EM iterations or improving the recognition hypothesis affects performance. It shows that the estimation procedure described in section 6.5 was reasonable since the difference in WER between the first and second iterations is

Acoustic Model	Compensation	Noise Est. Hypothesis	WER Iteration 1	WER Iteration 2	Log-likelihood Iteration 1	Log-likelihood Iteration 2
Clean	—		38.0		-74.4	
	M-Joint	Reference	8.9	8.5	-66.3	-66.0
		Recog. 1	9.2	8.6	-67.0	-66.3
		Recog. 2	8.4	8.5	-66.2	-66.2
	VTS	Reference	6.8	6.6	-64.8	-64.4
		Recog. 1	8.4	7.7	-65.9	-65.3
		Recog. 2	7.4	7.2	-65.2	-64.7
Matched	—		7.4		-63.6	

Table 8.13: WER (%) and log-likelihood for 16-diagonal M-Joint and VTS compensation of clean models, varying number of EM iterations and updating hypothesis, on Operations Room corrupted RM task at 20 dB SNR (0DA). Speaker-level ML noise model estimation.

small for both M-Joint and VTS compensation. Reducing the hypothesis error rate from 38% to less than 10% gives an improvement for both M-Joint and VTS noise model estimation. However, the VTS noise model estimation is more sensitive to errors in the hypothesis than M-Joint noise model estimation. This is indicated by the small difference in WER between results using the reference and recognition hypotheses. With M-Joint, the WER is similar between supervised and unsupervised training after two iterations of EM, whereas with VTS there is still a 0.6% difference after updating the hypothesis and a 2nd iteration of EM. This is expected since in M-Joint compensation a transform and estimation statistics are shared amongst similar components. Recognition errors mostly occur between similar components so do not substantially affect M-Joint noise model training.

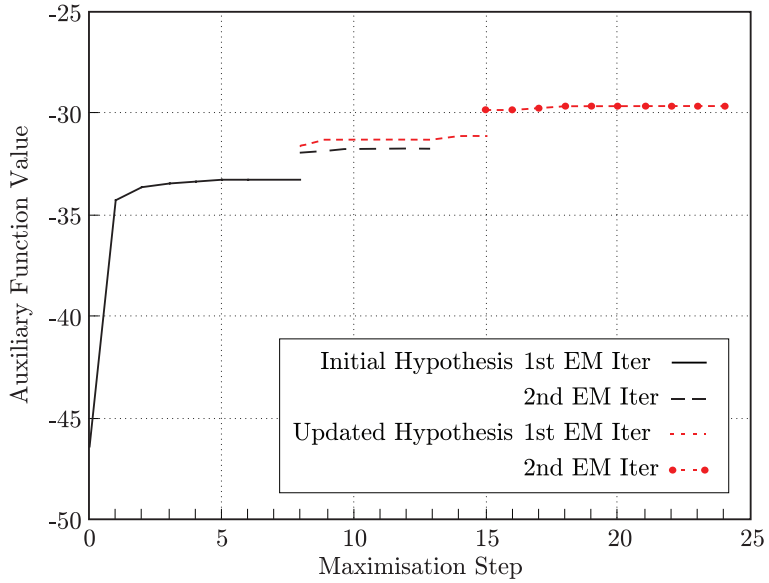


Figure 8.2: Graph of auxiliary function value during ML VTS noise model estimation.

To give more insight into the noise estimation procedure for the results shown in table 8.13,

the change in the auxiliary function value during VTS noise model estimation, with each step in the maximisation stage, is shown in figure 8.2. This is for a single speaker. In the first EM iteration, maximisation ends after seven steps. Each step involves the re-estimation of the noise means and variances as discussed in section 6.2. As illustrated in figure 6.1 of section 6.1, another iteration of EM may follow with the same hypothesis or an updated hypothesis produced using the updated noise model. The difference in auxiliary function values between at step seven is from the update of the complete data set. For either the same initial or an updated hypothesis, the maximisation stage fails to generate a large increase in the auxiliary function. Following the first EM iteration with the updated hypothesis, the maximisation step in the second iteration with the same updated hypothesis does not increase the auxiliary function much.

8.2.2.2 Noise Model Estimation Speed

The results conducted so far have used the full HMM speech model during noise model estimation. Alternatively a GMM may be used. If the GMM has far fewer components than the HMM, it may greatly increase the speed of all stages of noise model estimation as discussed in section 6.5. It also removes the need for a hypothesis for alignment and the associated initial decoding pass. However a GMM is less powerful than an HMM in capturing the temporal aspects of speech. Also using less adaptation data can improve estimation speed if fewer model components have data associated with them. Table 8.14 shows results using a GMM of

Recognition Acoustic Model	Noise Est. Acoustic Model		Noise Est. Hypothesis	Noise Est. Data	
				30 Utt.	1 Utt.
Clean	—			38.0	
	HMM		Reference	6.8	9.1
			Recognised	8.4	9.9
	GMM	256 comp.	—	7.9	8.6
		16 comp.	—	8.6	10.0
Matched	—			7.4	

Table 8.14: WER (%) for VTS compensation of clean models, varying noise estimation speech models, on Operations Room corrupted RM task at 20 dB SNR (0DA). Speaker-level ML noise model estimation.

varying complexity compared to the full HMM acoustic model during noise model estimation. The amount of adaptation data is also changed from all 30 utterances per speaker to just the first utterance per speaker. Estimation with a GMM is unexpectedly good considering that the noise model is optimised for VTS compensation of a GMM rather than the full HMM acoustic model. As was also shown in table 8.13 there is a difference of 1.6% absolute WER between using the reference hypothesis and the recognition for estimating the noise model with all 30 test utterances. Estimation with a 256-component GMM gives a WER of 7.9%, which is better than using an HMM with the recognition hypothesis, but not as good when the reference hypothesis is used. With only one utterance to estimate the noise model, the 256-component GMM surprisingly was more effective than estimation with a HMM regardless of the hypothesis used. This is perhaps to due to over-fitting the noise model for compensating the single utterance when using an HMM, whereas a GMM gives a more general model. The 16-component GMM was too simple and gave the worst results. Increasing the number of

components in the GMM to 1024 gave no improvements. These results indicate that when estimating noise models for compensating clean-trained acoustic models, the 256-component GMM is better than using the full HMM with a recognition hypothesis.

Compensation	Noise Est. Model	# of Params	Noise Est. Data	
			30 Utt.	1 Utt.
None	—		15.5	
CMLLR	HMM	0-1248	13.1	14.7
M-Joint	HMM	65	12.3	13.3
	GMM		12.7	12.8
VTS	HMM		12.0	13.3
	GMM		12.8	13.0
Matched	—		14.3	

Table 8.15: WER (%) for model-based compensation of multistyle models, comparing noise estimation speech model and amount of adaptation data, on Operations Room corrupted RM task at 14 dB SNR (0DA). Speaker-level parameter estimation. 16 diagonal transforms for M-Joint and CMLLR.

Table 8.15 shows how the amount of data available for noise estimation affects recognition performance for a multistyle trained acoustic model on 14 dB data. As in table 8.14, where in contrast a clean acoustic model is compensated, results are poorer when only a single utterance is available. However, on the multistyle the difference is negligible when using a GMM compared to an HMM for noise estimation. The predictive forms, i.e. M-Joint and VTS, are clearly less affected by having less data than the adaptive, i.e. CMLLR. For noise estimation with full HMM models, M-Joint suffers a 1% absolute loss, whereas the 16-diagonal transform CMLLR degrades by 1.6%. Since the average utterance is 3.4 seconds, or 340 frames, in length, a single full CMLLR transform, with 1560 free parameters, cannot be robustly estimated from one utterance. For diagonal CMLLR transforms systems using a regression tree, since the number of transforms varies with the amount of adaptation data available, the number of free parameters varies between 0 and 1248 (78 free parameters $\times$ 16 transforms). Typically only a single global transform is estimated since the minimum required number of training frames per transform, i.e. the split threshold, is set at 200. This explains the degradation in CMLLR performance such that there is only a 0.8% improvement over the uncompensated multistyle system. In contrast, the M-Joint system, even with a single utterance for noise model estimation was able to reduce the error rate by 2.2% absolute.

8.2.3 Joint Adaptive Training

It has been shown that compensating multistyle-trained acoustic models is more effective than compensating models trained on clean speech. Performance using multistyle models was close to the matched training. As an alternative to multistyle-training, chapter 7 presented adaptive training using M-Joint transforms. M-Joint transforms may be used to represent environmental noise during acoustic model training. The resulting JAT acoustic models may be purer representations of speech as they do not incorporate variability due to such noise. The initial model for JAT is the multistyle acoustic model. From this model four iterations of transform estimation and four iterations of model parameter re-estimation follow using

the interleaved training process shown in figure 7.1. For each successive model parameter iteration, the stabilising parameter ϑ, discussed in section 7.3.1, was reduced by increments of 0.5 starting from 2.5.

Acoustic Model	Compensation	Clean	Operations Room			Car
		47 dB	32 dB	20 dB	14 dB	20 dB
Clean	—	3.1	5.4	38.0	83.7	49.7
	M-Joint	3.1	4.4	9.2	22.6	8.0
Multistyle	—	11.7	5.3	7.0	15.5	43.5
	M-Joint	8.6	5.0	6.7	12.3	7.6
JAT	M-Joint	5.7	4.0	6.2	11.4	6.2
Matched	—	3.1	4.0	7.4	14.3	—

Table 8.16: WER (%) for 16-diagonal M-Joint compensation of clean, multistyle and JAT acoustic models, on clean and corrupted RM task (0DA). Recognition hypothesis used for speaker-level ML noise model estimation.

Table 8.16 presents the results of M-Joint compensation on clean, multistyle and adaptively trained acoustic models. Uncompensated multistyle models perform almost as well as matched models when tested on Operations Room noise since this is also what is used to corrupt the training data. However, these same multistyle models perform badly for unseen noise such as the clean test and car noise. It is clear that compensating multistyle models is superior to compensating clean models, on noisier data, but not the case at 32 dB SNR. This is not too far from the clean-training SNR of 47 dB, while M-Joint transforms are not actually appropriate for compensating multistyle models trained on data averaging 20 dB SNR to conditions less noisy at 32 dB. However, for all the noisier conditions the JAT system exceeds matched or compensated multistyle systems. While the gains over the compensated multistyle are small on Operations Room noise, there are larger gains on the unseen clean and car noise conditions not present in the multistyle training database. This demonstrates that JAT acoustic models are more amenable to being transformed to other noise conditions than multistyle models. Still, in a sense the models are not truly "clean" since adaptation to the clean condition gives a WER of 5.7%. This represents a substantial degradation compared to 3.1% WER with a matched clean system. However, such high SNR clean data is not normally available and data with some noise is more realistic. At 32 dB, JAT gives equivalent performance to matched at 4.0%. Hence JAT gives an effective method for training on heterogeneous data with varying noise levels.

JAT may be compared to other adaptive training techniques such as SAT with CMLLR. Two forms of SAT-CMLLR systems were built: the first used 16 diagonal CMLLR transforms and the second a pair of full matrix CMLLR transforms. A third NAT-CMLLR system used diagonal CMLLR transforms at a noise level rather than at a speaker level. Hence for each of the 5 main different SNR levels in the multistyle training database, only a single set of 16 diagonal transforms was estimated for each training iteration. Similar to JAT, these three systems were all trained using four iterations of interleaved transform and model parameter estimation steps as outlined in figure 2.14.

A comparison between JAT, SAT-CMLLR and NAT-CMLLR systems is presented in table 8.17. Clearly, the JAT system performs better than NAT-CMLLR demonstrating the effectiveness of the uncertainty variance bias term for noise adaptive training. However,

System	Transform Structure	Operations		Car
		20 dB	14 dB	20 dB
JAT	16 Diagonal	5.7	11.4	6.2
NAT-CMLLR	16 Diagonal	7.2	19.5	7.7
SAT-CMLLR	16 Diagonal	5.4	12.2	5.6
	2 Full	4.7	11.2	4.9

Table 8.17: WER (%) for JAT, NAT-CMLLR and SAT-CMLLR systems on Operations Room corrupted RM task (0DA). Recognition hypothesis used for speaker-level ML transform estimation.

compared to more standard SAT, at 20 dB JAT was worse than either transform structure for SAT-CMLLR. At 14 dB JAT was better than diagonal SAT, but marginally worse than SAT-CMLLR with full transforms. This again shows that the variance bias becomes more important as the SNR decreases. Since CMLLR transforms are effective for speaker adaptation, much of the gain of SAT-CMLLR over JAT may be due to the degree of speaker normalisation. For M-Joint transforms, the mismatch function used to predict the joint distribution does not address speaker differences, hence JAT cannot account for this factor very well. The NAT-CMLLR system compared to SAT-CMLLR, with diagonal transforms, shows the significant effect of accounting for the speaker for adaptive training. It was also expected that JAT would perform better than SAT-CMLLR on the car condition, but did not, indicating that CMLLR transforms may also train a relatively "clean" acoustic model.

8.2.4 Combined Systems

An important aspect for noise compensation schemes is how they interact with other techniques used in state-of-the-art ASR systems. This section presents experiments which combine M-Joint transforms with two such techniques: semi-tied and CMLLR transforms. Section 6.6 discussed how noise models may be estimated with a block-diagonal feature transform. Table 8.18 provides results when using a global, block-diagonal semi-tied transform for covariance

Acoustic Model	Compensation		Clean	Operations Room			Car
	STC	M-Joint	47 dB	32 dB	20 dB	14 dB	20 dB
Clean			3.1	5.4	38.0	83.7	49.7
		✓	3.1	4.4	9.2	22.6	8.0
	✓		2.8	5.2	41.9	84.0	74.1
	✓	✓	2.8	3.5	9.7	22.3	12.9
Multistyle			11.7	5.3	7.0	15.5	43.5
		✓	8.6	5.0	6.7	12.3	7.6
	✓		11.5	4.6	6.3	14.5	48.3
	✓	✓	7.6	4.5	6.1	11.7	9.7
Matched			3.1	4.0	7.4	14.3	—

Table 8.18: WER (%) for block-diagonal semi-tied transform combined with 16 diagonal M-Joint transforms with clean and multistyle acoustic models on Operations Room corrupted RM task (0DA). Recognition hypothesis used for speaker-level ML noise model estimation.

modelling in conjunction with M-Joint compensation.

First the interaction between M-Joint compensation and STC for clean acoustic models may be examined. The effectiveness of M-Joint compensation is more pronounced for the lower SNRs. While STC improves the matched clean baseline by 10% relative, there is no appreciable improvement on noisy tests. Combining M-Joint compensation with the clean-trained semi-tied system gave gains, although the results do not show a clear improvement over a non-STC clean acoustic model compensated with M-Joint transforms—e.g. at 20 dB, there is a loss from 9.2% for M-Joint compensation without STC, from figure 8.16 to 9.7% for M-Joint with STC on clean models. Hence for clean-trained acoustic models, using STC degrades the performance of M-Joint compensation.

For multistyle acoustic models, the addition of STC modelling improves performance by 0.7–1.0% for Operations Room noise. The semi-tied transform is capturing some of the correlations in the noise condition that are helpful when the test condition has the same noise characteristics. Moreover, this explains the 4.8% increase in WER on car noise over the non-STC system due to difference in the intra-frame correlations of the Operations Room noise in the multistyle training database. Adapting the multistyle STC system further with M-Joint transforms gives additional gains, which unlike what was seen on the clean models, are also better than the non-STC system with M-Joint transforms—e.g. at 14 dB, 11.7% to 12.3%. Nevertheless on the unseen car condition, M-Joint compensation performs worse due to the mismatched covariance modelling than compensating models without STC—i.e. 9.7% to 7.6%. The semi-tied transform embeds the training correlations in the acoustic models degrading recognition in unseen noise environments. Although combining a multistyle STC system with M-Joint compensation is effective, it does not perform as well as the diagonal variance JAT system; this is most clear for the unseen car condition when comparing between results given in tables 8.16 and 8.18. Alternatively, optimal feature space schemes may be a better approach to capture intra-frame correlations in changing, noisy environments [48].

Acoustic Model	Compensation		Clean	Operations	
	M-Joint	CMLLR	47 dB	20 dB	14 dB
Multistyle			11.7	7.0	15.5
		✓	5.0	5.8	13.1
	✓		8.6	6.7	12.3
	✓	✓	5.4	5.8	11.7
JAT	✓		5.7	6.2	11.4
	✓	✓	4.7	5.7	10.9
Matched			3.1	7.4	14.3

Table 8.19: WER (%) for 16-diagonal M-Joint with 2-full CMLLR compensation of multistyle and JAT models on Operations Room corrupted RM task (0DA). Recognition hypothesis used for speaker-level ML noise model estimation.

M-Joint compensated systems can be improved by further reducing mismatch between training and test conditions with CMLLR transforms. First, 16 diagonal M-Joint transforms are estimated for a noise condition. Subsequently, 2 full CMLLR transforms are estimated, using the M-Joint transformed feature space, to further adapt the system to a particular condition. Table 8.19 provides results for multistyle and JAT acoustic models, compensated with M-Joint and CMLLR transforms combined in such a manner. On the multistyle models,

although CMLLR transforms were more effective for the higher SNR conditions than M-Joint, at 14 dB M-Joint compensation proved better; in combination though they gave better results at low SNR. For example, adding the 2 CMLLR transforms improved M-Joint from 12.3% to 11.7% at 14 dB for multistyle model compensation. On the JAT models, the gains from combining M-Joint compensation with CMLLR are clearer: across all the conditions, there is gain in using CMLLR with JAT. The JAT models with CMLLR performed best across all the noisy test sets; although there was degradation on the clean test set over matched training, this is likely due to the lack of clean data in the multistyle training set.

8.3 Summary

The results in this chapter demonstrate that uncertainty decoding is an effective form of noise compensation on artificially corrupted small and medium vocabulary tasks. Two different forms of the conditional corrupted speech distribution selected in the front-end were examined: one modelled 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 normalisation schemes and a state-of-the-art enhancement technique such as SPLICE. The simple uncertainty variance bias provides measurable gains, especially in low SNR, over forms that only affect the observations such as FE-CMLLR and standard SPLICE. While feature-based uncertainty techniques such as SPLICEU and FE-Joint are effective, the model-based form of JUD, M-Joint, provides even better accuracy. It performs well with few transforms, and typically outperforms CMLLR when the mismatch is high such as when clean acoustic models are used or multistyle models on high noise. It was shown that front-end forms can exhibit problems in low SNR, especially if the language constraints on the search are weak. M-Joint, by having different variances biases for different regression classes, intrinsically avoids transforming all the models to the same noise model in low SNR.

An ML noise model estimation procedure was introduced that allows the simultaneous estimation of both the additive noise mean and variance and channel mean for both clean- and multistyle-trained acoustic models. This gives a powerful method to derive noise models for a specific speaker and noise condition even when there may be no background non-speech areas. On tests with VTS compensation, the noise models that were generated using this method were superior to the "known" noise estimated from the additive noise sample used to corrupt the test sets. Given a model of the noise, the complete joint distribution between the training and test conditions from a prior training speech class model can be predicted. M-Joint compensation particularly benefited from using an ML approach to estimating the noise model. Compensating acoustic models with these M-Joint transforms is much faster than VTS, yet gives almost the same level of accuracy. It was shown how M-Joint transforms may be integrated with other ASR techniques such as CMLLR and semi-tied transforms. Though multistyle systems were considerably more robust to noise than clean-trained models, adaptive training with M-Joint transforms called JAT provided superior results especially on unseen noise environments. While full CMLLR transforms may compensate for multiple factors such as noise and speaker, JAT specifically compensates for noise. Compared to noise adaptive training with CMLLR transforms, JAT was more effective.

CHAPTER 9

Experimental Results on Recorded Noisy Speech

This chapter presents results from experiments conducted on speech that is not artificially corrupted, but collected from environments where noise is already present in background. The Broadcast News task involves transcribing mostly prompted speech with a large, open vocabulary from actual aired news broadcasts. Another corpus was provide by Toshiba Research Europe Limited (TREL). This contains speech recorded in the office and in vehicles driving at various speeds. Users say phone numbers, city names and command and control requests. No stereo data was used to estimate any compensation parameters for these experiments. The noise models are estimated in the same manner as for the RM experiments.

9.1 Broadcast News Transcription

The system used here is a simplified version of the CU-HTK RT-03 BN-E system [81]. Acoustic models are trained in an ML fashion on approximately 143 hours of found data from recorded English broadcast news released by the LDC in 1997 and 1998. State-tied, cross-word triphone models were defined using decision tree clustering. This gave approximately 7000 distinct states, with each state modelled by 16 Gaussians, yielding about 110k acoustic model components. MFCC parameters were chosen over PLP, without CMN. While MFCC performance is comparable to PLP, the baseline system is slightly weaker without CMN. However cepstral normalisation makes it difficult to apply the predictive compensation forms. The same segmentation and clustering routines from the RT-03 system were used to provide homogeneous blocks of training and test data. Testing was conducted on the `bndev03` test set

Focus Condition	SNR(dB)	# Utts
Overall	26	32443
F0 – baseline broadcast speech	31	9948
F1 – spontaneous broadcast speech	23	6247
F2 – speech over telephone channels	28	1095
F3 – speech in the presence of background music	21	1385
F4 – speech under degraded acoustic conditions	26	9145
F5 – speech from non-native speakers	40	235
Fx – all other speech	22	4388

Table 9.1: SNR and number of utterances for focus conditions in test set `bneval98`.

totalling 2.5 hours of broadcast audio from news sources aired in January, 2001, and the `bneval98` test set comprised of 2.9 hours of usable audio from June, 1998. The `bneval98` test set is partitioned into different focus conditions such as read speech, spontaneous speech, acoustically degraded speech and non-native speech—this is useful to examine the effectiveness of predictive compensation in different conditions. The different focus conditions are described in table 9.1. There are large numbers of utterances in the F0, F1 and F4 conditions; these constitute over two thirds of the total number of utterances. The overall SNR of 26 is also fairly high compared to the testing conducted on AURORA and RM.

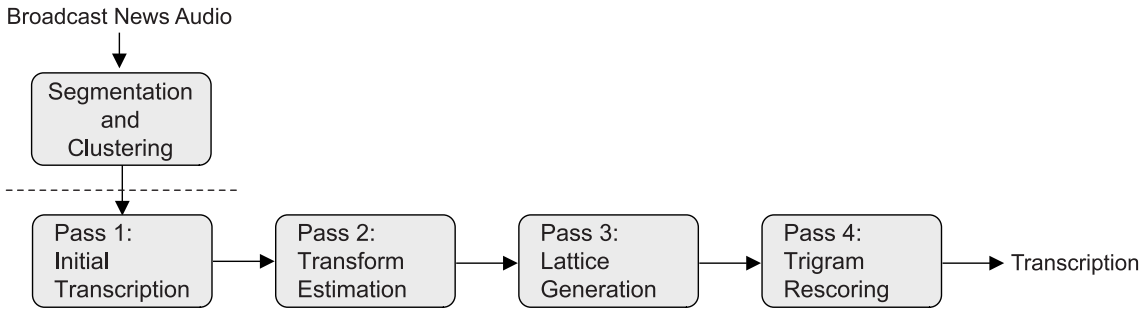


Figure 9.1: Broadcast News transcription system architecture.

The processing of BN audio to produce a transcription is shown in figure 9.1. An initial decoding pass over the test data was conducted to produce a 1-best hypothesised transcription. Another pass over the data is necessary to estimate transforms using this hypothesis. These transforms were used to perform decoding with a 59k-word dictionary and a bigram language model to generate lattices. A trigram language model then re-scores the lattices to find the final 1-best transcription. Only recognition on wide-band data was compensated using the techniques discussed; the same narrow-band results, from an uncompensated system, were used during scoring for all the systems described. A more complex system would typically use some form of feature projection scheme such as HLDA [87] or fMPE [116], advanced covariance modelling such as STC [41], and MMI [141] or MPE [115] training of model parameters—the use of such techniques were not investigated in these experiments.

9.1.1 Predictive Model Compensation

Table 9.2 shows results comparing 256 diagonal M-Joint transforms with VTS compensation. This number of transforms was used due to the significant difference in the number of acoustic model components in this system compared to the smaller ones in the previous chapter. With 16 transforms, M-Joint performance was about a half percentage point worse, yet 1024 transforms did not give gains over 256. On `bndev03`, M-Joint performed as well as VTS compensation. When the ML VTS noise model is used for M-Joint compensation, the WER increases by 0.3% to 19.1%. However, on the `bneval98` set, M-Joint is less effective than VTS compensation.

Compensation	`bneval98`	`bndev03`
—	21.2	20.8
M-Joint	19.0	18.8
VTS	18.5	18.8

Table 9.2: WER (%) for 256 diagonal M-Joint transform and VTS compensation of multistyle models on `bneval98` and `bndev03`.

Table 9.3 gives results for the `bneval98` test set broken down by the different focus conditions. On the cleaner broadcast news data (F0) there was only a modest reduction in WER of less than 1% absolute by using these predictive schemes. Despite the lower SNR of the F1 condition, the gain using the compensation forms was small. Much larger gains of more than 3% absolute WER were obtained on the noisier degraded acoustic condition F4. Nevertheless, the range in SNR is not large, and the difference between the training and test conditions minimal, which limits the gains possible by using predictive compensation. This explains why using 2 full CMLLR transforms is more effective, for example given error rates of 18.1% on `bndev03` and 18.3% on `bneval98`. This perhaps mirrors the performance of 2 full CMLLR transforms in compensating multistyle models on RM data corrupted at 20 dB, as was shown in table 8.19 of the previous chapter.

	`bneval98`							
Compensation	Overall	F0	F1	F2	F3	F4	F5	Fx
—	21.2	10.6	21.8	42.3	23.5	20.8	29.4	38.8
M-Joint	19.0	9.9	21.0	33.7	22.9	17.5	28.5	34.2
VTS	18.5	10.0	20.9	31.1	21.4	17.2	28.1	32.3

F0 – baseline broadcast speech
F1 – spontaneous broadcast speech
F2 – speech over telephone channels
F3 – speech in the presence of background music
F4 – speech under degraded acoustic conditions
F5 – speech from non-native speakers
Fx – all other speech

Table 9.3: WER (%) for 256 diagonal M-Joint transform and VTS compensation of multistyle models on `bneval98` broken down by focus condition.

9.1.2 Joint Adaptive Training

Table 9.4 compares using 256 diagonal M-Joint transforms to compensate multistyle-trained BN acoustic model against a JAT acoustic model estimated using the same number and type of M-Joint transforms. There was no gain in accuracy by using JAT. Unexpectedly, there was a slight degradation from 17.5% to 17.8% on the F4 "degraded acoustics" condition. Overall, the results using JAT on the BN task were disappointing, but this is due to the minimal difference in the conditions between the training and test data. Existing techniques such as CMLLR are sufficiently effective in reducing the mismatch.

Acoustic Model	Compensation	`bneval98`	`bndev03`
Multistyle	—	21.2	20.8
	M-Joint	19.0	18.8
JAT	M-Joint	18.9	18.9

Table 9.4: WER (%) for 256 diagonal M-Joint transform compensation of multistyle and JAT models on `bneval98` and `bndev03`.

9.2 Toshiba In-car Task

In 2004, Toshiba Research Europe Limited's Cambridge Research Laboratory (TREL-CRL) collected an internal corpus for noise robustness research. This corpus will be referred to as TREL-CRL04. It is a small/medium sized task with noisy speech collected in the office and in vehicles driving at various conditions. Three test sets that were available for use in this work are phone number, city name, and command and control recognition. The phone numbers are comprised of 80% in-country numbers and 20% international. The city name test consists of speakers saying one of 550 city names. The command and control task has speakers saying simple commands to operate various in-car functions.

Condition	Sub-condition	SNR(dB) μ	SNR(dB) σ
Office	—	34	3.5
In-car	Idle	35	5.7
	City	25	8.3
	Highway	18	5.2

Table 9.5: Average SNR level of TREL-CRL04 test set conditions.

Each task has a test set for four different conditions: office, idle, city driving and highway driving; the latter three are recorded in a vehicle. Results on these car conditions were reported on data collected from a rear-view mirror mounted microphone. The average SNR for each of the tests is shown in table 9.5. Although the office condition uses a close-talk microphone, the SNR is as about the same as the engine idling condition with the medium distance microphone. The SNR varies more in the city condition than in idle or highway conditions. The focus of this work will be on these three in-car conditions.

Table 9.6 provides additional details for each of the test sets. For the office data there were 20 speakers (10 male, 10 female), while for the in-car data there were 30 speakers (15 male,

Condition	Digits			City Names			Command & Control		
	μ	σ	n	μ	σ	n	μ	σ	n
Office	672	158	565	319	32.8	597	474	64.2	1194
Idle	656	158	824	304	37.3	928	452	67.2	1856
City	712	197	862	317	43.9	959	471	67.8	1915
Highway	726	179	898	328	41.8	988	492	70.1	1976
Total	694	178	3149	317	40.8	3472	472	69.3	6941

Table 9.6: Utterance length mean and standard deviation (frames) in TREL-CRL04 test sets. The # of utterances is denoted by n.

15 male) per session. Each provided about 30 utterances with twice as many utterances per speaker for the command and control task. All were native adult speakers of English. The digit task yielded the longest utterances on average while the city name utterances are the shortest. Notice that the mean length of the utterances increases for all the different tasks as the SNR decreases. This may be a reflection of hyper-articulation due to the Lombard Effect.

The test and training speech were parameterised in the following manner. The TREL-CRL04 data was recorded with 16-bit resolution at 16 kHz. Pre-emphasis was applied with a factor of 0.97. Magnitude spectra were obtained from a Hamming window and a filterbank of 24 channels. Applying the DCT gave MFCC of which the first 13 coefficients were used, including C_0, plus the first and second differentials. This yields a 39-dimensional feature vector. The WSJ SI284 training data was used to train a clean acoustic model in a similar manner to Woodland et al. [148]. There are 284 speakers from the WSJ0 and WSJ1 corpora yielding 66 hours of speech data. The acoustic models are decision tree clustered state, cross-word triphones, with three-emitting states per HMM, sixteen components per GMM and diagonal covariance matrices. In total there are just over 6000 states and 72k components.

A multistyle model was trained from this clean model using SPR with an artificially created stereo database of the clean and noise-corrupted multi-condition data. To generate the multi-condition training database, the WSJ SI284 corpus was artificially corrupted at a speaker/session level using in-car recordings from SpeechDat and Toshiba. The SpeechDat noise added was recorded at three different driving conditions: city, country and highway; and at Finland, Korea, Russia and Turkey. The Toshiba data comprised of one noise recording for each of the conditions plus an engine-on, idle recording. Hence the Toshiba idle condition data was used for corrupting a quarter of the training data, while a mixture of the SpeechDat and Toshiba noise data was used to corrupt the remainder of the training data. No samples from the actual evaluation data was used in the training. The noise to be added was selected at random intervals. The different noise recordings were scaled so that the average signal power is similar for each condition. Also, the WSJ0 training data was scaled down to a similar average power level to the WSJ1 data. The characteristics of the artificially created multi-condition training database are detailed in table 9.7. After SPR on the stereo data, the decision tree and state clustering were re-done to optimise them for noisy data and followed by another four iterations of BW re-estimation.

Decoding is conducted directly with a task-specific word network. The phone number digits task used an open digit loop. The city names grammar was a flat list of 550 cities. The command and control task was derived from the utterances in the data. A constant insertion penalty of -100 was used for the digits task and 0 for the others. The pruning threshold was

Condition	Portion of data	SNR μ	SNR σ
Idle	25%	35	5.5
City	25%	25	7.2
Country	25%	18	6.4
Highway	25%	15	5.8

Table 9.7: Summary of multistyle training data for TREL-CRL04 system, SNR in dB.

fixed at 400. As with RM, for adaptation and noise model estimation, an initial decoding pass was conducted on uncompensated clean or multistyle acoustic models to gives a recognition hypothesis. This is the baseline CUED configuration for this corpus.

9.2.1 Clean Acoustic Model Compensation

It was clear in the RM experiments that compensating multistyle trained acoustic models was more effective than compensating clean since the mismatch between training and testing conditions is less. Thus although many experiments for this corpus will focus on compensating multistyle and JAT models, some results for compensating clean models will be presented to evaluate the predictive techniques discussed on non-artificially corrupted test data. Table 9.8 provides some baseline experiments for clean-trained acoustic model compensation on the TREL-CRL04 corpus.

Compensation		Idle	City	H'way
None		2.7	33.8	64.4
CMLLR	2 Full	0.6	16.1	58.7
	16 Diag.	0.8	15.2	21.3
M-Joint		1.0	9.0	31.8
VTS		1.3	13.1	34.6

Table 9.8: WER (%) for CMLLR, 16-diagonal M-Joint and VTS compensation of clean models on TREL-CRL04 digits task. Recognition hypothesis used for speaker-level transform estimation with all test utterances.

It is clear that uncompensated clean acoustic models perform poorly. Based on the RM results, it is expected that simple feature normalisation schemes will give limited robustness. Adaptation provides a more powerful means to compensate the models to the test conditions for both the noise and speaker. Two forms of CMLLR were tested: two full matrix transforms, one for speech and one for silence, as it is often applied; and 16 diagonal transforms to better model the non-linear effect of noise. Transforms are estimated at a speaker-level with all available test data, which is about 3 minutes, for each speaker; this is sufficient for robust estimation of the transform parameters. As expected, when the noise level is low, i.e. the idle condition, 2 full CMLLR transforms are more effective than 16 diagonal transforms. As the mismatch between the training and test conditions grow, 2 full transforms performs more poorly—the most extreme example of this is clean models for highway with a compensated WER of 58.71%. With 16 transforms, CMLLR can more accurately model the non-linear effect of noise on the speech in a piece-wise fashion. Predictive M-Joint and VTS compensation

were also explored. The ML noise estimation procedure used for RM and BN are also used here; for all results, M-Joint noise model estimation was used to generate 16 transforms for M-Joint compensation. These results are less clear in comparison to CMLLR. Although CMLLR effectively compensates for both speaker and environment, it is expected that CMLLR should be worse than predictive forms in noisier conditions. In fact the WER of 21.3% is better than VTS or M-Joint at highways conditions. This may be explained by a sensitivity to the extremely poor recognition hypothesis, with an error rate of 64.4%. The poor alignment causes far more insertions in the predictive forms than with 16-diagonal CMLLR. In contrast, at the city condition the predictive forms are better.

9.2.2 Multistyle Acoustic Model Compensation

For the multistyle acoustic models, two forms of normalisation were examined: CMN+CVN and Gaussianisation with 4 components. These were conducted on a per utterance or per speaker level. The results in table 9.9 show that CMN and CVN performed poorly increas-

Compensation		Idle	City	H'way
None		1.9	5.8	19.4
CMN+CVN	Per Utt	2.4	7.5	12.8
	Per Spkr	2.7	6.8	12.0
Gaussianisation	Per Utt	2.5	6.5	9.4
	Per Spkr	3.3	7.2	11.0

Table 9.9: WER (%) for CMN+CVN and 4-component Gaussianisation with multistyle models on TREL-CRL04 digits task.

ing the WER by over 25% and 10% on the idle and city conditions compared to the uncompensated multistyle system although there were some gains on the highway condition. Gaussianisation generally did not improve on these results. The poor performance of these noise normalisation schemes might be due to the large SNR range in the multistyle training data as detailed in table 9.7. Part of the increase in errors may be attributed to a difference in the average utterance length that would affect the balance of speech and silence in the histograms; for WSJ0 and WSJ1 the average utterance length is 664 frames whereas there is variation around this mean in the test sets—table 9.6 showed a complete breakdown for the different test sets. The city names test has even shorter utterances, hence when Gaussianisation is applied, error rates in excess of 50% are obtained. Overall, the utterances in this task may simply be too short for normalisation to work effectively–too much discriminating information is lost due to the normalisation.

Alternatively, adaptation may be applied to improve performance. Table 9.10 provides results using CMLLR adaptation and predictive M-Joint and VTS compensation of multistyle models. The same methods to estimate the transforms were used as the previous section. Compared to the clean compensation results in table 9.8, with multistyle-trained models the mismatch is less and therefore the disparity between the full and diagonal results is smaller. As discussed for RM in section 8.2.2.2, the diagonal CMLLR transforms have many more free parameters than the predictive forms. In general, for both clean and multistyle systems, CMLLR outperformed the predictive M-Joint and VTS forms where on RM the opposite was found. This may be because CMLLR transforms are more flexible and can better adapt to the

Compensation		Idle	City	H'way
None		1.9	5.8	19.4
CMLLR	2 Full	0.6	1.9	5.1
	16 Diag.	0.8	1.8	3.5
VTS		1.8	3.4	5.9
M-Joint		1.6	3.0	4.8

Table 9.10: WER (%) for CMLLR, 16-diagonal M-Joint and VTS compensation of multistyle models on TREL-CRL04 digits task. Recognition hypothesis used for speaker-level transform estimation with all test utterances.

speaker; although the M-Joint and VTS compensation have more powerful model variance updates than CMLLR, they are constrained to the noise model. Unexpectedly, VTS was worse than M-Joint compensation. This may be due to sensitivity to initial noise model or differences between VTS noise model estimation and M-Joint noise model estimation.

Hypothesis	EM Iter	Idle	City	H'Way
—		1.9	5.8	19.4
Recognition	1	1.8	3.4	5.9
	2	1.8	3.3	5.3
Reference	1	1.7	3.4	5.4
	2	1.7	3.2	5.1

Table 9.11: WER (%) for VTS compensation of multistyle models, varying supervision mode and number of EM iterations, on TREL-CRL04 digits task. Noise model estimated at a speaker level with all utterances.

Table 9.11 investigates how VTS performance improves with further EM iterations to refine the noise model. Results with supervised noise model estimation are also presented for contrast. As expected results improve when a second EM iteration is conducting using the same recognition hypothesis. Though the gains are small on the idle and city conditions, the second iteration gives performance close to supervised estimation on the highway condition. Results are also better when the reference hypothesis is used, but not by much. This would indicate that the noise model estimation procedure is not very sensitive to errors in the hypothesis. Overall, the reported M-Joint compensation performance is still superior to any of these VTS model compensation results.

Compensation	Noise Est. Type	Idle	City	H'Way
—		1.9	5.8	19.4
M-Joint	VTS	1.6	3.3	7.6
	M-Joint	1.6	3.0	4.8
VTS	VTS	1.8	3.4	5.9
	M-Joint	1.7	3.0	4.1

Table 9.12: WER (%) for 16-diagonal M-Joint and VTS compensation of multistyle models, varying the noise estimation type, on TREL-CRL04 digits task. Recognition hypothesis used for speaker level noise model estimation with all test utterances

Table 9.12 examines M-Joint and VTS compensation with ML M-Joint or VTS noise models. As discussed previously, the compensation form should match the noise model estimation type. As expected the matched M-Joint compensation and noise models generally gave better results as was shown on RM. However, when VTS model compensation is performed with the ML M-Joint noise models, the results are better than with the VTS "tuned" noise models. This can be explained by the limited noise estimation procedure using VTS compensation. The statics fixed-point estimation only optimises the noise model for the static dimensions. The M-Joint noise model estimation uses numerical gradients to optimise the noise model for all dimensions. The similarity between VTS and M-Joint compensation allows VTS compensation to effectively use the M-Joint noise model. In fact, if the number of regression classes is increased to the number of model components, then M-Joint noise model estimation can be used to estimate VTS noise models since the compensation converge to be the same. The VTS results in table 9.12, using the ML M-Joint noise models, are the best of all predictive compensation schemes tested for multistyle acoustic models across all conditions. This was not the case on the RM task though; the best performance was obtained when the compensation matched the noise model estimation type.

The previous results use all the available adaptation data from the test set to estimate the compensation parameters. In some circumstances, such as interactive dialogue systems, the system should be compensated quickly such that accuracy is relatively high early in the interaction and latency in the system response is low. Thus results will now be presented where only the *first* utterance in the dialogue session is used to estimate the compensation parameters for rest of the session. Utterances average 6.9 seconds in length, hence full CMLLR transforms cannot be reliable estimated. Multiple diagonal CMLLR transforms can still be reliably estimated with a regression tree as applied in the RM task. In addition to evaluating CMLLR, M-Joint and VTS compensation, experiments with PCMLLR transforms will be conducted. PCMLLR has the same decoding form as CMLLR, however the transforms are estimated using predicted statistics from M-Joint transforms as discussed in section 5.8. Hence it is also a predictive compensation form, but without a model variance update as compared to M-Joint or VTS compensation.

Compensation	City		Highway	
	30 Utt	1 Utt	30 Utt	1 Utt
None	5.8		19.4	
CMLLR	1.8	8.4	3.5	8.8
PCMLLR	3.1	5.0	5.2	6.6
M-Joint	3.0	5.2	4.8	6.9

Table 9.13: WER (%) for CMLLR, PCMLLR or M-Joint compensation of multistyle models on TREL-CRL04 digits task comparing estimation with all utterances to only one utterance per speaker. Recognition hypothesis used for speaker-level 16-diagonal transform estimation.

Table 9.13 examines estimation with only the first utterance for each speaker compared with all 30. Only city and highway results are reported since the differences for the idle condition are small. While there is some expected degradation in performance when limiting the adaptation data, it is more substantial for CMLLR than the predictive forms. The WER more than quadruples on the city condition and doubles on the highway task. With the regression tree split threshold set at 200, transforms should be robustly estimated, although

on average only 3 transforms are produced per speaker since only a single utterance is available for adaptation. The transforms do not generalise well to the other utterances for the speaker, decreasing the overall likelihood of all test data. In contrast, PCMLLR transforms are estimated from M-Joint transforms and therefore a noise model. The noise model may be robustly estimated on little data. Thus PCMLLR and M-Joint perform better than CMLLR with limited adaptation data. It is surprising to find that PCMLLR gives similar performance to M-Joint. This is likely due to the mismatch between the multistyle acoustic models and the actual test conditions being not too large. For lower SNR conditions, this trend is unlikely to continue.

9.2.3 Joint Adaptive Training

This section examines JAT for noise robustness on this task. The training is conducted in the same manner as for the RM and BN systems. To achieve faster noise model estimation, limiting the adaptation data and using a GMM for noise model estimation during testing is also investigated for JAT and compared with the multistyle model results.

Acoustic Model	Compensation		Digits		
	M-Joint	CMLLR	Idle	City	H'way
Multistyle			1.9	5.8	19.4
	✓		1.6	3.0	4.8
	✓	✓	0.7	1.5	2.7
JAT	✓		1.1	2.4	4.5
	✓	✓	0.6	1.4	3.0

Table 9.14: WER (%) for 16-diagonal M-Joint combined with 2 full CMLLR transforms compensating multistyle and JAT models on TREL-CRL04 digits task. Recognition hypothesis used for speaker-level ML noise model estimation using all test utterances.

Table 9.14 gives results for multistyle compensation compared with JAT in combination with CMLLR. JAT was superior to the multistyle by about 0.5% absolute WER for all conditions. It was expected that gains would be larger for the highway condition just as they were for the city names task and the lower SNR conditions on RM in table 8.16, but this was not the case. This may be due to the higher average levels of noise in the multistyle training data for this task. The greatest relative reduction in error was in the idle condition at 30%. These results demonstrate that the JAT model is more amenable to being transformed to other conditions compared to the multistyle models. As was found on RM in table 8.19, the CMLLR transforms complemented M-Joint. The most substantial gains were on the idle condition where error rates are more than halved compared to compensation with only M-Joint transforms. There is large difference because this condition is less noisy than the average multistyle training SNR which is less appropriate for the M-Joint form to compensate. The mismatch function fundamental to generating M-Joint transforms represents combining clean speech with noise to give noisier speech, not cleaner speech. CMLLR transforms are not restrained in this manner. In combination the two provide results better than simply using CMLLR alone, as reported in table 9.10, especially on the highway condition.

The amount of adaptation data can also be reduced when testing the JAT system. Although during training, all the data are used to estimate transforms, for testing only the first

Acoustic Model	Compensation	Idle		City		Highway	
		30 Utt	1 Utt	30 Utt	1 Utt	30 Utt	1 Utt
Multistyle	—	1.9		5.8		19.4	
	M-Joint	1.6	1.7	3.0	5.2	4.8	6.9
JAT	M-Joint	1.1	1.1	2.4	3.2	4.5	4.7

Table 9.15: WER (%) for 16-diagonal M-Joint compensation on TREL-CRL04 digits task comparing estimation with all utterances to only one per speaker. Recognition hypothesis used for speaker-level ML noise model estimation.

utterance is used for noise model estimation. Results for this scenario are shown in table 9.15. Although there is reduced performance, between using all test data and only one, the overall results are still good. For example, using JAT with transforms estimated with only one test utterance was more effective than multistyle transforms estimated with all adaptation data in the idle and highway conditions. The large difference in the city condition, between estimating noise from 30 utterances or a single one for both compensated multistyle and JAT models, may be due to the more non-stationary nature of the noise in city driving.

For this task, one aspect of the noise model estimation process that has not been explored yet is the form of the acoustic model used during estimation. The JAT acoustic model is unchanged—only the noise model estimation for testing differs. The clean speech GMM is derived using the same approach described in section 5.4.1 to produce the clean speech class model. M-Joint transforms are generated using the VTS noise model estimated with the GMM. This is a limitation since there is a mismatch between the noise model estimation type, VTS, and the compensation the model will be used for, M-Joint. It was demonstrated on RM, BN and in table 9.12 for this task that it is important to match these.

Acoustic Model	Compensation	Idle		City		Highway	
		HMM	GMM	HMM	GMM	HMM	GMM
Multistyle	—	1.9		5.8		19.4	
	M-Joint	1.7	1.8	5.2	6.5	6.9	11.3
JAT	M-Joint	1.1	1.5	3.2	6.0	4.7	7.6

Table 9.16: WER (%) for 16-diagonal M-Joint compensation on TREL-CRL04 digits comparing HMM or GMM speech model for noise model estimation. Recognition hypothesis used for speaker-level ML noise model estimation using only the first utterance per speaker.

Table 9.16 provides results comparing GMM versus HMM speech models for noise model estimation on the first test utterance per speaker. With the 256-component GMM, M-Joint transforms are generated using the ML VTS noise model. Based on the experiments with VTS compensation on RM, 16 components are expected to be too few. Experiments on the multistyle system with a 1024-component GMM did not give improve improvements. Unlike the RM results in table 8.15 with VTS compensation, M-Joint performance degrades substantially when a GMM is used compared to the HMM. Part of this may be attributed to the mismatch between the VTS noise model estimation and the M-Joint compensation. The JAT results are still better than the multistyle, however on the city condition, the results with using a GMM for noise model estimation are worse than not compensating the multistyle models. This is likely due to the cumulative effects of estimating the noise model with a GMM

and only with a single utterance. Results should improve if the noise model is updated more frequently than once every 30 utterances.

Acoustic Model	Compensation	City Names		
		Idle	City	H'way
Multistyle	—	11.1	17.8	39.1
	M-Joint	9.5	15.4	22.8
JAT	M-Joint	9.4	14.8	16.9

Table 9.17: WER (%) for 16-diagonal M-Joint compensation of multistyle or JAT models on TREL-CRL04 city names task. Recognition hypothesis used for speaker-level noise model estimation with all test utterances.

Table 9.17 gives results comparing M-Joint compensation with multistyle or JAT acoustic models, but on the city names task to demonstrate that these results carry beyond a digit recognition task. The results are reported using all available adaptation data to estimate the noise models. Table 9.14 shows that performance levels on the city names task is worse than on digits, reflecting the increased task difficulty, and the gains from compensation less substantial. For predictive multistyle model compensation on digits at highway noise levels, the WER was reduced by about four times, whereas on city names it is not halved with M-Joint compensation. With the JAT models, results are only marginally better for the idle and city conditions, but more substantial on the highway. The WER is more than half of the uncompensated multistyle model. The results from table 9.14 and 9.17 show that JAT is an effective form of improving noise robustness.

9.3 Summary

This chapter has presented results examining uncertainty decoding on recorded noisy speech that is not artificially corrupted. Two corpora were examined: the large vocabulary Broadcast News task and unpublished Toshiba 2004 database. Experiments on Broadcast News demonstrated that predictive M-Joint and VTS compensation forms can successfully be applied to a large vocabulary task. Since the SNR did not vary by large amounts, joint adaptive training did not give any gains. The Toshiba 2004 database contains drivers saying phone numbers, city names, and commands whilst driving at various speeds. While the experimentation on this database focused on the phone number digits task, performance trends were similar for the other tasks. Many of the conclusions from RM testing were confirmed. While M-Joint compensation improved results for clean- and multistyle-trained acoustic models across all conditions, the best results were obtained using joint adaptive training. Experiments also showed that with little adaptation data, M-Joint compensation can still be effective compared with CMLLR compensation. Lastly some experiments were conducted using a GMM for noise model estimation rather than a HMM to improve estimation speech and were effective for the noisier conditions. Overall, positive results were obtained on these tasks involving speech recorded in noisy environments.

Chapter 10
Conclusions

This thesis has investigated uncertainty decoding for noise robust speech recognition. In particular, a new approach called joint uncertainty decoding (JUD) was introduced. JUD compensation parameters are derived in a straightforward manner from the joint distribution between the training and testing conditions. An important contribution is the discussion of inherent limitations of front-end uncertainty decoding forms like SPLICE with uncertainty and front-end JUD. The third major contribution is a detailed presentation of maximum likelihood noise model estimation for noise compensation. Lastly, the final contribution is noise adaptive training with JUD transforms, which is referred to as joint adaptive training or JAT. JAT directly takes into account the noise level of observations by de-weighting them in proportion to the uncertainty; noisier observations contribute less to the canonical model estimation leading to purer speech acoustic models. In conclusion, these contributions demonstrate the overall effectiveness of JUD as a competitive noise robustness technique for a wide variety of tasks, small to large vocabularies, clean- or multistyle-trained acoustic models, and on large range of SNR and real world data.

The next section reviews the key findings in this thesis in more detail and the last section in this chapter presents future work directions.

10.1 Summary of Results

JUD is a form of uncertainty decoding where the joint distribution between the training and test conditions is considered Gaussian. In comparison to other techniques, such as SPLICE with uncertainty or observation uncertainty, this naturally leads to more powerful non-diagonal transformations. If joint distributions are associated with different regions of the acoustic space, i.e. partitioning with a front-end GMM, then this results in a form of

front-end JUD referred to as FE-Joint compensation in this work. Alternatively if the joint distributions are linked to different regression classes of acoustic model components, then this results in model-based JUD, which is referred to as M-Joint compensation.

When the number of classes equals the number of model components. M-Joint compensation has the property of converging to whatever model compensation technique, such as VTS, PMC or SPR, is used to derive the joint distribution. However, by reducing the number of classes, the computational cost is lowered by a significant factor with little loss in accuracy. Efficiencies are attained in transform estimation and application by sharing transforms across a group of model components much like with MLLR adaptation. It was shown that with only 16 or 256 diagonal transforms, compared to 10k-100k model components, performance similar to VTS compensation could be achieved. Furthermore, the form of the M-Joint transform, which is a simple affine feature transformation and model variance bias, is far cheaper to apply than VTS, which requires full matrix multiplication of the model variances.

Additionally, the number of transforms is not restricted to the amount of adaptation data since M-Joint transforms are predicted by combining noise and speech models. Experiments also show that transforms can be more robustly estimated on less data than an adaptive form such as CMLLR. Results on the AURORA and Resource Management tasks demonstrate that JUD is effective over a wide range of SNR and superior to other noise compensation techniques such as standard SPLICE, SPLICE with uncertainty, and CMLLR. On the Toshiba tests, JUD gave results similar to VTS compensation and reduced the error rate substantially on the noisiest conditions. Hence, it may be concluded that from small to large vocabulary tasks, over a variety of artificially added noises and real conditions like the Toshiba In-car task, JUD is a fast, efficient yet effective, form of model-based noise compensation.

This thesis has provided a clarification between front-end uncertainty decoding and observation uncertainty. For the noise robustness framework used in this work, if the front-end chooses a global model mean and variance update—this is the definition of a front-end uncertainty decoding form—then problems in low SNR will arise. In contrast, observation uncertainty simply adds the variance of the feature enhancement process to the acoustic model variances; however, the literature has not provided a strong mathematical motivation for this approach. This may explain the larger than expected variances that reduce performance in cleaner conditions among the many other problems quoted using this technique. Hence the distinction between uncertainty decoding and observation uncertainty is a crucial one when assessing their advantages and limitations.

Another important contribution is the discussion of an inherent problem for all front-end uncertainty decoding forms such as, but not limited to, SPLICE with uncertainty and FE-Joint compensation. If a frame has low SNR, then the front-end will consider the region noise and choose a transformation that updates a model component to the noise distribution. Since a transformation is global, all the model components will be transformed to the same noise distribution. Hence in low SNR there may be no acoustic information available for the decoder. If this occurs for several frames, then errors in the search can arise. A language model can alleviate this problem, however in some tasks, such as recognising digit strings, it can be rather weak. In comparison model-based approaches do not have this problem since models are affected by different transforms. Moreover, the M-Joint variance update can be cached if the noise is stationary; otherwise M-Joint compensation has a similar computational cost to front-end uncertainty decoding or observation uncertainty for a comparable number of compensation parameters. Compared to the front-end forms, experiments on AURORA and Resource Management showed that model-based compensation provided superior results

especially with fewer transforms. These aspects lead to the conclusion that model-based uncertainty decoding is superior to front-end uncertainty decoding.

A ML noise model estimation framework using EM for state-of-the-art recognisers was put forward in this thesis. Some past work in this area has been limited to only static features, the log-spectral domain or models for enhancement. Here, the approach is used for model compensation forms such as M-Joint and VTS compensation and furthermore allows them to be applied to both clean and multistyle acoustic models. It was shown that using an ML noise model was superior to using an acoustic additive noise model computed from non-speech regions. Although this latter approach may be computationally efficient, the detection of speech may become more difficult and prone to error as the SNR becomes low. An EM approach can overcome this problem, and provides a noise model consistent with the noise compensation technique. Experimental results allow it to be concluded that ML noise model estimation is a useful technique for improving model-based noise compensation.

The last main contribution is noise adaptive training using JUD transforms called JAT. Instead of forcing the acoustic models to represent extraneous variability introduced by noise in the training data, as is the case for multistyle training, the noise effect is modelled by JUD transforms. Adaptive training with CMLLR or normalisation updates the features and subsequently treats cleaner observations the same as noisier ones. In contrast, during acoustic model training, JAT directly takes into account the noise level of observations by de-weighting them in proportion to the uncertainty—noisier observations contribute less to the canonical model estimation than cleaner ones. The resulting acoustic models are then purer representations of the speech variability. No gains were observed on the Broadcast News task because the mismatch between the training and test conditions is minimal. On the Resource Management and Toshiba tasks it was shown that joint adaptive training was superior to compensating clean or multistyle-trained acoustic models. The difference is most apparent on test conditions where the noise is not present in the multi-condition training data. Hence compared to multistyle training, JAT is a better, albeit more complex, method of training acoustic models on heterogeneous data with mixed levels of noise for noise robust speech recognition.

10.2 Future Work

The majority of this work focused on diagonal transforms. As the noise level increases, it becomes more important to model the correlations introduced by the noise. Using stereo data with artificially corrupted noise, full joint distributions were estimated giving results exceeding matched performance while compensating clean acoustic models—this demonstrates the gains possible by improving correlation modelling. It would be advantageous to extend the M-Joint transform and noise model estimation procedure to produce full, or at the least, block-diagonal transforms. A block-diagonal form could easily be derived by not diagonalising the Jacobian matrices or the predicted corrupted speech variance even though the additive noise variances are diagonal. This could be improved by estimating non-diagonal forms of additive noise covariance.

A goal of this work was to produce an efficient form of noise compensation. Although M-Joint compensation is more efficient than model-based VTS, the noise model estimation procedures used were not very efficient. For the ML M-Joint noise model, numerical derivatives were used. Analytic forms of these derivatives should be derived. As the noise level

increases, many models are effectively subsumed by the noise and could be tied to a single noise distribution to improve efficiency. Also, preliminary experiments in this work demonstrated that noise models could be estimated on only a single utterance with a GMM speech model. It would be interesting to develop a sub-utterance noise model estimation technique that would truly allow rapid adaptation to noisy environments.

Additional steps can be taken to improve the overall noise compensation capability of JUD. A major assumption in this work is that the noise is stationary. However, gains have been shown by introducing more complex noise models used for model-based compensation that handle non-stationary noises like babble speech [86] or machine gun noise [39]. Similarly M-Joint compensation can be extended to handle such situations by considering multi-state noise models. Each noise state may be associated with a set of JUD transforms. The noise state can be efficiently determined by associated each state with a component in a front-end GMM.

Furthermore, the joint distribution for M-Joint transforms has been assumed Gaussian, however it can be clearly non-Gaussian depending on the SNR and speech and noise variances. More complex forms of the joint distribution and hence the corrupted speech conditional should be explored to see how performance is affected by the single Gaussian approximation. The joint distribution has also been predicted using a noise mismatch function. If other factors, such as the speaker, can be captured in the joint distribution between the training and test conditions, then uncertainty decoding can be extended beyond noise compensation. This of course could complicate the JAT process. However, it could permit fast adaptation of an ASR system to new environments and speakers using a single set of predictive transforms that are estimated from small amounts of data and can be compactly stored.

Appendix A
Useful Derivations

This section contains simple, useful derivations involving multivariate Gaussian distributions and random vectors.

A.1 The Conditional Multivariate Gaussian

Let $\boldsymbol{s}$ and $\boldsymbol{o}$ be multivariate Gaussian distributed variables with mean parameters $\boldsymbol{\mu}_s$ and $\boldsymbol{\mu}_o$, and covariance matrices $\boldsymbol{\Sigma}_s$ and $\boldsymbol{\Sigma}_o$ respectively. The joint distribution of these two random vectors can be considered Gaussian distributed

$$\mathrm{p}(\boldsymbol{s}, \boldsymbol{o}) \sim \mathcal{N}(\boldsymbol{\mu}_{s,o}, \boldsymbol{\Sigma}_{s,o}) \tag{A.1}$$

where the mean and variance are given by

$$\boldsymbol{\mu}_{s,o} = \begin{bmatrix} \boldsymbol{\mu}_s \\ \boldsymbol{\mu}_o \end{bmatrix} \qquad \text{and} \qquad \boldsymbol{\Sigma}_{s,o} = \begin{bmatrix} \boldsymbol{\Sigma}_s & \boldsymbol{\Sigma}_{so} \\ \boldsymbol{\Sigma}_{os} & \boldsymbol{\Sigma}_o \end{bmatrix} \tag{A.2}$$

and $\boldsymbol{\Sigma}_{so}$ and $\boldsymbol{\Sigma}_{os}$ are the cross covariances between $\boldsymbol{s}$ and $\boldsymbol{o}$, and $\boldsymbol{\Sigma}_{so} = \boldsymbol{\Sigma}_{os}^{\mathsf{T}}$.

Bayes' rule dictates that

$$\mathrm{p}(\boldsymbol{s}|\boldsymbol{o}) = \frac{\mathrm{p}(\boldsymbol{s}, \boldsymbol{o})}{\mathrm{p}(\boldsymbol{o})} \tag{A.3}$$

If $\mathrm{p}(\boldsymbol{s}, \boldsymbol{o})$ is Gaussian, then the conditional PDF of $\boldsymbol{s}$ given $\boldsymbol{o}$ is also Gaussian

$$\mathrm{p}(\boldsymbol{s}|\boldsymbol{o}) = \mathcal{N}\big(\boldsymbol{\mu}_{s|o}, \boldsymbol{\Sigma}_{s|o}\big) \tag{A.4}$$

where

$$\boldsymbol{\mu}_{s|o} = \boldsymbol{\mu}_s + \boldsymbol{\Sigma}_{so}\boldsymbol{\Sigma}_o^{-1}(\boldsymbol{o} - \boldsymbol{\mu}_o) \tag{A.5}$$

$$\boldsymbol{\Sigma}_{s|o} = \boldsymbol{\Sigma}_s - \boldsymbol{\Sigma}_{so}\boldsymbol{\Sigma}_o^{-1}\boldsymbol{\Sigma}_{os} \tag{A.6}$$

A similar form can also be derived for $\mathrm{p}(\boldsymbol{o}|\boldsymbol{s})$. $\boldsymbol{\Sigma}_{s|o}$ is also referred to as the Schur decomposition of $\boldsymbol{\Sigma}_{s,o}$ w.r.t. $\boldsymbol{\Sigma}_o$ and denoted by $\boldsymbol{\Sigma}_{|\Sigma_o}$. A full derivation may be found in [122].

A.2 Convolution of Two Gaussian Distributions

In the marginalisation over the clean speech variable $\boldsymbol{s}_t$, an integral of this form appears

$$\begin{aligned}\mathrm{p}(\boldsymbol{o}_t|m,k) &= \int_{\mathcal{R}^D} |\boldsymbol{A}^{(k)}|\mathcal{N}\Big(\boldsymbol{A}^{(k)}\boldsymbol{o}_t + \boldsymbol{b}^{(k)}; \boldsymbol{s}_t, \boldsymbol{\Sigma}_{\mathsf{b}}^{(k)}\Big)\mathcal{N}\Big(\boldsymbol{s}_t; \boldsymbol{\mu}_s^{(m)}, \boldsymbol{\Sigma}_s^{(m)}\Big)\mathrm{d}\boldsymbol{s}_t \\ &= \int_{\mathcal{R}^D} |\boldsymbol{A}^{(k)}|\mathcal{N}\Big(\boldsymbol{A}^{(k)}\boldsymbol{o}_t + \boldsymbol{b}^{(k)} - \boldsymbol{s}_t; \boldsymbol{0}, \boldsymbol{\Sigma}_{\mathsf{b}}^{(k)}\Big)\mathcal{N}\Big(\boldsymbol{s}_t; \boldsymbol{\mu}_s^{(m)}, \boldsymbol{\Sigma}_s^{(m)}\Big)\mathrm{d}\boldsymbol{s}_t\end{aligned} \tag{A.7}$$

where $\boldsymbol{o}_t$ is the noisy observation vector at frame t, m indexing the model component parameters in the acoustic model $\mathcal{M}$ and k the transform index in the compensation parameter set $\check{\mathcal{M}}$. This integration can be considered the convolution of the two Gaussian distributions

$$\mathrm{p}(\boldsymbol{o}_t|m,k) = |\boldsymbol{A}^{(k)}|\mathcal{N}\big(\boldsymbol{A}^{(k)}\boldsymbol{o}_t + \boldsymbol{b}^{(k)} - \boldsymbol{s}_t; \boldsymbol{0}, \boldsymbol{\Sigma}_{\mathsf{b}}^{(k)}\big) * \mathcal{N}\big(\boldsymbol{s}_t; \boldsymbol{\mu}_s^{(m)}, \boldsymbol{\Sigma}_s^{(m)}\big) \tag{A.8}$$

The convolution of two Gaussian distributions results in another Gaussian distribution with a mean that is the sum of their means and a variance that is the sum of their variances [27]

$$\mathrm{p}(\boldsymbol{o}_t|m,k) = |\boldsymbol{A}^{(k)}|\mathcal{N}\Big(\boldsymbol{A}^{(k)}\boldsymbol{o}_t + \boldsymbol{b}^{(k)}; \boldsymbol{\mu}_s^{(m)}, \boldsymbol{\Sigma}_s^{(m)} + \boldsymbol{\Sigma}_{\mathsf{b}}^{(k)}\Big) \tag{A.9}$$

To prove this, consider the product of two Gaussians

$$\begin{aligned}&\mathcal{N}(\boldsymbol{y} - \boldsymbol{x}, \boldsymbol{\Sigma}_1)\mathcal{N}(\boldsymbol{x} - \boldsymbol{\mu}, \boldsymbol{\Sigma}_2) = \\ &\qquad \frac{1}{(2\pi)^D\sqrt{|\boldsymbol{\Sigma}_1||\boldsymbol{\Sigma}_2|}}\exp\left\{-\frac{1}{2}\Big[(\boldsymbol{y} - \boldsymbol{x})^{\mathsf{T}}\boldsymbol{W}_1(\boldsymbol{y} - \boldsymbol{x}) + (\boldsymbol{x} - \boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{W}_2(\boldsymbol{x} - \boldsymbol{\mu})\Big]\right\}\end{aligned} \tag{A.10}$$

where $\boldsymbol{W}_1 = \boldsymbol{\Sigma}_1^{-1}$ and $\boldsymbol{W}_2 = \boldsymbol{\Sigma}_2^{-1}$. In the square brackets, $\boldsymbol{x}$ can be brought out and isolated

$$\begin{aligned}&(\boldsymbol{y} - \boldsymbol{x})^{\mathsf{T}}\boldsymbol{W}_1(\boldsymbol{y} - \boldsymbol{x}) + (\boldsymbol{x} - \boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{W}_2(\boldsymbol{x} - \boldsymbol{\mu}) \\ &\qquad = \boldsymbol{y}^{\mathsf{T}}\boldsymbol{W}_1\boldsymbol{y} - 2\boldsymbol{x}^{\mathsf{T}}\boldsymbol{W}_1\boldsymbol{y} + \boldsymbol{x}^{\mathsf{T}}\boldsymbol{W}_1\boldsymbol{x} + \boldsymbol{x}^{\mathsf{T}}\boldsymbol{W}_2\boldsymbol{x} - 2\boldsymbol{x}^{\mathsf{T}}\boldsymbol{W}_2\boldsymbol{\mu} + \boldsymbol{\mu}^{\mathsf{T}}\boldsymbol{W}_2\boldsymbol{\mu} \\ &\qquad = \boldsymbol{x}^{\mathsf{T}}(\boldsymbol{W}_1 + \boldsymbol{W}_2)\boldsymbol{x} - 2\boldsymbol{x}^{\mathsf{T}}(\boldsymbol{W}_1\boldsymbol{y} + \boldsymbol{W}_2\boldsymbol{\mu}) + \boldsymbol{y}^{\mathsf{T}}\boldsymbol{W}_1\boldsymbol{y} + \boldsymbol{\mu}^{\mathsf{T}}\boldsymbol{W}_2\boldsymbol{\mu}\end{aligned} \tag{A.11}$$

For simplicity, define $\boldsymbol{W}_s = (\boldsymbol{W}_1 + \boldsymbol{W}_2)$, such that

$$\begin{aligned}&(\boldsymbol{y} - \boldsymbol{x})^{\mathsf{T}}\boldsymbol{W}_1(\boldsymbol{y} - \boldsymbol{x}) + (\boldsymbol{x} - \boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{W}_2(\boldsymbol{x} - \boldsymbol{\mu}) \\ &\qquad = \boldsymbol{x}^{\mathsf{T}}\boldsymbol{W}_s\boldsymbol{x} - 2\boldsymbol{x}^{\mathsf{T}}(\boldsymbol{W}_1\boldsymbol{y} + \boldsymbol{W}_2\boldsymbol{\mu}) + \boldsymbol{y}^{\mathsf{T}}\boldsymbol{W}_1\boldsymbol{y} + \boldsymbol{\mu}^{\mathsf{T}}\boldsymbol{W}_2\boldsymbol{\mu} \\ &\qquad = \boldsymbol{x}^{\mathsf{T}}\boldsymbol{W}_s\boldsymbol{x} - 2\boldsymbol{x}^{\mathsf{T}}(\boldsymbol{W}_1\boldsymbol{y} + \boldsymbol{W}_2\boldsymbol{\mu}) + (\boldsymbol{W}_1\boldsymbol{y} + \boldsymbol{W}_2\boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{W}_s^{-1}(\boldsymbol{W}_1\boldsymbol{y} + \boldsymbol{W}_2\boldsymbol{\mu}) + \\ &\qquad\quad \boldsymbol{y}^{\mathsf{T}}\boldsymbol{W}_1\boldsymbol{y} + \boldsymbol{\mu}^{\mathsf{T}}\boldsymbol{W}_2\boldsymbol{\mu} \qquad\qquad - (\boldsymbol{W}_1\boldsymbol{y} + \boldsymbol{W}_2\boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{W}_s^{-1}(\boldsymbol{W}_1\boldsymbol{y} + \boldsymbol{W}_2\boldsymbol{\mu})\end{aligned} \tag{A.12}$$

The first three terms in equation (A.12) can be factored as follows

$$\boldsymbol{x}^{\mathsf{T}}\boldsymbol{W}_s\boldsymbol{x}-2\boldsymbol{x}^{\mathsf{T}}(\boldsymbol{W}_1\boldsymbol{y}+\boldsymbol{W}_2\boldsymbol{\mu})+(\boldsymbol{W}_1\boldsymbol{y}+\boldsymbol{W}_2\boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{W}_s^{-1}(\boldsymbol{W}_1\boldsymbol{y}+\boldsymbol{W}_2\boldsymbol{\mu}) = (\boldsymbol{x}-\hat{\boldsymbol{x}})^{\mathsf{T}}\boldsymbol{W}_s(\boldsymbol{x}-\hat{\boldsymbol{x}}) \quad \text{(A.13)}$$

if $\hat{\boldsymbol{x}} = \boldsymbol{W}_s^{-1}(\boldsymbol{W}_1\boldsymbol{y}+\boldsymbol{W}_2\boldsymbol{\mu})$. The next two terms in equation (A.12) can be expanded as follows

$$\begin{aligned}\boldsymbol{y}^{\mathsf{T}}\boldsymbol{W}_1\boldsymbol{y} + \boldsymbol{\mu}^{\mathsf{T}}\boldsymbol{W}_2\boldsymbol{\mu} &= \boldsymbol{y}^{\mathsf{T}}(\boldsymbol{W}_1+\boldsymbol{W}_2)\boldsymbol{W}_s^{-1}\boldsymbol{W}_1\boldsymbol{y} \;+\; \boldsymbol{\mu}^{\mathsf{T}}(\boldsymbol{W}_1+\boldsymbol{W}_2)\boldsymbol{W}_s^{-1}\boldsymbol{W}_2\boldsymbol{\mu} \\ &= \boldsymbol{y}^{\mathsf{T}}\boldsymbol{W}_1\boldsymbol{W}_s^{-1}\boldsymbol{W}_1\boldsymbol{y} + \boldsymbol{y}^{\mathsf{T}}\boldsymbol{W}_2\boldsymbol{W}_s^{-1}\boldsymbol{W}_1\boldsymbol{y} + \\ &\quad \boldsymbol{\mu}^{\mathsf{T}}\boldsymbol{W}_1\boldsymbol{W}_s^{-1}\boldsymbol{W}_2\boldsymbol{\mu} + \boldsymbol{\mu}^{\mathsf{T}}\boldsymbol{W}_2\boldsymbol{W}_s^{-1}\boldsymbol{W}_2\boldsymbol{\mu} \end{aligned} \quad \text{(A.14)}$$

and the final term

$$\begin{aligned}&(\boldsymbol{W}_1\boldsymbol{y}+\boldsymbol{W}_2\boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{W}_s^{-1}(\boldsymbol{W}_1\boldsymbol{y}+\boldsymbol{W}_2\boldsymbol{\mu}) \\ &\qquad = \boldsymbol{y}^{\mathsf{T}}\boldsymbol{W}_1^{\mathsf{T}}\boldsymbol{W}_s^{-1}\boldsymbol{W}_1\boldsymbol{y} - 2\boldsymbol{y}^{\mathsf{T}}\boldsymbol{W}_1^{\mathsf{T}}\boldsymbol{W}_s^{-1}\boldsymbol{W}_2\boldsymbol{\mu} + \boldsymbol{\mu}^{\mathsf{T}}\boldsymbol{W}_2^{\mathsf{T}}\boldsymbol{W}_s^{-1}\boldsymbol{W}_2\boldsymbol{\mu}\end{aligned} \quad \text{(A.15)}$$

Since $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ are symmetric and therefore $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are as well, subtracting equation (A.15) from equation (A.14) gives

$$\begin{aligned}&\boldsymbol{y}^{\mathsf{T}}\boldsymbol{W}_1\boldsymbol{y} + \boldsymbol{\mu}^{\mathsf{T}}\boldsymbol{W}_2\boldsymbol{\mu} - (\boldsymbol{W}_1\boldsymbol{y}+\boldsymbol{W}_2\boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{W}_s^{-1}(\boldsymbol{W}_1\boldsymbol{y}+\boldsymbol{W}_2\boldsymbol{\mu}) \\ &\qquad = \boldsymbol{y}^{\mathsf{T}}\boldsymbol{W}_2\boldsymbol{W}_s^{-1}\boldsymbol{W}_1\boldsymbol{y} + \boldsymbol{\mu}^{\mathsf{T}}\boldsymbol{W}_1\boldsymbol{W}_s^{-1}\boldsymbol{W}_2\boldsymbol{\mu} - 2\boldsymbol{y}^{\mathsf{T}}\boldsymbol{W}_1^{\mathsf{T}}\boldsymbol{W}_s^{-1}\boldsymbol{W}_2\boldsymbol{\mu} \\ &\qquad = (\boldsymbol{y}-\boldsymbol{\mu})^{\mathsf{T}}(\boldsymbol{W}_1^{-1}+\boldsymbol{W}_2^{-1})^{-1}(\boldsymbol{y}-\boldsymbol{\mu})\end{aligned} \quad \text{(A.16)}$$

using the following identity $\boldsymbol{A}(\boldsymbol{A}+\boldsymbol{B})^{-1}\boldsymbol{B} = \boldsymbol{B}(\boldsymbol{A}+\boldsymbol{B})^{-1}\boldsymbol{A} = (\boldsymbol{A}^{-1}+\boldsymbol{B}^{-1})^{-1}$ found in Searle [126]. Equations (A.13) and (A.16) combine to form equation (A.12)

$$\begin{aligned}&(\boldsymbol{y}-\boldsymbol{x})^{\mathsf{T}}\boldsymbol{W}_1(\boldsymbol{y}-\boldsymbol{x}) + (\boldsymbol{x}-\boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{W}_2(\boldsymbol{x}-\boldsymbol{\mu}) \\ &\qquad = (\boldsymbol{x}-\hat{\boldsymbol{x}})^{\mathsf{T}}\boldsymbol{W}_s(\boldsymbol{x}-\hat{\boldsymbol{x}}) + (\boldsymbol{y}-\boldsymbol{\mu})^{\mathsf{T}}(\boldsymbol{W}_1^{-1}+\boldsymbol{W}_2^{-1})^{-1}(\boldsymbol{y}-\boldsymbol{\mu})\end{aligned} \quad \text{(A.17)}$$

and therefore

$$\mathcal{N}(\boldsymbol{y}-\boldsymbol{x},\boldsymbol{\Sigma}_1)\mathcal{N}(\boldsymbol{x}-\boldsymbol{\mu},\boldsymbol{\Sigma}_2) = \mathcal{N}(\boldsymbol{y}-\boldsymbol{\mu},\boldsymbol{\Sigma}_1+\boldsymbol{\Sigma}_2)\mathcal{N}(\boldsymbol{x}-\hat{\boldsymbol{x}},\boldsymbol{W}_s^{-1}) \quad \text{(A.18)}$$

This result allows the convolution to be simplified as follows

$$\begin{aligned}\int_{\mathcal{R}^D}\mathcal{N}(\boldsymbol{y}-\boldsymbol{x},\boldsymbol{\Sigma}_1)\mathcal{N}(\boldsymbol{x}-\boldsymbol{\mu},\boldsymbol{\Sigma}_2)\mathrm{d}\boldsymbol{x} &= \int_{\mathcal{R}^D}\mathcal{N}(\boldsymbol{y}-\boldsymbol{\mu},\boldsymbol{\Sigma}_1+\boldsymbol{\Sigma}_2)\mathcal{N}(\boldsymbol{x}-\hat{\boldsymbol{x}},\boldsymbol{W}_s^{-1})\mathrm{d}\boldsymbol{x} \\ &= \mathcal{N}(\boldsymbol{y}-\boldsymbol{\mu},\boldsymbol{\Sigma}_1+\boldsymbol{\Sigma}_2)\end{aligned} \quad \text{(A.19)}$$

Thus it is clear that for $\boldsymbol{y} = \boldsymbol{A}^{(k)}\boldsymbol{o}_t + \boldsymbol{b}^{(k)}$, $\boldsymbol{\mu} = \boldsymbol{\mu}_s^{(m)}$, $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_{\mathsf{b}}^{(k)}$ and $\boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}_s^{(m)}$

$$\begin{aligned}\mathrm{p}(\boldsymbol{o}_t|m,k) &= \int_{\mathcal{R}^D}|\boldsymbol{A}^{(k)}|\mathcal{N}\Big(\boldsymbol{A}^{(k)}\boldsymbol{o}_t+\boldsymbol{b}^{(k)}-\boldsymbol{s}_t;\boldsymbol{0},\boldsymbol{\Sigma}_{\mathsf{b}}^{(k)}\Big)\mathcal{N}\Big(\boldsymbol{s}_t;\boldsymbol{\mu}_s^{(m)},\boldsymbol{\Sigma}_s^{(m)}\Big)\mathrm{d}\boldsymbol{s}_t \\ &= |\boldsymbol{A}^{(k)}|\mathcal{N}(\boldsymbol{A}^{(k)}\boldsymbol{o}_t+\boldsymbol{b}^{(k)};\boldsymbol{\mu}_s^{(m)},\boldsymbol{\Sigma}_{\mathsf{b}}^{(k)}+\boldsymbol{\Sigma}_s^{(m)})\end{aligned} \quad \text{(A.20)}$$

Alternatively, the convolution of two probability "component" densities yields the density function for the sum of two random variables that are distributed according the two respective component densities [27]. If these two random variables are $\boldsymbol{w}_t \sim \mathcal{N}\big(\boldsymbol{0},\boldsymbol{\Sigma}_{\mathsf{b}}^{(k)}\big)$ and $\boldsymbol{v}_t \sim \mathcal{N}\big(\boldsymbol{\mu}_s^{(m)},\boldsymbol{\Sigma}_s^{(m)}\big)$ then

$$\boldsymbol{o}_t = \boldsymbol{w}_t + \boldsymbol{v}_t \sim \mathcal{N}\Big(\boldsymbol{\mu}_s^{(m)},\boldsymbol{\Sigma}_s^{(m)}+\boldsymbol{\Sigma}_{\mathsf{b}}^{(k)}\Big) \quad \text{(A.21)}$$

after noting the distribution of the sum two Gaussian distributed variables has a mean that is the sum of the component means and a variance the sum of the component variances [27]. It follows that the probability of the observation for a given front-end component k and model component m is given by equation (A.9).

A.3 Linear Models and Expected Values

Let $\boldsymbol{x}$ be a random vector and $\boldsymbol{y}$ a linear function of $\boldsymbol{x}$ such that

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} \tag{A.22}$$

where $\boldsymbol{A}$ is a square transformation matrix and $\boldsymbol{b}$ a bias vector. The expected value of $\boldsymbol{y}$ is then given by

$$\begin{aligned}\boldsymbol{\mu}_y &= \mathcal{E}\{\boldsymbol{y}\} \\ &= \mathcal{E}\{\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}\} \\ &= \boldsymbol{A}\boldsymbol{\mu}_x + \boldsymbol{b}\end{aligned} \tag{A.23}$$

where $\boldsymbol{\mu}_x$ is the mean of $\boldsymbol{x}$. The covariance of $\boldsymbol{x}$ is $\boldsymbol{\Sigma}_x$ whereas the covariance of $\boldsymbol{y}$ is given by

$$\begin{aligned}\boldsymbol{\Sigma}_y &= \mathcal{E}\Big\{(\boldsymbol{y} - \boldsymbol{\mu}_y)(\boldsymbol{y} - \boldsymbol{\mu}_y)^{\mathsf{T}}\Big\} \\ &= \mathcal{E}\Big\{(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} - (\boldsymbol{A}\boldsymbol{\mu}_x + \boldsymbol{b}))(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} - (\boldsymbol{A}\boldsymbol{\mu}_x + \boldsymbol{b}))^{\mathsf{T}}\Big\} \\ &= \mathcal{E}\Big\{(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}\boldsymbol{\mu}_x)(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}\boldsymbol{\mu}_x)^{\mathsf{T}}\Big\} \\ &= \mathcal{E}\Big\{\boldsymbol{A}(\boldsymbol{x} - \boldsymbol{\mu}_x)(\boldsymbol{x} - \boldsymbol{\mu}_x)^{\mathsf{T}}\boldsymbol{A}^{\mathsf{T}}\Big\} \\ &= \boldsymbol{A}\boldsymbol{\Sigma}_x\boldsymbol{A}^{\mathsf{T}}\end{aligned} \tag{A.24}$$

This result is useful for deriving the covariance terms of the corrupted speech distribution from a linear equation of the clean speech and noise means, e.g. equation (4.33). The covariance between $\boldsymbol{y}$ and $\boldsymbol{x}$ is

$$\begin{aligned}\boldsymbol{\Sigma}_{yx} &= \mathcal{E}\Big\{(\boldsymbol{y} - \boldsymbol{\mu}_y)(\boldsymbol{x} - \boldsymbol{\mu}_x)^{\mathsf{T}}\Big\} \\ &= \mathcal{E}\Big\{(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} - (\boldsymbol{A}\boldsymbol{\mu}_x + \boldsymbol{b}))(\boldsymbol{x} - \boldsymbol{\mu}_x)^{\mathsf{T}}\Big\} \\ &= \mathcal{E}\Big\{(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}\boldsymbol{\mu}_x)(\boldsymbol{x} - \boldsymbol{\mu}_x)^{\mathsf{T}}\Big\} \\ &= \mathcal{E}\Big\{\boldsymbol{A}(\boldsymbol{x} - \boldsymbol{\mu}_x)(\boldsymbol{x} - \boldsymbol{\mu}_x)^{\mathsf{T}}\Big\} \\ &= \boldsymbol{A}\boldsymbol{\Sigma}_x\end{aligned} \tag{A.25}$$

Hence the covariance between a random vector and its transformed version is simply a linear transform of the variance of the random vector.

APPENDIX B

Model-based VTS Compensation

The non-linear environmental mismatch function for relating clean speech $\boldsymbol{x}$, additive noise $\boldsymbol{z}$ and channel noise $\boldsymbol{h}$ to the corrupted speech $\boldsymbol{y}$ was given in chapter 3. Ignoring the time subscript, it is

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{h} + \boldsymbol{C}\mathbf{log}\big(\mathbf{1} + \mathbf{exp}\big(\boldsymbol{C}^{-1}(\boldsymbol{z} - \boldsymbol{x} - \boldsymbol{h})\big)\big) \tag{B.1}$$

Many approximations to this function have been proposed, such as selecting the maximum of either the noise or speech, i.e. noise masking [145] or PMC [39]. Another approach is to linearise it with a truncated vector Taylor series (VTS) [2, 80, 106] to individually update each model component. The first-order VTS approximation of the static corrupted speech for dimension i is

$$y_{\mathsf{vts},i} = y_i|_{\mu_0^{(m)}} + \boldsymbol{\nabla}_{\boldsymbol{x}}\, y_i|_{\mu_0^{(m)}} \bullet \big(\boldsymbol{x} - \boldsymbol{\mu}_x^{(m)}\big) + \boldsymbol{\nabla}_{\boldsymbol{z}}\, y_i|_{\mu_0^{(m)}} \bullet \big(\boldsymbol{z} - \boldsymbol{\mu}_z\big) + \boldsymbol{\nabla}_{\boldsymbol{h}}\, y_i|_{\mu_0^{(m)}} \bullet \big(\boldsymbol{h} - \boldsymbol{\mu}_h\big) \tag{B.2}$$

where $|_{\mu_0^{(m)}}$ indicates evaluation at the Taylor series expansion point of the clean speech component mean $\boldsymbol{\mu}_x^{(m)}$, and current estimates of the additive noise mean $\boldsymbol{\mu}_z$ and channel noise $\boldsymbol{\mu}_h$. The symbol $\bullet$ indicates the dot product and $\boldsymbol{\nabla}$ a gradient operator. Taking the expected value value of equation (B.2) associated with a given model component

$$\begin{aligned}\mu_{y,i}^{(m)} &\approx \mathcal{E}\left\{y_{\mathsf{vts},i}|m\right\} = y_i|_{\mu_0^{(m)}} \\ &= \mu_{x,i}^{(m)} + \mu_{h,i} + \boldsymbol{c}_i\mathbf{log}(\mathbf{1} + \mathbf{exp}(\boldsymbol{C}^{-1}(\boldsymbol{\mu}_z - \boldsymbol{\mu}_x^{(m)} - \boldsymbol{\mu}_h)))\end{aligned} \tag{B.3}$$

where the term $\boldsymbol{c}_{\bar{i}}$ is a row vector that is the ith row of the DCT matrix $\boldsymbol{C}$. Equation (B.3) can be written in vector form as

$$\boldsymbol{\mu}_y^{(m)} \approx \boldsymbol{\mu}_x^{(m)} + \boldsymbol{\mu}_h + \boldsymbol{C}\mathbf{log}(\mathbf{1} + \mathbf{exp}(\boldsymbol{C}^{-1}(\boldsymbol{\mu}_z - \boldsymbol{\mu}_x^{(m)} - \boldsymbol{\mu}_h))) \tag{B.4}$$

This gives a relationship between the corrupted speech mean, clean speech mean and the noise model assuming the clean speech, additive noise and channel noise are independent of each other and are Gaussian distributed random variables.

B.1 Compensating Dynamic Coefficients

In state of the art recognition systems, time derivatives improve performance by addressing the continuous nature of speech. The Continuous-Time approximation has been used to give an analytic approximation of the first- and second-order dynamic corrupted speech features [39]. Applying the chain rule to the first time derivative of the corrupted speech yields the following

$$\begin{aligned}\frac{\partial y_i}{\partial t} &= \frac{\partial y_i}{\partial \boldsymbol{x}} \bullet \frac{\partial \boldsymbol{x}}{\partial t} + \frac{\partial y_i}{\partial \boldsymbol{z}} \bullet \frac{\partial \boldsymbol{z}}{\partial t} + \frac{\partial y_i}{\partial \boldsymbol{h}} \bullet \frac{\partial \boldsymbol{h}}{\partial t} \\ &= \boldsymbol{\nabla}_{\boldsymbol{x}}\, y_i \bullet \frac{\partial \boldsymbol{x}}{\partial t} + \boldsymbol{\nabla}_{\boldsymbol{z}}\, y_i \bullet \frac{\partial \boldsymbol{z}}{\partial t} + \boldsymbol{\nabla}_{\boldsymbol{h}}\, y_i \bullet \frac{\partial \boldsymbol{h}}{\partial t}\end{aligned} \tag{B.5}$$

where recall $\boldsymbol{y}$ is the static corrupted speech, $\boldsymbol{x}$ the static clean speech, $\boldsymbol{h}$ the static channel noise and $\boldsymbol{z}$ the static additive noise variables where the subscript t has been omitted for simplicity. Substituting the first-order VTS approximation in equation (B.2) w.r.t. time for the actual corrupted speech, gives the following

$$\begin{aligned}\frac{\partial y_i}{\partial t} &\approx \frac{\partial y_{\mathsf{vts},i}}{\partial t} \\ &= \boldsymbol{\nabla}_{\boldsymbol{x}}\Big\{ y_i|_{\mu_0^{(m)}} + \boldsymbol{\nabla}_{\boldsymbol{x}}\, y_i|_{\mu_0^{(m)}} \bullet (\boldsymbol{x} - \boldsymbol{\mu}_x) + \boldsymbol{\nabla}_{\boldsymbol{z}}\, y_i|_{\mu_0^{(m)}} \bullet (\boldsymbol{z} - \boldsymbol{\mu}_z) + \boldsymbol{\nabla}_{\boldsymbol{h}}\, y_i|_{\mu_0^{(m)}} \bullet (\boldsymbol{h} - \boldsymbol{\mu}_h)\Big\} \bullet \frac{\partial \boldsymbol{x}}{\partial t} + \\ &\quad \boldsymbol{\nabla}_{\boldsymbol{z}}\Big\{ y_i|_{\mu_0^{(m)}} + \boldsymbol{\nabla}_{\boldsymbol{x}}\, y_i|_{\mu_0^{(m)}} \bullet (\boldsymbol{x} - \boldsymbol{\mu}_x) + \boldsymbol{\nabla}_{\boldsymbol{z}}\, y_i|_{\mu_0^{(m)}} \bullet (\boldsymbol{z} - \boldsymbol{\mu}_z) + \boldsymbol{\nabla}_{\boldsymbol{h}}\, y_i|_{\mu_0^{(m)}} \bullet (\boldsymbol{h} - \boldsymbol{\mu}_h)\Big\} \bullet \frac{\partial \boldsymbol{z}}{\partial t} + \\ &\quad \boldsymbol{\nabla}_{\boldsymbol{h}}\Big\{ y_i|_{\mu_0^{(m)}} + \boldsymbol{\nabla}_{\boldsymbol{x}}\, y_i|_{\mu_0^{(m)}} \bullet (\boldsymbol{x} - \boldsymbol{\mu}_x) + \boldsymbol{\nabla}_{\boldsymbol{z}}\, y_i|_{\mu_0^{(m)}} \bullet (\boldsymbol{z} - \boldsymbol{\mu}_z) + \boldsymbol{\nabla}_{\boldsymbol{h}}\, y_i|_{\mu_0^{(m)}} \bullet (\boldsymbol{h} - \boldsymbol{\mu}_h)\Big\} \bullet \frac{\partial \boldsymbol{h}}{\partial t}\end{aligned} \tag{B.6}$$

Since the clean speech, additive and channel noise may all considered independent of each other

$$\frac{\partial y_i}{\partial t} \approx \boldsymbol{\nabla}_{\boldsymbol{x}} \Big\{ \boldsymbol{\nabla}_{\boldsymbol{x}}\, y_i|_{\mu_0^{(m)}} \bullet \boldsymbol{x} \Big\} \bullet \frac{\partial \boldsymbol{x}}{\partial t} + \boldsymbol{\nabla}_{\boldsymbol{z}} \Big\{ \boldsymbol{\nabla}_{\boldsymbol{z}}\, y_i|_{\mu_0^{(m)}} \bullet \boldsymbol{z} \Big\} \bullet \frac{\partial \boldsymbol{z}}{\partial t} + \boldsymbol{\nabla}_{\boldsymbol{h}} \Big\{ \boldsymbol{\nabla}_{\boldsymbol{h}}\, y_i|_{\mu_0^{(m)}} \bullet \boldsymbol{h} \Big\} \bullet \frac{\partial \boldsymbol{h}}{\partial t} \tag{B.7}$$

Recall that the vector gradient quantity in the dot products, such as $\boldsymbol{\nabla}_{\boldsymbol{x}}\, y_i|_{\mu_0^{(m)}}$, is the gradient of the corrupted speech, w.r.t. $\boldsymbol{x}$, but with variables evaluated at $\boldsymbol{\mu}_0^{(m)}$ and hence is no longer a function of any of the random variables. Thus

$$\begin{aligned}\frac{\partial y_i}{\partial t} &\approx \Big\{ \boldsymbol{\nabla}_{\boldsymbol{x}}\, y_i|_{\mu_0^{(m)}} \bullet \boldsymbol{\nabla}_{\boldsymbol{x}} \boldsymbol{x} \Big\} \bullet \frac{\partial \boldsymbol{x}}{\partial t} + \Big\{ \boldsymbol{\nabla}_{\boldsymbol{z}}\, y_i|_{\mu_0^{(m)}} \bullet \boldsymbol{\nabla}_{\boldsymbol{z}} \boldsymbol{z} \Big\} \bullet \frac{\partial \boldsymbol{z}}{\partial t} + \Big\{ \boldsymbol{\nabla}_{\boldsymbol{h}}\, y_i|_{\mu_0^{(m)}} \bullet \boldsymbol{\nabla}_{\boldsymbol{h}} \boldsymbol{h} \Big\} \bullet \frac{\partial \boldsymbol{h}}{\partial t} \\ &= \boldsymbol{\nabla}_{\boldsymbol{x}}\, y_i|_{\mu_0^{(m)}} \bullet \frac{\partial \boldsymbol{x}}{\partial t} + \boldsymbol{\nabla}_{\boldsymbol{z}}\, y_i|_{\mu_0^{(m)}} \bullet \frac{\partial \boldsymbol{z}}{\partial t} + \boldsymbol{\nabla}_{\boldsymbol{h}}\, y_i|_{\mu_0^{(m)}} \bullet \frac{\partial \boldsymbol{h}}{\partial t}\end{aligned} \tag{B.8}$$

This may be re-expressed as

$$\frac{\partial \boldsymbol{y}}{\partial t} \approx \boldsymbol{J}_x^{(m)} \frac{\partial \boldsymbol{x}}{\partial t} + \boldsymbol{J}_z^{(m)} \frac{\partial \boldsymbol{z}}{\partial t} + \boldsymbol{J}_h^{(m)} \frac{\partial \boldsymbol{h}}{\partial t} \tag{B.9}$$

where the each row of the Jacobian matrices give the gradient of a dimension of the corrupted speech w.r.t. the clean speech, additive noise or channel noise vectors, all evaluated at the expansion point $\boldsymbol{\mu}_0^{(m)}$ and were expressed earlier in equations (4.28) and (4.30). The expected value of equation (B.9), across the clean speech and noise variables, gives the corrupted speech delta parameters for a given model component m

$$\begin{aligned} \boldsymbol{\mu}_{\Delta y}^{(m)} &\approx \mathcal{E}\left\{ \frac{\partial \boldsymbol{y}}{\partial t} \middle| m \right\} \\ &\approx \boldsymbol{J}_x^{(m)} \boldsymbol{\mu}_{\Delta x}^{(m)} + \boldsymbol{J}_z^{(m)} \boldsymbol{\mu}_{\Delta z} + \boldsymbol{J}_h^{(m)} \boldsymbol{\mu}_{\Delta h} \end{aligned} \tag{B.10}$$

If it is assumed that the additive noise is stationary, hence $\boldsymbol{\mu}_{\Delta z} = 0$, and the convolutional noise invariant, implying $\boldsymbol{\mu}_{\Delta h} = 0$, then

$$\boldsymbol{\mu}_{\Delta y}^{(m)} \approx \boldsymbol{J}_x^{(m)} \boldsymbol{\mu}_{\Delta x}^{(m)} \tag{B.11}$$

Finding the variance of $\frac{\partial \boldsymbol{y}}{\partial t}$ in equation (B.9) gives an approximation to the delta corrupted speech variance

$$\begin{aligned} \boldsymbol{\Sigma}_{\Delta y}^{(m)} &\approx \mathcal{E}\left\{ \frac{\partial \boldsymbol{y}}{\partial t} \frac{\partial \boldsymbol{y}}{\partial t}^{\mathsf{T}} \middle| m \right\} - \boldsymbol{\mu}_{\Delta y}^{(m)} \boldsymbol{\mu}_{\Delta y}^{(m)\mathsf{T}} \\ &\approx \boldsymbol{J}_x^{(m)} \boldsymbol{\Sigma}_{\Delta x}^{(m)} \boldsymbol{J}_x^{(m)\mathsf{T}} + \boldsymbol{J}_z^{(m)} \boldsymbol{\Sigma}_{\Delta z} \boldsymbol{J}_z^{(m)\mathsf{T}} + \boldsymbol{J}_h^{(m)} \boldsymbol{\Sigma}_{\Delta h} \boldsymbol{J}_h^{(m)\mathsf{T}} \end{aligned} \tag{B.12}$$

The assumption of channel invariance translates to zero channel variance, hence

$$\boldsymbol{\Sigma}_{\Delta y}^{(m)} \approx \boldsymbol{J}_x^{(m)} \boldsymbol{\Sigma}_{\Delta x}^{(m)} \boldsymbol{J}_x^{(m)\mathsf{T}} + \boldsymbol{J}_z^{(m)} \boldsymbol{\Sigma}_{\Delta z} \boldsymbol{J}_z^{(m)\mathsf{T}} \tag{B.13}$$

where $\boldsymbol{J}_x^{(m)}$ and $\boldsymbol{J}_z^{(m)}$ indicate the respective Jacobian matrices are instead evaluated at the component clean speech mean $\boldsymbol{\mu}_x^{(m)}$ and the noise means.

B.2 Delta-delta Coefficients

Delta-delta coefficients are also typically used in standard recognisers and may be approximated by second-order time derivatives. Differentiating equation (B.9) w.r.t. time, gives the equivalent form found in [2], however embeds the VTS approximation in the partial derivative.

If we start from equation (B.5), while again assuming invariant convolutional noise, then

$$\begin{aligned}\frac{\partial^2 y_i}{\partial t^2} &= \frac{\partial}{\partial t}\left\{\frac{\partial y_i}{\partial t}\right\} \\ &= \frac{\partial}{\partial t}\left\{\boldsymbol{\nabla}_{\boldsymbol{x}}\, y_i \bullet \frac{\partial \boldsymbol{x}}{\partial t} + \boldsymbol{\nabla}_{\boldsymbol{z}}\, y_i \bullet \frac{\partial \boldsymbol{z}}{\partial t}\right\} \\ &= \frac{\partial \boldsymbol{\nabla}_{\boldsymbol{x}}\, y_i}{\partial t} \bullet \frac{\partial \boldsymbol{x}}{\partial t} + \boldsymbol{\nabla}_{\boldsymbol{x}}\, y_i \bullet \frac{\partial^2 \boldsymbol{x}}{\partial t^2} + \frac{\partial \boldsymbol{\nabla}_{\boldsymbol{z}}\, y_i}{\partial t} \bullet \frac{\partial \boldsymbol{z}}{\partial t} + \boldsymbol{\nabla}_{\boldsymbol{z}}\, y_i \bullet \frac{\partial^2 \boldsymbol{z}}{\partial t^2} \\ &= \left\{\frac{\partial^2 y_i}{\partial \boldsymbol{x} \partial \boldsymbol{x}}\frac{\partial \boldsymbol{x}}{\partial t} + \frac{\partial^2 y_i}{\partial \boldsymbol{z} \partial \boldsymbol{x}}\frac{\partial \boldsymbol{z}}{\partial t}\right\} \bullet \frac{\partial \boldsymbol{x}}{\partial t} + \boldsymbol{\nabla}_{\boldsymbol{x}}\, y_i \bullet \frac{\partial^2 \boldsymbol{x}}{\partial t^2} + \\ &\quad \left\{\frac{\partial^2 y_i}{\partial \boldsymbol{x} \partial \boldsymbol{z}}\frac{\partial \boldsymbol{x}}{\partial t} + \frac{\partial^2 y_i}{\partial \boldsymbol{z} \partial \boldsymbol{z}}\frac{\partial \boldsymbol{z}}{\partial t}\right\} \bullet \frac{\partial \boldsymbol{z}}{\partial t} + \boldsymbol{\nabla}_{\boldsymbol{z}}\, y_i \bullet \frac{\partial^2 \boldsymbol{z}}{\partial t^2}\end{aligned} \tag{B.14}$$

Since there are no mixed terms in a VTS approximation of the clean speech, the mixed partial derivatives vanish, simplifying equation (B.14) to

$$\frac{\partial^2 y_i}{\partial t^2} \approx \frac{\partial^2 y_i}{\partial \boldsymbol{x} \partial \boldsymbol{x}}\frac{\partial \boldsymbol{x}}{\partial t} \bullet \frac{\partial \boldsymbol{x}}{\partial t} + \boldsymbol{\nabla}_{\boldsymbol{x}}\, y_i \bullet \frac{\partial^2 \boldsymbol{x}}{\partial t^2} + \frac{\partial^2 y_i}{\partial \boldsymbol{z} \partial \boldsymbol{z}}\frac{\partial \boldsymbol{z}}{\partial t} \bullet \frac{\partial \boldsymbol{z}}{\partial t} + \boldsymbol{\nabla}_{\boldsymbol{z}}\, y_i \bullet \frac{\partial^2 \boldsymbol{z}}{\partial t^2} \tag{B.15}$$

Substituting a first-order VTS approximation of the corrupted speech y_i will also result in the second-order partial derivatives being null

$$\frac{\partial^2 y_i}{\partial t^2} \approx \frac{\partial^2 y_{\mathsf{vts},i}}{\partial t^2} = \boldsymbol{\nabla}_{\boldsymbol{x}}\, y_i|_{\mu_0^{(m)}} \bullet \frac{\partial^2 \boldsymbol{x}}{\partial t^2} + \boldsymbol{\nabla}_{\boldsymbol{z}}\, y_i|_{\mu_0^{(m)}} \bullet \frac{\partial^2 \boldsymbol{z}}{\partial t^2} \tag{B.16}$$

which may be re-expressed as

$$\frac{\partial^2 \boldsymbol{y}}{\partial t^2} \approx \boldsymbol{J}_x^{(m)} \frac{\partial^2 \boldsymbol{x}}{\partial t^2} + \boldsymbol{J}_z^{(m)} \frac{\partial^2 \boldsymbol{z}}{\partial t^2} \tag{B.17}$$

Since equation (B.17) is similar in form to equation (B.9), it is obvious the mean and variance delta-delta coefficients of the corrupted speech distribution are also similar

$$\boldsymbol{\mu}_{\Delta^2 y}^{(m)} \approx \boldsymbol{J}_x^{(m)} \boldsymbol{\mu}_{\Delta^2 x}^{(m)} \tag{B.18}$$

$$\boldsymbol{\Sigma}_{\Delta^2 y}^{(m)} \approx \boldsymbol{J}_x^{(m)} \boldsymbol{\Sigma}_{\Delta^2 x}^{(m)} \boldsymbol{J}_x^{(m)\mathsf{T}} + \boldsymbol{J}_z^{(m)} \boldsymbol{\Sigma}_{\Delta^2 z} \boldsymbol{J}_z^{(m)\mathsf{T}} \tag{B.19}$$

This is the same result as found in [2], however investigating higher order VTS approximations for the delta-delta coefficients may be fruitful.

Alternatively, a second-order VTS approximation of the corrupted speech may be made to give a better estimation of the acceleration coefficients. This is

$$\begin{aligned}y_{2\mathsf{vts},i} = y_i|_{\mu_0^{(m)}} &+ \boldsymbol{\nabla}_{\boldsymbol{x}}\, y_i|_{\mu_0^{(m)}} \bullet (\boldsymbol{x} - \boldsymbol{\mu}_x) + \boldsymbol{\nabla}_{\boldsymbol{z}}\, y_i|_{\mu_0^{(m)}} \bullet (\boldsymbol{z} - \boldsymbol{\mu}_z) + \boldsymbol{\nabla}_{\boldsymbol{h}}\, y_i|_{\mu_0^{(m)}} \bullet (\boldsymbol{h} - \boldsymbol{\mu}_h) + \\ &\frac{1}{2}\frac{\partial^2 y_i}{\partial \boldsymbol{x} \partial \boldsymbol{x}} \bullet (\boldsymbol{x} - \boldsymbol{\mu}_x) \bullet (\boldsymbol{x} - \boldsymbol{\mu}_x) + \frac{1}{2}\frac{\partial^2 y_i}{\partial \boldsymbol{z} \partial \boldsymbol{z}} \bullet (\boldsymbol{z} - \boldsymbol{\mu}_z) \bullet (\boldsymbol{z} - \boldsymbol{\mu}_z) + \frac{1}{2}\frac{\partial^2 y_i}{\partial \boldsymbol{h} \partial \boldsymbol{h}} \bullet (\boldsymbol{h} - \boldsymbol{\mu}_h) \bullet (\boldsymbol{h} - \boldsymbol{\mu}_h)\end{aligned} \tag{B.20}$$

The second-order partial derivative matrices have elements

$$\begin{aligned}\frac{\partial^2 y_i}{\partial x_j \partial x_k} &= \frac{\partial^2 y_i}{\partial z_j \partial z_k} = \frac{\partial^2 y_i}{\partial h_j \partial h_k} \\ &= \sum_{d=1}^{D_s} c_{id} \left(\frac{\exp(\boldsymbol{c}_d^{-1}(\boldsymbol{\mu}_z - \boldsymbol{\mu}_x^{(m)} - \boldsymbol{\mu}_h))}{1 + \exp(\boldsymbol{c}_d^{-1}(\boldsymbol{\mu}_z - \boldsymbol{\mu}_x^{(m)} - \boldsymbol{\mu}_h))}\right)\left(\frac{1}{1 + \exp(\boldsymbol{c}_d^{-1}(\boldsymbol{\mu}_z - \boldsymbol{\mu}_x^{(m)} - \boldsymbol{\mu}_h))}\right) c_{dj}^{-1} c_{dk}^{-1}\end{aligned} \tag{B.21}$$

for row j and column k and $\boldsymbol{c}_d^{-1}$ gives the dth row of the inverse DCT.

B.3 Dynamic Cross-Covariance Coefficients

To compute the joint distribution for a regression class r, the cross-covariance between the clean and corrupted speech is needed. The static cross-covariance parameters were derived in section 5.4, however dynamic coefficients are also necessary if a system with dynamic coefficients is to be compensated. The delta cross-covariance is defined as

$$\boldsymbol{\Sigma}_{\Delta y \Delta x}^{(r)} = \mathcal{E}\left\{(\Delta \boldsymbol{y} - \boldsymbol{\mu}_{\Delta y}^{(r)})(\Delta \boldsymbol{x} - \boldsymbol{\mu}_{\Delta x}^{(r)})^{\mathsf{T}} \middle| r\right\} \tag{B.22}$$

The Continuous-Time approximation has the delta variables approximated by time derivatives, where the convolutional noise is not considered

$$\begin{aligned}\boldsymbol{\Sigma}_{\Delta y \Delta x}^{(r)} &\approx \mathcal{E}\left\{\left(\frac{\partial \boldsymbol{y}}{\partial t} - \boldsymbol{\mu}_{\Delta y}^{(r)}\right)\left(\frac{\partial \boldsymbol{x}}{\partial t} - \boldsymbol{\mu}_{\Delta x}^{(r)}\right)^{\mathsf{T}} \middle| r\right\} \\ &\approx \mathcal{E}\left\{\left(\boldsymbol{J}_x^{(r)}\frac{\partial \boldsymbol{x}}{\partial t} + \boldsymbol{J}_z^{(r)}\frac{\partial \boldsymbol{z}}{\partial t} - \boldsymbol{J}_x^{(r)}\boldsymbol{\mu}_{\Delta x}^{(r)} - \boldsymbol{J}_z^{(r)}\boldsymbol{\mu}_{\Delta z}\right)\left(\frac{\partial \boldsymbol{x}}{\partial t} - \boldsymbol{\mu}_{\Delta x}^{(r)}\right)^{\mathsf{T}} \middle| r\right\}\end{aligned} \tag{B.23}$$

An approximation to $\frac{\partial \boldsymbol{y}}{\partial t}$ was given in equation (B.9). Assuming independence between the speech and noise variables, simplifies this to

$$\begin{aligned}\boldsymbol{\Sigma}_{\Delta y \Delta x}^{(r)} &\approx \mathcal{E}\left\{\left(\boldsymbol{J}_x^{(r)}\frac{\partial \boldsymbol{x}}{\partial t} - \boldsymbol{J}_x^{(r)}\boldsymbol{\mu}_{\Delta x}^{(r)}\right)\left(\frac{\partial \boldsymbol{x}}{\partial t} - \boldsymbol{\mu}_{\Delta x}^{(r)}\right)^{\mathsf{T}} \middle| r\right\} \\ &= \boldsymbol{J}_x^{(r)}\mathcal{E}\left\{\left(\frac{\partial \boldsymbol{x}}{\partial t} - \boldsymbol{\mu}_{\Delta x}^{(r)}\right)\left(\frac{\partial \boldsymbol{x}}{\partial t} - \boldsymbol{\mu}_{\Delta x}^{(r)}\right)^{\mathsf{T}} \middle| r\right\} \\ &\approx \boldsymbol{J}_x^{(r)}\boldsymbol{\Sigma}_{\Delta x}^{(r)}\end{aligned} \tag{B.24}$$

The delta-delta cross-covariance between the clean and corrupted speech may be derived in a similar manner. The delta-delta cross-covariance is defined as

$$\boldsymbol{\Sigma}_{\Delta^2 y \Delta^2 x}^{(r)} = \mathcal{E}\left\{(\Delta^2 \boldsymbol{y} - \boldsymbol{\mu}_{\Delta^2 y}^{(r)})(\Delta^2 \boldsymbol{x} - \boldsymbol{\mu}_{\Delta^2 x}^{(r)})^{\mathsf{T}} \middle| r\right\} \tag{B.25}$$

The Continuous-Time approximation has the delta-delta variables approximated by second-order time derivatives where the convolutional noise again is not considered

$$\begin{aligned}\boldsymbol{\Sigma}_{\Delta^2 y \Delta^2 x}^{(r)} &\approx \mathcal{E}\left\{\left(\frac{\partial^2 \boldsymbol{y}}{\partial t^2} - \boldsymbol{\mu}_{\Delta^2 y}^{(r)}\right)\left(\frac{\partial^2 \boldsymbol{x}}{\partial t^2} - \boldsymbol{\mu}_{\Delta^2 x}^{(r)}\right)^{\mathsf{T}} \middle| r\right\} \\ &\approx \mathcal{E}\left\{\left(\boldsymbol{J}_x^{(r)}\frac{\partial^2 \boldsymbol{x}}{\partial t^2} + \boldsymbol{J}_z^{(r)}\frac{\partial^2 \boldsymbol{z}}{\partial t^2} - \boldsymbol{J}_x^{(r)}\boldsymbol{\mu}_{\Delta^2 x}^{(r)} - \boldsymbol{J}_z^{(r)}\boldsymbol{\mu}_{\Delta^2 z}\right)\left(\frac{\partial^2 \boldsymbol{x}}{\partial t^2} - \boldsymbol{\mu}_{\Delta^2 x}^{(r)}\right)^{\mathsf{T}} \middle| r\right\}\end{aligned} \tag{B.26}$$

An approximation to $\frac{\partial^2 \boldsymbol{y}}{\partial t^2}$ was given in equation (B.17). Assuming independence between the speech and noise variables, simplifies this to

$$\begin{aligned}
\boldsymbol{\Sigma}^{(r)}_{\Delta^2 x \Delta^2 y} &\approx \mathcal{E}\left\{ \left(\boldsymbol{J}^{(r)}_x \frac{\partial^2 \boldsymbol{x}}{\partial t^2} - \boldsymbol{J}^{(r)}_x \boldsymbol{\mu}^{(r)}_{\Delta^2 x} \right) \left(\frac{\partial^2 \boldsymbol{x}}{\partial t^2} - \boldsymbol{\mu}^{(r)}_{\Delta^2 x} \right)^{\mathsf{T}} \middle| r \right\} \\
&= \boldsymbol{J}^{(r)}_x \mathcal{E}\left\{ \left(\frac{\partial^2 \boldsymbol{x}}{\partial t^2} - \boldsymbol{\mu}^{(r)}_{\Delta^2 x} \right) \left(\frac{\partial^2 \boldsymbol{x}}{\partial t^2} - \boldsymbol{\mu}^{(r)}_{\Delta^2 x} \right)^{\mathsf{T}} \middle| r \right\} \\
&\approx \boldsymbol{J}^{(r)}_x \boldsymbol{\Sigma}^{(r)}_{\Delta^2 x}
\end{aligned} \tag{B.27}$$

Appendix C

Derivative of Auxiliary w.r.t. Additive Noise Variance

ML estimates of the additive and channel noise means, as well as the additive noise variance, maximise the likelihood of noise-corrupted speech data with clean speech models compensated using these estimates. Determining these estimates directly from a log-likelihood function is difficult. Hence and EM approach is taken. This requires optimising the auxiliary function

$$\begin{aligned}\mathcal{Q}_{\mathsf{vts}}(\mathcal{M}_n;\hat{\mathcal{M}}_n) &= \mathbb{E}_{\hat{\mathcal{M}}}\Big[\log \mathrm{p}(\boldsymbol{O},\boldsymbol{M}|\hat{\mathcal{M}}_n,\mathcal{M})\Big] \\ &= \sum_{t=1}^{T}\sum_{m=1}^{M}\gamma_{o,t}^{(m)}\Big[-\frac{1}{2}\log|\boldsymbol{\Sigma}_{y}^{(m)}| - \frac{1}{2}(\boldsymbol{y}_t-\boldsymbol{\mu}_{y}^{(m)})^{\mathsf{T}}\boldsymbol{\Sigma}_{y}^{(m)\text{-}1}(\boldsymbol{y}_t-\boldsymbol{\mu}_{y}^{(m)}) \\ &\qquad -\frac{1}{2}\log|\boldsymbol{\Sigma}_{\Delta y}^{(m)}| - \frac{1}{2}(\Delta\boldsymbol{y}_t-\boldsymbol{\mu}_{\Delta y}^{(m)})^{\mathsf{T}}\boldsymbol{\Sigma}_{\Delta y}^{(m)\text{-}1}(\Delta\boldsymbol{y}_t-\boldsymbol{\mu}_{\Delta y}^{(m)}) \\ &\qquad -\frac{1}{2}\log|\boldsymbol{\Sigma}_{\Delta^2 y}^{(m)}| - \frac{1}{2}(\Delta^2\boldsymbol{y}_t-\boldsymbol{\mu}_{\Delta^2 y}^{(m)})^{\mathsf{T}}\boldsymbol{\Sigma}_{\Delta^2 y}^{(m)\text{-}1}(\Delta^2\boldsymbol{y}_t-\boldsymbol{\mu}_{\Delta^2 y}^{(m)})\Big]\end{aligned} \tag{C.1}$$

given previously in equation (6.3) w.r.t. these noise means.

The partial derivative of equation (C.1) w.r.t. the additive noise variance is needed in 6.2.2

to estimate the additive noise variance. For the static dimensions this is

$$\frac{\partial \mathcal{Q}_{\mathsf{vts}}}{\partial \boldsymbol{\Sigma}_z} = -\frac{1}{2}\sum_{t=1}^{T}\sum_{m=1}^{M}\gamma_{o,t}^{(m)}\frac{\partial}{\partial \boldsymbol{\Sigma}_z}\left[\log|\boldsymbol{\Sigma}_y^{(m)}| + \left(\boldsymbol{y}_t - \boldsymbol{\mu}_y^{(m)}\right)^{\mathsf{T}}\boldsymbol{\Sigma}_y^{(m)\text{-}1}\left(\boldsymbol{y}_t - \boldsymbol{\mu}_y^{(m)}\right)\right] \tag{C.2}$$

from equation (6.8). Here the gradient w.r.t. the static additive noise variance is only a function of the static parameters. The two terms that are being differentiated can be examined separately and on a per dimension basis. First determine the derivative of the normalising determinant

$$\begin{aligned}\frac{\partial}{\partial \sigma_{z,i}^2}\log|\boldsymbol{\Sigma}_y^{(m)}| &= \frac{1}{|\boldsymbol{\Sigma}_y^{(m)}|}\frac{\partial|\boldsymbol{\Sigma}_y^{(m)}|}{\partial \sigma_{z,i}^2} \\ &= \frac{1}{|\boldsymbol{\Sigma}_y^{(m)}|}|\boldsymbol{\Sigma}_y^{(m)}|\mathtt{trace}\left\{\boldsymbol{\Sigma}_y^{(m)\text{-}1}\frac{\partial \boldsymbol{\Sigma}_y^{(m)}}{\partial \sigma_{z,i}^2}\right\}\end{aligned} \tag{C.3}$$

The partial derivative of the corrupted speech variance w.r.t. the additive noise variance is needed. The variance of the first-order VTS approximation of the corrupted speech in equation (4.34) provides the relationship. Hence,

$$\begin{aligned}\frac{\partial \boldsymbol{\Sigma}_y^{(m)}}{\partial \sigma_{z,i}^2} &\approx \frac{\partial}{\partial \sigma_{z,i}^2}\left\{\boldsymbol{J}_x^{(m)}\boldsymbol{\Sigma}_x^{(m)}\boldsymbol{J}_x^{(m)\mathsf{T}} + \boldsymbol{J}_z^{(m)}\boldsymbol{\Sigma}_z\boldsymbol{J}_z^{(m)\mathsf{T}}\right\} \\ &= 0 + \boldsymbol{J}_z^{(m)}\frac{\partial \boldsymbol{\Sigma}_z}{\partial \sigma_{z,i}^2}\boldsymbol{J}_z^{(m)\mathsf{T}} \\ &= \boldsymbol{J}_z^{(m)}\boldsymbol{\Delta}_{ii}\boldsymbol{J}_z^{(m)\mathsf{T}} \qquad \text{(C.4)} \\ &= \left[\boldsymbol{J}_z^{(m)}\right]_i\left[\boldsymbol{J}_z^{(m)}\right]_i^{\mathsf{T}} \qquad \text{(C.5)}\end{aligned}$$

Here the D_s-square matrix $\boldsymbol{\Delta}_{ij}$ is an all zero matrix save for a single entry of 1 at row i, column j. The notation $\left[\boldsymbol{J}_z^{(m)}\right]_i$ gives the ith column of the Jacobian matrix $\boldsymbol{J}_z^{(m)}$. Substituting this result into equation (C.3) gives

$$\frac{\partial}{\partial \sigma_{z,i}^2}\log|\boldsymbol{\Sigma}_y^{(m)}| \approx \mathtt{trace}\left\{\boldsymbol{\Sigma}_y^{(m)\text{-}1}\left[\boldsymbol{J}_z^{(m)}\right]_i\left[\boldsymbol{J}_z^{(m)}\right]_i^{\mathsf{T}}\right\} \tag{C.6}$$

If it is assumed the inverse corrupted speech variance is diagonal, and since the `trace` function only takes into account the diagonal terms, then this can be written as

$$\begin{aligned}\frac{\partial}{\partial \sigma_{z,i}^2}\log|\boldsymbol{\Sigma}_y^{(m)}| &\approx \mathtt{trace}\left\{\boldsymbol{\Sigma}_y^{(m)\text{-}1}\,\mathrm{diag}\left\{\left[\boldsymbol{J}_z^{(m)}\right]_i \circ \left[\boldsymbol{J}_z^{(m)}\right]_i\right\}\right\} \\ &= \sum_{d=1}^{D_s}\frac{1}{\sigma_{y,d}^{(m)2}}\left[\boldsymbol{J}_z^{(m)}\right]_{di}^2\end{aligned} \tag{C.7}$$

where the diag function converts the element-wise vector product to a matrix with the vector elements on the diagonal. Next, the derivative of the main probability term is

$$\begin{aligned}\frac{\partial}{\partial \sigma_{z,i}^2}\left(\boldsymbol{y}_t - \boldsymbol{\mu}_y^{(m)}\right)^{\mathsf{T}}\boldsymbol{\Sigma}_y^{(m)\text{-}1}\left(\boldsymbol{y}_t - \boldsymbol{\mu}_y^{(m)}\right) &= \left(\boldsymbol{y}_t - \boldsymbol{\mu}_y^{(m)}\right)^{\mathsf{T}}\frac{\partial \boldsymbol{\Sigma}_y^{(m)\text{-}1}}{\partial \sigma_{z,i}^2}\left(\boldsymbol{y}_t - \boldsymbol{\mu}_y^{(m)}\right) \\ &= \left(\boldsymbol{y}_t - \boldsymbol{\mu}_y^{(m)}\right)^{\mathsf{T}}\left(-\boldsymbol{\Sigma}_y^{(m)\text{-}1}\frac{\partial \boldsymbol{\Sigma}_y^{(m)}}{\partial \sigma_{z,i}^2}\boldsymbol{\Sigma}_y^{(m)\text{-}1}\right)\left(\boldsymbol{y}_t - \boldsymbol{\mu}_y^{(m)}\right)\end{aligned} \tag{C.8}$$

after applying the identity $\frac{\partial \boldsymbol{A}^{-1}}{\partial x} = -\boldsymbol{A}^{-1}\frac{\partial \boldsymbol{A}}{\partial x}\boldsymbol{A}^{-1}$. Substituting in equation (C.4) gives

$$\begin{aligned}\frac{\partial}{\partial \sigma_{z,i}^2}\left(\boldsymbol{y}_t - \boldsymbol{\mu}_y^{(m)}\right)^{\mathsf{T}} \boldsymbol{\Sigma}_y^{(m)\text{-}1}\left(\boldsymbol{y}_t - \boldsymbol{\mu}_y^{(m)}\right) &\approx -\left(\boldsymbol{y}_t - \boldsymbol{\mu}_y^{(m)}\right)^{\mathsf{T}}\left(\boldsymbol{\Sigma}_y^{(m)\text{-}1}\boldsymbol{J}_z^{(m)}\boldsymbol{\Delta}_{ii}\boldsymbol{J}_z^{(m)\mathsf{T}}\boldsymbol{\Sigma}_y^{(m)\text{-}1}\right)\left(\boldsymbol{y}_t - \boldsymbol{\mu}_y^{(m)}\right) \\ &= -\left[\left(\boldsymbol{y}_t - \boldsymbol{\mu}_y^{(m)}\right)^{\mathsf{T}}\boldsymbol{\Sigma}_y^{(m)\text{-}1}\boldsymbol{J}_z^{(m)}\right]_i^2 \end{aligned} \tag{C.9}$$

since $\boldsymbol{b}^{\mathsf{T}}\boldsymbol{\Delta}_{ii}\boldsymbol{b} = b_i^2$. For a diagonal $\boldsymbol{\Sigma}_y^{(m)}$, the right-hand side of equation (C.9) may be expressed as

$$\begin{aligned}&-\left[\left(\boldsymbol{y}_t - \boldsymbol{\mu}_y^{(m)}\right)^{\mathsf{T}}\boldsymbol{\Sigma}_y^{(m)\text{-}1}\boldsymbol{J}_z^{(m)}\right]_i^2 \\ &= -\left[\begin{bmatrix} a_1 & a_2 & \cdots & a_{D_s}\end{bmatrix}\begin{bmatrix}\frac{1}{b_1} & 0 & \cdots & 0 \\ 0 & \frac{1}{b_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{b_{D_s}}\end{bmatrix}\begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1D_s} \\ c_{21} & c_{22} & \cdots & c_{2D_s} \\ \vdots & \vdots & \ddots & \vdots \\ c_{D_s1} & c_{D_s2} & \cdots & c_{D_sD_s}\end{bmatrix}\right]_i^2 \\ &= -\left[\begin{bmatrix}\frac{a_1}{b_1} & \frac{a_2}{b_2} & \cdots & \frac{a_{D_s}}{b_{D_s}}\end{bmatrix}\begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1D_s} \\ c_{21} & c_{22} & \cdots & c_{2D_s} \\ \vdots & \vdots & \ddots & \vdots \\ c_{D_s1} & c_{D_s2} & \cdots & c_{D_sD_s}\end{bmatrix}\right]_i^2 \\ &= -\left[\begin{bmatrix} \frac{a_1}{b_1}c_{11} + \frac{a_2}{b_2}c_{21} + \cdots + \frac{a_{D_s}}{b_{D_s}}c_{D_s1} \\ \frac{a_1}{b_1}c_{12} + \frac{a_2}{b_2}c_{22} + \cdots + \frac{a_{D_s}}{b_{D_s}}c_{D_s2} \\ \vdots \\ \frac{a_1}{b_1}c_{1D_s} + \frac{a_2}{b_2}c_{2D_s} + \cdots + \frac{a_{D_s}}{b_{D_s}}c_{D_sD_s}\end{bmatrix}_i^{\mathsf{T}}\right]^2 \end{aligned} \tag{C.10}$$

where $a_d = y_{t,d} - \mu_{y,d}^{(m)}$, $b_d = \sigma_{y,d}^{(m)2}$, and $c_{ij} = \left[\boldsymbol{J}_z^{(m)}\right]_{ij}$ The index variable i selects the ith element from the vector

$$\begin{aligned}-\left[\left(\boldsymbol{y}_t - \boldsymbol{\mu}_y^{(m)}\right)^{\mathsf{T}}\boldsymbol{\Sigma}_y^{(m)\text{-}1}\boldsymbol{J}_z^{(m)}\right]_i^2 &= -\left[\frac{a_1}{b_1}c_{1i} + \frac{a_2}{b_2}c_{2i} + \cdots + \frac{a_{D_s}}{b_{D_s}}c_{D_si}\right]^2 \\ &= -\left[\sum_{d=1}^{D_s}\frac{y_{t,d} - \mu_{y,d}^{(m)}}{\sigma_{y,d}^{(m)2}}\left[\boldsymbol{J}_z^{(m)}\right]_{di}\right]^2 \end{aligned} \tag{C.11}$$

In another approximation, the cross-terms, $(y_{t,d} - \mu_{y,d}^{(m)})(y_{t,i} - \mu_{y,i}^{(m)})$ for $d \neq i$ may be ignored, diagonalising the process. This simplifies equation (C.11) to

$$\frac{\partial}{\partial \sigma_{z,i}^2}\left(\boldsymbol{y}_t - \boldsymbol{\mu}_y^{(m)}\right)^{\mathsf{T}}\boldsymbol{\Sigma}_y^{(m)\text{-}1}\left(\boldsymbol{y}_t - \boldsymbol{\mu}_y^{(m)}\right) \approx -\sum_{d=1}^{D_s}\left(\frac{y_{t,d} - \mu_{y,d}^{(m)}}{\sigma_{y,d}^{(m)2}}\right)^2\left[\boldsymbol{J}_z^{(m)}\right]_{di}^2 \tag{C.12}$$

Hence, the gradient of the auxiliary function w.r.t. the static noise variances from equa-

tion (C.2) simplifies to

$$\begin{aligned}
\frac{\partial \mathcal{Q}_{\mathsf{vts}}}{\partial \sigma_{z,i}^2} &\approx -\frac{1}{2}\sum_{m=1}^{M}\sum_{t=1}^{T}\gamma_{o,t}^{(m)}\left[\sum_{d=1}^{D_s}\frac{\left[\boldsymbol{J}_z^{(m)}\right]_{di}^2}{\sigma_{y,d}^{(m)2}} - \sum_{d=1}^{D_s}\left(\frac{y_{t,d}-\mu_{y,d}^{(m)}}{\sigma_{y,d}^{(m)2}}\right)^2\left[\boldsymbol{J}_z^{(m)}\right]_{di}^2\right] \\
&= -\frac{1}{2}\sum_{m=1}^{M}\sum_{d=1}^{D_s}\frac{\left[\boldsymbol{J}_z^{(m)}\right]_{di}^2}{\sigma_{y,d}^{(m)2}}\sum_{t=1}^{T}\gamma_{o,t}^{(m)}\left\{1-\left(\frac{y_{t,d}-\mu_{y,d}^{(m)}}{\sigma_{y,d}^{(m)}}\right)^2\right\} \\
&= -\frac{1}{2}\sum_{m=1}^{M}\sum_{d=1}^{D_s}\frac{\left[\boldsymbol{J}_z^{(m)}\right]_{di}^2}{\sigma_{y,d}^{(m)2}}\sum_{t=1}^{T}\gamma_{o,t}^{(m)}\left\{1-\frac{y_{t,d}^2-2y_{t,d}\mu_{y,d}^{(m)}+\mu_{y,d}^{(m)2}}{\sigma_{y,d}^{(m)2}}\right\} \\
&= -\frac{1}{2}\sum_{m=1}^{M}\sum_{d=1}^{D_s}\frac{\left[\boldsymbol{J}_z^{(m)}\right]_{di}^2}{\sigma_{y,d}^{(m)2}}\left\{\gamma^{(m)}-\frac{p_d^{(m)}-2q_d^{(m)}\mu_{y,d}^{(m)}+\gamma^{(m)}\mu_{y,d}^{(m)2}}{\sigma_{y,d}^{(m)2}}\right\} \\
&= -\frac{1}{2}\sum_{m=1}^{M}\sum_{d=1}^{D_s}\frac{\left[\boldsymbol{J}_z^{(m)}\right]_{di}^2}{\sigma_{y,d}^{(m)2}}\left\{\left(1-\frac{\mu_{y,d}^{(m)2}}{\sigma_{y,d}^{(m)2}}\right)\gamma^{(m)}-\frac{p_d^{(m)}-2q_d^{(m)}\mu_{y,d}^{(m)}}{\sigma_{y,d}^{(m)2}}\right\}
\end{aligned} \tag{C.13}$$

where the sufficient statistics $\boldsymbol{p}^{(m)}$ and $\boldsymbol{q}^{(m)}$ are defined as

$$p_d^{(m)} = \sum_{t=1}^{T}\gamma_{o,t}^{(m)}y_{t,d}^2 \qquad\qquad q_d^{(m)} = \sum_{t=1}^{T}\gamma_{o,t}^{(m)}y_{t,d} \tag{C.14}$$

and the component posterior $\gamma_{o,t}^{(m)} = \mathrm{P}(m_t = m|\boldsymbol{O}, \mathcal{W}_{\mathsf{h}}; \mathcal{M}, \hat{\mathcal{M}})$.

References

[1] A. Acero. *Acoustical and Environmental Robustness in Automatic Speech Recognition.* PhD thesis, Carnegie Mellon University, 1990. 3.1

[2] A. Acero, L. Deng, T.T. Kristjansson, and J. Zhang. HMM adaptation using vector Taylor series for noisy speech recognition. In *Proc. ICSLP*, Beijing, China, October 2000. 4.3.2, 4.4.3, 5.4, B, B.2, B.2

[3] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul. A compact model for speaker-adaptive training. In *Proc. ICSLP*, 1996. 2.5.2, 2.5.5, 2.5.5, 7

[4] J.A. Arrowood. *Using Observation Uncertainty for Robust Speech Recognition.* PhD thesis, Georgia Institute of Technology, 2003. 1, 4.5, 4.5.1, 1, 5.7

[5] J.A. Arrowood and M.A. Clements. Using observation uncertainty in HMM decoding. In *Proc. ICSLP*, Denver, Colorado, September 2002. 1, 4.5

[6] X.L. Aubert. A brief overview of decoding techniques for large vocabulary continuous speech recognition. In *ASR-2000*, 2000. 2.4.2

[7] L.R. Bahl, P.V. de Souza, P.S. Gopalakrishnan, D. Nahamoo, and M.A. Picheny. Context dependent modelling of phones in continuous speech using decision trees. In *Proc. DARPA Speech and Natural Language Processing Workshop*, pages 264–270, 1991. 2.3.3, 2.3.3

[8] J. Barker, M. Cooke, and P. Green. Robust ASR based on clean speech models: An evaluation of missing data techniques for connected digit recognition in noise. In *Proc. Eurospeech*, 2001. 4.5.3, 1, 4.5.3

[9] L.E. Baum and J.A. Eagon. An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. *Bull. Amer. Math. Soc.*, 73:360–363, 1967. 2.3.2

[10] C. Benítez, J.C. Segura, A. de la Torre, J. Ramírez, and A.J. Rubio. Including uncertainty of speech observations in robust speech recognition. In *Proc. ICSLP*, Jeju Island, Korea, October 2004. 4.5.1, 4.6, 5.2.2, 8.1.5

[11] S.F. Boll. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 27:113–120, 1979. 4.3.1

[12] M. Boros, W. Eckert, F. Gallwitz, G. Görz, G. Hanrieder, and H. Niemann. Towards understanding spontaneous speech: word accuracy vs. concept accuracy. In *Proc. ICSLP*, 1996. 2.4.3

[13] L. ten Bosch. Bridging the gap between human and automatic speech recognition. *Speech Communication*, 49:331–335, 2007. 1

[14] S.E. Bou-Ghazale and J.H.L. Hansen. Duration and spectral based stress token generation for HMM speech recognition under stress. In *Proc. ICASSP*, Adelaide, Australia, April 1994. 3.1

[15] S.E. Bou-Ghazale and J.H.L. Hansen. A comparative study of traditional and newly proposed features for recognition of speech under stress. *IEEE Trans. on Speech and Audio Processing*, 8, July 2000. 3.1

[16] W. Byrne. Minimum Bayes risk estimation and decoding in large vocabulary continuous speech recognition. *IEICE Special Issue on Statistical Modelling for Speech Recognition*, 2006. 2.3.5, 2.3.5

[17] S. Chen and R.A. Gopinath. Gaussianization. In *Proc. Advances in NIPS*, 2000. 2.5.4

[18] S. Chen, B. Kingsbury, L. Mangu, D. Povey, G. Saon, H. Soltau, and G. Zweig. Advances in speech transcription at IBM under the DARPA EARS program. *IEEE Trans. on Speech and Audio Processing*, 14:1596–1608, 2006. 1, 2.3.3

[19] 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. 4.5, 4.5.3, 4.5.3

[20] S.B. Davis and P. Mermelstein. Comparison of parametric representations for monosyllable word recognition in continuously spoken sentences. *IEEE Trans. on Speech and Audio Processing*, 28(4):357–366, 1980. 2.2

[21] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39:1–39, 1977. 2.3.2, 2.3.2

[22] L. Deng, A. Acero, M. Plumpe, and X.D. Huang. Large vocabulary speech recognition under adverse acoustic environments. In *Proc. ICSLP*, pages 806–809, Beijing, China, October 2000. 2.5.5, 2.5.5, 4.3.2, 4.3.2, 4.3.3, 4.4

[23] L. Deng, J. Droppo, and A. Acero. Exploiting variances in robust feature extraction based on a parametric model of speech distortion. In *Proc. ICSLP*, 2002. 4.5.1, 1

[24] L. Deng, J. Droppo, and A. Acero. Dynamic compensation of HMM variances using the feature enhancement uncertainty computed from a parametric model of speech distortion. *IEEE Trans. on Speech and Audio Processing*, 12(3), May 2005. 4.5.1

[25] J. Droppo and A. Acero. Noise robust speech recognition with a switching linear dynamic model. In *Proc. ICASSP*, Montreal, Canada, May 2004. 4.1

[26] J. Droppo, A. Acero, and L. Deng. Uncertainty decoding with SPLICE for noise robust speech recognition. In *Proc. ICASSP*, Orlando, Florida, May 2002. 1, 4.5.2, 4.5.2.1, 4.5.2.1, 4.5.2.1, 5.2.3, 8.1.2

[27] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley-Interscience, 2nd edition, 2001. 2.2, A.2, A.2, A.2

[28] Y. Ephraim. A Bayesian estimation approach for speech enhancement using hidden Markov models. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 40:725–735, 1992. 4.3.1

[29] Y. Ephraim. Statistical model based speech enhancement systems. *Proc. IEEE*, 80: 1526–1555, October 1992. 4.3.1

[30] Y. Ephraim, D. Malah, and B.-H. Juang. On the application of hidden Markov models for enhancing noisy speech. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 37:1846–1856, December 1989. 4.3.1

[31] G. Evermann and P.C. Woodland. Posterior probability decoding, confidence estimation and system combination. In *Proc. Speech Transcription Workshop*, College Park, MD, 2000. 2.1, 2.4.2

[32] G. Evermann, H.Y. Chan, M.J.F. Gales, T. Hain, X. Liu, D. Mrva, L. Wang, and P.C. Woodland. Development of the 2003 CU-HTK conversational telephone speech transcription system. In *Proc. ICASSP*, 2004. 1, 2.3.3, 3.3

[33] J.G. Fiscus. A post-processing system to yield reduced word error rates: Recogniser output voting error reduction (ROVER). In *Proc. ASRU*, pages 347–352, 1997. 2.4.2

[34] A. Franz and B. Milch. Searching the web by voice. In *Proc. COLING*, pages 1213–1217, 2002. 2.4.1

[35] B. Frey, L. Deng, A. Acero, and T.T. Kristjansson. Algonquin: Iterating Laplace's method to remove multiple types of acoustic distortion for robust speech recognition. In *Proc. Eurospeech*, Aalborg, Denmark, September 2001. 4.4.4, 4.6, 6.4

[36] B. Frey, T.T. Kristjansson, L. Deng, and A. Acero. Algonquin—Learning dynamic noise models from noisy speech for robust speech recognition. In *Proc. Advances in NIPS*, 2001. 6

[37] S. Furui. Speaker independent isolated word recognition using dynamic features of speech spectrum. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 34:52–59, 1986. 2.2.1

[38] S. Furui. 50 years of progress in speech recognition technology—Where we are, and where we should go—From a poor dog to a super cat, April 2007. Keynote Presentation, ICASSP. 1

[39] M.J.F. Gales. *Model-Based Techniques for Noise Robust Speech Recognition*. PhD thesis, Cambridge University, 1995. 2.2.1, 3.1, 3.2, 4.4.1, 4.4.2, 4.4.3, 5.4, 10.2, B, B.1

[40] M.J.F. Gales. The generation and use of regression class trees for MLLR adaptation. Technical Report CUED/F-INFENG/TR263, University of Cambridge, 1996. Available from http://mi.eng.cam.ac.uk/reports/index-speech.html. 2.5.1

[41] M.J.F. Gales. Semi-tied covariance matrices for hidden Markov models. *IEEE Trans. on Speech and Audio Processing*, May 1999. 2.3.1, 2.3.4, 2.3.4, 3.3, 9.1

[42] M.J.F. Gales. Cluster adaptive training of hidden Markov models. *IEEE Trans. on Speech and Audio Processing*, 8:417–428, 2000. 2.5.5

[43] M.J.F. Gales. Acoustic factorisation. In *Proc. ASRU*, 2001. 2.5.5, 2.5.5, 7

[44] M.J.F. Gales. Adaptive training for robust ASR. In *Proc. ASRU*, 2001. 2.5.5, 2.5.5

[45] M.J.F. Gales. Discriminative models for speech recognition. In *Information Theory and Applications Workshop*, UCSD, California, USA, 1997. 2.3.5

[46] M.J.F. Gales. Maximum likelihood linear transformations for HMM-based speech recognition. *Computer Speech and Language*, 12, January 1998. 2.5.1, 2.5.1, 2.5.2, 2.5.2, 2.5.2, 2.5.2, 2.5.5, 7

[47] M.J.F. Gales. Predictive model-based compensation schemes for robust speech recognition. *Speech Communication*, 25, 1998. 4.4, 4.4.2, 4.4.2, 5.4

[48] M.J.F. Gales and R.C. van Dalen. Predictive linear transforms for noise robust speech recognition. In *Proc. ASRU*, 2007. 1, 5.8, 8.1.4, 8.2.4

[49] M.J.F. Gales and P.C. Woodland. Mean and variance adaptation within the MLLR framework. *Computer Speech and Language*, 10:249–264, 1996. 2.5.1

[50] M.J.F. Gales and S.J. Young. An improved approach to hidden Markov model decomposition of speech and noise. In *Proc. ICASSP*, 1992. 4.4.2

[51] 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. 4.4.2

[52] M.J.F. Gales and S.J. Young. Robust continuous speech recognition using parallel model combination. *IEEE Trans. on Speech and Audio Processing*, 1996. 4.4.2

[53] M.J.F. Gales, D.Y. Kim, P.C. Woodland, H.Y. Chan, D. Mrva, R. Sinha, and S.E. Tranter. Progress in the CU-HTK broadcast news transcription system. *IEEE Trans. on Speech and Audio Processing*, 14:1513–1525, September 2006. 2.4.2, 3.3

[54] J.S. Garofolo, L.F. Lamel, W.M. Fisher, J.G. Fiscus, D.S. Pallett, and N.L. Dahlgren. DARPA TIMIT acoustic-phonetic continuous speech corpus CDROM, 1993. Available from NIST. 2.3.3

[55] J.L. Gauvain and C.-H. Lee. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Trans. on Speech and Audio Processing*, 2:291–298, 1994. 2.5.1, 4.4

[56] J.J. Godfrey, E.C. Holliman, and J. McDaniel. SWITCHBOARD: telephone speech corpus for research and development. In *Proc. ICASSP*, 1992. 1

[57] B. Gold and N. Morgan. *Speech and Audio Signal Processing: Processing and Perception of Speech and Music*. John Wiley & Sons, 2000. 2.2, 2.2

[58] Y. Gong. Speech recognition in noisy environments. A survey. *Speech Communication*, 16:261–291, 1995. 4.4

[59] A. Gunawardana. HTK complex back-end for Aurora 2. In *Proc. ICSLP*, September 2002. From Special Session on "Noise Robust Speech Recognition—Robust Algorithms and a Comparison of their Performance on the Aurora 2 & 3 Databases". 8.1

[60] R. Haeb-Umbach and V Ion. Soft features for improved distributed speech recognition over wireless networks. In *Proc. Interspeech*, 2004. 1

[61] R. Haeb-Umbach, X. Aubert, P. Beyerlein, D. Klakow, M. Ullrich, A. Wendemuth, and P. Wilcox. Acoustic modeling in the Philips Hub-4 continuous-speech recognition system. In *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, 1998. 2.3.1

[62] 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. 3.1

[63] J.H.L. Hansen, X. Zhang, M. Akbacak, U. Yapanel, B. Pellom, and W. Ward. Robust speech processing for in-vehicle voice navigation systems. In *ICA-2004: Inter. Congress on Acoustics*, volume 4, pages 2603–2606, Kyoto, Japan, April 2004. 2.2

[64] H. Hermansky. Perceptual linear predictive (PLP) analysis of speech. *Journal of the Acoustic Society of America*, 87(4):1738–1752, 1990. 2.2

[65] H. Hermansky. Mel cepstrum, deltas, double-deltas, ...—What else is new? In *Robust Methods for Speech Recognition in Adverse Conditions*, Tampere, Finland, 1999. 2.2

[66] H. Hermansky. Should recognizers have ears? In *Proc. of ESCA Tutorial and Research Workshop on Robust Speech Recognition for Unknown Communication Channels*, pages 1–10, France, 1997. 4.1

[67] H. Hermansky and N. Morgan. RASTA processing of speech. *IEEE Trans. on Speech and Audio Processing*, 2(4), October 1994. 1, 2.2, 4.2

[68] H.-G. Hirsch and D. Pearce. The Aurora experimental framework for the evaluation of speech recognition systems under noisy conditions. In *Proc. ASR-2000*, pages 181–188, September 2000. 4.3.2, 4.4, 4.6, 8.1

[69] J.N. Holmes and W.J. Holmes. *Speech Synthesis and Recognition*. Taylor & Francis, 2nd edition, 2002. 2.2

[70] J.N. Holmes, W.J. Holmes, and P.N. Garner. Using formant frequencies in speech recognition. In *Proc. Eurospeech*, Rhodes, Greece, September 1997. 4.5.1

[71] B.J. Huang, S.E. Levinson, and M.M. Sondhi. Maximum likelihood estimation for multivariate mixture observations of Markov chains. *IEEE Trans. Information Theory*, IT-32:307–309, March 1986. 2.3.2

[72] X.D. Huang, A. Acero, and H.W. Hon. *Spoken Language Processing*. Prentice Hall, 2001. 2.1, 2.2, 2.2, 2.2.1, 2.3, 2.3.2, 2.4.1, 2.4.2, 2.4.2, 2.4.2, 2.4.3, 2.5.1, 4.2

[73] Q. Huo and C.-H. Lee. A Bayesian predictive classification approach to robust speech recognition. *IEEE Trans. on Speech and Audio Processing*, 8(2), 2000. 4.5

[74] M.-Y. Hwang, W. Wang, X. Lei, J. Zheng, O. Çetin, and G. Peng. Advances in Mandarin broadcast speech recognition. In *Proc. Eurospeech*, 2007. 2.4.1

[75] B.-H. Juang, W. Hou, and C.-H. Lee. Minimum classification error rate methods for speech recognition. *IEEE Trans. on Speech and Audio Processing*, 5:257–265, 1997. 2.3.5

[76] J.-C. Junqua. The Lombard reflex and its role on human listeners and automatic speech recognizers. In *Proc. JASA*, pages 510–524, January 1993. 3.1

[77] J.-C. Junqua and Y. Anglade. Acoustic and perceptual studies of Lombard speech: Application to isolated-words automatic speech recognition. *Proc. ICASSP*, 1990. 3.1

[78] C. Kim, Y.-H. Chiu, and R.M. Stern. Physiologically-motivated synchrony-based processing for robust automatic speech recognition. In *Proc. ICSLP*, 2006. 4.2

[79] 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*, Rhodes, Greece, September 1997. 6.4

[80] 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. 4.4.3, 5.6, B

[81] D.Y. Kim, G. Evermann, T. Hain, D. Mrva, S.E. Tranter, L. Wang, and P.C. Woodland. Recent advances in broadcast news transcription. In *Proc. ASRU*, 2003. 9.1

[82] J. Koehler, N. Morgan, H. Hermansky, H.-G. Hirsch, and G. Tong. Integrating RASTA-PLP into speech recognition. In *Proc. ICASSP*, volume 1, pages 421–424, Albuquerque, New Mexico, 1994. 4.2

[83] S.S. Kozat, K. Visweswariah, and R. Gopinath. Efficient low latency adaptation for speech recognition. In *Proc. ICASSP*, 2007. 2.5.2

[84] T.T. Kristjansson. *Speech Recognition in Adverse Environments: a Probabilistic Approach*. PhD thesis, University of Waterloo, Waterloo, Canada, 2002. 4.4.4, 4.5, 4.5.2

[85] 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. 1, 4.5

[86] T.T. Kristjansson, B. Frey, L. Deng, and A. Acero. Towards non-stationary model-based noise adaptation for large vocabulary speech recognition. In *Proc. ICASSP*, 2001. 6, 10.2

[87] N. Kumar. *Investigation of Silicon-Auditory Models and Generalization of Linear Discriminant Analysis for Improved Speech Recognition*. PhD thesis, John Hopkins University, 1997. 3.3, 9.1

[88] L. Lee and R.C. Rose. Speaker normalisation using efficient frequency warping procedures. In *Proc. ICASSP*, 1996. 2.5.5

[89] C.J. Leggetter and P.C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density HMMs. *Computer Speech and Language*, 9:171–186, 1995. 2.5.1, 2.5.1, 7

[90] H. Liao and M.J.F. Gales. Uncertainty decoding for noise robust speech recognition. Technical Report CUED/F-INFENG/TR499, University of Cambridge, 2004. Available from http://mi.eng.cam.ac.uk/reports/index-speech.html. 8.2.1.1

[91] H. Liao and M.J.F. Gales. Issues with uncertainty decoding for noise robust speech recognition. Technical Report CUED/F-INFENG/TR549, University of Cambridge, 2006. Available from http://mi.eng.cam.ac.uk/reports/index-speech.html.

[92] H. Liao and M.J.F. Gales. Joint uncertainty decoding for robust large vocabulary speech recognition. Technical Report CUED/F-INFENG/TR552, University of Cambridge, 2006. Available from http://mi.eng.cam.ac.uk/reports/index-speech.html.

[93] H. Liao and M.J.F. Gales. Joint uncertainty decoding for noise robust speech recognition. In *Proc. Interspeech*, 2005. 4.3.2, 4.5.2, 5.2.1, 5.3, 5.4

[94] H. Liao and M.J.F. Gales. Issues with uncertainty decoding for noise robust speech recognition. In *Proc. Interspeech*, 2006. 4.5.3

[95] H. Liao and M.J.F. Gales. Adaptive training with joint uncertainty decoding for robust recognition of noisy data. In *Proc. ICASSP*, 2007.

[96] 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. 2.5.3

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

[98] R.P. Lippmann, E.A. Martin, and D.B. Paul. Multi-style training for robust isolated-word speech recognition. In *Proc. ICASSP*, 1987. 2.5.5, 4.4

[99] X. Liu, M.J.F. Gales, K.C. Sim, and K. Yu. Investigation of acoustic modelling techniques for LVCSR system. In *Proc. ICASSP*, 2005. 2.5.4

[100] A. Ljolje. The importance of cepstral parameter correlations in speech recognition. *Computer Speech and Language*, 8:223–232, 1994. 2.3.4

[101] 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. 4.3.1

[102] W. Macherey, L. Haferkamp, R. Schlüter, and H. Ney. Investigations on error minimizing training criteria for discriminative training in automatic speech recognition. In *Proc. Interspeech*, 2005. 2.3.5

[103] M. Mohri, M. Riley, D. Hindle, A. Ljolje, and F. Pereira. Full expansion of context-dependent networks in large vocabulary speech recognition. In *Proc. ICASSP*, 1998. 2.4.2

[104] S. Molau, F. Hilger, and H. Ney. Feature space normalization in adverse acoustic conditions. In *Proc. ICASSP*, 2003. 2.5.4

[105] R.K. Moore. Spoken language processing: Piecing together the puzzle. *Speech Communication*, 49:418–435, 2007. 1

[106] P.J. Moreno. *Speech Recognition in Noisy Environments*. PhD thesis, Carnegie Mellon University, 1996. 3.1, 4.3.2, 4.4.3, 4.6, 5.4, 6.2, 6.4, B

[107] N. Morgan, A. Stolcke Q. Zhu, K. Sönmez, S. Sivadas, T. Shinozaki, M. Ostendorf, P. Jain, H. Hermansky, Dan Ellis, G. Doddington, B. Chen, Ö. Çetin, H. Bourlard, and M. Athineos. Pushing the spectral envelope—aside. *IEEE Signal Processing Magazine*, pages 81–88, September 2005. 1

[108] A. Morris, J. Barker, and H. Bourland. From missing data to maybe useful data: soft data modelling for noise robust ASR. In *Proc. WISP*, Stratford-upon-Avon, England, March 2001. 1, 4.5.3, 4.5.3, 8.1.5

[109] L. Neumeyer and M. Weintraub. Probabilistic optimum filtering for robust speech recognition. In *Proc. ICASSP*, volume 1, pages 417–420, 1994. 4.3.2, 5.2.1

[110] L.R. Neumeyer, A. Sankar, and V.V. Digalakis. A comparative study of speaker adaptation techniques. In *Proc. Eurospeech*, 1995. 7

[111] J.J. Odell. *The Use of Context in Large Vocabulary Speech Recognition*. PhD thesis, University of Cambridge, 1995. 2.3.3

[112] M. Padmanabhan and M. Picheny. Towards super-human speech recognition. In *ASR-2000*, pages 189–194, Paris, France, 2000. 1

[113] D. Povey. *Discriminative Training for Large Vocabulary Speech Recognition*. PhD thesis, University of Cambridge, 2003. 2.3.5

[114] D. Povey and G. Saon. Feature and model space speaker adaptation with full covariance Gaussians. In *Proc. ICSLP*, 2006. 2.3.4

[115] D. Povey and P.C. Woodland. Minimum phone error and I-smoothing for improved discriminative training. In *Proc. ICASSP*, 2002. 9.1

[116] D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, and G. Zweig. fMPE: Discriminatively trained features for speech recognition. In *Proc. ICASSP*, 2005. 9.1

[117] 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. 8.2

[118] L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. of the IEEE*, 77(2):257–286, February 1989. 2.3, 2.3, 2.3.1, 2.3.1, 2.3.2

[119] B. Raj and R. Stern. Missing-feature approaches in speech recognition. *IEEE Signal Processing Magazine*, pages 101–116, September 2005. 4.5, 4.5.3, 4.5.3, 8.1.5

[120] B. Raj, M.L. Seltzer, and R.M. Stern. Robust speech recognition: The case for restoring missing features. In *Proc. Eurospeech, The Workshop on Consistent and Reliable Acoustic Cues*, Aalborg, Denmark, September 2001. 4.5.3, 4.5.3

[121] B. Ramabhadran, O. Siohan, L. Mangu, G. Zweig, M. Westphal, H. Schulz, and A. Soneiro. The IBM 2006 speech transcription system for European parliamentary speeches. In *Proc. Interspeech*, 2006. 2.3.3, 2.4.1, 2.4.2, 3.3

[122] A.-V.I. Rosti. *Linear Gaussian Models for Speech Recognition*. PhD thesis, University of Cambridge, 2004. A.1

[123] A. Sankar, L. Neumeyer, and M. Weintraub. An experimental study of acoustic adaptation algorithms. In *Proc. ICASSP*, 1996. 2.5.1

[124] G. Saon, G. Zweig, and M. Padmanabhan. Linear feature space projections for speaker adaptation. In *Proc. ICASSP*, 2001. 2.5.2

[125] R.W. Schafer and L.R. Rabiner. Digital representations of speech signals. *Proc. of the IEEE*, 63(4):662–677, April 1975. 2.2

[126] S.R. Searle. *Matrix Algebra Useful for Statistics*. John Wiley and Sons, 1982. A.2

[127] J.C. Segura, M.C. Benítez, , A. de la Torre, S. Dupont, and A.J. Rubio. VTS residual noise compensation. In *Proc. ICASSP*, 2002. 4.6

[128] C.W. Seymour and M. Niranjan. An HMM based cepstral-domain speech enhancement scheme. In *Proc. ICSLP*, pages 1595–1598, 1994. 4.3.1

[129] K. Shinoda and T. Watanabe. Speaker adaptation with autonomous control using tree structure. In *Proc. Eurospeech*, 1995. 2.5.1

[130] K.C. Sim. *Structured Precision Matrix Modelling for Speech Recognition*. PhD thesis, Cambridge University, 2006. 2.3.1

[131] R. Sinha, S.E. Tranter, M.J.F. Gales, and P.C. Woodland. The Cambridge University March 2005 speaker diarisation system. In *Proc. Interspeech*, 2005. 2.2

[132] O. Siohan, Y. Gong, and J. Haton. A Bayesian approach to phone duration adaptation for Lombard speech recognition. In *Proc. Eurospeech*, volume 3, pages 1639–1642, Berlin, September 1993. 3.1

[133] S. Srinivasan, N. Roman, and D. Wang. Exploiting uncertainties for binaural speech recognition. In *Proc. ICASSP*, 2007. 1

[134] A. Stolcke, X. Anguera, K. Boakye, O. Çetin, A. Janin, M. Magimai-Doss, C. Wooters, and J. Zheng. The SRI-ICSI spring 2007 meeting and lecture recognition system. *Proc. NIST Rich Transcription Workshop, Springer Lecture Notes in Computer Science*, 2007. 3.3

[135] V. Stouten, H. Van hamme, K. Demuynck, and P. Wambacq. Robust speech recognition using model-based feature enhancement. In *Proc. Eurospeech*, pages 17–20, Geneva, Switzerland, September 2003. 4.6, 5.4, 6

[136] V. Stouten, H. Van hamme, J. Duchateau, and P. Wambacq. Evaluation of model-based feature enhancement on the Aurora-4 task. In *Proc. Eurospeech*, pages 349–352, Geneva, Switzerland, September 2003. 4.6, 6.4

[137] V. Stouten, H. Van hamme, and P. Wambacq. Accounting for the uncertainty of speech estimates in the context of model-based feature enhancement. In *Proc. ICSLP*, volume I, pages 105–108, Jeju Island, Korea, October 2004. 4.3.3, 4.5.1

[138] V. Stouten, H. Van hamme, and P. Wambacq. Model-based feature enhancement with uncertainty decoding for noise robust ASR. *Speech Communication*, 2006. 4.3.3, 4.6

[139] 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. 2.5.4, 8.1.5

[140] A. de la Torre, A.M. Peinado, J.C. Segura, J.L. Pérez-Córdoba, C. Benítez, and A.J. Rubio. Histogram equalization of speech representation for robust speech recognition. *IEEE Trans. on Speech and Audio Processing*, 40(13):355–366, May 2005. 2.5.4

[141] V. Valtchev, J.J. Odell, P.C. Woodland, and S.J. Young. MMIE training of large vocabulary recognition systems. *Speech Communication*, 22:303–314, June 1997. 2.3.5, 9.1

[142] H. Van hamme. Robust speech recognition using missing feature theory in the cepstral or LDA domain. In *Proc. Eurospeech*, 2003. 4.5.3, 8.1.5

[143] A.P. Varga and R.K. Moore. Hidden Markov model decomposition of speech and noise. In *Proc. ICASSP*, 1990. 4.1

[144] A.P. Varga and H.J.M. Steeneken. Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems. *Speech Communication*, 12(3), 1993. 4.4, 8.2

[145] A.P. Varga, R.K. Moore, J. Bridle, K. Ponting, and M. Russell. Noise compensation algorithms for use with hidden Markov model based speech recognition. In *Proc. ICASSP*, 1988. 4.4.3, B

[146] T. Watanabe and K. Shinoda. Speech recognition using tree-structured probability density function. In *Proc. ICSLP*, 1995. 2.5.1

[147] M. Wölfel and F. Faubel. Considering uncertainty by particle filter enhanced speech features in large vocabulary continuous speech recognition. In *Proc. ICASSP*, 2007. 4.5.1

[148] P.C. Woodland, J.J. Odell, V. Valtchev, and S.J. Young. Large vocabulary continuous speech recognition using HTK. In *Proc. ICASSP*, 1994. 9.2

[149] H. Xu, L. Rigazio, and D. Kryze. Vector Taylor series based joint uncertainty decoding. In *Proc. Interspeech*, 2006. 5.4

[150] U.H. Yapanel and J.H.L. Hansen. A new perspective on feature extraction for robust in-vehicle speech recognition. In *Proc. Eurospeech*, 2003. 4.2

[151] U.H. Yapanel, J.H.L. Hansen, R. Sarikaya, and B. Pellom. Robust digit recognition in noise: An evaluation using the Aurora corpus. In *Proc. Eurospeech*, Aalborg, Denmark, September 2001. 2.5.3, 4.4

[152] S. Young, G. Evermann, M.J.F. Gales, T. Hain, D. Kershaw, G. Moore, J.J. Odell, D. Ollason, D. Povey, V. Valtchev, and P.C. Woodland. *The HTK Book (for HTK Version 3.3)*. University of Cambridge, March 2004. 2.2, 2.4.2, 2.4.3, 2.5.1, 8.2

[153] S.J. Young and N.H. Russell J.H.S. Thorton. Token passing: A simple conceptual model for connected speech recognition systems. Technical Report CUED/F-INFENG/TR38, University of Cambridge, 1989. Available from http://mi.eng.cam.ac.uk/reports/index-speech.html. 2.4.2

[154] S.J. Young, J.J. Odell, and P.C. Woodland. Tree-based state tying for high accuracy acoustic modelling. In *Proc. ARPA Workshop on Human Language Technology*, pages 307–312, 1994. 2.3.3

[155] D. Yu, L. Deng, X. He, and A. Acero. Use of incrementally regulated discriminative margins in MCE training for speech recognition. In *Proc. ICSLP*, 2006. 1, 2.3.5

[156] K. Yu and M.J.F. Gales. Adaptive training using structured transforms. In *Proc. ICASSP*, 2004. 2.5.5

[157] D. Zhu and Q. Huo. A maximum likelihood approach to unsupervised online adaptation of stochastic vector mapping function for robust speech recognition. In *Proc. ICASSP*, 2007. 2.5.5