UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

FAKULTÄT FÜR
ELEKTROTECHNIK,
INFORMATIK UND
MATHEMATIK

Transmission Error Robust Speech Recognition using Soft-Features

Zur Erlangung des akademischen Grades

DOKTORINGENIEUR (Dr.-Ing.)

der Fakultät für Elektrotechnik, Informatik und Mathematik
der Universität Paderborn
vorgelegte Dissertation
von

Dipl.-Ing. Valentin Ion
Bukarest

Referent:	Prof. Dr.-Ing. Reinhold Häb-Umbach
Korreferent:	Prof. Dr. H. Ney, RWTH, Aachen

Tag der mündlichen Prüfung: 13.08.2008

Paderborn, den 20.08.2008

Diss. EIM-E/244

Abstract

The fast-paced growth in cell phone usage experienced over the past few decades offers a huge potential market for speech enabled mobile services. A suitable technology is Remote Speech Recognition where the actual recognition task is carried out on a remote server in the network rather than on the mobile terminal. Despite the advantages of this client-server architecture, an inherent weakness is that the communication medium may introduce errors which impair recognition accuracy.

There are numerous research studies which have been concerned with methods aimed at the creation of remote speech recognition systems which are robust to transmission errors. A widely used error concealment technique is to replace the erroneously received speech feature by an estimate of the "true" transmitted one and the carrying out of recognition using the resulting point estimate. The improvement in recognition accuracy afforded by this technique has been limited, as the estimate does not perfectly match the transmitted value, i.e. is uncertain.

This thesis focuses on modification of the speech recognition framework to compensate for uncertain features. By reformulation of the classification problem we obtain a novel uncertainty decoding rule which, instead of a point estimate, employs the posterior probability density function of the clean feature. The conditional independence assumption, prevalent in Hidden Markov Model based ASR, is relaxed to obtain a feature posterior density that is conditioned on the complete feature vector sequence observed at the output of the communication channel. This is a more informative posterior than the one conditioned only on the current observation.

This novel decoding method is used to facilitate a transmission-error robust remote speech recognition system. It is shown how the clean feature posterior can be computed for communication links exhibiting either bit errors or packet loss. The probabilistic model which has been employed combines a priori knowledge about the clean features and bit reliability of the received data.

The proposed techniques are evaluated in experiments measuring recognition accuracy of small- and medium-vocabulary recognition tasks under various channel conditions. Recognition results are presented for two types of remote recognition: Distributed and Network Speech Recognition. In the latter case common Voice-over-IP codecs are employed.

Kurzfassung

Die Benutzung des Mobiltelefons, die sich in den letzten Jahrzehnten rasant verbreitet hat, bietet ein bedeutendes Entwicklungspotenzial für sprachbasierte Dienste an. Dafür ist die Remote-Spracherkennung eine geeignete Technologie, wobei für die Erfüllung der Erkennungsaufgabe, statt des mobilen Gerätes ein entfernter Server eingesetzt wird. Trotz der Vorteile einer Client/Server-Architektur, ist die Verschlechterung der Erkennungsgenauigkeit aufgrund Übertragungsfehler eine inhärente Schwachstelle dieses Verfahrens.

Die Robustheit der Remote-Spracherkennung gegen Übertragungsfehler wurde durch viele Forschungsarbeiten angesprochen. Eine sehr verbreitete Fehlerbehandlungstechnik basiert auf der Ersetzung des fehlerbehafteten empfangenen Merkmalsvektor durch einen Schätzwert des fehlerfreien Vektors. Der Schätzwert wird anschließend für die Klassifikation verwendet. Die durch dieses Verfahren ermöglichte Qualitätsverbesserung ist jedoch begrenzt, denn der geschätzte Merkmalsvektor stimmt nicht genau mit dem gesendeten Merkmalsvektor überein, d.h., der Schätzwert ist unsicher.

Diese Arbeit konzentriert sich auf die Änderungen in dem Rahmenwerk der Spracherkennung, die notwendig sind, um die Unsicherheitsinformation auszuwerten. Die neue Darlegung des Klassifikationsproblems ergibt eine neuartige Decodierregel, die anstatt einen Schätzwert anzuwenden, die Posterior-Verteilungsdichtefunktion des gesendeten Merkmalsvektors ausnutzt. Die Annahme, die häufig in der Hidden-Markov-Modellen basierten Spracherkennung gemacht wird, dass die einzelne Beobachtungen unabhängig voneinander sind, wird hier erleichtert. Somit hängt die Verteilungsdichtefunktion nicht nur von einer Beobachtung ab, sondern von der gesamten beobachteten Merkmalsvektorfolge. Dadurch wird die Aussagefähigkeit der Posterior-Verteilungsdichtefunktion erhöht.

Die neuartige Decodierregel ermöglicht die Realisierung eines gegen Übertragungsfehler robusten Remote-Spracherkennungssystems. Es wird aufgezeigt, wie die oben erwähnte Verteilungsdichtefunktion für Kommunikationsnetzwerke, die Bitfehlern oder Packetverluste aufweisen, ausgerechnet werden kann. Das zur Ausrechnung zugrunde gelegte wahrscheinlichkeitstheoretische Modell fasst sowohl A-priori Kenntnisse über den Merkmalsvektor als auch die Bitzuverlässigkeitsinformation über die empfangenen Daten zusammen.

Die Verbesserung der Robustheit unter verschiedenen widrigen Übertragungsumständen wird für Erkennungsaufgaben mit kleiner und mittlerer Vokabulargröße experimentell beurteilt. Die Ergebnisse für die beiden Modellen der Remote-Spracherkennung: verteilte und Netzwerk-basierte Spracherkennung sind dargestellt. Die Letztere setzte für Voice-over-IP verbreitete Sprachcodierungsverfahren ein.

Acknowledgments

The research reported in this thesis was carried out at the Dept. of Communications Engineering, University of Paderborn, Germany. I owe a great debt of gratitude and appreciation to Prof. Dr. Reinhold Häb-Umbach, my thesis advisor, for giving me the opportunity to work in his team, for introducing me in the world of speech recognition, and for directing this thesis through the years. I would also like to extend my appreciation to Prof. Dr. Hermann Ney for agreeing to be my second advisor.

My welcome at the University has been so very warm and cordial. A heart-felt ”thank you” to all my former colleagues! They have helped me in so many ways and made my stay a very enjoyable experience. Special thanks goes to Mr. Ernst Warsitz who provided me with the best possible support during my first days at Paderborn.

I am deeply appreciative to many of my colleagues and students for their understanding, patience, and indispensable cooperation. My colleagues Mr. Sven Peschke and Mr. Marc Lüdicke contributed directly to this research by setting up the simulation system for the GSM transmission. I gratefully acknowledge the assistance given by Dr. Thomas Hesse in correcting the manuscript.

I would like to express my gratitude to all those involved in making this research possible, especially to the Deutsche Forschungsgemeinschaft for their financial support (Contracts no. HA34455/2-1 and HA3455/2-3).

List of Acronyms

3GPP	3rd Generation Partnership Project
AMR	Adaptive Multi-Rate speech coder
ASR	Automatic Speech Recognition
AWGN	Additive White Gaussian Noise
BCH	error-correcting code invented by Hocquenghem, Bose and Ray-Chaudhuri
BIF	Bad Index Flag
BPC	Bayesian Predictive Classification
BPSK	Binary Phase Shift Keying
C/I	Carrier-to-Interference ratio
C1, C2, C3, C4	IP channel conditions, or packet loss patterns
CDMA	Code Division Multiple Access
CELP	Code Excited Linear Prediction
CRC	Cyclic Redundancy Check
CS-ACELP	Conjugate Structure Algebraic Code Excited Linear Prediction
DSR	Distributed Speech Recognition
EC	Error Concealment
ELRA	European Language Resources Association
EP	Error Pattern
ETSI	European Telecommunications Standards Institute
ETSI-AFE	ETSI advanced front-end for DSR

ETSI-DSR	ETSI Distributed Speech Recognition
FB	Forward-Backward
FEC	Forward Error Correction
FLI	Feature Loss Indicator
GBER	Gross Bit Error Rate
GMSK	Gaussian Minimum Shift Keying
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
GSM-EFR	GSM-enhanced full-rate speech coder 13 kbit/s
GSM-TCH/F4.8	GSM traffic channel full rate 4800 bps
HMM	Hidden Markov Model
HTK	Hidden Markov Model Toolkit
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
LC	Layered Coding
LSFs	Line Spectral Frequencies
M	Marginalization
MAP	Maximum A Posteriori Probability
MDC	Multiple Description Coding
MDT	Missing Data Theory
MFCC	Mel Frequency Cepstral Coefficient

MFT	Missing Feature Technique
MMSE	Minimum Mean Squared Error
MMSE0	MMSE considering 0^{th} order a priori knowledge
MMSE1	MMSE considering 1^{st} order a priori knowledge
MMSE1-dyn	MMSE1 using the source model with dynamic components
MOS	Mean Opinion Score
NFR	Nearest Frame Repetition
NSR	Network-based (Network) Speech Recognition
PCM	Pulse-Code Modulation
PDA	Personal Digital Assistant
PDF	Probability Density Function
PLC	Packet Loss Concealment
PLI	Packet Loss Indicator
PLP	Perceptual Linear Prediction
PPP	Point-to-Point Protocol
RSR	Remote Speech Recognition
RTP	Real Time Protocol
SES	Speech Enabled Services
SNR	Signal to Noise Ratio
SPARK	Speech Processing and Recognition Toolkit
SVQ	Split Vector Quantization
TCP	Transmission Control Protocol
TRAU	Transcoder and Rate Adaption Unit

UD	Uncertainty Decoding
UD0	UD considering the 0^{th} order a priori knowledge
UD1	UD considering the 1^{st} order a priori knowledge
UD1-dyn	UD1 using the source model with dynamic components
UDP	User Datagram Protocol
UDP-Lite	a variant of UDP, that deliver packets even if their checksum is invalid
UEP	Unequal Error Protection
VAD	Voice Activity Detection
VoIP	Voice-over-IP
VQ	Vector Quantization
WAcc	Word Accuracy of a recognition system
WER	Word Error Rate
WSJ0	Wall Street Journal 5000-words corpus
WV	Weighted Viterbi recognition
XOR	exclusive OR logical operation

List of Symbols

$\mathbf{x}_t$	the feature vector transmitted at time instance t
$\mathbf{x}_1^T$	a sequence of feature vectors $\mathbf{x}_1 \ldots \mathbf{x}_T$
$\mathbf{y}_t$	the feature vector received at time instance t
$\mathbf{b}_t$	the bit pattern at the input of the channel coder
$\mathbf{b}_t^{(i)}$	the i^{th} bit pattern from a set of 2^M patterns
$P(\mathbf{b}_t)$	a priori probability mass function of a bit pattern, i.e. the 0^{th} order a priori knowledge
$P(\mathbf{b}_t\|\mathbf{b}_{t-1})$	the probability mass function of $\mathbf{b}_t$ conditioned on $\mathbf{b}_{t-1}$, i.e. the 1^{st} order a priori knowledge
$\hat{\mathbf{b}}_t$	the decoded bit pattern, at the output of the channel decoder
$\mathbf{z}_t$	the channel state at the time instance t
$P(\hat{\mathbf{b}}_t\|\mathbf{b}_t, \mathbf{z}_t)$	the channel transition probability of the bit pattern conditioned on the channel state
$P(\mathbf{b}_t\|\hat{\mathbf{b}}_t)$	the posterior probability mass function of the sent bit pattern conditioned on the received bit pattern $\hat{\mathbf{b}}_t$
$P(\mathbf{b}_t\|\hat{\mathbf{b}}_1^T)$	the posterior probability mass function of the sent bit pattern conditioned on the received sequence of bit patterns $\hat{\mathbf{b}}_1^T$
$p(\mathbf{x}_t\|s_t)$	the observation probability of $\mathbf{x}_t$ in the state s_t, i.e. the acoustic model
$p(\mathbf{x}_t)$	a priori PDF of $\mathbf{x}_t$, i.e. the 0^{th} order a priori knowledge
$p(\mathbf{x}_t\|\mathbf{x}_{t-1})$	the PDF of $\mathbf{x}_t$ given $\mathbf{x}_{t-1}$, i.e. the 1^{st} order a priori knowledge
$p(\mathbf{y}_t\|\mathbf{x}_t)$	the instantaneous channel transition probability
$p(\mathbf{x}_t\|\mathbf{y}_t)$	the posterior PDF of $\mathbf{x}_t$ conditioned on the observed (received) $\mathbf{y}_t$
$p(\mathbf{x}_t\|\mathbf{y}_1^T)$	the posterior PDF of $\mathbf{x}_t$ conditioned on the observed sequence $\mathbf{y}_1^T$
$\mathbf{v}_t$	any of the seven subvectors of the feature vector produced by ETSI-AFE
$\Delta\mathbf{x}_t$	the dynamic component *delta* (velocity) of a feature vector
$\Delta^2\mathbf{x}_t$	the dynamic component *delta-delta* (acceleration) of a feature vector

Contents

Chapter 1

Introduction

The proliferation of mobile communications technology has been considered to be the most important lifestyle revolution of the century. While in its early stages of development the telephone was aimed at oral communication, it now in its new digital guise provides numerous additional services such as multimedia access, web surfing, email and instant messaging. Most of the new features need a more intensive and user-friendly interaction with the user than is available by using a small keypad or touch-screen. To facilitate the access to these new services one has to overcome the limitations of the traditional user interfaces. The extension of human/machine interface methods to include automatic speech recognition (ASR) technology has been received with great interest.

At the same time, the widespread adoption of wireless networks offers a huge potential for deployment of query-based information systems providing constantly updated travel information, stock price quotations, weather reports, etc. Voice operation of a device is far more agreeable and effective than manual methods: This is a key factor in expecting a rapid take up of such speech based services, once the technology has been perfected. In addition a speech-enabled interface permits “hands free” operation and is thus ideal for use by vehicle drivers etc..

To date, in the main, two approaches to a speech-enabled interface have been proposed. One is the embedding of the ASR technology into the mobile terminal itself. Some basic functionality, e.g. spoken name dialing, is already provided by most modern cell phones. However, for more complex applications requiring large-vocabulary ASR, like dictation, the technical limitations are still a hurdle. Limited computational resources and memory constraints constitute a challenge to achieving satisfactory recognition accuracy. Nontheless, energy consumption is strongly dependent on the computational burden and should be low in order to pro-

vide the battery supplied device with long autonomy times. Note that to date, competitive commercial dictation software available on PC platforms requires much more resources than cell phones can provide. Moreover, since speech recognition is still an area of dynamic development, costly updates of the cell phone ASR software might often be required. This limits the applicability of terminal based ASR solutions to relatively simple tasks.

In the other approach, so-called Remote Speech Recognition (RSR) the ASR task is hosted on a dedicated server in the communication network infrastructure. The task of the mobile terminal is reduced to capturing the speech signal, coding it into a suitable representation, and sending it over the communication channel to the remote server. In consequence, sophisticated applications like natural language understanding, translation and dictation become accessible even on low-end mobile terminals. The next section provides the necessary background to the two approaches to RSR: Network-based (NSR) and Distributed Speech Recognition (DSR). The latter constitutes the recommended approach to speech-enabled services in the third generation of mobile networks and it is reviewed in Section 1.2.

In RSR, a parameterization of the speech signal must be transmitted to the recognition server using the communication medium. The transmission may induce errors in the bitstream of compressed parameters. E.g. in a mobile network noise, fading, and interference may temporarily corrupt the bitstream inducing bit errors or, in packet-oriented transmission, erroneous or late datagrams are dropped resulting in packet loss. If real-time constrains and bandwidth limitations apply, the errors cannot be corrected by simple mechanisms like data retransmission.

In the context of RSR, this thesis addresses a problem which arrises as a consequence of an imperfect communication medium - the degradation of recognition accuracy due to transmission errors. Techniques which aim to reduce or even eliminate the effect of transmission errors on the quality as perceived by the consumer of the transmitted data are termed “error concealment” (EC) . In classical speech transmission, where the data consumer is a human, the EC attempts to reconstruct the signal, i.e. to reconstruct the original signal, so as to reduce the annoying effects of corrupted bits or lost packets. If the data consumer is a speech recognizer, as is in RSR, the goal of reconstruction is to reduce the word error rate. Chapter 2 presents the current state of research regarding EC techniques employed in a distributed ASR scenario.

The main idea of this work is that the speech recognizer, which is in fact a statistical classifier, can benefit from knowledge about the estimation error variance

or, in other words, the reliability of the reconstructed speech feature. The reconstructed speech feature, along with its reliability information, has been termed "soft-feature" since, initially, the reliability computation was based on "soft-outputs" from the channel decoder provided to the ASR unit [1]. Chapter 3 presents the objectives and the organization of this dissertation.

1.1 Remote Speech Recognition

Remote Speech Recognition is built on a client-server architecture. The basic rationale is to displace the computationally intensive part of ASR to a recognition server residing in the infrastructure. This allows the users to share network resources and facilitates service and technology upgrades. The task of the mobile client is reduced to generating a suitable parameterization of the input speech signal. Depending on what kind of speech parameterization is performed by the mobile client, we can distinguish between Network-based Speech Recognition and Distributed Speech Recognition. These are detailed in the following paragraphs.

1.1.1 Network-based Speech Recognition

In Network-based Speech Recognition the speech signal captured by the terminal is coded into a low bit rate data stream using a network specific conventional speech coding algorithm such as GSM-EFR (GSM-Enhanced Full-Rate) or AMR (Adaptive Multi-Rate). The resulting bitstream is transmitted over a dedicated speech channel to the server. The speech channel comprises specific channel coding, decoding and the actual physical wireless or wired channel. At the server side, the speech signal must be first re-synthesized from the received bitstream and subsequently processed using feature extraction in order to obtain speech features for ASR.

A typical NSR architecture is shown in Figure 1.1.

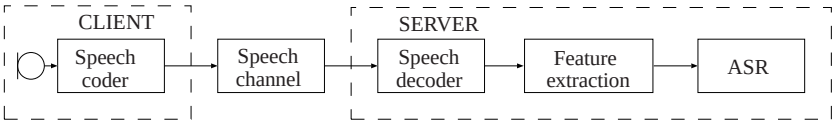

Figure 1.1: Block diagram of Network-based Speech Recognition

Note, that some variations from this architecture exist. For example terminals, such as Personal Digital Assistants (PDA's) , operating in a wireless environment may use voice-over-ip (VoIP) speech coding and a packet-switched channel. While the primary goal of speech coding is to achieve high compression rates with a gracefull degradation of subjective quality, experiments have proven that the recognition of decoded speech suffers from serious performance degradation [2]. This is the result of lossy compression and of channel errors. Hence, to avoid the reconstruction of signal waveform, some authors proposed the so called "bitstream based" NSR in which the speech features are directly computed from the coded speech [3, 4, 5]. Although bitstream based NSR is more robust, it cannot be used effectively if the transmission occurs over networks using different codecs, as is often the case nowadays, since this requires transcoding.

In conclusion, as the speech channel has to adhere to bandwidth and low latency constrains, NSR recognition performance is highly dependent on the codec and network conditions. These aspects have been extensively studied by researchers and are addressed by DSR.

1.1.2 Distributed Speech Recognition

An alternative approach to NSR is "Distributed Speech Recognition" (DSR). This eliminates the network dependent speech channel and uses instead a data channel to send coded features more suitable for recognition. By computing the speech features at the client side, lossy speech coding/decoding steps are avoided.

Figure 1.2 shows the block diagram of a DSR system. The ASR task is split between a "front-end" running on the terminal and a "back-end" hosted on the server. The front-end computes the speech features from the microphone signal. They are compressed, i.e. quantized in order to reduce the bit rate, and provided with error protection codes. The resulting low-bitrate data stream is sent over an error protected data channel. At the server side, the back-end has first to decompress the incoming data stream and mitigate the possible transmission channel errors. The speech features are then decompressed and used in ASR.

In the year 2000, the Aurora working group of the European Telecommunications Standards Institute (ETSI) standardized this approach to support the compatibility between various networks providing speech services. Section 1.2 gives an overview of DSR standardization by ETSI and describes the DSR front-end employed throughout this work.

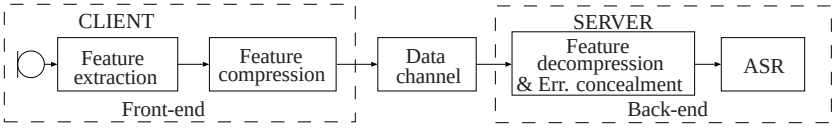

Figure 1.2: Block diagram of Distributed Speech Recognition

1.1.3 Network-based vs. Distributed Speech Recognition

This section gives a brief comparison between NSR and DSR in terms of recognition word accuracy, bit rates required, noise and transmission error-robustness, and compatibility with existing devices.

According to numerous studies, recognition word accuracy suffers due to using a speech codec before feature extraction [3], [2]. The low bit rate representation of the speech signal is not suitable for speech recognition. Moreover, due to nonlinear transformations, it is difficult to suppress additive ambient noise from the low bit rate coded speech. This leads to degraded performance in noisy environments. Hence, the main benefit of DSR is the avoidance of the use of a speech codec and, instead, performing the feature extraction at the terminal side on the original waveform.

In [6] it has been shown that the noise and channel robustness of DSR in a GSM network is superior to that of NSR using the GSM-EFR speech codec. The degradation is more pronounced in the case of large-vocabulary systems. They concluded that DSR provides a viable and robust alternative to NSR.

Another benefit of the DSR solution is the inherent support for multimodal interfaces and combined speech and data services. The data channel used to transmit the compressed features can be used as well to send other kinds of additional information. In implementing the same services using NSR, an additional data channel needs to be available, besides the dedicated speech channel.

Unlike speech recognition, some other speech-enabled applications require, e.g. for validation purposes, the signal waveform to be stored. While reconstruction of the speech waveform at the receiving end is the main goal of speech coding, a DSR codec does not necessarily provide this. MFCC features alone are not enough for speech reconstruction. However, the latest variant of the DSR standard includes pitch and voicing classes of each frame thus allowing for speech waveform reconstruction.

A very important economic advantage of NSR is its compatibility with any existing speech terminal. Implementation of an additional codec, i.e. the DSR front-end, is not necessary.

NSR and DSR were the subject of an extensive evaluation conducted by 3GPP (3rd Generation Partnership Project) aimed at selecting the most suitable codec for "Speech Enabled Services" (SES). The two candidate codecs were the ETSI-DSR codec and the AMR speech codec. Evaluation results published in [7] confirmed the advantage of the DSR solution over the network-based solution using the AMR speech codec in terms of recognition accuracy. The ETSI-DSR codec became the recommended codec for SES in 3GPP networks.

In conclusion the main benefits (+) and disadvantages (-) of NSR and DSR are summarized in Table 1.1.

Table 1.1: NSR vs. DSR

Criterion	NSR	DSR
Word Accuracy	-	+
Bit Rate	-	+
Noise Robustness	-	+
Transmission Error Robustness	-	+
Compatibility	+	-
Multimodality	-	+
Reconstruction	+	-,+

1.2 DSR Standardization

1.2.1 Overview of standardization activities

The compatibility between terminals and remote recognizers plays a crucial role in introducing the benefits of DSR to the wider mobile communications market. The STQ Aurora DSR Working Group of the European Telecommunications Standards Institute was formed to develop DSR standards that provide client-server interoperability. The standards need to adhere to a series of requirements as follows:

- Low data transmission rate
- Low computational and memory requirements for implementation in mobile terminals
- Low latency
- Robustness in noisy acoustic environments
- Robustness to transmission errors

The first DSR standard, the "ETSI standard front-end for DSR" (ETSI-SFE) [8] was published in February 2000. It defines the front-end processing stages to obtain speech features using a Mel-cepstrum algorithm and the compression algorithm to obtain a 4800 bps (bits per second) data stream. The feature extraction algorithm processes independently each 25 ms length segment of speech sampled at 8 kHz to generate a feature vector which consists of 13 cepstral coefficients and the logarithmic energy parameter. The frame overlap is 10 ms. The resulting feature rate is 100 vectors per second. Operation modes at 11 and 16 kHz sampling frequencies are also provided.

In order to evaluate the performance of the front-end in noisy conditions, an audio database has been designed and made publicly available through the European Language Resources Association (ELRA) under the name "Aurora 2". The database contains connected spoken digits originating from the TIdigits database. The clean speech signal has been mixed artificially with various real-world noises at signal-to-noise ratios between -5 and 20 dB.

An evaluation of the front-end in noisy conditions [9] showed that using acoustic models trained in clean conditions, the average accuracy over all noisy conditions dropped to 61% (although the word accuracy achieved on the clean test set was about 99%). By retraining the acoustic models under "multi-conditions", i.e. using both clean and noisy training data, the average accuracy increased to 86%. However, the main concern was that in a real-world scenario the statistics of the noise which contaminates the speech signal are not fully predictable at the training stage. Hence, since the models have not been trained with the specific noise, this multi-condition training might improve accuracy only to a small extent. The evaluation addressed also the degradation of accuracy due to low bit rate compression, but the observed loss of performance was negligible. An overview of the first ETSI standard for DSR can be found at [10]. The reference acoustic models and their WER performance are to be found at [9].

The next objective of Aurora DSR Working Group was to devise a noise robust front-end. In [11] the set of performance requirements for the so-called "advanced front-end for DSR" (ETSI-AFE) are specified. According to these, the new standard must provide at least a 50% performance improvement over the previous standard on the small-vocabulary task and under high mismatch conditions, i.e. without multi-condition training. The computational complexity must allow implementation within the typical resources of a cell phone terminal and must therefore not exceed that of the AMR speech codec used in GSM.

For evaluation purposes, two other databases were recommended. "Aurora 3", a small-vocabulary task consisting of spoken digits collected in noisy environments (driving vehicle) in five languages: Danish, Finnish, German, Italian and Spanish. The other is "Aurora 4", a large-vocabulary database obtained by artificial addition of noise to the Wall Street Journal (WSJ0) 5000-words corpus.

The ETSI Advanced Front-end for DSR [12] was published in October 2002, after a selection process. The key component in achieving noise robustness has been a noise reduction approach composed of two-stage Wiener filtering. The front-end achieved 53% reduction in word error rate [13] compared to the previous front-end.

The activity of the Aurora DSR Working Group continued with the development of an extended version of the standard which enables the reconstruction of the speech waveform from speech features and which has better support for tonal languages. This is the ETSI Extended Advanced Front-end for DSR [14], published in November 2003.

Regarding the protocols for the transport of compressed features, the DSR standards provide a packetization scheme for a circuit-switched data network. Transmission over packet-based network is also possible using the Real Time Protocol (RTP) payload for DSR recommended by Internet Engineering Task Force (IETF) in [15].

The experimental evaluations reported in this thesis have been performed using the ETSI Advanced Front-end for DSR. For this reason an overview of it is given in Section 1.2.2, however, it is only focused on transmission issues and robustness to transmission errors. The topic of robustness to environmental noise is not addressed.

1.2.2 The ETSI Advanced Front-end for DSR

Figure 1.3 depicts the functional blocks of the ETSI Advanced Front-end [12]. The front-end processing is distributed between the terminal side (*Terminal front-end*) and the server side (*Server front-end*). The terminal front-end specifies the algorithms for computation of the feature vectors from the sampled input speech (*Feature extraction*), for quantization of the speech feature vectors into a bitstream (*Feature compression*), and the subsequent processing of the bitstream (*Bitstream formatting* and *Error protection*) in order to be transmitted through the channel.

The server front-end specifies the inverse processing steps (*Bitstream decoding*, *Feature decompression*) required to obtain the feature vectors from the received bitstream. In addition, the *Server feature processing* block computes the feature temporal derivatives, i.e. delta and delta-delta features, resulting in the complete feature vector for ASR, usually having 39 components.

A brief description of each block is given in the remainder of this section.

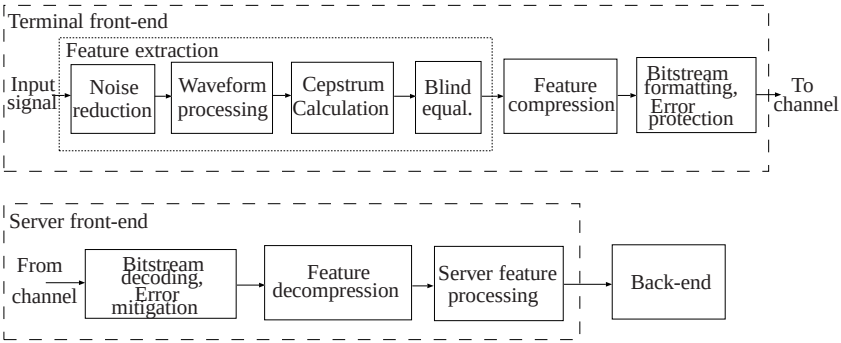

Figure 1.3: Block diagram of the ETSI Advanced Front-end for DSR

1.2.2.1 Feature extraction

The feature extraction part consists of:

- *Noise Reduction*
  This is performed basically by a two-stage Wiener filtering. The algorithm exceeds the scope of this work and therefore is not detailed.

- *Waveform Processing*
  The portions of signal with a good SNR are further enhanced in order to improve the overall SNR.

- *Cepstrum Calculation*
  Computation of cepstral coefficients (e.g. for 8 kHz) consist of: segmentation into overlapping frames of 25 ms (200 samples), multiplication with a Hamming window and zero padding to 256 samples, computation of magnitude spectrum, computation of 23-channels Mel frequency filter bank outputs, computation of the natural logarithm of each Mel filter output, and transformation of logarithmic filter bank outputs into 13 cepstral coefficients, $c_0, \ldots c_{12}$ using Discrete Cosine Transform.

- *Blind Equalization* The cepstrum is normalized in order to increase the robustness to convolutional distortion, e.g. caused by different microphone characteristics in training and testing.

The output of the feature extraction block is a 14-dimensional vector consisting of 13 MFCCs ($c_1, \ldots, c_{12}, c_0$) and the logarithm of frame energy $logE$.

1.2.2.2 Feature compression

The feature vector $(c_1, \ldots, c_{12}, c_0, logE)$ is compressed using a Split Vector Quantization (SVQ) schema. This is done by splitting the 14 dimensional feature vector into 7 two-dimensional subvectors $sv_1, \ldots, sv_7$ as shown in Table 1.2.

Table 1.2: Splitting and bit allocation scheme of ETSI Advanced Front-end

Subvector	sv_1	sv_2	sv_3	sv_4	sv_5	sv_6	sv_7
Features	c_1, c_2	c_3, c_4	c_5, c_6	c_7, c_8	c_9, c_{10}	c_{11}, c_{12}	$c_0, logE$
M	6	6	6	6	6	5	8

Each subvector is vector-quantized with M bits using its own codebook (which is provided by the standard). The front-end also computes a Voice Activity Detection (VAD) flag which is coded using one bit. The resulting 88 bits of two consecutive frames (feature vectors), i.e. a frame pair (FP), are completed using a 4-bit

CRC (Cyclic Redundancy Check) code for error detection purposes. This sums up to 92 bits (11.5 Bytes) per frame pair.

1.2.2.3 Framing, bitstream formatting and error protection

For transmission over circuit-switched channels the standard defines a multiframe bitstream format schematized in Figure 1.4. A multiframe consists of a synchronization sequence (2 Bytes), header field (4 Bytes) and 12 frame pairs (138 Bytes). This totals to 144 Bytes per multiframe and codes a portion of 240 ms of speech. The equivalent bit rate is 4800 bps. At the receiving end, the beginning of a multiframe in the bitstream is detected by means of the synchronization sequence. The header field contains information about the sampling rate, the front-end employed, and a multiframe counter. The synchronization and header data fields are critical and are therefore protected by a (31, 16) extended systematic code which is able to correct up to three bit errors and to detect up to 7 bit errors.

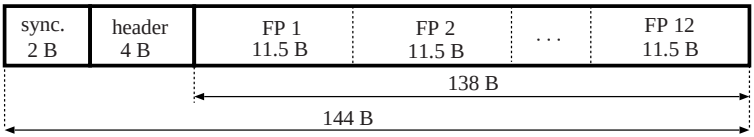

Figure 1.4: Multiframe format used for DSR over circuit-switched channels

In a packet-oriented network the transmission of the feature bitstream using Real Time Protocol (RTP) is envisaged. This is the de facto standard for real-time media streaming in IP networks. The data payload to be accommodated in a RTP packet was defined in [15]. Figure 1.5 shows the structure of a data packet. The first 40 Bytes are allocated for IP, UDP and RTP protocol headers. The following Bytes represent payload data. The choice of the number of frame pairs accommodated in a packet is flexible but must be specified in the RTP header. The choice of the number of frame pairs per packet must be chosen in accordance with data rate limitations, latency and channel robustness considerations. E.g. If it is one, only 12 Bytes of the total 52 Bytes are user data whereas the other 40 are header data. Bandwidth is not efficiently used in this case. On the other hand, fewer frame pairs per packet provides higher robustness against packet loss, since less information is lost on any one failure occasion. Regarding latency: it increases proportionally to the number of frame pairs per packet.

It is expected that future networks will provide a robust header compression scheme (RoHC) which will reduce the overhead from 40 to 4 Bytes and therefore even a packetization with one frame pair per packet will use the bandwidth efficiently. However, according to [7], for current GPRS data networks the usage of 4 or 8 frame pairs per packet is recommended.

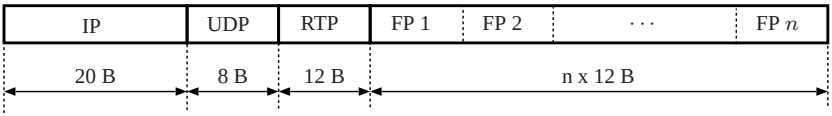

Figure 1.5: Packetization scheme for DSR over packet-networks. The packet contains n frame pairs (FP) representing $2n$ feature vectors.

In the case of a circuit-switched network, transmission errors may affect any of the bits of the multiframe. While the synchronization sequence and the header are relatively robust to bit errors, the bitstream of features is not. The standard provides a method for detecting the errors in a frame pair at reception. This consists of computing and appending a 4-bit CRC to the frame pair before transmission.

In the case of a packet-switched network, e.g. IP network, the packets which are erroneously received are usually rejected by the low-layer network protocols. The user either receives a correct packet or does not receive it at all. The absence of a packet in the received packet sequence is signalled by the RTP protocol for error concealment purposes. Note, that packet loss may also have other causes, as will be explained in Chapter 5.

1.2.2.4 Bitstream decoding, Error mitigation

The presence of bit errors is detected by two methods: CRC validation (media-independent FEC) and a data integrity check (media-specific FEC). On reception the 4-bit CRC is computed again from the received, possibly corrupted, bits and compared with the received CRC code. In case of mismatch the whole frame pair is deemed erroneous. Furthermore, a data integrity check is performed on the frame pairs sourrounding the erroneous one. The underlying rationale is that since the errors appear in short series (error bursts), it may be possible that the neighboring frame pairs are also erroneous but have not been detected as such by CRC. This may happen since CRC cannot detect every error pattern. The data integrity check is a heuristic algorithm based on the continuity of feature vector

components. The algorithm computes a *Bad Index Flag* (BIF) for each dimension of the feature vector as:

$$BIF_i = \begin{cases} 1 & \text{if } |x_i(t+1) - x_i(t)| > T_i \text{ or} \\ & |x_{i+1}(t+1) - x_{i+1}(t)| > T_{i+1} \\ 0 & \text{otherwise,} \end{cases} \tag{1.1}$$

where x_i, x_{i+1} are components of any subvector and T_i are subvector dependent thresholds estimated on training speech in absence of channel errors. If the BIF is one for at least two subvectors, the frame pair is deemed erroneous.

To increase robustness against transmission errors, the standard provides a simple error concealment method based on repetition of the nearest correct frame. If a sequence of L frame pairs ($2L$ feature vectors) encoded by the ETSI front-end for DSR has been corrupted, the first L feature vectors are replaced by copies of the last correct feature vector before the corrupted sequence, and the last L are replaced by copies of the first correct feature vector after the corrupted sequence. The sequence of corrupted or lost frames/features is called "error burst" or simply "burst" throughout this work.

Despite its simplicity, the error concealment method specified by the standard provides satisfactory robustness to relatively short error bursts, such as the loss of a single frame pair.

1.2.2.5 Feature decompression

Feature decompression consist of retrieving the multidimensional real-valued feature vector corresponding to a received bit pattern from the set of codebook centroids. This is a simple inverse mapping operation.

1.2.2.6 Server feature processing

The temporal derivatives delta and delta-delta denote the first and second-order temporal derivatives of the static feature components (MFCC and $\log E$) computed in the terminal front-end. They are computed from the static features by linear regression over a limited interval, see (6.24)-(6.25) for more details.

Chapter 2

Review of error-robustness techniques

2.1 Introduction

In Chapter 1 it has been shown that there are two main approaches to implementing a remote speech recognition system. The most straightforward approach is NSR as it can be directly deployed in the existing networks. On the other hand, DSR is more robust against lossy speech coding, noise and transmission channel errors.

A side effect of using speech channels with NSR is that the existing techniques for robust digital speech transmission are implicitly used. For example the channel coding and error concealment algorithm of a GSM speech channel are specified in the standard [16]. Therefore, the channel robustness of an NSR system deployed in GSM networks either relies on existing network/codec standard techniques, or requires the development of additional techniques, often on top of existing ones.

The goal of error-robustness techniques for speech transmission is to minimize the detrimental effect of errors on the perceptual speech quality, i.e. as it is perceived by the human listener. Such techniques were extensively studied since the beginning of digital speech transmission and are currently regarded as being at a relatively mature stage. A good reference regarding this topic will be found at [17]. However, it has been argued that the perceptual quality of speech quantified by the MOS (Mean Opinion Score) and the accuracy of an automatic speech recognition system WAcc (Word Accuracy) are not directly related. Therefore, robustness techniques aimed at improving perceptual quality do not necessarily achieve the optimum performance of ASR. An example is “muting”, a commonly used channel robustness technique for speech transmission. Muting prevents annoying noise patterns generated by erroneously received frames by replacing the corresponding regions of the synthesized waveform by silence. In speech recognition, if muting

occurs within a word, this leads mostly to end-point detection errors [18].

With NSR, transmission transmission errors affect the bitstream of speech codec parameters. Since one coded frame is usually involved in synthesizing several frames of the speech waveform, one transmission error smears over more speech features rendering them unreliable. In [19] it has been shown that if features are extracted directly from the bitstream of codec parameters, this effect is minimized. Gòmez et al. employed this technique for bitstream-based NSR using the GSM-EFR speech codec. The proposed transcoding technique exploits the bad frame indicator (BFI) flag for error mitigation purposes. Although though it even outperforms DSR in terms of channel robustness, the pitfall is that the BFI flag is usually not accessible at the server side. Unfortunately this implies modifications in the network, or, installing the feature extraction system mentioned above in the "Transcoder and Rate Adaption Unit" (TRAU) where the BFI is available [20].

With DSR, the transmission errors affect the bitstream of compressed features. An error which corrupts bits of one frame therefore affects only one feature vector, or even only some of its components, and has no effect on neighboring features. The error does not spread over more features as is the case with NSR.

This chapter gives an overview of the state of research on channel error-robustness techniques for DSR. They constitute the background for understanding the motivation for, and objectives of, this thesis. Error-robustness techniques for NSR operating in other that mel-cepstrum domain are not included in this scope, however, a study concerning the applicability of the proposed techniques to NSR using VoIP is detailed in Chapter 8.

Error-robustness techniques were categorized in transmitter-driven techniques and receiver-based techniques [20]. The first category requires direct participation of the transmitter. The participation can be active, as required e.g. for retransmission of an erroneous packet, or it can be passive, as in case of FEC and interleaving. The receiver-based techniques do not imply any coordination with the sender. The receiver alone has to detect the errors and to mitigate them: Involved in this are interpolation, estimation and recognizer-based techniques.

The transmitter-driven techniques are also referred to as error recovery techniques, while the receiver-based techniques are referred to as error concealment [21].

In the following, the error-robustness techniques are classified into error detection and recovery, error concealment based on "reconstruction" of lost or erroneous features, and error concealment based on modification of the classification

decision rule, i.e. ASR decoder-based techniques.

2.2 Error detection and recovery techniques

Error detection and recovery techniques rely on transmitting redundant information together with the bitstream of the source. This can be either the residual redundancy present in the source stream after source coding, or, it is artificially added by channel coding as "controlled redundancy". The amount of redundancy that one can afford to transmit depends on specific bit rate and latency limitations. While in some cases this amount can be enough even to enable error correction, in other cases it suffices to only detect it. Although by itself error detection does not directly contribute to channel robustness, it constitutes the basis for the error concealment. It has been proven that for remote speech recognition applications, error detection is a very important attribute. The authors of [22] have shown that the undetected channel errors may have a disastrous effect on recognition accuracy, but the recognizer can operate with no loss of accuracy with up to 15% channel erasure, when the erroneous frames are detected and simply erased.

Controlled redundancy can be added by FEC algorithms like block codes and convolutional codes or by sending same or similar information over different channels, as multiple description coding (MDC) or layered coding (LC) does. The joint source and channel coding exploits characteristics of the source to provide better error protection.

Interleaving does not add redundancy to the signal, but instead reorders the information before transmission in such a way that error bursts are spread over several frames. Thus, the loss of information per frame decreases which results in improved performance of the complementary error recovery or concealment techniques.

2.2.1 Forward Error Correction

With the Forward Error Correction (FEC) approach, the transmitter generates redundant data which is then combined with the source data and sent. At the reception end this redundancy is exploited to detect or even correct the errors. Depending on how the redundant information is created, there are principally two approaches in existence: linear block coding and convolutional coding.

In linear block coding the source bit stream is segmented into blocks of length k and each block is then independently encoded into a codeword of n bits. The code is referred to as (n, k) code and its error detection and correction capability is given by the minimum *Hamming distance*, d_{min} between any pair of code words. The code is able to:

- detect up to t errors if $d_{min} \geq t + 1$
- correct up to t errors if $d_{min} \geq 2t + 1$

Hamming, cyclic, Golay, BCH and Reed Solomon belong to the family of block codes. In convolutional codes the coder has memory and a coded bit not only depending on the current input to the coder but also on previous input. The information of one source symbol is distributed over more encoded bits and therefore better protected. Convolutional codes are particularly powerful since their decoding can provide a reliability measure for each decoded bit. Typical algorithms for decoding convolutional codes are the soft-output Viterbi algorithm [23] and the Max-Log-MAP [24]. The bit reliability measure can either be used for error detection, i.e. by imposing a reliability threshold to decide about bit correctness, as it was done in [25], or for more sophisticated error concealment algorithms like soft-feature reconstruction or ASR decoder-based techniques as will be shown in Section 2.3.3 and 2.4, respectively. Examples of channel coding using convolutional codes are the speech channel GSM-EFR and the data channel GSM-TCH/F4.8 in GSM networks [16].

For DSR, block codes are preferred due to their smaller delay and lower complexity as well as their independency between blocks [21]. In Section 1.2.2 we have seen that the ETSI-AFE algorithm employs a Golay code to protect the header of the multiframe and a 4-bit cyclic code to provide error detection to each feature pair. Tan and Dalsgaard [26] showed that error detection at the frame level, e.g. 4-bit CRC for each frame rather than for each frame pair, can considerably improve the performance of ETSI-AFE error concealment while the bitrate increases slightly from 4800 to 5000 bps. In their experiments the recognition word accuracy (WAcc) was raised from 47.1% to 85.6% under channel conditions with a 2% bit error rate.

In [22] the authors studied the performance of various linear block codes under fading channels when employing PLP (perceptual linear prediction) features for recognition. The vector consisted of 5 LSFs (line spectral frequencies) each quantized with 7 up to 10 bits. For packet-erasure networks, Reed-Solomon codes

were successfully employed in [27] in combination with interleaving.

2.2.2 Multiple description and layered coding

Multiple Description Coding (MDC) and Layered Coding (LC) utilize independent channels to deliver sub-streams of encoded source information. The degree of independence between channels ensures that it is very unlikely that all sub-streams will be corrupted simultaneously. In MDC the original signal can be reconstructed to a satisfactory quality from any of the descriptions [28]. If more descriptions are correctly received, they can be combined to increase the quality of the reconstructed signal.

MDC is suitable when all channels provide the same level of error protection. An example is the use of two IPv4 channels. However, when channels with unequal error protection (UEP) level are used, LC is a better choice. In LC there is a base layer stream and several enhancement streams. For the reconstruction the base layer is mandatory while the other layers are optional and can progressively refine the quality of reconstruction. However, they cannot be used alone. Thus, the base layer stream is transmitted using the channel providing highest protection and the other layers using channels with less protection. A possible application scenario is remote speech recognition over IP channels implementing the next generation Internet Protocol (IPv6) [29]. This provides priority assignment to the packets and implicitly UEP.

As a packet-loss recovery technique for voice transmission, MDC has been proven to outperform FEC techniques based on Reed-Solomon codes [30]. In the area of remote speech recognition there are also studies in existence about employing MDC and LC. Zhong et al. [31] suggest the superiority of the MDC technique in a VoIP-based NSR scenario. Srinvasamurthy et al. [32] proposed an efficient scalable speech compression method for NSR using layered coding based on a differential pulse code modulation (DPCM) with two loops. They generated a coarse and a fine reproduction of the MFCCs that constituted the base and the enhancement layer, respectively. The authors of [33] proposed the use of an MDC technique to transport the bitstream produced by the DSR front-end. They used two independent channels, one transmitting the sub-stream containing odd-numbered frames and other, the even-numbered. Since the redundancy between consecutive frames of the ETSI-AFE front-end is quite high, the full feature stream can be approximated at the reception from only one sub-stream by a single repetition of each of its frames.

2.2.3 Joint source and channel coding

Joint source-channel (de)coding is a research topic that has attracted much recent attention. Techniques in this category exploit certain properties, often termed *a priori knowledge*, of the source. This knowledge can be exploited in the design of the source coder, as it is done in channel-optimized vector quantization [34], or can provide "inherent" channel robustness as in channel-constrained vector quantization [35].

Another approach is to allocate the FEC codes of the channel coder according to the importance of the data. The underlying rationale is that for speech transmission we are actually not interested in minimizing the bit error rate, but the speech quality degradation. In the case of a Pulse-Code Modulation (PCM) encoded speech signal, for example, the most significant bits of the source symbols can be provided with more controlled redundancy than the less significant bits. The effect is that the most significant bits are better protected than the others thus increasing the overall transmission error robustness. The same amount of bandwidth available for channel coding can in this way be used more efficiently. Such a technique is utilized at channel coding of the bitstream generated by the GSM-EFR speech codec.

In another technique, termed "source-controlled channel decoding" [36, 37], the residual redundancy left in the bitstream after source coding is then exploited at the receiver end. In addition to the source redundancy, the information about the reliability of decoded bits, provided by the soft-output Viterbi algorithm [23], can be employed resulting in *soft decision* or *softbit source decoding*. Applications of this technique in robust speech decoding were presented in [38] and [39]. They provided good speech quality over GSM channels down to a C/I (carrier-to-interference) ratio of 6 dB whereas standard error concealment does badly already at a C/I of 7 dB.

Publications more focused on the remote speech recognition application are [25], [27], and [40]. Potamianos and Weerackody were the first to introduce the concept of *soft-feature speech decoding* in [25]. They used an UEP scheme for channel coding and computed the bit error probability of each received bit using the Max-Log-MAP algorithm [24]. Subsequently bit error probability was employed to estimate the confidence of each feature and the ASR decoder was modified to use that confidence, see also Section 2.4. Boulis et al. [27] proposed an UEP scheme for DSR over packet channels based on FEC with Reed-Solomon codes and combined with interleaving. Riskin et al. [40] addressed the issue of

how to assign unequal amounts of FEC to different sub-vectors to minimize the word error rate.

2.2.4 Interleaving

Interleaving itself can neither correct nor detect errors. It must be used in conjunction with other techniques to improve their robustness against error bursts. This is easy to understand if we consider channel coding employing linear block codes FEC. In Section 2.2.1 we have seen that each code is capable of correcting/detecting a certain number of errors. If errors occur in bursts it is likely that the capabilities of the code are exceeded and thus errors propagate to the source decoding. In general, error recovery and concealment techniques lose their effectiveness when the errors appear in long bursts [20].

By interleaving, the sequence of coded bits is rearranged before transmission so that error bursts affecting the reordered sequence are converted into random errors after restoring the original sequence order. This is relatively easy to implement by writing the coded bits into a matrix in row and reading for transmission in column order. At reception the inverse operation needs to be carried out.

Interleaving can also be implemented at frame level, i.e. reordering the sequence of feature vectors rather than the sequence of bits.

The main disadvantage of interleaving is the latency involved. In the previous implementation example, the first column can be transmitted only after the entire matrix has been filled. Whereas for speech transmission this latency is crucial, for remote ASR it is not. A reasonably small amount of latency is, however, desirable in order to ensure fluent interaction [20].

On the other hand, interleaving does not require extra bandwidth and can be easily combined with other error recovery and concealment techniques.

The performance of various interleaving schemas in a DSR scenario has been extensively studied by James and Milner [41].

2.3 Feature reconstruction techniques

In contrast to applications were data integrity is crucial, e.g. File Transfer Protocol (FTP); speech transmission and remote ASR are more tolerant of transmission errors. They can still perform satisfactorily using a degree of approximation

to the original signal when the original itself cannot be recovered. For example the speech signal synthesized from a moderately corrupted bitstream suffers some quality degradation but mostly remains intelligible.

Error concealment based on feature reconstruction attempts to create a replacement for the lost or erroneous packet which is then optimal with respect to a specific criterion. For example, in error concealment for speech transmission one attempts to minimize the perceptual speech quality degradation, whereas in a speech recognition application the goal is to minimize the word error rate.

In order to know when to perform reconstruction instead of normal source decoding, the erroneous frames must firstly be identified. Therefore error detection plays a crucial role in feature reconstruction. Errors can be detected and/or corrected by media independent FEC, as discussed in Section 2.2.1 or by media-specific FEC. The latter relies on intrinsic redundancy in the source to decide about the reliability of data as described in Section 1.2.2.4.

Generally, error detection is carried out on a frame (block of data) basis. That is, it can ascertain as to whether the block of data has been corrupted or not, but can neither indicate the number of errors nor their position in the block. It is therefore likely that some considerable part of the frame is still intact and this can be exploited in the reconstruction process. This is done by some sub-vector insertion-based techniques, e.g. [42].

Another powerful source of information to exploit at reconstruction are the statistical properties of the source signal. The reconstruction of a feature can be seen probabilistically as an estimation problem. Usually, the unknown value of the feature vector has to be estimated given a set of previous and/or following reliable values.

The remainder of this section reviews some widely used feature reconstruction techniques: insertion, interpolation, and statistics based reconstruction.

2.3.1 Insertion

Originally employed in audio streaming applications [43], this method inserts *fill-in* packets in the data stream where losses occurred. In GSM speech transmission it is known as muting or silence insertion. The filled-in frame can be simply silence or ambient noise, in which case it does not take into account any signal characteristics; it can also be an estimated value such as the mean of the error-free training data or it can be the nearest correctly received frame.

Boulis et al. [27] investigated usability of insertion methods as EC for DSR. Three approaches were tested: Replacing the erroneous frames with zeros (similar to replacement by silence), dropping the erroneous frames from the received sequence, and replacing the frame with a mean value computed on the training data. The latter gave the best performance and "dropping" the worst performance.

Insertion of a replica of the nearest correctly received frame, or simply "nearest frame repetition" (NFR), is the method adopted by the ETSI-DSR standard, see Section 1.2.2. After its publication, researchers have continued to address some problems manifested by this EC technique. One problem of NFR is that, in spite of its good performance in concealing short bursts, it fails when longer bursts occur. This can be attributed to the fact that repetition can exploit only the short-term self-similarities [43] of the signal. During longer error bursts the signal properties change and cannot be modeled appropriately by constant values along the burst. In this situation it is favorable to drop the features from the middle of the long burst. The method described in [44] has been termed *partial splicing* and consists in repetition applied at the extremities of the burst combined with splicing in the middle.

Another problem is that the error detection is carried out at the frame pair level. In case of error, the whole information content of that frame pair is disregarded, although possibly only one frame of the pair has been corrupted. The authors of [26] proposed the provision of each frame of the frame pair with CRC codes. They achieved significant improvement by doing so, however, this came at the price of increasing the required data rate and losing compliance with the standard.

Even if the error detection works satisfactorily at the frame level as proposed above, some information will still be disregarded. Several uncorrupted bits within an erroneous frame may be dropped possibly due to a one bit error. Tan et al. [42] have shown how this error-free portion of the frame can be exploited. They applied a data consistency check, similar to that of ETSI-DSR, to each subvector of a frame and identified the subvectors likely to be corrupted. The subvectors labeled as erroneous were replaced by copies of the nearest reliable subvector (subvector-based EC). Obviously, the data consistency check cannot determine exactly whether the data has been corrupted or not. However it was shown that this approach reduced the word error rate of a Danish digits task from 9.7% using the standard EC to 1.5%.

2.3.2 Interpolation

In contrast to insertion techniques, interpolation attempts to create a replacement for the lost or erroneous feature, which preserves the continuity of the original feature. Instead of replicating one (or two) feature values along the error burst as is done by insertion techniques, interpolation models the feature variation between a start value $\mathbf{x_1}$ which is the last correctly received feature before the burst and an end value $\mathbf{x}_T$ which is the first correctly received feature after the burst. In a general form the variation is described by a function f as:

$$\hat{\mathbf{x}}_t = f(t; \mathbf{x}_{-B+2}, \ldots, \mathbf{x}_1, \mathbf{x}_T, \ldots, \mathbf{x}_{T+F-1}). \tag{2.1}$$

$\hat{\mathbf{x}}_t$ represents the feature at time t relative to the beginning of burst ($1 < t < T$) obtained by interpolation using B reliable values before and F after the burst. The widely used approach is the Lagrange polynomial interpolation where the function f is a polynomial of degree $B + F - 1$. The polynomial coefficients depend on B past vectors $\mathbf{x}_{-B+2}, \ldots, \mathbf{x}_1$ and on F future vectors $\mathbf{x}_T, \ldots, \mathbf{x}_{T+F-1}$.

Milner and Semnani [45] deployed a linear interpolation for DSR in that the function was a first degree polynomial ($F = 1, B = 1$):

$$\hat{\mathbf{x}}_t = \mathbf{x}_1 + (t-1)\frac{\mathbf{x}_T - \mathbf{x}_1}{T-1}. \tag{2.2}$$

The feature trajectory is a straight line joining the points $\mathbf{x}_1$ and $\mathbf{x}_T$. Contrary to the general expectation that linear interpolation should perform better than repetition as the interpolated features are have a smaller Euclidian distance to the original ones, a number of publications [7, 46, 44] proved experimentally that it performs worse. The study of Tan et al. [42] reveals that it is not the Euclidian distance that accounts for the word error rate, but the *dynamic programming distance*.

James and Milner [41] have shown that repetition can be outperformed by interpolation using a cubic Hermite polynomial function ($B + F - 1 = 3$). This ensures not only the continuity of the feature trajectory but also that of the first derivative [20].

Some particular cases of interpolation are obtained with a 0^{th} degree polynomial: For $B = 1$, $F = 0$ and $\hat{\mathbf{x}}_t = \mathbf{x}_1$ we obtain the so called forward repetition, whereas for $B = 0$, $F = 1$ and $\hat{\mathbf{x}}_t = \mathbf{x}_T$ we obtain the backward repetition. This is why other authors [20] considered the EC of ETSI-DSR as belonging to the class

of interpolation techniques, being a combination of the forward and backward repetition described above.

Another aspect which must be considered is that an interpolation schema generates a latency of $T-2+F$ frames, with T being the burst length. This is higher than the latency of $T-2$ frames produced by the EC of ETSI-DSR.

2.3.3 Statistics based reconstruction

The previously presented repetition technique is more or less a crude method to exploit signal redundancy. If there was no residual redundancy in the coded source parameters, the consecutive frames would be statistically independent. In this case the repetition of the nearest correct frame would yield the same word error rate as randomly generating the fill-in frames. The rationale is that, assuming statistical independence, the frames within the burst do not depend on the other correct frames and thus, cannot be inferred from them. However, as it will be shown in Section 6.4.1, there exists a large amount of residual redundancy in the DSR source coded bitstream. The repetition technique models this redundancy by assuming that the speech feature changes slowly and thus the nearest received neighbor is a reasonable approximation of the lost feature, i.e. $\hat{\mathbf{x}}_t = \mathbf{x}_1$ on the first half and $\hat{\mathbf{x}}_t = \mathbf{x}_T$ on the second half of the burst.

In [47] this model of redundancy was referred to as a 0^{th} order data-source model. The authors have proven that the word accuracy can be improved using higher order data-source models. For example given a burst of length $2L$, in the 1^{st} order model, the sequence of L reconstructed values of the first half of the burst depends on the last value before the burst $\mathbf{x}_1$. However, unlike with simple repetition, the reconstructed values are not constant along the burst. Their sequence is the average of all sequences of length L of the clean training data prefixed by the VQ-index of $\mathbf{x}_1$. Thus, to each possible VQ-index there corresponds a sequence of length L stored in a lookup table. The second half of the burst depends on the $\mathbf{x}_T$ and is obtained in a similar fashion as the average of sequences suffixed by the VQ-index of $\mathbf{x}_T$. In the second-order data-source model, the reconstructed values depend on the two nearest correct frames. The approach has been even extended to a N^{th} order source, however, the memory requirements to store the lookup tables increase exponentially with N so that a special storage strategy is necessary. Nontheless, the limited amount of training data may also constitute a problem in estimating the lookup table if the number of table entries is too high, as is the case with $N > 1$.

In recent years a number of EC techniques have been developed that include the use of statistical a priori knowledge of the data in a more elaborate way. They rely on a priori information in form of speech models and employ estimation methods such as Maximum A Posteriori (MAP) or Minimum Mean Squared Error (MMSE) to reconstruct the clean feature.

MAP estimation has been successfully used to increase ASR robustness against environmental noise. In a first approach the problem of noisy observation has been cast into the missing feature framework, see also Section 2.4.1: the regions of the spectrogram that exhibit low signal-to-noise ratio are "deleted" resulting in incomplete spectrograms. Raj [48] proposed to reconstruct the missing components of the spectrograms using the surrounding clean speech components. The estimation criterion was to maximize the likelihood of the reconstructed components conditioned on observed (clean) components and a priori knowledge of clean data.

A similar approach to that described above has been applied as packet loss concealment (PLC) for DSR, i.e. EC for packet-loss channels, by James et al. [41]. They estimated the lost feature vectors so as to maximize their likelihood conditioned on the received feature vectors and on the a priori feature distribution. This method has been proven as more robust than cubic interpolation, particularly with regard to long bursts. A drawback was the high computational complexity due to the matrix inversion operations involved.

Gomez at al. [49] proposed to combine their data-source method [47] with the MAP estimation to trade-off memory reduction for computational expense.

Besides residual redundancy, another source of information which can be exploited for reconstruction is the error-free portion of erroneous frames. Subvector-based repetition utilizes the reliable subvectors of an erroneous vector yielding superior performance to vector-based repetition. In the first place, the error-free subvectors must be identified. This is relatively easy in a wireless communication scenario if the reliability information about each received bit is available as a complementary output of the channel decoder. Source decoding employing bit reliability has been addressed in a number of studies [39, 38, 37, 22, 46, 50]. Fingscheidt and Vary [39] termed their approach to EC for speech transmission *soft-bit speech decoding*. Instead of "hard"-decoding the speech waveform from the possibly erroneous received bits of coded speech parameters they softened the decoding by computing the MMSE estimate of the transmitted speech conditioned by previous hard-decoded speech parameters. Conditioning on more than one hard-decoded speech parameter allowed for modeling the dependencies between con-

secutive speech coded frames.

Peinado et al. [46] deployed a similar technique for DSR in that the source and channel were jointly modeled by a Hidden Markov Model (HMM), as depicted in Figure 2.1. The transmitted speech features, which are unobservable at the

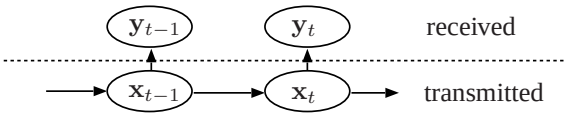

Figure 2.1: Source-channel modeled by HMM

receiver end, were represented by the hidden states $\mathbf{x}_t$ of the HMM. The observations were the received, possibly corrupted speech features $\mathbf{y}_t$. The probability of observing the received parameters $\mathbf{y}_t$ conditioned on each state $\mathbf{x}_t$ was computed using the bit reliability, similarly to Section 6.3 of this work. Subsequently, the computation of the posterior probability density of the transmitted speech features, here the model states, conditioned on the received one, here the observation sequence, reduces to one of the three fundamental problems of HMM theory [51]. This can be solved by the *forward-backward* (FB) algorithm. Knowing the feature posterior density, the MMSE estimate can be taken as the reconstructed feature:

$$\hat{\mathbf{x}}_t = E[\mathbf{x}_t|\mathbf{y}_1^T] = \int \mathbf{x}_t p(\mathbf{x}_t|\mathbf{y}_1^T) d\mathbf{x}_t. \tag{2.3}$$

Other approaches exploiting bit reliability for reconstruction are given in [25, 1, 22, 50]. However, they do not only reconstruct the erroneous feature but also provide it with a measure of reliability which is the variance of the estimator. This assembly has been initially termed *soft-feature* in [25]. As ASR decoding with soft-features requires modification of the speech recognizer, this is in fact within the scope of the next Section.

2.4 ASR decoder-based techniques

In classical speech transmission the recipient of the transmitted data is the human listener. The acoustic signal is captured by a microphone at the sending terminal end, parameterized and transmitted over a digital channel to the receiver. After

decoding, the resynthesized waveform is fed into the loudspeaker to generate a pattern of acoustic pressure waves which can be perceived by the human ear. Here, the goal of EC is to reconstruct the erroneous data such that the annoying effects of corrupted bits or lost packets [52] are reduced.

The feature reconstruction EC techniques for DSR act in a similar fashion, i.e. they reconstruct the lost or erroneous features, however, their goal is to reduce the degradation of recognition accuracy due to transmission errors. Unlike speech transmission where the only possible input to the human listener is the acoustic wave generated by the resynthesized waveform, in ASR the data recipient is a statistical classifier. In addition to the reconstructed speech feature, the classifier can also benefit from knowledge about the quality of the reconstruction. The reconstructed features are actually estimates of the true values and thus, reliable only to a restricted extent. Their contribution to the classification decision must be according to their reliability. Therefore, the feature deemed totally unreliable must produce no discrimination whereas the contribution of the reliable features is kept unchanged.

So far there three approaches to modification of the ASR decoder (classifier) have been proposed to take into account the feature reliability.

One approach is the missing feature technique (MFT). In this approach the reliability of a feature is quantified on two levels: either reliable or not. The reliable features are used in the conventional way whereas the unreliable ones have no contribution to recognition hypothesis, i.e. they are marginalized.

Another approach is the use of weighted Viterbi decoding. Here reliability is modeled by a weighting factor, usually denoted by γ, which takes values at the interval $[0; 1]$. The modification of an HMM based ASR decoder consist in raising the HMM state conditioned observation probability to the power of γ. Thus, a reliable feature has $\gamma = 1$ and its observation probability does not change. In contrast to this, a completely unreliable feature has $\gamma = 0$ and the observation probability becomes 1 independent of the HMM state and therefore does not produce discrimination between word hypotheses.

A relatively new approach termed “Uncertainty Decoding” (UD) has been recently investigated with the goal of attaining noise-robust ASR. In contrast to the other two approaches it involves a probabilistic formulation and avoids heuristics.

This section gives an overview of employing the first two methods in the context of channel-robust DSR. Uncertainty decoding constituted the starting point of the present work and had not been directly proposed as EC for DSR prior to our

work [50]. Therefore, an overview of the state of research on ASR using uncertainty decoding is first given in Section 4.2.

2.4.1 Missing Feature Theory

The Missing Data Theory (MDT), or Missing Feature Technique (MFT) in the context of speech recognition, has increasingly received researchers' attention over the past decade. The "missing data" problem has been initially examined in computer vision [53] where objects may be occluded by others such that only incomplete evidence is available for their identification. A similar problem occurs when parts of the spectro-temporal representation of the target speech signal are 'occluded' by environmental noise, by other competing speech signals or even by some band-limiting transfer function. Numerous research studies have investigated the application of MDT to noise-robust ASR and more recently to channel-robust ASR.

The fundamental idea of MFT for noise-robust ASR is to treat the noise dominated regions of the spectrogram as missing or unreliable [54]. Consequently, the classification relies solely on the regions dominated by the target speech. This approach requires solving a two-fold problem; on the one hand, the unreliable regions in the spectral representation must be detected and, on the other hand, the classification algorithm must be modified to handle incomplete data.

The identification of (un)reliable spectrogram regions given an additive mix of speech and noise is still an open research topic. A widely adopted solution is to evaluate the instantaneous SNR and declare reliable those regions where the local SNR exceeds some predefined threshold. The success of this approach is highly dependent on the accuracy of the SNR estimation. In a DSR system where the speech features such as Mel frequency cepstral coefficients are transmitted through the channel, the potential corruption due to the imperfect channel occurs in the time-cepstral domain. Identifying the missing regions is straightforward for packet-oriented transmission where the lost packets generate contiguous missing regions in the time-cepstral domain. When bit errors occur instead of packet loss, the unreliable regions can be identified as the feature vectors/subvectors failing the FEC check. Another possibility is to derive the feature reliability from the bit reliability if the latter is available as a by-product of soft-output channel decoding.

Regarding the second issue of MFT, i.e. employing the reliability of the data at the decoding, two solutions have been proposed [54, 48]: *data imputation* and *marginalization*. The former is in fact a feature reconstruction technique since it estimates the unreliable regions using a feature posterior density conditioned on

those of reliable regions and on the recognition hypothesis. The latter modifies the computation of the observation probability by considering only the reliable parts and integrating over, or marginalizing, the unreliable parts.

Potamianos and Weerackody applied MFT to EC for DSR over wireless channels in [25]. They coded each component of the feature vector separately into a binary codeword and used the Max-Log-MAP algorithm [24] to obtain the posterior probability for each bit of the decoded codeword. The bit was declared unreliable if its posterior probability was below a predefined threshold. Subsequently, the feature reliability was obtained by the use of the following heuristic: if the first and second bit of the codeword (in the sense of most significant bits) were unreliable, the complete feature component was deemed unreliable. Considering the feature vector as composed of reliable $\mathbf{x_{r}}_t$ and unreliable components $\mathbf{x_{u}}_t$, i.e. $\mathbf{x}_t = (\mathbf{x_{r}}_t, \mathbf{x_{u}}_t)$, the computation of the HMM state dependent observation probability $p(\mathbf{x}_t|s_t)$ was replaced by:

$$p(\mathbf{x_{r}}_t|s_t) = \int p(\mathbf{x}_t|s_t) d\mathbf{x_{u}}_t = \int p(\mathbf{x_{r}}_t, \mathbf{x_{u}}_t|s_t) d\mathbf{x_{u}}_t. \tag{2.4}$$

where s_t is the HMM state and $p(\mathbf{x}_{r,t}, \mathbf{x}_{u,t}|s_t)$ is the observation probability of the full feature vector, stored in the acoustic models.

Endo et al. [55] applied MFT to robust speech recognition over IP networks. They detected the packet (feature) loss by means of the sequence number in the RTP transport protocol header. The lost features were considered to have no reliable components resulting in an observation probability of one, cf. (2.4). The work claimed that marginalization is more effective than data imputation and splicing in the case of high packet loss ratio and long bursts.

James et al. [56] compared MAP estimation, cubic interpolation and marginalization in a packet-loss DSR environment. The results suggested that marginalization is more beneficial than the other two techniques, especially for long bursts. Thus, they concluded that ASR decoder-based techniques outperform feature reconstruction ones. They also investigated the problem of inferring the reliability of the temporal derivatives of the feature vector which are computed at the server side from the received static components. The best results were achieved by computing the derivatives from interpolated static components and performing marginalization of the feature only if its static components were lost.

2.4.2 Weighted Viterbi

Quantifying reliability with only two levels, reliable and unreliable, as it is done by MFT is disadvantageous for two reasons. Firstly, the features that are superficially damaged, though not enough to be declared corrupt, are used as if they were error-free ("true") transmitted features. Secondly, features declared uninformative, e.g. because some of their bits were considered erroneous, may contain useful information in the other bits which is lost in this way.

From a probabilistic perspective, the estimate of a feature is completely unreliable, or uninformative, if it is statistically independent of the "true" sent feature. This leads to the equality of the feature prior density with the posterior density of the sent features. The estimate is fully reliable if the conditioned density (posterior) is a Dirac delta function. In practice, however, intermediate situations occur in that the posterior density has an arbitrary form denoting that the estimate is neither fully reliable nor completely uninformative. For example in [57, 58] it has been found experimentally that in case of the NFR approach the estimated feature exhibits a high confidence level of being correct for the first and last frames of a burst, i.e. for those frames close to one burst end, but reliability decreases towards the middle of the burst. Clearly, a continuous measure of reliability would be more appropriate than a binary reliability indicator.

Even in packet-loss networks where the features are effectively lost, their estimate $\hat{\mathbf{x}}_t$ may become a useful source of information when considering the inter-frame correlation. This information is neglected by marginalization. In [59] it has been confirmed that repetition is slightly better than marginalization for short bursts, since in this case the NFR estimate is likely to be reliable.

In the weighted Viterbi approach the reliability of a feature is mapped onto a continuous parameter γ_t taking values in the interval $[0; 1]$. This allows a better modeling of intermediate situations in that the estimate is neither fully reliable nor completely uninformative.

This reliability is utilized in the speech recognizer to weigh the contribution of speech features to the acoustic likelihoods. This can be done by modification of the ASR decoder by raising the HMM state dependent observation probability to the power of γ_t:

$$\hat{p}(\mathbf{x}_t|s_t) = [p(\hat{\mathbf{x}}|s_t)]^{\gamma_t}. \tag{2.5}$$

The index t denotes that the exponential weighting factor varies with time.

In the view of Eq. 2.5 the unreliable observations, for which $\gamma_t \simeq 0$, yield an observation probability close to one, independent of the state s_t. In this way their contribution to the word hypothesis is neutralized. As reliability increases, γ_t tends to 1 and the observation probability gains discriminatory value.

A weighted Viterbi algorithm was proposed in [60] to increase the robustness of ASR against additive and convolutional noise. The basic idea was to take into consideration the reliability in noise canceling (spectral subtraction) by weighting each frame according to its segmental SNR. With acoustic models having a single Gaussian per state and diagonal covariance, the exponential weighting factor has been computed separately for each state as:

$$\gamma_{s_t} = \frac{1}{D} \sum_{d=1}^{D} \frac{\sigma^2_{s_t,d}}{\sigma^2_{s_t,d} + Var[\hat{x}_{t,d}]}, \tag{2.6}$$

where D is the dimensionality of the feature vector, $\sigma^2_{s_t,d}$ is the variance of the Gaussian of state s_t and $Var[\hat{x}_{t,d}]$ denotes the variance of the clean speech estimate. Obviously, in the clean conditions the clean speech estimate is nearly perfect and therefore the estimation variance tends to zero and the weighting factor equals one.

Potamianos and Weerackody [25] proposed to modify each m^{th} component of the Gaussian mixture independently:

$$\hat{p}(\mathbf{x}_t|s_t) = \sum_{m=1}^{M} c_m \prod_{d=1}^{D} [\mathcal{N}(\hat{x}_t; \mu_{m,d}, \sigma^2_{m,d})]^{\gamma_{t,d}}, \tag{2.7}$$

c_m being the weight and $\mu_{m,d}, \sigma^2_{m,d}$ the Gaussian mean and variance of the m^{th} mixture component of the state s_t. In order to obtain the factor $\gamma_{t,d}$ they defined first a confidence $C_{t,d}$ of each feature component as a function of the estimation variance $\sigma^2_{\hat{x}_{t,d}}$ and the a priori feature variance $\sigma^2_{apriori,d}$:

$$C_{t,d} = 1 - \frac{\sigma^2_{\hat{x}_{t,d}}}{\sigma^2_{apriori,d}} \tag{2.8}$$

The weighting factor $\gamma_{t,d}$ has been computed by smoothing the confidence according to:

$$\gamma_{t,d} = \frac{\alpha + C_{t,d}}{\alpha + 1}, \tag{2.9}$$

where the smoothing constant α has been tuned to obtain best performance on the test set.

Eq. 2.8 states that if the estimation variance $\sigma^2_{\hat{x}_{t,d}}$ equals the a priori variance $\sigma^2_{apriori,d}$, the confidence becomes zero, denoting uninformative feature. In contrast, if the variance of the estimator is close to zero, i.e. nearly perfect estimate, the confidence approaches one. In order to compute the estimation variance Potamianos and Weerackody employed the bit error probabilities delivered by the channel decoder, similarly to Eq. 2.3 but computing the centered second order moment of the estimator.

Bernard and Alwan [61] proposed to use the bit probabilities of the channel decoder to compute the weighting factor in a different way. They postulated that a good measure of reliability is the relative difference β of the first d_1 and second d_2 smallest Euclidian distances between any quantization codebook centroid and the received vector:

$$\beta = \frac{d_2 - d_1}{d_1}. \tag{2.10}$$

This relative Euclidian distance difference has been subsequently mapped to the factor γ_t by mean of a sigmoid function:

$$\gamma_t = \frac{1}{1 + e^{-21.8(\beta - 0.3)}} \tag{2.11}$$

Delaney proposed a stochastic weighted Viterbi recognition in [62] extending the idea given in [63]. Although classified as a weighted Viterbi approach, it is rather an uncertainty decoding approach [64, 65]. Instead of weighting the output probability Delaney modified the variance of the original acoustic models by the estimation variance, in this case the variance of interpolation error. This was computed prior to recognition as a function of the burst length and the relative position of the missing feature in the burst. Additionally, an experimentally tuned scaling factor has been applied to the estimation variance.

Bernard and Alwan [22] employed the weighted Viterbi algorithm for packet-loss concealment in DSR. They proposed to compute the exponential weighting factor based on the time auto-correlation of features. The weighting factor of the d^{th} component of the feature vector has been computed as:

$$\gamma_t = \sqrt{\rho_d(t - t_c)}, \tag{2.12}$$

$\rho_d(t)$ denoting the normalized auto-correlation function and $t - t_c$ the difference between the current time instance and that of the last correctly received feature.

A comprehensive study of the performance of weighted Viterbi decoding in a packet-network environment has been carried out in [57, 66]. The work investigated three methods of obtaining the weighting factor γ_t during the loss burst.

In a first approach γ_t was constant along the burst. The results on the Aurora 3 task (multi language small vocabulary speech database) in a DSR environment with packet loss have shown that the optimal value of γ_t depends strongly on the burst length. This dependency has been accounted for in the second approach where the weighting factor varied along the burst. The variation rule was chosen heuristically such that γ_t is close to one at the beginning and end of the burst and decreases toward the middle of the burst. Linearly and exponentially decreasing laws have been experimented with, with the latter giving the best performance. Considering for notational simplicity that the last received frame before the burst has the index $t = 1$ and the first after the burst has the index $t = T$, i.e. a burst of length T, the exponential variation law is given by:

$$\gamma_t = \begin{cases} \alpha^{t-1} & \text{for } t = 1, \ldots \frac{T}{2} \\ \alpha^{(T-t)} & \text{if } t = \frac{T}{2} + 1, \ldots T. \end{cases} \tag{2.13}$$

The parameter α has been given various values between 0 and 1 to find the best setting. The best results were obtained with $\alpha = 0.8$ consistently for all languages.

In the third approach they employed a feature vector component dependent variation law for the weighting coefficient rather than same one for all components. When weighting was performed separately for each component, the results were slightly better, however, with the drawback of increased computational load.

Cardenal et al. [66] attempted to avoid empirical choice of the variation rule and investigated a “probabilistic data-driven” approach. The weighting factor has

been estimated in advance as the cumulative distribution function of the Euclidian distance $d(\mathbf{x}_1, \mathbf{x}_t)$ during the first half of the burst:

$$\gamma_t = P(d(\mathbf{x}_1, \mathbf{x}_t) < \delta), t = 1, \ldots, \frac{T}{2} \tag{2.14}$$

The parameter δ has been obtained by assuming a 95% confidence interval of the first repetition, i.e. $\gamma_2 = 0.95$, and then solving the equation $P(d(\mathbf{x}_1, \mathbf{x}_2) < \delta) = \gamma_2$. Interestingly, they observed that the weighting factor has the same variation in the first 4 frames of a burst for all languages of Aurora 3, but exhibits a language-dependent variation in the subsequent frames.

2.5 Discussion

Each error concealment technique presented in this chapter has its own advantages and disadvantages. Choosing the most appropriate one certainly depends on particular requirements of the DSR system. Beside the word error rate, a performance comparison has to also consider other factors, e.g. bit-rate, computational complexity, and compatibility with the ETSI-DSR standards. Such performance comparison of some error-robustness techniques are given in [21].

Comparisons of decoder- based EC techniques in terms of WER performance were presented in our publication [59] and in the chapter “Error Concealment” of the more recent [52]. Several experimental results using various EC techniques can be found in the extensive work of Peinado [20]. Note, however, that the absolute word error rates may depend from site to site due to inherent variations in training of the acoustic models, different recognition engines, possible specific details of numerical implementation, and differing alignment of the error pattern with the bit stream of speech features. This obviously makes an accurate comparison between results obtained by different research groups difficult.

According to [21], in a DSR scenario over GSM under adverse channel conditions, a Multiple Description Coding (MDC) scheme using two descriptions, was ranked first in terms of WER performance. The feature extraction of ETSI-DSR was used and the descriptions were the bit streams of odd- and even-numbered frames, respectively. In spite of its best performance, the technique is not compatible with the ETSI standard for DSR as it needs a bit rate of 5200 bps (2600 bps per stream), two uncorrelated channels, and another payload type. The MMSE estimation using bit reliability and assuming the first-order Markov source [67] (also

denoted MMSE1 in the present work) achieved the second best performance followed by MAP estimation. These are compatible with the ETSI standard for DSR however the drawback is a high computational load at the server side of the DSR system. Frame reconstruction methods based on repetition have a lower computational load, such as subvector repetition, frame-level repetition and frame-pair repetition of the ETSI-DSR standard. Unfortunately, due to the simplified source model the performance is degraded, especially under channel conditions exhibiting longer error bursts. Performance can be further enhanced if long bursts are avoided by interleaving, but the compatibility with the standard is lost as the interleaved data stream is transmitted. Decoder-based techniques such as marginalization and weighted Viterbi have been outperformed in this scenario by the MMSE feature reconstruction technique.

In a DSR scenario over a packet channel, the marginalization and weighted Viterbi deliver better performance than the feature reconstruction techniques, including MMSE. Chapter 7 discusses the possible reasons for this and gives experimental results.

Chapter 3

Objectives and organization of this thesis

In remote speech recognition the errors occurring during transmission of the speech parameters from the speech capturing unit, e.g. cell phone, to the recognition server cause loss of recognition accuracy. The goal of error concealment techniques is to minimize the degradation and thus, increase the channel error-robustness of the remote speech recognition system. An overview of recent channel error-robustness techniques has been given in Chapter 2.

The decoder-based techniques reviewed in Section 2.4 are particularly interesting for remote speech recognition purposes. They either leave out the erroneous features from classification, as MFT does, or, modify their discrimination capabilities, as in weighted Viterbi. While these approaches are more or less heuristic, the first objective of this work is to provide a probabilistic well-founded framework for classification with unreliable features. This can be achieved by reformulation of the conventional Bayesian framework for ASR. A similar reformulation has been done in Bayesian Predictive Classification [68] where, however, the uncertainty is not in the observations but in the parameters of the acoustic model. To this end, Chapter 4 reformulates the conventional Bayesian framework of speech recognition such that the reliability in the speech features appears explicitly in the likelihood maximization expression (4.2). The reliability in the feature is modeled by the feature posterior density which denotes the probability density of the unobserved clean feature conditioned on all observed unreliable features. The reformulation leads to a novel decision rule which requires the feature posterior at each time instance. Hence, instead of evaluating the acoustic probability in a simple point estimate of the clean feature as in conventional ASR, the uncertainty decoding rule leads to an integration of the acoustic probability over the feature space. Here, special assumptions about the form of the feature posterior have to be made in order to obtain a numerically feasible solution.

A key element of the novel decision rule is the feature posterior. In order to properly estimate it in a remote speech recognition scenario, i.e. make it most informative, there are two sources of knowledge which can be exploited. One is the reliability of the received data in terms of individual bit error probabilities of the compressed feature, and the other is the redundancy in the feature sequence.

Chapter 5 aims at estimating data reliability, i.e. bit reliability information, in each of the two representative network types: a circuit-switched network where the errors consist of corrupted bits and a packet-switched network where the errors occur when the network drops contiguous sequences of bits, i.e. data packets. In [50], which built upon the concept of softbit speech decoding introduced by Fingscheidt and Vary [39], the bit reliability information of the channel decoder was used to compute the posterior probability of the transmitted bit pattern. This was then employed in the computation of feature posterior. In practice, the assumption that bit reliability information computed by the communication network channel decoder is available for reconstruction of corrupted features at the DSR back-end, is arguable. The channel decoder might not compute or at least not output this information, or, the transmission of the bit reliability information from the channel decoder to the DSR back-end requires bandwidth which one might not be willing to dedicate. Thus, another objective of this work is to explore methods to estimate the data reliability at the DSR back-end without requiring the soft-output information of the channel decoder.

Furthermore, the intention is to obtain an unified error concealment approach by separating the network-dependent data reliability estimation from the network-independent uncertainty decoding framework. This is an important feature since in the new generation of network protocols IPv6 or the existing UDP-Lite, bit errors and packet loss may coexist. It would therefore be beneficial if the error concealment method could deal with both types of error.

Chapter 6 describes how to estimate the feature posterior assuming models of various complexity for the redundancy in the feature vector sequence. The goal is to find a model which ensures a good trade-off between computational complexity and ability to realistically reproduce the redundancy. The simplest model assumes a memoryless source, i.e. does not utilize the temporal correlation between features. This assumption has been widely used in other works on uncertainty decoding regarding noise robustness, see Section 4.2. However, it yields poor performance when the errors affect the whole feature vector at once, as in case of packet loss. Employing the temporal correlation is expected to perform better. Furthermore, models including first and second order temporal derivatives

of the static feature components are studied.

The performance of a DSR system employing the proposed uncertainty decoding is evaluated in Chapter 7. The transmission from client to server is simulated for both bit-error and packet-loss network scenarios. The recognition tasks comprise a small- and medium-vocabulary task, Aurora 2 and WSJ, respectively. The word error rates versus transmission quality are evaluated using the proposed method. For comparison purposes, the word error rates achieved by some representative techniques reviewed in Chapter 2 are given.

Chapter 8 extends the application of the novel decoding rule to an NSR system where the coded speech is transmitted using voice-over-IP. Since most speech codecs used in voice-over-IP are provided with Packet Loss Concealment (PLC) algorithms which deliver the clean speech estimate in case of packet loss, the feature vectors can be computed straightforwardly. However, it is expected that considering the feature vector estimation variance caused by imperfect clean speech estimation, i.e. by PLC, the recognition performance can be improved.

Chapter 9 is dedicated to computational complexity issues. Employing the uncertainty decoding rule incurs a slowdown of the recognition process for at least three reasons; the computation of the feature posterior, the more complex expression of the observation probability, and the expansion of the acoustic search space due to reduced discrimination between word hypotheses during poor network conditions. It is desired to reduce the computational complexity eventually by further simplifying approximations, but without significantly degrading performance.

The relation between the other decoder-based error concealment techniques, e.g. marginalization and weighted Viterbi, and the novel uncertainty decoding rule is studied in Chapter 10.

A summary, and conclusions drawn in the Chapter 11, conclude this work.

Chapter 4

Uncertainty decoding for ASR

Statistical pattern classification requires knowledge of class conditioned probabilities and a priori class probabilities. They are used in Bayes' theorem to obtain the posterior probability of each class. The MAP decision rule, i.e. choosing the class with highest posterior probability, guarantees the minimum classification error. In Bayesian speech recognition, which is nothing else than a pattern classification problem encompassing the dimension of time, the classes are words modeled by temporal sequences of states. The class conditioned probabilities are known as "acoustic models" whereas the a priori probabilities of words are known as "language models".

The first section of this chapter reviews the classical Bayesian framework of speech recognition. Section 4.2 reviews the state-of-the-art approaches proposed by other authors to compensate for unreliability. The last section presents the novel approach of this thesis and how it is obtained by reformulating the classification task to consider unreliability in observations.

4.1 Bayesian framework of speech recognition

In ASR the aim of pattern classification is to map the sequence of feature vectors $\mathbf{x}_1^T = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$ to a sequence of words of a given vocabulary. This task comes down to finding the sequence of words $\hat{\mathbf{W}}$ which maximizes the joint probability $p(\mathbf{W}, \mathbf{x}_1^T)$ or, equivalently by using Bayes' theorem:

$$\hat{\mathbf{W}} = \underset{\mathbf{W}}{\operatorname{argmax}}\, p(\mathbf{W}, \mathbf{x}_1^T) \tag{4.1}$$

$$= \underset{\mathbf{W}}{\operatorname{argmax}}\, p(\mathbf{x}_1^T|\mathbf{W}) \cdot P(\mathbf{W}). \tag{4.2}$$

Eq. (4.2) is more convenient since it allows for separation into $P(\mathbf{W})$ which is the a priori probability of the word sequence $\mathbf{W}$ and $p(\mathbf{x}_1^T|\mathbf{W})$, the probability that the sequence of feature vectors $\mathbf{x}_1^T$ is emitted by uttering that word sequence. While the former term is a property of the *language model*, the computation of the latter is the main concern of the *acoustic model*. As the language model is independent of the observations, it is not affected by observation uncertainties.

A widely used acoustic modeling method is to represent the words as sequences of states in a Hidden Markov Model (HMM). Thus, a word or utterance is described by a certain sequence of hidden states, each state s_t emitting one observation $\mathbf{x}_t$ at a time. In a isolated word speech recognizer the sequence $\mathbf{W}$ consist of only one word. For example W_1 in Fig. 4.1 can be modeled by various sequences of states (or paths through the model): $s^{(1)}, s^{(1)}, s^{(2)}, s^{(3)}, s^{(4)}$ or $s^{(1)}, s^{(2)}, s^{(2)}, s^{(3)}, s^{(4)}$, etc., actually all possible paths starting in $s^{(1)}$ and ending in $s^{(4)}$. Since the length of the observation sequence $\mathbf{x}_1^T$ is known, only the paths of length T are allowed. Similarly, in a continuous speech recognizer $\mathbf{W}$ consists of more words resulting in sequences of states crossing the word boundaries, e.g. the path $s^{(1)}, s^{(1)}, s^{(2)}, s^{(3)}, s^{(4)}, s^{(5)}, s^{(6)}$ can model the word sequence W_1, W_2.

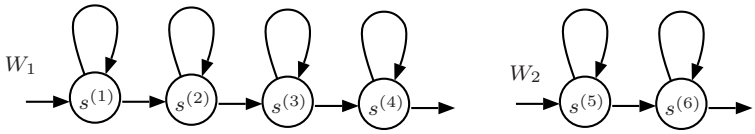

Figure 4.1: Words W_1, W_2 modeled as sequences of HMM states

Introducing the sequence of hidden states s_1^T underlying the sequence of observations $\mathbf{x}_1^T$ the acoustic probability can be expressed as:

$$p(\mathbf{x}_1^T|\mathbf{W}) = \sum_{\{s_1^T\}} p(\mathbf{x}_1^T, s_1^T|\mathbf{W}) \tag{4.3}$$

where the summation is carried out over all possible paths traversing the word sequence $\mathbf{W}$. This can be further written as:

$$\begin{aligned} p(\mathbf{x}_1^T|\mathbf{W}) &= \sum_{\{s_1^T\}} p(\mathbf{x}_1^T|s_1^T, \mathbf{W}) \cdot P(s_1^T|\mathbf{W}) \\ &= \sum_{\{s_1^T\}} p(\mathbf{x}_1^T|s_1^T) \cdot P(s_1^T|\mathbf{W}) \end{aligned} \tag{4.4}$$

The latter equality is supported by the fact that the conditioning on s_1^T and $\mathbf{W}$ in $p(\mathbf{x}_1^T|s_1^T, \mathbf{W})$ is equivalent to conditioning on only s_1^T, since the s_1^T implicitly models the word sequence.

Eq. 4.5 can be solved recursively if both terms under the sum are factorized:

$$p(\mathbf{x}_1^T|s_1^T) = \prod_{t=1}^{T} p(\mathbf{x}_t|\mathbf{x}_1^{t-1}, s_1^T) \tag{4.5}$$

$$P(s_1^T|\mathbf{W}) = \prod_{t=1}^{T} P(s_t|s_1^{t-1}, \mathbf{W}). \tag{4.6}$$

Assuming the *conditional independence* which states that $\mathbf{x}_t$ is statistically independent of its neighboring feature vectors if s_t is given, (4.5) turns into:

$$p(\mathbf{x}_1^T|s_1^T) = \prod_{t=1}^{T} p(\mathbf{x}_t|s_t). \tag{4.7}$$

The conditional independence assumption allows the expression of the emission probability of the sequence as a product over individual state emission probabilities $p(\mathbf{x}_t|s_t)$. The emission probability of each state is usually modeled as a multivariate mixture of Gaussian densities whose parameters are learned from the training speech data. Consequently, the estimated parameters are representative for speech features exhibiting the same statistical properties as those of the training data set.

Eq. 4.6 can also be simplified by exploiting the Markov property of s_1^T that $P(s_t|s_1^{t-1}) = P(s_t|s_{t-1})$, thus:

$$P(s_1^T|\mathbf{W}) = \prod_{t=1}^{T} P(s_t|s_{t-1}, \mathbf{W}). \tag{4.8}$$

Using (4.7) and (4.8) in (4.5) we arrive at the well known result:

$$p(\mathbf{x}_1^T|\mathbf{W}) = \sum_{\{s_1^T\}} \prod_{t=1}^{T} p(\mathbf{x}_t|s_t) \cdot P(s_t|s_{t-1}, \mathbf{W}). \tag{4.9}$$

An approximate value of (4.9) is then computed by the Viterbi algorithm as:

$$p(\mathbf{x}_1^T|\mathbf{W}) \simeq \max_{\{s_1^T\}} \prod_{t=1}^{T} p(\mathbf{x}_t|s_t) \cdot P(s_t|s_{t-1}, \mathbf{W}). \tag{4.10}$$

The approximation assumes that there exists a path having a much higher probability than all others and which thus dominates in the sum.

4.2 State of research on decoding unreliable data

Perfect knowledge of acoustic and language models, as required by the Bayesian decision rule for minimal word error rate (4.2), poses some practical difficulties. On the one hand, the estimation of the acoustic models is often performed in an environment different from that of practical system usage. Therefore, the estimated class conditioned probabilities do not well represent the true ones. On the other hand, the language model probabilities are usually not trained on speech corpora with identical statistical properties as in testing. The problem is well-known in ASR as the mismatch between training and testing conditions. It has been already demonstrated in [69] that the "plug-in" rule which uses the estimated probabilities as if they were the true ones is not optimal in the case of mismatch. The probabilities or the parameters of probability density functions (PDF) are in fact estimates of the true values and thus have their own estimation error variances. They are uncertain to some degree. To maintain the optimality of the Bayesian decision rule, this uncertainty must be used in a novel formulation of the classification problem.

In the "Bayesian Predictive Classification" (BPC) [68] the vector of model parameters is described by a PDF rather than a sole point estimate. The variance of the PDF is a measure of uncertainty in model parameters. By integrating the observation probability over the model space the mismatch between training and testing conditions is reduced. The BPC framework can also be used to benefit from the

uncertainty in the feature. This requires the posterior PDF of the clean feature conditioned on the observed feature and performs integration over the feature space rather than model space.

In the context of noise robust speech recognition, employing the uncertainty in the feature at the decoding stage has been proposed in [70, 71, 64, 65] and yielded very promising results.

The optimal classification strategy proposed by Kristjanson et al. [71] was based on using the class posterior conditioned on noisy speech as:

$$p(\mathbf{y}_1^T|s_1^T) = \prod_{t=1}^{T} \int \frac{p(\mathbf{x}_t|\mathbf{y}_t)}{p(\mathbf{x}_t)} p(\mathbf{x}_t|s_t) d\mathbf{x}_t \cdot P(s_t|s_{t-1}), \tag{4.11}$$

where $\mathbf{y}_1^T$ is the sequence of observations, $\mathbf{y}_t$ the current observation, and $\mathbf{x}_t$ the current unobserved clean feature.

Comparing with the conventional class posterior conditioned on the clean feature sequence:

$$p(\mathbf{x}_1^T|s_1^T) = \prod_{t=1}^{T} p(\mathbf{x}_t|s_t) P(s_t|s_{t-1}), \tag{4.12}$$

it can be readily observed that the sole modification is changing the observation probability $p(\mathbf{x}_t|s_t)$ into $\tilde{p}(\mathbf{y}_t|s_t)$ given by:

$$\tilde{p}(\mathbf{y}_t|s_t) = \int \frac{p(\mathbf{x}_t|\mathbf{y}_t)}{p(\mathbf{x}_t)} p(\mathbf{x}_t|s_t) d\mathbf{x}_t \tag{4.13}$$

If the $p(\mathbf{x}_t)$ is assumed to be constant within the interval where $p(\mathbf{x}_t|\mathbf{y}_t) \cdot p(\mathbf{x}_t|s_t)$ is not zero, the denominator $p(\mathbf{x}_t)$ can be neglected, as done in [64], since it results in a multiplicative constant which does not affect the MAP decision. A closed form solution of (4.13) was then obtained assuming Gaussian feature posterior $p(\mathbf{x}_t|\mathbf{y}_t) = \mathcal{N}(\mathbf{x}_t; \mu_{\mathbf{x}_t|\mathbf{y}_t}, \mathbf{\Sigma}_{\mathbf{x}_t|\mathbf{y}_t})$ and a Gaussian mixture with weights $P(m|s_t)$ for $p(\mathbf{x}_t|s_t)$:

$$p(\mathbf{x}_t|s_t) \quad = \quad \sum_m P(m|s_t) \mathcal{N}(\mathbf{x}_t; \mu_m, \mathbf{\Sigma}_m). \tag{4.14}$$

Under these assumptions Eq. (4.13) becomes:

$$\begin{aligned}\tilde{p}(\mathbf{y}_t|s_t) &= \sum_m P(m|s_t) \int \mathcal{N}(\mathbf{x}_t; \mu_{\mathbf{x}_t|\mathbf{y}_t}, \mathbf{\Sigma}_{\mathbf{x}_t|\mathbf{y}_t}) \mathcal{N}(\mathbf{x}_t; \mu_m, \mathbf{\Sigma}_m) d\mathbf{x}_t \\ &= \sum_m P(m|s_t) \mathcal{N}(\mu_{\mathbf{x}_t|\mathbf{y}_t}; \mu_m, \mathbf{\Sigma}_m + \mathbf{\Sigma}_{\mathbf{x}_t|\mathbf{y}_t}) \end{aligned} \quad (4.15)$$

The last expression shows that in order to account for uncertainty in estimating the value of the clean feature $\mu_{\mathbf{x}_t|\mathbf{y}_t}$, the original acoustic models have to be adapted by increasing the variances of the Gaussian mixture density components by the estimation variance $\mathbf{\Sigma}_{\mathbf{x}_t|\mathbf{y}_t}$.

4.3 Bayesian framework in the presence of corrupted features

In a DSR scenario and many other practical cases, such as in noisy environments, there is a mismatch between training and testing conditions. That is, the probabilities of the acoustic model estimated on a training set are not representative of the speech features of the testing situation as the latter may be affected by other factors not present in training, e.g. channel errors or acoustic noise. In the following let us denote by $\mathbf{x}_1^T$ the sequence of clean features which are representative of the training conditions but are not directly observable. Instead of this, a corrupted version $\mathbf{y}_1^T$ can be observed. The corruption may either be caused by environmental noise, or by errors during transmission over a communication network in a remote ASR setup.

The relation between the observed $\mathbf{y}_1^T$ and the hidden $\mathbf{x}_1^T$ cannot usually be analytically described since it may depend on some unknown factors. Hence, it is more convenient to model it statistically as channel *transition probability* $p(\mathbf{y}_1^T|\mathbf{x}_1^T)$. In DSR, $\mathbf{x}_t$ is the channel input value and $\mathbf{y}_t$ the output. The channel transition probability is the PDF of the channel output for a given input. Obviously, in the case of error-free transmission, input and output are equal. The estimation of the transition probability in a remote ASR scenario is described in Chapter 5. In the following it is assumed that $p(\mathbf{y}_t|\mathbf{x}_t)$ is known.

In a remote recognition scenario the classification is carried out at the server side where the uncorrupted speech features $\mathbf{x}_1^T$ are not available. Thus, the conventional pattern recognition problem of Section 4.1 has to be reformulated. The

task is now to find the word sequence $\mathbf{W}$ most likely to yield $\mathbf{y}_1^T$ at the other end of the channel:

$$\hat{\mathbf{W}} = \underset{\mathbf{W}}{\operatorname{argmax}}\, p(\mathbf{y}_1^T|\mathbf{W}) \cdot P(\mathbf{W}). \tag{4.16}$$

The most straightforward approach to solve (4.16) is to retrain the acoustic models in the particular mismatch condition and apply the conventional decoding rule of Section 4.1. The practical drawback of retraining, i.e. learning $p(\mathbf{y}_1^T|\mathbf{W})$, is that it requires computing power and a large amount of noisy (corrupted) data. Instead of retraining the models, we can simply consider $\mathbf{y}_1^T$ as an estimate of $\mathbf{x}_1^T$ and use it in (4.2). But, this results in the well-known poor performance of speech recognition in mismatched conditions.

To maintain the optimality of the Bayesian framework but still using the acoustic models trained in error-free conditions, we introduce the sequence of clean speech features $\mathbf{x}_1^T$ in (4.16) as a hidden variable:

$$p(\mathbf{y}_1^T|\mathbf{W}) = \int_{\{\mathbf{x}_1^T\}} p(\mathbf{y}_1^T|\mathbf{x}_1^T) p(\mathbf{x}_1^T|\mathbf{W}) d\mathbf{x}_1^T, \tag{4.17}$$

where the notation $\{\mathbf{x}_1^T\}$ indicates that the integration has to be carried out over all possible feature sequences of length T. By again introducing the sequence of hidden HMM states s_1^T to model the acoustic probability $p(\mathbf{x}_1^T|\mathbf{W})$ we obtain:

$$\begin{aligned} p(\mathbf{y}_1^T|\mathbf{W}) &= \int_{\{\mathbf{x}_1^T\}} p(\mathbf{y}_1^T|\mathbf{x}_1^T) \sum_{\{s_1^T\}} p(\mathbf{x}_1^T|s_1^T) \cdot P(s_t|s_{t-1}) d\mathbf{x}_1^T & (4.18)\\ &= \sum_{\{s_1^T\}} \int_{\{\mathbf{x}_1^T\}} p(\mathbf{y}_1^T|\mathbf{x}_1^T) p(\mathbf{x}_1^T|s_1^T) d\mathbf{x}_1^T P(s_t|s_{t-1}). & (4.19) \end{aligned}$$

Applying Viterbi approximation similarly to Eq. 4.10, the observation probability of a corrupted sequence becomes:

$$p(\mathbf{y}_1^T|\mathbf{W}) \simeq \max_{\{s_1^T\}} \int_{\{\mathbf{x}_1^T\}} p(\mathbf{y}_1^T|\mathbf{x}_1^T) p(\mathbf{x}_1^T|s_1^T) d\mathbf{x}_1^T P(s_t|s_{t-1}). \tag{4.20}$$

In comparison to Eq. 4.10, the observation probability of a corrupted sequence considers the acoustic model probability $p(\mathbf{x}_1^T|s_1^T)$ and the channel transition probability $p(\mathbf{y}_1^T|\mathbf{x}_1^T)$. This is equivalent to a combined source-channel model with the emission probability given by:

$$p(\mathbf{y}_1^T|s_1^T) = \int_{\{\mathbf{x}_1^T\}} p(\mathbf{y}_1^T|\mathbf{x}_1^T)p(\mathbf{x}_1^T|s_1^T)d\mathbf{x}_1^T. \tag{4.21}$$

It is instructive to investigate (4.21) in the extreme cases of an error-free transmission and of completely corrupted observations. In error-free conditions the received $\mathbf{y}_1^T$ is equal to the transmitted $\mathbf{x}_1^T$. In our statistical framework this relation turns into $p(\mathbf{y}_1^T|\mathbf{x}_1^T) = \delta(\mathbf{y}_1^T - \mathbf{x}_1^T)$, where $\delta(.)$ denotes the Dirac delta function. The integration of (4.21) over $\mathbf{x}_1^T$ reduces to the evaluation of the originally trained acoustic probability $p(\mathbf{x}_1^T|s_1^T)$ at $\mathbf{x}_1^T = \mathbf{y}_1^T$. This is equivalent to carrying out the recognition with the received feature sequence. On the other hand, the case of completely uninformative observations is expressed by statistical independence between the received and the sent feature sequences, i.e. $p(\mathbf{y}_1^T|\mathbf{x}_1^T) = p(\mathbf{y}_1^T)$. Replacing this in (4.21) yields the same emission probability equal to $p(\mathbf{y}_1^T)$ independent of the state sequence, since the integral of $p(\mathbf{x}_1^T|s_1^T)$ over the feature space equals one. Therefore, in this case the recognizer relies only on the prior word probabilities $P(\mathbf{W})$ to find $\hat{\mathbf{W}}$.

In order to recursively compute the most likely path of the Viterbi algorithm, the observation probability (4.21) must be factorized, i.e. must be put in a form similar to (4.10). The following subsections present simplifying assumptions which allow us to elegantly factorize (4.21) and evaluate it in an existing HMM based recognizer.

4.3.1 Conditional independence assumption

In a first approximation we consider that the transmitted feature vector sequence contains no temporal auto-correlation, which is equivalent to conditional independence of $\mathbf{x}_t$ from its neighbors $\ldots, \mathbf{x}_{t-1}, \mathbf{x}_{t+1}, \ldots$ etc. The Bayesian network associated to this model is shown in Fig. 4.2.

Bayes' theorem allows decomposition of $p(\mathbf{y}_1^T|\mathbf{x}_1^T)$ as:

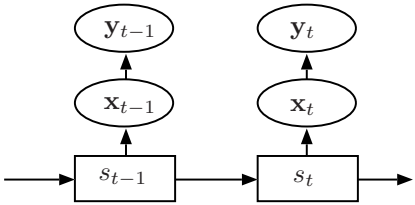

Figure 4.2: Bayesian network assuming conditional independence

$$\begin{aligned} p(\mathbf{y}_1^T|\mathbf{x}_1^T) &= p(\mathbf{y}_1,\ldots,\mathbf{y}_T|\mathbf{x}_1^T) \\ &= p(\mathbf{y}_1|\mathbf{y}_2,\ldots,\mathbf{y}_T,\mathbf{x}_1^T)\cdot p(\mathbf{y}_2,\ldots,\mathbf{y}_T|\mathbf{x}_1^T) \\ &= p(\mathbf{y}_1|\mathbf{y}_2,\ldots,\mathbf{y}_T,\mathbf{x}_1^T)\cdot p(\mathbf{y}_2|\mathbf{y}_3,\ldots,\mathbf{y}_T,\mathbf{x}_1^T)\ldots \quad (4.22) \end{aligned}$$

Considering, for example, the first term of the product (4.22), it can be readily observed that due to the statistical dependencies expressed by the graphical model, the conditioning on $\mathbf{y}_2,\ldots,\mathbf{y}_T,\mathbf{x}_1^T$ is equivalent to conditioning on $\mathbf{x}_1$. This equivalence can be explained on the basis of the Bayesian network by the fact that the graph node $\mathbf{y}_1$ can only be reached by traversing the node $\mathbf{x}_1$. Thus, the product (4.22) simplifies to:

$$\begin{aligned} p(\mathbf{y}_1^T|\mathbf{x}_1^T) &= p(\mathbf{y}_1|\mathbf{x}_1)\cdot p(\mathbf{y}_2|\mathbf{x}_2)\ldots p(\mathbf{y}_T|\mathbf{x}_T) \quad (4.23) \\ &= \prod_{t=1}^{T} p(\mathbf{y}_t|\mathbf{x}_t) \quad (4.24) \end{aligned}$$

Using (4.22) and (4.7) allows factorization of (4.21):

$$\begin{aligned} p(\mathbf{y}_1^T|s_1^T) &= \int_{\{\mathbf{x}_1^T\}} \prod_{t=1}^{T} p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|s_t)d\mathbf{x}_t \quad (4.25) \\ &= \prod_{t=1}^{T}\int_{\mathbf{x}_t} p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|s_t)d\mathbf{x}_t \quad (4.26) \end{aligned}$$

Assuming conditional independence, the reformulation of the Bayesian framework in the presence of corrupted features comes close to the approach presented

in [72] in the context of environmental noise compensation. A very similar result already discussed in Section 4.2 [71, 65, 73, 74] is obtained by applying Bayes' rule for conditional probabilities to (4.26):

$$p(\mathbf{y}_1^T|s_1^T) \propto \prod_{t=1}^{T} \int_{\mathbf{x}_t} \frac{p(\mathbf{x}_t|\mathbf{y}_t)}{p(\mathbf{x}_t)} p(\mathbf{x}_t|s_t) d\mathbf{x}_t. \tag{4.27}$$

Comparing the last equation with (4.7) yields that the uncertainty in observation is taken into account by integration over the feature space.

In other works [75], [76] the denominator $p(\mathbf{x}_t)$ has been neglected, which is definitely an approximation, but has not been identified as such. This approximation can be suggested on the grounds that the prior $p(\mathbf{x}_t)$ should have a larger variance than posterior. Thus the denominator can be considered constant for the range of values of $\mathbf{x}_t$, where the posterior density assumes values significantly larger than zero. However, this argument holds no longer in the presence of strong distortions, e.g. low SNR or long error bursts when the posterior $p(\mathbf{x}_t|\mathbf{y}_t)$ tends to equal the denominator. The use of the approximate decision rule, which neglects the prior density, results in poor performance [77].

The major drawback of the model presented in Fig. 4.2 is that it does not exploit the temporal correlation between consecutive features. Thus, the redundancy in the transmitted feature sequence is not used at all. The conditioning in the feature posterior $p(\mathbf{x}_t|\mathbf{y}_t)$ is only on the received $\mathbf{y}_t$. To illustrate this disadvantage, let us consider that the feature $\mathbf{y}_t$ is completely unreliable, whereas all other features of the received sequence are reliable. As discussed previously, complete unreliability denotes statistical independence or, $p(\mathbf{y}_t|\mathbf{x}_t) = p(\mathbf{y}_t)$. Using this in (4.26) the observation at time t is marginalized, i.e. it does not contribute to classification. On the other hand, it is known that the feature $\mathbf{x}_t$ could be estimated from neighboring features which, due to the strong correlation, would have resulted in performance better than marginalization, see also experimental results of Chapter 7.

Obviously, adopting a model able to handle the temporal correlation in the sequence $\mathbf{x}_1^T$ would be advantageous. To this end, the stringent conditional independence assumption is relaxed in the following section.

4.3.2 Relaxed conditional independence assumption

Whereas in the previous section the correlation between the feature vectors was neglected, in this section the dependency between features is approximated by a first-order Markov model:

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_{t-2,\ldots}) = p(\mathbf{x}_t|\mathbf{x}_{t-1}) \tag{4.28}$$

Figure 4.3 depicts the associated Bayesian network.

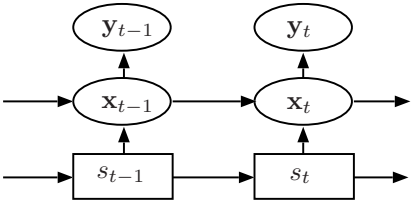

Figure 4.3: Bayesian network modeling temporal correlation between features

The factorization of $p(\mathbf{x}_1^T|s_1^T)$ becomes in this case:

$$\begin{aligned} p(\mathbf{x}_1^T|s_1^T) &= p(\mathbf{x}_T|\mathbf{x}_{T-1}, s_1^T)p(\mathbf{x}_{T-1}|\mathbf{x}_{T-2}, s_1^T)\ldots & (4.29)\\ &= p(\mathbf{x}_1|s_1)\prod_{t=2}^{T} p(\mathbf{x}_t|\mathbf{x}_{t-1}, s_1^T) & (4.30)\\ &\approx p(\mathbf{x}_1|s_1)\prod_{t=2}^{T} p(\mathbf{x}_t|\mathbf{x}_{t-1}, s_t) & (4.31)\end{aligned}$$

The last approximation was made to keep the observation probability computationally tractable and is justified by the fact that the dependency between $\mathbf{x}_t$ and $(\mathbf{x}_{t-1}, s_t)$ is stronger than between $\mathbf{x}_t$ and the other states (s_{t+1}^T) and therefore the latter can be neglected.

The probabilities $p(\mathbf{x}_t|\mathbf{x}_{t-1}, s_t)$ describe an acoustic model which takes into account the temporal correlation. However, in a conventional recognizer only $p(\mathbf{x}_t|s_t)$ are usually estimated during training. A convenient approximation for

$p(\mathbf{x}_t|\mathbf{x}_{t-1}, s_t)$ can be obtained assuming that $\mathbf{x}_{t-1}$ and s_t are statistically independent:

$$p(\mathbf{x}_{t-1}, s_t) = p(\mathbf{x}_{t-1})p(s_t) \tag{4.32}$$

Note that this assumption is less stringent than the conditional independence assumption of the previous section as it still captures some inter-frame correlation. Eq. 4.32 allows the double conditioning to be split into:

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}, s_t) = \frac{p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_t|s_t)}{p(\mathbf{x}_t)} \tag{4.33}$$

The observation probability of $\mathbf{y}_1^T$ becomes:

$$\begin{aligned}
p(\mathbf{y}_1^T|s_1^T) &= \int_{\{\mathbf{x}_1^T\}} p(\mathbf{y}_1^T|\mathbf{x}_1^T) \cdot p(\mathbf{x}_1|s_1) \prod_{t=2}^{T} \frac{p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_t|s_t)}{p(\mathbf{x}_t)} d\mathbf{x}_1^T \\
&= \int_{\{\mathbf{x}_1^T\}} p(\mathbf{y}_1^T|\mathbf{x}_1^T) \cdot \frac{p(\mathbf{x}_1)\prod_{t=2}^{T} p(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{x}_1)\prod_{t=2}^{T} p(\mathbf{x}_t)} \prod_{t=1}^{T} p(\mathbf{x}_t|s_t) d\mathbf{x}_1^T \\
&= \int_{\{\mathbf{x}_1^T\}} p(\mathbf{y}_1^T|\mathbf{x}_1^T) \cdot p(\mathbf{x}_1^T) \cdot \prod_{t=1}^{T} \frac{p(\mathbf{x}_t|s_t)}{p(\mathbf{x}_t)} d\mathbf{x}_1^T \\
&= \int_{\{\mathbf{x}_1^T\}} p(\mathbf{x}_1^T, \mathbf{y}_1^T) \cdot \prod_{t=1}^{T} \frac{p(\mathbf{x}_t|s_t)}{p(\mathbf{x}_t)} d\mathbf{x}_1^T.
\end{aligned} \tag{4.34}$$

The last expression can be rearranged as:

$$\begin{aligned}
p(\mathbf{y}_1^T|s_1^T) &= \prod_{t=1}^{T} \int_{\mathbf{x}_t} \frac{p(\mathbf{x}_t|s_t)}{p(\mathbf{x}_t)} \cdot \\
&\quad \left[\int_{\{\mathbf{x}_1^{t-1}\}} \int_{\{\mathbf{x}_{t+1}^T\}} p(\mathbf{x}_1^T, \mathbf{y}_1^T) d\mathbf{x}_1^{t-1} d\mathbf{x}_{t+1}^T \right] d\mathbf{x}_t \\
&= \prod_{t=1}^{T} \int_{\mathbf{x}_t} \frac{p(\mathbf{x}_t|s_t)}{p(\mathbf{x}_t)} p(\mathbf{x}_t, \mathbf{y}_1^T) d\mathbf{x}_t.
\end{aligned} \tag{4.35}$$

To obtain (4.35) we exploited that the integral of $p(\mathbf{x}_1^T, \mathbf{y}_1^T)$ over the space of feature vector sequences, excluding $\mathbf{x}_t$ is nothing else but the marginal density $p(\mathbf{x}_t, \mathbf{y}_1^T)$.

A more intuitive expression is obtained by applying Bayes' rule to $p(\mathbf{x}_t, \mathbf{y}_1^T)$ and leaving out the resulting term $p(\mathbf{y}_1^T)$, as it is irrelevant for the classification task, i.e. has the same contribution to all word hypothesis and thus does not change the maximum. Therefore we obtain:

$$p(\mathbf{y}_1^T|s_1^T) \quad \propto \quad \prod_{t=1}^{T} \int_{\mathbf{x}_t} \frac{p(\mathbf{x}_t|\mathbf{y}_1^T)}{p(\mathbf{x}_t)} p(\mathbf{x}_t|s_t) d\mathbf{x}_t. \tag{4.36}$$

The observation probability of expression (4.36) differs from (4.27) in that it is able to exploit the correlation in the sequence $\mathbf{x}_1^T$ since the posterior $p(\mathbf{x}_t|\mathbf{y}_1^T)$ is computed by conditioning not only on the instantaneously observed $\mathbf{y}_t$ but also on past and future observations. Comming back to the example with one uninformative observation $\mathbf{y}_t$ in a sequence of reliable ones, the observation probability (4.36) no longer results in marginalization. The discrimination is now possible as long as the feature posterior $p(\mathbf{x}_t|\mathbf{y}_1^T)$ does not equal the prior $p(\mathbf{x}_t)$ since the former is more informative than the latter. That is, some part of the lost information can be recovered from the reliable neighboring features. For this reason, the approach of this section has the potential of being superior to that of the previous one.

The computation of (4.36) or (4.27) requires knowledge in addition to the conventional acoustic model $p(\mathbf{x}_t|s_t)$. The feature prior $p(\mathbf{x}_t)$ can be easily estimated on the same training data as the acoustic model. The computation of the posterior, on the contrary, may turn out to be very complicated. A suitable model of the perturbation, be it acoustic noise or distortions due to the communication channel is additionally required, as well as the model of interaction with the clean speech feature. Chapter 6 shows how the feature posterior can be obtained for the case of remote speech recognition where the perturbation is the error prone transmission.

In conclusion, the reformulation of the Bayesian framework yields a modified expression of the observation probability which if factorized appropriately, can be further used in a Viterbi decoder. The original observation probability of the acoustic models $p(\mathbf{x}_t|s_t)$ changes either into:

$$\int_{\mathbf{x}_t} \frac{p(\mathbf{x}_t|\mathbf{y}_t)}{p(\mathbf{x}_t)} p(\mathbf{x}_t|s_t) d\mathbf{x}_t \tag{4.37}$$

if the inter-frame correlation is neglected, which is denoted by "Uncertainty Decoding considering the 0^{th} order a priori knowledge" (**UD0**) throughout this work, or into:

$$\int_{\mathbf{x}_t} \frac{p(\mathbf{x}_t|\mathbf{y}_1^T)}{p(\mathbf{x}_t)} p(\mathbf{x}_t|s_t) d\mathbf{x}_t \tag{4.38}$$

if inter-frame correlation is considered, denoted by "Uncertainty Decoding considering the 1^{st} order a priori knowledge" (**UD1**).

The following section presents a possible way to integrate the observation probability computation of (4.37) or (4.38) into an existing HMM based speech recognizer.

4.3.3 Integration into the recognizer: Gaussian assumption

It is well-known that the observation probability computation is the most time consuming processing step in a speech recognizer. Certainly, the numerical evaluation of the integral in the modified observation probability (4.37) or (4.38) would be prohibitive since it would increase the computational burden beyond the limits of practical interest. This section presents an approach to solving the integral analytically under the following simplifying assumptions:

1. The state conditioned observation probability of the uncorrupted feature is a Gaussian mixture:

$$p(\mathbf{x}_t|s_t) = \sum_{m=1}^{M} c_{s_t,m} \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_{s_t,m}, \boldsymbol{\Sigma}_{s_t,m}) \tag{4.39}$$

   where $c_{s_t,m}$ is the weight, $\boldsymbol{\mu}_{s_t,m}$ the mean vector and $\boldsymbol{\Sigma}_{s_t,m}$ the covariance matrix of the m^{th} mixture component of the observation probability of state s_t. This is a standard assumption in speech recognition.

2. The a priori probability density of the uncorrupted feature is a Gaussian:

$$p(\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{x}}). \tag{4.40}$$

Experimental data show that this assumption is quite valid, with certain reservations concerning the log energy component, its PDF having two modes. The Gaussian parameters $\boldsymbol{\mu}_{\mathbf{x}}$ and $\boldsymbol{\Sigma}_{\mathbf{x}}$ are learned from training data.

3. The feature posterior, given the observations, is Gaussian:

$$p(\mathbf{x}_t|\mathbf{y}) \approx \hat{p}(\mathbf{x}_t|\mathbf{y}) = \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_{\mathbf{x}_t|\mathbf{y}}, \boldsymbol{\Sigma}_{\mathbf{x}_t|\mathbf{y}}). \tag{4.41}$$

Here the notation $\mathbf{x}_t|\mathbf{y}$ stands for either $\mathbf{x}_t|\mathbf{y}_t$ or $\mathbf{x}_t|\mathbf{y}_1^T$, depending on which observation probability is going to be used, (4.37) or (4.38). Eq. 4.41 is the most debatable assumption, as we often observed a multi-modal shape of the posterior. Some researchers therefore suggested the use a Gaussian mixture model instead [76]. As this, however, has a major impact on the computational effort, we prefer to stick to a single Gaussian model here.

Under these assumptions the integral of (4.37) or (4.38) turns out to be the evaluation of the modified observation probability of an equivalent feature:

$$\sum_{m=1}^{M} c_{s_t,m} \int_{\{x_t\}} \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_{s_t,m}, \boldsymbol{\Sigma}_{s_t,m}) \frac{\mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_{\mathbf{x}_t|\mathbf{y}}, \boldsymbol{\Sigma}_{\mathbf{x}_t|\mathbf{y}})}{\mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{x}})} d\mathbf{x}_t$$

$$= \sum_{m=1}^{M} c'_{s_t,m} \mathcal{N}(\boldsymbol{\mu}_e; \boldsymbol{\mu}_{s_t,m}, \boldsymbol{\Sigma}_{s_t,m} + \boldsymbol{\Sigma}_e) \tag{4.42}$$

$$\propto \sum_{m=1}^{M} c_{s_t,m} \mathcal{N}(\boldsymbol{\mu}_e; \boldsymbol{\mu}_{s_t,m}, \boldsymbol{\Sigma}_{s_t,m} + \boldsymbol{\Sigma}_e) \tag{4.43}$$

The equivalent feature $\boldsymbol{\mu}_e$, the variance $\boldsymbol{\Sigma}_e$ and the new mixture weights $c'_{s_t,m}$ are given by following equations:

$$\boldsymbol{\Sigma}_e = \boldsymbol{\Sigma}_{\mathbf{x}_t|\mathbf{y}} (\boldsymbol{\Sigma}_{\mathbf{x}} - \boldsymbol{\Sigma}_{\mathbf{x}_t|\mathbf{y}})^{-1} \boldsymbol{\Sigma}_{\mathbf{x}} \tag{4.44}$$

$$\boldsymbol{\Sigma}_e^{-1} \boldsymbol{\mu}_e = \boldsymbol{\Sigma}_{\mathbf{x}_t|\mathbf{y}}^{-1} \boldsymbol{\mu}_{\mathbf{x}_t|\mathbf{y}} - \boldsymbol{\Sigma}_{\mathbf{x}}^{-1} \boldsymbol{\mu}_{\mathbf{x}} \tag{4.45}$$

$$c'_{s_t,m} = c_{s_t,m} \frac{\mathcal{N}(\mathbf{0}; \boldsymbol{\mu}_{\mathbf{x}_t|\mathbf{y}}, \boldsymbol{\Sigma}_{\mathbf{x}_t|\mathbf{y}})}{\mathcal{N}(\mathbf{0}; \boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{x}}) \mathcal{N}(\mathbf{0}; \boldsymbol{\mu}_e, \boldsymbol{\Sigma}_e)}. \tag{4.46}$$

Since the new mixture weights differ from the original ones by a multiplicative constant, the modified observation probability is proportional to expression (4.43). It is therefore not necessary to compute the new mixture weights.

Whereas the Eq. 4.44-4.46 are for the general case when the Gaussians have full covariance matrices, more intuitive expressions are obtained assuming diagonal covariance matrices. In this case, along one feature vector dimension the observation probability becomes:

$$\sum_{m=1}^{M} c_{s_t,m} \int_{x_t} \mathcal{N}(x_t; \mu_{s_t,m}, \sigma^2_{s_t,m}) \cdot \frac{\mathcal{N}(x_t; \mu_{x_t|\mathbf{y}}, \sigma^2_{x_t|\mathbf{y}})}{\mathcal{N}(x_t; \mu_x, \sigma^2_x)} dx_t$$

$$= \sum_{m=1}^{M} c'_{s_t,m} \mathcal{N}(\mu_e; \mu_{s_t,m}, \sigma^2_{s_t,m} + \sigma^2_e) \tag{4.47}$$

$$\propto \sum_{m=1}^{M} c_{s_t,m} \mathcal{N}(\mu_e; \mu_{s_t,m}, \sigma^2_{s_t,m} + \sigma^2_e) \tag{4.48}$$

where the equivalent mean, variance and weight are given by the following equations:

$$\frac{1}{\sigma^2_e} = \frac{1}{\sigma^2_{x_t|\mathbf{y}}} - \frac{1}{\sigma^2_x} \tag{4.49}$$

$$\frac{\mu_e}{\sigma^2_e} = \frac{\mu_{x_t|\mathbf{y}}}{\sigma^2_{x_t|\mathbf{y}}} - \frac{\mu_x}{\sigma^2_x} \tag{4.50}$$

$$c'_{s_t,m} = c_{s_t,m} \frac{N(0; \mu_{x_t|\mathbf{y}}, \sigma^2_{x_t|\mathbf{y}})}{N(0; \mu_x, \sigma^2_x) N(0; \mu_e, \sigma^2_e)}. \tag{4.51}$$

Comparing (4.15) and (4.43) it can be noted that both increase the variances of the original mixtures and evaluate the resulting PDF of an equivalent feature. However, the way in which the equivalent feature is computed and the amount added to variances is different. In the former, the equivalent feature $\boldsymbol{\mu}_e$ is the mean value of the Gaussian posterior. In the latter, it depends also on the prior parameters $\mu_{\mathbf{x}}$ and $\Sigma_{\mathbf{x}}$. Similarly, the equivalent variance $\boldsymbol{\Sigma}_e$ encompasses the effect of the prior variance.

The evaluation of (4.49) and (4.50) has to be performed only for unreliable features. The case $\sigma^2_{x_t|\mathbf{y}} = 0$ is excluded since this correspond to a fully reliable feature for which the posterior $p(\mathbf{x_t}|\mathbf{y})$ is equal to $\delta(\mathbf{x_t} - \mathbf{y_t})$. Therefore the evaluation of expressions (4.37) and (4.38) simplifies to evaluation of $p(\mathbf{x_t}|s_t)$ at $\mathbf{x_t} = \mathbf{y_t}$.

However, some numerical problems may appear in the case of a completely unreliable feature when $\sigma^2_{x_t|\mathbf{y}} \simeq \sigma^2_x$. This results in a large, potentially infinite

variance σ_e^2. To avoid such a situation (4.49) is evaluated only if the ratio between the variance of the posterior density and that of the prior density is smaller than one minus a small threshold. If this is not the case, i.e. the variances are very close to each other, we also force the means to be equal ($\mu_{x_t|\mathbf{y}} = \mu_x$) and thus the denominator in (4.43) or (4.48) becomes equal to the posterior. This results in marginalization since the observation probability of the uncorrupted feature is integrated over the feature space yielding unity. The approximation that we have made here can be seen to be based on information theory: the entropy of the random variable $\mathbf{x}_t$ is greater than that of $x_t|\mathbf{y}$ and the equality occurs when the observations are uninformative and thus the two PDF's are equal, having implicitly the same mean. Note that, practically, the prior and the posterior are estimated under simplifying assumptions such as being Gaussians. Due to this reason it may occasionally occur that the posterior variance is greater than the prior variance. However, as long as we know that this is only an artifact, we upper bound the variance $\sigma^2_{x_t|\mathbf{y}}$ to the a priori variance so that the validity of (4.49)-(4.50) is ensured. Similar consideration has to be made for (4.44)-(4.45).

The computation of (4.48) is certainly more time consuming than the evaluation of the original observation probability (4.39). Section 9.3 of this work shows how (4.39) is usually evaluated in a HMM-based recognizer with diagonal covariance mixture densities and what slowdown occurs due to (4.48). Section 9.4 of this work evaluates the impact of uncertainty on the acoustic search space and implicitly on the recognition speed.

Chapter 5

Data reliability estimation

Applying the uncertainty decoding of the previous chapter in a remote recognition setup requires knowledge of the statistical dependency between the received and sent feature. This can be modeled by the channel transition probability $p(\mathbf{y}_t|\mathbf{x}_t)$. Since we assume a digital channel connecting the client and server, the speech features are compressed prior to transmission, i.e. quantized into a bit pattern. The instantaneous bit error probability (reliability) of each decoded bit can be used to determine the channel transition probability of the bit pattern. This chapter presents approaches to obtain data reliability at the bit level which is subsequently used to infer the feature reliability in a remote speech recognition setup. The two possible types of network that we are studying are a circuit-switched transmission in which each bit may be independently corrupted, and a packet-switched transmission where the smallest unit which may be corrupted is a data packet. In both cases, the causes of the channel distortion, the channel models employed for the performance evaluation of EC, and the estimation of the instantaneous bit error probability are described.

5.1 Data reliability estimation in mobile networks

A typical example of a circuit-switched channel is the data channel of a GSM network, e.g. GSM-TCH/F4.8 [16]. For data transmission over GSM the user data is channel coded and modulated resulting in a signal which is then transmitted over the air interface in the form of a GSM burst. At the receiving end the signal consists typically of a sum of channel noise and delayed, attenuated replicas of the original signal, as a consequence of various reflections. Demodulation of the received signal followed by channel decoding results in a sequence of received

bits.

5.1.1 Transmission errors in mobile networks

This section briefly summarizes the causes of transmission errors in mobile networks. They serve as a basis for the channel models employed in the experimental section of this work to simulate data transmission in an RSR system.

In a wireless communication medium it is very common for noise to be additively combined with the transmitted signal. The signal-to-noise ratio (SNR), defined as the power of the signal divided by the power of the noise, is the typical measure of this distortion. As the distance between the sender and receiver increases, the signal strength is attenuated whereas the background noise level remains approximately constant. The SNR decreases correspondingly. This degradation is referred to as *path loss* and results typically in randomly distributed bit errors.

Another degradation factor of the radio channel is *fading*. Due to reflections from various objects, the transmitted signal may arrive at the receiver through multiple paths. Hence, the incoming signals summed in the receiver antenna have different amplitudes and phases which can either have a constructive contribution to the sum, or a destructive one, in the case of opposite phases. Since the receiver and other reflecting objects may be moving, the received signal exhibits rapid fluctuations of the envelope, an effect known as “Rayleigh” or “fast fading”. The temporary gaps of the envelope strength have very low SNR and cause bit errors concentrated in bursts. The bursty errors are more difficult to eliminate than randomly distributed errors. This is because commonly used error protection techniques such as FEC or channel coding, have specific correction capabilities that are easily exceeded if the errors are concentrated.

A particularly important type of additive distortions is caused by signals from other radio channels, known as *interference*. This may be an effect of intermodulation in the receiver front end circuitry, or as result of sharing the same communication medium by multiple users, as in networks with CDMA access. A measure of the interference from adjacent channels is the carrier-to-interference (C/I) ratio.

In GSM networks the most important degrading factors are interference and fast fading. The recommendations [78] present some typical scenarios of GSM channels which have to be taken into account at the network planning phase. According to this, the C/I should usually be about 10 dB when the mobile terminal

is located well inside a cell, 7 dB at the cell boundary and less than 4 dB outside the cell. The recommendations also specify some topological profiles, e.g. typical urban, typical rural etc. with a predefined number of propagation paths and delay spreads.

5.1.2 Channel Models

A very flexible and accurate way to model wireless transmission is to simulate the network physical layer. Such a simulation includes the (de)modulation, channel (de)coding, (de)interleaving and the channel model encompassing multipath propagation, fast fading, and co-channel interference. Physical layer simulation is computationally expensive and needs software modules which implement the above mentioned processing steps. In the experimental part of this work we employed the GSM library of the "Signal Processing Worksystem" (SPW) software suite [79] to simulate a complete transmission over the GSM-TCH/F4.8 data channel. The software allows for flexibly setting the channel model parameters such as C/I, receiver input SNR, GSM profile, terminal speed etc.

A much simpler and widely used simulation method is error patterns injection. This is more convenient in situations where the simulation of a set of a few particular conditions is needed. The representative error pattern $(e_1, \ldots, e_T)$ for each particular channel condition has to be prepared in advance, either by physical layer simulation, or by actual measurements. This can be done by comparing each bit of the transmitted bit stream b_t with the corresponding bit of the received bit stream $\hat{b}_t$ by an exclusive OR (XOR) operation: $e_t = b_t \oplus \hat{b}_t$. The length T must be large enough so that the pattern is statistically representative. Modeling the erroneous transmission by error patterns injection consists in computing the "received" data $\hat{b}_t$ by an XOR operation: $\hat{b}_t = b_t \oplus e_t$.

In comparison with physical layer simulation, error patterns injection is less flexible but easier to implement, provided that the error patterns are available.

Pearce [10] used GSM error patterns originally created for evaluation of speech quality under adverse channel conditions and adapted for the ETSI-DSR data rate of 4.8 kb/s. The error patterns labeled EP0, EP1, EP2, EP3 represent an error-free channel, good, medium and poor channel quality, respectively. They correspond to C/I ratios of ∞, 10, 7 and 4 dB and gross bit error rates (GBER) of 0%, 5%, 8% and 13%. The GSM error patterns became the de facto standard for evaluating DSR channel robustness, being widely used in works on this topic.

Another alternative for modeling the error characteristics of a wireless channel is the *Gilbert-Eliot* model [80, 81, 82]. It consist of a Markov chain with two states. The one represents the temporarily *Good* and the other the *Bad* channel condition, each one being assigned a constant bit error rate. In each state the bit errors occur independently of each other but with a higher rate in the *Bad* state. The state transition probabilities of the chain determine the frequency of bursts and their average length. A similar approach was applied in [46], where the GSM channel was simulated by superimposition of two additive white Gaussian noise (AWGN) channels with different noise levels: N_g - the background and N_b - the burst noise level, with $N_b >> N_g$ (see also Section 5.1.3). The resulting noise level was a weighted sum of N_g and N_b. By changing the weighting coefficients according to a Poisson distributed random variable, the model generates short periods of high noise levels, sporadically producing higher bit error rates.

The channel modeling methods of the previous paragraph are less complex than physical layer simulation and more flexible than the error patterns. However, they have the disadvantage that their parameters depend on each particular network condition. They can be directly related neither to channel parameters such as C/I, number of propagation paths, terminal speed, etc. nor to a specific channel coding scheme. The common practice is to tune them experimentally so that the resulting bit error distribution approaches that of the true channel.

Note, however, that the soft-output information of the channel decoder needed to estimate the feature posterior in this work can be obtained by physical layer simulation but it is not available if GSM error patterns are employed.

5.1.3 Instantaneous bit error probability estimation

This section presents two methods for obtaining the instantaneous bit error probability, which is the probability that the bit is decoded with an error. The first method exploits the soft-output of the channel decoding stage, i.e. the log-likelihood ratio of each bit. In the absence of soft-output information, we propose an alternative method based on CRC and data consistency checking. Both methods are described in the following paragraphs.

5.1.3.1 Estimation based on the soft-output from a channel decoder

In the receiving device, obtaining the sequence of transmitted bits from the real or complex valued output sample of the matched filter can be seen as a classification

problem. For simplicity let us consider uncoded binary phase shift keying (BPSK) transmission over an additive white Gaussian noise (AWGN) channel with constant power spectral density $N_0/2$. Each real valued output sample q_t carries in this case one bit b_t. Assuming maximum likelihood (ML) decoding, it is well-known that the average bit error probability is given by:

$$p_e = \frac{1}{2} erfc \sqrt{\frac{E_b}{N_0}} \tag{5.1}$$

where the E_b is the energy per transmitted bit and $erfc()$ denotes the Gaussian error function.

While (5.1) is an average, the instantaneous bit error probability is the probability of a specific bit $\hat{b}_t$ being erroneously decoded. This depends on the value q_t which corresponds to that bit and can be readily computed as:

$$p_{et} = \frac{1}{1 + exp(|\frac{4E_b}{N_0} q_t|)}. \tag{5.2}$$

The combination of the decoded bit $\hat{b}_t$ and its error probability p_{et} has been termed *softbit* in the literature [39].

Haykin [83] found that an expression similar to (5.2) approximates well to the instantaneous bit error probability for the Gaussian minimum shift keying (GMSK) modulation technique used in GSM.

The ratio E_b/N_0 which appears in (5.2) is related to the signal-to-noise power ratio (SNR). This has to be known or estimated in order to obtain p_{et} using the following equation:

$$SNR = \frac{E_b}{N_0} \frac{R_s}{B}. \tag{5.3}$$

Here R_s is the bitrate and B the bandwidth.

In the presence of channel coding, the so-called "soft-output" channel decoders, e.g. the Bahl decoder [84] and the less complex soft-output Viterbi algorithm [23], can be used to provide the instantaneous bit error probability (5.5). They are able to deliver the decoded bit stream and the log-likelihood ratio of each bit, the latter being defined as:

$$L_t = \log \frac{P(b_t = 1|\mathbf{q})}{P(b_t = 0|\mathbf{q})} , \tag{5.4}$$

The bold notation $\mathbf{q}$ denotes that a sequence of matched filter output samples may be used instead of one sample, as the channel coder may spread the information of one bit over more samples. With the soft-output (5.4) the instantaneous bit error probability is given by:

$$p_{et} = \frac{1}{1 + \exp|L_t|} . \tag{5.5}$$

5.1.3.2 Estimation based on CRC and data consistency

In wireless communication, a likely architecture is that the GSM base station (or an equivalent platform) performs the soft-output channel decoding and sends the decoded bits over another, e.g. wired reliable network, to the DSR server. However, transmission of bit reliability information would increase the required bandwidth and render this approach unattractive. Hence, there arises the need to estimate the instantaneous bit error probability p_{et} of the decoded bit $\hat{b}_t$ at the recognition server side in the absence of soft-output from the channel decoder.

The idea is to estimate the average bit error rate over a sequence of bits, e.g. the bits of a DSR frame or of a subvector within the frame, as the ratio of the number of (assumed) bit errors $\hat{N}_e$ to the total number of bits N within the sequence. Assuming that the instantaneous bit error rate p_{et} is constant for all bits of that sequence, an estimator of p_{et} is:

$$\hat{p_{et}} = \frac{\hat{N}_e}{N}. \tag{5.6}$$

One option is to employ the CRC check of ETSI-AFE to detect the bit errors within a frame of 92 bits, and estimate the average bit error rate on this interval. A problem is, however, that although the CRC check detects the errors with a high level of confidence, it cannot provide information about their number $\hat{N}_e$. Another problem is that the bit sequence on which the estimation is performed is relatively long (corresponds to 20 ms). During this relatively long time interval the assumption that the instantaneous bit error probability is constant may not hold.

A more accurate estimate $\hat{N}_e$ of the number of bit errors can be obtained by the data consistency test of ETSI-DSR, see Section 1.2.2.4. Basically, this checks

the continuity of a parameter within the frame pair. If the difference between two consecutive values of a parameter exceeds a fixed threshold it is decided that the frame pair is not reliable since some bit errors occurred. Certainly, not every bit error can be caught by this test, but only those that lead to exceeding the thresholds. Thus, there is a high probability of underestimating $\hat{p_e}_t$. In spite of this, our experiments have shown [85] that better results are obtained, compared to the case when the cyclic redundancy check (CRC) alone is employed.

The length of the sequence, over which N_e is estimated has to be chosen according to conflicting requirements. On one hand, a long interval is preferable, because this will deliver more reliable estimates $\hat{N}_e$. The estimation error decreases with the length of observation interval. On the other hand, a small interval is desirable since it is unrealistic to assume that $\hat{p}_e$ is constant over longer periods of time.

Two obvious choices were used. In the first one, referred as *frame oriented* (**FR**), the errors were counted over the whole frame-pair of 86 bits (excluding the 4 bits for CRC code) by applying the data consistency test to each parameter. For each consistency test failure the error counter for that frame is increased by one. Note that in this case the same value of $\hat{p_e}$ is shared by all 86 bits of the frame-pair. In the second procedure the bits of one subvector constitute the estimation sequence - *subvector oriented* (**SV**). The consistency test is triggered by a CRC failure, and the errors are counted on a subvector basis. We assumed that a consistency test failure reveals one bit error. Note that a subvector encodes two components of the feature vector. Thus, $\hat{p}_e$ for all bits of the subvector is estimated as the number of bit errors occurring in that subvector divided by the number of bits allocated to that subvector over the frame-pair, which is $2M$, see Table 1.2.

In [85] it has been shown that both methods **FR** and **SV** achieve a performance in a DSR scenario using GSM data channel which comes close to the performance using the instantaneous bit error probability computed from the soft-output of the channel decoder. In the experimental part of this work we compare EC performance of DSR employing the “true” soft-output of the channel decoder and the proposed estimation approach based on data consistency (**SV**).

5.2 Data reliability estimation in IP networks

In a packet-switched network the user data is fragmented into blocks named “data packets” which are then routed from source to destination. A prominent exam-

ple of a packet-switched network is the public Internet. Due to numerous advantages over the circuit-switched solution, the packet-switched solution has witnessed rapid expansion over the last decades. Its main benefit is the ability to share network resources among the users. Unlike circuit-switched networks, where a physical channel is exclusively used for connecting the source and destination points, packet-switched networks allow the use of the same channel to transport data from/to several users. Furthermore, there may exist multiple routing paths between two connection points so that the bandwidth resources can be more efficiently managed. An inherent property arising from this architecture is that the network cannot ensure whether a packet is going to arrive at destination or how long this process will take. Hence, the network offers so called *best effort* delivery service.

The packets that do not arrive at the destination in a reasonable time are considered lost. The maximum delay which can be allowed depends on the real time constraints of the application. For example in VoIP the latency should not exceed 200 ms, otherwise the conversation flow is impaired.

5.2.1 TCP/IP protocol suite

TCP/IP denotes a family of protocols developed by the Internet Engineering Task Force (IETF) aiming at providing a simple and flexible framework for developing network applications and services. The protocols are organized into a stack of four layers, each layer being implemented on top of the other and providing specific services. The layers at the top are closer to the application while those near the bottom are closely related to the physical channel.

1. *Application layer* provides methods to pass the data from the program to the transport layer in an application-specific format. E.g. HTTP, FTP.

2. *Transport layer* is responsible for transferring data from source to destination in an abstract manner, independently of the underlying network. Applications are connected through the use of ports. The transmission can be either connection-oriented (TCP) or connectionless (UDP). TCP provides a reliable link which guarantees that packets arrive in order with minimal error. This is ensured mainly by retransmission of lost or discarded packets. UDP does not guarantee a reliable transmission. The lost packets are not retransmitted and the erroneous one, detected by a checksum algorithm,

are discarded. UDP is typically used in data streaming applications such as VoIP where timely delivery is more important than reliability.

3. *Network layer* provides mechanisms to transmit the packets of the transport layer from source to destination independently of the data link layer. This is specified by the IP protocol. The whole network is virtualized by associating IP addresses with the source and destination points.

4. *Data link layer* is network dependent. There are a large variety of networks (including mobile networks) in which TCP/IP can be deployed. Examples of protocols on this layer are the PPP protocol for internet access over a dial-up modem, IEEE 802.11 for a local wired network, GPRS for packet data over GSM, etc.

For the particular case of data streaming application, such as VoIP, where on-time delivery of the packets is crucial, the IETF developed the Real Time Transfer Protocol (RTP) [86]. The protocol is built on top of UDP and provides methods to reorder the incoming packets, handle the time delay spread and synchronization of multiple data streams transmitted over separate channels. Like UDP, it is a best effort data delivery service since the data reliability cannot be guaranteed.

The data packet in an RTP communication consists of IP, UDP and RTP headers, and the real-time user data. The IP header (in IPv4) contains 20 bytes of protocol specific data protected by a checksum. If the checksum of the header bits at reception does not match the checksum field, the packet is dropped. The UDP comprises 8 bytes and contains the source and destination ports, the length of the header and encapsulated data, and the checksum of the whole packet (including encapsulated data). In IPv4 the packets containing errors are discarded. IPv6 may also accept erroneous UDP packets. This feature was introduced in the new version of IP in order to support those applications which are more sensitive to packet loss than to bit errors in payload.

The RTP header has 12 bytes of protocol specific data including the payload type and a sequence number. The payload type identifies the kind of encapsulated data (e.g. the speech codec used) so that they can be properly decoded at the reception. The sequence number is used to reorder the sequence of packets in the jitter buffer at the reception, since their trip delay may not be constant. The size of the jitter buffer is dictated by the maximum latency tolerated by the application. The packets that do not arrive within that time span are considered to be lost packets.

5.2.2 Transmission errors in IP networks

In IP networks there are two factors responsible for transmission errors. One is the error prone communication medium and the other arises from the specific network architecture. Both are explained as follows:

The data link layer is error prone due to the inherent noise in the communication medium. If the data link layer consists of a wireless or mobile network, the same distorting factors, e.g. path loss, fading, as discussed in the previous section, apply. If the data link layer is a wired network, such as Ethernet, the SNR level in the communication medium is usually high so that the transmission approaches the noiseless scenario. The result of such channel distortion is that the data link layer of the receiver may pass erroneous packets to the upper layer. In the latter, however, the checksum of the data is checked and the erroneous packets are discarded (in IPv4), that is, the application layer does not receive erroneous packets at all. Therefore, in IP networks the bit errors occurring in the data link layer usually turn into packet loss at the application layer level.

The other degradation is related to the IP network architecture. The users are connected to the network by means of *router* devices. A router has a number of input and output *queues* connected by a *switcher*. The packets received from the user are buffered first in an input queue and the switcher decides, according to a *switching logic* algorithm, the route that the packet has to follow, i.e. to which output queue the packet is to be redirected to. The packets in the output queues are stored ready to be sent; however, transmission does not occur immediately. The packets may wait an undefined time until they are sent, depending on network load, packet and queue size, routing policy, etc. Thus, they are likely to be randomly delayed. This is known as *time delay spread*. Besides variable latency at reception, packet-switched architecture may be confronted with two critical situations in practice [87]: either some input queue may overflow due to an input flow higher than the processing capacity of the switcher, or, some output queue may overflow when the switcher delivers packets at a higher rate than the transmission medium can manage. Overflows lead immediately to the dropping of packets from queues in order to ensure the continuation of data flow. Since dropped packets never reach their destination, this effect is known as *packet loss*. Furthermore, it is likely that once an overflow has occurred, it may last some time until the queue is in a free-flowing steady state. Dropping successive packets leads to bursty packet loss [88, 27].

Further loss of packets may occur at reception in the application layer if real-

time constraints apply. The media streaming applications, such as VoIP, expect data to arrive within a specific time span. This is necessary in order to keep the latency within reasonable limits by avoiding the accumulation of packet delays. At the receiving end the incoming packets are buffered in the *jitter buffer* before proceeding with decoding. The aim of buffering is twofold. First, it enables delayed packets to be stored in an orderly fashion. The playback starts when the buffer is full, that is, after a time proportional to the buffer size has elapsed. Secondly, since the packets usually do not arrive in order, they can be reordered in the jitter buffer by mean of the sequence number in the RTP header. As the playback must be performed synchronously, the frame which has to be decoded next is expected to be already in the jitter buffer. If not present, the application has to consider it lost (since it is useless to decode it later) and needs to employ a specific packet loss concealment (PLC) technique in order to reconstruct the speech signal of the lost segment.

Although with IPv6 or UDP-Lite the propagation of bit errors towards the application layer is not excluded, to date it has been widely assumed that the main source of degradation in IP networks is packet loss [21, 20]. The next section describes channel models that can be employed to simulate packet loss in a remote speech recognition scenario over an IP network.

5.2.3 Channel Models

As mentioned in Section 5.1.2 the burstiness of channel errors can be conveniently reproduced by a Markov chain [80, 81]. The same principle was applied in [27, 89, 90, 20] to model packet loss in an IP network. The simplest and mostly used model consists of a 2-states Markov model, similar to the Gilbert model [80], schematized in Figure 5.1.

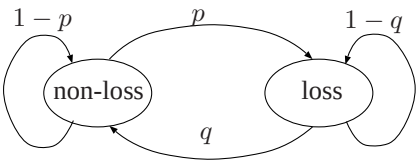

Figure 5.1: 2-state Markov model for packet loss in IP networks

Each packet triggers a transition from the current to the next state according to the transition probabilities of the chain. The packet is correctly received if the cur-

rent state is *non-loss* otherwise it is declared lost. This model is completely defined by the two parameters p and q representing the transition probabilities between the states. These can be used to derive two other more intuitive parameters. One is the *mean loss probability* (mlp) which is the average probability of packet loss and is given by $mlp = p/(p+q)$. The other is the *conditional loss probability* (clp) defined as the probability of packet loss given that the previous packet was lost $clp = 1 - q$. The values of model parameters can be chosen so that the generated loss pattern approaches that corresponding to a certain network load, packet size, etc. They can be obtained by estimating mlp and clp from real packet traces characterizing that network scenario. For simulation purposes some authors including [27, 91, 47] extensively used the settings given in Table 7.3 which model realistic situations of low losses and short loss bursts (C1,C2) up to high losses and longer bursts (C3,C4). These conditions were also used in the experimental part of this work.

In addition to the two-state Markov chain, other models of higher complexity have been proposed. Milner and James [89] proposed adding a third state which models randomly received packets inside the loss periods, a situation which may occur by freeing the router queues for a very short time period. The model proposed by ETSI [92] for VoIP Quality of Service evaluation adds a fourth state which models the isolated (not bursty) packet losses. The correlation between losses can be even more accurately represented by employing higher order Markov models as shown in [93], however, the disadvantage is the higher number of parameters that have to be set.

While an accurate modeling of packet loss is desired, the purpose of this work is, however, not to deliver absolute values of the word accuracy for a particular network scenario, but to allow performance comparison with other state-of-the-art approaches under the same network conditions. From this point of view the two-state Markov chain is a convenient choice.

Alternatively, if the absolute WER of a recognition task needs to by predicted for a particular network condition, e.g. a technique similar to VoIP Quality of Service evaluation applicable to DSR, the burst length distribution of that particular condition has to be known. The WER can be estimated as a weighted sum of word error rates achieved by the task under random losses of fixed length. They can be estimated in advance as shown in [94].

5.2.4 Instantaneous bit error probability estimation

In packet-loss networks, the term "bit error probability" may at first seem inappropriate. The bits within a lost packet do not exist in practice since they have not actually been received. Even though the bits are lost, we can assume (without loosing generality) that they are generated at random in the application layer. This can be equivalent to a situation of severe channel degradation in which the channel decoder generates uninformative bits.

Assuming that the sent bits are uniformly distributed, by generating zero or one at random in the application layer with probabilities a and $1-a$, respectively, the probability of error becomes:

$$\begin{aligned} p_{e_t} &= P(\hat{b}_t = 0|b_t = 1) \cdot P(b_t = 1) + \\ &\quad P(\hat{b}_t = 1|b_t = 0) \cdot P(b_t = 0) \\ &= a \cdot \frac{1}{2} + (1-a) \cdot \frac{1}{2} = \frac{1}{2}, \end{aligned} \tag{5.7}$$

independent of the value of a.

Note the bits need not be generated in fact. If the bit error probability is $1/2$, their actual value does not matter. As it will become more clear in Chapter 6, the aim of estimating instantaneous bit error probability is to compute the channel transition probability $p(\mathbf{y}_t|\mathbf{x}_t)$. By randomly generating the lost bits independently of the sent ones, it is in fact ensured that the "received" feature $\mathbf{y}_t$ and the sent one $\mathbf{x}_t$ are statistically independent. Thus, $p(\mathbf{y}_t|\mathbf{x}_t)$ equals $p(\mathbf{y}_t)$ which does not depend on $\mathbf{x}_t$, and therefore does not change the feature posterior.

In conclusion, the bit error probabilities of the correctly received bits are zero denoting that the bits are reliable, while those of the lost bits are set to $1/2$ according to (5.7). Note that in general $p_{e_t} = \frac{1}{2}$ denotes the maximum level of degree of unreliability. Higher values are not possible assuming that the decoder uses MAP decision.

5.3 Discussion

Working with instantaneous bit error probability in both bit error and packet loss cases, allows for a unified error concealment approach [85] in that the details of the underlying network do not need to be known. The feature posterior computation

presented in Chapter 6 can be applied in both circuit and packet-switched networks if the received bits $\hat{b}_t$ and their error probabilities p_{et} are available.

Moreover, the next generation of IP networks (IPv6) is supposed to support applications such as VoIP better, in that the receiving of partially damaged payload is preferable to discarding it in the transport layer.

For example the UDP-Lite [95] transport protocol allows packets with failing checksums to reach the application. This enables the application to decide itself how to handle the erroneous packets. This feature has also already been proven to be advantageous for DSR applications. Here, the nearest frame repetition method is more efficient on a subvector basis, i.e. reconstructing only the erroneous part of the vector, than on frame basis where the whole frame is discarded possibly due to a single bit error [42].

Chapter 6

Feature posterior estimation

This Chapter describes our approach to estimate the feature posterior in DSR systems, where compressed features are transmitted over an error-prone channel. The features can be extracted and compressed at the terminal side using any of the ETSI-DSR standards [8, 12, 14]. For simplicity, however, our approach is explained in detail here for the ETSI Advanced Front-end (ETSI-AFE) [12] which has been described in Section 1.2.2. The short-hand notation $\mathbf{v}_t$ denotes in the following any of the seven subvectors of the 14-dimensional real valued vector produced by the feature extraction part, see Table 1.2.

Figure 6.1 depicts the DSR system considered in this section. The *feature extraction* block, *Quantizer*, *Index generator* and *Codebook* are those of the ETSI-AFE. Note that only one subvector $\mathbf{v}_t$ is being considered since, as long as it is not otherwise specified, the operations are performed identically for any subvector.

The subvector $\mathbf{v}_t$ is quantized ($\mathbf{c}_t$) and coded into a bit pattern $\mathbf{b}_t$ of M bits. This constitutes the source coding part of the system. The bit pattern $\mathbf{b}_t$ is then transmitted through an equivalent channel which includes the channel (de)coding and the other components. The channel output consists of the decoded bit pattern $\hat{\mathbf{b}_t}$ and the instantaneous bit error probabilities of each of its M bits. The dependency between received bit, transmitted bit, and channel state $\mathbf{z}_t$ at that time is modeled statistically by the transition probability $P(\hat{\mathbf{b}}_t|\mathbf{b}_t, \mathbf{z}_t)$. The latter variable $\mathbf{z}_t$ has been introduced to allow us to model the time varying channel properties (state) using instantaneous bit error probability computed as described in Chapter 5.

The channel state gives in fact the probability of a bit error at that time. The posterior probability of the bit pattern is then computed using the redundancy of the bitstream (*a priori* knowledge) and the transition probability. Once the bit

pattern posterior is known, the static feature posterior can easily be inferred. The above processing steps are described in the following.

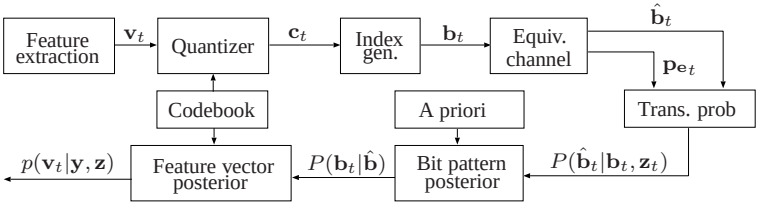

Figure 6.1: Block diagram of processing elements relevant for feature posterior estimation in a DSR system.

6.1 Source coding

Generally, a source coder is a mapping of the N-dimensional Euclidian space into a finite set of 2^M indices. It consists of two components: the quantizer and the index generator. The quantizer searches the finite codebook for that N-dimensional codeword (centroid) $\mathbf{c}$ which best represents the N-dimensional input vector. The search criterion is usually the Euclidian distance measure. The codeword is then mapped to an index $\mathbf{b}$ of length M bits.

The source coder of ETSI-AFE employs a split vector quantizer (SVQ) for the quantization of the static components of feature. The input is the 14-dimensional vector consisting of 13 MFCCs and the logarithmic frame energy. This is split into 7 two-dimensional subvectors ($N = 2$) which are then separately quantized to bit patterns of length M according to Table 1.2.

6.2 Equivalent channel

Both circuit and packet-switched channels are modeled here by an equivalent time-variant binary symmetric channel. The binary input symbols are corrupted with probability p_{e_t} which may vary with each transmitted bit. The input of the channel at each time instance t is the bit pattern produced by source coding corresponding to the centroid $\mathbf{c}_t$. The bit patterns $\mathbf{b}_t$ are sent in a sequence $b_t(0), \ldots, b_t(M-1)$, M being the number of bits allocated for the subvector. Note that in Chapter 5 this

detail was intentionally omitted for clarity since there the index t was associated with each bit of the sequence. The channel output consists of the decoded bit pattern $\hat{\mathbf{b}}_t = \hat{b}_t(0), \ldots, \hat{b}_t(M-1)$ and the vector of bit error probabilities $\mathbf{p}_{e\,t} = p_{e\,t}(0), \ldots, p_{e\,t}(M-1)$ of each decoded bit.

6.3 Channel transition probability

The channel transition probability is defined here as the probability of receiving a particular pattern given the sent pattern and the present channel state. This can be obtained considering the Bayesian network of the Figure 6.2. Here we use the capitalized notation $P(\hat{\mathbf{b}}_t|\mathbf{b}_t, \mathbf{z}_t)$ since the bit patterns can be seen as discrete random variables.

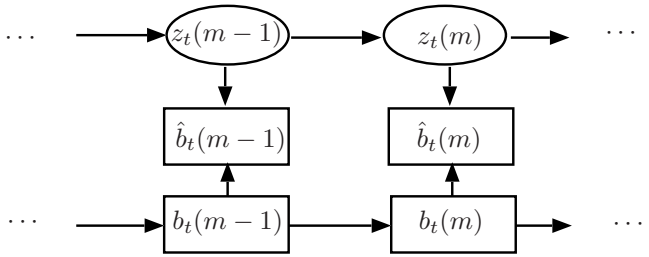

Figure 6.2: Bayesian Network for computing channel transition probability of the bit pattern $(b_t(0), \cdots b_t(M-1))$

The model captures both, possible dependencies between the bits of $\mathbf{b}_t$, which corresponds to a source with memory, and dependencies between consecutive channel states, which correspond to a channel with memory such as a bursty error channel. However, the received bit $\hat{b}_t(m)$ depends only on the sent bit $b_t(m)$ and $z_t(m)$ at that time instance. This allows us to express the transition probability of each bit as:

$$P(\hat{b}_t(m)|b_t(m), z_t(m)) = \begin{cases} 1 - p_{e\,t}(m) & \text{if } \hat{b}_t(m) = b_t(m) \\ p_{e\,t}(m) & \text{if } \hat{b}_t(m) \neq b_t(m). \end{cases} \tag{6.1}$$

The transition probability of the bit pattern becomes:

$$P(\hat{\mathbf{b}}_t|\mathbf{b}_t, \mathbf{z}_t) = \prod_{m=0}^{M-1} P(\hat{b}_t(m)|b_t(m), z_t(m)) \tag{6.2}$$

For the purpose of further processing it is enough to evaluate the above expression only at the particular received pattern $\hat{\mathbf{b}}_t$ but for all possible values of $\mathbf{b}_t$, i.e. $0, \ldots, 2^M - 1$. Thus, the output of the block *Transition probabilities* in the Figure 6.1 is a vector of 2^M probability values $P(\hat{\mathbf{b}}_t|\mathbf{b}_t^{(i)}, \mathbf{z}_t)$, with $i = 0 \ldots 2^M - 1$.

It is instructive to analyse the expression (6.2) for a packet erasure channel. In this case, a data packet which carries a number of bit patterns is either completely lost or received without any bit errors. Thus, the channel state $z_t(m)$ is binary, the possible values corresponding to two situations: bit m belongs to a received ($z_t(m) = 0$) or to a lost packet ($z_t(m) = 1$). As shown in the previous chapter, in the packet loss scenario the bit error probability can take only two values: 0 and $1/2$. Assuming that all bits of a bit pattern are carried in the same data packet, i.e. $z_t(m) = z_t$, for all $m = 0 \ldots M - 1$, the transition probability becomes now:

$$P(\hat{\mathbf{b}}_t|\mathbf{b}_t, \mathbf{z}_t) = \begin{cases} 1 & \text{if } \mathbf{z}_t = 0 \text{ and } \hat{\mathbf{b}}_t = \mathbf{b}_t \\ 0 & \text{if } \mathbf{z}_t = 0 \text{ and } \hat{\mathbf{b}}_t \neq \mathbf{b}_t \\ (\frac{1}{2})^M & \text{if } \mathbf{z}_t = 1 \end{cases} \tag{6.3}$$

In the view of (6.3) it makes no difference how the channel decoder "reconstructs" the lost bits. The equation states that if the packet is lost, $P(\hat{\mathbf{b}}_t|\mathbf{b}_t, \mathbf{z}_t)$ is independent of $\mathbf{b}_t$.

6.4 Bit pattern posterior computation

In Chapter 4 we have seen that for the purpose of uncertainty decoding, the temporal correlation of the feature vectors can either be neglected (Section 4.3.1), or it can be taken into account (Section 4.3.2). The first case yields a memoryless discrete source of bit patterns whereas the second corresponds to a discrete Markov source. This section shows how the bit pattern posterior can be computed in each case using the bit reliability information and priori knowledge of the source.

6.4.1 A priori knowledge

The knowledge of source statistics is contained in the block *a priori* of Figure 6.1. This is modeled by the prior probability mass function of the bit pattern $P(\mathbf{b}_t^{(i)})$, with $i = 0 \dots 2^M - 1$, and the conditional probability mass function $P(\mathbf{b}_t^{(i)}|\mathbf{b}_{t-1}^{(j)})$, $i, j = 0 \dots 2^M - 1$ for the case of first-order Markov source. Both are estimated in advance on a training set by quantizing the feature vectors and counting the occurences $N^{(i)}$ of each bit pattern $\mathbf{b}_t^{(i)}$ and the occurrences of the sequences $(\mathbf{b}_t^{(i)}, \mathbf{b}_{t-1}^{(j)})$ denoted by $N^{(i,j)}$. The ML estimates (notation with $\hat{\cdot}$ for estimate was omitted for convenience) are:

$$P(\mathbf{b}_t^{(i)}) = \frac{N^{(i)}}{N} \tag{6.4}$$

$$P(\mathbf{b}_t^{(i)}, \mathbf{b}_{t-1}^{(j)}) = \frac{N^{(i,j)}}{N \cdot N} \tag{6.5}$$

$$P(\mathbf{b}_t^{(i)}|\mathbf{b}_{t-1}^{(j)}) = \frac{P(\mathbf{b}_t^{(i)}, \mathbf{b}_{t-1}^{(j)})}{P(\mathbf{b}_t^{(j)})}$$

Here the process stationarity was assumed such that the statistic does not change over time, i.e. $P(\mathbf{b}_{t-1}^{(j)}) = P(\mathbf{b}_t^{(j)})$.

The average information contained in a bit pattern can be measured by the source entropy:

$$H(\mathbf{b}_t) = -\sum_{i=0}^{2^M-1} P(\mathbf{b}_t^{(i)}) \log P(\mathbf{b}_t^{(i)}) \tag{6.6}$$

The first line of Table 6.1 gives the entropies of the bit patterns corresponding to individual subvectors. The values have been obtained on the training set of the Aurora 2 database using the ETSI advanced feature extraction front-end. It can be observed that the entopy values are close to M, the number of bits used at quantization. This denotes that the redundancy is small within a bit pattern.

An indication of inter-frame correlation is the *mutual information* of consecutive vectors defined as:

$$I(\mathbf{b}_t; \mathbf{b}_{t-1}) = H(\mathbf{b}_t) - H(\mathbf{b}_t|\mathbf{b}_{t-1}) \tag{6.7}$$

Table 6.1: *Entropies and mutual information among the subvectors produced by the ETSI advanced DSR front-end.*

Subvector	1	2	3	4	5	6	7
M	6	6	6	6	6	5	8
$H(\mathbf{b}_t)$	5.8	5.8	5.8	5.8	5.8	4.8	7.7
$I(\mathbf{b}_t;\mathbf{b}_{t-1})$	2.6	2.1	1.6	1.4	1.2	1.0	3.4
$I(\mathbf{b}_t;\underline{\mathbf{b}}_{t-1})$	3.0	2.4	1.9	1.7	1.5	1.3	4.5

where the conditional entropy is computed as:

$$H(\mathbf{b}_t|\mathbf{b}_{t-1}) = -\sum_{i=0}^{2^M-1}\sum_{j=0}^{2^M-1} P(\mathbf{b}_t^{(i)},\mathbf{b}_{t-1}^{(j)})\log P(\mathbf{b}_t^{(i)}|\mathbf{b}_{t-1}^{(j)}) \tag{6.8}$$

The mutual information $I(\mathbf{b}_t;\mathbf{b}_{t-1})$ is a measure of how much information about the current bit pattern $\mathbf{b}_t$ is already present in the previous vector $\mathbf{b}_{t-1}$. The higher the mutual information, the better a bit pattern can be predicted from its predecessor. Statistical independence (no correlation) of bit patterns at t and $t-1$ would lead to $H(\mathbf{b}_t|\mathbf{b}_{t-1}) = H(\mathbf{b}_t)$ and the mutual information would become zero.

The second line of Table 6.1 gives the mutual information computed for each subvector. Here can be observed that the values are relatively high, denoting that the assumption of a Markov source is more appropriate than the assumption of a memoryless source.

Even more accurate modeling of the memory source can be achieved by considering a higher-order Markov process. For example in the second-order Markov source, the current bit pattern $\mathbf{b}_t$ depends on two predecessors $\mathbf{b}_{t-1}, \mathbf{b}_{t-2}$. However, the complexity and memory requirements increase exponentially with the order of the Markov source. A good compromise can be made by still assuming first-order but extending the static feature vector by appending the dynamic features, e.g. first and second-order temporal differences. By doing so, the first predecessor of a feature vector contains to some extent information about the second, third, etc. predecessor.

The last line of Table 6.1 gives the mutual information between the current

bit pattern of the static components $\mathbf{b}_t$ and the bit pattern of the previous frame $\underline{\mathbf{b}}_{t-1} = (\mathbf{b}_{t-1}, \Delta\mathbf{b}_{t-1}, \Delta^2\mathbf{b}_{t-1})$, which consists of the coded static feature components $\mathbf{b}_{t-1}$ and the coded first- and second-order derivatives. While the static components were coded by the ETSI-AFE, for the dynamic components of each subvector we had to design our own vector quantizers [96]. For the experiment reported in Table 6.1 we employed $D_1 = 3$ bit vector quantizers for Δ (velocity) and $D_2 = 1$ bit for Δ^2 (acceleration). Obviously, the dynamic parameters of the previous frame provide additional knowledge about the static parameters of the current frame, since the mutual information is higher than the one observed between $\mathbf{b}_t$ and $\mathbf{b}_{t-1}$.

6.4.2 Memoryless source

Assuming a memoryless source, the correlation between $\mathbf{b}_t$ and $\mathbf{b}_{t-1}$, which represents most of the bitstream redundancy, is neglected. Without temporal dependencies, the bit pattern posterior $P(\mathbf{b}_t|\hat{\mathbf{b}}_t)$ can be easily computed by applying Bayesian rules for conditional probabilities:

$$\begin{aligned} P(\mathbf{b}_t^{(i)}|\hat{\mathbf{b}}_t) &= P(\mathbf{b}_t^{(i)}|\hat{\mathbf{b}}_t, \mathbf{z}_t) \\ &= \frac{P(\hat{\mathbf{b}}_t|\mathbf{b}_t^{(i)}, \mathbf{z}_t)P(\mathbf{b}_t^{(i)}|\mathbf{z}_t)}{P(\hat{\mathbf{b}}_t|\mathbf{z}_t)} \\ &= \frac{P(\hat{\mathbf{b}}_t|\mathbf{b}_t^{(i)}, \mathbf{z}_t)P(\mathbf{b}_t^{(i)})}{P(\hat{\mathbf{b}}_t|\mathbf{z}_t)} \\ &= \frac{1}{\mathcal{K}}P(\hat{\mathbf{b}}_t|\mathbf{b}_t^{(i)}, \mathbf{z}_t)P(\mathbf{b}_t^{(i)}), \\ &\quad i = 0 \ldots 2^M - 1 \end{aligned} \tag{6.9}$$

Here we theorized that the sent bit pattern and channel state are independent resulting in $P(\mathbf{b}_t^{(i)}|\mathbf{z}_t) = P(\mathbf{b}_t^{(i)})$. The denominator $P(\hat{\mathbf{b}}_t|\mathbf{z}_t)$ is assimilated into the constant $\mathcal{K}$ since it does not depend on i. It can be computed by forcing the posterior to sum to one:

$$\mathcal{K} = \sum_{i=0}^{2^M-1} P(\hat{\mathbf{b}}_t|\mathbf{b}_t^{(i)}, \mathbf{z}_t)P(\mathbf{b}_t^{(i)}) \tag{6.10}$$

The vector of probabilities $P(\hat{\mathbf{b}}_t|\mathbf{b}_t^{(i)}, \mathbf{z}_t)$, $i = 0 \dots 2^M - 1$ is provided by the block *transition probabilities*, see Figure 6.1.

6.4.3 Markov source

The feature posterior $p(\mathbf{x}_t|\mathbf{y}_1^T)$ which captures the temporal correlation between features can be computed from the bit pattern posterior $P(\mathbf{b}_t|\hat{\mathbf{b}}_1^T)$. In specifying the correlation a good compromise between accuracy and complexity is to assume that the sequence $\mathbf{b}_t$, $t = 1, 2, \dots$ is a homogeneous first-order Markov process. Let $\mathbf{b}_t^{(i)}$, $i = 0, \dots, 2^M - 1$ be the discrete state space of this process, i.e. the values that the bit pattern may take, and $a_{ij} = P(\mathbf{b}_t^{(j)}|\mathbf{b}_{t-1}^{(i)})$, $i, j = 0 \dots 2^M - 1$ are the elements of the transition probability matrix.

The bit pattern posterior $P(\mathbf{b}_t^{(i)}|\hat{\mathbf{b}}_1^T)$ conditioned on all received bit patterns $\hat{\mathbf{b}}_1^T = (\hat{\mathbf{b}}_1, \dots, \hat{\mathbf{b}}_T)$ can be computed by the Forward-Backward algorithm [51]. Let $\alpha_t(i)$ be the forward probabilities, $\beta_t(i)$ the backward probabilities and $\gamma_t(i)$ the posterior probabilities of each bit pattern $i = 0 \dots 2^M - 1$. They are defined as follows:

$$\alpha_t(i) = P(\hat{\mathbf{b}}_1^t, \mathbf{b}_t^{(i)}|\mathbf{z}_1^T) \tag{6.11}$$

$$\beta_t(i) = P(\hat{\mathbf{b}}_{t+1}^T|\mathbf{b}_t^{(i)}, \mathbf{z}_1^T) \tag{6.12}$$

$$\gamma_t(i) = P(\mathbf{b}_t^{(i)}|\hat{\mathbf{b}}_1^T) \tag{6.13}$$

The algorithm details are given below:

Initialization:

$$\alpha_1(i) = P(\mathbf{b}_1^{(i)})P(\hat{\mathbf{b}}_1|\mathbf{b}_1^{(i)}, \mathbf{z}_1) \tag{6.14}$$

$$\beta_T(i) = 1$$

Recursion:
(starting from $t = 1$ for forward and $t = T$ for backward probabilities)

$$\alpha_{t+1}(i) = \sum_{j=0}^{2^M-1} \alpha_t(j)P(\mathbf{b}_{t+1}^{(i)}|\mathbf{b}_t^{(j)}) \cdot P(\hat{\mathbf{b}}_{t+1}|\mathbf{b}_{t+1}^{(i)}, \mathbf{z}_{t+1}) \tag{6.15}$$

$$\beta_{t-1}(i) = \sum_{j=0}^{2^M-1} \beta_t(j)P(\mathbf{b}_t^{(j)}|\mathbf{b}_{t-1}^{(i)})P(\hat{\mathbf{b}}_t|\mathbf{b}_t^{(j)}, \mathbf{z}_t)$$

Posterior computation:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=0}^{2^M-1} \alpha_t(j)\beta_t(j)} \tag{6.16}$$

The implementation of recursion (6.15) needs special precautions since the multiplication of small probability values can quickly lead to numerical problems. In order to avoid this, the forward and backward probabilities were scaled, i.e. multiplied by a constant $\mathcal{K}_{\alpha_t}$ and $\mathcal{K}_{\beta_t}$, respectively, at each step t such that they summed to one. The scaling coefficients do not affect the final result since:

$$\gamma_t(i) = \frac{\mathcal{K}_{\alpha_t}\alpha_t(i)\mathcal{K}_{\beta_t}\beta_t(i)}{\sum_{j=0}^{2^M-1} \mathcal{K}_{\alpha_t}\alpha_t(j)\mathcal{K}_{\beta_t}\beta_t(j)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=0}^{2^M-1} \alpha_t(j)\beta_t(j)} \tag{6.17}$$

6.4.4 Extended Markov source

A typical feature vector for ASR consists of static mel frequency cepstral coefficients (MFCC) and their first- and second-order temporal differences, the so-called dynamic features. In [96] we have shown that the source modeling can be improved if we consider that the sequence $\underline{\mathbf{b}}_t = (\mathbf{b}_t, \Delta\mathbf{b}_t, \Delta^2\mathbf{b}_t)$,$t = 1, 2, \ldots$, i.e. coded static, delta and delta-delta components, is a Markov process. The bit pattern posterior can be determined in a similar fashion as above. The complete bit pattern has in this case $N = M + D_1 + D_2$ bits and the state space is extended to 2^N elements. D_1 and D_2 denote the number of bits for quantization of delta and delta-delta components. Since only the bit pattern corresponding to the static components is transmitted, the channel transition probability of the complete pattern is:

$$P(\hat{\mathbf{b}}_t|\underline{\mathbf{b}}_t^{(i,m,n)}, \mathbf{z}_t) = P(\hat{\mathbf{b}}_t|\mathbf{b}_t^{(i)}, \mathbf{z}_t), \tag{6.18}$$

for all

$$\begin{aligned} i &= 0, \ldots, 2^M - 1, \\ m &= 0, \ldots, 2^{D_1} - 1, \\ n &= 0, \ldots, 2^{D_2} - 1. \end{aligned}$$

Here $\underline{\mathbf{b}}_t^{(i,m,n)} = (\mathbf{b}_t^{(i)}, \Delta\mathbf{b}_t^{(m)}, \Delta^2\mathbf{b}_t^{(n)})$ is the bit pattern consisting of the i^{th} static, m^{th} velocity and n^{th} acceleration bit pattern.

The bit pattern posterior of the static components $P(\mathbf{b}_t^{(i)}|\hat{\mathbf{b}}_1^T)$ is obtained from the bit pattern posterior of the whole vector (computed by the forward-backward as previously shown) by marginalizing the dynamic components:

$$P(\mathbf{b}_t^{(i)}|\hat{\mathbf{b}}_1^T) = \sum_{m=0}^{2^{D_1}-1} \sum_{n=0}^{2^{D_2}-1} P(\mathbf{b}_t^{(i,m,n)}|\hat{\mathbf{b}}_1^T), \quad \text{for any } i = 0, \ldots, 2^M - 1. \tag{6.19}$$

Similarly to Eq. 6.19, the bit pattern posterior of the dynamic components can be obtained from the posterior of the full vector by marginalizing the static components. Experimental evaluation showed, however, that this method yields poor results due to the rough quantization for delta and delta-delta that was employed. Therefore, the posterior of dynamic components is computed using an alternative suboptimal approach presented in Section 6.5.2.

Note that there are technical limits to increasing the resolution of the dynamic features. The number of states to be evaluated by the Forward-Backward algorithm grows exponentially with the number of bits of the bit pattern. Even with the rough quantization of dynamic components used in our experiment, we already needed 12 bits (8 for static, 3 for delta and 1 for delta-delta) for the seventh subvector, resulting in 4096 model states. More details on computational complexity can be found in Chapter 9.

6.5 Feature posterior

The feature posterior is the probability density function of the sent feature $\mathbf{x}_t$ conditioned either on only the received feature $\mathbf{y}_t$ or on the whole sequence $\mathbf{y}_1^T$, see also Eq. 4.37 and 4.38. In speech recognition the feature vector $\mathbf{x}_t$ usually consists of static components, e.g. cepstral coefficients, and dynamic components which are obtained by linear regression of static components of neighboring vectors. While in (4.37) and (4.38) the notation $\mathbf{x}_t$ has been used for the complete feature vector, eventually including dynamic components, in this section $\mathbf{x}_t$ denotes only the static components. The full vector is denoted here by $(\mathbf{x}_t, \Delta\mathbf{x}_t, \Delta^2\mathbf{x}_t)$. This distinction has to be made as in DSR only the static components are transmitted through the channel. Thus, the corruption process affects directly only the

static components. The indirect corruption of the dynamic components as an effect of erroneous static components is discussed separately in Section 6.5.2.

6.5.1 Posterior of static components

As the observed feature vector $\mathbf{y}_t$ is obtained by one-to-one mapping of the received bit pattern of each subvector into the corresponding VQ centroid, the conditioning on $\mathbf{y}_t$ is equivalent to conditioning on $\hat{\mathbf{b}}_t$. The continuous posterior of the subvector can therefore be expressed as:

$$p(\mathbf{v}_t|\mathbf{y}_t) = p(\mathbf{v}_t|\hat{\mathbf{b}}_t) = \sum_{i=0}^{2^M-1} p(\mathbf{v}_t|\mathbf{b}_t^{(i)})P(\mathbf{b}_t^{(i)}|\hat{\mathbf{b}}_t), \tag{6.20}$$

or, conditioning on the whole sequence:

$$p(\mathbf{v}_t|\mathbf{y}_1^T) = p(\mathbf{v}_t|\hat{\mathbf{b}}_1^T) = \sum_{i=0}^{2^M-1} p(\mathbf{v}_t|\mathbf{b}_t^{(i)})P(\mathbf{b}_t^{(i)}|\hat{\mathbf{b}}_1^T). \tag{6.21}$$

The term $p(\mathbf{v}_t|\mathbf{b}_t^{(i)})$ is the probability density function of the subvector given the i^{th} bit pattern. That is, the distribution of those subvectors which are coded into the bit pattern i. It must be zero outside the i^{th} cluster and non zero within. While this can be specified exactly through knowing the prior distribution of the feature and the quantization clusters, we prefer to use an approximative parameterization having less parameters. Thus, we employed Gaussian PDFs, $\mathcal{N}(\mathbf{v}_t; \mathbf{c}_t^{(i)}, \mathbf{\Sigma}_t^{(i)})$ for each cluster. The mean value parameter $\mathbf{c}_t^{(i)}$ is the VQ centroid mapped to $\mathbf{b}_t^{(i)}$, and $\mathbf{\Sigma}_t^{(i)}$ the within-cell covariance matrix. The latter can be easily estimated on the training data by assuming the mean value to be $\mathbf{c}_t^{(i)}$ and computing the second order moment of feature subvectors quantized into the bit pattern $\mathbf{b}_t^{(i)}$. An even simpler parameterization is the Delta-Dirac distribution centered on the VQ centroid, $\delta(\mathbf{v}_t - \mathbf{c}_t^{(i)})$ which does not need any parameter estimation. This neglects, however, the uncertainty due to the quantization process. Thus, in explaining the approach we use the Gaussian approximation since it is more general. To particularize for Delta-Dirac distribution the Gaussian covariances must be set to zero.

In the following, we use the notation $\mathbf{y}$ for either $\mathbf{y}_t$ or $\mathbf{y}_1^T$ and $\hat{\mathbf{b}}$ for either $\hat{\mathbf{b}}_t$ or $\hat{\mathbf{b}}_1^T$.

For the reason explained in Section 4.3.3, the subvector posterior density needs to be approximated by a Gaussian density, but Eq. 6.20 and 6.21 are Gaussian mixtures with 2^M components. The mixture coefficients are the posterior bit pattern probabilities $P(\mathbf{b}_t^{(i)}|\hat{\mathbf{b}})$. The single Gaussian $\hat{p}(\mathbf{v}_t|\mathbf{y}) = \mathcal{N}(\mathbf{v}_t; \boldsymbol{\mu}_{\mathbf{v}_t|\mathbf{y}}, \boldsymbol{\Sigma}_{\mathbf{v}_t|\mathbf{y}})$ which best approximates the mixture is obtained by minimizing the Kullback-Leibler divergence between $\hat{p}(\mathbf{v}_t|\mathbf{y})$ and the original posterior $p(\mathbf{v}_t|\mathbf{y})$. This yields the following estimates:

$$\boldsymbol{\mu}_{\mathbf{v}_t|\mathbf{y}} = \sum_{i=0}^{2^M-1} P(\mathbf{b}_t^{(i)}|\hat{\mathbf{b}})\mathbf{c}_t^{(i)} \tag{6.22}$$

$$\boldsymbol{\Sigma}_{\mathbf{v}_t|\mathbf{y}} = \sum_{i=0}^{2^M-1} P(\mathbf{b}_t^{(i)}|\hat{\mathbf{b}}) \cdot ((\mathbf{c}_t^{(i)} - \boldsymbol{\mu}_{\mathbf{v}_t|\mathbf{y}}) \cdot (\mathbf{c}_t^{(i)} - \boldsymbol{\mu}_{\mathbf{v}_t|\mathbf{y}})^T + \boldsymbol{\Sigma}_t^{(i)}). \tag{6.23}$$

In Eq. 6.23 the covariance matrix can be seen as the sum of the average between-cell covariance and the average within-cell covariance. It has been observed experimentally that if M is sufficiently large, as it is the case with the ETSI-AFE quantization, the within-cell covariance can be neglected. This is none other than using Delta-Dirac approximation for $p(\mathbf{v}_t|\mathbf{b}_t^{(i)})$.

As the correlation between the subvectors of the same feature vector is rather small [85], the posterior of the static feature can be obtained as the product of all subvector posteriors. This yields a Gaussian with parameters $\boldsymbol{\mu}_{\mathbf{x}_t|\mathbf{y}}$ and $\boldsymbol{\Sigma}_{\mathbf{x}_t|\mathbf{y}}$ which are obtained by simply concatenating the parameters of all subvectors.

6.5.2 Posterior of dynamic components

Let $\mathbf{x}_t$, $\Delta\mathbf{x}_t$ and $\Delta^2\mathbf{x}_t$ denote the static, first-order and second-order temporal derivatives of subvectors of $\mathbf{x}_t$. They are usually computed as a linear function of the static components (linear regression):

$$\Delta\mathbf{x}_t = \sum_{k=-K}^{K} w_k \mathbf{x}_{t+k} \tag{6.24}$$

$$\Delta^2\mathbf{x}_t = \sum_{k=-L}^{L} v_k \Delta\mathbf{x}_{t+k} \tag{6.25}$$

Here, w_k and v_k are the linear regression parameters. Usual settings for the regression interval length are $K = 3$ and $L = 2$.

The posterior of the delta components $p(\Delta\mathbf{x}_t|\mathbf{y})$ could be computed as:

$$p(\Delta\mathbf{x}_t|\mathbf{y}) = \int_{\mathcal{D}_{\Delta\mathbf{x}_t}} p(\mathbf{x}_{t-K}, \ldots, \mathbf{x}_{t+K}|\mathbf{y}) d\mathbf{x}_{t-K} \ldots d\mathbf{x}_{t+K}, \tag{6.26}$$

where $\mathcal{D}_{\Delta\mathbf{x}_t}$ is the domain containing all sequences $\mathbf{x}_{t-K}, \ldots, \mathbf{x}_{t+K}$ which yield the same value of $\Delta\mathbf{x}_t$. A similar expression can be obtained for $\Delta^2\mathbf{x}_t$. Unfortunately, this approach is not practically applicable due to the complexity of computing the joint probability $p(\mathbf{x}_{t-K}, \ldots, \mathbf{x}_{t+K}|\mathbf{y})$.

The suboptimal approach which we employed is to infer the posterior of dynamic features from the posterior of static ones. The fundamental approximation allowing this is that the consecutive features $\mathbf{x}_{t-K}, \ldots, \mathbf{x}_{t+K}$ are Gaussians and statistically independent, given $\mathbf{y}$. According to (6.24), $\Delta\mathbf{x}_t$ is a linear combination of independent and normally distributed random variables and therefore it is also a Gaussian distribution with parameters:

$$\begin{aligned} \boldsymbol{\mu}_{\Delta\mathbf{x}_t|\mathbf{y}} &= \sum_{k=-K}^{K} w_k \boldsymbol{\mu}_{\mathbf{x}_{t+k}|\mathbf{y}} \\ \boldsymbol{\Sigma}_{\Delta\mathbf{x}_t|\mathbf{y}} &= \sum_{k=-K}^{K} w_k^2 \boldsymbol{\Sigma}_{\mathbf{x}_{t+k}|\mathbf{y}} \end{aligned} \tag{6.27}$$

Note that the expression for the covariance in (6.27) is valid for diagonal covariance matrices which is implicitly our case since we assumed that the individual feature vector dimensions are independent.

With the same considerations the posterior of delta-delta is computed as:

$$\begin{aligned} \boldsymbol{\mu}_{\Delta^2\mathbf{x}_t|\mathbf{y}} &= \sum_{k=-L}^{L} v_k \boldsymbol{\mu}_{\Delta\mathbf{x}_{t+k}|\mathbf{y}} \\ \boldsymbol{\Sigma}_{\Delta^2\mathbf{x}_t|\mathbf{y}} &= \sum_{k=-L}^{L} v_k^2 \boldsymbol{\Sigma}_{\Delta\mathbf{x}_{t+k}|\mathbf{y}} \end{aligned} \tag{6.28}$$

In practice, since the independence approximation does not hold, the variances of dynamic components tend to be over-estimated.

6.5.3 Posterior of complete vector

Assuming now that the three types of feature, static, delta and delta-delta, are independent (given the observations), the posterior of the complete feature vector becomes:

$$p(\mathbf{x}_t, \Delta\mathbf{x}_t, \Delta^2\mathbf{x}_t|\mathbf{y}) = p(\mathbf{x}_t|\mathbf{y}_1^T) \cdot p(\Delta\mathbf{x}_t|\mathbf{y}) \cdot p(\Delta^2\mathbf{x}_t|\mathbf{y}). \tag{6.29}$$

Thus, the Gaussian feature vector posterior to be used in (4.37) or (4.38) has the parameters:

$$\boldsymbol{\mu}_{\mathbf{x}_t,\Delta\mathbf{x}_t,\Delta^2\mathbf{x}_t|\mathbf{y}} = (\boldsymbol{\mu}_{\mathbf{x}_t|\mathbf{y}}, \boldsymbol{\mu}_{\Delta\mathbf{x}_t|\mathbf{y}}, \boldsymbol{\mu}_{\Delta^2\mathbf{x}_t|\mathbf{y}}) \tag{6.30}$$

$$\boldsymbol{\Sigma}_{\mathbf{x}_t,\Delta\mathbf{x}_t,\Delta^2\mathbf{x}_t|\mathbf{y}} = (\boldsymbol{\Sigma}_{\mathbf{x}_t|\mathbf{y}}, \boldsymbol{\Sigma}_{\Delta\mathbf{x}_t|\mathbf{y}}, \boldsymbol{\Sigma}_{\Delta^2\mathbf{x}_t|\mathbf{y}}) \tag{6.31}$$

Chapter 7

The application of uncertainty decoding to channel-error robust DSR

In this Chapter it is shown how uncertainty decoding can be used as an error concealment technique in a DSR scenario to alleviate the detrimental effect of channel errors. There are three sections. The first describes an experimental setup of a DSR system and the speech recognition tasks employed for performance evaluation. The second section evaluates a simulated DSR system in a network where the channel exhibits bit errors, a typical case for circuit-switched data links such as the GSM data channel. In the third section the DSR system is simulated in a packet-switched network where the channel errors consist of packet losses. Typical example used here is the transmission of data over the public Internet (TCP/IP).

The word error rates were determined for each scenario under various channel conditions and employing different error concealment techniques. The results obtained with the error concealment of ETSI-AFE and some other state-of-the-art techniques such as weighted Viterbi and MFT are also presented.

7.1 Experimental setup

The simulation system was realized for the most part in SPARK (Speech Processing and Recognition Toolkit), a simulation software package developed at the Department of Communications Engineering of the University of Paderborn. It consists in feature extraction at the client side, a transmission channel model, and recognition supporting uncertainty decoding at the server side. The front-end part is compliant with ETSI-AFE standard. The simulation of the GSM physical layer has been carried out using the GSM library of the CoWare SPW (Signal Processing

Worksystem) software suite [79].

For the training of the acoustic models we employed the HTK (Hidden Markov Model Toolkit) speech recognition software suite [97]. The speech recognition was carried out using our own speech decoder software with SPARK as it offers more flexibility in modifying the observation probability computation required by the uncertainty decoding rule. This is an one-pass Viterbi decoder with dynamically constructed lexical pronunciation trees which supports n-gram language models.

7.1.1 DSR simulation system

The block diagram of the DSR simulation system is given in Figure 7.1.

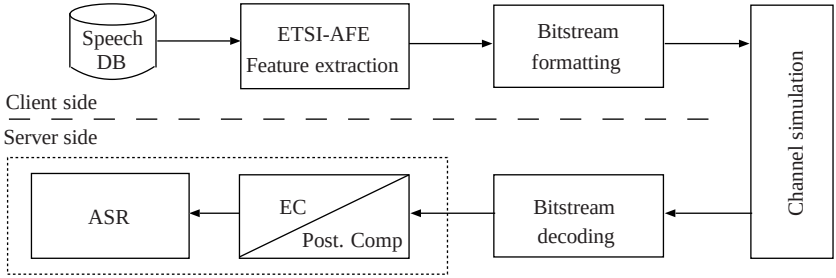

Figure 7.1: Block diagram of the DSR simulation system.

The block *Speech DB* comprises the speech data of the test set. They are processed in *ETSI-AFE* resulting in the bitstream of compressed feature vectors. This is formatted according to ETSI specifications and enters the *Channel simulation* block, which is described in Section 7.2 and 7.3. The bitstream obtained at the channel output is decoded into feature vectors which enter the error concealment (*EC*) block and, subsequently, the speech recognizer. While this is the case of reconstruction-based EC, in decoder-based techniques the error concealment takes place within the recognizer itself. For simulations with uncertainty decoding the block EC is labeled alternatively *posterior computation*.

The performance in terms of WER has been evaluated for each of the EC techniques listed below:

- **ETSI-DSR**: the EC provided by the DSR standard that is basically nearest frame repetition (NFR), see Section 2.3.1. This setup yields the reference

WER performance.

- **M**: marginalization, according to MFT, see Section 2.4.1. Unreliable features do not contribute to the acoustic score. The features were deemed unreliable either by means of CRC or, in case of packet loss, using packet loss indication.

- **WV**: weighted Viterbi EC as in Section 2.4.2. In case of the GSM channel the weighting coefficient γ has been computed by (2.9) using bit reliability. The parameter α in the above equation was tuned for best performance under the worst channel conditions. In case of the IP channel we employed the exponentially decaying weighting coefficient during loss bursts (2.13), a method proposed in [66].

- **UD0**: uncertainty decoding with feature posterior conditioned on instantaneous observation only (4.37).

- **UD1**: uncertainty decoding with feature posterior conditioned on the whole sequence of observations (4.38).

- **UD1-dyn**: uncertainty decoding with feature posterior conditioned on the whole sequence of observations (4.38) and employing extended source modeling of Section 6.4.4.

- **MMSE0**: the MMSE point estimate of the clean feature is the Gaussian mean of the feature posterior conditioned on instantaneous observation only (4.37). ASR uses the conventional decoding rule.

- **MMSE1**: the MMSE point estimate of the clean feature is the Gaussian mean of the feature posterior conditioned on the whole sequence of observations (4.36). ASR uses the conventional decoding rule.

- **MMSE1-dyn**: the MMSE point estimate of clean feature is the Gaussian mean of the feature posterior computed as for **UD1-dyn**.

Note that **ETSI-DSR**, **MMSE0**, **MMSE1**, and **MMSE1-dyn** do not imply changes in the speech recognizer. The EC is therefore separated from the recognition process. In contrast, the other techniques are decoder-based and require modification of the decoding rule in the recognizer. In all cases there are, however, no changes at the client side.

The experimental results are given in terms of WERs under various channel conditions: different C/I ratios for the GSM and a set of representative packet-loss conditions for IP.

7.1.2 Speech recognition tasks

The EC performance has been evaluated on two recognition tasks: Aurora 2 which is a small-vocabulary digit recognition task without a language model, and Wall Street Journal (WSJ0) a medium-vocabulary task with approx. 5000 words and a bigram language model. Each task is briefly described below.

1. *Aurora 2 task*. The small-vocabulary task is the clean test set of the Aurora 2 database, see [9] for an overview, consisting of 4004 utterances (continuously spoken digit strings) from 52 male and 52 female speakers, distributed over four subsets of 1001 utterances. The sampling rate is 8kHz. There is no language model for this task. The acoustic models, trained in clean conditions as described in [9], have 16 states per word with 3 Gaussians per state and diagonal covariance. With feature vectors computed by the ETSI DSR standard [12] the WER in error-free conditions (i.e. without channel-induced errors) was 0.86%.

2. *WSJ0 task*. The medium-vocabulary task is the Wall Street Journal WSJ0 5k Nov. '92 evaluation test set [98] comprising about 5000 words in 330 utterances of 4 male and 4 female speakers, summing up to 40 min of speech. Here, the sampling rate is 16kHz. Recognition experiments were carried out using a closed vocabulary bigram language model. The acoustic model consisted of 3437 tied states. The parameters of the 10-component mixture densities were trained on the SI-84 set of the WSJ corpus as in [99] using the HTK toolkit. With features computed by ETSI DSR standard [12], 16kHz extension, the WER in error-free conditions was 8.99%.

7.2 Evaluation of robustness to bit errors

A GSM data channel was considered as an example of a transmission channel exhibiting bit errors. Figure 7.2 shows the details of the DSR simulation system for this particular scenario. Note that the block diagram is relevant only for **MMSE0**, **MMSE1**, **UD0**, and **UD1** techniques.

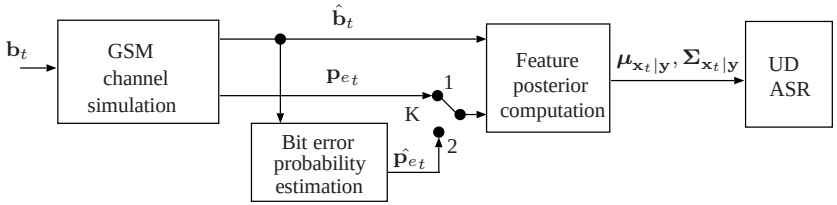

Figure 7.2: Block diagram of the DSR simulation system over GSM, channel and server side processing.

The transmission of the formatted bitstream $\mathbf{b}_t$ obtained at the client side is simulated in the block *GSM channel simulation*. The method employed here is either physical layer simulation or error pattern injection. The output consists of the decoded bitstream $\hat{\mathbf{b}}_t$ and, in the case of physical layer simulation, the associated bit error probabilities $\mathbf{p}_{e\,t}$. The computation of feature posterior is carried out using either the bit error probabilities delivered by the physical layer (the switch K on position 1), or the estimated bit error probabilities (the switch K on position 2). For details regarding bit error probability estimation the reader is referred to Chapter 5.

The Gaussian parameters $\boldsymbol{\mu}_{\mathbf{x}_t|\mathbf{y}}$, $\boldsymbol{\Sigma}_{\mathbf{x}_t|\mathbf{y}}$ of the feature posterior are fed into the recognizer where the classification takes place according to uncertainty decoding rules (4.37) or (4.38).

Under each channel condition, a complete recognition task has been performed using one of the EC technique listed in Section 7.1.1. Note that since the transmission errors are random, the word error rates obtained may slightly differ if the initialization of random number generators or the alignment with the utterance is changed. This effect has been also noticed in [94] the WER differences between simulations with different initializations were in average 0.04% (max. 0.69%) for Aurora 2 and 0.16% (max. 4.32% !) for WSJ0. Hence, it is particularly difficult to exactly reproduce the WERs obtained by other research groups using MMSE, weighted Viterbi or even NFR. In order to reduce these variations, a much larger test set should be used, which is not possible since this is determined by the speech database, or enough results obtained with different initializations should be averaged. In this work, for a fair comparison of performances, the various EC techniques are examined under identical error patterns, that is the bit errors or packet loss occurred exactly at the same positions in the utterance.

7.2.1 GSM physical layer simulation

In this experiment the GSM physical layer was simulated, including channel coding/decoding, interleaving/deinterleaving, modulation/demodulation, interference and fading of the channel. The channel coding was TCH/F4.8 [16], which uses convolutional coding with rate 1/3. The channel decoding employed the Bahl (Forward-Backward) algorithm after [84] which provides bit reliability along with the decoded bits. The channel model approximated a "typical urban" (TU) profile as specified by COST 207 [78] with 12 propagation paths, delay spread 1.03 μs and Rayleigh fading at a terminal velocity of 50 km/h. The Carrier-to-Interference (C/I) power ratio was set to 2.5, 4, 5.5 , 7 and 10 dB resulting in average bit error rates of 3.6%, 1.2%, 0.34%, 0.078% and 0.0025%, respectively. Note that a C/I of 10 dB is close to error-free transmission.

One set of simulations was carried out using the bit reliability information of the channel decoder (K on position 1 in Figure 7.2). The other set was carried out using the bit reliability estimated from the decoded bit pattern (K on position 2) as described in Section 5.1.3.2. This allows the quantifying of loss of performance by imperfect estimation of bit reliability, when the soft-output of the channel decoder is not available.

For the Aurora 2 task, Figure 7.3 presents the word error rates versus channel quality expressed by the C/I ratio. The bit error probabilities $\mathbf{p}_{\mathbf{e}_t}$ were obtained from the channel decoder (soft-output).

In Figure 7.4, the simulations were repeated on this occasion using bit error probabilities estimated from received data. The corresponding curves have the suffix **-EST** (estimated). For ease of comparison they are plotted along with WERs computed with channel decoder soft-output (those of Figure 7.3).

The same testing procedure was applied to the WSJ0 task and the word error rates are shown in Figures 7.5 and 7.6.

7.2.2 Channel errors simulated by GSM error patterns

Another method to simulate the bit errors in a GSM transmission is the injection of channel specific error patterns into the data stream, as mentioned in Section 5.1.2. The simulation with error patterns can be easily set up since it involves only an "exclusive or" operation between the sent bitstream and the error pattern in order to obtain the received bitstream. A disadvantage is that the method is unable to provide bit reliability information (decoder soft output) as the channel decoder is

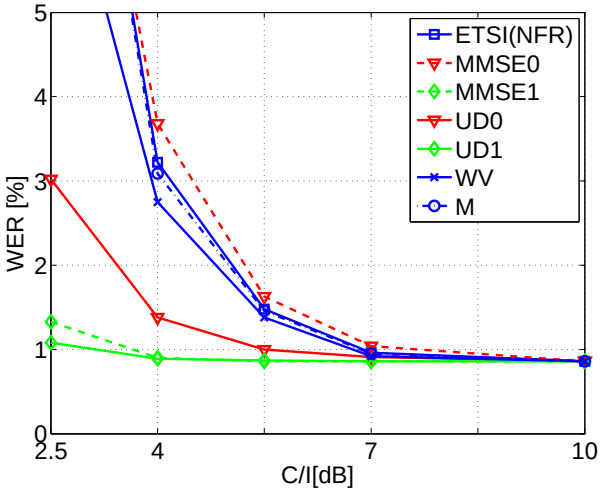

Figure 7.3: WERs on the Aurora 2 task, with bit error probabilities from the channel decoder

not explicitly part of the setup. To circumvent this problem an estimate of the bit reliability with the method described in Section 5.1.3.2 was employed, i.e. switch K on position 2 in Figure 7.2.

Tables 7.1 and 7.2 present the word error rates using different EC techniques, for the Aurora 2 and WSJ0 tasks, respectively. The transmission errors were simulated by the GSM error patterns EP1, EP2, and EP3 introduced in Section 5.1.2.

Table 7.1: Word error rates on the Aurora 2 task for GSM error patterns

EP	ETSI(NFR)	M	WV	UD0	UD1	MMSE0	MMSE1
EP1	0.90	0.90	0.90	0.90	0.90	0.90	0.90
EP2	0.90	1.01	1.22	0.97	0.97	1.43	0.96
EP3	5.43	7.68	8.20	2.67	1.95	11.80	2.26

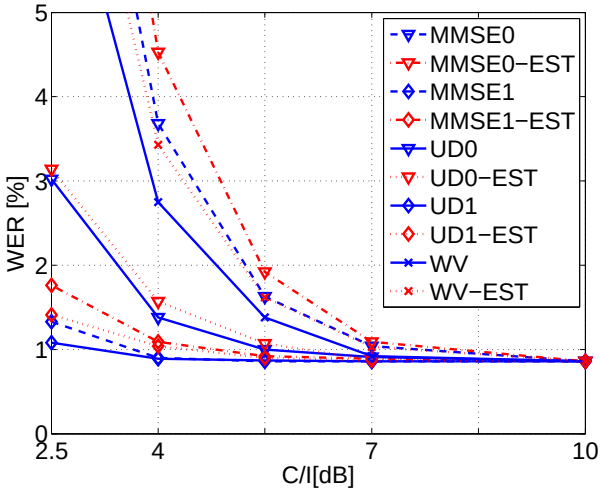

Figure 7.4: WERs on the Aurora 2 task, with bit error probabilities estimated from data

Table 7.2: Word error rates on the Wall Street Journal (WSJ0) task for GSM error patterns

EP	ETSI(NFR)	M	WV	UD0	UD1	MMSE0	MMSE1
EP1	8.97	8.95	8.85	8.96	8.95	8.99	8.97
EP2	9.75	9.73	9.19	9.00	9.02	9.42	9.02
EP3	32.75	26.23	17.63	14.33	11.43	22.06	12.24

7.2.3 Discussion

The results on the small-vocabulary task Aurora 2 and the medium-vocabulary task WSJ0 show analogous behaviour. The following observations apply to both tasks.

The importance of considering the inter-frame correlation for error concealment is emphasized by comparing the curves **UD0** and **UD1**. While both of them account for unreliability due to transmission errors, the latter performs better since, under the same conditions, it delivers a feature posterior with a smaller variance.

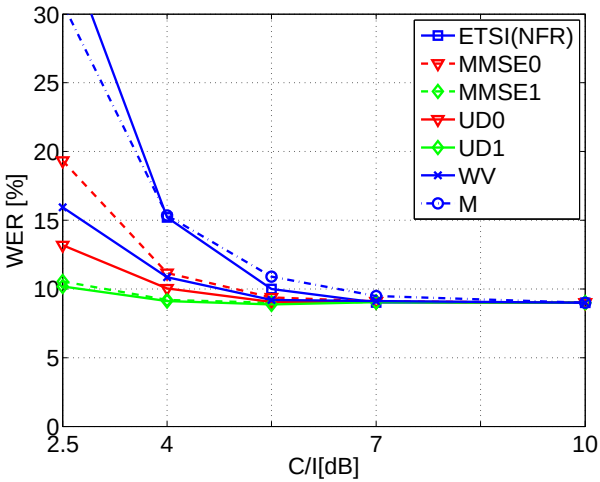

Figure 7.5: WERs on the WSJ0 task, with bit error probabilities from the channel decoder

This means that the clean feature estimate is frequently close to the true transmitted value. The variance is so small that even neglecting it, as **MMSE1** does, yields hardly any degradation in WER. This explains the slight difference between **MMSE1** and **UD1**.

If the inter-frame correlation is not considered, e.g. **MMSE0** and **UD0**, the clean feature is poorly estimated. Consequently, **MMSE0** performs even worse than the EC of ETSI-AFE. The improvement by **UD0** over **MMSE0** shows that if the variance of the feature posterior is high, speech recognition can benefit from the use of the modified decoding rule.

The marginalisation **M** exploits neither the inter-frame correlation nor the error-free portion of the feature vector. A feature vector does not produce discrimination even if it is only partially corrupted, e. g. only one component was affected. **WV** does not exploit the inter-frame correlation, too, but contrary to **M**, it is able to gradually deemphasise the discrimination produced by unreliable features.

When the instantaneous bit error probability is estimated from received data, small performance losses occur for all techniques employing it, however only at low C/I ratios, i.e. bad channel conditions. This validates our method used to derive the instantaneous bit error probability and proves that powerful EC tech-

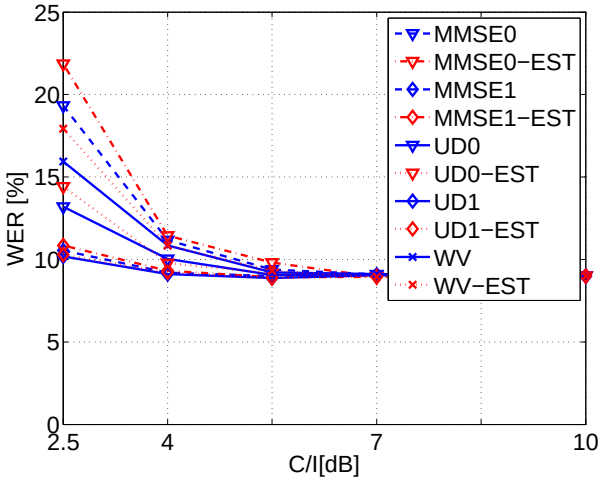

Figure 7.6: WERs on the WSJ0 task, with bit error probabilities estimated from data

niques such as **MMSE1** and **UD1** can be applied as well, even if the soft-output information of the channel decoder is not accessible at the server side of the DSR system.

7.3 Evaluation of robustness to packet loss

In order to evaluate the impact of packet loss on the recognition performance, a DSR system deployed in an IP network was considered. As shown in Section 5.2 packet erasure is the dominant error pattern in this scenario. The setup of the simulation system is shown in Figure 7.7.

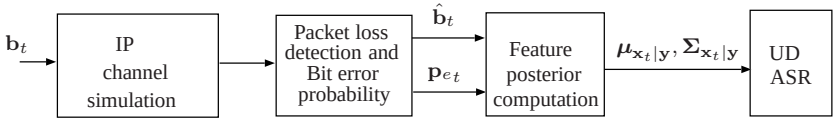

Figure 7.7: Block diagram of the DSR simulation system over IP, channel and server side processing.

The bitstream produced by the front end is packetized according to the RTP payload for DSR. In the block *IP channel simulation* the resulting data packets are randomly dropped. At the server side the missing packets are detected by means of the RTP sequence number. The packet sequence is then recovered by inserting packets containing random data (or zeros, since it makes no difference, see Section 5.2.4). The bit error probabilities of the bits of received packets is zero since they are reliable, while the bit error probabilities of the inserted bits are set to 1/2. The feature posterior is then computed and its parameters are fed into the uncertainty decoding ASR.

The following section presents the word error rates of Aurora 2 and WSJ0 tasks while packet loss has been simulated by a 2-state Markov chain.

7.3.1 IP channel with packet loss simulated by 2-state Markov chain

The simulation of packet loss was carried out by dropping packets according to the state of a 2-state Markov chain, a model already presented in Section 5.2.3. The parameters of the model were set to obtain particular channel conditions C1, ..., C4 as given in Table 7.3.

Table 7.3: The conditional loss probability (clp) and mean loss probability (mlp) of the IP network conditions simulated in this work.

Condition	C1	C2	C3	C4
clp	0.147	0.33	0.5	0.6
mlp	0.006	0.09	0.286	0.385

Note that the conditions C1 and C2 are characterized by short bursts and a relatively low packet loss ratio while C3 and C4 exhibit longer bursts and high packet loss ratios, e.g. up to 38%. Additionally the condition C0 denoting no loss was simulated.

The number of feature vectors per packet was either two, in one set of experiments, or four in the other set. Note that by using more features per packet, the average feature loss rate does not change, but the average burst length increases.

Figures 7.8 and 7.9 show the word error rates as a function of the channel

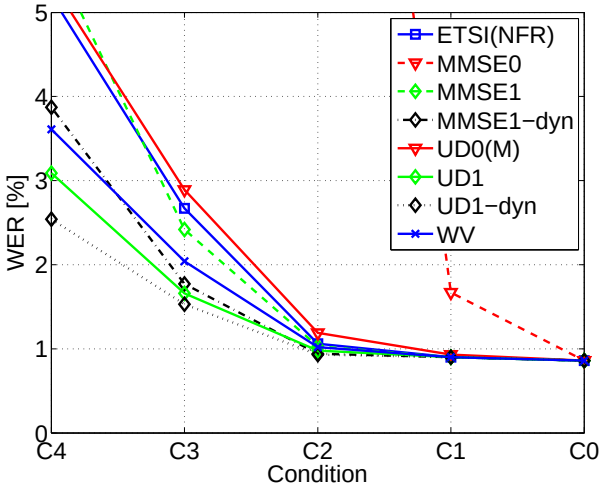

Figure 7.8: Word error rates on the Aurora 2 task at transmission over the packet-switched network with 2 feature vectors per packet.

condition. The results of Figure 7.8 were obtained with 2, whereas those of the latter were obtained with 4 features per packet.

The procedure was repeated with the WSJ0 task and the results are given in Figure 7.10 and 7.11.

7.3.2 Discussion

In the packet loss scenario the feature vector during an error burst has no reliable components. Thus, the feature posterior, without conditioning on reliable vectors before and after the burst, is same as the feature prior (unconditioned) probability density. The **UD0** methods reduces in this case to marginalisation **UD0(M)**, see also discussion in Section 4.3.1. Another consequence is that **MMSE0** reduces to simply inserting the feature priori mean value in the gap periods. The experimental results show that **UD0(M)** and **MMSE0** perform poorly.

An important improvement is obtained if the inter-frame correlation is considered (**UD1** and **MMSE1**). The feature posterior becomes more informative in this case, i.e. more focused on the clean (transmitted) feature vector. Contrary to the

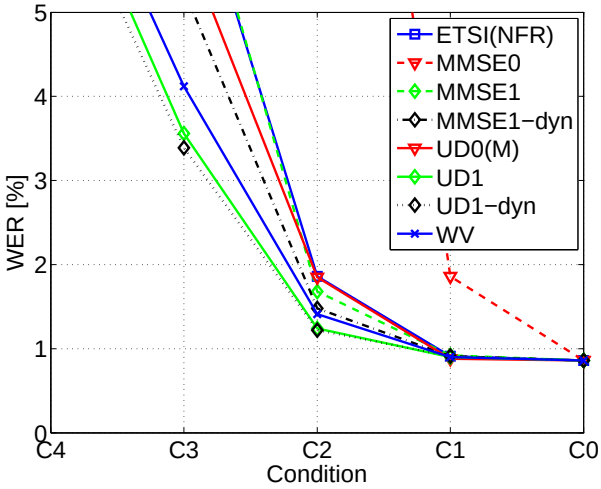

Figure 7.9: Word error rates on the Aurora 2 task at transmission over the packet-switched network with 4 feature vectors per packet.

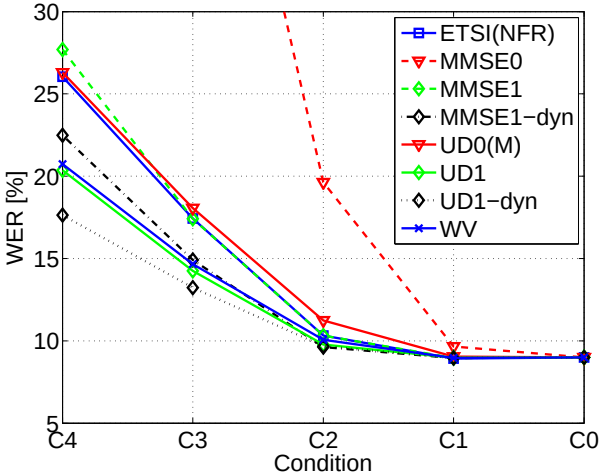

Figure 7.10: Word error rates on the WSJ0 task at transmission over the packet-switched network with 2 feature vectors per packet.

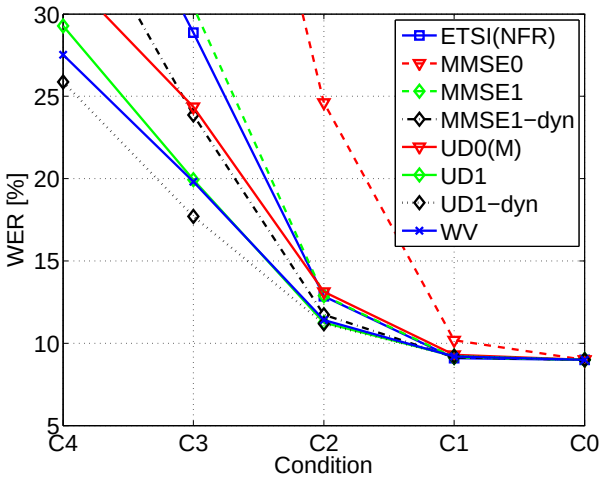

Figure 7.11: Word error rates on the WSJ0 task at transmission over the packet-switched network with 4 feature vectors per packet.

bit error scenario (GSM transmission), here employing the variance of the feature posterior density at recognition by **UD1**, yields better results than by neglecting it, as is done by **MMSE1**. This can be attributed to the fact that the feature posterior variance can be high, especially in the middle of long bursts, where it approaches the variance of the feature prior. Obviously, since all information is lost during bursts, the longer the distance from the unreliable feature to the burst ends, the smaller the amount of mutual information that can be utilized.

The use of more complex source models, which are better able to reproduce redundancy, e.g. **UD1-dyn**, **MMSE1-dyn**, is also beneficial in this case, contrary to the scenario with bit errors where modeling of feature static components alone was enough to obtain a feature posterior having a small variance.

The **WV** technique performs close to **UD1**. Although it does not explicitly exploit the correlation between consecutive features, **WV** assumes that the observation probability of the lost feature vector depends on that of the closest neighboring vector. As shown in [66], an exponential dependency is appropriate and can be confirmed by measurements on experimental data.

If the loss bursts are very long, which is more likely to happen, for example, by

using more features per packet under poor channel conditions, as shown in Figure 7.11, the uncertainty decoding tends to lose effectiveness comparing to **WV**. This can be an effect of the simplifying approximations that were made about the feature posterior and prior densities, and possibly insufficient training of the latter. They become critical in the middle of long bursts when the feature posterior and prior densities should be almost equal, but the estimation yields constantly different values. The observation probability is therefore still HMM state dependent which means that marginalization is not correctly performed. This can be avoided in practice by simply applying marginalization in middle of long bursts rather than by computing the posterior by FB, see also Section 9.2.3.1.

Note that the similar performance of **WV** and **UD1** is normally to be expected. In the Chapter 10 it is even demonstrated that weighted Viterbi is nothing more than a particular case of more general uncertainty decoding.

Chapter 8

The application of uncertainty decoding to channel-error robust NSR

In Network Speech Recognition the feature vectors are computed at the server side from the decoded speech waveform. The channel errors affect in this case the data stream of coded speech rather than the compressed features. Error concealment methods able to deliver the posterior probability of the decoded speech have been already proposed in [39, 38]. While the goal there was to compute an estimate (MMSE or MAP) of the decoded speech, we are interested here to infer the posterior of the feature vector from the posterior of the decoded speech. However, since the computation of one feature vector involves several samples of speech and non-linear operations, this inference is difficult. This chapter shows that in the case of packet-oriented transmission of coded speech, such as voice-over-IP, an approximation of the feature posterior can be readily obtained. The basic idea is to consider the feature vector lost, similarly to DSR over packet channels, if all (or most of the) speech samples required for its computation had been lost at transmission and subsequently reconstructed by PLC. The feature posterior computation methods of Section 7.3 can be easily adapted. The packet loss indicator (PLI) which indicates loss of coded speech frames is used to derive a feature loss indicator (FLI) for the estimation of the feature posterior.

The following section presents the setup of a conventional NSR system which has been used for evaluating WER under adverse channel conditions. Section 8.2 explains in detail how the FLI is obtained from the PLI in order to estimate the feature posterior. The last section of this chapter presents comparative results on the Aurora 2 task (see Section 7.1.2) using three widely used speech codecs: G.711, G.729A and G.723.1. The WERs achieved using the codec specific packet loss concealment (PLC) to reconstruct the speech waveform and conventional recogni-

tion are compared to those employing proposed uncertainty decoding.

8.1 NSR simulation system using VoIP

The conventional approach to NSR is depicted in Figure 8.1.

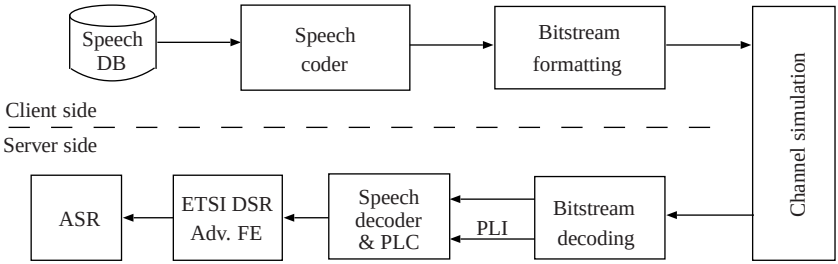

Figure 8.1: Block diagram of the conventional NSR simulation system.

The coded speech is sent to the server through a packet channel. Since the common transport protocol for VoIP is RTP, the data must be formatted according to some codec specific payload specifications in the *Bitstream formatting* block. At reception, the incoming data is decoded and usually packet loss concealment is applied to reconstruct the missing speech. ETSI feature extraction is then performed and the features used for speech recognition.

The performance of NFR over VoIP channels using this setup was evaluated on the Aurora 2 task. The speech coding was carried out using G.711 [100], G.729a [101], and G.723.1 [102]. The packet erasure channel was modeled as in the scenario with DSR over packet networks of Section 7.3.1. The remainder of this section gives a brief description of the speech codecs and their PLC algorithm:

- G.711 [100] has to be supported by all VoIP equipments [103]. The output of the coder represents logarithmic pulse-code modulation samples obtained at 8 kHz. In order to increase robustness in packet networks the decoding stage can be optionally provided with a packet loss concealment (PLC) algorithm to hide the transmission losses [104]. The PLC assumes transmission in packets of 10 ms speech. When a packet is lost, the pitch is estimated using the most recent 20 ms of speech. Using the estimate, a synthetic signal is generated for the duration of the lost frame (10 ms). For loss lengths

exceeding one frame, the synthetic signal is linearly attenuated by 20% per frame and then suppressed (silence) after 6 consecutive lost frames.

- G.729A is a speech codec using conjugate structure algebraic-code-excited linear-prediction (CS-ACELP) [101]. The speech signal sampled at 8 kHz is framed into 10 ms chunks. This codec provides a more advanced PLC. The LSF parameters for the lost frame are repeated from the previously received frame. Similarly, the adaptive and fixed codebook gains are taken from the previous frame but gradually attenuated. The excitation for the lost frame depends on the classification of the previous frame as voiced or unvoiced.

- G.723.1 is a dual rate coder for multimedia communication [102]. It belongs to the class of CELP analysis-by-synthesis hybrid codecs. The algorithm codes frames of 30 ms signal and can operate at two bit rates: high rate, 6.3 kbit/s (used in our simulations) and low rate, 5.3 kbit/s. Similarly to G.729A, it also has a built-in PLC.

Note that for each codec the RTP payload is specified by IETF recommendations. Due to latency constraints a data packet transports 10 ms of speech for G.711 and G.729A, and 30 ms for G.723.1. Packetization with more speech frames per packet, such as is done for DSR, are therefore not of interest here.

8.2 Derivation of feature loss indicator

The speech reconstructed by PLC in the case of a packet erasure is still corrupted. Consequently, this leads to an unreliable feature in the view of Chapter 4. In order to apply the uncertainty decoding rule, the posterior of the feature vector conditioned on decoded speech is required. The method proposed here is to cast the packet loss indicator (PLI) of the received coded speech into a binary indicator of feature reliability (FLI) as follows: If the feature is computed from a speech segment containing predominantly corrupted speech samples, i.e. speech reconstructed by PLC in the speech decoder, it is marked unreliable by setting the feature loss indicator. The FLI is subsequently used for feature posterior computation, analogous to PLI in DSR over packet networks.

The block diagram of the NSR system with uncertainty decoding is given in Figure 8.2. For convenience only the server side is depicted. The feature vectors are computed from the decoded speech and quantized into bit patterns $\hat{\mathbf{b}}_t$. The

FLI is computed in the block *Equivalent loss indicator*. The algorithm to compute the FLI depends of the speech codec, as it will be shown later. The FLI is then used to generate the bit error probability, i.e. 0 if the feature was extracted from reliable speech samples (FLI=0) and 1/2 if the speech samples were unreliable (FLI=1). Note that the vector quantization and bit error probability are still necessary since the a priori knowledge used for feature posterior computation consists of prior probabilities of the bit patterns. The posterior computation and uncertainty decoding are carried out as already shown in Section 7.3.

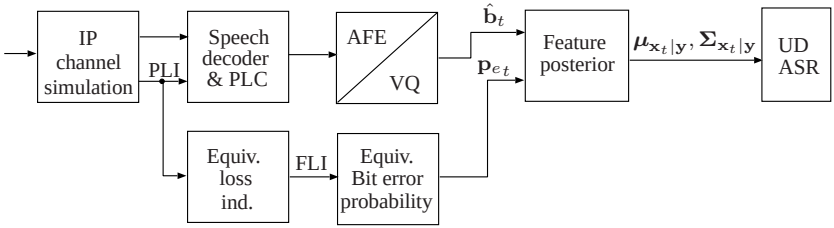

Figure 8.2: Block diagram of the simulation system for NSR over IP with uncertainty decoding (only server side processing).

The algorithm to derive the feature reliability from the packet loss indication provided by the communication channel is given in the following assuming the G.711 speech codec. In the absence of losses, the decoding of one speech frame uses only the information from one data packet. Thus, a lost packet affects exactly one speech frame and has no effect on subsequent frames. The feature extraction process is depicted in Figure 8.3.

The dashed region in the upper part of the figure denotes that the t^{th} frame of coded speech has been lost. The 10 ms of speech corresponding to this packet is reconstructed by PLC and thus is unreliable, and is represented in dark gray, whereas the other frames are reliable and represented as white rectangles. According to ETSI-AFE, the feature vectors are computed from overlapping speech segments of 25 ms (200 samples at an 8 kHz sample rate). For each vector the window shifts 10 ms (80 samples). E.g. the window t from which the feature t is computed contains 10 ms from decoded speech frame t, 10 ms from $t-1$ and 5 ms from $t-2$. Thus, it can be easily observed that an unreliable speech frame at time t affects not only the feature t but also its successors $t+1$ and $t+2$ since their windowed signal contains samples of unreliable speech. As the window is centered on the middle of the frame, the feature at $t+1$ (dark gray) is more affected

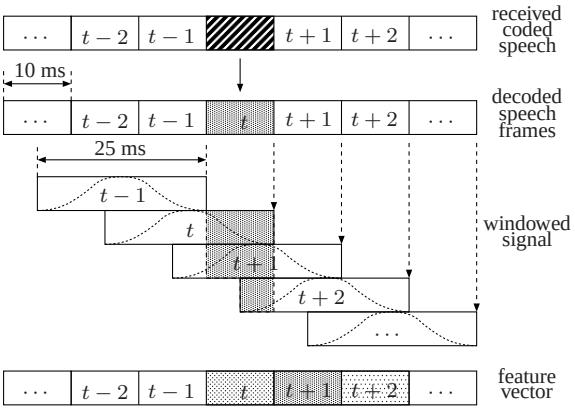

Figure 8.3: Derivation of the feature loss indicator (FLI) from the packet loss indicator (PLI)

than those at t and $t+2$ (light gray). It means that the most unreliable feature is obtained in the next frame after the loss, i.e. after τ_{FE}=10 ms. By the subscript FE (Feature Extraction) it is meant that the latency in reaching the maximum of unreliability after a lost frame is a characteristic of the feature extraction and does not depend on the speech codec.

Another effect which has to be considered when deriving the FLI is "codec memory noise", as it was termed in [19]. The authors observed that, due to the predictive nature of the encoding process, the speech decoder needs some time (frames) to recover after a lost/bad frame until it can produce an output which is again close to the original uncoded speech. This means that even if the PLI indicates a correctly received frame, the decoded speech might still be affected by some preceding lost frames. The time elapsed between the packet loss event and observing its effect on decoded output is denoted by τ_C and depends on the speech codec. Note that G.711 codes each sample independently of the others and shows therefore no memory noise effect. The packet loss is immediately noted at the output ($\tau_C = 0$), as the reconstructed frame significantly differs from the original one. For the G.729A and G.723.1 we have experimentally determined a τ_C of 10 and 15 ms, respectively. This has been done by measuring uncertainty decoding performance with various τ_C to find an optimum.

In order to encompass both effects described above, the PLI delayed by $\tau =$

$\tau_{FE} + \tau_C$ gives a binary indication about the reliability of each 10 ms segment of decoded speech. Under these considerations we devised the following rule to classify a feature vector: The feature vector is considered corrupted, i.e. FLI indicates feature loss, if at least B segments of 10 ms involved in its computation contain corrupted speech. Since a feature vector is computed using 25 ms of speech, B may take the values 1, 2, or 3 and therefore can be easily determined on an experimental basis.

This is shown in Table 8.1 which gives the WERs obtained using the G.711 (a-law) speech codec with PLC and performing uncertainty decoding for possible values of B. The best choice is B=2 which yields the best performance. With a smaller B, i.e. $B = 1$, the feature is declared lost even if only 10 ms of the 25 ms window (less than half) is affected by the erasure. This discards too much useful information. With B=3, features classified as reliable might exist although up to 20 ms of the windowed speech is corrupt. Thus, corrupted features are deemed reliable which leads to mismatch.

Table 8.1: WERs [%] on the Aurora 2 task with NSR employing G.711 (a-law) with PLC and uncertainty decoding. B is the minimum number of corrupted 10 ms-segments required to declare a feature vector unreliable.

Condition	C1	C2	C3	C4
B=1	0.99	1.19	4.83	10.25
B=2	0.99	1.02	1.68	2.95
B=3	0.99	1.06	2.20	3.61

8.3 Evaluation of robustness in an NSR scenario

The setup of Figure 8.1 was employed first to carry out recognition tasks using each of G.711, G729A and G.723.1 codecs. The channel model and channel conditions (C0,...,C4) were those used for DSR over IP networks. The goal of these experiments was to obtain the performance of the conventional NSR system where the features are extracted from decoded, possibly PLC reconstructed, speech.

The simulations with uncertainty decoding used the setup of Figure 8.2 and the threshold for declaring the feature unreliable as $B = 2$.

The WERs obtained with G.711 speech coding algorithm are shown in the Table 8.2.

Table 8.2: WERs [%] on the Aurora 2 task with NSR employing G.711.

Condition	C0	C1	C2	C3	C4
G.711	0.90	0.95	5.23	29.5	48.9
G.711-PLC	0.90	0.94	1.22	3.58	8.64
G.711-UD	0.91	0.93	0.98	1.60	2.95

Since the PLC is an optional part of G.711, the row labeled G.711 shows the WERs obtained without PLC. The missing segments of the speech waveform were simply filled with zero (silence). The second row shows the WERs when PLC is performed. It can be seen that the PLC is beneficial for recognition since WER drastically decreases particularly under adverse conditions. The row G.711-UD shows the results for recognition with uncertainty (**UD1**, exploiting temporal correlation, Eq. 4.38). The features computed from more than 20 ms speech reconstructed by PLC ($B = 2$) were considered lost and their posterior density was estimated as in DSR over packet channels scenario, otherwise they were deemed reliable. By considering the uncertainty, the system becomes very robust as regards packet erasure, e.g. at 38% loss ratio (C4) the WER is more than halved compared to G.711-PLC.

The Tables 8.3 and 8.4 present the WER achieved using **UD1** along with the G729A and G.723.1 codecs, respectively.

Table 8.3: WERs [%] on the Aurora 2 task with NSR employing G.729A.

Condition	C0	C1	C2	C3	C4
G.729A-PLC	1.60	1.77	3.99	16.47	27.36
G.729A-UD	1.60	1.65	2.91	7.62	11.90

Note that even in the error-free condition (C0) the mismatch between the acoustic model trained with uncoded speech and the features computed from decoded speech produces a degradation in WER. While with G.711 approximately the same

Table 8.4: WERs [%] on the Aurora 2 task with NSR employing G.723.1.

Condition	C0	C1	C2	C3	C4
G.723.1-PLC	1.88	2.60	7.30	26.76	40.00
G.723.1-UD	1.88	1.98	3.56	11.36	19.97

word error rate is achieved as without speech coding (0.86%), it increases to 1.60% and 1.88% for G.729A and G.723.1, respectively.

For all three codecs, the improvement achieved by uncertainty decoding compared with the performance of the same codec by standard recognition ignoring uncertainty is considerable. Note, however, that the intention here is not a performance comparison between codecs, as this would also need the consideration of bit rates, packetization schemes etc.

Chapter 9

Computational complexity and speed-up methods

The novel uncertainty decoding rule proposed in this work implies inherent latency (algorithmic delay), additional computational expense for the feature posterior estimation and for the feature log-likelihood evaluation, and also an expansion of the acoustic search space. This chapter analyses the negative factors mentioned above and proposes simplifying assumptions and techniques aimed at speeding up the computation.

9.1 Error bursts detection

In the computation of the feature posterior at the time t according to (4.38), all past and future observations $\mathbf{y_1}, \ldots, \mathbf{y_T}$ of the utterance are required. This implies that the computation has to be performed off-line after the whole sequence of features has been observed. Whereas this is the general case where all observations are assumed unreliable, in a practical situation, such as a packet loss scenario, the reliable and unreliable regions of an utterance alternate so that the latter can be isolated in bursts as depicted in Figure 9.1. The figure shows the received sequence of feature vectors (observations) of an utterance, $T = 9$. The unreliable observations, i.e. where transmission errors occurred, are represented as dark areas while the reliable observations are shown in white. The feature posterior of a reliable feature vector is, as discussed in Chapter 4, a delta Dirac distribution, e.g. $p(\mathbf{x}_2|\mathbf{y}_1^9) = p(\mathbf{x}_2|\mathbf{y}_2) = \delta(\mathbf{x}_2 - \mathbf{y}_2)$. Since the observed $\mathbf{y}_2$ is reliable, it already contains all knowledge about $\mathbf{x}_2$ rendering the neighboring observations unnecessary. Due to similar considerations, within an error burst, e.g. at $t = 5$, the

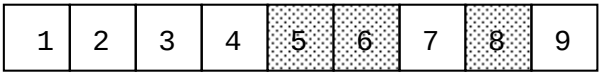

Figure 9.1: Sequence of feature vectors of an utterance. Dotted areas are corrupted.

feature vector posterior is conditioned on all unreliable observations within that burst as well as on the last and first reliable observation before and after the burst, respectively: $p(\mathbf{x}_5|\mathbf{y}_1^9) = p(\mathbf{x}_5|\mathbf{y}_4^7)$.

In the case of DSR the burst detection by CRC and data consistency check described in [12] can be used for our purpose. Thus, it can be concluded that if the observation $\mathbf{y}_t$ is reliable, the computation of the feature vector posterior $p(\mathbf{x}_t|\mathbf{y}_1^T)$ implies no algorithmic delay, i.e. does not need the future observation(s). Within an error burst the algorithmic delay can be as high as the burst length, as in the case of the nearest frame repetition approach of ETSI-AFE.

Note that the speech decoding process itself, even under real-time constraints, implies a variable latency from the end of word, until the word sequence up to that point is decoded. Compared to this, the latency due to the feature vector posterior computation during error bursts is quite small, but is an additional disadvantage.

Besides making on-line processing possible, error burst detection significantly contributes to computational reduction. This is because the computationally expensive estimation of the posterior by the forward-backward algorithm needs to be done only within bursts, whereas outside them the posterior estimation is trivial, as it is a delta Dirac PDF.

9.2 Feature posterior computation

Assuming that the feature posterior is Gaussian with a diagonal covariance matrix, the evaluation of (4.43) requires first the estimation of the D-dimensional mean $\boldsymbol{\mu}_e$ and D-dimensional variance vector $\boldsymbol{\sigma}_e^2$. These parameters are obtained from the probability mass function of the transmitted bit patterns as described in Section 6.5 at relatively low computational expense. The most computationally intensive step is obtaining the posterior probability mass function of the transmitted bit patterns, since it (only that of **UD1**) implies the forward-backward (FB) recursion. The following paragraphs evaluate the computational complexity of the FB algorithm and

propose approximations that lead to a drastic reduction of complexity at marginal cost or even no loss in accuracy.

9.2.1 Complexity of the Forward-Backward algorithm

In the view of Section 6.4.3, the unobserved sent bit patterns represent the states of a hidden Markov model. There are as many states as there are possible values for the sent bit pattern (2^M, with M being the number of quantization bits). The forward recursion (6.15) gives a computational complexity of the order of 2^{2M} operations per iteration if we assume for simplicity that the HMM states are fully connected. This is because there are $2^M \cdot 2^M$ possible transitions that must be evaluated at each iteration. Since there is a forward and a backward recursion, a factor of two must be also considered. These altogether result in a complexity of the order 2^{2M+1} per subvector and frame. For the vector quantization scheme of ETSI-AFE, the contribution of the seventh subvector sv_7 ($M = 8$) dominates, as the other subvectors are quantized with 6 or 5 bits. Hence the complexity may be as high as about 2^{17} operations per frame. This is an upper bound of complexity since in practice the HMM states are not fully connected resulting in fewer transitions to be evaluated. This is a consequence of continuity constraints along each dimension to which the feature vector has to adhere. The variation between two frames cannot therefore exceed some specific threshold (data consistency test of ETSI-AFE exploits the same principle) and therefore not all transitions between consecutive VQ centroids (HMM states) are possible. By considering only the non-zero probability transitions in the FB recursion, the complexity is well below the computed upper bound. Measurements on a workstation with 2.3 GHz Intel Xeon 5140 have shown that feature posterior computation (Section 6.4.3) including FB and subsequent steps, needs about 0.3 ms per frame, corresponding to a real time (RT) factor of 0.03. Depending on the recognition task this may represent a significant portion of the processing time. This consideration shows that a reduction of computational effort is highly desirable.

9.2.2 Forward approach for computation of the feature posterior

The computational burden of the feature posterior computation can be easily halved by considering only the forward recursion. This approximation has been used in [50, 67, 90]. By doing so, the correlation between the current bit pattern and its

successors is neglected and thus, only the dependency on the predecessors is considered:

$$P(\mathbf{b}_t^{(i)}|\hat{\mathbf{b}}_1^T) \approx P(\mathbf{b}_t^{(i)}|\hat{\mathbf{b}}_1^t) \tag{9.1}$$

In this case the bit pattern posterior is computed using the forward probabilities $\alpha_t(i)$ (see Eq. 6.15) as:

$$P(\mathbf{b}_t^{(i)}|\hat{\mathbf{b}}_1^t) = \frac{\alpha_t(i)}{\sum_{j=0}^{2^M-1} \alpha_t(j)} \tag{9.2}$$

Note that by using only the forward recursion, the algorithmic delay, see Section 9.1, associated to the backward recursion is eliminated. This can be an advantage for applications that do not tolerate such a processing delay, e.g. [39].

The approximation (9.1) results in performance loss especially under poor channel conditions. Tables 9.1 and 9.2 show for comparison the recognition WERs on the Aurora 2 task obtained using the forward-backward recursion UD-FB (which is same as **UD1** of Section 7.1.1) and forward-only recursion UD-F. The experimental setup was DSR over GSM with physical layer simulation, for Table 9.1 and DSR over IP for Table 9.2, respectively.

Table 9.1: WERs on the Aurora 2 task for DSR over GSM

C/I [dB]	2.5	4	5.5	7	10
UD-FB	1.15	0.90	0.89	0.87	0.86
UD-F	1.32	0.93	0.90	0.88	0.86

It can be observed that in the GSM scenario, the loss of performance by only making the forward approximation is not significant. This can be attributed to the fact that, unlike with the IP scenario, usually not all information of a frame is erroneous (lost). The observed feature vectors are still informative enough to produce a good posterior estimate, even conditioned on predecessors only. The amount of information gained by also considering the successors is small.

In the IP scenario, the gap between forward and forward-backward is more pronounced. Since there are no observations during the loss bursts, the posterior is

Table 9.2: WERs on the Aurora 2 task for DSR over IP, two feature vectors per packet

Cond.	C4	C3	C2	C1	C0
UD-FB	3.09	1.66	0.98	0.90	0.86
UD-F	4.93	2.50	1.09	0.90	0.86

in fact conditioned on the last observation before and the first observation after the burst, respectively. Clearly, the forward recursion exploits only the last observation before the burst which becomes less informative toward the end of burst, resulting in degraded performance.

In conclusion, the forward approach is suitable for DSR over circuit-switched channels but yields poor robustness to packet loss.

9.2.3 Table lookup approach

This section presents a method to speed up the forward-backward recursion, which is applicable in case of DSR over lossy packet channels. Figure 9.2 depicts a sequence of bit patterns observed at the channel output, where the dark areas correspond to lost bit patterns. The bit pattern sequence was renumbered starting from the last correctly received bit pattern before the loss burst. We are interested in computing the bit pattern posterior for $t = 2 \dots T-1$, since for the correctly received frames conventional decoding can be used, see Section 9.1.

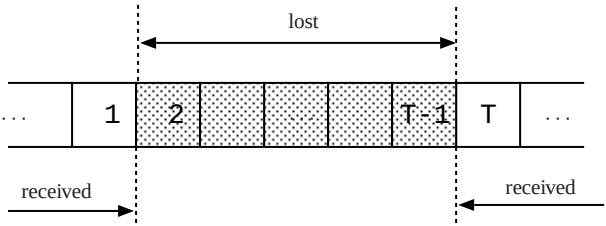

Figure 9.2: Sequence of bit patterns affected by a loss burst. Dotted areas are lost.

As has already been mentioned in Section 6.3, the transition probability for the

bit patterns within the burst is constant and independent of the bit pattern index (i), i.e. $P(\hat{\mathbf{b}}_t|\mathbf{b}_t^{(i)}, \mathbf{z}_t) = \text{const.}$, $i = 0, \ldots, 2^M - 1$; $t = 2 \ldots T - 1$. Thus, using that the posterior is invariable to particular scaling of $\alpha(i)$ and $\beta(i)$, the forward-backward (6.15) recursion can be simplified to:

$$\begin{aligned} \alpha_{t+1}(i) &= \sum_{j=0}^{2^M-1} \alpha_t(j) P(\mathbf{b}_{t+1}^{(i)}|\mathbf{b}_t^{(j)}) \\ \beta_{t-1}(i) &= \sum_{j=0}^{2^M-1} \beta_t(j) P(\mathbf{b}_t^{(j)}|\mathbf{b}_{t-1}^{(i)}) \end{aligned} \tag{9.3}$$

Further, by defining the row vectors:

$$\begin{aligned} \boldsymbol{\alpha}_t &= (\alpha_t(0), \ldots, \alpha_t(2^M - 1)) \\ \boldsymbol{\beta}_t &= (\beta_t(0), \ldots, \beta_t(2^M - 1)) \end{aligned} \tag{9.4}$$

and the $(2^M \times 2^M)$-dimensional matrix of HMM state transitions, $(\mathbf{A})_{ij} = P(\mathbf{b}_t^{(j)}|\mathbf{b}_{t-1}^{(i)})$, the Eqs. 9.3 become:

$$\boldsymbol{\alpha}_{t+1} = \boldsymbol{\alpha}_t \mathbf{A} = \boldsymbol{\alpha}_1 \mathbf{A}^t \tag{9.5}$$

$$\boldsymbol{\beta}_{t-1} = \boldsymbol{\beta}_t \mathbf{A}' = \boldsymbol{\beta}_T (\mathbf{A}^{T-t+1})', \tag{9.6}$$

where $\mathbf{A}'$ denotes the transposed matrix and the initialization vectors $\boldsymbol{\alpha}_1$ and $\boldsymbol{\beta}_T$ are:

$$\boldsymbol{\alpha}_1(i) = P(\hat{\mathbf{b}}_1|\mathbf{b}_1^{(i)}) = \delta(i - i_s) \tag{9.7}$$

$$\boldsymbol{\beta}_T(i) = P(\hat{\mathbf{b}}_T|\mathbf{b}_T^{(i)}) = \delta(i - i_e), \tag{9.8}$$

with i_s and i_e being the indices of $\hat{\mathbf{b}}_1$ and $\hat{\mathbf{b}}_T$ respectively.

9.2.3.1 Matrix lookup

The Eqs. 9.5 and 9.6 show that if $\mathbf{A}^t$ and $\mathbf{A}^{T-t+1}$ are computed in advance and stored, the most computation during runtime can be saved, since computing $\boldsymbol{\alpha}_{t+1}$

reduces to selecting the i_s-th row of the matrix $\mathbf{A}^t$, whereas computing $\boldsymbol{\beta}_t$ reduces to selecting the i_e-th column of $\mathbf{A}^{T-t+1}$. If the matrices $\mathbf{A}, \mathbf{A}^2, \ldots, \mathbf{A}^L$ are stored, the Eqs. 9.5 and 9.6 allow the processing of bursts of a maximum length L. Obviously, the memory demands increase correspondingly as we need to store L matrices instead of one. For the quantization scheme used in the ETSI-AFE for DSR [12] this amounts to $L \times (5 \cdot 2^{2 \cdot 6} + 1 \cdot 2^{2 \cdot 5} + 1 \cdot 2^{2 \cdot 8}) = L \times 87040$ values.

The memory demands are therefore dependent on the maximum burst length which may occur. However, this cannot be known in advance. In [58] we observed that the depth L can be limited to a suitable value without loosing performance. The rationale is that there exists a depth L beyond that the matrix $\mathbf{A}^k$, $k > L$, does not significantly changes and can be approximated by $\mathbf{A}^\infty$. This is a known property of a Markov chain [105] stating that, independent of the initial state distribution, after enough iterations, stationarity is achieved. This means that the state distribution at time t, which is obtained as a product of initial state distribution and $\mathbf{A}^t$ tends to the a priori distribution. The forward and backward recursions are schematized in Fig. 9.3.

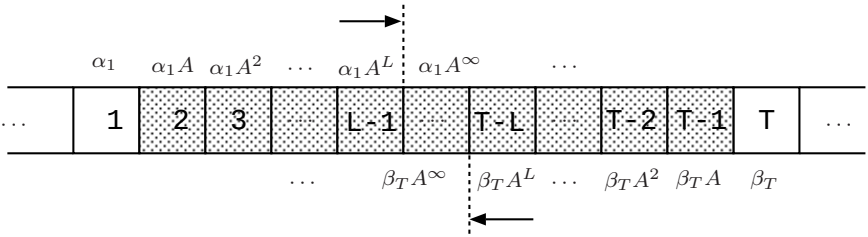

Figure 9.3: Forward and backward recursion if transition probability is constant within burst. Iterations stopped after L steps in each direction.

We carried out experiments for various L in order find a suitable value. Table 9.3 shows the WERs on the Aurora 2 task assuming a DSR over IP setup and using the table lookup method with limited depth L. For convenience the baseline results using the original forward-backward algorithm (UD-FB) are given as well.

It can be concluded that for this recognition task, a depth $L = 4$ provides enough accuracy for processing bursts of any length.

Table 9.3: WERs on the Aurora 2 task for DSR over IP with lookup method, two feature vectors per packet

Cond.	C4	C3	C2	C1	C0
UD-FB	3.09	1.66	0.98	0.90	0.86
L=1	4.10	2.20	1.09	0.90	0.86
L=2	3.42	1.88	1.04	0.90	0.86
L=3	2.97	1.68	0.99	0.90	0.86
L=4	2.80	1.60	0.97	0.90	0.86
L=5	2.80	1.60	0.97	0.90	0.86
L=6	2.92	1.60	0.97	0.90	0.86
L=8	2.99	1.66	0.98	0.90	0.86
L=12	3.08	1.66	0.98	0.90	0.86
L=14	3.09	1.66	0.98	0.90	0.86

9.2.3.2 Suboptimal approach

The above solution is aimed at reducing the computational effort but at the cost of increased memory demands. While this could be achieved with only minor approximations to the original FB algorithm, a dramatic reduction of memory demands can be achieved by a suboptimal approach.

This is based on the fact that the correlation between two features $\mathbf{x}_t$ and $\mathbf{x}_{t-\tau}$ weakens by increasing τ. This has been experimentally proven as shown in Table 9.4 which gives the mutual information $I(\mathbf{b}_t;\mathbf{b}_{t-\tau}) = H(\mathbf{b}_t) - H(\mathbf{b}_t|\mathbf{b}_{t-\tau})$, where $H(\mathbf{b}_t)$ denotes the entropy of the bit pattern $\mathbf{b}_t$. $I(\mathbf{b}_t;\mathbf{b}_{t-\tau})$ is a measure of the information about $\mathbf{b}_t$, that is contained in $\mathbf{b}_{t-\tau}$ and thus indicates whether it is useful to utilize $\mathbf{b}_{t-\tau}$ for the reconstruction of $\mathbf{b}_t$. The values have been obtained using the ETSI-AFE on the Aurora 2 training set. It can be seen that $I(\mathbf{b}_t;\mathbf{b}_{t-\tau})$ becomes smaller (tends to zero theoretically) as τ increases. Note that if the conditional entropy equals the unconditional, the rows of A^τ, become equal to the a priori distribution $P(\mathbf{b}_t^{(j)})$ [105, p.161]. This property was used also in the matrix lookup approach. Eq. 9.5 states that $(\mathbf{A}^\tau)_{ij}$ is none other than $P(\mathbf{b}_t^{(j)}|\mathbf{b}_{t-\tau}^{(i)})$ which becomes independent of i for a large enough τ.

This property can be exploited at least for long bursts where the distance τ between the lost feature and one burst end is large enough so that the statistical

Table 9.4: Entropies and mutual information among b_t and b_{t-1} produced by the ETSI advanced DSR front-end.

Subvector	sv_1	sv_2	sv_3	sv_4	sv_5	sv_6	sv_7
M	6	6	6	6	6	5	8
$H(b_t)$	5.8	5.8	5.8	5.8	5.8	4.8	7.7
$I(b_t;b_{t-1})$	2.6	2.1	1.6	1.4	1.2	1.0	3.4
$I(b_t;b_{t-2})$	1.7	1.3	0.9	0.8	0.7	0.6	2.8
$I(b_t;b_{t-3})$	1.2	0.9	0.7	0.6	0.5	0.4	2.1
$I(b_t;b_{t-4})$	0.9	0.7	0.5	0.4	0.3	0.3	1.8
$I(b_t;b_{t-5})$	0.7	0.5	0.3	0.3	0.2	0.2	1.4

dependency of that burst end can be neglected. Thus, in the first half of the burst the bit pattern posterior depends mainly on $\hat{\mathbf{b}}_1$ so that is can be approached by the forward probabilities $\boldsymbol{\alpha}$. Similarly, in the second half of the burst the posterior depends mainly on $\hat{\mathbf{b}}_T$ so that only backward probabilities can be used.

$$P(\mathbf{b}_t^{(i)}|\hat{\mathbf{b}}_1^T) \approx \begin{cases} P(\mathbf{b}_t^{(i)}|\hat{\mathbf{b}}_1) & \text{if } t \leq \frac{T}{2} \\ P(\mathbf{b}_t^{(i)}|\hat{\mathbf{b}}_T) & \text{if } t > \frac{T}{2} \end{cases} \tag{9.9}$$

According to the definition of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, the above equation comes out to:

$$P(\mathbf{b}_t^{(i)}|\hat{\mathbf{b}}_1^T) \approx \begin{cases} \frac{\alpha_t(i)}{\sum_{j=0}^{2^M-1}\alpha_t(j)} & \text{if } t \leq \frac{T}{2} \\ \frac{\beta_t(i)P(\mathbf{b}^{(i)})}{\sum_{j=0}^{2^M-1}\beta_t(j)P(\mathbf{b}^{(j)})} & \text{if } t > \frac{T}{2} \end{cases} \tag{9.10}$$

Using the vector notations (9.4) and

$$\mathbf{c} = \left[c^{(0)}, \ldots, c^{(2^M-1)}\right], \tag{9.11}$$

$$\mathbf{c^2} = \left[(c^{(0)})^2, \ldots, (c^{(2^M-1)})^2\right] \tag{9.12}$$

for the vector of codebook centroids and their squared values, respectively, the mean and variance of the feature posterior can be expressed as:

$$\mu_{x_t|\hat{\mathbf{b}}_1} = \sum_{i=0}^{2^M-1} c^{(i)} \frac{\alpha_t(i)}{\sum_{j=0}^{2^M-1} \alpha_t(j)} = \frac{\mathbf{c}\boldsymbol{\alpha_t}'}{\sum \boldsymbol{\alpha_t}} = \frac{(\mathbf{A}^t)'}{\sum \boldsymbol{\alpha}_1 \mathbf{A}^t} \cdot \boldsymbol{\alpha}'_1 \tag{9.13}$$

$$\begin{aligned} \sigma^2_{x_t|\hat{\mathbf{b}}_1} &= \sum_{i=0}^{2^M-1} (c^{(i)} - \mu_{x_t|\hat{\mathbf{b}}_1})^2 \cdot \frac{\alpha_t(i)}{\sum_{j=0}^{2^M-1} \alpha_t(j)} \\ &= \mathbf{c^2} \cdot \frac{(\mathbf{A}^t)'}{\sum \boldsymbol{\alpha}_1 \mathbf{A}^t} \cdot \boldsymbol{\alpha}'_1 - 2\mu_{x_t|\hat{\mathbf{b}}_1} \mathbf{c} \cdot \frac{(\mathbf{A}^t)'}{\sum \boldsymbol{\alpha}_1 \mathbf{A}^t} \cdot \boldsymbol{\alpha}'_1 + \mu^2_{x_t|\hat{\mathbf{b}}_1}, \end{aligned} \tag{9.14}$$

where the $\sum \boldsymbol{\alpha}_t$ denotes the sum of all elements of the vector $\boldsymbol{\alpha}_t$.

The advantage is that the expressions $\mathbf{c} \cdot (\mathbf{A}^t)'$, $\mathbf{c^2} \cdot (\mathbf{A}^t)'$, and $\sum \boldsymbol{\alpha}_1 \mathbf{A}^t$ are vectors of length 2^M, which need considerably less storage than the matrix $\mathbf{A}^t$ of size $2^M \times 2^M$ and they can be computed prior to recognition.

Similar expressions can be found for the second half of the burst, if we denote by P a matrix having a priori probabilities $P(\mathbf{b}^{(i)})$ on the main diagonal:

$$\mu_{x_t|\hat{\mathbf{b}}_T} = \sum_{i=0}^{2^M-1} \frac{c^{(i)} \beta_t(i) P(\mathbf{b}^{(i)})}{\sum_{j=0}^{2^M-1} \beta_t(j) P(\mathbf{b}^{(j)})} = \frac{\mathbf{c}(\boldsymbol{\beta_t}\mathbf{P})'}{\sum \boldsymbol{\beta_t}\mathbf{P}} = \frac{\mathbf{cPA}^{T-t}}{\sum \mathbf{PA}^{T-t}\boldsymbol{\beta_T}} \cdot \boldsymbol{\beta_T}' \tag{9.15}$$

$$\begin{aligned} \sigma^2_{x_t|\hat{\mathbf{b}}_T} &= \sum_{i=0}^{2^M-1} (c^{(i)} - \mu_{x_t|\hat{\mathbf{b}}_T})^2 \cdot \frac{\beta_t(i) P(\mathbf{b}^{(i)})}{\sum_{j=0}^{2^M-1} \beta_t(j) P(\mathbf{b}^{(j)})} \\ &= \frac{\mathbf{c^2PA}^{T-t}}{\sum \mathbf{PA}^{T-t}\boldsymbol{\beta_T}} \cdot \boldsymbol{\beta_T}' - 2\mu_{x_t|\hat{\mathbf{b}}_1} \frac{\mathbf{cPA}^{T-t}}{\sum \mathbf{PA}^{T-t}\boldsymbol{\beta_T}} \cdot \boldsymbol{\beta_T}' + \mu^2_{x_t|\hat{\mathbf{b}}_T} \end{aligned} \tag{9.16}$$

Here, the expressions $\mathbf{cPA}^{T-t}$, $\mathbf{c}^2\mathbf{PA}^{T-t}$, and $\sum \mathbf{PA}^{T-t}\boldsymbol{\beta_T}$ are vectors of size 2^M.

Thus, for the ETSI-AFE quantization scheme the memory requirements are reduced from $L \times 87040$, in the previous approach, to $L \times 6 \times (5 \cdot 2^6 + 2^5 +$

2^8)=$L \times 3648$. This is a dramatical reduction (23 times) achieved by combining the bit pattern posterior estimation and the feature vector posterior estimation into one step. Moreover, because $\boldsymbol{\alpha}_1$ is a vector of zeros except of a one at position s, its multiplication by another vector, e.g. $\mathbf{c^2} \cdot (\mathbf{A}^t)'$, results in simply selecting the s^{th} element of that vector. A similar operation can be done for multiplication by $\boldsymbol{\beta}_T$.

Table 9.5 gives the WERs on the Aurora 2 task assuming a DSR over IP setup and using the above approximation (and limited depth $L = 4$) in comparison to the baseline using the original forward-backward algorithm.

Table 9.5: WERs on the Aurora 2 task for DSR over IP using the lookup method, forward probabilities in the first half of burst and backward in the second.

Cond.	C4	C3	C2	C1	C0
UD-FB	3.09	1.66	0.98	0.90	0.86
UD-F & UD-B	3.16	1.79	1.03	0.90	0.86

9.2.4 Multi-resolution approach

A complexity reduction approach which can be used for both bit-error or packet-loss channels has been proposed in [106]. Since the computational complexity increases exponentially with the number of quantization bits it is obvious that a lower resolution would be beneficial regarding reduction of computation. However, the vector quantization scheme of the ETSI-AFE DSR standard [12] ensures a good trade-off between a limited channel bit rate and a loss of recognition accuracy due to quantization errors, thus, changing it is not an option.

The idea of the multi-resolution approach is to assume that the source emits features quantized at a lower resolution only within the error burst periods. The reason behind this thinking is that due to channel errors, the transmitted feature cannot be recovered at original resolution anyway, thus, a coarse representation should be sufficient. Note that the assumption of lower resolution during error bursts does not incur any modification of the standard. The proposed scheme is still fully compatible with the standard!

The following example might be useful in helping to understand this principle. Let us assume that the channel is completely unreliable so that the feature posterior $p(\mathbf{x}_t|\mathbf{y})$ equals the prior $p(\mathbf{x}_t)$, see also Section 4.3. This happens no matter which resolution was used for quantization. Consequently, using the original ETSI-AFE resolution in (6.21) yields the same result as using an extremely coarse quantization with one centroid, i.e. $M = 0$. Indeed in this case there is only one term in the sum of Eq. 6.21 and its bit pattern posterior is $P(\mathbf{b}^{(0)}|\hat{\mathbf{b}}_1^T) = 1$. The source model has only one state whose cluster conditioned probability density is the prior feature probability density $p(\mathbf{x}_t|\mathbf{b}_t^{(0)}) = p(\mathbf{x}_t)$. Using this in (6.21) we indeed deduce that $p(\mathbf{x}_t|\mathbf{y}) = p(\mathbf{x}_t)$, however, avoiding the FB algorithm at the original ETSI resolution.

Note that the channel transition probabilities required in the FB algorithm must be projected on the lower resolution space by:

$$P(\hat{\mathbf{b}}_t^{(k)}|\underline{\mathbf{b}}_t^{(n)}) = \sum_{i=0}^{2^M-1} P(\hat{\mathbf{b}}_t^{(k)}|\mathbf{b}_t^{(i)}) \cdot P(\mathbf{b}_t^{(i)}|\underline{\mathbf{b}}_t^{(n)}) \tag{9.17}$$

where $\underline{\mathbf{b}}_t^{(n)}$ denotes the n^{th} bit pattern of the lower resolution ($\underline{M}$) codebook, $P(\hat{\mathbf{b}}_t^{(k)}|\mathbf{b}_t^{(i)})$ are the channel transition probabilities of the original resolution bit pattern and $P(\mathbf{b}_t^{(i)}|\underline{\mathbf{b}}_t^{(n)})$ represents the probability of the sent bit pattern $\mathbf{b}_t^{(i)}$ when the corresponding lower resolution quantization index n (bit pattern $\underline{\mathbf{b}}_t^{(n)}$) is known. This latter term is assumed constant for those i's falling in the n^{th} lower resolution cluster and zero otherwise. In a packet loss scenario this computation can be saved, except for the burst begin and end ($t = 1$ and $t = T$) since the transition probabilities within the burst do not affect the bit pattern posterior.

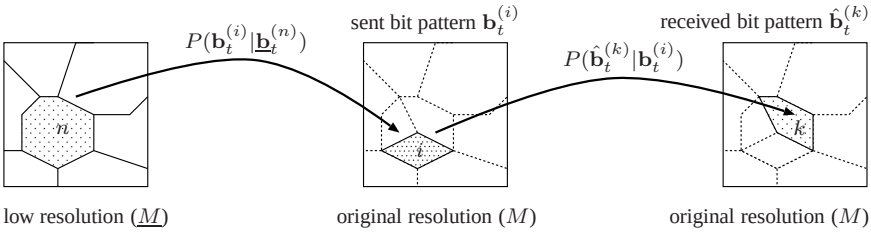

Figure 9.4: Channel transition probabilities from the low resolution feature to the received one at original resolution.

While the 0-bit resolution of the previous example is equivalent to marginalization, known to cause performance degradation, a finer resolution but still coarser than the original can provide computational savings at a cost of graceful degradation of performance. This is shown in Table 9.6 where the Aurora 2 task over an IP channel was carried out assuming various fixed resolutions during the loss bursts. E.g. in the first column, the string 6666658 denotes the resolution of the quantization scheme used in the ETSI-DSR standard: 6 bits for the first five subvectors $sv_1, \ldots, sv_5$, then 5 bits for sv_6 and 8 bits for the last subvector sv_7. For the other resolution settings, we trained correspondingly a vector quantizer for each subvector using the Generalized Lloyd Algorithm.

The second column gives the upper bound of the computational complexity per frame, computed as in Section 9.2.1 and cumulating the contribution of each subvector. Additionally, the computing time for error concealment per lost frame, measured on a workstation with 2.3 GHz Intel Xeon 5140, is given in the third column. Although the absolute values are highly dependent on the processor speed, they indicate a reduction of complexity close to the estimated one. Comparing the complexities of resolution 6666658 and 5555555 it can be seen that if all HMM states were connected, reduction of complexity by a factor of 9 would be possible without any performance loss. In practice, the computing time was reduced by a factor of 6.6.

At resolutions lower than 4 bits the performance starts to degrade, where the degradation is strongest for the worst channel model C4.

Table 9.6: WERs on the Aurora 2 task at various resolutions during loss bursts.

	$\mathcal{O}$	t[μs]	C1	C2	C3	C4
6666658	$\simeq 2^{17}$	337	0.90	0.98	1.68	3.13
6666657	$\simeq 2 \cdot 2^{15}$	190	0.90	0.98	1.73	3.19
6666656	$\simeq 6 \cdot 2^{13}$	148	0.90	1.00	1.73	3.13
5555555	$\simeq 7 \cdot 2^{11}$	51.0	0.90	1.00	1.76	3.21
4444444	$\simeq 7 \cdot 2^{9}$	17.5	0.90	1.02	1.83	3.19
3333333	$\simeq 7 \cdot 2^{7}$	6.4	0.90	1.07	1.80	3.37
2222222	$\simeq 7 \cdot 2^{5}$	3.2	0.90	1.27	2.59	4.93
1111111	$\simeq 7 \cdot 2^{3}$	1.93	0.90	2.36	7.67	13.14

Improved recognition accuracy can be achieved if a Markov model is used for both the clean static and dynamic components of the feature vector, i.e. the source model of Section 6.4.4. In the second experiment the improvement by using this augmented source model is evaluated. In order to limit complexity, only a coarse

quantization for the dynamic components was used: 3 bits for first-order derivatives (velocity) and 1 bit for second-order derivatives (acceleration). Still, the complexity increase is considerable. While for the quantization table of the static components only, the finest resolution, i.e. sv_7 with 8 bit quantization, resulted in a value of the complexity measure of 2^{17}, this is now increased to $2(2^{8+3+1})^2 = 2^{25}$ in the case of a source model for static and dynamic features. Note that the actual complexity is much lower since the HMM transition matrix is sparse at that resolution. While the number of bits for the dynamic features was kept fixed at 3 bit for delta and 1 bit for delta-delta, the resolution of the static components was successively decreased, as is indicated by the left column of Table 9.7. The results show that for resolutions down to 5 bits, dynamic features yield noticeable improvements. At resolutions of 4 bits and lower, the word accuracy is limited by the resolution of the static features, i.e. the augmented source model no longer has a pay off in increased word accuracy.

Table 9.7: WERs with the augmented source model (static and dynamic components): 3 bit delta, 1 bit delta-delta.

	$\mathcal{O}$	t[ms]	C1	C2	C3	C4
6666658	$\simeq 2^{25}$	64	0.90	0.93	1.53	2.54
6666657	$\simeq 2 \cdot 2^{23}$	30	0.90	0.94	1.56	2.62
6666656	$\simeq 7 \cdot 2^{21}$	22	0.90	0.93	1.53	2.60
5555555	$\simeq 7 \cdot 2^{19}$	7.3	0.90	0.97	1.61	2.71
4444444	$\simeq 7 \cdot 2^{17}$	2.3	0.90	0.98	1.73	3.17

It can be concluded that a considerable speed-up (up to 6 times), with hardly any degradation in word accuracy, can be achieved by simply assuming a lower quantization resolution of the received feature subvectors, i.e. $M = 5$. With the extended Markov source, which includes dynamic features (keeping $M = 5$ for static components, $M = 3$ for delta and $M = 1$ for delta-delta), the state space and thus computational complexity of the FB algorithm is considerably increased requiring as much time as the feature sampling period (10 ms). Note that the average computation time per frame is actually lower since the bit pattern posterior need to be evaluated by FB only for unreliable/lost frames, whereas for reliable ones it is a Dirac PDF.

The resolution at which the erroneous frames are to be processed can be either fixed or can be changed on-the-fly according to the burst length, as we did in [106]. In the latter case the variable resolution contributes to further reduction of

computational complexity. E.g. by processing the bursts up to length 8 at $M = 3$ and those longer at $M = 4$, the processing time per frame decreases from 17.5 (fixed $M = 4$, see Table 9.6) to 10 μs. This corresponds to a 30-times reduction of the processing time per frame at original resolution 6666658 (337μs).

9.3 Computation of observation log-likelihood

In an HMM based recognizer with diagonal covariance mixture densities, the computation of (4.39) is usually done in log-likelihood domain. Considering for simplicity one mixture component, the log-likelihood computation is usually implemented as:

$$\log p(\mathbf{x}_t|s_t) = -\frac{1}{2}\log\prod_{d=1}^{D} 2\pi\sigma_{s_t,d}^2 - \frac{1}{2}\sum_{d=1}^{D}\frac{(x_{t,d}-\mu_{s_t,d})^2}{\sigma_{s_t,d}^2} \tag{9.18}$$

with D denoting the feature vector dimensionality. Since the product in (9.18) does not depend on $\mathbf{x}_t$, its logarithm can be computed in advance for each state s_t and stored as part of the acoustic model. By doing so, the computational load reduces to the evaluation of the sum.

By applying the uncertainty decoding rule (4.48) the right hand side of (9.18) becomes:

$$-\frac{1}{2}log\prod_{d=1}^{D} 2\pi(\sigma_{s_t,d}^2+\sigma_{e_t,d}^2) - \frac{1}{2}\sum_{d=1}^{D}\frac{(\mu_{e_t,d}-\mu_{s_t,d})^2}{\sigma_{s_t,d}^2+\sigma_{e_t,d}^2} \tag{9.19}$$

Since the variance $\sigma_{e_t,d}^2$ changes according to instantaneous channel properties, the logarithm of the product and the sum in (9.19) have to be evaluated for each active state. The computation of the logarithm of the product can no longer be done in advance. In the experiments the computation of (4.48) took 1.4 times longer than (4.39). Note, however, that (4.48) needs to be computed only for unreliable features, i.e. within the error bursts. For the reliable features the simpler expression (4.39) can be used.

9.4 The impact of uncertainty on the acoustic search space

An unfavorable side effect of techniques which broaden the observation probability densities, such as weighted Viterbi recognition or uncertainty decoding, is that the search space increases considerably due to reduced discrimination capabilities between the word hypotheses in the presence of uncertain observations. As the observation probability of an unreliable feature tends to be the same for all model states, the beam pruning loses efficiency and the number of "active" states increases. This effect is illustrated in Figure 9.5 where the number of states is plotted over the time during an utterance. The curve PLI denotes the binary packet (feature) loss indicator which is zero for a lost feature vector. It can be observed that during periods without losses the number of states evaluated per frame using uncertainty decoding rule tends to be the same as that obtained in no-loss conditions. When a feature loss occurs, the uncertainty decoding requires to evaluate significantly more states than in no-loss conditions.

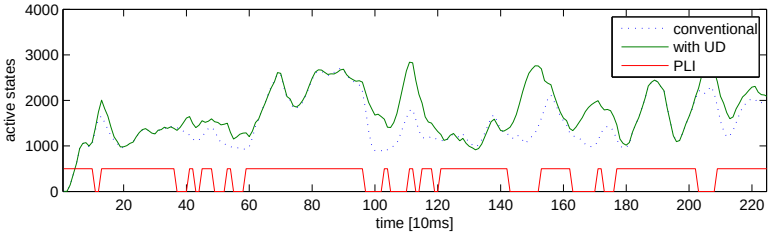

Figure 9.5: Number of active states vs. time in the conventional and uncertainty decoding of an utterance.

Table 9.8 shows the average number of active states per frame for the WSJ0 task during the **UD1** experiment of Section 7.3. The beam pruning threshold was kept constant. The last line shows the slowdown factor of the recognition, relative to decoding time in error-free condition, when the uncertainty decoding rule is employed for unreliable features. Under the most critical channel condition (C4) recognition was slowed down by a factor of 2.3. This is explained by the increased search space on one hand, and by the more intensive computation of (4.48) on the other.

Table 9.8: Average number of active states per frame for the WSJ0 task.

Condition	C0	C1	C2	C3	C4
Active states/frame	1035	1038	1098	1319	1516
slowdown factor	1.0	1.0	1.2	1.8	2.3

Chapter 10

Special cases of uncertainty decoding

This chapter is aimed at strengthening the focus of this work by studying how other decoder-based robustness techniques, namely the Missing Feature Technique and weighted Viterbi recognition (see Section 2.4) are related to uncertainty decoding. It is shown that they can be obtained as special cases of the more general uncertainty decoding rule proposed in this thesis.

10.1 Missing Feature Theory

According to MFT, a feature vector (or a component of it) is either reliable, in which case its contribution to the acoustic score, i.e. the observation probability, is computed as usual, or it is unreliable and does not contribute at all. There is no intermediate case in that the feature vector is partly unreliable, to be accounted for by an attenuated contribution to the score. The principle can be applied to each feature vector component. By separating the full vector $\mathbf{x}_t$ into reliable $\mathbf{x}_{\mathbf{r}t}$ and unreliable $\mathbf{x}_{\mathbf{u}t}$ subvectors, the observation probability can be expressed by Eq. 2.4.

In the context of DSR, the reliable feature vector components correspond to a delta shaped posterior probability density function:

$$p(\mathbf{x}_{\mathbf{r}t}|\mathbf{y}_1^T) = \delta(\mathbf{x}_{\mathbf{r}t} - \mathbf{y}_{\mathbf{r}t}), \tag{10.1}$$

where $\mathbf{y}_{\mathbf{r}t}$ are the observations deemed reliable at the channel output.

The posterior probability density function of unreliable feature vector components equals their a priori probability density function, as the observations are not

relevant (not informative) for the transmitted feature components:

$$p(\mathbf{x}_{\mathbf{u}t}|\mathbf{y}_1^T) = p(\mathbf{x}_{\mathbf{u}t}). \tag{10.2}$$

Replacing the posterior probability in observation probability for uncertainty decoding (4.36) yields:

$$\begin{aligned}
&\int_{\mathbf{x}_{\mathbf{r}t}} \int_{\mathbf{x}_{\mathbf{u}t}} \frac{p(\mathbf{x}_{\mathbf{r}t}|\mathbf{y}_1^T)}{p(\mathbf{x}_{\mathbf{r}t})} \cdot \frac{p(\mathbf{x}_{\mathbf{u}t}|\mathbf{y}_1^T)}{p(\mathbf{x}_{\mathbf{u}t})} p(\mathbf{x}_{\mathbf{r}t}, \mathbf{x}_{\mathbf{u}t}|s_t) d\mathbf{x}_{\mathbf{r}t} d\mathbf{x}_{\mathbf{u}t} && (10.3)\\
&= \int_{\mathbf{x}_{\mathbf{r}t}} \int_{\mathbf{x}_{\mathbf{u}t}} \frac{\delta(\mathbf{x}_{\mathbf{r}t} - \mathbf{y}_{\mathbf{r}t})}{p(\mathbf{x}_{\mathbf{r}t})} \cdot \frac{p(\mathbf{x}_{\mathbf{u}t})}{p(\mathbf{x}_{\mathbf{u}t})} p(\mathbf{x}_{\mathbf{r}t}, \mathbf{x}_{\mathbf{u}t}|s_t) d\mathbf{x}_{\mathbf{r}t} d\mathbf{x}_{\mathbf{u}t} && (10.4)\\
&= \int_{\mathbf{x}_{\mathbf{u}t}} \frac{p(\mathbf{y}_{\mathbf{r}t}, \mathbf{x}_{\mathbf{u}t}|s_t)}{p(\mathbf{y}_{\mathbf{r}t})} d\mathbf{x}_{\mathbf{u}t} && (10.5)\\
&\propto \int_{\mathbf{x}_{\mathbf{u}t}} p(\mathbf{y}_{\mathbf{r}t}, \mathbf{x}_{\mathbf{u}t}|s_t) d\mathbf{x}_{\mathbf{u}t} && (10.6)\\
&= p(\mathbf{y}_{\mathbf{r}t}|s_t) && (10.7)
\end{aligned}$$

Note that $p(\mathbf{y}_{\mathbf{r}t})$ is not state dependent and thus can be neglected, having same contribution for all HMM states. The last expression proves that, assuming binary reliability, uncertainty decoding and MFT result in the same observation probability, namely that of reliable components only.

10.2 Weighted Viterbi recognition

As we have seen in Section 2.4.2, a plethora of recipes have been proposed for the computation of the weighting coefficient γ_t employed in the weighted Viterbi recognition technique. Some of them are purely heuristic, e.g. [61] which uses a particular sigmoid function to map a Euclidian distance onto the weighting coefficient, or assuming an exponential decay of the weighting coefficient during the error bursts [66]. Most of these methods need parameter tuning and thus are prone to be suboptimal in an unknown real-world scenario. Other methods use training data to find a relationship between the weighting coefficient and some others exploit statistical properties such as the temporal auto-correlation [22, 66].

Since most of these methods lack an obvious probabilistic interpretation, we are looking here for certain approximations which would allow us to derive the

weighting coefficient γ_t from the mathematically well founded uncertainty decoding rule proposed in this work.

In the following, the same assumptions (1-3) of Section 4.3.3 are made and additionally, we assume acoustic models sharing a global diagonal covariance matrix $\Sigma_g = diag(\sigma^2_{g,d}), d = 1 \dots D$. Note that speech recognizers using a global covariance matrix are widely used to simplify the observation likelihood computation.

Under these assumptions, the observation probability of Eq. 4.48 yields:

$$\sum_{m=1}^{M} c_m \mathcal{N}(\boldsymbol{\mu}_e; \boldsymbol{\mu}_m, \boldsymbol{\sigma}_g^2 + \boldsymbol{\sigma}_e^2) \tag{10.8}$$

$$= \sum_{m=1}^{M} c_m \prod_{d=1}^{D} \mathcal{N}(\mu_{e,d}; \mu_{m,d}, \sigma^2_{g,d} + \sigma^2_{e,d}) \tag{10.9}$$

$$\propto \sum_{m=1}^{M} c_m \prod_{d=1}^{D} \exp\left[-\frac{1}{2}\frac{(\mu_{e,d} - \mu_{m,d})^2}{\sigma^2_{g,d} + \sigma^2_{e,d}}\right] \tag{10.10}$$

$$\propto \sum_{m=1}^{M} c_m \prod_{d=1}^{D} \left[\mathcal{N}(\mu_{e,d}; \mu_{m,d}, \sigma^2_{g,d})\right]^{\frac{\sigma^2_{g,d}}{\sigma^2_{g,d}+\sigma^2_{e,d}}} \tag{10.11}$$

For notational convenience we omitted the state index s_t and the time index t, the expression being evaluated for each feature vector and each state.

Comparing (10.11) with (2.7) leads to the conclusion that the appropriate weighting coefficient for each dimension d is given by:

$$\gamma_d = \frac{\sigma^2_{g,d}}{\sigma^2_{g,d} + \sigma^2_{e,d}}. \tag{10.12}$$

Note the similarity of the last expression to (2.6) in that the weighting coefficient was state dependent but averaged over the feature vector components.

Chapter 11

Summary and conclusions

This thesis addresses the channel robustness of a remote speech recognition system by using a novel decoding rule which takes into account the uncertainty in the received speech features. As the channel may be error prone, the clean features emitted by the source are not observable and must be reconstructed from the observed ones. Unlike point estimation methods, our approach takes advantage of both the optimally reconstructed clean features and information about the reliability of reconstruction. The proposed decoding rule is obtained by reformulating the classical Bayesian framework of speech recognition to carry out the classification with features observed at the communication channel output. Under certain assumptions this simply results in a modification of the observation probability computation, while the structure of the decoder, which is based on Viterbi search, remains unchanged.

This chapter summarizes our conclusions based on simulations of small- and medium-vocabulary remote speech recognition tasks using both bit and packet oriented transmission between a terminal and a recognition server. The major contributions made by this work are highlighted and some ideas for further research are suggested.

11.1 Summary of results

In the case of DSR over a GSM network exhibiting bit errors, a corrupted feature vector can be reconstructed accurately resulting in a small variance of the feature posterior. Consequently, point estimation techniques, such as MMSE, which neglect the estimation variance, achieve competitive performance levels. However, when the variance of the feature posterior is high, as is the case in DSR over

IP channels, a considerable improvement can be obtained by uncertainty decoding. Uncertainty decoding using the extended source model with dynamic feature components outperformed the competing error concealment techniques.

In the case of NSR using VoIP channels, the uncertainty decoding achieved drastic improvement over the performance of the same codec obtained by conventional recognition under moderate and poor channel conditions. This was noticed for all investigated speech codecs.

Using the uncertainty decoding rule comes, however, at the expense of increased computational complexity. This is due to additional operations needed for the computation of the observation probability on the one hand, and due to the expansion of the acoustic search space caused by the reduced discrimination capabilities of unreliable features.

11.2 Contributions

The major contributions of this work are summarized below:

- We proposed a novel uncertainty decoding rule which takes advantage of correlation among successive feature vectors. Taking into account all, i.e. also past and future observable values, reduces the uncertainty about the clean feature compared to conditioning the feature posterior only on the currently observed vector. To the best of our knowledge, the uncertainty decoding rules proposed by other authors in the context of noise robust speech recognition do not employ correlation in this form.

- Our decoding rule is derived from the very first principle of speech recognition for minimizing the word error rate and avoids any heuristic. It delivers therefore the optimal solution, provided that the simplifying approximations hold.

- We applied the decision rule for decoder-based error concealment for distributed speech recognition and network speech recognition. In both cases the inter-frame correlation turned out to be a powerful knowledge source to overcome temporarily poor channel conditions.

- We proposed an approach to estimating data reliability based on media-specific FEC which relies on the intrinsic redundancy of source. This allows

the employment of uncertainty decoding in situations where the soft-output from the channel decoder is not available.

- We showed that MFT and weighted Viterbi decoding are special cases of uncertainty decoding.

11.3 Suggestions for further research

The proposed decoding rule is applicable wherever there is a mismatch between training and testing conditions, e.g. due to environmental noise, channel and speaker variations. However, it may turn out to be a major challenge to compute the key element of the decoding rule, the clean speech feature posterior given all observed feature vectors, for a particular distortion scenario.

While noise robustness techniques based on uncertainty decoding rules which neglect the inter-frame correlation have already been proposed, e.g. in [70, 71, 65, 64], it would be interesting to study the extent to which the noise robustness can benefit from using the decoding rule proposed in this work.

Another topic worth exploring is the exploitation of inter-frame correlation in a clean scenario, where the observed features are not degraded. It is well-known that the conditional independence assumption, i.e. neglecting inter-frame correlation, represents a major drawback of HMM-based speech recognition. The Bayesian network of Figure 4.3 allows for using inter-frame correlation only if the observation is uncertain. In this case the uncertainty in observation is reduced by considering the dependency of neighboring frames. In an error-free environment the uncertainty in observations is already minimal so that it cannot be further reduced by knowledge of neighboring frames. It would be interesting to explore whether the model proposed here can be also extended to relax the conditional independence assumption in a clean scenario.

Bibliography

[1] V. Weerackody, W. Reichl, and A. Potamianos, "An error-protected speech recognition system for wireless communications," *IEEE Trans. Wireless Communications*, vol. 1, no. 2, pp. 282–291, 2002.

[2] Hans-Guenter Hirsch, "The influence of speech coding on recognition performance in telecommunication networks," in *Proc. of ICSLP 2002*, 2002.

[3] J. Huerta and R. Stern, "Speech recognition from GSM codec parameters," in *Proc. of ICSLP-98*, 1998.

[4] R. Hong Kook Kim; Cox, "Bitstream-based feature extraction for wireless speech recognition," in *Proc. of ICASSP,*, Oct. 2000.

[5] Peláez-Moreno C., Gallardo-Antolín A., Gómez-Cajas D. F., and Díaz de María F., "A comparison of front-ends for bitstream-based ASR over IP," *Speech Communication*, vol. 86, pp. 1502–1508, 2006.

[6] Imre Kiss, "A comparison of distributed and network speech recognition for mobile communication systems," in *Proc. of ICSLP 2000*, 2000.

[7] David Pearce, "Robustness to Transmission Channel - the DSR Approach," in *COST278 and ISCA Tutorial and Research Workshop on Robustness Issues in Conversational Interaction University of East Anglia, Norwich, UK*, 2004.

[8] ES.201.108, "Speech processing, Transmission and Quality aspects (STQ); Distributed speech recognition;Front-end feature extraction algorithm; Compression algorithms," *ETSI*, April 2000.

[9] H. Hirsch and D. Pearce, "The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *ISCA ITRW Workshop ASR2000, Paris, France*, 2000.

[10] David Pearce, "Enabling New Speech Driven Services for Mobile Devices: An overview of the ETSI standards activities for Distributed Speech Recognition Front-ends," in *Applied Voice Input/Output Society Conference (AVIOS2000), San Jose, CA*, May 2000.

[11] David Pearce, "Developing the ETSI Aurora advanced distributed speech recognition front-end what next ?," in *ASRU 2001, Dec 2001*, 2001.

[12] ES.202.050, "Speech processing, Transmission and Quality aspects (STQ); Distributed speech recognition; Advanced front-end feature extraction algorithm; Compression algorithms," *ETSI*, Oct 2002.

[13] Duncan Macho, Laurent Mauuary, Bernhard Noé, Yan Ming Cheng, Doug Ealey, Denis Jouvet, Holly Kelleher, David Pearce, and Fabien Saadoun, "Evaluation of a noise-robust DSR front-end on Aurora databases," in *Proc. of ICSLP2002*, 2002.

[14] ES.202.212, "Speech processing, Transmission and Quality aspects (STQ); Distributed speech recognition; Extended advanced front-end feature extraction algorithm; Compression algorithms; Back-end speech reconstruction algorithm.," *ETSI*, Nov 2003.

[15] Q. Xie and D. Pearce, "RTP Payload Formats for European Telecommunications Standards Institute (ETSI) ES 202 050, ES 202 211, and ES 202 212 Distributed Speech Recognition Encoding," *Internet Official Protocol Standards*, May 2005.

[16] TS.100.909.v8.7.1, "Digital cellular telecommunications system (Phase 2+); Channel coding (3GPP TS 05.03 version 8.7.0 Release 1999)," *ETSI*, Apr. 2003.

[17] P. Vary and R. Martin, *Digital Speech Transmission. Enhancement, Coding and Error Concealment*, John Wiley, 2006.

[18] Lamia Karray, Chafic Mokbel, and Jean Monne, "Solutions for robust speech/non-speech detection in wireless environment," in *Proc. of Interactive Voice Technology for Telecommunications Applications, Torino, Italy September 29-30, 1998*, 1998.

[19] A.M. Gómez, A.M. Peinado, V. Sánchez, and A.J. Rubio, "Recognition of coded speech transmitted over wireless channels," *IEEE Trans. on Wireless Communications*, vol. 5, No. 9, pp. 2555–2562, 2006.

[20] Antonio M. Peinado and Jose C. Segura, *Speech Recognition over Digital Channels*, John Wiley & Sons Ltd., 2006.

[21] Z.-H. Tan, P. Dalsgaard, and B. Lindberg, "Automatic speech recognition over error-prone wireless networks," *Speech Communication*, vol. 47, no. 1-2, pp. 220–242, Sep.-Oct. 2005.

[22] A. Bernard and A. Alwan, "Low-bitrate distributed speech recognition for packet-based and wireless communication," *IEEE Trans. Speech and Audio Processing*, vol. 10, no. 8, pp. 570–579, 2002.

[23] J. Hagenauer and P. Höher, "A Viterbi algorithm with soft-decision outputs and its applications," in *Proc. of IEEE Global Communications Conference, Dallas*, 1989.

[24] P. Robertson, P. Hoeher, and E. Villebrun, "Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo-decoding," in *European Trans. Telecomm*, 1997.

[25] A. Potamianos and V. Weerackody, "Soft-feature decoding for speech recognition over wireless channels," in *Proc. of ICASSP, Salt Lake City, Utah*, 2001.

[26] Zheng-Hua Tan and Paul Dalsgaard, "Channel error protection scheme for distributed speech recognition," in *Proc. ICSLP, September 16-20, 2002 Denver, Colorado, USA*, 2002.

[27] C. Boulis, M. Ostendorf, E:A. Riskin, and S. Otterson, "Graceful degradation of speech recognition performance over packet-erasure networks," *IEEE Trans. Speech and Audio Processing*, vol. 10, no. 8, pp. 580–590, Nov. 2002.

[28] V. K. Goyal, "Multiple description coding: compression meets the network," *IEEE Signal Processing Magazine*, vol. 18, pp. 74–93, 2001.

[29] S. Deering and R. Hinden, "RFC 2460 - Internet Protocol, Version 6 (IPv6) Specification," *Internet Official Protocol Standards*, 1998.

[30] M. Y. Kim and W.B. Kleijn, "Comparison of transmitter-based packet-loss recovery techniques for voice transmission," in *Proc. of ICSLP 2004*, 2004.

[31] X. Zhong, J. Arrowood, A. Moreno, and Clements M., "Multiple description coding for recognizing voice over IP," in *Proc. of IEEE Digital Signal Processing Workshop*, 2002.

[32] N. Srinivasamurthy, A. Ortega, and S. Narayanan, "Efficient scalable speech compression for scalable speech recognition," in *Proc. of Eurospeech 2001, Aalborg, Denmark*, 2001.

[33] Zheng-Hua Tan, Paul Dalsgaard, and Borge Lindberg, "Adaptive multi-frame-rate scheme for distributed speech recognition based on a half frame-rate front-end," in *IEEE 7th Workshop on Multimedia Signal Processing*, 2005.

[34] N. Farvardin and V. Vaishampayan, "On the performance and complexity of channel-optimized vector quantizers," *IEEE Transactions on Information Theory*, vol. 37, 1991.

[35] Mikael Skoglund, "On channel-constrained vector quantization and index assignment for discrete memoryless channels," *IEEE Transactions on Information Theory*, vol. 45, 1999.

[36] J. Hagenauer, "Source controlled channel decoding," *IEEE Transactions on Communications*, vol. 43, 1995.

[37] Tim Fingscheidt, Thomas Hindelang, Richard V. Cox, and Nambi Seshadri, "Joint source-channel (de)coding for mobile communications," *IEEE Transactions on Communications*, vol. 50, pp. 200–212, 2002.

[38] Tim Fingscheidt and Olaf Scheufen, "Robust GSM speech decoding using the channel decoder's soft output," in *Proc. of EUROSPEECH*, 2001.

[39] T. Fingscheidt and P. Vary, "Softbit speech decoding: a new approach to error concealment," *IEEE Trans. Speech and Audio Processing*, vol. 9, no. 3, pp. 1–11, 2001.

[40] E. A. Riskin, C. Boulis, and M. Otterson, S. Ostendorf, "Graceful degradation of speech recognition performance over lossy packet networks," in *Proc. of Eurospeech 2001*, 2001.

[41] A. B. James and B. P. Milner, "An analysis of interleavers for robust speech recognition in burst-like packet loss," in *Proc. of ICASSP*, 2004.

[42] Z.-H. Tan, P. Dalsgaard, and B. Lindberg, "A subvector-based error concealment algorithm for speech recognition over mobile networks," in *Proc. of ICASSP, Montreal*, 2004.

[43] C. Perkins, O. Hodson, and V. Hardman, "A survey of packet loss recovery techniques for streaming audio," *IEEE Network*, vol. 12, pp. 40–48, 1998.

[44] Z.-H. Tan, P. Dalsgaard, and B. Lindberg, "Partial splicing packet loss concealment for distributed speech recognition," *IEE Electron. Lett.*, vol. 39, pp. 1619–1620, 2003.

[45] B. Milner and S. Smnani, "Robust speech recognition over IP networks," in *Proc. of ICASSP, Istanbul*, 2000.

[46] A.M. Peinado, V. Sanchez, J.L. Perez-Cordoba, and A. de la Torre, "HMM-based channel error mitigation and its application to distributed speech recognition," *Speech Communication*, vol. 41, no. 6, pp. 549–561, Nov. 2003.

[47] A.M. Gómez, A.M. Peinado, V. Sánchez, and A.J. Rubio, "A source model mitigation technique for distributed speech recognition over lossy packet channels," in *Proc. of EUROSPEECH*, 2003.

[48] Bhiksha Raj, *Reconstruction of incomplete spectrograms for robust speech recognition*, Ph.D. thesis, Department of Electrical and Computer Engineering Carnegie Mellon University, 2000.

[49] Angel M. Gómez, Antonio M. Peinado, Victoria Sánchez, Ben P. Milner, and Antonio J. Rubio, "Statistical-based reconstruction methods for speech recognition in ip networks," in *COST278 and ISCA Tutorial and Research Workshop (ITRW) on Robustness Issues in Conversational Interaction University of East Anglia, Norwich, UK*, 2004.

[50] R. Haeb-Umbach and V. Ion, "Soft features for improved distributed speech recognition over wireless networks," in *Proc. of ICSLP, Jeju, Korea*, 2004.

[51] L. Rabiner and B.H. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, 1993.

[52] R. Haeb-Umbach and V. Ion, *"Error Concealment", in Automatic Speech Recognition on Mobile Devices and over Communication Networks*, Springer, London, 2008.

[53] S. Ahmad and Volker Tresp, "Some solutions to the missing feature problem in vision," *Advances in Neural Information Processing Systems*, 1993.

[54] M. Cooke, P. Green, L. Josifovski, and A. Vizio, "Robust automatic speech recognition with missing and unreliable acoustic data," *Speech Communication*, vol. 34, pp. 267–285, 2001.

[55] T. Endo, S. Kuroiwa, and S. Nakamura, "Missing feature theory applied to robust speech recognition over IP networks," in *Proc. of EUROSPEECH, Geneva, Switzerland*, 2003.

[56] Alastair James, Ángel Gómez, and Ben Milner, "A comparison of packet loss compensation methods and interleaving for speech recognition in burst-like packet loss," in *Proc. of Interspeech 2004 - ICSLP*, 2004.

[57] A. Cardenal-López, L. Docío-Fernández, and C. García-Mateo, "Soft Decoding Strategies for Distributed Speech Recognition over IP Networks," in *Proc. of ICASSP, Montreal*, 2004.

[58] V. Ion and Reinhold Haeb-Umbach, "An inexpensive packet loss compensation scheme for distributed speech recognition based on soft-features," in *In Proc. ICASSP 2006*, 2006.

[59] V. Ion and R. Haeb-Umbach, "Comparison of decoder-based transmission error compensation techniques for distributed speech recognition," in *Proc. of ITG-Fachtagung Sprachkommunikation 2006, Kiel*, 2006.

[60] Nestor Becerra Yoma, Fergus R. McInnes, and Mervyn A. Jack, "Weighted viterbi algorithm and state duration modelling for speech recognition in noise," in *Proc. of ICASSP 98*, 1998.

[61] A. Bernard and A. Alwan, "Joint channel decoding - Viterbi recognition for wireless applications," in *Proc. of EUROSPEECH, Aalborg, Denmark*, 2001.

[62] B. Delaney, "Increased robustness against bit errors for distributed speech recognition in wireless environments," in *Proc. of ICASSP 2005*, 2005.

[63] N.B. Yoma and M. Villar, "Speaker verification in noise using a stochastic version of the weighted Viterbi algorithm," *IEEE Transactions on Speech and Audio Processing*, vol. 10, pp. 158–166, 2002.

[64] L. Deng, J. Droppo, and A. Acero, "Exploiting variances in robust feature extraction based on a parametric model of speech distortion," in *Proc. of ICSLP, Denver, USA*, 2002.

[65] J. Droppo, A. Acero, and L. Deng, "Uncertainty decoding with SPLICE for noise robust speech recognition," in *Proc. of ICASSP, Orlando, Florida, May 2002*.

[66] A. Cardenal-López, C. García-Mateo, and L. Docío-Fernández, "Weighted Viterbi decoding strategies for distributed speech recognition over IP networks," *Speech Communication*, vol. 48, Issue 11, pp. 1422–1434, November 2006.

[67] Antonio M. Peinado, Victoria Sánchez, José L. Pérez-Córdoba, and Antonio J. Rubio, "Efficient MMSE-based channel error mitigation techniques. application to distributed speech recognition over wireless channels," *IEEE Transactions on Wireless Communications*, vol. vol 4, no. 1, 2005.

[68] Q. Huo and C-H. Lee, "A Bayesian predictive approach to robust speech recognition," *IEEE Trans. Speech and Audio Processing*, vol. 8, no. 8, pp. 200–204, 2000.

[69] A. Nadas, "Optimal solution of a training problem in speech recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 33, pp. 326–329, 1985.

[70] M. J. F. Gales, *Model-based techniques for noise robust speech recognition*, Ph.D. thesis, University of Cambridge, 1995.

[71] Trausti T. Kristjansson and Brendan J. Frey, "Accounting for uncertainty in observations: A new paradigm for robust automatic speech recognition," in *Proc. of ICASSP*, 2002.

[72] H. Liao and M. J. F. Gales, "Joint uncertainty decoding for noise robust speech recognition," in *Proc. of Interspeech, Lisbon, Portugal, Sept. 2005*.

[73] J. Barker, M. Cooke, and D. Ellis, "Decoding speech in the presence of other sources," *Speech Communication*, vol. 45, pp. 5–25, 2005.

[74] Andrew Morris, Jon Barker, and Hervé Bourlard, "From missing data to maybe useful data: soft data modelling for noise robust ASR," *Proc. WISP*, vol. 06, 2001.

[75] 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. Speech and Audio Processing*, vol. 13, no. 3, pp. 412–421, 2005.

[76] V. Stouten, H. van Hamme, and P. Wambacq, "Model-based feature enhancement with uncertainty decoding for noise robust ASR," *Speech Communication*, vol. 48, Issue 11, Nov. 2006.

[77] H. Liao and M. J. F. Gales, "Issues with uncertainty decoding for noise robust speech recognition," in *Proc. of Interspeech, Pittsburgh, PA, USA, September 17-21*, 2006.

[78] COST 207, "Digital land mobile radio communication - final report," Tech. Rep., Office for Official Publications of the European Communities, Luxembourg, 1989.

[79] CoWare Design Systems, "CoWare SPW 4," *http://www.coware.com/products/spw4.php*, 2004.

[80] E.N. Gilbert, "Capacity of a burst-noise channel," *The Bell System Technical Journal*, Sep. 1960.

[81] E. O. Elliot, "Estimates of error rates for codecs on burst-noise channels," *Bell Syst. Tech. J.*, vol. 42, pp. 1977–1997, 1963.

[82] H. S. Wang and N. Moayeri, "Finite state Markov channel - a useful model for radio communication channels," *IEEE Transactions on Vehicular Technology*, vol. 44, pp. 163–171, 1995.

[83] S. Haykin, *Communication systems*, Wiley, 2000.

[84] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Information Theory*, vol. 10, pp. 284–287, March 1974.

[85] V. Ion and R. Haeb-Umbach, "A unified probabilistic approach to error concealment for distributed speech recognition," in *Proc. of Interspeech, Lisbon*, 2005.

[86] Henning Schulzrinne, Stephen L. Casner, Ron Frederick, and Van Jacobson, "RTP: A transport protocol for real-time applications," *Internet-Draft ietf-avt-rtp-new-01.txt*, 1998.

[87] J. Kurose and K. Ross, *Computer network: a top-down approach featuring the internet,* Addison-Wesley, 2003.

[88] J. Bolot, S. Fosse-Parisis, and D. Towsley, "Adaptive FEC-based error control for interactive audio in the internet," in *Proc. of IEEE Infocom 99*, 1999.

[89] B. Milner and A. James, "An analysis of packet loss models for distributed speech recognition," in *In Proc. of ICSLP 2004. Jeju Island, Korea*, 2004.

[90] V. Ion and R. Haeb-Umbach, "Uncertainty decoding for distributed speech recognition over error-prone networks," *Speech Comm.*, vol. 48, pp. 1435–1446, 2006.

[91] L. Docío-Fernández and C. García-Mateo, "Distributed speech recognition over IP networks on the Aurora 3 database," in *Proc. of ICSLP, Denver, USA*, 2002.

[92] ETSI TIPHON TS 101-329-5, "QoS measurements for Voice over IP," *Technical Report*, 2000.

[93] Yajnik M., Moon S., Kurose J., and Towsley D., "Measurement and modelling of the temporal dependence in packet loss," *In Proc. of IEEE Infocom*, pp. 345–352, 1999.

[94] Matthias Botte, "Modellierung und Aufbau eines verteilten Spracherkennungssystems basierend auf einer RTP-Netzwerkkommunikation," M.S. thesis, University of Paderborn, Department of Communications Engineering, 2006.

[95] Network Working Group, "The Lightweight User Datagram Protocol (UDP-Lite)," *IETF RFC 3828 / RFC3828*, 2004.

[96] V. Ion and R. Haeb-Umbach, "Improved source modeling and predictive classification for channel robust speech recognition," in *Proc. of Interspeech, Pittsburgh*, 2006.

[97] Cambridge University Engineering Department, "The hidden markov model toolkit (HTK)," http://htk.eng.cam.ac.uk/.

[98] D. Paul and J. Baker, "The design for the Wall Street Journal based CSR corpus," in *Proc. of the workshop on Speech and Natural Language, Harriman, New York*, Feb. 1992, pp. 357–362.

[99] P.C. Woodland and S. J. Young, "The HTK tied-state continuous speech recogniser," in *Proc. of Eurospeech, Lisbon, pages 2207-2210*, 1993.

[100] ITU-T, "Recommendation G.711. Pulse code modulation (PCM) of voice frequencies," *ITU-T*, 1988.

[101] ITU-T, "Recommendation G.729/Annex A - Reduced Complexity 8 kbit/s CS-ACELP Speech Codec," *ITU-T*, 1996.

[102] ITU-T, "Recommendation G.723.1 - Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s," *ITU-T*, 1996.

[103] Jari Turunen, Pekka Loula, and Tarmo Lipping, "Assessment of objective voice quality over best-effort networks," *Computer Communications*, vol. 28, pp. 582–588, 2004.

[104] ITU-T, "Recommendation G.711 Appendix I, A high quality low-complexity algorithm for packet loss concealment with G.711," *ITU-T*, 1999.

[105] I. Mitrani, *Probabilistic modelling*, Cambridge University Press, 1998.

[106] V. Ion and R. Haeb-Umbach, "Multi-resolution soft features for channel-robust distributed speech recognition," in *Proc. of Interspeech 2007, Antwerp*, 2007.

List of own publications

[HI04] R. Haeb-Umbach and V. Ion, "Soft features for improved distributed speech recognition over wireless networks," in *Proc. of ICSLP, Jeju, Korea*, 2004.

[IH05a] V. Ion and R. Haeb-Umbach, "A comparison of soft-feature distributed speech recognition with candidate codecs for speech enabled mobile services," in *Proc. of ICASSP, Philadelphia*, 2005.

[IH05b] ——, "A unified probabilistic approach to error concealment for distributed speech recognition," in *Proc. of Interspeech, Lisbon*, 2005.

[IH06a] ——, "Comparison of decoder-based transmission error compensation techniques for distributed speech recognition," in *In Proc. of ITG-Fachtagung Sprachkommunikation, Kiel*, 2006.

[IH06b] ——, "Improved source modeling and predictive classification for channel robust speech recognition," in *Proc. of Interspeech, Pittsburgh*, 2006.

[IH06c] ——, "Uncertainty decoding for distributed speech recognition over error-prone networks," *Speech Comm.*, vol. 48, pp. 1435–1446, 2006.

[IH06d] ——, "An inexpensive packet loss compensation scheme for distributed speech recognition based on soft-features," in *In Proc. ICASSP*, 2006.

[IH07] ——, "Multi-resolution soft features for channel-robust distributed speech recognition," in *In Proc. of Interspeech, Antwerp*, 2007.

[IH08] ——, "A novel uncertainty decoding rule with applications to transmission error robust speech recognition," *Accepted for publication in IEEE Trans. on Audio, Speech and Language Processing*, 2008.

[TL08] Z.-H. Tan and B. Lindberg, Eds., *"Error Concealment", in Automatic Speech Recognition on Mobile Devices and over Communication Networks*. Springer, London, 2008.