Multi-stream Processing for Noise Robust Speech Recognition

THÈSE N^o 3508 (2006)

presentée a la Facultè des sciences et techniques de l'ingènieur

École Polytechnique Fédérale de Lausanne

pour l'obtention du grade de docteur ès sciences

par

HEMANT MISRA

Bachelor of Engineering in Electronics and Communication,
Sardar Vallabhbhai National Institute of Technology, Surat, India
and
Master of Science by Research in Electrical Engineering,
(Thesis title: Spectral Mapping as a Feature for Speaker Verification)
Indian Institute of Technology, Madras, India

Thesis committee members:

Prof. Touradj Ebrahimi, Président du jury, EPFL, Switzerland
Prof. Hervé Bourlard, Directeur de thèse, IDIAP/EPFL , Switzerland
Prof. Roger K. Moore, Rapporteur, University of Sheffield, UK
Prof. Simon King, Rapporteur, University of Edinburgh, UK
Prof. Drygajlo Andrzej, Rapporteur, EPFL, Switzerland

Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland
March 2006.

To

Late Smt. Shanti Devi Mishra, my Ammaji

and

Late Shri. P. S. Misra, my Papa

Abstract

In this thesis, the framework of multi-stream combination has been explored to improve the noise robustness of automatic speech recognition (ASR) systems. The central idea of multi-stream ASR is to combine information from several sources to improve the performance of a system. The two important issues of multi-stream systems are which information sources (feature representations) to combine and what importance (weights) be given to each information source.

In the framework of hybrid hidden Markov model/artificial neural network (HMM/ANN) and Tandem systems, several weighting strategies are investigated in this thesis to merge the posterior outputs of multi-layered perceptrons (MLPs) trained on different feature representations. The best results were obtained by inverse entropy weighting in which the posterior estimates at the output of the MLPs were weighted by their respective inverse output entropies.

In the second part of this thesis, two feature representations have been investigated, namely pitch frequency and spectral entropy features. The pitch frequency feature is used along with perceptual linear prediction (PLP) features in a multi-stream framework. The second feature proposed in this thesis is estimated by applying an entropy function to the normalized spectrum to produce a measure which has been termed spectral entropy. The idea of the spectral entropy feature is extended to multi-band spectral entropy features by dividing the normalized full-band spectrum into sub-bands and estimating the spectral entropy of each sub-band. The proposed multi-band spectral entropy features were observed to be robust in high noise conditions. Subsequently, the idea of embedded training is extended to multi-stream HMM/ANN systems.

To evaluate the maximum performance that can be achieved by frame-level weighting, we investigated an “oracle test”. We also studied the relationship of oracle selection to inverse entropy weighting and proposed an alternative interpretation of the oracle test to analyze the complementarity of streams in multi-stream systems.

The techniques investigated in this work gave a significant improvement in performance for clean as well as noisy test conditions.

Version abrégée

L'idée centrale des systèmes à flux multiples est de combiner plusieurs sources d'information pour améliorer la performance finale d'un système. Cette thèse explore la combinaison de flux multiples pour améliorer la résistance au bruit d'un système de reconnaissance automatique de la parole (ASR). Deux directions complémentaires sont considérées : quel flux d'information utiliser (type de représentation des données), et quel importance relative donner à chaque flux (un poids pour chaque flux d'information).

Dans le cadre de la reconnaissance de la parole avec systèmes hybrides chaîne de Markov cachée/réseau neuronal (HMM/ANN) d'une part, et systèmes Tandem d'autre part, cette thèse propose trois stratégies pour définir les poids relatifs. Un poids est attribué à chaque perceptron multi-couches (MLPs), et chaque perceptron est entraîné sur un flux d'information différent. Les deux stratégies "postérieure maximum" (MP) et "entropie inverse" définissent les poids à partir des probabilités *a posteriori* estimées par chaque MLP. La troisième stratégie "vraisemblance maximale" (ML) définit les valeurs des poids de façon à maximiser la vraisemblance des données de test. Les expériences de reconnaissance montrent que la stratégie d'entropie inverse conduit aux meilleurs résultats.

Cette thèse considère aussi deux types de flux d'information : fréquence de timbre et entropie spectrale. La fréquence de timbre est liée au signal d'excitation du conduit vocal. Elle est ici utilisée en concaténation avec les informations de prédiction linéaire perceptuelle (PLP). L'entropie spectrale est l'entropie du spectre normalisé. Une extension "bandes multiples" de l'entropie spectrale est proposée : le spectre normalisé est divisé en sous-bandes, et l'entropie est estimée dans chaque sous-bande. Les expériences de reconnaissance montrent que l'entropie spectrale est robuste aux conditions fortement bruitées.

Pour connaître l'intégralité du bénéfice potentiel offert par les systèmes à flux multiples, nous avons étudié un "test d'oracle". Ce test indique la performance maximale qui peut être obtenue par les différentes stratégies de combinaison. Nous avons ensuite étudié les relations entre le choix de l'oracle et la stratégie d'entropie inverse. Ceci a conduit à une autre interprétation du test d'oracle, qui permet d'analyser la complémentarité des flux dans les systèmes multi-flux. Enfin, l'idée d'entraînement incorporé a été étendue aux systèmes HMM/ANN à flux multiples.

Les techniques étudiées dans cette thèse, à savoir la stratégie de combinaison par entropie inverse, l'entropie spectrale et l'entraînement incorporé à flux multiples, apportent une amélioration significative aux performances de reconnaissance. Ceci est vérifié aussi bien en conditions non-bruitées que bruitées. On en conclut donc que les techniques proposées rendent plus robustes les systèmes de reconnaissance.

Contents

List of Figures

List of Tables

Acknowledgement

To begin with, I would like to thank my supervisor, Prof. Hervé Bourlard, for giving me a chance to do my thesis at IDIAP. His guidance and help were readily available almost 24-hours a day throughout my stay at IDIAP. He has provided an excellent structure at IDIAP to carry out state-of-the-art research. Also, during the last 2 years of my Ph.D., I got advise and comments from Prof. Hynek Hermansky which improved my thesis considerably. Dr. Samy Bengio was a great help when it came to clarifying things related to machine-learning or making right assumptions in the ML weighting.

I am grateful to Dr. Andrzej Drygajlo, Prof. Roger Moore and Dr. Simon King for accepting to be a part of my thesis review committee. Their comments and suggestions have improved the presentation of this thesis significantly.

The journey till this point would not have been possible without the support of several people who helped me during the thick and thin of life. I am indebted to Prof. B. Yegnanarayana, who has been my mentor for last several years. Special thanks to Dr. Samudravijaya, Dr. Hema Murthy and Dr. Chandrasekhar who helped me understand speech recognition from scratch. My colleagues at IIT Madras, Anjani, Jyotsna, Karna, Kishore, Kutty Murthy, Manish, Murthy Bhaiya, Pragathi, Siva and Tamil, had a profound effect on my thinking process. I would also like to thank my colleagues at SSTIL who helped me with my career growth.

It was a pleasure to work with Andrew, Artem, Astrid, Bertrand, Cuong, Darren, Guillaume, Hamed, Hari, Iain, Itsik, Jitendra (Ajmera), Jithendra (Vepa), Joel, John, Katrin, Michael, Momo, Sunil, Todd, Victoria and Vivek, the members of speech group at some point during the last few years. Sincere thanks to Mathew and Ikbal for their useful comments on the thesis and to Guillaume for the French abstract. My other friends at IDIAP, including Agnes, Christos, Datong, Haiyan, Johnny and Mael, made my stay at IDIAP a memorable one.

Special thanks to Norbert, Frank and Tristan for all the system related help. Their help in recovering the data after the big crash cannot be expressed in words. The administrative staff of IDIAP, Nadine, Sylvie and Mr. Dal-Pont, were a great help in taking care of the administrative issues inside and outside IDIAP.

Francina, Jasmine, Jahnavi, Partha, Sarangi family and Shakila, the Indian friends in Switzerland, were a good source of encouragement and entertainment. Thanks to Christie, D'Souza family, Marco-Lousila family and Carruzzo family for their help and support.

My family members, Mummy (LKO), Papa, Mummy (CHD), Manu, Rakhee, Prashant, Sumala, Rahul, Rohit, Vedika, Aryan and Vedant, were a constant source of inspiration during the whole Ph.D. Thank you all for your love and support.

"Behind every successful endeavour there is a woman" is true in the case of my PhD as well. Thank you, Bhavna, for all the things you did for me in these last few years (including spell checks of the thesis!). We (Bhavna and myself) expect that our baby (who is due next month), will be a born speech expert :)

Hemant Misra

Chapter 1

Introduction

Speech is a natural means of communication among human beings to convey the intended message. Apart from the message content, the speech signal also carries variabilities such as speaker characteristics, information about the environment in which it is produced and the properties of the channel through which it propagates. In spite of these additional variabilities present in the signal, human beings are able to extract the message content of the speech without much difficulty.

Automatic speech recognition (ASR) is the task of transforming the intended message content of the speech into text with the help of a machine. In the ASR task, variabilities due to different speakers, environment and channels degrade the performance of the system and are undesirable. The goal of ASR is to have speech as a medium of interaction between man and machine and it is desired that an ASR system is robust to these unwanted variabilities.

1.1 Objective of the Work

Typically, the performance of an ASR system drops in presence of noise (Hermansky and Morgan, 1994; Gong, 1995), and this noise can be additive, convolutional (channel or reverberant) or a combination of any of them. In the present work, we have investigated a multi-stream combination framework to address the issue of robustness towards additive noise in ASR.

In multi-stream systems, information from more than one source is combined to improve the performance, assuming that different sources considered for combination carry complementary information. The resulting additional information from multiple sources can lead to an improvement in performance over a single-stream system (Kirchhoff, 1998; Neti et al., 2001). The two important issues studied in this thesis in the framework of multi-stream ASR are: which information sources (feature representations) to combine and how much importance (weight) be given to each information source.

1.2 Motivation for the Present Work

1.2.1 Human Speech Recognition

The multi-stream approach towards ASR is inspired to a certain extent by human speech recognition (HSR). The examples of multi-stream in HSR include:

1. The ability of human beings to perceive weak (low amplitude) sounds in presence of noise can be explained by binaural hearing (Koenig, 1950; Kock, 1950). When compared to monaural hearing, the ability to detect a signal in a background masking signal is greatly improved with binaural hearing. In ideal conditions, the detection threshold for binaural hearing can exceed monaural hearing by 25 dB (Arons, 1992). This phenomenon of robust speech recognition is referred to as the binaural masking level difference (BMLD) (Kock, 1950).

2. It has been observed that humans use different acoustic cues to gain robustness, for example, the cocktail party effect indicates that humans rely on different features to lock-in to the voice of a particular speaker in presence of many other competing voices from different speakers (Cherry, 1953; Cherry and Taylor, 1954; Arons, 1992).

3. The effect of lip movement of the speakers in HSR was investigated in (McGurk and McDonald, 1976), and it was reported that human beings use visual cues in speech perception (McGurk effect).

4. Fletcher's perceptual experiments (Fletcher, 1953; Allen, 1994) on HSR demonstrated that human beings process different frequency bands independently and the independent estimates of the speech sounds in each frequency band are merged optimally at some higher level.

1.2.2 Multi-stream vs Single-stream ASR

1. In the past, it has been observed that combining evidence from more than one source improves the performance of ASR systems if the different sources carry complementary information and

are given importance according to their reliability (Dupont and Luettin, 1998; Morgan et al., 1998; Kirchhoff, 1998; Sharma et al., 2000; Zhu et al., 2005b). In Chapter 6, the performance of multi-stream systems is compared with single-stream systems, and the potential of multi-stream systems is demonstrated using an "oracle test".

2. Including additional information by appending temporal features (deltas and double-deltas) to static features[1], also known as feature-level combination or early integration, has shown to improve ASR performance (Furui, 1986).

3. Processing techniques such as spectral subtraction and mean normalization, which make individual feature streams more robust, further help multi-stream systems in improving the performance.

Besides these motivations, multi-stream systems are a step towards fail-safe systems from an engineering perspective as well. In the case of failure of one of the streams, the system can still work, perhaps with a reduced performance. Considering these properties of multi-stream systems, multi-stream combination is an interesting research area and has been investigated in this thesis.

1.3 Contribution of the Thesis

Out of several factors which need to be considered in multi-stream combination, this thesis focuses on the following two important issues:

1. **Weights given to the posterior outputs** of each classifier[2] in the combination: The following three weighting techniques are investigated:

   (a) **Maximum-posterior (MP) weighting:** The posterior outputs of a classifier are weighted in proportion to its maximum-posterior probability. The approach is motivated by the reasoning that a classifier gives a high probability for a class if it is confident about that class. Furthermore, it will yield low probabilities for all the classes if it is unable to distinguish among them. By this reasoning, a classifier with high maximum-posterior probability is more confident about its decision and its outputs should be given more weight.

[1] The feature extraction process is described in Section 2.1.1.

[2] In this work, we have used hybrid hidden Markov model/artificial neural network (HMM/ANN) systems, where a multi-layered perceptron (MLP) is trained as a classifier (Bourlard and Morgan, 1994).

(b) **Inverse entropy weighting:** The entropy at the output of a classifier is low when the classifier gives high probability for a particular class and low probabilities for rest of the classes. Such a classifier is more confident about its classification and its outputs should be given more weight. In contrast, a classifier having equal probabilities for all the classes has the highest entropy and is least confident about its decision. Therefore, the outputs of a classifier with high entropy should be given less weight (Okawa et al., 1999; Heckmann et al., 2002; Misra et al., 2002, 2003).

(c) **Maximum-likelihood (ML) weighting:** ML weighting is inspired by the reasoning that an increase in likelihood of test data might help in improving the recognition accuracy also. The weights at the output of the classifiers are estimated for each class and classifiers' outputs are combined such that the likelihood of the test data is maximized.

2. **New sources of complementary information (feature representations)** to be used in the combination: We have studied two feature representations which are expected to carry complementary information when compared with baseline perceptual linear prediction (PLP) features (Hermansky, 1990).

   (a) **Pitch frequency feature:** The pitch frequency carries information about the vocal-tract excitation signal. It may have information complementary to commonly used cepstral features which capture the characteristics of the vocal-tract. Further, pitch frequency is one of the cues on which humans rely to improve robustness in the cocktail party effect (Arons, 1992).

   (b) **Multi-band spectral entropy features:** The multi-band spectral entropy features are proposed to capture the spectral peaks of the spectrum which are more resistant to additive noise compared to other parts of the spectrum (Misra et al., 2005a; Misra and Bourlard, 2005). Recent studies on the spectral entropy feature have shown that the features are robust to noise and have properties different from energy based features (McClellan and Gibson, 1997; Shen et al., 1998; Subramanya et al., 2005).

The two main contributions of this work, namely inverse entropy weighting and spectral entropy features, are further investigated with the help of the following:

1. **Oracle analysis:** An Oracle test is suggested to evaluate the potential of multi-stream systems by selecting the outputs of the classifier at every time instant that has the highest pos-

terior for the correct class. The Oracle test gives support to inverse entropy weighting and illustrates the complementarity of spectral entropy features (Misra et al., 2005b).

2. **Multi-stream embedded training:** Single-stream embedded training is known to yield improved performance in HMM/ANN systems. In the present work, such a scheme is proposed and investigated for multi-stream HMM/ANN systems. The proposed method yields a significant improvement in ASR performance, both in clean as well as noisy test conditions (Misra et al., 2005b).

3. **Multi-stream Tandem:** The Tandem model (Hermansky et al., 2000) for ASR is a two stage sequential model where the feature processing is done by an MLP and the outputs of the MLP (after some processing to make them Gaussian-like and uncorrelated) are used as features in a standard hidden Markov model/Gaussian mixture model (HMM/GMM) system. In multi-stream Tandem, the outputs of MLPs trained on different feature streams are combined and the combined output is modelled by an HMM/GMM system. Inverse entropy weighting and spectral entropy features are investigated in this framework, showing an improvement over a single-stream Tandem system (Misra and Bourlard, 2005).

4. **Large vocabulary ASR:** The methods developed on a small vocabulary task were further verified on a large vocabulary task. The improvements obtained on the small vocabulary task were observed on the large vocabulary task as well.

1.4 Organization

The present thesis has been organized into 9 chapters. The next two chapters (Chapters 2 and 3) give the general overview and the later chapters are related to the research work carried out in this thesis.

Chapter 2 presents an overview of the current state-of-the-art HMM-based ASR systems. In this chapter, we explain different components of an ASR system, giving a description of the two main HMM based approaches to ASR, namely HMM/GMM and HMM/ANN. We conclude the chapter with a discussion on robustness issues in ASR systems.

In Chapter 3, the motivation for multi-stream combination is presented. We identify the important constituents of multi-stream combination systems, and review some of the contributions made to multi-stream ASR in the past. Subsequently, we explain the "full-combination multi-stream" approach, a special case of multi-stream combination. In this chapter, we also describe the OGI Numbers95 spontaneously spoken connected digit US English database. This database has been used to carry out the ASR experiments related to robustness studies in this thesis.

Chapter 4 gives details about weighting techniques for multi-stream combination that have been studied in this thesis. We give the motivation for each weighting technique and present the results for them on hybrid HMM/ANN systems.

The feature streams considered for combination should carry complementary information for a multi-stream system to yield better performance compared to an ASR system using a single feature stream (typically cepstral features). We investigate pitch frequency and spectral-entropy as additional feature streams in multi-stream combination in Chapter 5.

Chapter 6 is organized to analyze the performance of an Oracle in a multi-stream setup and study its relationship with inverse entropy weighting. In the same chapter, we explain embedded training, and then propose the idea of embedded training in the multi-stream framework for hybrid systems.

Chapter 7 gives an introduction to Tandem systems where we describe two existing variants of Tandem systems, and discuss multi-stream Tandem systems. The weighting techniques and the features studied in the previous chapters are used in multi-stream Tandem systems and the results are presented.

The studies conducted on OGI Numbers95 database (Chapters 4, 5, 6 and 7) are subsequently tested on a large vocabulary conversational telephone speech database in Chapter 8.

In the last chapter, we summarize the techniques studied in this thesis and draw conclusions. We also suggest future directions that can be pursued to improve the performance further.

Chapter 2

Speech Recognition: An Overview

Speech is a sequence of sounds which follows the phonological, semantic, lexical and syntactical constraints of the language in which it has been produced.

The block diagram of a typical automatic speech recognition system is shown in Fig. 2.1. Similar to many other pattern recognition tasks, there are two phases in ASR, training and testing. In

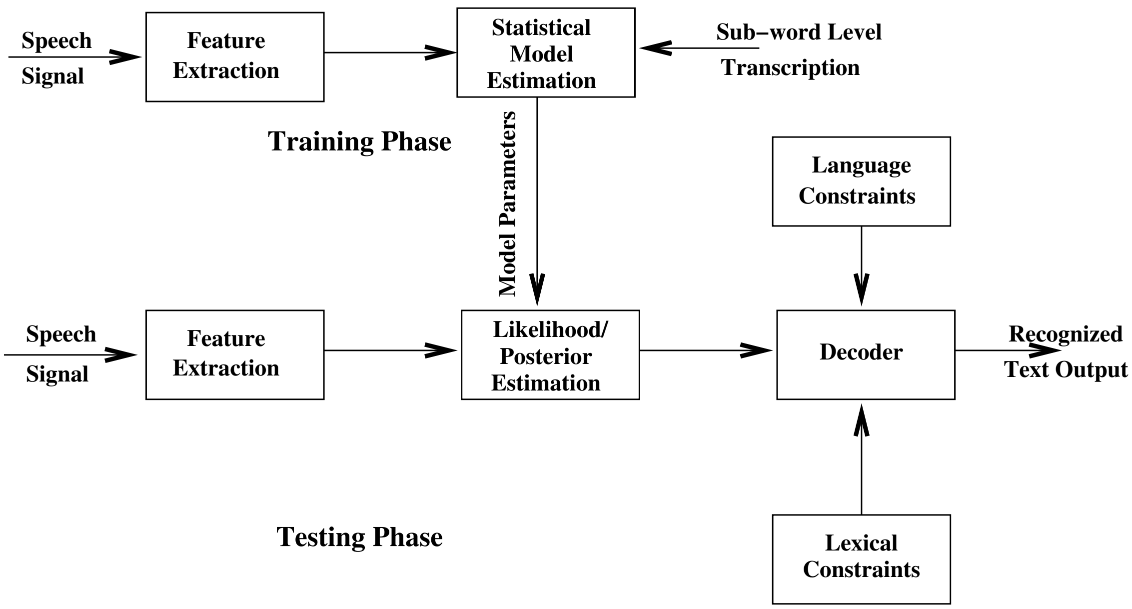


Figure 2.1. *A standard speech recognition system with training and testing phases.*

ASR, feature extraction involves computing a sequence of vectors to capture the linguistic information present in the speech signal (Section 2.1.1). Training consists of estimation of parameters of the different statistical models in the ASR setup, typically using feature vectors extracted from a limited amount of training data. In the testing phase, given the feature vectors extracted from the test utterance and the model parameters learned during training, pattern matching is performed using the constraints of the lexicon and language. The most likely word sequence resulting from the pattern matching is output as the recognized text.

The list of words to be recognized are given in a dictionary (lexicon). A word in the dictionary is represented in two forms, sequence of letters and sequence of phonemes, and the phoneme sequence defines the pronunciation of the word. A phoneme is the smallest unit of speech that affects the meaning of a word and distinguishes one word from another in a given language. Most languages do not follow a one-to-one mapping between letters and phonemes, English being a particularly good example.

On the higher level, words are made from sequence of phonemes, and in turn, sequence of words forms sentences. Typically, phones (the acoustical realization of phonemes) are the preferred basic units (or classes) in ASR[1]. It is possible to have words as classes and have one model for each word, but in practice it is difficult to have sufficient realizations of each word while developing a large vocabulary system.

The rest of the chapter is organized as follows: in Section 2.1, we explain the different components of a typical ASR system. The issue of robustness towards noise in ASR is discussed in Section 2.2. In addition, a few of the current techniques pursued to address the robustness issue are reviewed briefly in this section.

2.1 Components of A Speech Recognition System

Typically there are three main components in ASR systems, feature extraction, acoustic modelling and decoding, each shown as a block in Fig. 2.1. In the following sections, we explain each of these components in detail.

2.1.1 Feature Extraction

Feature extraction is the process of retaining useful information of the signal while discarding redundant and unwanted information. However, in practice, while removing the unwanted informa-

[1]In state-of-the-art ASR systems, context-dependent phones are used instead of context-independent phones.

tion, we may also lose some useful information in the process. Feature extraction may also involve transforming the signal into a form appropriate for the models used for classification. In developing an ASR system, a few desirable properties of the features are:

1. High discrimination between sub-word classes

2. Low speaker variability

3. Invariance to degradations in the speech signal due to channel and noise.

Different feature representations have been developed to emphasize one or more desirable properties of the features mentioned above. The features used in state-of-the-art ASR systems are usually derived from short-time Fourier transform (STFT) of the speech signal. Commonly used features for ASR include Mel-frequency cepstral coefficients (MFCC) (Davis and Mermelstein, 1980), perceptual linear prediction (PLP) (Hermansky, 1990) cepstral coefficients and RelAtive SpecTrAl (RASTA)-PLP-derived cepstral coefficients (Hermansky and Morgan, 1994). Each feature has different properties, for example, PLP and MFCC perform well for clean speech conditions and degrade for noisy conditions while RASTA-PLP generally works well for conditions where noise is not speech-like and yields inferior performance in clean conditions.

In the present work, we have used PLP-derived cepstral coefficients to develop the baseline system. In Fig. 2.2, a block diagram of the PLP feature extraction process is shown.

The process of short-time feature extraction is explained in the following steps:

1. **Pre-emphasis:** Typically, the speech signal produced by human beings has a spectral slope of approximately -6 dB/octave for voiced sounds. This slope is because of two reasons: a) The shape of the glottal pulse introduces a slope of -12 dB/octave, and b) The lip radiation introduces a slope of +6dB/octave. Therefore, the resultant slope of approximately -6dB/octave exists in the recorded voiced speech sounds. Pre-emphasis is performed to remove this slope of -6dB/octave. To accomplish the task, the speech signal is passed through a high-pass finite impulse response (FIR) filter of order 1. The pre-emphasis is defined by,

$$y[n] = s[n] - P * s[n-1]$$

   where s[n] is the n^{th} speech sample, y[n] is the corresponding pre-emphasized sample and P is the pre-emphasis factor typically having a value between 0.9 and 1. Pre-emphasis ensures that in the frequency domain all the formants[2] of the speech signal have similar amplitude so that they get equal importance in subsequent processing stages.

2. **Frame Blocking:** Speech is a quasi-stationary signal and is stationary only for a short interval of time. This allows us to use block processing techniques such as discrete Fourier transform (DFT) to analyze speech signal. Typically, the signal analysis is performed by dividing the speech signal into small blocks of size 20-35 ms, and a shift of 5-15 ms is introduced between adjacent blocks.

3. **Windowing:** Frame blocking in the time domain corresponds to truncating the signal by a rectangular window. In the frequency domain, this leads to convolution of the Fourier transforms of the signal and the rectangular window. The frequency domain equivalent of a rectangular window has one main-lobe and several side-lobes. The width of the main-lobe of the window is dependent on the window type and size and should be as narrow as possible for better estimates of the spectrum. Also, the ratio between magnitudes of the side-lobes and main lobe of the window should be as large as possible to minimize interference from neighbouring frequency components. The Hamming window (Oppenheim and Schafer, 1975) is a common choice for windowing in short-time speech signal processing as it has a high ratio between amplitudes of side-lobes and main-lobe.

4. **Static Feature Computation:** The most commonly used short-time features in ASR are MFCC (Davis and Mermelstein, 1980) and PLP-derived cepstral coefficients (Hermansky, 1990) or a variation of them. To give an example of static features, we will describe the PLP feature extraction process in brief. As mentioned earlier, PLP-derived cepstral coefficients are used as features for developing the baseline ASR system in this thesis. Fig. 2.2 shows the different stages of PLP feature extraction process. To begin with, the short-time Fourier transform of the windowed signal is obtained. An estimate of energy in each band of the filter bank defined on a bark scale is obtained. The resultant filter bank energies are multiplied by an equal-loudness curve, and to simulate the power law of hearing, the output amplitudes undergo cube-root compression. The final smooth spectrum thus obtained is transformed by IFFT and using auto-regressive modelling[3] (Makhoul, 1975) the PLP coefficients are estimated. The order of LP analysis, p, is usually kept between 8 and 14 for telephone-quality

[2] Formants are peaks in the spectrum caused by resonances in the vocal tract. The position of the formants characterizes the speech sound.

[3] Also known as linear prediction (LP) analysis.

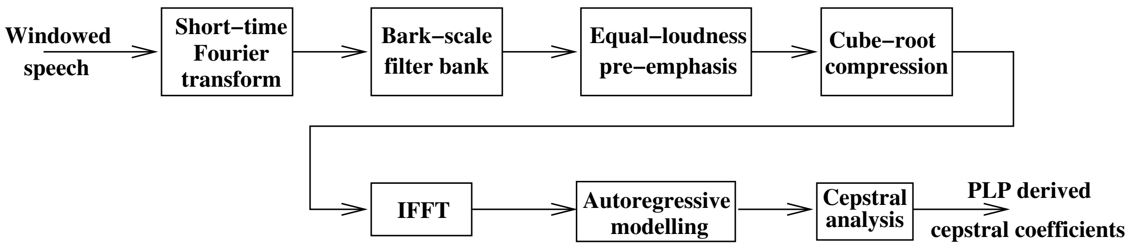


Figure 2.2. *Extraction of PLP derived cepstral coefficients from short time windowed speech signal.*

speech. p also has a physical interpretation and represents the number of poles of the signal (or system response). LP analysis is also known as all-pole modelling for the reason that it can model only the poles of the signal. In LP analysis, each complex-conjugate pole-pair corresponds to a peak of the spectrum and real poles model the roll-off of the spectrum. PLP-derived cepstral coefficients are obtained from PLP coefficients using a recursive equation (Makhoul, 1975; Rabiner, 1989).

5. **Dynamic Feature Computation:** The static features have information about the present frame only and do not carry any temporal or dynamic information. Dynamic features are obtained from static features and they capture the time trajectory of the static features. The temporal information is included by taking first and second order time derivatives of each feature component and appending them to the static features (Furui, 1981, 1986). The delta ($\Delta c_{t,l}$) and double delta ($\Delta\Delta c_{t,l}$) cepstral coefficients are the first and second order time derivative of the cepstral coefficients ($c_{t,l}$) respectively, and are obtained by,

$$\begin{aligned} \Delta c_{t,l} &= \frac{\sum_{k=-i}^{k=i} k \cdot c_{t+k,l}}{\sum_{k=-i}^{k=i} |k|} \\ \Delta\Delta c_{t,l} &= \frac{\sum_{k=-i}^{k=i} k \cdot \Delta c_{t+k,l}}{\sum_{k=-i}^{k=i} |k|} \end{aligned}$$

where t is the frame index, l is the index for feature component and i is generally kept between 2 and 4 to have a context of 5 to 9 frames (Furui, 1986).

Appending of dynamic features to static features is shown to improve ASR performance, and the gains are usually more in presence of noise (Misra et al., 2003; Yang et al., 2005).

6. **Cepstral Mean Subtraction (CMS) and Variance Normalization:** The channel through

which speech is captured imposes its characteristics on the speech signal. Channel effects are convolutional noise in the time domain, multiplicative noise in the frequency domain and additive noise in the cepstral domain. CMS (Atal, 1976) helps in reducing the channel mismatch. In CMS, the mean vector is computed from all the vectors of an utterance, and is subtracted from each feature vector of that utterance. Similarly, variance normalization helps in robustness against additive noise (Jain and Hermansky, 2001; Molau et al., 2003). In state-of-the-art ASR systems, CMS and variance normalization are applied to the feature vectors to alleviate the problem of channel mismatch and improve noise robustness respectively. Depending upon the application, the normalization might be done at utterance-level or online.

7. **Vocal-Tract Length Normalization:** Typically, the vocal-tract length of females are shorter than that of males, and as a consequence, the formant center frequencies between speakers can vary upto 25% (Lee and Rose, 1996). The mismatch caused by the difference in vocal-tract shape of speakers can lead to high inter-speaker variabilities (Wakita, 1977). This mismatch can be reduced by vocal-tract length normalization (VTLN). This technique is usually applied at the spectrum level where the spectrum is either stretched or compressed along the frequency axis by a factor (constant for each speaker) to enhance the likelihood of the observation (Cohen et al., 1995; Lee and Rose, 1998). VTLN has emerged as one of the important techniques to improve the performance of large vocabulary speaker-independent continuous speech recognition systems. VTLN is a two stage algorithm where the VTLN parameter on test data needs to be computed first (an iterative expectation-maximization procedure), and then applied to the spectrum to compute the features from the modified spectrum.

2.1.2 Acoustic Modelling

The ASR task can be defined as finding the word sequence given the feature vector sequence, $X = [x_1, x_2, \cdots, x_T]$[4]. This can be expressed as maximum-a-posteriori (MAP) problem (Jelinek, 1976; Rabiner, 1989):

$$\widehat{W} = \underset{W}{\arg\max} \quad \{P(W|X, \Theta)\} \tag{2.1}$$

[4]The feature vector sequence could be the same as the cepstral feature vectors appended by first and second order time derivatives described earlier.

where $\widehat{W}$ is the most likely word sequence, W is the set of all possible word sequences from the lexicon and Θ represents the set of parameters of the model (which needs to be estimated from training data).

It is difficult to estimate $P(W|X, \Theta)$ directly. However, we can convert it into a maximum-likelihood form using Bayes rule.

$$\widehat{W} = \arg\max_{W} \quad \left\{ \frac{p(X|W, \Theta)P(W|\Theta)}{p(X|\Theta)} \right\} \tag{2.2}$$

In (2.2), the term $p(X|\Theta)$ is common to all the hypotheses and thus can be dropped. The term Θ is assumed to have two parts, acoustic model parameters, Θ_a, and language model parameters, Θ_l. Usually, the two parameters are estimated separately and assumed to be independent of each other. The term $p(X|W, \Theta_a)$ is referred as the likelihood of the feature vector sequence X given the word sequence W and the acoustic model parameters Θ_a. Further, the term $P(W|\Theta_l)$ is the prior probability of the word sequence and is defined by a language model. Accordingly, (2.2) can be rewritten as:

$$\widehat{W} \approx \arg\max_{W} \quad \{p(X|W, \Theta_a)P(W|\Theta_l)\} \tag{2.3}$$

In state-of-the-art ASR systems, some parametric representation is assumed for the acoustic model, Θ_a, and parameters of the model are estimated using a training database and its transcription:

$$\widehat{\Theta}_a = \arg\max_{\Theta_a} \quad \left\{ \prod_{X \in \mathscr{X}} p(X|W, \Theta_a) \right\} \tag{2.4}$$

where $\mathscr{X}$ represents all the utterances in the training set.

In contrast to acoustic model, the language model, Θ_l, is typically estimated by counting the frequency of word-sequences (Bahl et al., 1983; Clarkson and Rosenfeld, 1997) and is expressed in terms of n-grams (for example, n=2 gives a bigram language model).

In maximum-likelihood (ML) training given by (2.4), the aim is to find Θ_a which maximizes the likelihood of the training set. Expectation-maximization (EM) (Baum et al., 1970; Dempster et al., 1977) is a popular iterative algorithm employed to estimate Θ_a. In EM, a few hidden variables are added to the parameter set (Θ_a) to simplify the otherwise intractable problem.

In each iteration of EM, the parameter set Θ_a is reestimated using the previous parameter estimates Θ_a^s such that the likelihood of the training set is increased. That is,

$$\prod_{X \in \mathscr{X}} p(X|W, \Theta_a) \geq \prod_{X \in \mathscr{X}} p(X|W, \Theta_a^s) \tag{2.5}$$

Each EM iteration involves two steps, estimation and maximization. In the estimation step of EM, the posterior distribution of the hidden variables is estimated using the parameters of the previous step (Θ_a^s). In the maximization step, the posterior estimates are used to estimate the new parameters (Θ_a). The two steps when repeated increase the likelihood of the data. The proof for convergence of EM can be found in (Baum et al., 1970; Dempster et al., 1977).

In state-of-the-art ASR systems, typically hidden Markov model (HMM) are used for acoustic modelling (Rabiner, 1989; Bourlard and Morgan, 1994), and are discussed next.

Hidden Markov Models (HMM) for ASR

The HMM is a finite state automaton and each state is associated with an output process which is stochastic. Further, the transition between the states is non-deterministic. From the feature extraction module, we extract sequence of feature vectors denoted by $X = [x_1, x_2, \cdots, x_T]$. This sequence of feature vectors is called the observation sequence in Markov modelling terminology. In the HMM, it is assumed that this observation sequence is piecewise stationary and has been generated by a sequence of states. The state sequence is hidden in HMM systems and is the hidden variable of EM. The state sequence in the HMM is represented by $Q = [q_1, q_2, \cdots, q_T]$, $q_t \in 1, \cdots, N$, where N is the number of states, and transition between states takes place with time. The stochastic output process associated with each state generates the feature vectors for that state. Generally, some physical interpretation can be assigned to the states. In standard ASR systems, the states usually represent phones (or parts of phones). We need to determine the parameters of the HMMs, represented by Θ_a in (2.4), from the observation sequence during training.

The state sequence is hidden in HMM systems and has to be interpreted from the observation sequence. Including the hidden variable Q in the likelihood term $p(X|W, \Theta_a)$, the likelihood term can be rewritten as[5]:

$$\begin{aligned} p(X|W) &= \sum_Q p(X,Q|W) && (2.6)\\ &= \sum_Q p(X|Q,W)P(Q|W) && (2.7) \end{aligned}$$

In the equation, $\sum_Q$ refers to set of all possible state sequences. We have two terms in the equation, $p(X|Q,W)$ and $P(Q|W)$, and we can solve them separately.

To solve the first term, $p(X|Q,W)$, we make the assumption that the probability of the current observation x_t depends only on the current state q_t, and is independent of all other observations as well as states (independent and identical distribution). Also, the probability is independent of time.

$$\begin{aligned} p(X|Q,W) &= p(x_1,\cdots,x_T|q_1,\cdots,q_T,W)\\ &\approx \prod_{t=1}^{T} p(x_t|q_t) && (2.8) \end{aligned}$$

The second term is solved as follows using first order Markov assumption (the current state q_t depends only on the previous state q_{t-1}):

$$\begin{aligned} P(Q|W) &= P(q_1,\cdots,q_T|W)\\ &\approx P(q_1)\prod_{t=2}^{T} P(q_t|q_{t-1}) && (2.9) \end{aligned}$$

Substituting (2.8) and (2.9) in (2.7), we get

$$p(X|W) = \sum_Q P(q_1)p(x_1|q_1)\prod_{t=2}^{T} P(q_t|q_{t-1})p(x_t|q_t) \tag{2.10}$$

The solution to evaluate the likelihood of an observation sequence for an HMM is given by (2.10). An efficient procedure known as Baum-Welch algorithm (Baum et al., 1970) (also known as the forward-backward algorithm) exists to compute (2.10). An approximation to (2.10) is the likelihood of the best state sequence as realized by Viterbi decoding (Viterbi, 1967; Forney, 1973; Rabiner, 1989) and is given as follows:

$$p(X|W) = \max_Q P(q_1)p(x_1|q_1)\prod_{t=2}^{T} P(q_t|q_{t-1})p(x_t|q_t) \tag{2.11}$$

[5]In the following derivation, the dependency term Θ has been dropped for readability.

The three parameters of HMMs in (2.10) are:

1. *State-transition probability distribution*: It is represented by $A = \{a_{i,j}\}$, where

$$a_{i,j} = P(q_t = j | q_{t-1} = i) \qquad 1 \leq i, j \leq N \tag{2.12}$$

defines the probability of transition from state i to j at time t, q_t is the state at time instant t and N is the total number of states in the HMM.

2. *Observation probability distribution*: It is given by $B = \{b_j(x_t)\}$, in which

$$b_j(x_t) = p(x_t | q_t = j) \tag{2.13}$$

defines the probability density of observation vectors in state j, $j = 1, \cdots, N$.

3. *The initial state distribution*: It is represented by $\Pi = \{\pi_i\}$, where

$$\pi_i = P(q_1 = i) \qquad 1 \leq i \leq N \tag{2.14}$$

The set of parameters of the HMM (A, B and Π) is represented by Θ_a in (2.4).

As mentioned earlier, the hidden variables of the HMMs in the EM training are state sequence. A description of EM for HMM training can be found in (Rabiner, 1989; Bourlard and Morgan, 1994; Bilmes, 1998).

The two popular techniques used in the current ASR systems for modelling the emission probabilities $p(x_t|q_t)$, namely Gaussian mixture model (GMM) and artificial neural network (ANN), are discussed in the next section.

HMM/GMM System

HMM/GMM systems are used extensively for ASR. They are also known as continuous HMMs as the emission probability is computed by a mixture of Gaussians in these systems.

The emission probability density, $p(x_t|q_t = j)$, in HMM/GMM systems is given by

$$b_j(x_t) = \sum_{m=1}^{M_j} c_{j,m} N(x_t; \mu_{j,m}, \Sigma_{j,m}) \tag{2.15}$$

where M_j is the number of mixture components in state j, $c_{j,m}$ is the weight of the m^{th} component such that $\sum_{m=1}^{M_j} c_{j,m} = 1$, and $N(x_t; \mu_{j,m}, \Sigma_{j,m})$ is a multivariate Gaussian with $\mu_{j,m}$ and $\Sigma_{j,m}$ as mean vector and covariance matrix respectively, that is,

$$N(x_t; \mu_{j,m}, \Sigma_{j,m}) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_{j,m}|}} exp(-0.5(x_t - \mu_{j,m})^T \Sigma_{j,m}^{-1} (x_t - \mu_{j,m})) \tag{2.16}$$

where d is the dimensionality of x_t.

HMM/ANN System

In the typical ASR system illustrated in Fig. 2.1, for hybrid HMM/ANN systems, the density estimation stage is performed by an ANN. In hybrid HMM/ANN systems, a multi-layered perceptron (MLP) with single hidden layer is trained using back-propagation (McClelland et al., 1986; Bishop, 1995; Yegnanarayana, 1999). The input to the MLP are feature vectors, usually with a context of 4 frames on either side, that is 9 frames as input. Typically, the number of nodes in the output layer of the MLP is equal to the number of classes (phonemes in the database) and each phoneme class is represented by an HMM. The number of nodes in the hidden layer is generally kept more than the number of nodes in the input or output layers (Bourlard and Morgan, 1994). When the MLP is used as a classifier, typically the activation function for the nodes in the input layer is linear where as for the nodes in hidden and output layers, the activation function is non-linear, such as sigmoid or softmax. In HMM/ANN ASR systems, the output is typically softmax (Bourlard and Morgan, 1994) and the training is done with *one-hot-encoding*. In one-hot-encoding, the desired output of the MLP is 1 for the correct class and 0 for rest of the classes. Usually the training is done in batch mode and increase in cross-validation error (Bishop, 1995) is used as a stopping criterion. The softmax output ensures that the outputs of the MLP are posterior probability estimates for the phones given the input data (Bourlard and Wellekens, 1989; Richard and Lippmann, 1991). The posterior estimates are represented as:

$$P(q_k | x_{t-4}^{t+4}) \qquad k = 1, \cdots, K \tag{2.17}$$

where t is the current frame index, x_{t-4}^{t+4} represents the input vector with a context of 4 frames on either side (Bourlard and Morgan, 1994), and K is the number of phonemes.

These posterior estimates can be converted into scaled likelihoods by Bayes rule

$$\frac{p(x_{t-4}^{t+4}|q_k)}{p(x_{t-4}^{t+4})} = \frac{P(q_k|x_{t-4}^{t+4})}{P(q_k)} \tag{2.18}$$

where $P(q_k)$ is the phoneme prior probability computed from the training data and $p(x_{t-4}^{t+4})$ is the likelihood of the data. Assuming that the likelihood of the data is same for all the classes, it can be treated as a scaling factor. The scaled likelihoods thus obtained are treated as emission probabilities and given to the HMM for Viterbi decoding. The HMMs used in hybrid systems usually have fixed state-transition probabilities of 0.5. Further, each phone has a one state model and the minimum duration of each phone is modelled by forcing one to three repetitions of the same state for each phone (Robinson et al., 1996). The scaled likelihoods are provided as emission probabilities to the state models.

In hybrid HMM/ANN systems, the ANN is trained for classification and it provides frame-level phonetic discriminant learning. Given the input pattern, ANN outputs estimate posterior probabilities of output classes when the network is trained for classification to minimize one of the several common cost functions such as least mean square error or relative entropy (Bourlard and Wellekens, 1989; Richard and Lippmann, 1991; Bourlard and Morgan, 1994). Additionally, a system that estimates these posterior probabilities minimizes the error rate while maximizing discrimination between the correct output class and the competing classes (discriminant training). In practice, ANN-based systems can be trained to generate good estimates of these ideal probabilities (one for the correct class and zeros for the rest of the classes).

Moreover, ANNs can incorporate multiple constraints and find optimal combinations of constraints. Therefore, there is no need for strong assumptions about the statistical distributions of the input features or about higher order correlation of the input data. In theory, the correlation can be discovered automatically by the ANNs during training. This ability of ANNs to model higher order correlation is very useful when dealing with features which have high correlation between different components. In contrast, in HMM/GMM systems, we need to decorrelate the features first as the usual assumption in the GMM is that the covariance matrix is diagonal. The assumption of diagonal covariance matrix simplifies the estimation procedure. Further, it is difficult to reliably estimate the correlation between different dimensions from a limited amount of training data.

2.1.3 Pattern Matching or Decoding

The third important component of an ASR system involves pattern matching. In a large vocabulary ASR systems, pattern matching can be done at more than one stage, for example, at acoustical, word and syntax levels.

In the typical ASR system shown in Fig. 2.1, while decoding, the lexical constraints are imposed by a dictionary which contains words along with their corresponding pronunciation (phonetic transcription). A word might be described by more than one phonetic transcription in the dictionary to take care of pronunciation variants (Lucassen and Mercer, 1984).

Similarly, syntactic constraints are imposed by a language model where the probabilities of the words, word pairs, word triplets and more are stored. The use of a language model is dependent on the task in-hand. For instance, in a connected digit recognition task, a uniform language model is sufficient as any digit can follow any digit. In contrast, a language model is complex for a conversational speech recognition tasks. In general, the language model is obtained from written text data and is stored in terms of fixed probabilities. The language model is usually expressed in terms of "n-grams", where n describes the dependency of the current word on the past n-1 words. In a large vocabulary ASR task, unigrams (n=1), bigrams (n=2) and trigrams (n=3) are the commonly used language models. While using a particular n-gram model, in case sufficient statistics are not available for a particular word sequence of length n, k-gram statistics are used instead of n-gram statistics, where k $<$ n. This is known as backed-off language modeling (Katz, 1987). As expected, an n-gram language model is described by a Markov process of order n-1. For example, the bigram model probabilities for words are computed by:

$$P(w_r|w_{r-1}) = N(w_r, w_{r-1}) / \sum_k N(w_k, w_{r-1}) \tag{2.19}$$

In the equation, $N(\cdot)$ is the count of word pairs, w_k represents the k^{th} word in the dictionary and $P(w_r|w_{r-1})$ is the probability of word w_{r-1} being followed by word w_r. The language model, represented by Θ_l in (2.3), comprises a set of n-gram probabilities discussed above, and is used to incorporate the constraints of the language.

2.2 Noise Robustness in ASR

Speech can get affected by the channel through which it passes or the environment in which it is produced and/or recorded. For instance, the channel imposes its characteristics in the form of convolutional and additive noises and the environment can introduce additive or reverberant (which is convolutive) noises to the speech. These undesirable changes in the speech signal can affect the performance of an ASR system considerably.

The problem of robustness can be addressed at various stages of ASR. We will discuss a few of them in this section:

1. *Robust feature extraction*: Features that are less affected by certain kinds of noises can be extracted from the speech signal. For example, RASTA (Hermansky and Morgan, 1994) feature extraction uses filters to remove those components of the signal which do not follow the dynamics of speech. Therefore, the feature gets less affected by stationary noises compared to PLP or MFCC features. Similarly, phase-autocorrelation (PAC) features (Ikbal et al., 2003b) are able to enhance the peaks of the spectrum and were shown to be less affected by additive noise. In the same vein, the Mel-cepstrum modulation spectrum (MCMS) (Tyagi et al., 2003) captures different dynamics of the Mel-cepstral features by projecting them on sine and cosine bases to yield an improved performance under different noise conditions. Spectro-Temporal Activity Pattern (STAP) features (Ikbal et al., 2004a) were suggested to capture the pattern around the peaks of the spectrum and were shown to be less vulnerable to noisy conditions.

   Each of these feature representations captures different characteristics of the speech signal. However, a single feature representation might not perform well under all conditions. This makes multi-stream a promising approach where the information from different feature representations is combined to have more information and hence a better description of the speech signal.

2. *Feature normalization*: Normalization techniques like cepstral mean (Furui, 1986) and variance normalization can help in reducing the mismatch caused by channel and additive noise respectively. Additionally, techniques like histogram normalization (Molau et al., 2001) can also help in reducing the mismatch between train and test features.

3. *Spectral Subtraction*: Spectral subtraction (Boll, 1979; Berouti et al., 1979; Lockwood and Boudy, 1992; Ris and Dupont, 2001) is a popular approach to reduce the effect of additive

uncorrelated noise in a signal. In this technique, an estimate of the noise spectrum, usually a time average of the spectrum in regions where only noise is present, is subtracted from the spectrum of noisy signal.

4. *Missing data approach*: In this approach, it is assumed that noise affects only a few regions in the spectro-temporal plane and it is possible to identify such regions. These regions are treated as unavailable or unreliable and methods are suggested for the two possible conditions to enhance the signal and recognition accuracy (Cooke et al., 2001). If it is assumed that the noise affected regions are unavailable, they are marginalized out while doing the estimation. In contrast, when it is assumed that the regions affected by noise are unreliable, they are estimated from the reliable parts by conditioning.

   The approach is hindered by the fact that it is always difficult to detect the reliable and unreliable parts with high accuracy in the presence of noise.

5. *Better model generalization*: Models can be made robust by training them on different conditions (various noise types and levels (Hirsch and Pearce, 2000), different speaking styles (Lippmann, 1987)) or adapting them to new test conditions using techniques such as maximum-likelihood linear regression (Leggetter and Woodland, 1995). However, in practice, it is difficult to simulate all possible conditions and performance is affected for unseen conditions.

6. *Multi-band combination*: Inspired by Fletcher's studies (Fletcher, 1953), in the multi-band combination approach, the full-band spectrum is divided into smaller sub-bands and features are computed from individual sub-bands. Separate classifiers are trained for each sub-band's features and the outputs of different classifiers are combined. If the noise is band-limited, only certain sub-bands get corrupted and accordingly only the classifiers for those sub-bands are affected. Though multi-band approaches work very well for band-limited noises, they perform poorly for clean and white noise conditions (Bourlard and Dupont, 1996; Hermansky et al., 1996).

   Multi-band is a special case of multi-stream combination, and will be reviewed briefly in the next chapter (Section 3.3).

In this thesis, the issue of robustness in ASR towards additive noise has been addressed, and the multi-stream combination approach has been proposed to alleviate this problem. Many of the techniques mentioned in this section, such as feature normalization and spectral subtraction, which make the individual feature representations more robust, can be used directly to further improve the performance of the multi-stream systems. In the next chapter, we discuss multi-stream combination systems and their issues.

Chapter 3

Multi-stream Combination

The use of information from more than one source to arrive at a decision is the basic concept of multi-stream. Perhaps, the most common example of multi-stream combination observed in day-to-day life is binaural hearing (hearing by two ears). Binaural hearing helps us in finding the direction and distance of the sound sources. In addition, binaural hearing enables the auditory system to detect certain sounds at much lower intensity levels compared to using only one ear (binaural masking level difference) (Kock, 1950). Multi-stream combination can also exist across different modalities, for instance, hearing and vision are two different senses and are complementary. Evidence from both of them can be combined to arrive at a decision (McGurk and McDonald, 1976; Chen, 2001).

The rest of the chapter is organized as follows: in the next section, the motivation for multi-stream combination for ASR will be presented. The issues of multi-stream ASR are discussed in Section 3.2. In Sections 3.3 and 3.4, we review multi-band and multi-stream ASR respectively, listing some of the important contributions in each field. In Section 3.5, full-combination multi-stream (FCMS), a special case of multi-stream, is described. Finally, in Section 3.6, we explain the setup and database used to carry out the experiments reported in this thesis.

3.1 Motivation

Multi-stream combination is one of the ways to improve the robustness of a system. Multi-stream has been studied in various areas of pattern recognition. Multi-stream combination, classifier ensembles (Kuncheva, 2005), multiple expert systems, classifier fusion (Buxton and Langdon, 2001) and several other synonyms are frequently used to describe the class of systems where the outputs (classification decisions) of more than one classifier is combined to get an improved performance (Poh and Bengio, 2005). The underlying principle of multi-stream combination is to obtain a better estimate of the optimal decision rule by combining outputs of several classifiers having complementary source of information.

As pointed out by Morgan et al. (1998), it is advantageous to combine as many sources of information as possible in a recognition process with the condition that weaker and stronger information sources get less and more importance respectively. The importance given to the decision of a classifier is generally referred to as weight.

In state-of-the-art ASR systems, delta and double-delta cepstral features are appended to the static cepstral features to include additional information, and thereby improve the performance (Furui, 1986). This is an example of multi-stream combination where the feature representations (feature streams) carrying different information are combined at the feature-level.

3.1.1 Multi-stream Processing in Human Speech Recognition

Audio-visual cues: McGurk effect

McGurk and McDonald (1976) reported that in human beings acoustic information is combined with visual information at a sub-conscious level, and the perceived phoneme category is influenced by lip movement. This particular phenomenon is known as *McGurk effect*. The McGurk effect demonstrates how human beings use visual speech information and also the bimodal nature of speech perception. Especially in noisy speech conditions, it has been shown that combining mouth shape with acoustic data can improve the recognition performance of an ASR system (Dupont and Luettin, 1998).

Hearing physiology

Evidence exists that multiple experts (specialized cells) are used in the first stage of central auditory processing in the mammalian auditory system. In the cochlear nucleus, each fiber in the auditory nerve splits and carries the same data by about seven different types of specialized cells. Each of these cells have a very different characteristic response and the outputs from these cells are combined at higher levels of processing (Pickles, 1998).

Multi-band processing

Multi-band processing was investigated by Fletcher (1953). His studies on speech perception to improve the intelligibility of speech over telephone network under conditions of filtering and noise support the idea of multi-stream processing. His work has been summarized in (Allen, 1994) and is an important contribution towards understanding human speech recognition (HSR).

In his studies on phone[1] articulation (empirical probability of correct recognition of sounds in absence of any context) for various channel frequency responses and channel noises, Fletcher passed the speech signal through low-pass and high-pass filters and did perceptual experiments on several listeners. He found that the listening errors made in the low-pass filtered band are independent of the listening errors made in the high-pass filtered band, and a phone is misrecognized only if a listener makes an error in both the sub-bands (if any of the sub-band is recognized correctly, the recognition will be correct). This is an important result which shows that the phones are processed in independent frequency channels (*articulation bands*) and these independent estimates of the speech sounds in each frequency band are merged "optimally" (in humans).

This relationship was later interpreted by Allen as, "we are listening to independent sets of phone features in the two bands and processing them independently up to the point where they are fused to produce the phone estimates" (Allen, 1994, Page 572).

The two band model was generalized to a multichannel articulation band model by Fletcher (originally proposed by J. Q. Stewart) and is given as,

$$\varepsilon = \varepsilon_1 \cdot \varepsilon_2 \ldots \varepsilon_K \tag{3.1}$$

where K is the number of independent articulation bands and ε_k is the error of k^{th} band. This model was referred to as *Fletcher-Stewart multi-independent channel model of phone perception*. This is also referred to as the *product of errors rule* in the literature, for example Bourlard (1999). This analysis of Fletcher on the independence of sub-bands is the basis of multi-band approaches in ASR.

[1] A phone is the acoustical realization of a phoneme.

3.1.2 Engineering Aspects

From an engineering perspective, multi-stream systems could be a solution for designing fail-safe systems. The redundancy which exists in multi-stream systems makes them more robust against failures. Multi-stream systems can give reasonable performance in the case of failure of some streams in the system. In short, multi-stream systems are useful from a reliability perspective. Further, these systems may give an improved performance when all the classifiers trained on different feature streams work reliably and their outputs are combined optimally. We will discuss more about it in Chapter 6.

The techniques which make the individual feature streams robust, such as spectral-subtraction, CMS, variance normalization and VTLN, improve the performance of multi-stream systems further when robust individual feature streams are used for combination.

Variance reduction

The outputs of a single classifier are typically affected by large variances. The goal of an ideal multi-stream combination system is to reduce this variance and hence the confusion between the classes (Bishop, 1999; Poh and Bengio, 2005).

In an ensemble of classifiers, if the errors (deviation from the true value) of the classifiers have zero means and are uncorrelated, the combination of classifiers' outputs can reduce the expected square error by a factor equal to the number of classifiers in the ensemble (Bishop, 1999, Chapter 9). In practice, errors might not be uncorrelated and thus the reduction in error is considerably smaller than the theoretical limit. This variance reduction usually results in a better performance by combining the classifiers compared to the performance of a single classifier in the ensemble.

Oracle performance in multi-stream systems

In (Kuncheva, 2002), performance of several multi-stream combination rules was investigated for a two class problem. It was reported that the error can be reduced close to zero if outputs of a large number of classifiers are combined by an 'oracle'. In the proposed oracle experiments, a classification decision was considered to be correct if at least one of the classifiers had made the correct classification. This is similar to Fletcher's "product of errors rule" given by (3.1).

In (Shire and Chen, 2000), the MLP classifiers were trained on different feature representations and the outputs of the classifiers were combined to study the performance of multi-stream ASR for reverberant acoustic conditions. A frame-level oracle was used to pick the outputs of the classifier that had the highest posterior probability for the correct class. It was shown that the performance of the oracle was significantly better than the performance of the individual classifiers used in the combination.

An oracle has also been employed to ascertain the improvement that can be gained while scoring several hypotheses in a lattice in an ASR task (Roark et al., 1994) compared to using the single best hypothesis. In ASR, a lattice contains several alternative paths which can be chosen while decoding (the path of maximum-likelihood being the usual choice). The paths in a lattice could be viewed as an example of multi-stream system where the combination is done at the decoder-level (discussed in Section 3.2.1, Fig. 3.3). It was shown that the 'oracle-path' can reduce the word-error-rates on large vocabulary Switchboard database by a significant margin.

In practice, it is not known how to design an oracle which can choose the "right" classifier or the "right" path. Nevertheless, an oracle analysis can indicate the potential performance of multi-stream systems and demonstrates the advantage of multi-stream systems over single-stream systems.

In Chapter 6, we have investigated an oracle test for multi-stream ASR setup. The proposed oracle test does frame-level weighting where the outputs of the classifier which has the highest posterior for the right phoneme class are chosen at every time instant for decoding (Shire and Chen, 2000). An analysis of the oracle reveals some interesting results which are discussed in detail in Chapter 6.

3.2 Issues in Multi-stream ASR

3.2.1 Combination Level

The multi-stream combination can be performed at various stages. In feature-level combination (Furui, 1986; Okawa et al., 1998), different feature representations are concatenated (Fig. 3.1) and a model is trained for the single concatenated feature vector stream. The most common example of concatenation and modelling in ASR is appending of delta features to the static features $(c_t, \Delta c_t, \Delta\Delta c_t)$[2]. Feature-level combination is commonly known as early integration in the field of machine learning.

[2] Discussed in detail in Section 2.1.1.

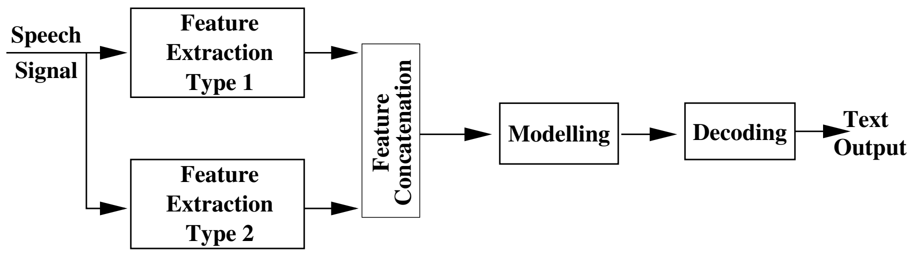


Figure 3.1. *Feature combination: Different feature representations are obtained from the speech signal and appended (concatenated) to form a feature vector. The combined feature vector is modelled and decoded by the usual ASR system.*

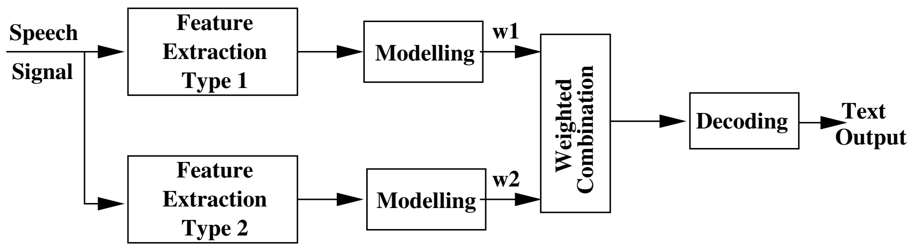


Figure 3.2. *Posterior (or likelihood) combination: Different feature representations are obtained from the speech signal and modelled separately. The outputs of the models (either posteriors or likelihoods) are weighted and combined. The combined outputs are decoded.*

In posterior-level combination shown in Fig. 3.2 for hybrid HMM/ANN systems, different feature representations obtained from the speech signal are modelled separately and the outputs of the models (MLP classifiers) are weighted and combined (Bourlard and Dupont, 1996; Hermansky et al., 1996; Misra et al., 2003; Hagen and Morris, 2005). The combined outputs thus obtained are used for decoding. Combination at the posterior-level is also known as late integration in the area of machine learning.

In ROVER (Fiscus, 1997; Schwenk and Gauvain, 2000), the combination is done at the outputs of the decoder (Fig. 3.3). In ROVER, first the text outputs obtained from different decoders are time

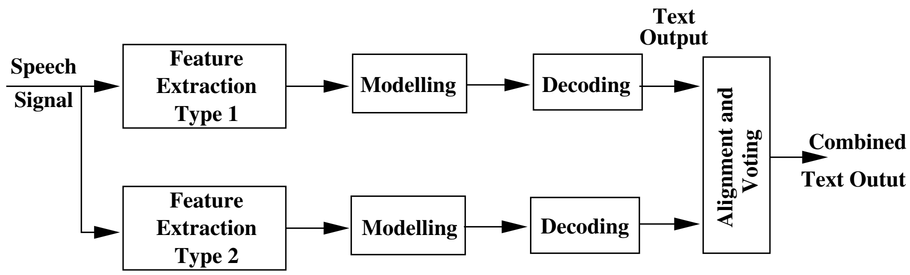


Figure 3.3. *Decoder output combination: Different feature representations obtained from the speech signal are modelled and decoded separately by individual ASR systems. The decoded output (which is text) is combined with techniques such as majority voting to get the final text output.*

aligned and then weighted or unweighted majority voting is performed over the aligned outputs.

3.2.2 Features for Different Streams

An important task in multi-stream combination is to identify feature representations that carry complementary information. When the outputs of the classifiers trained on such feature representations are combined correctly, the combined outputs yield fewer errors compared to the outputs of individual classifiers used in the combination. In rest of the thesis, the term *feature stream* is used to denote feature representation and the term *stream* is used to represent outputs of the classifier trained on the feature representation.

In a multi-stream system, an improvement in performance can be achieved by combination only if the error characteristics of the outputs of the classifiers trained on different feature streams are different. In addition to this, for noise robustness, all the streams in the system should not undergo the same degradation in presence of noise.

3.2.3 Weights for Different Streams

In order to combine the outputs of different classifiers at posterior or likelihood-level, weights need to be assigned to the outputs of each classifier. Posterior-level combination was described in Section 3.2.1. In rest of the thesis, the term weight is associated with posterior-level combination of different classifiers.

The outputs of the classifiers trained on different feature representations must be assigned importance in accordance with the reliability of the classifiers which might be dependent on noise conditions, phoneme classes and time. Assigning appropriate weights to different streams is a critical step in multi-stream systems and if the weights are not chosen properly, the combined system might perform poorly.

The weights given to different streams (outputs of the classifiers) can be defined a-priori (static weighting) or estimated at the time of testing (dynamic weighting). In static weighting, the reliability of the streams is estimated on some development data. Static weighting may not work well if testing conditions are different from the development data conditions. In contrast, in dynamic weighting, the weights are computed at the time of testing from some reliability estimates of the individual streams. In practice, the reliability of the streams might change over time. Therefore, dynamic weighting is expected to perform better than static weighting.

In this work, we have studied a few dynamic weighting techniques which are reported in Chapter 4.

3.2.4 Combination Method

There are several methods to combine the outputs of the classifiers, and the common ones include: *majority voting*, *max*, *min*, *sum* and *product* rule (Kittler et al., 1998; Kirchhoff and Bilmes, 2000; Kuncheva, 2002). The equations we present in the following paragraphs are for posterior combination since we have used HMM/ANN systems (described in Section 2.1.2) in our studies. Similar equations were developed for likelihood combination (HMM/GMM systems) in (Hagen, 2001).

In the methods discussed below, K is the number of classes at the output of a classifier and I is the number of classifiers in multi-stream combination setup. Classifier i is trained on feature stream x_t^i and $x_t = \{x_t^1, \cdots, x_t^I\}$ is the set of all feature streams in the combination. It is assumed that all the classifiers have the same number of output classes.

Majority voting

In *majority voting*, at the output of a classifier, the class that gets the highest posterior probability is considered the right class. For I classifiers, a count is made of how many times a class is selected as the right class. The counting is done for each class. The combined probability for a class is the number of votes received by that class divided by the number of classifiers. The relevant equations are:

$$P(q_k|x_t) = \frac{\sum_{i=1}^{I} \delta_{k,i}}{I} \tag{3.2}$$

where x_t is the set of all feature streams in the combination and

$$\delta_{k,i} = \begin{cases} 1 & : \text{ if } P(q_k|x_t^i) = \max_{m=1}^{K} P(q_m|x_t^i) \\ 0 & : \text{ otherwise} \end{cases} \tag{3.3}$$

is an element of a matrix of dimension $K \times I$. The matrix has a single entry of one and the rest zeros in each column. A one in the k^{th} row of the i^{th} column indicates that classifier i had the highest probability for the class q_k. In the above equation, x_t^i represents the feature stream used for classifier i.

Max rule

In the *max* rule, the posterior probability of a class is given by the maximum over the posterior probabilities for the class from different classifiers.

$$P(q_k|x_t) = \frac{\max_{i=1}^{I} P(q_k|x_t^i)}{\sum_{m=1}^{K} \max_{i=1}^{I} P(q_k|x_t^i)} \tag{3.4}$$

Min rule

In the *min* rule, the posterior probability of a class is computed by the minimum over the posterior probabilities for the class from different classifiers.

$$P(q_k|x_t) = \frac{\min_{i=1}^{I} P(q_k|x_t^i)}{\sum_{m=1}^{K} \min_{i=1}^{I} P(q_k|x_t^i)} \tag{3.5}$$

Sum rule

In the case of multiple classifiers with posterior probabilities $P(q_k|x_t^i)$, $i = 1 \cdots I$, and $x_t = \{x_t^1, \cdots, x_t^I\}$, the combined output for the k^{th} class can be decomposed as:

$$\begin{aligned} P(q_k|x_t) &= \sum_{i=1}^{I} P(q_k, b_i|x_t) \\ &= \sum_{i=1}^{I} P(q_k|b_i, x_t) P(b_i|x_t) \\ &\approx \sum_{i=1}^{I} w_i P(q_k|x_t^i) \end{aligned} \tag{3.6}$$

where $w_i = P(b_i|x_t)$ and $\sum_{i=1}^{I} w_i = 1$. In the derivation of (3.6), we make the assumption that events b_i, denoting the occurrence of stream i, are mutually exclusive and exhaustive.

Product rule

In the case of the outputs of the classifiers being independent, we can develop the product rule as follows (Hagen, 2001):

$$\begin{aligned} P(q_k|x_t) &= \frac{P(q_k)}{p(x_t)} p(x_t|q_k) \\ &= \frac{P(q_k)}{p(x_t)} p(x_t^1, \cdots, x_t^I|q_k) \\ &\approx CP(q_k)^{1-I} \prod_{i=1}^{I} \Big[P(q_k|x_t^i)\Big] \end{aligned} \tag{3.7}$$

where C is a constant such that $\sum_{k=1}^{K} P(q_k|x_t) = 1$.

In the above equation, the weights for the outputs of each classifier is kept as 1. In the case of the weights to classifiers' outputs being different, the equation is modified to (Hagen, 2001):

$$\begin{aligned} P(q_k|x_t) &\approx \frac{P(q_k)}{p(x_t)} \prod_{i=1}^{I} \Big[p(x_t^i|q_k)\Big]^{w_i} \\ &\approx CP(q_k)^{1-\sum_i w_i} \prod_{i=1}^{I} \Big[P(q_k|x_t^i)^{w_i}\Big] \end{aligned} \tag{3.8}$$

In (3.8), the weights need not sum to one and the normalization factor is used so that the combined posteriors sum to 1 ($\sum_{k=1}^{K} P(q_k|x_t) = 1$).

Properties of combination methods

Majority voting is a *hard combination method* in which the outputs of the classifiers are converted into 'all zeros and single one', and this output is used for combination. In comparison, the rest of the combination methods are *soft combination methods* where the posterior outputs of the classifiers are used for combination.

In the maximum rule and sum rule, the output probability is high for a class if any of the classifiers used for the combination has a high probability for that class. In contrast, the minimum rule and product rule yield a high output probability for a class only if all the classifiers used for the combination have high probability for that class.

Also, the sum rule is a weighted arithmetic mean operation while the product rule is a weighted geometric mean operation.

3.3 Multi-band Combination in ASR

Multi-band speech recognition, a special case of multi-stream combination, has been studied in detail in the recent past, leading to some important contributions to the field of ASR. The approach of multi-band ASR was motivated by Fletcher's product of errors rule given by (3.1). In multi-band approaches, the full-band spectrum is divided into sub-bands and separate models are trained for features extracted from each sub-band. At the time of testing, outputs of the models (likelihoods or posteriors) are weighted and combined before sending the combined outputs for decoding. The multi-band approach is very useful in the case of band-limited noise where only a few of the sub-bands get corrupted. At the combination stage, if the outputs of the models which are corrupted by noise can be deemphasized, a better performance can be achieved compared to a full-band system.

A few important issues in multi-band systems are: a) how to define the sub-bands, b) what features to extract from the sub-bands, c) at what level combination can be performed, and d) how to combine the outputs. Some of the contributions made to multi-band ASR are listed in this section.

Multi-band combination has been studied in the framework of HMM/GMM systems (Cerisara, 1999; Okawa et al., 1998, 1999) as well as HMM/ANN systems (Mirghafori, 1999; Sharma, 1999; Dupont, 2000; Glotin, 2000; Hagen, 2001).

In most of the studies, 4 sub-bands defined by critical bands were used for band division (Okawa et al., 1999; Cerisara, 1999; Mirghafori, 1999; Hagen, 2001). In Hermansky et al. (1996) and Tibrewala and Hermansky (1997), the authors studied the effect of changing the number of sub-bands (2, 4 and 7). Similarly, in Bourlard and Dupont (1996), the effect of the number of sub-bands was studied for 3, 4 and 6 sub-bands. Increasing the number of sub-bands reduced the information content in each sub-band and the performance of individual sub-bands degraded. However, when the outputs of the sub-band models were combined, no significant difference was observed between the systems using 4 and 7 sub-bands (Tibrewala and Hermansky, 1997).

The feature representations used for the multi-band ASR included critical band energies (Hermansky et al., 1996; Bourlard and Dupont, 1996), linear prediction cepstral coefficients (LPCC) (Bourlard and Dupont, 1996), J-RASTA-LPCC (Bourlard and Dupont, 1996), PLP (Hermansky et al., 1996; Hagen, 2001) and MFCC (Hagen, 2001). In Hermansky et al. (1996), the authors reported a better performance with PLP features compared to critical band energies.

In multi-band combination approaches, the features extracted from sub-bands were either modelled jointly by a single model (feature combination) (Okawa et al., 1998) or modelled separately and then the outputs of the models were combined (likelihood or posterior combination) (Okawa et al., 1999; Cerisara, 1999; Mirghafori, 1999; Hagen, 2001). Combinations were tried at state-, phone- and syllable-level by Bourlard and Dupont (1996), and no significant difference in performance was reported among the three combining methods.

For combining the outputs of the models, different combination techniques such as SNR weighting (Bourlard and Dupont, 1996; Bourlard et al., 1996; Dupont and Bourlard, 1997; Hermansky et al., 1996; Okawa et al., 1998; Hagen, 2001) and MLP merging (Bourlard and Dupont, 1996; Bourlard et al., 1996; Dupont and Bourlard, 1997; Hermansky et al., 1996; Cerisara, 1999; Mirghafori, 1999) were tried. The merging by MLP gave good performance for matched conditions. Voicing (Berthommier and Glotin, 1999) and spatial localization (Glotin et al., 1999) cues were also explored for weighting in multi-band studies. Relative-frequency weighting was explored in (Bourlard and Dupont, 1996; Hagen, 2001). In (Okawa et al., 1998, 1999), the authors investigated information theoretic measures for combination.

An important finding of the multi-band studies was that an improved performance is achieved if the full-band system is used along with the sub-band based systems (Cerisara, 1999; Mirghafori, 1999; Hagen, 2001). In (Tibrewala and Hermansky, 1997; Hagen, 2001; Glotin, 2000), full-combination multi-stream (FCMS)[3], an elegant way of combining the full-band system to sub-band systems was suggested. In FCMS, all possible combinations of the sub-bands are considered and one model is trained for each combination. It was reported that non-adjacent sub-bands carry useful and complementary information which helps in improving the performance.

The issue of asynchrony between different sub-bands was investigated in (Cerisara et al., 2000; Mirghafori, 1999). It was reported that, in spite of added computational complexity, releasing the synchrony constraint did not give an improvement in performance (Mirghafori, 1999).

Discussion on multi-band ASR

Multi-band combination was found useful in the case of band-limited noise (Bourlard and Dupont, 1996; Hermansky et al., 1996; Tibrewala and Hermansky, 1997). However, for wide-band noise, the scheme often failed to perform better than a full-band system and the combination led to a degraded performance (Hagen, 2001; Tibrewala and Hermansky, 1997; Okawa et al., 1999). This is counterintuitive to Fletcher's product of errors rule given in (3.1). The reason for this contradiction

[3]FCMS is described in detail in Section 3.5.

between HSR and ASR is that Fletcher's product of errors rule works for optimal combination (in HSR), and in practice it is not known how to derive this optimal rule for ASR systems (Hermansky et al., 1996).

Some of the limitations of multi-band approaches are:

1. Dividing the full-band into sub-bands and processing them separately has the adverse affect that information content in each sub-band is reduced compared to the information content of the full-band (Tibrewala and Hermansky, 1997). Moreover, the spectral correlation between the sub-bands is lost while processing each sub-band separately. This loss of information generally gave a reduced accuracy for sub-band recognizers or a combination of them in clean speech (Hagen, 2001).

2. The noises present in an environment are often wide-band and, as discussed earlier, multi-band systems have not yielded any improvement for speech corrupted by wide-band noises. In fact, the performance was generally poor compared to a full-band recognizer (Tibrewala and Hermansky, 1997; Okawa et al., 1999).

3. The choice of number of sub-bands and their positions is still an open issue in multi-band systems. Usually no straightforward rule is available to define the best setup.

4. In multi-band studies, it has been shown by Hermansky et al. (1996) that cepstral features yield better performance than critical band energy features. However, it is difficult to conclude which features must be extracted from each sub-band to obtain the best possible performance (Hagen, 2001).

3.4 Multi-stream Combination in ASR

In the multi-stream approach, evidences from various sources of information are combined to achieve a better performance. In general, in multi-stream combination, the information sources may include the following:

1. Feature representations from different modalities, or

2. Different feature representations from the same modality, or

3. Different classifiers (classifiers having different architecture or the same architecture but with a different number of parameters and/or initialization) trained on the same feature stream or different feature streams.

In ASR, for example, the combination of PLP, MFCC and J-RASTA-PLP features was investigated in (Hagen et al., 2000). In (Hagen and Bourlard, 2000), the authors employed features obtained from different time-scales for the combination. The multi-stream approach was used for reverberant speech in (Shire, 2001) where two models were trained for different noise conditions and their outputs were combined at the time of testing. Multi-stream combination was also pursued in audio-visual ASR where visual features derived from lip movement were combined with audio features (Tomlinson et al., 1996; Dupont and Luettin, 1998; Heckmann et al., 2002; Bengio, 2003). In (Bengio, 2003), asynchrony between the streams when streams are from different modalities was explored.

In this section, we discuss a few of the contributions made to multi-stream combination for the ASR task.

Feature representations in multi-stream ASR

In (Antoniou and Reynolds, 2000), the authors used different feature representations (MFCC, PLP and LPC) to train separate MLPs for each phoneme class and the outputs of these MLPs were combined using another set of MLPs. Similarly, PLP, J-RASTA-PLP and MFCC features were investigated for multi-stream studies in (Christensen et al., 2000; Hagen, 2001), where separate MLPs were trained for each feature representation (HMM/ANN system) and outputs of the MLPs were combined. New features such as phase-autocorrelation (PAC) (Ikbal et al., 2003b,a), spectro-temporal activity pattern (STAP) (Ikbal et al., 2004a), modulation-filtered spectrogram[4] (Wu et al., 1998a,b; Kirchhoff, 1998; Hermansky and Sharma, 1998; Janin et al., 1999; Shire, 2001), articulatory features (place and manner of articulation) (Kirchhoff, 1998) and TempoRAl Patterns[5] (Sharma et al., 2000) have also been studied in the framework of multi-stream ASR. Features obtained from different time-scales (Hagen and Morris, 2000; Hagen and Bourlard, 2000) were also explored. It was observed that the feature representations considered for combination were the primary reason for improved performance in multi-stream combination compared to the size or initialization of the neural network model (Antoniou and Reynolds, 2000; Christensen et al., 2000). It indicates

[4]MSG (Greenberg and Kingsbury, 1997).
[5]TRAPs (Hermansky and Sharma, 1998).

that each feature representation captures different characteristics of the speech signal and hence different feature representations carry some complementary information.

Neural network size and initialization in multi-stream ASR

The effect of the size and initialization of neural network was studied in (Janin et al., 1999; Antoniou and Reynolds, 2000; Christensen et al., 2000). The conclusion of the studies was that size and initialization of the neural network were less important in improving the performance compared to different feature representations used for training the networks. A similar observation is reported in Section 6.2.2 of this thesis where two MLPs of different sizes are trained on the same feature stream and the outputs of the MLPs are combined by an oracle.

In (Zhu et al., 2005a), the authors studied 4-layered MLPs in hybrid HMM/ANN systems and combined the posterior outputs of different MLPs to improve the performance in a large vocabulary conversational ASR task.

Multi-condition training in multi-stream ASR

Multi-condition training in the framework of multi-stream combination for reverberant speech was investigated in (Shire and Chen, 2000). For the same feature representation, one MLP was trained for each environmental condition. RASTA-PLP and RASTA-PLP with different linear-discriminant analysis (LDA) filters were used as two feature streams to carry out the tests. Single-stream and multi-stream combination setups were studied. For multi-stream combination, the posterior outputs of the MLPs trained on different conditions for the same feature representation were merged at the frame-level. It was reported that training on one condition and testing on the other degraded the performance. Also, combining the outputs of the MLPs (one trained on the clean condition and two others on noisy conditions but at different levels) gave a performance which was inbetween the performance of the matched condition and mismatched conditions. In the study, the average, log average and MLP mergers were investigated to combine the posterior outputs of different MLP classifiers (Section 3.2.1, Fig. 3.2). It was reported that RASTA-PLP performed well for less noisy conditions and RASTA-PLP with LDA filters gave better performance for high noise conditions.

In (Shire, 2001), the authors used two different feature representations for the multi-stream setup. PLP and MSG (Greenberg and Kingsbury, 1997) features were considered for combination, training one MLP for each feature representation. Also, for each feature representation, one separate MLP was trained for each noise condition. The outputs of MLPs trained on different feature representations were combined. The authors reported a better performance on mismatched conditions and slightly inferior performance on matched conditions compared to the baseline systems trained only on matched conditions.

Tandem system for multi-stream

In (Sharma et al., 2000), PLP, RASTA-PLP-like features with LDA, MSG and TRAP (Hermansky and Sharma, 1998) were considered as feature streams. Except PLP features, which are short-time features, the rest of the features are obtained over 1 sec windows to have complementary information between the streams considered for combination. An MLP was trained for each feature stream and the outputs of the MLPs before the softmax nonlinearity were obtained (Hermansky et al., 2000)[6]. The outputs of the MLPs trained on different feature streams were combined by averaging (Section 3.2.4, (3.6), sum rule with equal weights). The combined output was then decorrelated by principal component analysis (PCA) and given as a feature vector to an HMM/GMM system. Several combinations of feature streams were tested, and an improved performance was achieved on the AURORA (Hirsch and Pearce, 2000) task compared to the baseline MFCC features.

Using the Tandem system, our partners in the Defense Advanced Research Projects Agency (DARPA) Effective, Affordable, Reusable Speech-to-Text (EARS) project at International Computer Science Institute (ICSI), Berkeley, studied multi-stream combination for combining several feature streams for a large vocabulary conversational speech recognition task (Zhu et al., 2004; Chen et al., 2004; Zhu et al., 2005a,b). In these studies, the outputs of MLPs trained on separate feature streams were combined by several weighting methods, including the inverse entropy weighting (Misra et al., 2003) explained in Section 4.2 of this thesis. The feature streams used were short-time PLP features and a variation of long-term TRAP features, and an improved performance was reported on the task using inverse entropy weighting method.

In (Ikbal et al., 2004c) and (Misra and Bourlard, 2005), the authors reported an improved performance using PAC (Ikbal et al., 2003b) and spectral entropy features (Misra et al., 2004, 2005a) respectively, in a Tandem setup. The spectral entropy features are described later in this thesis in Section 5.2

[6]The Tandem framework is explained in detail in Chapter 8.

Combination level

In (Kirchhoff et al., 2000), the combination was considered at three levels, the feature-, state- and word-level. The state-level combination yielded the best performance for a large vocabulary ASR task using an HMM/GMM system. In the HMM/ANN framework, the authors reported the combination of the output posteriors of the MLPs to be better than feature-level combination (Kirchhoff and Bilmes, 2000).

In (Wu et al., 1998a,b), the authors studied frame-, syllable- and utterance-level combinations. In frame-level combination, the posteriors at the output of the MLPs were multiplied. HMM-recombination was utilized to do syllable-level combination, and for utterance-level combination, merging and rescoring was carried out on N-best list. The combination at syllable-level gave the best performance, closely followed by frame-level combination.

In (Ellis and Bilmes, 2000), feature-level and posterior-level combinations were analyzed. It was argued that conditional mutual information (CMI) between the feature streams, given the knowledge of the correct class, can estimate the amount of information content common in the two feature streams. The authors hypothesized that if the CMI between two feature streams was high, they were better suited for feature-level combination. The experiments on the Aurora database with matched conditions and multi-condition training could support this conjecture only weakly. A consistent relationship could not be established between higher CMI and better performance improvement by feature-level combination. The authors found that combination at the posterior-level was more suitable for feature streams which were very different. Further, the best results were obtained by a mix of feature and posterior-level combinations.

Studies on audio-visual ASR

In the recent past, several positive contributions have been made to the field of audio-visual ASR (Tomlinson et al., 1996; Dupont and Luettin, 1998; Rogozan and Deléglise, 1998; Teissier et al., 1999; Glotin and Berthommier, 2000; Heckmann et al., 2001, 2002; Bengio, 2003). The studies pointed out that combination of features from audio and visual streams make the ASR systems more robust. Especially in presence of noise, when the audio stream gets heavily corrupted, combining the visual stream can lead to a much better performance. The audio stream in most of the studies was represented by cepstral features and information from the shape of the lips was used as visual features.

Discussion on multi-stream ASR

The main observations of the studies on multi-stream combination are as follows:

1. The choice of feature streams considered for combination is very important, and the feature streams must carry complementary information to yield an improved performance when the outputs of the classifiers trained on the feature streams are combined.

2. Weights chosen and combining strategy play an important role.

3. Posterior and state-level combinations usually work better than feature-level combination.

4. In HMM/ANN systems, the size of the network (number of parameters) does not play an important role. The general observation is that if the network size is reasonable, changing its size does not alter the performance significantly. The feature representations and the methods considered for combination are the most important factors in improving the ASR performance (Janin et al., 1999; Antoniou and Reynolds, 2000; Christensen et al., 2000).

3.5 Full-combination Multi-stream ASR

In full-combination multi-stream (FCMS) ASR, more than one feature representation is extracted from the speech signal and every possible combination of the feature representations is treated as a separate feature stream (Morris et al., 2001; Hagen and Morris, 2005). Fig. 3.4 illustrates FCMS framework for hybrid HMM/ANN systems. In the figure, one multi-layered perceptron (MLP) with single hidden layer is trained for each such feature stream. In (Hagen and Morris, 2000; Misra et al., 2003), the authors used cepstral coefficients, delta cepstral coefficients and delta-delta cepstral coefficients as separate feature streams. An improvement over simple concatenation of the three feature streams was observed when these feature streams were used in FCMS.

3.6 Database and the Experimental Setup

In Chapters 4, 5, 6 and 7 of this thesis, the results are reported on the Numbers95 (Cole et al., 1995) connected digit task. Results on a large vocabulary conversational telephone speech (CTS) database are reported in Chapter 8, and the CTS database is described in the same chapter.

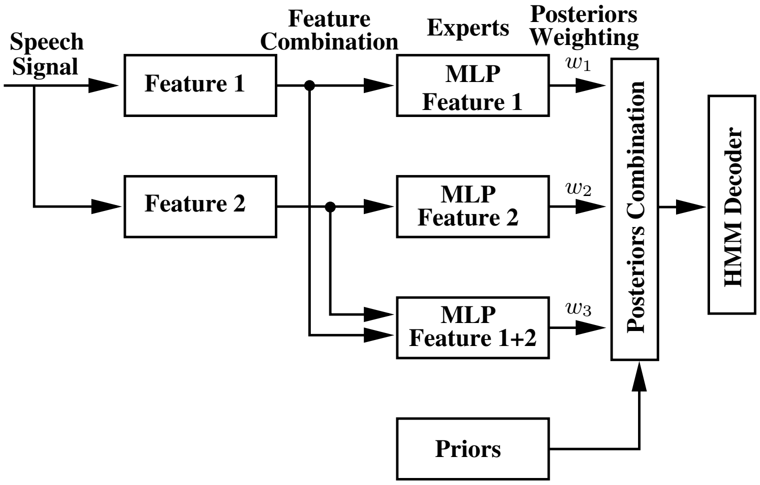


Figure 3.4. *Full-combination multi-stream for a hybrid HMM/ANN system: All possible combinations of the two feature representations are treated as separate feature streams and an MLP is trained for each feature stream. The posteriors at the output of the MLPs are weighted and combined. The combined posteriors thus obtained are passed to an HMM decoder to generate text output.*

3.6.1 Numbers95 Database

Numbers95 contains digit strings spoken in US English over a telephone channel and is a small vocabulary database. There are 30 word types in the database modelled by 27 context-independent phones. The training set consists of 3330 sentences (2996 sentences were used for training the MLP and 304 for cross-validation) and 2250 sentences were used for testing. Training was performed on clean speech only. The lexicon had a single pronunciation per word.

In this thesis, to study robustness towards additive noise, the factory and lynx noise data from the Noisex92 (Varga et al., 1992) database was added to the clean speech test utterances of the Numbers95 database at different SNRs. Factory noise is a wide-band noise recorded in a factory environment while lynx noise is collected from a running helicopter environment. In addition, we also investigated the ASR performance in presence of additive car noise (provided by Daimler Benz). The car noise was collected inside a car running at 120 km/h with closed windows.

3.6.2 System Details

We used a hybrid HMM/ANN system for our multi-stream studies presented in Chapters 4, 5 and 6. The hybrid system is preferred for the following reasons:

1. Discriminative training is possible in the HMM/ANN architecture.

2. The outputs of the MLP in a hybrid system are estimate of posterior probabilities for the

classes and each of them ranges between 0 and 1 (for each frame, the sum of output posterior probabilities is 1). The posteriors give an indication of how confident the MLP classifier is about a particular class. In contrast, the likelihoods of an HMM/GMM system are dependent upon the dimension of the input feature vector and do not have an upper bound. In multi-stream ASR, it is easier to combine the posterior probabilities that are constrained between 0 and 1 compared to likelihoods that have a high dynamic range.

3. Correlation between different components of the feature space can be learned by an MLP, therefore no strong assumption about the statistical distribution of the acoustic space is required. In contrast, the features need to be decorrelated in HMM/GMM systems because of the usual assumption of diagonal covariance matrix.

4. In the case of less data, better robustness can be achieved by an HMM/ANN system (Bourlard and Morgan, 1994).

In our studies, the input to the MLP is 9 frames of feature vectors (4 consecutive frames on either side of the center frame) and the targets of the MLP are phoneme labels (Bourlard and Morgan, 1994). One-hot-encoding has been used for training the MLP. We worked with context-independent phone models, that is, there was one output node for each phoneme (or class) and the number of nodes in the outer layer was same as the number of phonemes. The activation function for nodes in the output layer was softmax. The hidden layer had a number of nodes proportional to the input feature vector dimension[7].

A Viterbi decoder was used for generating the output text sequences. The HMM for each context-independent phone had a single state model. The minimum duration of each phone was enforced by repeating 1 to 3 states of the same model with a transition probability of 0.5 to the same state and 0.5 to the next state. The posteriors obtained at the output of the MLP were divided by the prior probability of the respective phonemes, and the obtained scaled likelihoods were used as emission probabilities in the HMMs.

In the Numbers95 ASR task, where any number can follow any other number, a uniform language model having equal probability for all possible word transitions is sufficient. The *phone deletion penalty* factor was found empirically for the clean test conditions for every setup such that the *number of insertions* was same as the *number of deletions*. This factor was not changed while

[7] If the number of nodes are reasonable, changing them does not change the performance of the system significantly (Janin et al., 1999; Antoniou and Reynolds, 2000; Christensen et al., 2000).

doing experiments on noisy conditions for the same setup.

In Chapter 7, we introduce the Tandem system. The MLP of the Tandem system was the same as that of the hybrid system. The HMM/GMM part of the Tandem system consists of 80 context-dependent phones with 3 left-to-right states per context-dependent phone. For each state, emission probabilities were estimated by a mixture of 12 Gaussians. We used HTK (Young et al., 1997) for the HMM/GMM system.

3.6.3 Performance Evaluation: Statistical Significance Test

We used the test suggested in (Bisani and Ney, 2004) to measure the statistical significance of the difference between the performance of two speech recognizers. In this test, the total number of word errors (insertions, substitutions and deletions) of both the recognizers are obtained for each test utterance. The difference between the errors by two recognizers are listed for each utterance along with the total number of words in the utterance. Resampling this data (for each utterance: difference of errors vs number of words in the utterance) several times using bootstrap, we obtain an estimate of the difference in average errors of the two recognizers for each resampling. A distribution of average errors is plotted and depending upon where 0 average error difference point lies, the statistical significance between the errors of two speech recognizers is determined.

Chapter 4

Multi-stream ASR: Weighting Techniques

In a multi-stream system, a stream (outputs of the classifier) should be assigned a weight according to the relative reliability of its information content. A stream which is more reliable should get more weight and vice-versa. The ideal system would be the one where the "right" stream gets a weight of 1 and rest of the streams get a weight of 0. However, in practical situations, it is not possible to judge which stream is "right". Thus, assigning proper weights to different streams is an integral and important issue in multi-stream systems.

In a multi-stream setup, we can assign weights to streams according to one of the following criteria:

1. same weight to all the classes of a particular classifier i which changes with time t (w_t^i), or

2. different weights to different classes k of every classifier (w_k^i), or

3. different weights to different classes of every classifier which change with time ($w_{t,k}^i$).

In the work reported here, we have investigated examples of all three weighting strategies mentioned above.

In the present chapter, we propose and investigate some weighting techniques for combining different streams. The individual feature representations considered for multi-stream combination in this chapter are 13 PLP-derived cepstral coefficients (c), 13 delta cepstral coefficients (Δc) and

13 delta-delta cepstral coefficients ($\Delta\Delta c$). These three individual feature representations give rise to seven possible feature streams in a full-combination multi-stream (FCMS)[1] setup (Misra et al., 2003). One MLP classifier was trained for each feature stream.

The remaining chapter is organized as follows: in Section 4.1, we present motivation for maximum-posterior (MP) weighting, its implementation and the results obtained by it. Inverse entropy weighting and its details are presented in Section 4.2. Maximum-likelihood (ML) weighting is discussed in Section 4.3, followed by a summary of the chapter.

4.1 Maximum-Posterior (MP) Weighting

4.1.1 Motivation

In general, an MLP which is well trained in classification mode outputs a high posterior probability for one class and low posterior probabilities for the rest of the classes. This estimate of posterior probability for a phoneme, given the input feature, can be treated as a frame-level confidence measure (Bourlard and Morgan, 1994; Robinson et al., 2002). At the output of the MLP, we can find out which class has got the highest posterior probability and what that value is (maximum posterior probability, $\mathscr{P}$).

$$j = \arg\max_k \quad \{P(q_k|x_t, \theta)\} \tag{4.1}$$

$$\mathscr{P} = \max_k \quad P(q_k|x_t, \theta), \quad k = 1, \cdots, K \tag{4.2}$$

where q_k is the class (typically context-independent phone in HMM/ANN based ASR), x_t is the input feature vector to the MLP at time instant t, θ is the parameter set of the MLP and K is the total number of classes at the output of the MLP.

Fig. 4.1 illustrates the empirical relationship between *maximum posterior probability* and *number of frames correctly classified* for clean as well as different noisy test conditions. The noise conditions are simulated by adding factory noise from the Noisex92 database to the test utterances of the Numbers95 database and are represented by SNR12 and SNR6 for signal-to-noise ratio (SNR) of 12 dB and 6 dB respectively. In Fig. 4.1(a), we observe that a large number of frames are classified correctly when the maximum posterior probability is high and, when maximum posterior probability is low, only a few frames are correctly classified. Also, the curve has a peak for high maximum

[1]FCMS was described in Section 3.5.

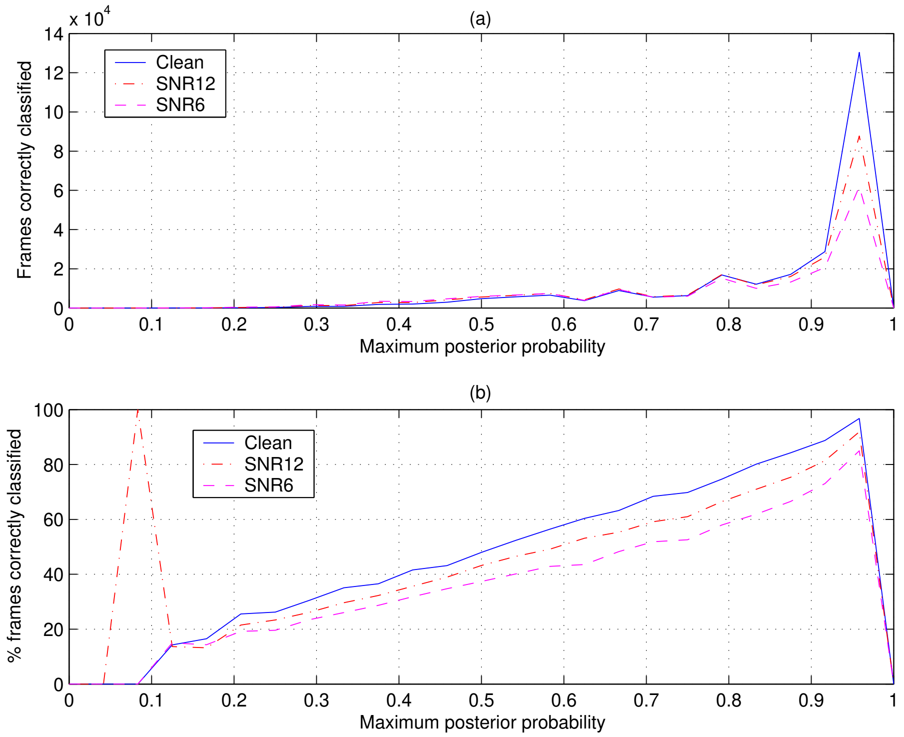


Figure 4.1. *Empirical relationship between maximum-posterior probability at the output of an MLP and: (a) the number of frames correctly classified, (b) the percentage of frames correctly classified. The plot is for an MLP trained on clean PLP features and for the following test conditions: Clean (—), SNR12 (-.-.) and SNR6 (- - -). The database for training and testing is Numbers95 and noise conditions are simulated by adding factory noise from the Noisex92 database.*

posterior probability indicating that a significant percentage of the frames have high maximum posterior probability. However, this figure does not reveal the complete information as the total number of frames for a particular maximum posterior probability is not known. Fig. 4.1(b) shows how *maximum posterior probability* is related to *percentage correct classification*. The percentage correct classification is obtained by dividing the number of frames correctly classified by the total number of frames for each maximum posterior probability value. We see that an approximately linear relationship exists over a wide region between maximum posterior probability and percentage correct classification. A similar relationship between posterior probability estimates at the output of the MLP and percentage correct classification was shown in Bourlard and Morgan (1994). The unexpected peak for noise at SNR6 for low value of maximum posterior probability is because there is only 1 frame at that value and that frame is correctly classified.

In (Robinson et al., 2002), the authors estimated the log of the posteriors and then averaged it over time for the phoneme and word durations. Robinson et al. called the estimates "acoustic confidence measures" and used them for pronunciation modelling, searching the lattice and combining the outputs of the decoders[2].

Motivated by the relationship between maximum posterior probability and percentage correct classification, we suggest a weighting technique such that the weight to the outputs of an MLP classifier (stream) is proportional to its maximum posterior probability.

4.1.2 Implementation

For each time frame t, the maximum posterior probability is found for every feature stream i at the output of its MLP

$$\mathscr{P}_t^i = \max_k \quad P(q_k|x_t^i, \theta_i), \quad k = 1, \cdots, K \tag{4.3}$$

where K is the number of output classes or phonemes, x_t^i is the input acoustic feature vector for the i^{th} MLP classifier for the t^{th} frame, θ_i is the parameter set of the i^{th} MLP, and $P(q_k|x_t^i, \theta_i)$ is the posterior probability estimate for the k^{th} class by the i^{th} MLP.

Weight, w_t^i, for the i^{th} MLP classifier is computed as

$$w_t^i = \frac{\mathscr{P}_t^i}{\sum_{j=1}^{I} \mathscr{P}_t^j} \tag{4.4}$$

where I is the total number of classifiers considered for combination.

As discussed in Section 3.2.4, the sum rule (3.6) or the product rule (3.8) can be applied to combine the weighted outputs of different classifiers.

A variation of maximum-posterior (MP) weighting could be to select the stream that has the maximum posterior probability among all the streams at each time frame t.

$$j = \arg\max_i \quad \{\mathscr{P}_t^i\} \tag{4.5}$$

We refer to this weighting as **Maximum MP**.

4.1.3 Results

The results of the MP weighting and its variation (Maximum MP) are presented in Table 4.1 in terms of word error rates (WERs). We observe that MP weighting does not help in improving the

[2] Using ROVER (Fiscus, 1997)

Word Error Rates for Maximum-Posterior Weighting				
Feature	**Clean**	**SNR12**	**SNR6**	**SNR0**
c	12.5	21.3	35.3	57.8
Δc	15.0	23.7	34.9	56.1
$\Delta\Delta c$	15.8	22.4	36.3	60.0
c,Δc	11.0	19.0	31.1	52.7
c,$\Delta\Delta c$	10.7	18.0	30.0	52.3
Δc,$\Delta\Delta c$	12.6	19.3	30.7	51.7
c,Δc,$\Delta\Delta c$ (Baseline)	**10.0**	17.7	29.6	51.0
Sum Rule				
Equal Weight	11.7	18.4	30.1	50.8
MP	11.2	17.6	29.4	50.0
Product Rule				
Equal Weight	10.2	16.4*	28.5	50.1
MP	**10.0**	**16.1***	27.8*	49.0*
Maximum MP	10.1	16.3*	**27.5***	**48.7***

Table 4.1. *Word Error Rates (WERs) in % for the 7 possible PLP streams and their combination by MP weighting. c, Δc and $\Delta\Delta c$ represent static, delta and delta-delta features respectively. The noise conditions are simulated by adding factory noise from the Noisex92 database at different SNRs to the utterances of the Numbers95 database. SNR12, SNR6 and SNR0 represent the SNR of 12, 6 and 0 dBs respectively. The numbers in* **bold** *show the best performance and * indicates that the improvement in performance compared to the baseline system is significant.*

performance when combined by the sum rule, but the product rule gives an improvement in performance. Maximum MP weighting, where we choose the stream having highest posterior probability, also helps in improving the performance. The MP with the product rule and the Maximum MP methods yield similar improvements in performance and the improvement is more for low SNR cases.

Though the improvement in performance is statistically significant[3] in a few cases, it is not very high. A possible reason for this could be that the feature streams used for training the classifiers are not carrying enough complementary information to yield an improved performance[4]. Nevertheless, even such a simple weighting improves the relative average WER performance over the PLP baseline system[5] by 4.7%.

In Table 4.1, performance is also shown for the case of equal weighting. The improvement by equal weighting is observed to be less compared to that by MP weighting. Still, equal weighting in the case of the product rule gives a relative average WER improvement of 2.7% over the baseline PLP system, showing the advantage of a multi-stream system over a single-stream system. It is

[3] Statistical significance test used in this work was explained in Section 3.6.3.

[4] In Section 5.2.5 (Page 80), Section 6.2.2 (Page 89) and Appendix D, we compare the performance of different streams/feature streams in a multi-stream setup and analyze the complementarity of streams/feature streams.

[5] In this thesis, baseline PLP features are 39 dimensional with 13Δc and 13$\Delta\Delta c$ concatenated to 13c.

interesting to note that modelling the feature streams separately and combining the outputs of the models (MLPs in HMM/ANN framework) with simple weights can bring an improvement over a baseline system where the feature streams are concatenated first and then modelled (referred as feature-level combination or early integration in Section 3.2.1). The result that posterior-level combination is better compared to feature-level combination was reported in (Kirchhoff and Bilmes, 2000) as well.

The important results of the study can be summarized as follows:

1. Combination by the sum rule gave poor results compared to combination by the product rule.

2. We observed that posterior-level combination (late integration) gives better performance than feature-level combination (early integration). A similar observation was reported in (Kirchhoff and Bilmes, 2000)

3. In posterior-level combination using the product rule, equal weights and MP weights yielded a relative average WER improvement of 2.7% and 4.7% respectively, over the baseline.

4. The improvements obtained by posterior-level combination were significant in few cases, but the average improvement was low. The reason for this could be that the feature representations used for combination did not have enough complementary information. In Section 6.2.2, we use an oracle test to indicate the complementarity of several feature streams in a multi-stream system. The oracle test indicates that the 7 PLP feature streams carry less complementary information compared to RASTA-PLP features and spectral entropy features (discussed later in this thesis in Section 5.2).

4.2 Inverse Entropy Weighting

4.2.1 Motivation

Entropy is used in information theory to measure the randomness or uncertainty of a process that has several possible outcomes (Shannon and Weaver, 1949). The outcome of a completely certain process has a probability of 1 for one event and 0 for the rest of the events. Such a process has zero entropy. In contrast, the outcome of a completely unpredictable process has equal probabilities for all the events and has maximum entropy. So entropy can tell us about the degree of uncertainty in a process.

We can measure entropy at the output of an MLP classifier from the estimates of its output a posteriori probabilities. The entropy at the output of the i^{th} MLP classifier for the t^{th} frame, h_t^i, is computed in the following way,

$$h_t^i = -\sum_{k=1}^{K} P(q_k|x_t^i, \theta_i) \cdot \log_2 P(q_k|x_t^i, \theta_i) \tag{4.6}$$

where K is the number of output classes (states or phones in an HMM/ANN system) in the MLP, x_t^i is the input acoustic feature vector for the i^{th} MLP classifier for the t^{th} frame, and θ_i is the parameter set of the i^{th} MLP classifier.

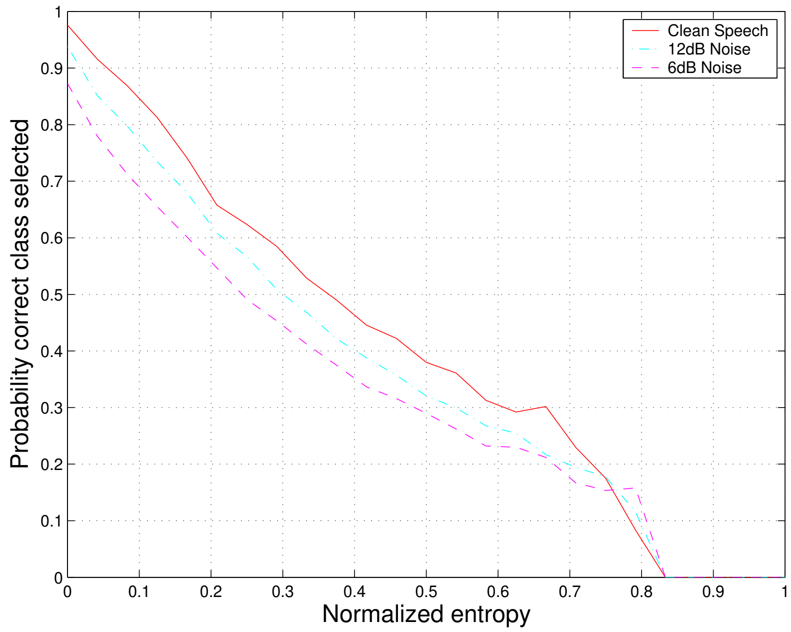


Figure 4.2. *Plot of normalised entropy vs probability that the correct class is selected. The plot is for the MLP trained on clean PLP features and for the following test conditions: Clean (—), SNR12 (-.-.) and SNR6 (- - -). The database for training and testing is Numbers95 and noise conditions are simulated by adding factory noise from the Noisex92 database to the test data.*

Fig. 4.2 shows the inverse relation between the *normalized entropy* at the output of an MLP classifier and the *percentage correct classification*. The normalized entropy is the entropy at the output of the classifier divided by the maximum possible entropy ($\log_2 K$ for the number of classes being K). The figure shows that percentage correct classification is high for low entropy and vice-versa, that is, an MLP classifier is more reliable when the entropy at its output is low and is less reliable when its output entropy is high.

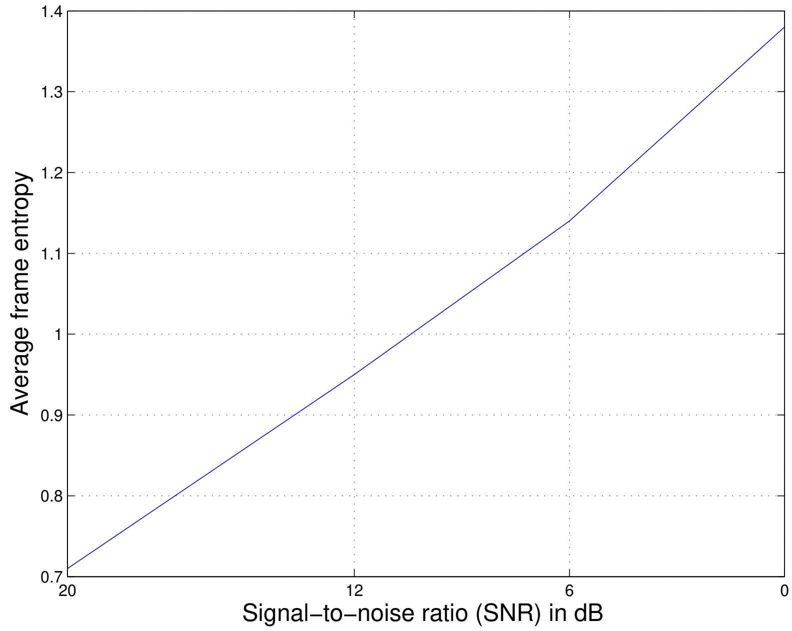


Figure 4.3. *Change in average frame entropy at the output of an MLP with increase in noise-level at the input of the MLP. The MLP was trained on features extracted from clean speech and tested on features extracted from clean as well as noisy test utterances. The Numbers95 database was used or training and testing. Noise conditions were simulated by adding factory noise from the Noisex92 database at various SNRs to the clean speech. In the plot, the clean test condition is represented by 20 dB SNR.*

In our study, we observed that if an MLP has been trained on clean speech, the average entropy (averaged over all the frames) at the output of the MLP increases in the case of noisy speech (Fig. 4.3). This indicates that for noisy speech, the discriminatory power of the MLP decreases, and the a posteriori probabilities tend to become more uniform. This mismatch between the training and testing conditions is reflected through the entropy at the output of an MLP. We have used this information in our FCMS approach for weighting the streams (outputs of different MLP classifiers).

At the time of testing, the MLP classifiers associated with the feature streams that are more corrupted by noise will face more mismatched conditions. Consequently, their respective output entropies will increase indicating that the a posteriori probabilities are approaching *equal probabilities for all the classes*. The MLP classifiers having high entropy are less discriminatory, therefore the outputs of such classifiers should be weighted less. Similarly, the MLP classifiers having low entropy will have higher discrimination among classes and their outputs should be assigned more weight.

It needs to be emphasized that though it is possible that a frame might be wrongly classified and has low output entropy also, Fig. 4.2 illustrates that such cases are in a minority. Further, entropy is used as a statistical measure, that is, it is correct most of the time. It might not be possible for any single statistical measure to define the "oracle selection" described in Chapter 6.

In the framework of multi-band ASR systems, a similar concept of entropy weighting was reported in (Okawa et al., 1998). The authors computed entropy from normalized likelihood in HMM/GMM systems to weight the likelihoods of different streams in a multi-band system. In a similar study on audio-visual ASR, features were obtained from both the modalities and one MLP was trained for each feature representation. Entropy estimates at the output of the MLPs were employed as a measure to weight the posterior outputs of different MLPs. Furthermore, the idea of entropy at the output of an MLP was explored to discriminate between speech and music in (Ajmera et al., 2003). In their work, the authors reported that an MLP trained on speech gives low entropy at the time of testing for speech and high entropy for music. In an internal email communication of the Thematic Indexing of Spoken Language (THISL) project, correlation between average entropy at the output of an MLP and WER was reported[6]. The author observed that most often WER was high when average entropy was high and WER was low when entropy was low.

4.2.2 Implementation

To achieve the idea of giving more weight to the MLP outputs which have low output entropy and vice-versa, the idea of inverse entropy weighting is investigated. The weight assigned to the outputs of the i^{th} MLP classifier, w_t^i, is given by,

$$w_t^i = \frac{1/h_t^i}{\sum_{j=1}^{I} 1/h_t^j} \tag{4.7}$$

where I is the total number of classifiers considered for combination.

The inverse entropy weights, w_t^i, can be applied to the sum rule (3.6) as well as the product rule (3.8) to combine posterior outputs of different MLP classifiers.

4.2.3 Variations of Inverse Entropy Weighting

Inverse entropy weighting with static threshold (IEWST)

In this variation, a fixed maximum threshold is chosen for the entropy (empirically optimized for clean speech and set to 1.0 in our studies). If the entropy of a particular classifier for a frame is

[6] E-mail of Daniel P. W. Ellis, 26 January 2000, THISL project

more than the threshold, the outputs of that classifier are penalized by a *static weight* proportional to $\frac{1}{10000}$ (other values of static weight gave similar performance). For the same frame, the outputs of the classifiers with entropy lower than the threshold are weighted inversely proportional to their respective entropies. The modified equations for IEWST are:

$$\widetilde{h}_t^i = \begin{cases} 10000 & : \quad h_t^i > 1.0 \\ h_t^i & : \quad h_t^i \leq 1.0 \end{cases} \tag{4.8}$$

$$w_t^i = \frac{1/\widetilde{h}_t^i}{\sum_{j=1}^{I} 1/\widetilde{h}_t^j} \tag{4.9}$$

Inverse entropy weighting with average entropy at each frame-level as threshold (IEWAT)

In this weighting scheme, the average entropy of all the classifiers for a frame is calculated by the equation,

$$\bar{h}_t = \frac{\sum_{i=1}^{I} h_t^i}{I} \tag{4.10}$$

This average entropy is used as a dynamic threshold for the frame and outputs of all the classifiers having entropy greater than the threshold are assigned a weight proportional to $\frac{1}{10000}$, whereas the outputs of the classifiers having entropy lower than the threshold are weighted inversely proportional to their respective entropies. The equations in case of IEWAT are:

$$\widehat{h}_t^i = \begin{cases} 10000 & : \quad h_t^i > \bar{h}_t \\ h_t^i & : \quad h_t^i \leq \bar{h}_t \end{cases} \tag{4.11}$$

$$w_t^i = \frac{1/\widehat{h}_t^i}{\sum_{j=1}^{I} 1/\widehat{h}_t^j} \tag{4.12}$$

Minimum entropy criterion

In this approach, at every time instant, the outputs of the classifier that has the minimum entropy are chosen and used for decoding while the outputs of the rest of the classifiers are ignored. The modified equations in this case are:

$$\widehat{P}(q_k|x_t, \Theta) = P(q_k|x_t^j, \theta_j) \tag{4.13}$$

where $\Theta = \{\theta_i, \cdots, \theta_I\}$, I is the total number of feature streams (and MLP classifiers) and $x_t = \{x_t^i, \cdots, x_t^I\}$, such that

$$j = \arg\min_i \quad \{h_t^i\} \tag{4.14}$$

In essence, minimum entropy criterion is a 1/0 weighting, where the outputs of the MLP classifier having least entropy get a weight of 1 and the outputs of rest of the classifiers are assigned a weight of 0.

4.2.4 Results

Table 4.2 shows the results for different variations of inverse entropy weighting suggested in this thesis. In the table, IEWST and IEWAT represent *inverse entropy weighting with static threshold* and *inverse entropy weighting with average entropy at each frame as threshold* respectively. The results once again show that the *product rule* gives slightly better performance compared to the *sum rule*. Moreover, among all the possible variations of inverse entropy suggested in this thesis, the *average entropy weighting (IEWAT)* works the best. The relative average WER improvement over the baseline by this method is 6.6%. As mentioned before, the reason for this small improvement could be that the individual feature streams used to train the MLPs do not have enough complementary information to yield a better performance when outputs of the MLPs are combined.

WERs for Inverse Entropy Weighting				
Feature	**Clean**	**SNR12**	**SNR6**	**SNR0**
c, Δc, $\Delta\Delta c$ (Baseline)	10.0	17.7	29.6	51.0
Sum Rule				
IEWST	10.6	17.2	28.8	49.8
IEWAT	10.0	16.2*	27.4*	**48.3***
Product Rule				
IEWST	**9.7**	16.0*	27.4*	49.0*
IEWAT	**9.7**	**15.7***	**27.3***	48.9*
Minimum Entropy	10.0	16.2*	27.7*	48.7*

Table 4.2. *WERs in % for the baseline PLP features and combination of the 7 PLP streams by inverse entropy weighting. c, Δc and $\Delta\Delta c$ represent static, delta and delta-delta features respectively. The noise conditions are simulated by adding factory noise from the Noisex92 database at different SNRs to the utterances of the Numbers95 database. The numbers in* **bold** *show the best performance and * indicates that the improvement in performance as compared to the baseline system is significant.*

4.2.5 Relationship between MP and Inverse Entropy Weightings

In the experimental studies presented in Tables 4.1 and 4.2, we observed that MP and inverse entropy weighting gave a relative average WER improvement of 4.7% and 6.6% respectively. MP weighting and inverse entropy weighting are closely related except for the fact that MP weighting relies only on the maximum posterior probability value while inverse entropy weighting captures the relation among all the posterior probabilities. Entropy, which has information about the output distribution, is a better estimate of the classifier's reliability, and this could be the reason why inverse entropy performs better than MP weighting.

In this section, we analyze the relationship between entropy and the maximum posterior probability when the maximum posterior probability ≈ 1 (which is generally the case at the output of an MLP, as shown in Fig. 4.1 (a)).

$$\begin{aligned}
h &= -\sum_{k=1}^{K} P(q_k|x_n)\log P(q_k|x_n) \\
&= -P(q_j|x_n)\log P(q_j|x_n) - \sum_{k=1,\neq j}^{K} P(q_k|x_n)\log P(q_k|x_n) \quad \left[P(q_j|x_n) = \max_k P(q_k|x_n); P(q_j|x_n)\approx 1\right] \\
&\approx -(1-y)\log(1-y) \quad \left[P(q_j|x_n) = 1-y;\ K >> 2;\ q_k \approx 0\right] \\
&\approx -(1-y)(-y-y^2/2-y^3/3) \quad \left[\text{truncating the Taylor series}\right] \\
&\approx y \quad \left[y\approx 0 \text{ and neglecting higher power of y}\right] \\
&\approx 1-P(q_j|x_n) \qquad (4.15)
\end{aligned}$$

It is observed that the relationship between maximum posterior probability (≈ 1) and entropy is linear and inverse. But the relationship holds under the assumptions that K (number of classes) is large and one of the classes has posterior probability close to 1 (the rest of the classes have posterior probability close to 0). The empirical and the approximate relationship given by (4.15) between *maximum posterior probability* and *entropy* is shown in Fig. 4.4 for $K = 27$. We have used the MLP trained on the Numbers95 database to plot the empirical curve. For the values of maximum posterior probability between 0.7 and 0.99, the approximate curve closely follows the empirical curve. It shows that MP and inverse entropy weighting methods are closely related.

The two weighting methods discussed above, namely MP and inverse entropy, are of the kind w_t^i where the same weight is given to all the classes at the output of a classifier and the weight changes

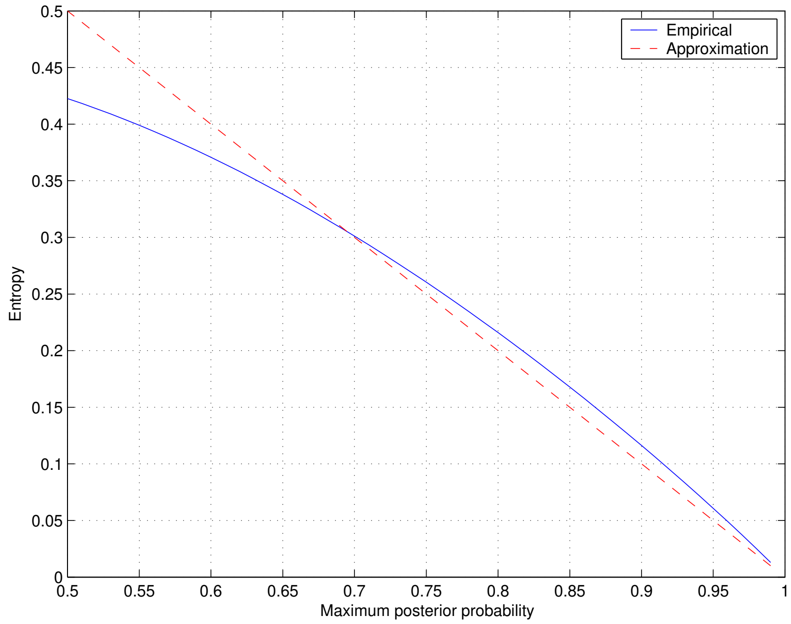


Figure 4.4. *Relationship between maximum-posterior probability and entropy at the output of an MLP for number of classes (K) equal to 27. The MLP was trained on the Numbers95 database. Solid line (—) is the empirical relationship obtained from the data and dashed line (- - -) is the relationship developed by the approximation of* (4.15).

with time.

4.2.6 Discussion: Entropy at a Classifier Output

In Shannon and Weaver (1949, Page 19), we find the following statement:

Suppose for the moment that one knows that a certain signal symbol has actually been received. Then each message symbol takes on a certain probability – relatively large for the symbol identical with or the symbols similar to the one received, and relatively small for all others. Using this set of probabilities, one calculates a tentative entropy value. This is the message entropy on the assumption of a definite known received or signal symbol. Under any good conditions its value is low, since the probabilities involved are not spread around rather evenly on the various cases, but are heavily loaded on one or a few cases. Its value would be zero in any case where noise was completely absent, for then, the signal symbol being known, all message probabilities would be zero except for one symbol (namely the one received), which would have a probability of unity.

We can consider a trained MLP as a channel with feature representation (extracted from signal) as input and probabilities for each class at its output. An MLP trained using cross-validation data as stopping criterion does not give 100% correct classification even for the training data, and acts like a noisy channel. Entropy at the output of an MLP indicates how good the classification abilities of the MLP are for a given input. An MLP whose input feature representation is less noisy will have lower entropy at its output compared to another MLP which has a feature representation more affected by noise at its input.

Based on this principle, inverse entropy seems to be a good weighting method and experimental results confirm this reasoning. This observation is further supported in Section 6.2.3 where we investigate the relationship between "oracle selection" and inverse entropy weighting.

4.3 Maximum Likelihood (ML) Weighting

4.3.1 Motivation

The idea of the ML weighting originates from the expectation maximization (EM) algorithm used for training Gaussian mixture models (GMMs) and hidden Markov models (HMMs). In the EM algorithm, we try to maximize the likelihood of the data and in the process estimate the hidden parameters of the models which will maximize the likelihood.

Based on the same principle, in ML weighting, the goal is to find the weights assigned to each stream (outputs of the classifier) such that the likelihood of the combined data increases. In the ML weighting proposed in this thesis, we increase the likelihood of the test data and we do not have any segment information (phoneme labels) available for the data.

At this juncture, we would like to compare the proposed ML weighting with common ML approaches used for training or adaptation.

1. It is different from the ASR training approach discussed in Chapter 2 where the training data is provided along with its segmental information. The iterative Baum-Welch algorithm (Baum et al., 1970) is applied to train the HMM parameters such that the likelihood of the training data is increased from one iteration to another.

2. It is different from techniques used for model adaptation. In maximum-likelihood linear regression (MLLR) (Leggetter and Woodland, 1995), trained models are tested on some development data and the segmentation of the development data is obtained. Based on some confidence measure, the correctness of the segmentation is established. The segments which are recognized with high confidence are used to adapt the trained model. The adaptation procedure is similar to ML training employed for training the ASR models. The models are adapted

(only the means of the Gaussians are modified) such that the likelihood of the development data is increased. The adapted models are used for ASR.

Though ASR training as well as MLLR adaptation increase the likelihood of the data, they are different from the ML weighting suggested for multi-stream combination in this thesis. In our case, test data is employed for finding the weights (hidden parameters of ML weighting) given to outputs of different classifiers such that the likelihood of the test data is increased. It is expected that the combined data with higher likelihood will yield better discrimination between the classes.

4.3.2 Derivation

EM for Multi-stream ASR: HMM Framework

An HMM has three parts, *Initial* parameters. *Emission* parameters and *Transition* parameters.

The corresponding equation to compute the likelihood of the HMM, given the data, is:

$$\begin{aligned} p(X, Q|\theta) \quad = \quad & \prod_{k=1}^{K} P(q_1 = k)^{z_{k,1}} \\ & \prod_{t=1}^{T} \prod_{k=1}^{K} \left[\left(\prod_{i=1}^{I} p(x_t, b_i | q_t = k, \theta_i)^{z_{i,k,t}} \right) \right. \\ & \left. \prod_{m=1}^{K} P(q_t = k | q_{t-1} = m)^{z_{k,t} \cdot z_{m,t-1}} \right] \end{aligned} \tag{4.16}$$

In (4.16), the first term is initial-state probabilities, the second term is emission probabilities, and the last one is transition-state probabilities. In the above equation, X represents the feature vector sequence obtained from test utterance, Q is the set of all possible state sequences in a fully connected HMM, θ is the set of parameters of the model, b_i is the indicator variable to represent different classifiers, θ_i is the set of MLP parameters for classifier i, K is the number of states (phones or classes in HMM/ANN system), I is the total number of classifiers and T is the number of frames in the utterance. $z_{k,1}$, $z_{i,k,t}$ and $z_{m,t-1}$ are the indicator variables for initial-state probabilities, emission probabilities and state-transition probabilities respectively, and can take the value of either 0 or 1 depending on whether the event occurs or not.

In (4.16), the second term (related to emission probabilities) can be expanded as,

$$
\begin{aligned}
p(X,Q|\theta) \quad = \quad & \prod_{k=1}^{K} P(q_1=k)^{z_{k,1}} \\
& \prod_{t=1}^{T}\prod_{k=1}^{K}\Bigg[\bigg(\prod_{i=1}^{I}\Big[P(b_i|q_t=k,\theta_i)\;\; p(x_t|b_i,q_t=k,\theta_i)\Big]^{z_{i,k,t}}\bigg) \\
& \prod_{m=1}^{K} P(q_t=k|q_{t-1}=m)^{z_{k,t}\cdot z_{m,t-1}}\Bigg]
\end{aligned}
\tag{4.17}
$$

The second term now suggests that the emission probabilities (scaled likelihoods in our HMM/ANN case) from individual classifiers, $p(x_t|b_i, q_t = k, \theta_i)$, are weighted to give the combined likelihood. The weights (hidden parameters of ML) are given by $P(b_i|q_t = k, \theta_i)$, and the aim of ML weighting is to find the weights that maximize the combined likelihood.

Taking $\log$ on both the sides,

$$
\begin{aligned}
\log p(X,Q|\theta) \quad = \quad & \sum_{k=1}^{K} z_{k,1}\log P(q_1=k) + \\
& \sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{i=1}^{I} z_{i,k,t}\Big[\log p(x_t|b_i,q_t=k,\theta_i)+\log P(b_i|q_t=k,\theta_i)\Big] + \\
& \sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{m=1}^{K} z_{k,t}\cdot z_{m,t-1}\log P(q_t=k|q_{t-1}=m)
\end{aligned}
\tag{4.18}
$$

Instead of solving (4.18), we can define an auxiliary function $A(\theta, \theta^s)$ as,

$$A(\theta,\theta^s) = E_Q\Big[\log p(X,Q|\theta)|X,\theta^s)\Big] \tag{4.19}$$

where, $E_Q[\cdot]$ is the expectation over Q. We can show that if A is maximized, it leads to maximization in likelihood of $p(X|\theta^{s+1})$[7], that is,

$$\theta^{s+1} = \arg\max_{\theta} \quad \{A(\theta,\theta^s)\} \tag{4.20}$$

[7] See Appendix A for proof.

The auxiliary equation for (4.18) is given by,

$$\begin{aligned} A(\theta,\theta^s) &= \sum_{k=1}^{K} E_Q[z_{k,1}|X,\theta^s]\log P(q_1=k) + \\ & \sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{i=1}^{I} E_Q\big[z_{i,k,t}|X,\theta^s\big]\Big[\log p(x_t|b_i,q_t=k,\theta_i)+\log P(b_i|q_t=k,\theta_i)\Big] + \\ & \sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{m=1}^{K} E_Q[z_{k,t}\cdot z_{m,t-1}|X,\theta^s]\log P(q_t=k|q_{t-1}=m) \end{aligned} \tag{4.21}$$

As mentioned earlier, we want to combine the *emission probabilities* to maximize the likelihood. In order to achieve that, we are required to solve the second term of (4.21) which has three parts in it, the posterior part $\big(E_Q\big[z_{i,k,t}|X,\theta^s\big]\big)$, emission probability part $\big(\log p(x_t|b_i,q_t=k,\theta_i)\big)$ and the weight part $\big(\log P(b_i|q_t=k,\theta_i)\big)$.

First, we will compute the posterior for the second term $(E_Q\big[z_{i,k,t}|X,\theta^s\big]$. Also, we replace the sum of the first and last term in (4.21) by a constant, $Constant$.

Forward and Backward variables

Before computing the posterior, we will introduce two variables from the HMM theory, *forward variable* given by α and *backward variable* given by β. In the following derivation[8], the parameter θ_i has been dropped for convenience and Markov model is assumed to be of order 1.

$\alpha(k,t)$ is defined as *the likelihood of having generated the sequence* $x_1^t=\{x_1,\cdots,x_t\}$ *and being in state* k *at time instant* t. It is given by,

$$\begin{aligned} \alpha(k,t) &= p(x_1^t,q_t=k) \\ &\approx \left(\sum_{i=1}^{I} p(x_t|b_i,q_t=k)\ P(b_i|q_t=k)\right)\left(\sum_{m=1}^{K} P(q_t=k|q_{t-1}=m)\ \alpha(m,t-1)\right) \end{aligned} \tag{4.22}$$

$\beta(k,t)$ is defined as *probability to generate the rest of the sequence* $x_{t+1}^T=\{x_{t+1},\cdots,x_T\}$ *given that we are in state* k *at time instant* t. It is given by,

$$\begin{aligned} \beta(k,t) &= p(x_{t+1}^T|q_t=k) \\ &\approx \sum_{m=1}^{K}\left(\sum_{i=1}^{I} p(x_{t+1}|b_i,q_{t+1}=m)\ P(b_i|q_{t+1}=m)\right)\beta(m,t+1)\ P(q_{t+1}=m|q_t=k) \end{aligned} \tag{4.23}$$

[8] Refer to Appendix B for the derivation.

Estimation Step: Compute the posterior

Using (4.22) and (4.23), we can estimate $z_{i,k,t}$ as follows:

$$\begin{aligned}
E_Q[z_{i,k,t}|X,\theta^s] &= P(q_t = k, b_i|X,\theta^s) &(4.24)\\
&= P(q_t = k|X,\theta^s)\ P(b_i|q_t = k, X, \theta^s)\\
&= \frac{p(q_t = k, x_1^T)}{p(x_1^T)}\ P(b_i|q_t = k, X, \theta^s)\\
&= \frac{p(x_1^t, q_t = k)p(x_{t+1}^T|q_t = k)}{p(x_1^T)}\ P(b_i|q_t = k, X, \theta^s)\\
&= \frac{\alpha(k,t)\beta(k,t)}{\sum_{m=1}^{K}\alpha(m,T)}\ P(b_i|q_t = k, X, \theta^s) &(4.25)
\end{aligned}$$

In the equation, X is the whole data and θ^s is the set of parameters of the models.

Maximization Step: Compute the weights

From (4.21), replacing the sum of first and the last term by $Constant$

$$\begin{aligned}
A(\theta,\theta^s) &= Constant + \sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{i=1}^{I} P(q_t = k, b_i|X,\theta^s)\Big[\log p(x_t|b_i, q_t = k, \theta_i)\\
&\quad + \log P(b_i|q_t = k, \theta_i)\Big] &(4.26)
\end{aligned}$$

In (4.26), we need to find the weights which will maximize the likelihood of the auxiliary function. Therefore, we differentiate $A(\theta,\theta^s)$ with respect to $P(b_i|q_t = k,\theta)$, with the constraint that $\sum_{i=1}^{I} w_k^i = \sum_{i=1}^{I} P(b_i|q_t = k,\theta_i) = 1$, $\forall$ k.

$$\begin{aligned}
\frac{\partial}{\partial P(b_i|q_t = k,\theta_i)} \Bigg[\sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{i=1}^{I} P(q_t = k, b_i|X,\theta^s)\Big[\log p(x_t|b_i, q_t = k, \theta_i) \quad +\\
\log P(b_i|q_t = k,\theta_i)\Big] + \lambda(\sum_{i=1}^{I} P(b_i|q_t = k,\theta_i) - 1)\Bigg] \quad = \quad 0 \qquad (4.27)
\end{aligned}$$

In (4.27), λ is the Lagrange multiplier required to satisfy the constraint $\sum_{i=1}^{I} P(b_i|q_t = k,\theta_i) = 1$. Solving (4.27), we get

$$\sum_{t=1}^{T} P(q_t = k, b_i|X,\theta^s)\frac{1}{P(b_i|q_t = k,\theta_i)} + \lambda = 0 \qquad (4.28)$$

Summing both the variables over all i, we get $\lambda = -\sum_{i=1}^{I}\sum_{t=1}^{T} P(q_t = k, b_i|X, \theta^s)$. Substituting λ back in (4.28), we get,

$$P(b_i|q_t = k, \theta_i) = \frac{\sum_{t=1}^{T} P(q_t = k, b_i|X, \theta^s)}{\sum_{i=1}^{I}\sum_{t=1}^{T} P(q_t = k, b_i|X, \theta^s)} \tag{4.29}$$

$$= \frac{\sum_{t=1}^{T} \frac{\alpha(k,t)\beta(k,t)}{\sum_{m=1}^{K}\alpha(m,T)} \cdot P(b_i|q_t = k, X, \theta^s)}{\sum_{i=1}^{I}\sum_{t=1}^{T} \frac{\alpha(k,t)\beta(k,t)}{\sum_{m=1}^{K}\alpha(m,T)} \cdot P(b_i|q_t = k, X, \theta^s)} \tag{4.30}$$

In the derivation, (4.30) is obtained by substituting (4.25) in (4.29). Still, $P(b_i|q_t = k, X, \theta^s)$ cannot be solved directly. We modify it in terms of known variables as follows:

$$P(b_i|q_t = k, X, \theta^s) = \frac{p(X|b_i, q_t = k, \theta^s)\ P(b_i, q_t = k, \theta^s)}{p(q_t = k, X, \theta^s)} \tag{4.31}$$

Making the assumption of HMM that the emission probability at time instant t depends only on the state of HMM at that instant, that is, $p(x_t|q_t = k)$ is independent of any other variable, $p(X|b_i, q_t = k, \theta^s) = p(x_t|b_i, q_t = k, \theta^s)$.

$$\begin{aligned} P(b_i|q_t = k, X, \theta^s) &\approx \frac{p(x_t|b_i, q_t = k, \theta^s)\ P(b_i, q_t = k, \theta^s)}{\sum_{j=1}^{J} p(X, b_j, q_t = k, \theta^s)} \\ &\approx \frac{p(x_t|b_i, q_t = k, \theta^s)\ P(b_i|q_t = k, \theta^s)}{\sum_{j=1}^{J} p(x_t|b_j, q_t = k, \theta^s) P(b_j|q_t = k, \theta^s)} \\ &\approx \frac{P(q_t = k|x_t, b_i, \theta^s)\frac{p(x_t|b_i,\theta^s)}{P(q_t=k|b_i,\theta^s)}\ P(b_i|q_t = k, \theta^s)}{\sum_{j=1}^{J} P(q_t = k|x_t, b_j, \theta^s)\frac{p(x_t|b_j,\theta^s)}{P(q_t=k|b_j,\theta^s)}\ P(b_j|q_t = k, \theta^s)} \end{aligned} \tag{4.32}$$

Now we make the following assumptions: $P(q_t = k|b_i)$ is independent of b_i (phoneme prior probability is independent of classifier), $p(x_t|b_i)$ is independent of b_i (likelihood of observation at time instant t is independent of classifier). Using these assumptions, (4.32) gets simplified to:

$$P(b_i|q_t = k, X, \theta^s) \approx \frac{P(q_t = k|x_t, b_i, \theta^s) \cdot P(b_i|q_t = k, \theta^s)}{\sum_{j=1}^{I} P(q_t = k|x_t, b_j, \theta^s) \cdot P(b_j|q_t = k, \theta^s)} \tag{4.33}$$

$$= w_k^i \tag{4.34}$$

Depending on the temporal context used to estimate the weights, we can have either an *on-line* (different weights for each class at the output of a classifier and weights change with time, $w_{k,t}^i$) or an *off-line* (different weights for each class at the output of a classifier, w_k^i) implementation. In the present study, the on-line estimate was done on a per-utterance basis (a set of weights was estimated for each utterance) and the off-line estimate was performed on all the utterances in the test database (a single set of weights was estimated for all the test utterances in the database).

The dilemma of EM is evident in ML weighting. While a smaller temporal context can capture the finer dynamics of the streams, estimates of the weights in the absence of large amounts of data are not reliable in this case. On the other hand, while a larger temporal context (having more data) gives better weight estimates, it loses the finer dynamics of the streams.

4.3.3 Results and Discussion

In this setup, the outputs of the MLPs trained on the 7 PLP feature streams were used for the combination (FCMS). In Fig. 4.5, the average likelihood per utterance is shown for the first 9 iterations (starting with uniform weights in the 0th iteration). As expected, the likelihood of the test data increases from one iteration to another. The values are high because instead of working with likelihoods, we worked with scaled likelihoods obtained by dividing the posterior estimates of the phoneme classes at the output of the MLPs by their prior probabilities.

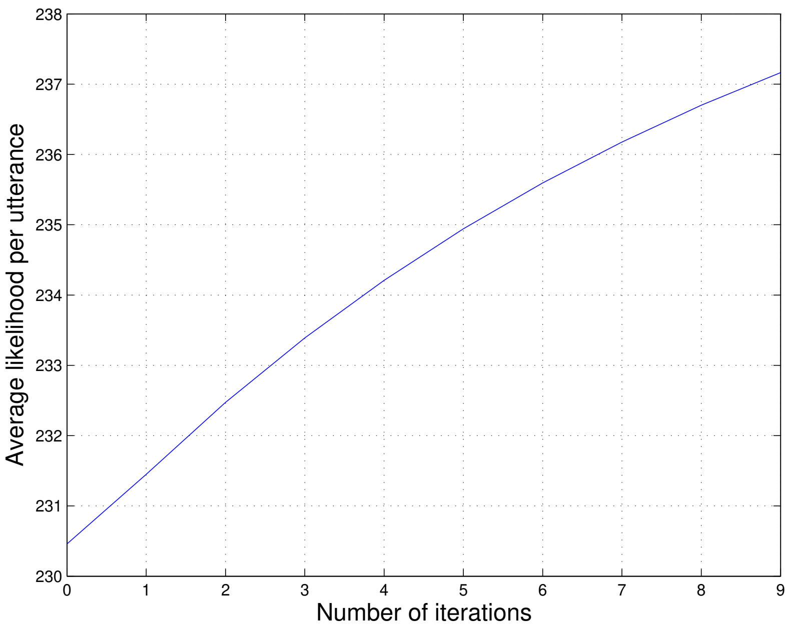


Figure 4.5. *ML Weighting (Batch Mode): Evolution of average likelihood per utterance from one iteration to another.*

In Fig. 4.6, we show the evolution of the weights from one iteration to another for the batch-mode (off-line) method of weight estimation by ML. The outputs of the MLPs trained on 7 PLP feature streams were combined and each MLP classifier had 27 phoneme classes. We started with equal weights for each class in each classifier (similar results were obtained for initialization with

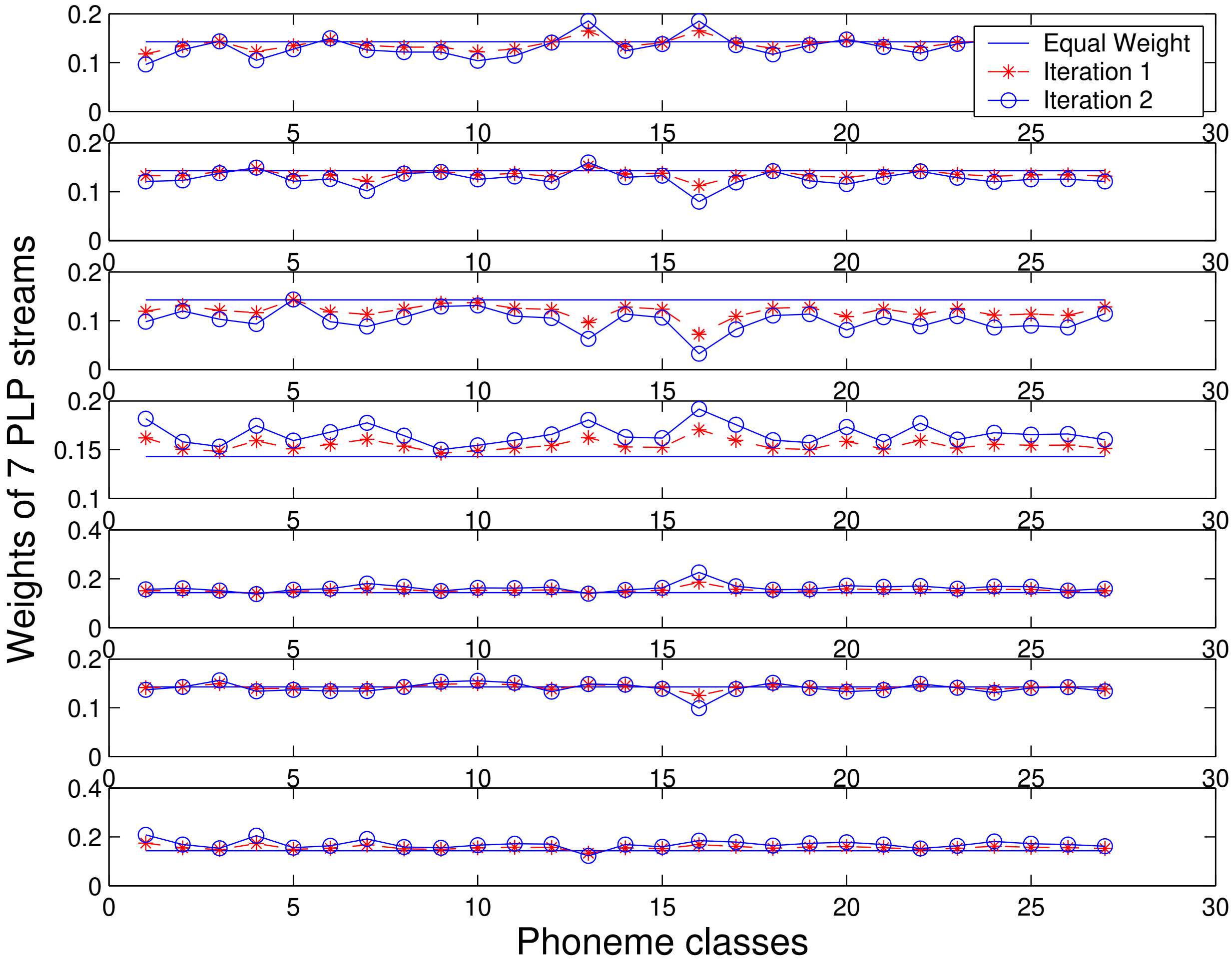


Figure 4.6. *ML Weighting (Batch Mode): Evolution of weights for first three iterations.*

inverse entropy weights) such that $\sum_{i=1}^{I} w_{i,k} = 1$ (the sum of the weights for a particular class over all the classifiers is 1), where $k = 1, \cdots, K$, and K the number of phoneme classes at the output of the MLP classifiers. The weight evolution is shown for the first two iterations. We observe that individual classifiers give different importance to various phoneme classes. This is reasonable that the classifiers used in the combination have unequal importance for different phonemes.

The WER for the on-line and off-line ML weighting are shown in Table 4.3. The performance of the ML weighting is inferior compared to the baseline system and does not change from one iteration to another. An analysis of the results shows that the *off-line ML* weighting is slightly better than the *on-line ML* weighting.

Though we could combine the streams (outputs of different classifiers) in such a manner that likelihood increased from one iteration to another, we could not achieve a better discrimination

WERs for ML Weighting								
	On-line ML weighting				Off-line ML weighting			
Iterations	**Clean**	**SNR12**	**SNR6**	**SNR0**	**Clean**	**SNR12**	**SNR6**	**SNR0**
0	11.7	18.4	30.1	50.8	11.7	18.4	30.1	50.8
1	12.1	19.2	30.8	51.2	11.6	18.4	30.3	51.5
2	12.3	19.5	31.2	51.4	11.5	18.4	30.6	51.7
3	12.4	19.7	31.6	51.5	11.5	18.4	30.4	51.7
4	12.4	19.7	31.3	51.4	11.5	18.4	30.5	51.7
5	12.4	19.8	31.2	51.4	11.5	18.4	30.7	51.7

Table 4.3. *WERs in % for the 7 possible PLP feature streams combined by ML weighting. The noise conditions are simulated by adding factory noise from the Noisex92 database at different SNRs to the utterances of the Numbers95 database. Iteration 0 corresponds to equal weighting.*

between the classes. The following is the possible explanation for these results:

1. When we increase the likelihood, in the absence of a target, we do not necessarily achieve a better discrimination. In EM training for HMM, we have explicit targets (either sentences, words or phonemes) for which we increase the likelihood. Explicit targets improve the discrimination between the classes[9].

2. The test data is same as the adaptation data. In approaches like MLLR, models are adapted on development data (either the transcription for the development data exists or is obtained first) and then the performance is evaluated on test data. In our case, we estimated the weights from the test data itself.

Another drawback of the ML weighting is its high computational cost. The iterative nature of the EM procedure for estimating weights requires time (computational cost) which is an order of magnitude more than the time taken by techniques like MP and inverse entropy weighting.

4.4 Summary

In this chapter, we studied three weighting techniques to combine streams in an FCMS setup. The two computationally simple techniques, namely MP and inverse entropy weightings, gave good improvements in performance. In contrast, computationally expensive ML weighting performed below expectation.

The conclusions of this study on weighting techniques are:

[9] In a strict sense, it is generative training. While training the models of one class, negative examples from other classes are not used.

1. Late integration (combination at posterior-level) performed better than early integration (combination at feature-level).

2. ML weighting gave a lower performance when the outputs of the MLPs trained on the 7 PLP feature streams were combined in the FCMS setup. MP weighting gave 4.7% relative average WER improvement on the same task and inverse entropy weighting yielded a relative average WER improvement of 6.6%.

3. The two methods of combination, namely the sum and product rules, yielded similar improvement in performance, and the product rule performed slightly better than the sum rule in almost all the cases.

4. The variation of inverse entropy weighting, where at each time frame the average entropy obtained from different classifiers was used as a dynamic threshold (IEWAT), gave the best performance.

Even though we obtained an improved performance by MP and inverse entropy weighting techniques, the improvement in performance was relatively low. The possible reason for this can be attributed to the fact that the individual feature streams used to train the classifiers were not carrying enough complementary information[10]. This highlights the issue that besides weighting techniques, the feature streams also need careful consideration.

In the next chapter, we propose and investigate new features in an FCMS setup which may carry complementary information to PLP features. The IEWAT method is used for combining the streams as it gave the best performance in the studies reported in the present chapter. The issue of complementarity of the feature streams is further discussed in Chapter 6, where we observe the oracle performance and illustrate its relationship with inverse entropy weighting.

[10]Refer to Section 5.2.5 for a comparison where the outputs of classifiers trained on a different set of 7 feature streams are combined by inverse entropy weighting method.

Chapter 5

Features in Multi-stream ASR

In Section 3.4, for posterior-level combination, we discussed several methods of generating multi-stream systems. It was mentioned that using different feature representations to train separate classifiers and combining the outputs of the classifiers (streams) is one type of multi-stream framework.

Multi-stream combination using several feature representations can improve the performance only if the different feature streams carry complementary information. Also, the more complementary information the feature streams carry, the better the improvement in the performance can be expected by combining the outputs of classifiers trained on such feature streams.

In the present chapter, we use PLP derived cepstral coefficients $(c, \Delta c, \Delta\Delta c)$ as the baseline features and explore additional feature streams which, when combined with PLP features, might improve the performance of an ASR system. Following is the organization of rest of the chapter: in Section 5.1, fundamental frequency ($F0$) feature is introduced as an additional feature in the multi-stream setup and motivation and the experimental results for the same are presented. Spectral entropy features are proposed in Section 5.2, and we discuss the results when the spectral entropy features are used along with PLP features. We conclude the chapter with a short summary.

5.1 Fundamental Frequency Feature

PLP features capture the characteristics of vocal tract (system information) and may not carry the characteristics of excitation signal (source information). However, source features might also carry important information which is complementary to the PLP features. Fundamental frequency, the periodicity of the speech signal, is a source feature and carries several important information. Fundamental frequency relates closely to the pitch frequency, a perceptual phenomenon, and is often used as a measure for pitch frequency. In this thesis, we have used the term pitch frequency for fundamental frequency.

The following are some of the important information present in the pitch frequency feature:

1. Voicing/unvoicing: Pitch is perceived for those speech signals which have periodicity in them. Periodicity in the signal exists because of periodicity of the excitation signal, which in turn occurs due to opening and closing of the vocal-folds while producing voiced sounds. Therefore, the presence or absence of pitch frequency can be a good indicator of whether a particular sound is voiced or unvoiced.

2. Gender: In males, typically the time difference between two consecutive excitations is larger compared to that for females. Therefore, the pitch frequency of males is typically lower than that of females.

3. Emotional state of the speaker: In case of emotional stress, generally the rate of opening and closing of the vocal-folds gets affected. In turn, this changes the pitch frequency of the speaker.

Including pitch frequency as a feature in ASR might be helpful in distinguishing between voiced and unvoiced sounds as well as gender of the speakers. However, it has been observed earlier that pitch frequency feature, when appended to the cepstral features, does not give an improvement in the ASR performance (Fujinaga et al., 2001). The possible reasons for this could be:

1. It is difficult to estimate the pitch frequency reliably

2. The information that the pitch frequency carries is already present in some parts of the cepstral features

3. The pitch frequency feature of dimension 1 gets submerged in the cepstral features of higher dimension, unable to show its usefulness.

4. The information present in pitch frequency cannot be exploited by frame-based Markov model, like the HMM, because the "units of information" of pitch frequency are suprasegmental.

The possibility that pitch frequency estimates are error prone is true (Bagshaw et al., 1993), especially in noisy environments. Moreover, cepstral features do carry voicing information to a certain extent in the form of energy of the spectral envelope, but gender information is not present in them (cepstral features are obtained from a smoothed spectral envelope, and the pitch frequency information present in the original spectrum is lost in the smooth spectrum). Finally, it is possible that pitch frequency is not a very useful feature for ASR and is unable to contribute significantly when appended to the cepstral features (pitch frequency is a single dimensional feature vector while the dimension of cepstral features is typically much higher[1]).

Recently, it has been shown that proper integration of a pitch frequency feature (either by marginalization or by appending) (Doss et al., 2003; Doss, 2005) can yield an improvement in the performance of ASR systems. In marginalization of pitch feature, separate models were trained for males and females. During testing, the outputs of the models were merged to get the combined output (Doss et al., 2003). Gender-dependent modelling (Konig et al., 1991), where separate models are created for male and female speakers, has also helped in improving the performance of ASR systems. In (Lei et al., 2005), the authors showed that appending the pitch frequency feature and its derivative can improve ASR performance for the Chinese language. Motivated by these findings, we studied the pitch frequency feature in the framework of multi-stream combination.

In the work reported in the following section, we analyze the pitch frequency feature and investigate whether and how the pitch frequency feature can be incorporated in an ASR system so as to get an improvement in the performance. In turn, we also investigate the validity of the four reasons listed above as to why the appending of pitch frequency feature to the cepstral features usually does not improve the ASR performance.

5.1.1 Implementation

We extracted the pitch frequency feature from the speech signal by using a simplified inverse filter transform (SIFT) method (Markel, 1972). The steps of the SIFT algorithm are shown in Fig. 5.1. The speech signal is passed through a low-pass finite impulse response (FIR) filter with cut-off frequency of 1 KHz and then it is down-sampled. The filtered signal undergoes low order LP analysis (4 to 6) and the residual signal is obtained. In auto-correlation of the residual signal, the position of the first peak (in number of samples) is identified as the pitch period. In this study, the search for the pitch period is restricted between sample numbers $20(= 8000/400)$ and $115(\approx 8000/70)$, which correspond with the pitch frequencies between 400 and 70 Hz respectively. The value 8000 men-

[1]PLP feature vector dimension is 39 in the present studies.

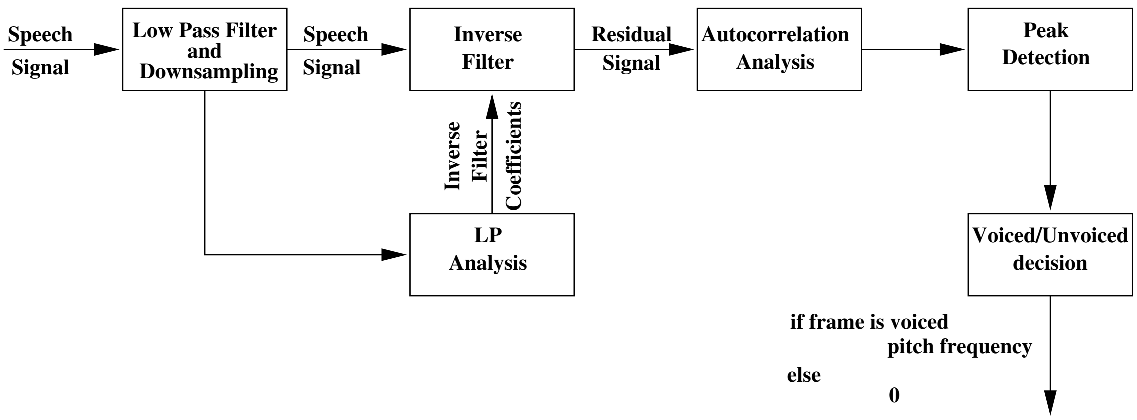


Figure 5.1. *SIFT algorithm for extracting pitch frequency.*

tioned above corresponds to the sampling frequency of the speech signal (in Hz). Pitch frequency is obtained by dividing the sampling frequency of the signal by the pitch period (in samples). Only the frames which had high residual energy were considered as voiced frames and pitch frequency was extracted for them. The pitch frequency contour obtained for the speech utterance was smoothened by median filtering to remove isolated jumps from voiced to unvoiced and vice-versa.

The pitch frequency feature being one dimensional may not be useful as an individual feature stream in multi-stream framework. In fact, one dimensional pitch frequency feature yielded a poor performance (a WER of 93.7%) when used in the hybrid system. Therefore, we used the pitch frequency in the FCMS framework along with PLP features in the following way: as in the previous chapter, there were 7 PLP feature streams (c, Δc, $\Delta\Delta c$ and their all possible combinations). Pitch frequency was appended to each of the 7 feature streams and a separate MLP was trained for each feature stream. The problem of the pitch frequency feature being submerged in the PLP features of higher dimension is reduced by appending the pitch frequency to the feature streams of lower dimension. The pitch frequency feature is used 7 times, but with lower dimensional feature vectors (the dimensionality of the 7 streams are 13, 13, 13, 26, 26, 26 and 39). It is different from appending the pitch frequency to a single feature stream 7 times to have enhanced contribution of pitch frequency[2]. Still, the problem of unreliable pitch frequency feature estimation cannot be overcome by this method. By appending the pitch frequency feature to all the 7 PLP feature streams in FCMS framework, we may also find out if the information carried by the pitch frequency feature is confined to some specific part of the baseline PLP feature vector.

[2] In such a case, the replicas of the feature do not carry ant additional information and hence may not improve the performance.

5.1.2 Results

Feature	**Clean**	**SNR12**	**SNR6**	**SNR0**
c, Pitch	12.5 (0.0)	21.3 (-0.3)	35.3 (-0.4)	57.8 (-0.3)
Δc, Pitch	15.0 (1.7)	23.7 (1.7)	34.9 (0.6)	56.1 (1.0)
$\Delta\Delta c$, Pitch	15.8 (1.2)	22.4 (-0.3)	36.3 (0.5)	60.0 (0.8)
c, Δc, Pitch	11.0 (-0.2)	19.0 (-0.4)	31.1 (-0.8)	52.7 (-0.1)
c, $\Delta\Delta c$, Pitch	10.7 (0.0)	18.0 (-0.4)	30.0 (-0.8)	52.3 (0.4)
Δc, $\Delta\Delta c$, Pitch	12.6 (1.1)	19.3 (0.5)	30.7 (0.6)	51.7 (-0.7)
c, Δc, $\Delta\Delta c$, Pitch	10.0 (-0.4)	17.7 (0.1)	29.6 (-0.2)	51.0 (-0.1)
IEWAT (Sum Rule)	10.0 (0.3)	16.2 (-0.1)	27.4 (-0.7)	48.3 (-1.0)
IEWAT (Product Rule)	9.7 (0.0)	15.7 (-0.6)	27.3 (-0.4)	48.9 (-0.1)

Table 5.1. *The baseline results and in brackets (absolute change in % WERs) for the 7 possible PLP streams appended with the pitch frequency feature and their combination by inverse entropy weighting in an FCMS framework. Results of Table 4.1 (Page 47) are used as a baseline. A + change indicates improved performance and a - change shows degraded performance. c, Δc and $\Delta\Delta c$ represent static, delta and delta-delta features respectively. The Numbers95 database corrupted by factory noise from the Noisex92 database at different SNRs is used for running the experiments.*

In this section, we present the results of the pitch frequency feature in an FCMS setup as described in the previous section. Pitch frequency is normalized by dividing it by 400 Hz, the highest value of the pitch frequency assumed in the SIFT algorithm while searching for the peak in the residual of the autocorrelation. We observe that pitch frequency feature being appended to the standard PLP features does not improve the system's performance (row 7: c, Δc, $\Delta\Delta c$, Pitch). In fact, either there is no improvement or a drop in absolute performance. Another interesting observation is that appending the pitch feature to any feature stream where static features are present (rows 1, 4, 5 and 7) generally hurts the performance. When we compare this with the situation when pitch feature is appended to the feature streams having dynamic PLP features only (rows 2, 3 and 6), there is generally a slight improvement in the performance. Though these degradations or improvements are not significant, they still present a trend. A simple conclusion could be that in the presence of static features, we do not observe an improvement in the performance. The reason for this behavior could be: the information that the pitch feature carries is present in some form in the static features. In general, the pitch frequency is estimated for the frames having high energy (voiced frame) and the information about the energy of the frame is present in the cepstral-coefficient of order 0. So it is possible that the information present in the pitch frequency feature is already represented by the 0^{th} order cepstral-coefficient in the static features (c).

In the framework of FCMS, we get a slight improvement in the performance of the system for clean speech. There is no improvement in the performance for noisy speech and the reason can be attributed to a) pitch frequency estimate is not robust for noisy speech, and/or b) pitch feature is not useful for ASR even in the FCMS framework.

To understand this aspect, we did an experiment where we appended the pitch frequency feature obtained from clean speech to the PLP features obtained from noisy speech to run the experiments on noisy test conditions (Table 5.2).

Feature	**Clean**	**SNR12**	**SNR6**	**SNR0**
c, Pitch	0.0	0.1	0.4	1.0
Δc, Pitch	1.7	3.2	4.0	3.7
$\Delta\Delta c$, Pitch	1.2	1.4	4.4	6.4
c, Δc, Pitch	-0.2	0.1	0.1	0.6
c, $\Delta\Delta c$, Pitch	0.0	0.0	0.3	1.1
Δc, $\Delta\Delta c$, Pitch	1.1	1.7	3.4	4.2
c, Δc, $\Delta\Delta c$, Pitch	-0.4	0.6	0.6	0.8
IEWAT (Sum Rule)	0.3	1.0	1.3	1.9
IEWAT (Product Rule)	0.0	0.6	1.4	2.7

Table 5.2. *Absolute change in % WERs for the 7 possible PLP streams appended with the pitch frequency feature from clean speech. Also their combination by the inverse entropy weighting in the FCMS framework. Results of Table 4.1 (Page 47) are used for computing the difference. A + change indicates improved performance and a - change shows degraded performance. c, Δc and $\Delta\Delta c$ represent static, delta and delta-delta features respectively. The Numbers95 database corrupted by factory noise from the Noisex92 database at different SNRs is used for running the experiments.*

We observe some interesting trends which need further discussion.

1. Appending the clean pitch frequency feature to the static features or to a feature stream having static features (rows 1, 4, 5 and 7) improves the performance slightly. In contrast, the feature streams without static features (rows 2, 3 and 6) show a better improvement when the pitch frequency feature is appended to them. The trend is consistent and easily noticeable. It strengthens our previous conjecture that static features already carry the pitch frequency information in some form. It also confirms that appending the pitch frequency feature to the standard PLP features (row 7: c, Δc, $\Delta\Delta c$, Pitch), even if the pitch frequency estimate is reliable, does not improve the ASR performance.

2. The improvement in the results by appending the pitch frequency feature (compare Tables 5.1 and 5.2) obtained from clean speech confirms that the pitch frequency estimate is not robust for noisy speech. Improvement in performance for feature streams having only dynamic features (rows 3, 4, and 6) is consistent and significant.

3. If the pitch frequency estimate is robust (Table 5.2), we notice an improvement in performance

of the FCMS ASR system. Considering the fact that the pitch frequency feature is a single dimensional feature, improvement in performance is worth reporting. Also, the potential of the pitch frequency feature could be realized in the proposed FCMS setup.

In this study, we observed that appending the pitch frequency feature to PLP cepstral features in ASR may not be a correct approach as the static PLP features carry similar information. In the proposed FCMS setup, though the pitch frequency feature helped in clean speech, in the absence of a robust pitch estimation method, it could not yield a better performance for noisy test conditions. Robustness of the pitch frequency feature is an issue and we need better methods to estimate the pitch frequency feature reliably.

We observed that if the pitch feature can be estimated reliably, the proposed FCMS method using the pitch frequency feature can improve the ASR performance. In this method, the feature streams with dynamic cepstral features improve their individual performance when pitch feature is appended to them. Use of better individual feature streams led to a better performance when the outputs of the classifiers trained on them were combined. Also, the improvement by including the pitch frequency feature of dimension one is considerable and it could be worthwhile to investigate the feature further.

5.2 Multi-band Spectral Entropy Features

5.2.1 Motivation

In STFT spectra of speech, we observe distinct peaks and the position of these peaks (formants) in the spectra are dependent on the phoneme under consideration. Generally, the formants are considered important for robust speech recognition (McCandless, 1974; Moore, 1997; Welling and Ney, 1998; Strope and Alwan, 1998). In additive noise conditions, typically the formants of the spectrum are less affected than other parts of the spectrum. In (Padmanabhan, 2000) the author tried to use the location of spectral peaks as an additional feature in ASR. In the framework of HMM2 (Weber et al., 2003), the authors extracted robust features from the spectrum and showed that they closely follow the formants in the spectrogram. Similarly, in (Ikbal et al., 2004a), the authors suggested spectro-temporal activity pattern (STAP) features centered around the spectral peaks for robust ASR. Instead of picking the formants or their position, in (Ikbal et al., 2003a, 2004b) the authors used a non-linear transformation to enhance the peaks of the spectrum. In all these approaches, the goal was to have a robust feature extraction method having information about the formants of the spectrum, as formants are considered to be less affected by noise.

Entropy can be used to capture the "peakiness" of a probability mass function (PMF). A PMF with sharp peaks will have low entropy while a PMF with flat distribution will have high entropy. In their work, Niyogi and Sondhi (2002) investigated the detection of stop-consonants for event-based ASR (different features for different phoneme classes instead of the usual approach of the same features for all phoneme classes). In the study, the spectral entropy rate was used to measure the spectral flatness and explored as one of the features for detecting stop-consonants in continuous speech. In their study, the spectral entropy rate was computed from STFT of the speech signal, assuming the signal to be normally distributed in the time domain (Papoulis, 1991, Page 568).

In a similar endeavour, sub-band spectral entropy was used for voice activity detection across bands (McClellan and Gibson, 1997). The authors pointed out that spectral entropy can measure the flatness of sub-band spectrum. In (McClellan and Gibson, 1997), the authors used spectral entropy for deriving mode and rate allocation cues for a variable-rate code-excited linear prediction (CELP) coder.

On similar lines, the central idea in (Misra et al., 2004, 2005a; Misra and Bourlard, 2005) while using multi-resolution spectral entropy as a feature was to capture the peaks of the spectrum and their location. To compute entropy of a spectrum, we converted the spectrum into a PMF-like function by normalizing it.

$$x_i = \frac{X_i}{\sum_{j=1}^{N} X_j} \qquad \text{for } i = 1 \text{ to } N \tag{5.1}$$

where X_i is the energy of i^{th} frequency component of the spectrum, $\mathbf{x} = (x_1, \cdots, x_N)$ is the PMF of the spectrum and N is the number of points in the spectrum (the order of the STFT). Spectral entropy for each frame is computed from $\mathbf{x}$ as:

$$H = -\sum_{i=1}^{N} x_i \log_2 x_i \tag{5.2}$$

Fig. 5.2(b) shows the spectral entropy contour computed on a full-band spectrum for clean speech. We observe that speech sound segments, usually characterized by distinct spectral peaks, have lower spectral entropy compared to silence segments. Therefore spectral entropy computed on the full-band can be used as an estimate for speech/silence detection. In presence of noise, the formants

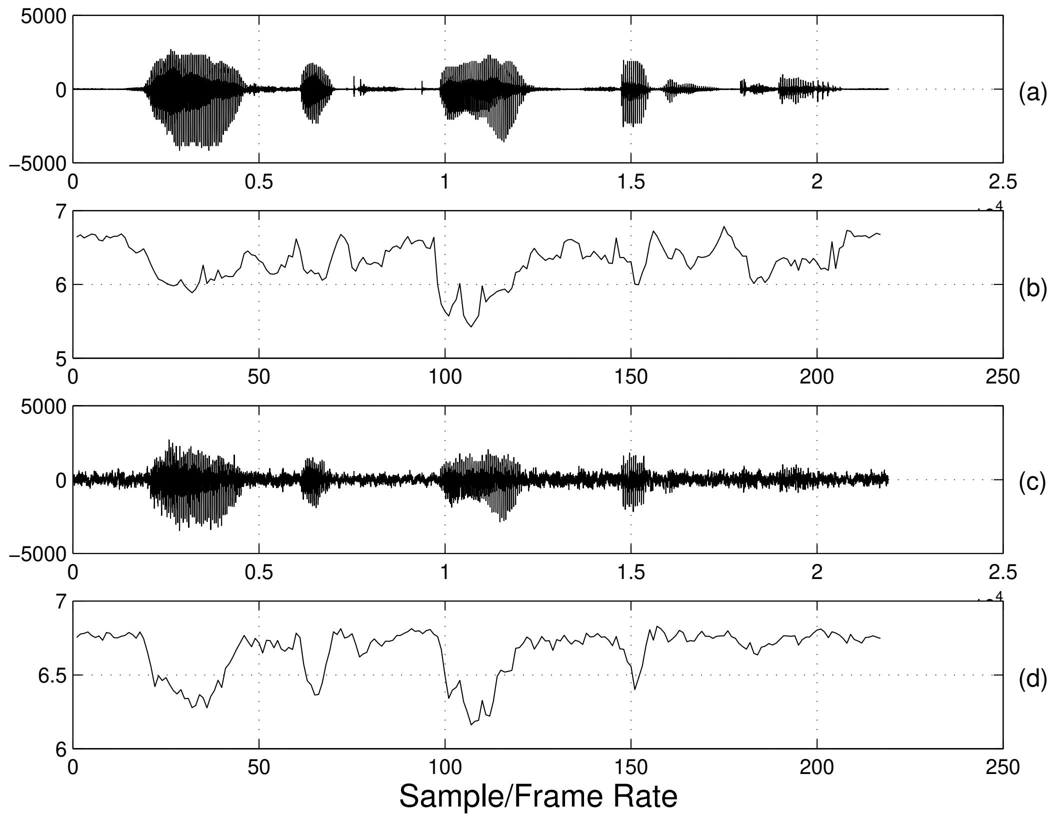


Figure 5.2. *Entropy computed from the full-band spectrum. (a) A clean speech waveform from the Numbers95 database, (b) Spectral entropy contour for the clean speech waveform, (c) Speech waveform corrupted by factory noise at 6 dB SNR, and (d) Spectral entropy contour for the speech waveform corrupted by factory noise at 6 dB SNR. Factory noise is taken from the Noisex92 database and added to the clean speech waveform.*

are less affected compared to the other parts of the spectrum. So we can assume that entropy of the spectrum, if used for speech/silence detection, will be robust to noise, and indeed it appears to be true as observed in Fig. 5.2(d). Though the dynamic range of the spectral entropy contour is reduced in the presence of noise, it retains its discriminatory property. In (Shen et al., 1998; Huang and Yang, 2000; Subramanya et al., 2005), the authors successfully used spectral entropy for end point detection of speech in noisy environmental conditions.

5.2.2 Multi-band/Multi-resolution Spectral Entropy

The full-band spectral entropy feature can capture only the gross peakiness of the spectrum but not the position of the formants. In (Misra et al., 2004, 2005a), we suggested multi-resolution/multi-band spectral entropy features. The main motivation behind this was to identify the presence or absence of spectral peaks in each sub-band and use that as a pattern. To estimate multi-band spectral entropy features, we divided the spectrum into sub-bands and computed the entropy of each sub-band.

Non-overlapping sub-bands

In the beginning, we used non-overlapping sub-bands to extract multi-band spectral entropy features. The normalized STFT spectrum was divided into $\mathbf{J}$ non-overlapping sub-bands of equal size. Entropy was estimated for each sub-band and we obtained one spectral entropy value for each sub-band. These sub-band spectral entropy estimates indicate the presence or absence of spectral peaks in that sub-band. When $\mathbf{J} = \mathbf{1}$, we have the full-band spectrum and get one spectral entropy estimate per frame. When there are two sub-bands ($\mathbf{J} = \mathbf{2}$), we obtain two spectral entropy estimates, one for each sub-band and so on. In our experiments, we changed the parameter $\mathbf{J}$ from $\mathbf{1}$ to $\mathbf{32}$. This process of estimating spectral entropy from each sub-band is equivalent to *estimating the spectral entropy contribution of each sub-band to the full-band spectral entropy*.

Overlapping sub-bands

We also made an attempt to study the performance for overlapping sub-bands. When the sub-bands are overlapping, a very high number of alternatives need to be explored to account for all possible sub-band sizes and their positions. We restricted ourselves to overlapping sub-bands defined on the well known Mel scale (Davis and Mermelstein, 1980). In (Misra et al., 2005a), we obtained the best results by dividing the normalized full-band spectrum into 24 overlapping sub-bands defined on a Mel scale and estimated spectral entropy for each sub-band. In the overlapping case, the sub-bands' spectral entropy estimates do not sum to the spectral entropy of the full-band.

Temporal information

In standard ASR features, temporal information is introduced by appending first and second order time derivatives (dynamic features) to the static features. Similarly, we appended the first and second order time derivatives of the spectral entropy features to incorporate temporal information in the present setup.

5.2.3 Results

The results obtained by static multi-band spectral entropy features for different numbers of sub-bands are shown in Table 5.3. The results are shown only for clean speech. An analysis of the results reveals that as the number of sub-bands increases, WER decreases. Moreover, the initial improvement with the increase in number of sub-bands is more and it reduces as we keep increasing the number of sub-bands. With an increase in number of sub-bands, the number of frequency components left in a sub-band decreases and the spectral entropy is estimated from these fewer components only.

WERs for spectral entropy features		
Feature	**Feature Dimension (J)**	**WER**
Full-band Entropy	1	91.6
2-bands Entropy	2	74.4
3-bands Entropy	3	59.5
4-bands Entropy	4	42.7
8-bands Entropy	8	24.3
16-bands Entropy	16	18.6
24-bands Entropy	24	16.2
32-bands Entropy	32	15.1
24 Mel-bands Entropy	24	15.7

Table 5.3. *WERs in % for clean speech for multi-band spectral entropy features in a hybrid system for different number of sub-bands. Only* **Mel-bands are overlapping***. Rest of the sub-bands are non-overlapping.The experimental results are obtained on the Numbers95 database.*

The effect of appending time derivatives to multi-band spectral entropy features is shown in Table 5.4. In this table, we present the results on noisy speech also (noise conditions were simulated

WERs: Spectral entropy and its time derivatives				
Feature	**clean**	**SNR12**	**SNR6**	**SNR0**
16-bands	15.5	22.0	31.9	53.2
24-bands	14.0	20.2	29.3	50.1
32-bands	14.0	20.4	28.8	47.1*
24 Mel-bands	12.8	18.3	**27.0***	**45.1***
PLP (Baseline)	**10.0**	**17.7**	29.6	51.0

Table 5.4. *WERs in % for entropy features with its* **first and second order time derivatives appended** *in hybrid system for clean and noisy test conditions. Only Mel-bands are overlapping. Factory noise from the Noisex92 database added to the utterances of the Numbers95 database at different SNRs. Performance of the PLP features is given for comparison. The numbers in* **bold** *show the best performance and * indicates that the improvement in performance compared to the baseline system is significant.*

by adding factory noise from the Noisex92 database at different SNRs). We show the results for 16 or more non-overlapping sub-bands and the overlapping sub-bands defined on a Mel scale as these gave the best results for clean speech in Table 5.3.

The results show that including the time derivatives of spectral entropy features help in improving the ASR performance. This observation holds good for clean as well as noisy test conditions and for all the sub-bands divisions considered in Table 5.4. Furthermore, the overlapping sub-bands defined on a Mel scale yielded the best performance.

On comparing spectral entropy features with PLP features, we realize that PLP features work better for low noise conditions while spectral entropy features work well for high noise conditions. This is an indication that the two features may carry complementary information and their combination might yield an improved ASR performance. To investigate this issue further, we performed multi-stream ASR studies where we combined the PLP features with multi-band spectral entropy features in the framework of FCMS. One MLP was trained for each feature stream and the IEWAT method discussed in Section 4.2.3, which gave the best performance, was chosen to combine the output posterior estimates of the different MLP classifiers.

5.2.4 Spectral Entropy Features in Multi-stream

We studied the combination of spectral entropy features with PLP features at the following two levels:

1. Combination at the feature-level

2. Combination at the posterior-level

As discussed previously, in feature-level combination, the features are appended and modelled jointly, while in posterior-level combination, features are modelled separately and then the outputs of the classifiers are combined (Section 3.2.1).

PLP features constituting the baseline were used along with spectral entropy features in the multi-stream combination experiments presented in this section. Unlike the pitch feature, which was single dimensional, the spectral entropy features have higher dimensionality and can be utilized as a separate feature stream.

The performance of two multi-stream methods, namely feature and posterior combination, are presented in Table 5.5. The results show that appending the features (Table 5.5: PLP,24-Mel) helps in low noise conditions and the advantage of such combination is lost in high noise cases. The reason for this could be that the features affected by noise influence the overall performance when features are appended, and the features less affected by noise lose their advantage. In contrast, when

Feature	Clean	SNR12	SNR6	SNR0
PLP (Baseline)	10.0	17.7	29.6	51.0
24-Mel	12.8	18.3	27.0*	45.1*
PLP, 24-Mel	9.6	15.8*	28.1	51.7
Sum Rule				
FCMS: PLP,24-Mel	**9.2**	**15.0***	**24.5***	**44.5***
Product Rule				
FCMS: PLP,24-Mel	**9.3**	**15.1***	**24.3***	**44.6***

Table 5.5. *WERs in % for PLP features, 24 Mel-band spectral entropy feature and its time derivatives (24-Mel), the two features appended (PLP, 24-Mel), and the two features in full-combination multi-stream (FCMS: PLP,24-Mel) in a hybrid system. The noise conditions are simulated by adding factory noise from the Noisex92 database at different SNRs to the utterances of the Numbers95 database. The numbers in* **bold** *show the best performance and * indicates that the improvement in performance as compared to the baseline system is significant.*

features are modelled separately and then combined (as in FCMS), the individual feature streams retain their properties. At the time of combination, we can give more weight to the stream (outputs of the classifier) which was less affected by noise. Still, there exists the issue of identifying the streams according to their reliability and we discussed some of the methods for weighting different streams in Chapter 4. A significant relative average WER improvement of 14.5% is achieved when we use spectral entropy features along with PLP features in FCMS with inverse entropy weighting.

The performance of the spectral entropy features is affected when the noise contains sinusoidal components with high energy. High amplitude at those frequencies produces a spectral peak in the corresponding sub-band and the spectral entropy has a low value for that sub-band. However, the sub-band methods mitigates the effect of such noises on the overall performance. The results obtained on Numbers95 database corrupted by additive lynx noise, which contains sinusoidal components with high energy, support this reasoning.

In Figs. 5.3 and 5.4, we see the performance of PLP features, spectral entropy features, spectral entropy features appended to PLP features and spectral entropy features along with PLP features in FCMS for lynx and car noises respectively.

The performance of PLP features, which involves smoothing, is good for lynx noise. In comparison, spectral entropy features perform poorly for lynx noise in all the noise levels studied. Still, it is noticeable that the performance gap between PLP features and spectral entropy features narrow downs as the noise level increases. Moreover, the two features in FCMS framework perform better than the PLP baseline features alone, indicating the complementarity of the two streams. Once again, feature-level combination yields less improvement in performance compared to posterior-level combination, specially in high noise conditions.

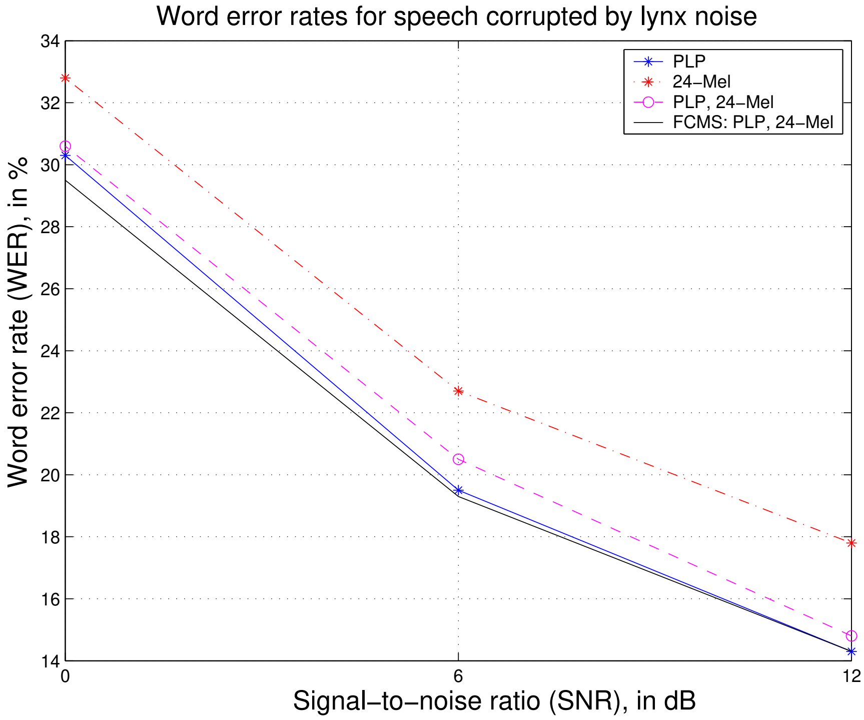


Figure 5.3. *Performance in % WER of different feature streams for a hybrid system for* **lynx noise** *at different SNRs. The plot shows baseline PLP (-*-), spectral entropy features derived from overlapping sub-bands defined on a Mel scale (-*.), spectral entropy features appended to the PLP features (-o-) and the two features in FCMS (—). Lynx noise is taken from the Noisex92 database and added to the utterances of the Numbers95 database at different SNRs.*

The trends observed for lynx noise are observed on car noise also. However, the improvement in performance is more for car noise when PLP and spectral-entropy features are used in the FCMS framework compared to the two features being appended.

5.2.5 Combination of PLP, RASTA-PLP and Spectral Entropy Features

RelAtive SpecTrAl (RASTA)-PLP (Hermansky and Morgan, 1994) is a well-known feature representation for robust ASR. In this section, we compare the performance when CJRASTA-PLP features (a variation of RASTA-PLP features) and spectral entropy features are used individually with PLP features. The FCMS setup was used for carrying out the experiments and inverse entropy weighting was used for combining the outputs of the classifiers. In Table 5.6, the results are presented for the above mentioned setup. We observe that most often combining spectral entropy features with PLP features gives a better improvement than combining CJRASTA-PLP features with PLP features.

In the next experiment, we combine all the three features in an FCMS setup, giving rise to 7

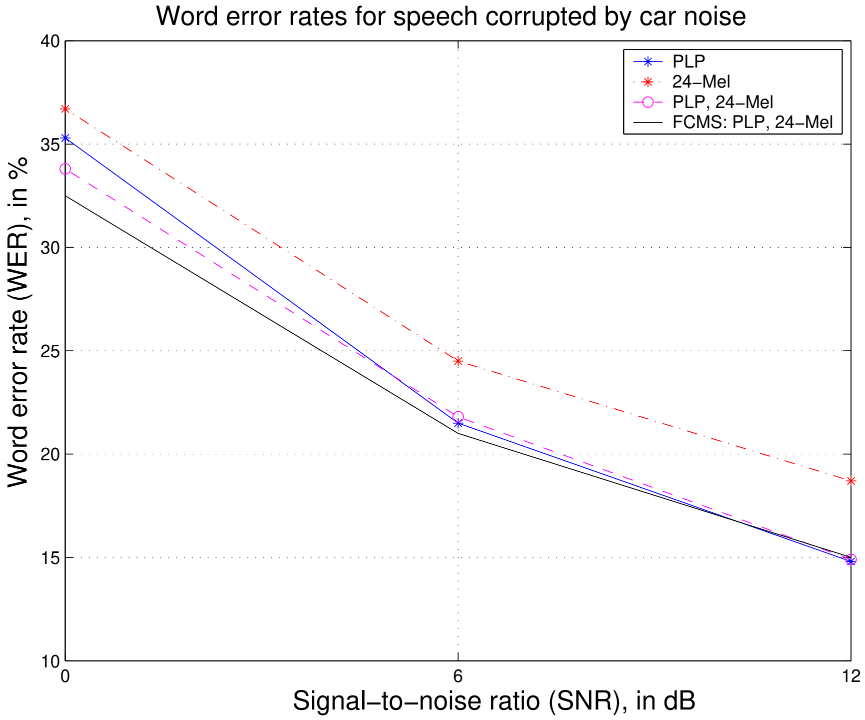


Figure 5.4. *Performance in % WER of different feature streams for a hybrid system for* **car noise** *at different SNRs. The plot shows baseline PLP (-*-), spectral entropy features derived from overlapping sub-bands defined on a Mel scale (-*.), spectral entropy features appended to the PLP features (-o-) and the two features in FCMS (—). Car noise is taken from the Noisex92 database and added to the utterances of the Numbers95 database at different SNRs.*

Features	Clean	Factory Noise			Car Noise		
		SNR12	SNR6	SNR0	SNR12	SNR6	SNR0
FCMS (PLP, CJRASTA)	9.4	15.3*	26.4*	46.8*	**13.7***	**20.0***	33.3*
FCMS (PLP, 24-Mel)	**9.2**	**15.0***	**24.5***	**45.5***	15.0	21.0	**32.5***

Table 5.6. *WERs in % for CJRASTA-PLP features and 24 Mel-band spectral entropy feature with its time derivatives (24-Mel) along with PLP features in FCMS with inverse entropy weighting. Results are for a hybrid system on the Numbers95 database. The noise conditions are simulated by adding the factory and car noises from the Noisex92 database at various SNRs. The numbers in* **bold** *show the best performance and * indicates that the improvement in performance as compared to the baseline system is significant.*

streams. Table 5.7 shows that the combination of the three streams gives a better improvement than the two stream combination results presented in Table 5.6 where CJRASTA-PLP and spectral entropy features were individually combined with PLP features. This indicates that CJRASTA-PLP features carry some information that is complementary to the information carried by the combination of PLP and spectral entropy features, and inclusion of CJRASTA-PLP as one of the feature streams is indeed beneficial in improving the overall performance. Moreover, this result emphasizes that better gains can be achieved if we consider a large number of feature representations having complementary information and combine them in an FCMS setup. It is also noted that

Features	Clean	Factory Noise			Car Noise		
		SNR12	SNR6	SNR0	SNR12	SNR6	SNR0
PLP	10.0	17.7	29.6	51.0	14.8	21.5	35.3
CJRASTA	10.6	17.1	27.9*	48.6*	15.6	23.4	38.5
24-Mel	12.8	18.3	27.0*	45.1*	18.7	24.5	36.7
PLP, CJRASTA	9.8	16.9	27.8*	49.0*	14.4	21.3	34.3
PLP, 24-Mel	9.6	15.8*	28.1	51.7	14.9	21.8	33.8*
CJRASTA, 24-Mel	9.6	15.3*	25.4*	46.7*	14.4	19.8*	31.9*
PLP, CJRASTA, 24-Mel	9.5	15.1*	26.4*	49.3	14.3	20.4	31.7*
FCMS (PLP, CJRASTA, 24-Mel)	**8.6***	**13.7***	**23.1***	**44.0***	**13.6***	**19.4***	**31.0***

Table 5.7. *WERs in % for PLP features, CJRASTA-PLP features, 24 Mel-band spectral entropy feature with its time derivatives (24-Mel), and the three features in full-combination multi-stream (FCMS) with inverse entropy weighting. Results are for a hybrid system on the Numbers95 database. The noise conditions are simulated by adding the factory and car noises from the Noisex92 database at various SNRs. The numbers in* **bold** *show the best performance and * indicates that the improvement in performance as compared to the baseline system is significant.*

concatenation of CJRASTA-PLP and spectral entropy features is most beneficial among all the two feature stream concatenations considered in Table 5.7.

In Appendix D, an oracle (described later in Chapter 6) is used to combine CJRASTA-PLP and spectral entropy features individually in a hybrid FCMS setup. The oracle analysis (to show the complementarity of streams, Section 6.2.2) indicates that spectral entropy features carry more complementary information than CJRASTA-PLP features, when each of the features are used with PLP features.

Comparing this result (Table 5.7) with the results presented in Table 4.2 (Page 53), where the outputs of the MLPs trained on 7 PLP feature streams were combined by inverse entropy weighting, the complementarity of the streams can also be noticed. In Chapter 4, it was suggested that the improvement observed by combining the 7 PLP streams in low because the streams were not carrying enough complementary information.

5.3 Summary

In this chapter, we analyzed two feature representations in the multi-stream framework. The first feature we studied is the well-known fundamental frequency (referred as pitch frequency in this study). Giving the reasons why pitch frequency could be of help in ASR, we did some analysis to find out why appending the pitch feature to the standard ASR features usually does not improve the performance. Then we proposed an FCMS setup where pitch frequency was used along with PLP features. We observed that if the pitch frequency estimates are robust to noise, the pitch feature in the proposed FCMS framework can yield a considerable improvement in performance over the baseline PLP based system. However, in the absence of a robust estimation method for pitch extraction, we get an improvement only for clean speech.

In the next part of the study, we presented multi-band spectral entropy features. We observed the robustness of the new features towards additive wide-band noises at low SNRs. We also found that the new features are complementary to PLP features and when both the features are used in an FCMS setup, we obtain an improvement in the performance. The improvement was observed for different kinds of noise conditions at various SNRs. We noticed that the improvement is larger in high noise conditions compared to improvement in low noise conditions.

The conclusions of the studies in this chapter are as follows:

1. Appending of the pitch frequency feature to the usual cepstral features did not improve the performance of the ASR system. This behaviour can be explained by the reasoning that pitch frequency information is present in the static cepstral features, therefore appending the pitch frequency feature does not contribute new information to improve ASR performance.

2. Utilizing the pitch features in an FCMS framework, as proposed in this work, can yield an improved performance. Considering that pitch frequency is a single dimensional feature, the improvement obtained by including the feature is good.

3. The SIFT method used for pitch frequency estimation is based on LP analysis. LP analysis is known to be prone to noise and as a result the pitch frequency estimates are not reliable. The ASR performance could be improved only for clean speech where pitch frequency estimates are more reliable. It is worth exploring some other methods for pitch extraction which are less affected by noise.

4. Multi-band spectral entropy features (appended with first and second order time derivatives) were observed to be robust to additive noise conditions at low SNRs. In contrast, PLP features performed well in low noise conditions.

5. Appending the two features (early integration) yielded an improved performance for low noise conditions. However, in high noise conditions, the appending did not give any improvement over the baseline. The reason for this could be that when the features are appended, if one feature is affected by noise, it affects the whole model and hence the performance.

6. The multi-band spectral entropy features when used with PLP features in an FCMS framework with inverse entropy weighting gave consistent and most often significant improvement in performance for all noise conditions and various noise levels studied in this chapter. Modelling the features separately, and then combining the posteriors at the output of the MLPs (late integration), is a better approach (less affected by noise) compared to early integration if a proper weighting technique is employed at the time of combination. Kirchhoff and Bilmes (2000) reported a similar observation in multi-stream ASR task.

7. Combination of PLP, RASTA-PLP and spectral entropy features in the FCMS setup yielded the best performance, indicating that the combination of more number of streams brings more complementary information into the ASR system. In this setup, once again, late integration was found to be better than the early integration.

In the next chapter, we do the analysis of an "oracle" in an attempt to investigate the issue of complementarity of feature streams in multi-stream ASR. We also explore how oracle selection correlates with inverse entropy weighting.

Chapter 6

Oracle Test and Embedded Training

In posterior-level combination in a multi-stream system, if at every time instant we can select the stream (outputs of the classifier trained on the feature stream) that is "best" among all the streams considered for combination, it will lead to the best performance that can be achieved by frame-level weighting techniques[1]. Such "oracle tests" have been used earlier to find out the oracle performance in pattern recognition tasks (Hermansky et al., 1996; Shire and Chen, 2000; Kuncheva, 2002).

In this chapter, we also propose a different interpretation of the oracle test to analyze the issue of complementarity of feature streams in a multi-stream system. It is assumed that the outputs of the classifiers (streams) are more complementary if the feature streams used to train them are more complementary, that is, outputs of a classifier are a representation of the feature stream used to train that classifier.

The aim of investigating the oracle test is three fold:

1. To find out what is the best performance that can be achieved by frame-level weighting while combining outputs of classifiers trained on a given set of feature streams. One separate classifier is trained for each feature stream.

2. To find out whether the streams considered for combination have complementary information.

[1] It might not yield the best word recognition rate because of weak correlation between frame and word accuracies in an ASR task. Also, we are maximizing the likelihood of the best path without taking into consideration the likelihood of the competing paths.

3. To find out how well the inverse entropy weighting method studied in this thesis corresponds with the oracle choice.

In Chapters 4 and 5, we trained the MLP classifiers using hand-labelled data. However, embedded training is known to improve the performance of ASR systems. In the second part of this chapter, we investigate the idea of embedded training for multi-stream systems. We examine whether the improvements achieved in previous chapters by inverse entropy weighting and spectral entropy features extend to multi-stream systems trained with embedded procedure.

The rest of the chapter is organized as follows: the first part of this chapter is related to the oracle studies. In Section 6.1, we discuss the oracle test and its properties. The experimental analyses of the oracle test are presented in Section 6.2, followed by a discussion in Section 6.3. Embedded training experiments are reported in the second part of the chapter, and in Section 6.4 we describe the single-stream embedded training for HMM/ANN systems. The idea of multi-stream embedded training is presented in Section 6.5 along with the experimental results. A summary of the chapter is presented in the last, mentioning the important outcomes of the studies presented in this chapter.

6.1 Oracle Test

6.1.1 Oracle Performance in Multi-stream ASR

In the 'Oracle' experiment, at every time instant (frame), we choose the outputs of the classifier that has the highest posterior for the correct class (Shire and Chen, 2000; Kuncheva, 2002). In essence, the oracle does 1/0 weighting, that is, the outputs of the "best classifier" get a weight of 1 while the outputs of rest of the classifiers get a weight of 0. The oracle takes its decision based on the outputs of the classifier and the correct class label. The correct class was obtained by forced alignment of the test data by the baseline PLP system ($13c$,$13\Delta c$, $13\Delta\Delta c$)[2]. This simple oracle test can let us know the best performance that can be achieved by frame-level weighting for a given set of streams in multi-stream combination.

[2]In general, if hand-labelled data is present for the test utterances, the oracle test can be devised very easily. In absence of hand-labelled data, we can consider forced aligned test data to setup the oracle test.

6.1.2 Complementarity of Feature Streams

Apart from the usual best frame-level performance, the oracle test can also give an indication of the complementarity of the feature streams (feature representations) and streams (outputs of the classifier trained on feature streams). This property of the oracle test could be a step in finding complementary feature streams for a multi-stream system. The proposed interpretation of the oracle test to indicate the complementarity of feature streams has the following argument: If two streams carry exactly the same information, combining those two streams by oracle cannot improve the accuracy of the system. On the other hand, if the two streams carry complementary information, we can achieve an improvement in performance by combining them by oracle. In essence, the more complementary information is available between two streams, the better gains we can attain by combining those two streams.

This property of the oracle test can help in finding whether the feature streams considered for combination carry any complementary information. The oracle test can stop us from looking at feature streams which do not give improvement even in the ideal case. To begin with, we can consider only those feature streams which give good improvement when the outputs of the classifier trained on them are combined by an oracle. It could be a fast method to check whether the streams considered for combination will yield any improvement when combined by sub-optimal methods (Hagen and Bourlard, 2000; Shire and Chen, 2000; Misra et al., 2003). In practice, the improvements achieved by oracle might not be reached by statistical combination methods which rely on the average behavior of the streams. In this setup, it is assumed that the classifiers (MLPs in HMM/ANN systems) are well trained on their respective feature streams.

6.2 Oracle Performance

In this section, we present the performance obtained by oracle for different multi-stream setups. This performance is not the upper bound because the "goodness" of forced-aligned data itself depends on the posterior outputs used for finding the alignment. We have used the posterior outputs of the baseline PLP system to obtain the forced alignment.

In this discussion, we evaluate the performance for two multi-stream systems. The first system uses 7 PLP feature streams, namely static PLP features (c), delta PLP features (Δc), delta-delta PLP features ($\Delta\Delta c$), and all their possible combinations in the FCMS setup, studied in Chapter 4.

The second system uses the baseline PLP features and spectral entropy features defined on a Mel scale in the FCMS setup (3 feature streams) investigated in the previous chapter (Section 5.2.2). In both the systems, one separate MLP is trained for each feature stream and outputs of the MLP classifiers (streams) are considered for combination.

6.2.1 Number of Streams

In the first setup, we increased the number of streams considered for combination from 1 to 7 for the 7 streams PLP system. Fig. 6.1 (a) shows the average word error rates for n streams chosen out

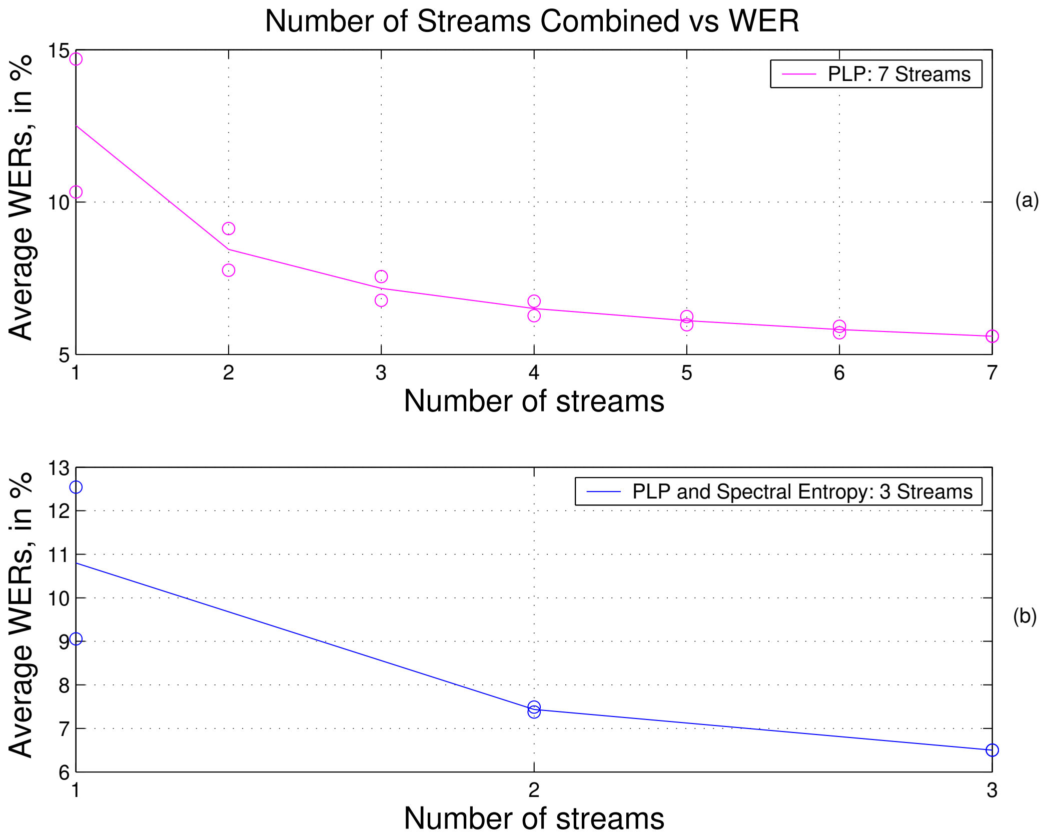


Figure 6.1. *Performance of oracle in % WERs for multi-stream combination. (a) The streams are all possible combinations of the static PLP features and their first and second order time derivatives (7 streams). Out of 7 streams, 'n' streams were used for combination, 'n' varying from 1 to 7. (b) The streams considered for combination were PLP features with first and second order time derivatives, spectral entropy features derived from 24-Mel band with their first and second order time derivatives and concatenation of the two features (3 streams). 'n' varied from 1 to 3.*

of 7 possible streams (we have $C_n^N = \frac{N!}{n!(N-n)!}$ possibilities to choose n streams for combination out of N streams and we considered all the possible combinations to compute the average word error rates). The circles (o) in the figure show the variation of one standard deviation around the average WERs. Similarly, Fig. 6.1 (b) shows the plot for PLP and 24-Mel band derived spectral entropy features used in a full-combination multi-stream setup (3 possible streams), with n varied from 1 to 3.

We observe from these figures that the performance of the oracle improves as the number of streams increases. In the hypothetical case of all the streams used for combination carrying the same information, no improvement can be achieved by combining such streams. Further, the curve starts flattening out when more streams are added, indicating that the additional streams do not bring much complementary information into the system.

6.2.2 Complementarity of Streams

The property of the oracle that it can give information about the complementarity of the streams (as well as feature streams) is depicted in Fig. 6.2. In the figure, we start with the baseline PLP system and start combining other streams to it. When we combine another PLP stream (choosing one from the six remaining streams and considering all the 6 possible combinations) to the baseline PLP stream, we see an improved performance. When we combine the spectral entropy streams (choosing one from the two remaining streams and considering both the combinations), the improvement is more compared to the one observed by adding the PLP streams[3]. A similar trend was observed when we considered 7 PLP streams for combination (Chapter 4) and compared the results with 3 PLP and spectral entropy streams used for combination (Chapter 5). This supports our earlier studies presented in Chapter 5, and indicates that spectral entropy features bring more complementary information into the system. The circles (o) in the figure show the variation of one standard deviation around the average WERs.

We further investigate the following streams/feature streams in a multi-stream setup to compare their complementarity to PLP features:

1. Different MLP sizes: In this setup, PLP features were used to train two MLPs of different sizes. The first MLP is the one through which baseline results are reported in this thesis. The second MLP had 4 times the number of units in the hidden layer than the baseline MLP. The posterior outputs of the two MLPs were combined by an oracle.

[3] We change the number of streams from 1 to 3 because the PLP baseline features when used with the spectral entropy features in the FCMS setup give rise to 3 possible feature streams only.

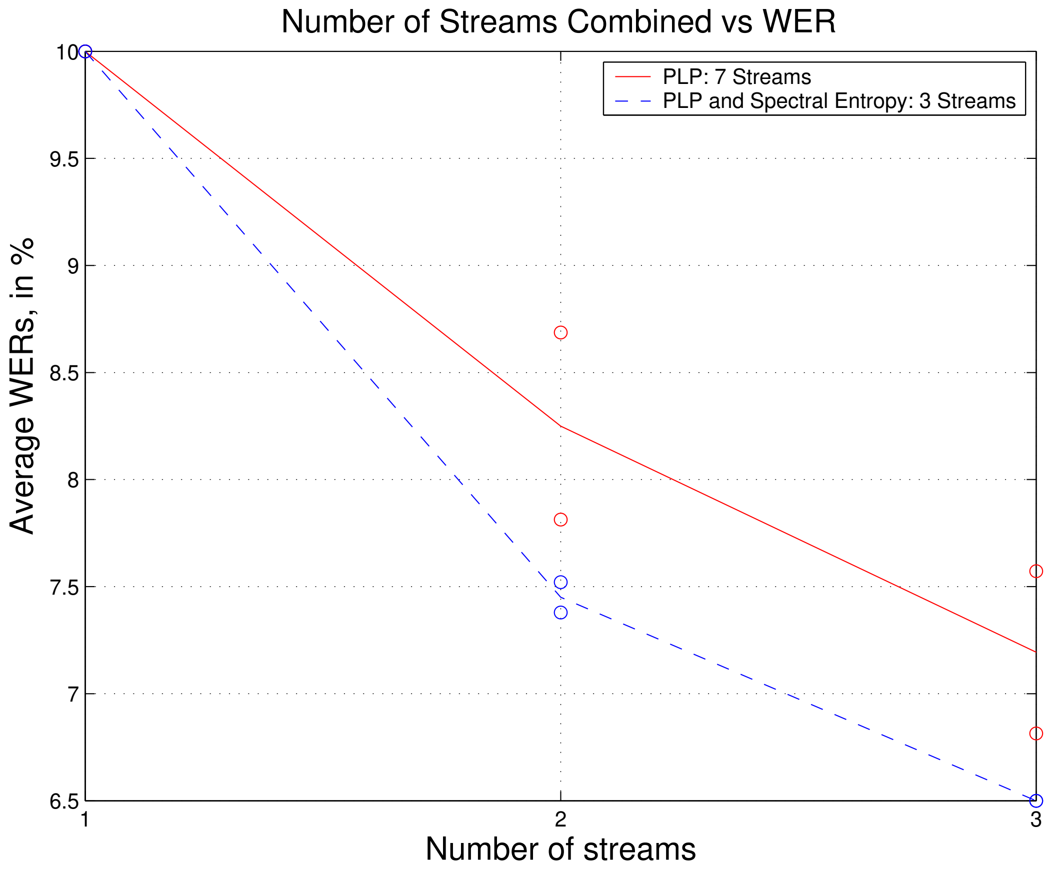


Figure 6.2. *Oracle performance in % WERs to find out complementarity of streams used in multi-stream combination. The performance is compared for PLP features (static, delta and delta-delta) in FCMS (7 streams: —-) and PLP features along with spectral entropy features in FCMS (3 streams: - - -)*

2. 7 PLP streams: This setup was used in Chapters 4. The PLP baseline was used to train an MLP. The remaining 6 possible streams were used as separate streams and one MLP was trained for each of them. In this setup, 6 experiments were conducted, one with each stream. Out of the 6 streams, 1 stream was chosen and posterior outputs of that MLP were combined with the posterior outputs of baseline MLP using an oracle.

3. CJRASTA-PLP: In this setup, one MLP was trained for each feature representation (PLP and CJRASTA-PLP) and the outputs of the two MLPs were combined by an oracle. This is different from FCMS setup where all possible combinations of the two feature representations are used as a separate feature stream.

4. Spectral entropy: The experiments of spectral entropy features and baseline PLP features in an FCMS setup were discussed in Chapters 5. In the present setup, one MLP was trained for

each feature representation and the outputs of the two MLPs were combined by an oracle.

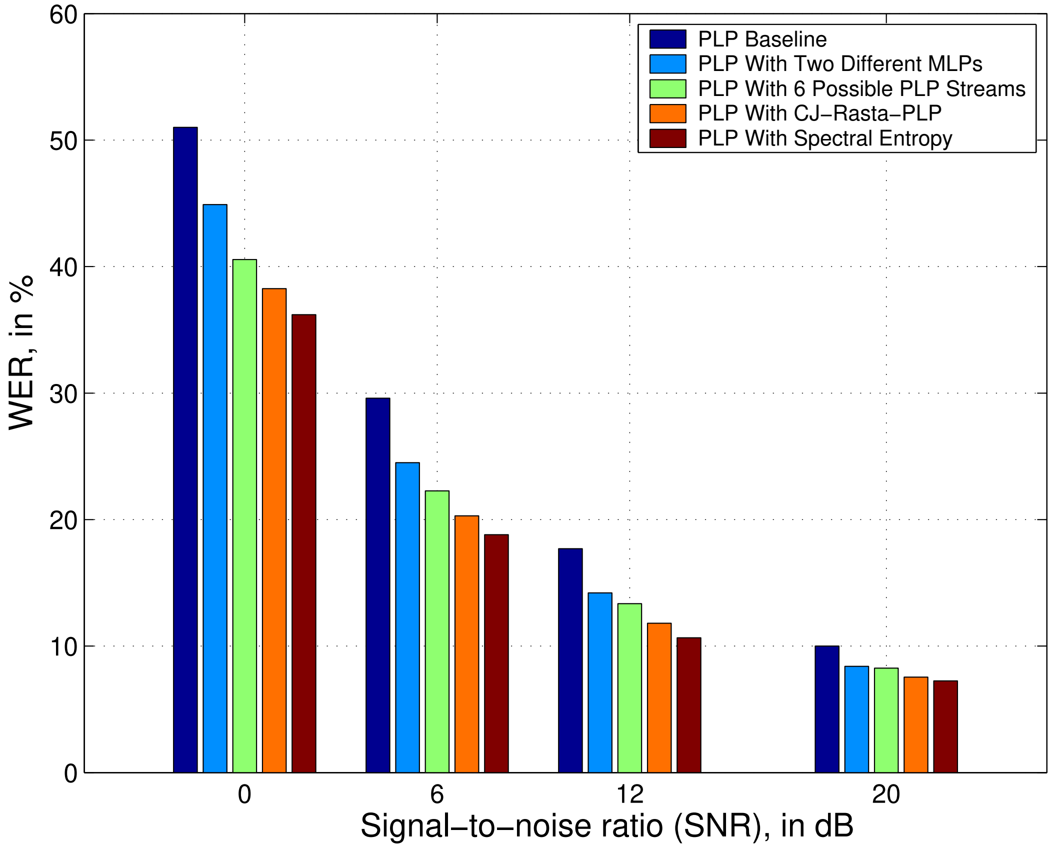


Figure 6.3. *Complementarity of different multi-stream setups using oracle test. The oracle performance in % WERs is compared for: a) Same PLP baseline features with different MLP sizes, b) 6 possible PLP feature streams with baseline PLP features, c) CJRASTA-PLP features with baseline PLP features, and d) Spectral entropy features with baseline PLP features.*

The results of the above 4 setups are shown in Fig. 6.3. We observe from the figure that different MLP sizes trained on the same feature representation to create different streams bring the least complementary information. It was reported in (Janin et al., 1999; Antoniou and Reynolds, 2000; Christensen et al., 2000) that changing the number of parameters in the MLP to create streams brings least improvement in the performance as compared to the improvements obtained by creating streams using different feature representations. The performance of 6 PLP streams shows that they carry more complementary information than creating streams by using different MLP sizes on the same feature representation. CJRASTA-PLP performs better than 6 PLP streams and spectral entropy performs the best. It indicates that CJRASTA-PLP features are more complementary than 6 PLP feature representations and spectral entropy features are most complementary among all the feature representations investigated in this thesis.

The 6 PLP streams are part of the baseline PLP features themselves, therefore they are expected to bring less complementary information than CJRASTA-PLP features. At the same time, like PLP features, CJRASTA-PLP are spectral energy based features and have less complementary information than spectral entropy features.

6.2.3 Relationship with Minimum Entropy

In this section, we analyze how the oracle chooses a particular stream from amongst all the streams. We restrict our studies to analyzing the relationship between oracle selection and the entropy at the output of the MLPs trained on their respective feature streams (in any case, it is rather difficult, if not impossible, to completely understand oracle selection from a statistical point).

In this experimental setup, we computed the entropy of the stream selected by the oracle at each time step, and compared it with the entropy of all other streams. Interestingly, in the case of 7 streams PLP features used for combination, in clean speech, 75.7% of the times the oracle selected the stream with minimum entropy. In case of multi-stream combination of PLP features along with spectral entropy features in full-combination, the oracle selected the minimum entropy stream 79.2% of the times.

Fig. 6.4 shows how many times (frames) the oracle selected the minimum entropy stream for different noise levels (additive factory noise at various SNRs). We notice that as the noise level increases, the preference for minimum entropy frames diminishes, but the minimum entropy frames still enjoy a majority in oracle selection (random selection is 14.4% for 7 streams case and 33.3% for 3 streams case). This suggests that entropy at the output of a classifier is a reasonable choice for weighting.

6.3 Discussion

The important conclusions that can be drawn from this oracle study are:

- It can give us an approximate upper bound on the frame-level performance that can be achieved using a multi-stream setup where the weights are chosen at frame-level.
- The proposed interpretation of an oracle can let us know about the complementarity of the streams used for combination. This can save us from running costly experiments on streams

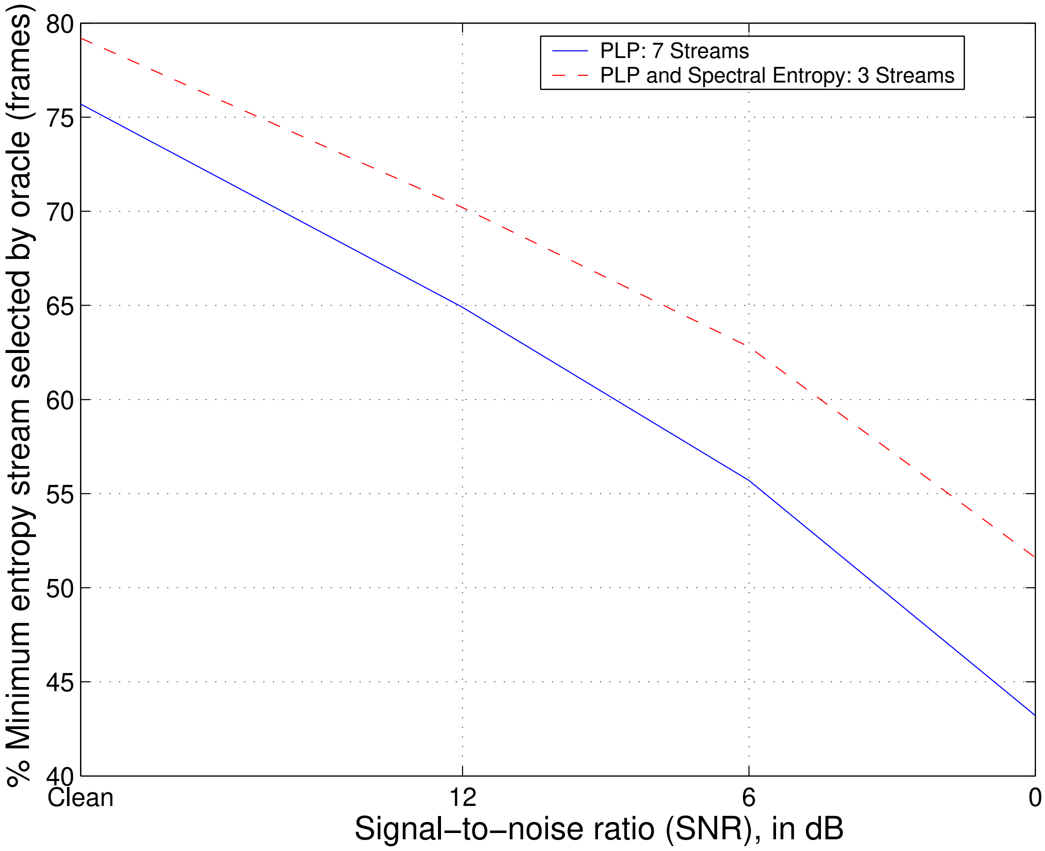


Figure 6.4. *Number of times (in percentage of frames) oracle selected the stream with minimum entropy in the FCMS hybrid systems. The plot is for clean as well as noisy test conditions. Noise conditions are simulated by adding factory noise from the Noisex92 database to the utterances of the Numbers95 database at various SNRs. In the plot, clean speech condition is represented by 20 dB SNR.*

(or feature streams) which might not have enough complementary information to give an improvement in a multi-stream combination system.

- Further, the oracle test establishes that using spectral entropy features along with PLP features in multi-stream combination was a reasonable choice. We observed that spectral entropy features were indeed bringing new information into the system.

- In clean speech, approximately 80% of the times the oracle selected the stream with minimum entropy. Even in case of noise, minimum entropy stream was selected most of the times. However, it might not be possible to define the oracle selection by a statistical measure such as entropy at the output of the classifiers.

6.4 Embedded Training

Embedded training of a single-stream hybrid systems is known to yield an improved performance (Renals et al., 1994; Mirghafori and Morgan, 1997). In this work, we investigate if the improvements achieved on multi-stream combination in Chapter 5 also extend to embedded training of multi-stream combination systems. First, we give the details of single-stream embedded training (Renals et al., 1994; Mirghafori and Morgan, 1997). Next, we propose an embedded training procedure for multi-stream systems.

Embedded training of single-stream hybrid HMM/ANN system was implemented as follows:

1. We started with hand-labelled frame segmentation and trained an MLP, as done in one-hot-encoding hybrid training.

2. The training data was passed as test data through the MLP and the posteriors were obtained.

3. A new state-level segmentation was obtained by Viterbi forced alignment of the posteriors obtained on training data.

4. A new MLP with the same initialization was trained from scratch using the new segmentation.

5. Steps 2, 3 and 4 were repeated several times. In general, 2 iterations might be adequate, but in the present setup we did 15 iterations to observe the trend. In the end, we had one trained MLP for each iteration.

6. Test data was passed through each MLP to obtain the posteriors which were then scaled by their respective priors to obtain scaled likelihoods. The decoding was done using the scaled-likelihood estimates to get the word transcription.

The performance in terms of WER for clean test conditions for every iteration is shown in Fig. 6.5 for the baseline PLP features. We see that the first iteration (going from hand-labelled segments to forced-aligned segmentation for training) gives an absolute improvement of 2% over the baseline. For further iterations, we do not see an obvious pattern between the number of iterations and WER, except that WER hovers around 8% mark achieved in the very first iteration. Further, to study the effect of embedded training in noisy test conditions, we investigated the additive factory noise test conditions. In Fig. 6.6, we present the results obtained from an MLP trained with a forced-aligned segmentation obtained from first iteration. As in all the previous studies, training was performed

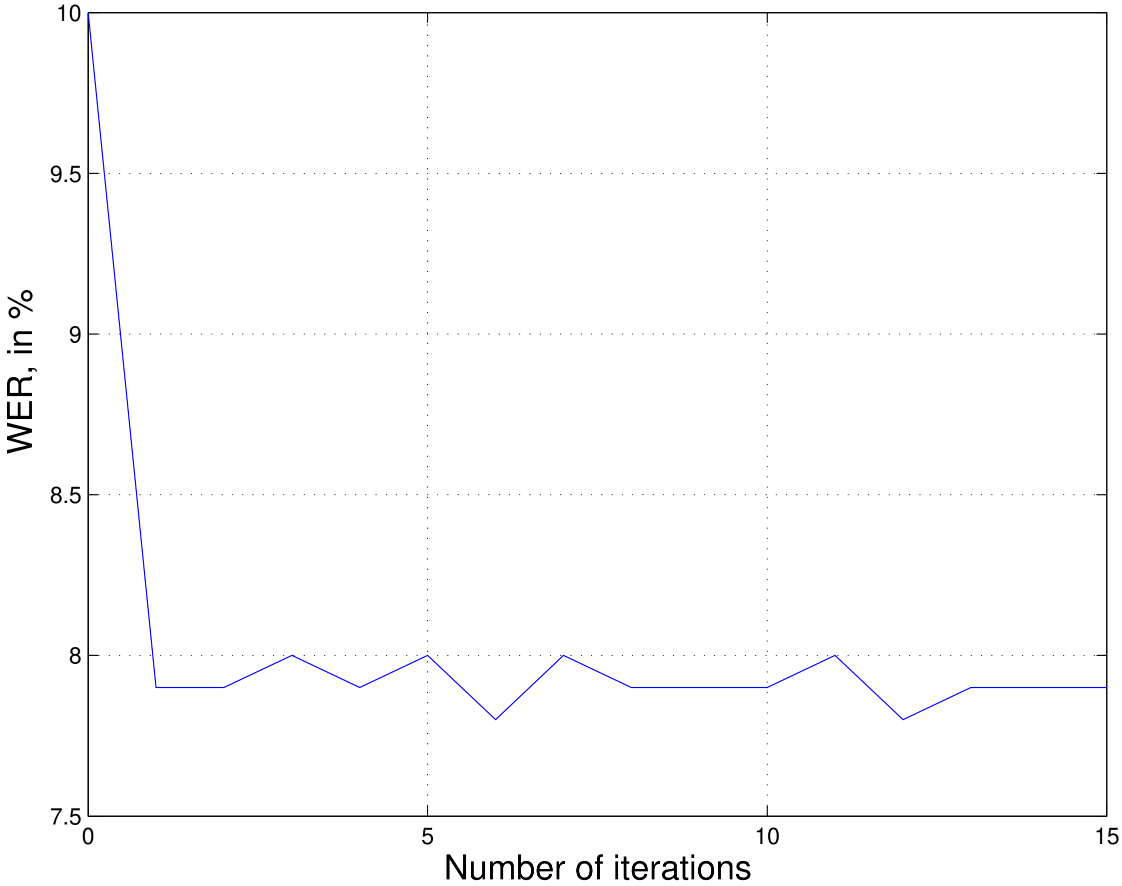


Figure 6.5. *Embedded hybrid training for a single-stream (baseline PLP features): 'number of iterations' vs WER.* 0^{th} *iteration has hand-labelled segmentation for training the MLP. The first iteration gives the maximum improvement.*

on clean data and testing was done on both clean and noisy data. We observe that the improvement seen in clean test conditions is achieved in noisy conditions as well. We obtain a relative average WER improvement of 14.2% over the baseline.

The results obtained by single-stream embedded training encouraged us to examine whether similar trends exist in the case of multi-stream embedded training also. The system setup and the performance of this approach are presented in the next section.

6.5 Multi-stream Embedded Training

There could be more than one method to perform embedded training in a multi-stream system. For example, separate embedded training can be performed for each individual feature stream and the streams (outputs of all the classifiers) can be combined at the time of testing. We followed a rather

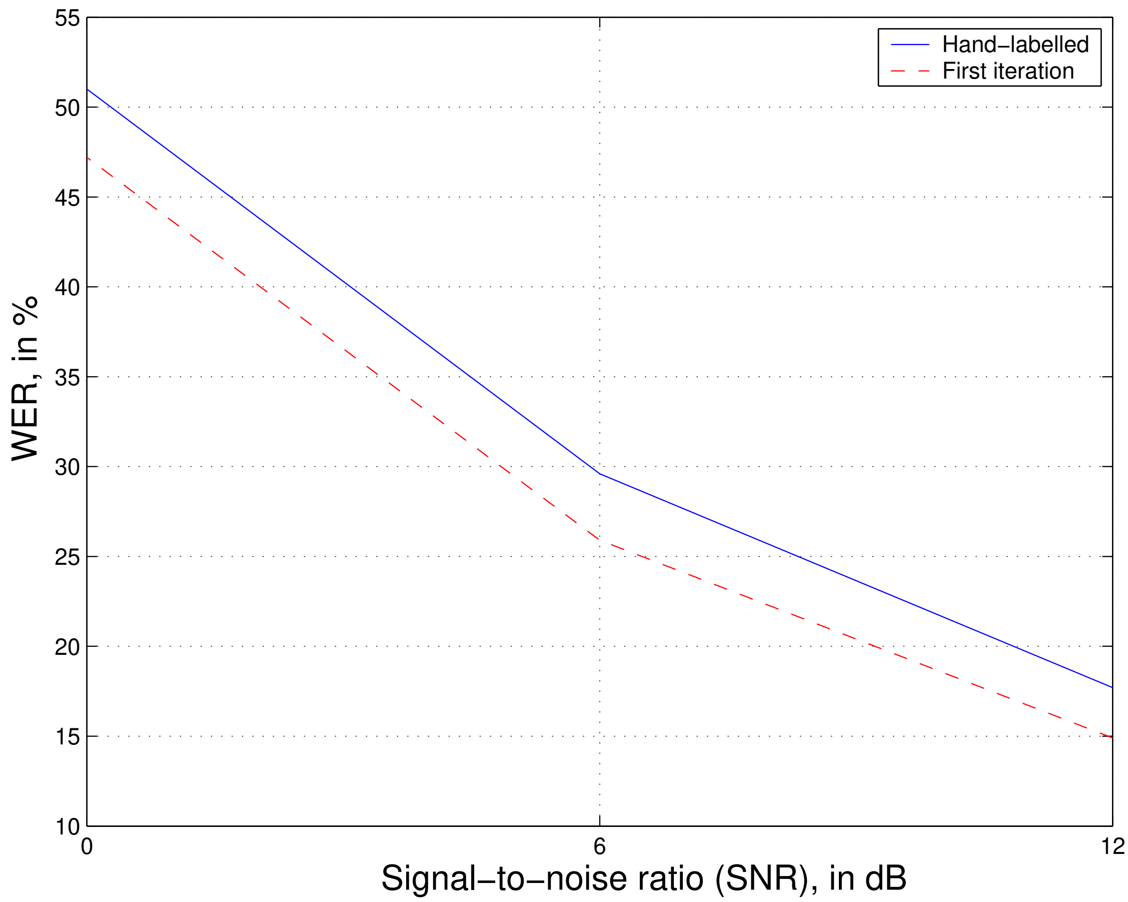


Figure 6.6. *Embedded hybrid training for a single-stream (baseline PLP features): Comparison between WERs obtained by hand-labelled segmentation and with segmentation obtained by forced alignment (first iteration). Noise conditions are simulated by adding factory noise from the Noisex92 database to the utterances of the Numbers95 database at various SNRs.*

easy approach where we trained one MLP for each feature stream but the labels of all the feature streams were same and were obtained by the combined posteriors. The steps of the training were:

1. Starting with hand-labelled frame segmentation, one MLP was trained for each feature stream.

2. The training data of each feature stream was passed as test data through the corresponding MLP and the posteriors were obtained at the output of the respective MLPs.

3. The posterior outputs of different MLPs were combined using the inverse entropy weighting studied in the previous chapters.

4. A new state-level segmentation was obtained by Viterbi forced alignment of the combined outputs obtained on training data.

5. New MLPs with the same initialization were trained from scratch for every feature stream using the new segmentation.

6. Steps 2, 3, 4 and 5 were repeated several times (4 iterations in the present setup). In the end, we had one trained MLP for each feature stream and each iteration.

7. To test the MLPs for each iteration, test data for each feature stream was passed through the respective MLP to obtain the posterior outputs. The posteriors from different MLPs were combined by inverse entropy weighting. The combined posteriors were scaled by their respective priors to obtain scaled likelihoods and decoding was done to get the word transcription.

8. WERs were obtained for each iteration.

The feature streams considered for multi-stream embedded training were: PLP features, spectral entropy features from 24-Mel bands and the concatenation of the two features. The WER performances for different iterations for clean as well as noisy test data (additive factory noise from the Noisex92 database) are shown in Fig. 6.7. As observed in single-stream embedded training, the first iteration with the forced-aligned data gave the maximum improvement and the improvement for the next iteration is subdued. For the third iteration, no improvement in performance is observed (in fact, the performance degrades slightly). The same trend is noticed for different SNRs investigated in this work.

In the bar plot (Fig. 6.8), the performance of the PLP baseline, PLP trained with embedded training (first iteration), multi-stream baseline and multi-stream system with embedded training (first two iterations) are shown for comparison. Embedded training helps in improving the baseline PLP performance as well as the performance of the multi-stream system. The improvement is consistent and generalizes for different noise levels studied in this work. The proposed multi-stream embedded system yields relative average WER improvements of 27.2% compared to the PLP baseline system and 15.3% compared to the single-stream embedded PLP system. Also, the embedded training improves the relative average WER performance of the multi-stream system by 13.4%

The results of embedded training give an impression that we have achieved the performance of the oracle (WER of 6.2% in Fig. 6.1 (b) when all the 3 feature streams are used) by embedded training, but this is not entirely true. In the presence of better segmentation obtained by embedded training and hence better modelling of the acoustic features by the MLPs, the oracle performance

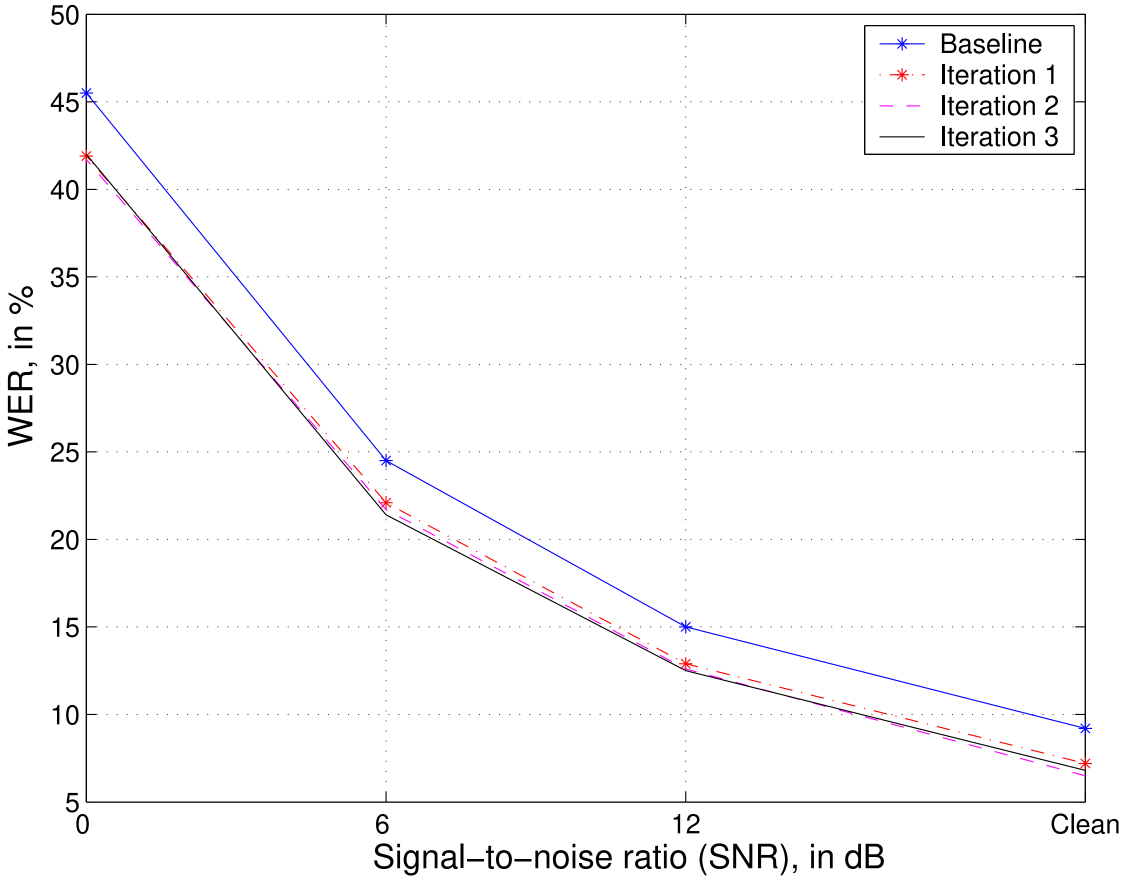


Figure 6.7. *Embedded hybrid training for FCMS: Comparison of performance in % WER with training performed on hand-labelled segmentation and segmentation obtained by forced alignment after each iteration. The performance is compared for clean test condition (20 dB SNR) as well as noisy test conditions. Noise conditions are simulated by adding factory noise from the Noisex92 database to the utterances of the Numbers95 database at various SNRs.*

improves from 6.2% to 4.5%. Embedded training on hybrid systems is similar to embedded training of HMM/GMM systems, and leads to maximization of the likelihood of the training data, often yielding better discrimination between classes (Rabiner, 1989; Rabiner and Juang, 1993; Bourlard and Morgan, 1994). The WERs for the clean test conditions obtained by the different systems are presented in Table 6.1.

Segmentation	**Baseline PLP**	**Multi-stream**	**Oracle**
Hand	10.0	9.1	6.2
Forced-alignment	7.9*	6.5*	4.5*

Table 6.1. *WER in % for training with hand-segments and segments obtained by forced alignment using embedded training (best result for second iteration is shown). a) PLP baseline features, b) multi-stream combination of PLP features with spectral entropy features in FCMS, and c) Oracle. Testing on clean conditions only. In the table, * indicates that the improvement in performance as compared to the system trained on hand-segmented data is significant.*

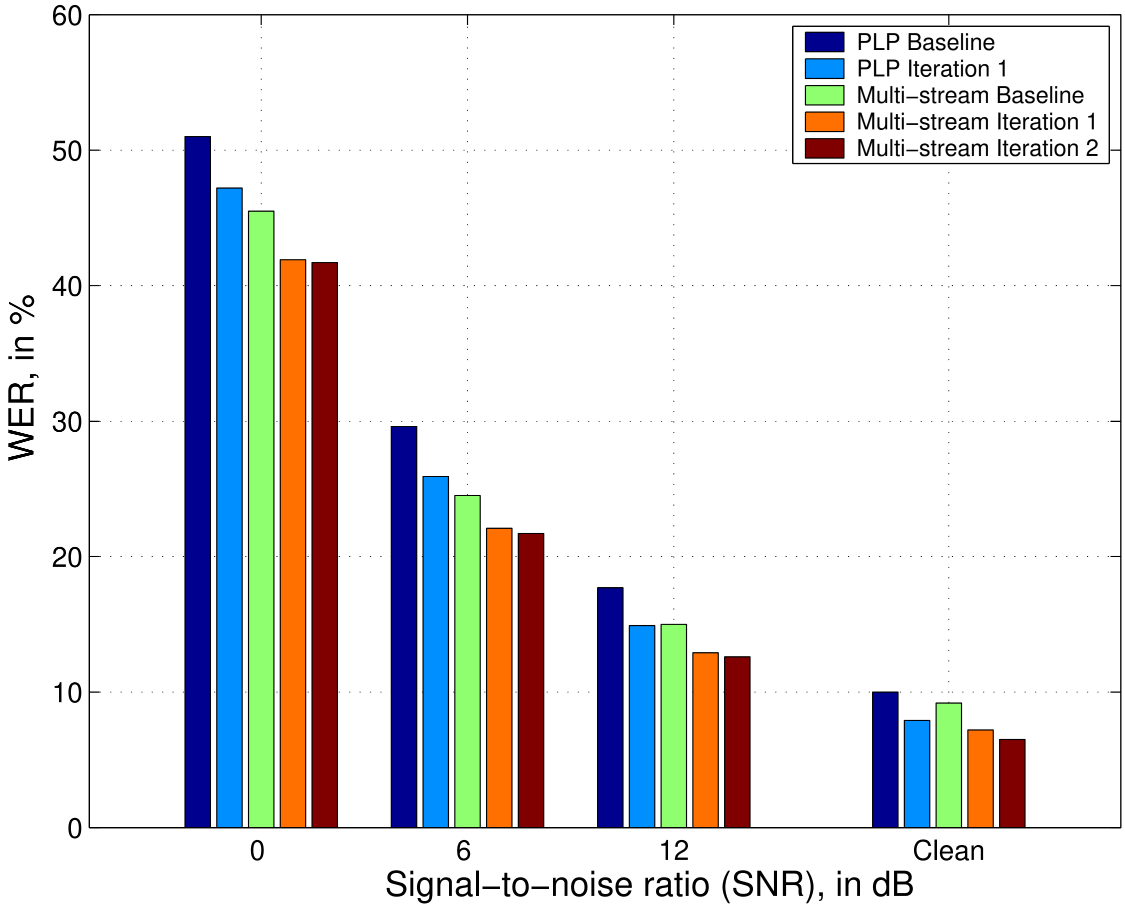


Figure 6.8. *Performance in % WER for PLP features with hand segmentation, PLP features with segmentation obtained by forced alignment during embedded training (first iteration), PLP and spectral entropy features in FCMS with inverse entropy weighting and hand segmented labels, PLP and spectral entropy features in FCMS with segmentation obtained by forced alignment during embedded training (first two iterations). The performance is shown for clean test condition (20 dB SNR) as well as noisy test conditions. Noise conditions are simulated by adding factory noise from the Noisex92 database to the utterances of the Numbers95 database at various SNRs.*

6.6 Summary

In this chapter, we presented a frame-level oracle test for multi-stream systems and analyzed its characteristics. We outlined the need for the oracle test to investigate the complementary properties of the new feature representations. We could show the complementarity of spectral entropy features using the oracle test. Also, we found that the oracle tends to choose the outputs of the MLP classifiers (trained on feature streams) that had the least entropy at their outputs. This further supported our proposed method of inverse entropy weighting for combining the outputs of the classifiers.

In the second part of the chapter, we proposed an embedded training procedure for hybrid multi-stream systems. The study was carried out to investigate whether the gains obtained on simple multi-stream systems by techniques studied in earlier chapters, namely inverse entropy weighting and spectral entropy features, extend to multi-stream systems trained with an embedded procedure. We observed that the multi-stream embedded training can lead to improved performance, not only in clean test conditions but for noisy test conditions as well. In clean condition, we achieved a WER improvement of 2% absolute (20.0% relative) over the baseline PLP system by employing single-stream embedded training. We further gained a WER drop of close to 1.5% absolute (17.7% relative) on clean conditions by multi-stream embedded training over the single-stream embedded training applied to PLP baseline features.

Chapter 7

Multi-stream Combination in Tandem ASR Systems

In Chapters 4, 5 and 6, we presented the results for hybrid HMM/ANN systems using models of context-independent phones. The outputs of the MLP in an HMM/ANN system are posterior estimates and it is easy to interpret them. In addition, posteriors being bound between 0 and 1, it is convenient to use them in multi-stream combination systems. HMM/ANN systems can be trained for context-dependent phones also where the number of outputs units of the ANN is equal to the number of context-dependent phones. However, with an increase in vocabulary size, the number of context-dependent phones increases, and training the MLP gets computationally expensive (Doss, 2005). For this reason, ANNs are generally trained for context-independent phones.

State-of-the-art HMM/GMM systems employ state-tying and context-dependent training with ease. However, the outputs of the HMM/GMM system being likelihoods, it is not easy to use them in multi-stream combination. In the next section, we describe the Tandem system which is a combination of HMM/ANN and HMM/GMM systems (Hermansky et al., 2000).

The organization of the remaining chapter is as follows: in Sections 7.1 and 7.2, we describe single-stream and multi-stream Tandem systems respectively. The experimental results obtained by using the inverse entropy weighting and spectral entropy features in multi-stream Tandem system are presented in Section 7.3. We summarize the important contributions of the chapter in Section 7.4.

7.1 Tandem System

In a Tandem system, we train the MLP of the usual HMM/ANN system. The output posteriors of the MLP are fed to the HMM/GMM system after some processing so that the inputs to the HMM/GMM system are Gaussian-like and decorrelated. The HMM/GMM system uses standard techniques like state-tying and context-dependent modelling to model the HMM parameters. The Tandem system has been shown to be noise robust (Sharma et al., 2000; Hermansky et al., 2000) and yields improved performance compared to a standard HMM/ANN system or HMM/GMM system.

In (Zhu et al., 2004), the authors showed that the speaker variations at the output of an MLP were much less compared to the speaker variations in the PLP features at the input of the MLP. This reduction is speaker variance is beneficial for the HMM/GMM stage of a Tandem systems.

Depending upon how the outputs of the MLP are obtained and processed, there are two variations of Tandem systems proposed in the literature.

7.1.1 Tandem: Softmax Outputs

The Tandem system, as suggested originally in (Hermansky et al., 1999), is depicted in Fig. 7.1. In

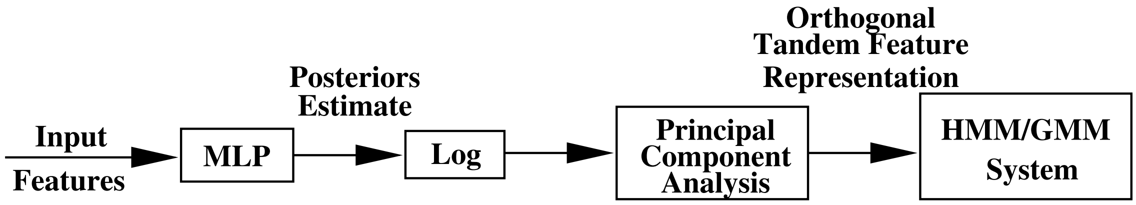


Figure 7.1. *Tandem Posterior Model: Posteriors from the MLP are* log *scaled and then decorrelated by PCA. The transformed posteriors are used as features in a standard HMM/GMM system (Hermansky et al., 1999).*

this system, the posterior outputs of the MLP are first transformed by log and then decorrelated by principal component analysis (PCA) before being fed as features to an HMM/GMM system. Both the steps are necessary so that the transformed posteriors can be modelled by a mixture of Gaussians having a diagonal covariance matrix. Techniques like state-tying and context-dependent modelling can be utilized to train the HMM/GMM system. The steps of developing a Tandem-based ASR system are listed below:

1. An MLP is trained for a given feature representation (same as in hybrid HMM/ANN system). It is possible to use task-specific training data or data from some other databases (task independent training data) to train the MLP. In (Sivadas and Hermansky, 2004), the authors ob-

served that training on task-specific data performs better than training on task-independent data for an ASR task. In our studies, we used features obtained from task-dependent data to train the MLP.

2. The features obtained from the training data are passed through the trained MLP (forward pass) to generate the posteriors for the training data.

3. The posteriors from training data are log transformed and decorrelated by PCA. In this work, all the dimensions are preserved while doing PCA and the dimension of the decorrelated posterior vectors is the same as that of the dimension of the posterior vectors. In some recent papers (Zhu et al., 2004, 2005a), the authors evaluated the role of PCA to reduce the dimension before feeding the decorrelated posterior vectors to the HMM/GMM system. The PCA basis obtained from the training data is stored to be used at the time of testing.

4. Using the transformed posteriors as feature vectors ("termed orthogonal Tandem feature representation" in Fig. 7.1), an HMM/GMM system is trained.

5. During testing, the features obtained from the test data are forward passed through the trained MLP to generate posteriors.

6. The log of the posteriors is taken and, and the resultant outputs are projected on the PCA basis obtained from the training data.

7. The transformed test posteriors are used as test features in the HMM/GMM system and decoding is performed to obtain the word sequence.

7.1.2 Tandem: Linear Outputs

The output posteriors of the Tandem system discussed above are obtained by a softmax activation function in the output layer of the MLP. The softmax function is given by,

$$P(q_k|x_n) = \frac{exp(f(y_k|x_n))}{\sum_i exp(f(y_i|x_n))} \tag{7.1}$$

where $f(y_k|x_n)$ and $P(q_k|x_n)$ are the linear and softmax outputs respectively, for k^{th} class and feature vector x_n at time instant n. The output after softmax, $P(q_k|x_n)$, is the estimated posterior probability at the output of the MLP for the k^{th} class. The relationship between output before and after the softmax is a many-to-one mapping and we lose some information in the process. This information loss can be avoided if the MLP output is taken from linear output units instead of softmax outputs.

The "linear" Tandem topology was suggested in (Sharma et al., 2000) and it gives a better performance compared to the one obtained with softmax output. The schematic diagram of a linear Tandem system is shown in Fig. 7.2. Comparing the linear and softmax Tandem systems, we notice

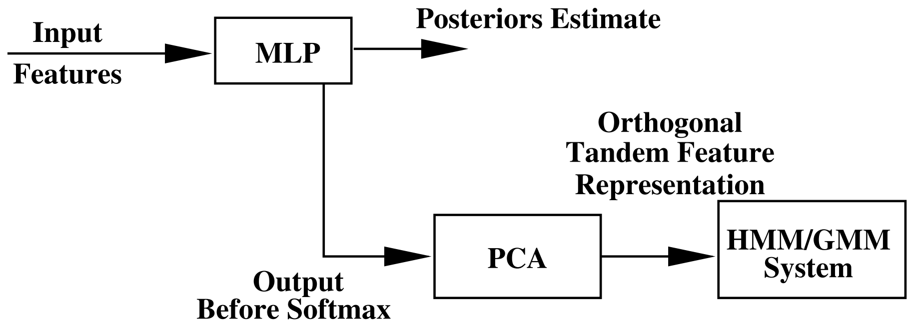


Figure 7.2. *Tandem Linear Model: 'Outputs before softmax' from the MLP are decorrelated by PCA and used as features in a standard HMM/GMM system.*

that the log module is not used in the linear system. The linear outputs are Gaussian like and do not require the log transformation. The rest of the system details are same as explained for softmax Tandem system.

7.2 Multi-stream Tandem

The MLP in the first stage facilitates the use of a Tandem system in multi-stream combination. The outputs of MLPs trained on different feature streams can be combined with the methods discussed in Chapter 4. The combined outputs can be used as features after some required pre-processing in a standard HMM/GMM system. As discussed above, there are two variations of Tandem system. We have used both of them in our multi-stream combination studies. The system details are explained further in the following discussion.

Multi-stream Tandem: Softmax outputs

Fig. 7.3 explains the working of a multi-stream Tandem system with softmax outputs. In this system, we combine the posteriors obtained from MLPs trained on different feature streams. In FCMS, as shown in the figure, for each possible feature combination one MLP is trained. All the

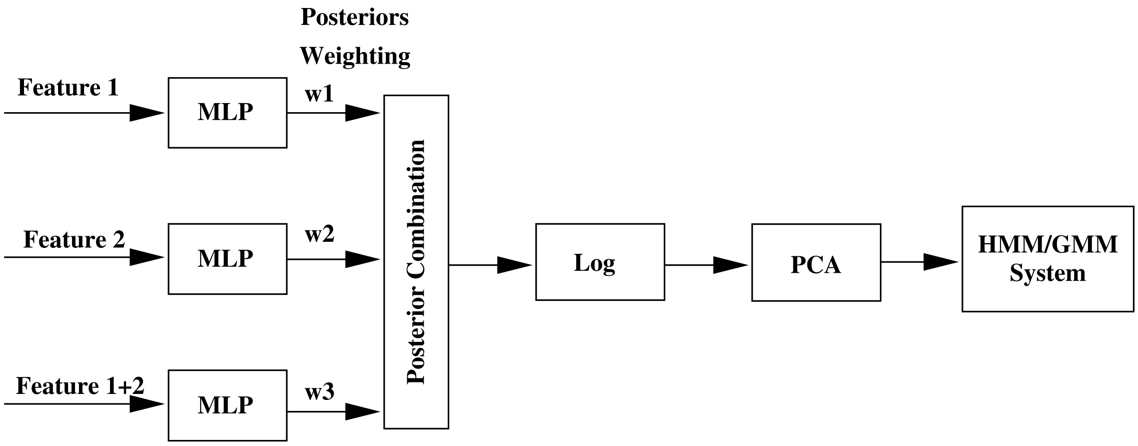


Figure 7.3. *Multi-stream Softmax Tandem: Posteriors from different MLPs are weighted and combined. The combined output undergoes* log *scaling followed by PCA before being fed as features into an HMM/GMM system.*

weighting techniques studied in Chapter 4 can be used directly to combine the outputs of MLP classifiers.

Multi-stream Tandem: Linear outputs

The multi-stream Tandem system for linear outputs is shown in Fig. 7.4. There are certain issues when dealing with the linear outputs. For example, the outputs are no longer restricted to be

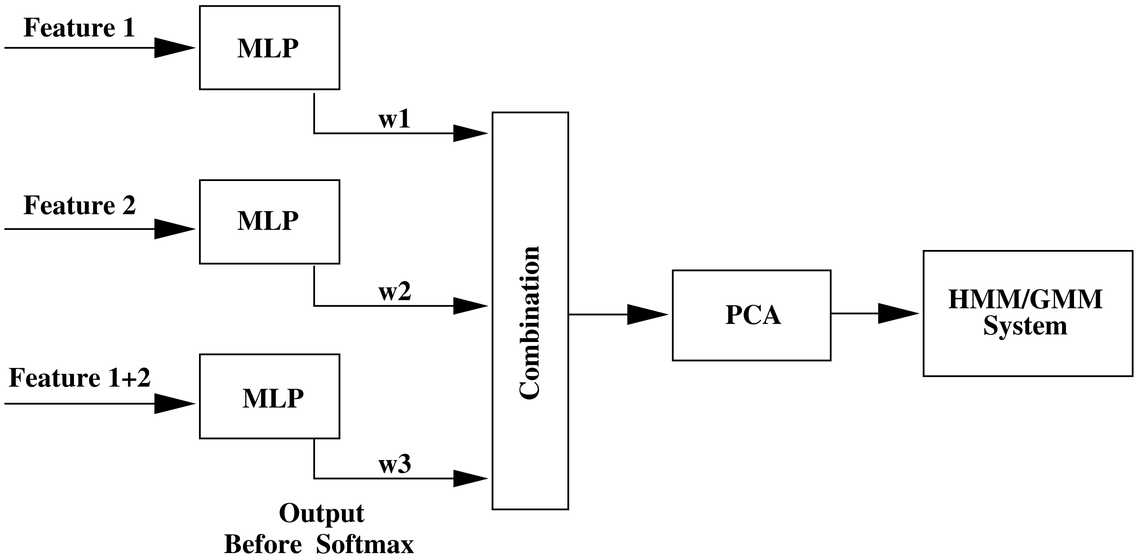


Figure 7.4. *Multi-stream Linear Tandem: 'Outputs before softmax' from different MLPs are weighted and combined. The combined output undergoes PCA before being fed as features into an HMM/GMM system.*

between 0 and 1, and they can take positive as well as negative values. Therefore, we cannot apply product rule to combine their outputs (summation is still possible). Also, the weighting strategies of Chapter 4 are not applicable directly.

With a little modification, we can use the inverse entropy weighting in the multi-stream linear Tandem setup. We converted the linear outputs of an MLP into posteriors using (7.1) and computed entropy from these posteriors at the output of each MLP employing (4.6). The linear outputs of the MLPs were weighted by their respective inverse entropies using (4.11) to get the combined linear outputs by a summation operation.

7.3 Experimental Setup and Results

In the previous section, we discussed two variations of Tandem systems. In this section, the results are presented when the two of them are used in multi-stream combination.

The MLP of the Tandem system was the same as that of the hybrid system. The HMM/GMM part of the Tandem system consists of 80 context-dependent phones with 3 left-to-right states per context-dependent phone. Emission probabilities for each state were modelled by a mixture of 12 Gaussians. PLP and spectral entropy features studied in Chapter 5 were used as individual feature streams in the FCMS Tandem framework. The inverse entropy weighting technique investigated in Chapter 4 was used for combining the outputs of the MLP classifiers.

Table 7.1 shows the results of two variations of the Tandem system for clean and noisy test conditions. Noise conditions are simulated by adding factory noise from the Noisex92 database to the test utterances of the Numbers95 database at various SNRs. We notice that the performance

	Softmax Tandem				**Linear Tandem**			
Feature	**Clean**	**SNR12**	**SNR6**	**SNR0**	**Clean**	**SNR12**	**SNR6**	**SNR0**
PLP	5.5	12.0	22.1	44.2	4.3	10.3	20.1	41.9
24-Mel	8.6	13.9	22.1	40.8*	7.1	12.1	19.9	37.7*
PLP, 24-Mel	5.5	11.9	22.2	45.1	4.2	9.7	18.5*	41.1
FCMS: PLP,24-Mel	**5.2**	**10.9***	**19.6***	**39.8***	**4.0**	**9.6**	**17.6***	**37.5***

Table 7.1. *WERs in % for PLP features, 24 Mel-band spectral entropy feature and its time derivatives (24-Mel), the two features appended (PLP, 24-Mel), and the two features in full-combination multi-stream (FCMS: PLP,24-Mel) in the Tandem systems for the Numbers95 database corrupted by additive factory noise from the Noisex92 database at various SNRs. The numbers in* **bold** *show the best performance and * indicates that the improvement in performance as compared to the baseline system is significant.*

of the Tandem system with softmax outputs (Softmax Tandem) is inferior than the performance of the Tandem system with linear outputs (Linear Tandem). This trend is observed for single as well as multi-stream systems for different noise conditions presented in the table and is in-line with the results reported in (Ellis et al., 2001).

Further, the table shows that PLP features work well in low noise conditions whereas spectral entropy features work well in high noise conditions. We notice that the concatenation of two features (Row 3: PLP,24-Mel) improves the performance for low noise conditions, however the improvement is more when the two features are used in the FCMS framework with the inverse entropy weighting (Row 4: FCMS: PLP,24-Mel). These observations are similar to the trends obtained on hybrid system (In Table 5.5, we saw that the FCMS framework gave better performance than feature concatenation).

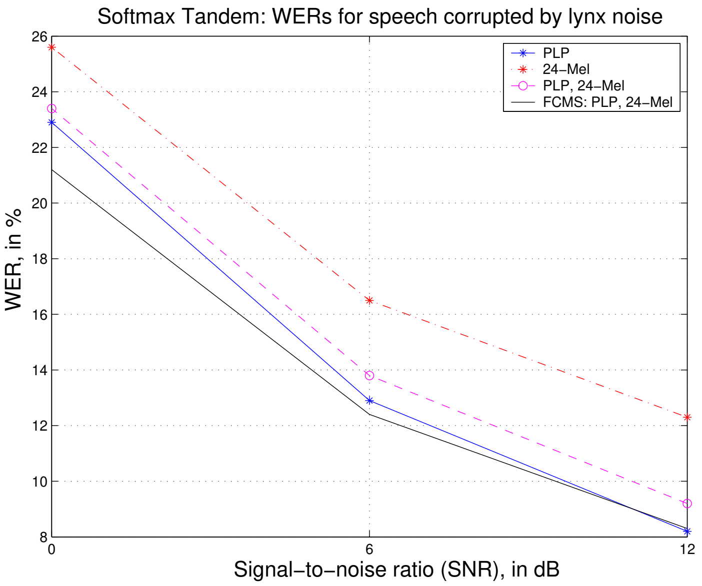
The relative average WER improvements of 9% and 9.2% were obtained by using the inverse entropy weighting and the spectral entropy features in the framework of multi-stream Softmax and Linear Tandem systems respectively.

The results for lynx and car noises for different SNR conditions for multi-stream Softmax Tandem system are presented in Fig. 7.5. Once again we notice that the FCMS framework gives better performance than concatenating the feature streams and the trends observed for hybrid system (Figs. 5.3 and 5.4) are replicated in the Softmax Tandem system.

Similar plots for the multi-stream Linear Tandem system for additive lynx and car noises at different noise levels are given in Fig. 7.6. For car and lynx noises, the difference in performance between concatenating the features and FCMS framework is reduced. This suggests that for these noises, inverse entropy weighting which works well for posteriors combination is less effective for linear outputs combination. The reason for this could be that the linear outputs have high dynamic range. It is difficult to reduce the effect of wrong high output for a phoneme class using outputs of different streams for the same phoneme class. Similar problem affects the usefulness of multi-stream in HMM/GMM systems where emission likelihoods have a high dynamic range.

7.4 Summary

In this chapter, we investigated Tandem systems using multi-stream combination. Spectral entropy features (Chapter 5) were used along with PLP features in the framework of FCMS extended to the Tandem systems. Two topologies of Tandem were explored in the framework of FCMS. The



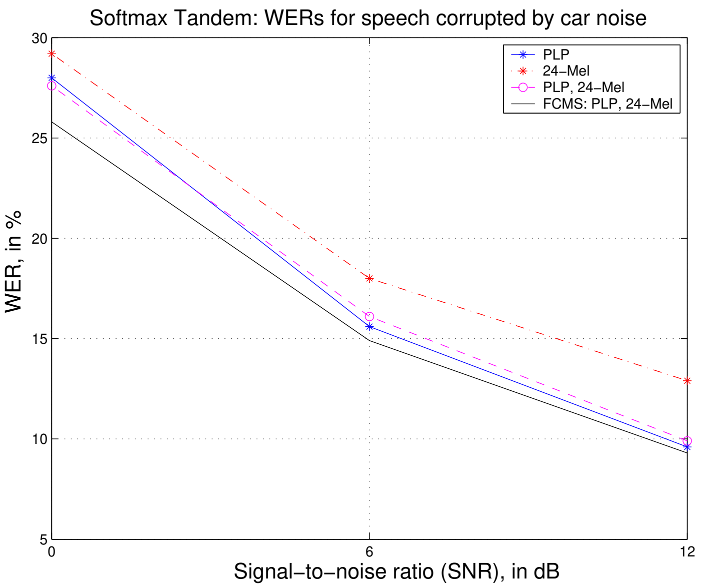


Figure 7.5. *Multi-stream Softmax Tandem (outputs after softmax): Plot of WERs for different feature streams for the Numbers95 database (Top):* **lynx noise** *added from the Noisex92 database at various SNRs, (Bottom):* **car noise** *added from the Noisex92 database at various SNRs. PLP features (-*-), spectral entropy features (-*.), the two features concatenated (-o-) and the two features in the FCMS framework with inverse entropy weighting (—).*

following is the summary of the results presented in this chapter:

1. The Tandem systems with linear outputs were found to be consistently better than the Tandem systems with softmax outputs. This trend was observed for clean speech as well as various noise conditions studied in this chapter.

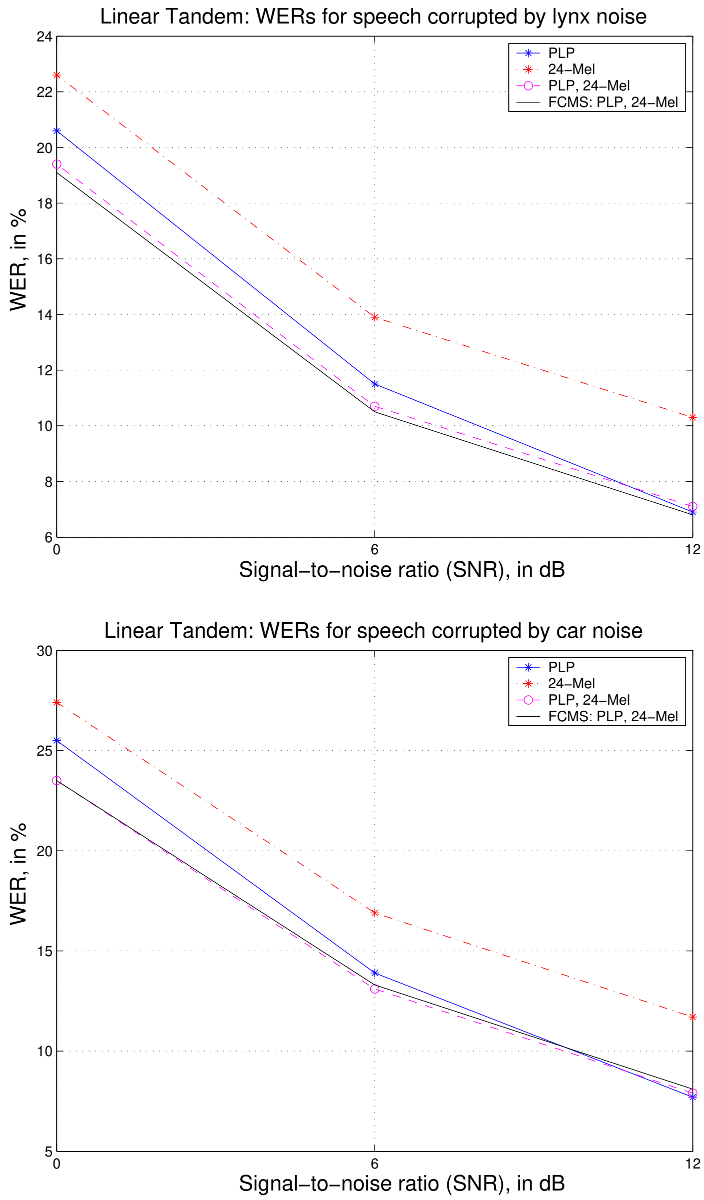


Figure 7.6. *Multi-stream Linear Tandem (outputs before softmax): Plot of WERs for different feature streams for the Numbers95 database (Top):* **lynx noise** *added from the Noisex92 database at various SNRs, (Bottom):* **car noise** *added from the Noisex92 database at various SNRs. PLP features (-*-), spectral entropy features (-*.), the two features concatenated (-o-) and the two features in FCMS with inverse entropy weighting (—).*

2. Similar to the hybrid system, PLP features performed well in low noise conditions while the performance of spectral entropy features was good in high noise conditions.

3. The inverse entropy weighting technique studied in the previous chapters gave an improvement in multi-stream Tandem system also.

4. Once again, we found that modelling the features separately and then combining the outputs of the MLPs usually performs better or the same as modelling the features jointly. The combined system performed better than the baseline for different noise cases and various SNRs, and the improvement was more for low noise conditions.

The results for Tandem systems presented in this chapter followed the similar trend that was observed in Chapter 5 while studying hybrid HMM/ANN systems. This validates the spectral entropy features and inverse entropy weighting combination on two different models.

Chapter 8

Large Vocabulary ASR

The analysis and results presented in Chapters 4, 5, 6 and 7 were a on small vocabulary database (Numbers95). In this chapter, the methods proposed in this thesis are validated on a large vocabulary task. The task, setup and results are explained in the next few sections.

Spectral entropy features gave good performance when combined with PLP features on the Numbers95 task. In this chapter, we investigate the performance of spectral entropy features on a large vocabulary conversational telephone speech (CTS) recognition task when they are combined with PLP features in a multi-stream setup using inverse entropy weighting. We used multi-stream Tandem systems to develop the large vocabulary ASR system. The feature extraction process in the CTS task was same as the one used in the connected digit recognition task reported in Chapters 4, 5, 6 and 7. Also, the MLP architecture was kept the same. However, the associated decoding process used for the CTS task was modified significantly and is explained later in the chapter.

The remaining chapter is organized as follows: First, we give a description of the CTS database used in this study. In Section 8.2, we briefly explain the feature streams and the ASR system details employed to carry out the experiments. We present the results obtained on the CTS task in Section 8.3, followed by a short summary.

8.1 Database and MLP Training

The CTS database used in this study consists of telephone quality speech collected from a subset of the following six corpora: *Switchboard 1*, *Switchboard 2*, *Switchboard Cellular*, *Callhome English*, *Fisher* and *Switchboard Credit-Card*. The original database had 32.87 hours of training data for males and 36.08 hours for females. Out of this, 15 hours of training data was identified as representative data for each gender to reduce the training time. The representative training data was defined such that it gave a performance that was similar to the performance obtained by using the original data for training. Separate male and female MLPs were trained using the respective training database of each gender. Approximately 10% of the training data was used for cross-validation while training the MLP. For female speakers, this resulted in 14420 sentences for training the MLP and 1575 for cross-validation. Similarly, male speakers had 13522 and 1489 sentences for training and cross-validation of the MLP respectively.

The database is represented by 47 phonemes, including 'laugh' and 'reject' also as phoneme classes. The phonemes which could not be categorized into any specific phoneme were put in the 'reject' phoneme class. The phoneme 'reject' was not used while training the MLP. So, effectively only 46 phonemes were used for training and accordingly the MLP had 46 output units, one for each phoneme.

The 1000 most common words occurring in the DARPA Effective, Affordable, Reusable Speech-to-Text (EARS) project CTS task mentioned above were identified for testing. For males[1], the development set comprised of 951 utterances and was used for tuning the parameters related to the back-end such as grammar model and state-tying. The test set had 1009 utterances in it.

8.2 Components

The main components of the ASR system employed for the CTS task were as follows:

8.2.1 Feature Streams

The feature representations used in the multi-stream CTS ASR system were:

1. PLP-derived cepstral coefficients: The PLP features[2] had undergone vocal-tract normalization, and mean and variance normalization on per speaker basis.

2. Spectral entropy feature: In Chapter 5, spectral entropy features obtained from 24 overlapping sub-bands defined on a Mel scale yielded the best performance on the Numbers95

[1] As pointed out by Zhu et al. (2004), our partners at ICSI, female set showed similar results on CTS task.
[2] Obtained from International Computer Science Institute (ICSI), Berkeley.

database. Therefore, we used the same setup to obtain spectral entropy features on the CTS task. While vocal-tract normalization was not performed to obtain the spectral entropy features, mean and variance normalization were done on per utterance basis.

8.2.2 Inverse Entropy Weighting

In Chapter 4, inverse entropy weighting with average threshold was observed to give the best improvement. This method also gave a reasonably good performance when PLP and spectral entropy features were combined in hybrid and Tandem systems in Chapter 5 and 7 respectively.

The inverse entropy weighting technique was successfully employed by ICSI (Morgan et al., 2004; Zhu et al., 2004; Chen et al., 2004) to reduce WER on a CTS task[3] using a different set of features (PLP and TRAPs), validating its usefulness for other setups. In (Morgan et al., 2004), the authors unintentionally combined outputs of a classifier obtained from a badly degraded TRAPs feature stream with the other features. The inverse entropy weighting could reduce "the importance of the poor stream so that the overall performance essentially matched what was achieved for a feature vector that consisted of the baseline PLP features concatenated with the PLP/MLP feature alone" (Morgan et al., 2004, Page 538).

8.2.3 Multi-stream Tandem ASR

We used the full-combination multi-stream (FCMS) framework in this study, and a separate MLP was trained for each individual feature stream and their combination by concatenation. The number of output units in the MLPs was 46 (compared to 27 for Numbers95) and the rest of the MLP structure was kept the same.

Tandem: Softmax Output

The training of the multi-stream Softmax Tandem system was performed as follows:

1. We had three feature streams, namely PLP features, spectral entropy features and the combination of the two features by concatenation. Three MLPs were trained in the FCMS framework, one MLP for each feature stream.

[3] The size of the database was much larger in these tasks.

2. Features obtained from training data were passed through their respective MLPs and the posteriors were obtained at the output of the MLPs.

3. Inverse entropy weighting studied in Chapter 4 was used to weight the posterior outputs of individual MLP classifiers.

4. The combined output was transformed by log and decorrelated using PCA before being fed to the HMM/GMM system as features for training.

The testing of the multi-stream Softmax Tandem system involved the following steps:

1. At the time of testing, the posterior outputs for all the feature streams from their respective MLPs were obtained.

2. The posteriors were combined using inverse entropy weighting.

3. The combined posteriors went through a log transformation and projected on the PCA basis obtained from the training data. The transformed outputs were used as features to test the HMM/GMM system.

Tandem: Linear Outputs

In the Linear Tandem setup, the outputs before the softmax nonlinearity were taken from all the MLP classifiers and combined by inverse entropy weighting. The combined outputs were decorrelated by PCA and used as a feature to train and test the HMM/GMM system.

8.2.4 HMM Training and Decoding

The training of the HMM/GMM system was performed using the HTK system (Young et al., 1997). The setup used in the system was:

1. Context-dependent phone models were used and there were 3 emitting states for each model. The states were connected left to right.

2. Gaussian mixture with 32-components were used for modelling the emission probability density of each state.

3. State-tying was performed to merge the states having very little data. Two methods were used to perform the state-tying. In the first method, if the number of observation vectors in a

state were below a predefined threshold the state was merged with the neighbouring state. In the second method, the number of states were fixed a-priori and accordingly the states having less data were merged.

4. Bigram language models were used while decoding[4].

8.3 Results

The results for baseline and different multi-stream systems for the male set are presented in Table 8.1 for comparison. In the CTS task, we do not have noisy test conditions, therefore results

Feature	**Tandem: Softmax**	**Tandem: Linear**
PLP	50.0	48.2
24-Mel	60.7	60.2
PLP + 24-Mel	50.1	48.7
FCMS	**49.0**	**47.9**

Table 8.1. *WERs in % on the CTS database for different feature streams and their combinations. 24-Mel represents the spectral entropy features obtained from 24 overlapping sub-bands defined on a Mel scale. PLP + 24-Mel represents the spectral entropy features appended to the PLP features (feature combination). The numbers in* **bold** *show the best performance.*

are presented for the clean condition only. In the case of the Tandem system with softmax outputs, the result for PLP features is better than for the spectral entropy features obtained from 24 overlapping sub-bands defined on a Mel scale. The reasons for the performance difference between PLP and spectral entropy features can partially be attributed to the fact that PLP features were obtained after VTLN and cepstral normalization on a per speaker basis, which are known to improve the performance of individual feature streams. It is noticed that concatenating the two features does not improve the performance over the baseline. In contrast, using the two feature streams in FCMS with inverse entropy weighting gives an improvement. Similar results were obtained on the Numbers95 database in Chapter 7.

Comparing the two Tandem systems, once again we notice that the Tandem system having linear outputs performs better than the Tandem system with softmax outputs. In the case of linear outputs, the spectral entropy features' performance is inferior to that of the PLP features and concatenation did not achieve any improvement in the performance. In the framework of FCMS, again the two feature streams give an improved performance. Similar trends were observed on the

[4] HTK training modules and the language models were provided by the University of Washington, one of our partners in the EARS project.

Numbers95 database in Chapter 7. These results validate our findings on the CTS task and support the methods we have proposed in this work.

The important results of these experiments can be summarized as follows:

1. Tandem with linear outputs gave better performance than Tandem with softmax outputs.

2. Combination at posterior-level is found to be better than combination at feature-level.

3. The performance of PLP features in clean speech was better than the performance of spectral entropy features. The difference in performance may be partially due to VTLN and speaker level mean and variance normalization performed on the PLP features. However, when the two features were combined by inverse entropy weighting in the FCMS setup, we observed an improvement in the performance of the ASR system.

8.4 Summary

In this chapter, the promising methods proposed in the earlier chapters were investigated using a CTS database. Spectral entropy features and FCMS with inverse entropy weighting in a Tandem system, which gave a significant improvement in performance on the limited-vocabulary Numbers95 database (Chapter 7), gave an improvement in performance on the CTS task as well. This validates the usefulness of our methods and further supports our analysis presented on the Numbers95 database.

Chapter 9

Conclusions

This thesis addressed on the important issue of robustness in ASR towards additive noise. Motivated by the reasoning that combining evidences from complementary sources of information can improve the robustness of a system (Furui, 1986; Dupont and Luettin, 1998; Morgan et al., 1998; Kirchhoff, 1998; Hagen and Morris, 2005), we pursued a multi-stream combination approach to address the problem. After introducing various components of multi-stream systems, the following two important issues were investigated further:

1. new weighting techniques such that the streams (outputs of the classifiers) get weight according to their reliability;
2. new feature streams which might carry complementary information when compared to existing feature streams (PLP features were used as a baseline).

The frameworks of hybrid HMM/ANN and Tandem systems were used to carry out the experimental work. The posterior outputs of MLPs in the two systems make them a good candidate for multi-stream combination studies.

9.1 Weighting Techniques

While investigating the weighting techniques, we developed a maximum-posterior (MP) weighting method where the outputs of an MLP classifier get a weight directly proportional to the maximum posterior at the output of that classifier. This simple technique gave us a reasonable improvement in performance for the 7 PLP streams considered in the FCMS approach. Next, we proposed inverse entropy weighting and suggested that the outputs of an MLP classifier can be weighted inversely proportional to the entropy at the output of that classifier. The inverse entropy weighting had an advantage over maximum posterior weighting because entropy captures the posterior distribution at the output as opposed to maximum posterior which uses only the highest posterior probability value. FCMS with inverse entropy weighting gave consistently better performance over the baseline. The third and the last method proposed was maximum-likelihood (ML) weighting. In maximum-likelihood (ML) weighting, the goal was to increase the likelihood of the combination and investigate whether it improves the discrimination between the classes as well. In the absence of targets, though the likelihood of the data was increased, the discrimination between the classes could not be improved, and therefore the performance of the system was also not improved.

9.2 Features

We studied two features in the framework of multi-stream ASR. Fundamental frequency (referred as pitch frequency in this thesis), a feature that captures the characteristics of the excitation signal, was considered for the first set of experiments along with PLP features which characterize the response of vocal tract. It was observed that appending the pitch feature does not improve the ASR performance. However, the suggested full-combination multi-stream framework was able to obtain an improvement using the pitch feature in the clean speech condition. In the case of noise, due to unreliable estimation of the pitch feature, the performance could not be improved over the baseline.

Subsequently, we proposed and investigated spectral entropy as a feature for ASR. We divided the normalized spectrum into sub-bands (overlapping as well as non-overlapping sub-bands were studied) and computed entropy of each sub-band. The multi-band spectral entropy was used as a feature representation in ASR. We found that the overlapping sub-bands defined on a Mel scale gave the best performance. Spectral entropy features were observed to be robust to additive noise and gave slightly lower performance in low noise conditions (when compared with PLP features). It was suggested that combining spectral entropy features with PLP features might give improved performance across all conditions. In the framework of FCMS with inverse entropy weighting, when spectral entropy features were combined with PLP features, a consistent improvement was observed for clean speech as well as for different additive noise types and noise levels studied in this thesis.

We also realized that modelling the features separately and then combining the posteriors (late integration) was a better approach than combining the features and then modelling them (early integration). This result is similar to observations reported in (Kirchhoff and Bilmes, 2000).

9.3 Oracle and Embedded Training

To evaluate the potential performance of a multi-stream system that could be achieved for a set of feature streams, we investigated an "oracle test". In this setup, one separate classifier was trained for each feature stream and outputs of the classifiers (streams) were considered for combination. We explored a frame-level weighting where an "oracle" gave us the correct label for each frame, and then out of all the available streams, the stream having the highest posterior for the correct class was chosen. Through this oracle test, we could find the gap between the performance achieved by our proposed techniques and the best performance achievable by frame-level weighting. A new interpretation of the oracle test proposed in this thesis gave us insight into the complementarity of the streams suggested for combination.

It was observed that the oracle selected minimum entropy streams most of the times, even at high noise levels, suggesting that the inverse entropy weighting was an attempt in the right direction.

In the same chapter, we suggested embedded training for multi-stream combination for hybrid systems. The proposed training method led to a significant improvement in performance for both clean as well as noisy test conditions.

9.4 Multi-stream Tandem ASR

In this thesis, we introduced inverse entropy weighting and spectral entropy features to multi-stream Tandem systems. It was observed that inverse entropy weighting and spectral entropy features brought similar improvements in performance in the Tandem system to those that were observed in the hybrid system setup. Using the proposed techniques, we improved the basic Tandem system itself and also bring an additional gain over a very strong baseline on the Numbers95 ASR task (Cerisara, 1999; Mirghafori, 1999; Sharma, 1999; Hagen, 2001; Ikbal, 2004; Doss, 2005).

9.5 CTS Task

The proposed techniques (inverse entropy weighting in multi-stream, spectral entropy features and multi-stream linear Tandem) were used for a CTS task. In this task, a large vocabulary telephone quality continuous speech database defined for the DARPA EARS project was used. In the CTS task, in clean conditions, we observed a similar improvement over the baseline with the suggested methods that was observed on the Numbers95 database.

In short, the methods proposed in this thesis were validated on two different types of systems and two different databases, yielding a reasonable improvement in performance in almost all cases.

9.6 Future Directions

This thesis made an attempt to address the issue of robustness in ASR systems. In the framework of multi-stream combination, we investigated some weighting techniques and new features. Some of the issues pertaining to multi-stream ASR which need further investigation are:

1. The weighting techniques studied in this work used local frame-level confidence measures to combine the posterior outputs of MLPs trained on different feature streams (ML weighting considered global optimization, but in absence of targets it did not yield good results). A recently proposed hierarchical multi-stream approach combines the posterior outputs of different MLPs with some contextual knowledge (Ketabdar et al., 2005). It is assumed in this approach that the contextual information available over the whole utterance should be used along with available prior knowledge to obtain better posterior estimates when the streams are combined. Furthermore, it has been suggested that similar hierarchical combination can be performed at phoneme and word-levels as well. Such a method of combination is promising because it does not rely only on local confidence measures.

2. In the methods suggested in this thesis, the posterior combination and modelling are done locally and independently. A local piecewise approach like this is unable to capture the dependencies among the streams in an optimal manner. A promising model to integrate the information from multiple sources is a dynamic Bayesian network (DBN) (Zweig et al., 2002). DBNs can model multiple sources of information by providing flexibility in modelling the dependencies between the sources and integrating prior knowledge about them. Such an architecture is

more suitable for multi-stream combination systems than the piecewise architecture of combination followed my modelling. However, modelling different dependencies between sources leads to a more complex model along with higher computational cost.

3. The oracle test investigated in this work showed that if weights are chosen properly, even simple streams can give significant improvement in performance. We observed that oracle and minimum entropy weightings had a close relationship, but minimum entropy cannot describe the complete nature of the oracle. It is a possibility that using more than one confidence measure can explain the oracle selection more closely.

Appendix A

Auxiliary Function Maximization

As required in Section 4.3, in this appendix we show that maximization of auxiliary function, $A(\theta, \theta^s)$, leads to likelihood maximization, $p(X|\theta^{s+1})$, in ML weighting.

From (4.19), the auxiliary function is given by,

$$\begin{aligned} A(\theta, \theta^s) &= E_Q\big[\log p(X, Q|\theta)|X, \theta^s)\big] \\ &= \sum_q P(q|X, \theta^s) \log p(X, Q|\theta) \\ &= \sum_q P(q|X, \theta^s) \log\big(P(q|X, \theta)\ p(X|\theta)\big) \\ &= \sum_q \Big(P(q|X, \theta^s) \log P(q|X, \theta)\Big) + \log p(X|\theta) \end{aligned} \tag{A.1}$$

Substituting θ^s in (A.1), we get

$$A(\theta^s, \theta^s) = \sum_q \Big(P(q|X, \theta^s) \log P(q|X, \theta^s)\Big) + \log p(X|\theta^s) \tag{A.2}$$

Subtracting (A.2) from (A.1), we obtain

$$\begin{aligned} A(\theta, \theta^s) - A(\theta^s, \theta^s) &= \sum_q \Big(P(q|X, \theta^s) \log P(q|X, \theta)\Big) - \sum_q \Big(P(q|X, \theta^s) \log P(q|X, \theta^s)\Big) \\ &\quad + \log p(X|\theta) - \log p(X|\theta^s) \\ &= \sum_q \Big(P(q|X, \theta^s) \log \left[\frac{P(q|X, \theta)}{P(q|X, \theta^s)}\right]\Big) + \log \left[\frac{p(X|\theta)}{p(X|\theta^s)}\right] \end{aligned} \tag{A.3}$$

We can rearrange (A.3) as,

$$\log\left[\frac{p(X|\theta)}{p(X|\theta^s)}\right] = A(\theta,\theta^s) - A(\theta^s,\theta^s) + \sum_q \Big(P(q|X,\theta^s)\log\left[\frac{P(q|X,\theta^s)}{P(q|X,\theta)}\right]\Big) \tag{A.4}$$

The left-hand side of (A.4) is the likelihood ratio with change in parameters of the model from θ^s to θ. The right-hand side of (A.4) has two parts, a) the difference between the two auxiliary functions evaluated at θ^s and θ, and b) Kullback-Leibler (KL) divergence which is either 0 or positive.

If the value of the auxiliary function increases going from θ^s to θ, the log-likelihood ratio will also increase. Therefore, maximizing the auxiliary function by changing the parameters of the model (θ) ensures that we also maximize the likelihood of the data.

Appendix B

Forward and Backward Variables

As mentioned in Section 4.3, the derivation for forward variable, α, and backward variable, β, has been provided in this appendix.

$\alpha(k,t)$ is defined as *the likelihood of having generated the sequence* $x_1^t = \{x_1, \cdots, x_t\}$ *and being in state* k *at time instant* t. It is given by,

$$
\begin{aligned}
\alpha(k,t) &= p(x_1^t, q_t = k) \\
&= p(x_t | x_1^{t-1}, q_t = k)\; p(x_1^{t-1}, q_t = k) \\
&\approx p(x_t | q_t = k) \sum_{m=1}^{K} p(x_1^{t-1}, q_t = k, q_{t-1} = m) \\
&\approx \left(\sum_{i=1}^{I} p(x_t, b_i | q_t = k) \right) \left(\sum_{m=1}^{K} p(x_1^{t-1}, q_{t-1} = m)\; P(q_t = k | x_1^{t-1}, q_{t-1} = m) \right) \\
&\approx \left(\sum_{i=1}^{I} p(x_t | b_i, q_t = k)\; P(b_i | q_t = k) \right) \left(\sum_{m=1}^{K} P(q_t = k | q_{t-1} = m)\; p(x_1^{t-1}, q_{t-1} = m) \right) \\
&\approx \left(\sum_{i=1}^{I} p(x_t | b_i, q_t = k)\; P(b_i | q_t = k) \right) \left(\sum_{m=1}^{K} P(q_t = k | q_{t-1} = m)\; \alpha(m, t-1) \right)
\end{aligned}
$$

Initial condition for α is, $\alpha(k,0) = P(q_0 = k) =$ Initial state probability.

$\beta(k,t)$ is defined as *probability to generate the rest of the sequence* $x_{t+1}^T = \{x_{t+1}, \cdots, x_T\}$ *given* *that we are in state k at time instant t*. It is given by,

$$
\begin{aligned}
\beta(k,t) &= p(x_{t+1}^T|q_t = k) \\
&= \sum_{m=1}^{K} p(x_{t+1}^T, q_{t+1} = m|q_t = k) \\
&= \sum_{m=1}^{K} p(x_{t+1}|x_{t+2}^T, q_{t+1} = m, q_t = k)\; p(x_{t+2}^T, q_{t+1} = m|q_t = k) \\
&\approx \sum_{m=1}^{K} p(x_{t+1}|q_{t+1} = m)\; p(x_{t+2}^T|q_{t+1} = m)\; P(q_{t+1} = m|q_t = k) \\
&\approx \sum_{m=1}^{K} \left(\sum_{i=1}^{I} p(x_{t+1}, b_i|q_{t+1} = m) \right) \beta(m, t+1)\; P(q_{t+1} = m|q_t = k) \\
&\approx \sum_{m=1}^{K} \left(\sum_{i=1}^{I} p(x_{t+1}|b_i, q_{t+1} = m)\; P(b_i|q_{t+1} = m) \right) \beta(m, t+1)\; P(q_{t+1} = m|q_t = k)
\end{aligned}
$$

Final condition for β is, $\beta(k,T) = 1$ if k is the final state (otherwise it is 0).

Appendix C

Comparison of Spectral Entropy and Spectral Variance Features

In this appendix, the results obtained by multi-band spectral entropy features and multi-band spectral variance features are presented. In these preliminary experiments, we used a 15-point smooth PLP spectrum to obtain the features.

The feature extraction process can be summarized as follows:

1. The PLP spectrum was normalized and divided into (J) non-overlapping sub-bands. The number of sub-bands was varied from 1 to 5.

2. For J sub-bands, spectral entropy and variance were estimated for each sub-band, and a spectral entropy feature vector and spectral variance feature vector were formed by concatenating the estimates from each sub-band. The dimensionality of both the feature vectors was the same as number of sub-bands (J).

3. Feature vectors from $J = 1$ to 5 were concatenated to form a feature vector of dimension 15 ($= 1 + 2 + 3 + 4 + 5$).

4. First and second order time derivatives were appended to include the temporal information for both spectral entropy and spectral variance features.

The results for both the feature vectors in a hybrid HMM/ANN system framework for clean as well as noisy test conditions are presented in Table C.1. The table shows that the spectral entropy

Feature	Clean	SNR12	SNR6	SNR0
Spectral Variance	17.7	27.4	43.3	66.9
Spectral Entropy	**15.3***	**25.2***	**42.0***	**66.0**

Table C.1. *WERs in % for spectral entropy and spectral variance features obtained from a smooth PLP spectrum using a hybrid system. The results are for the Numbers95 database and the noise conditions are simulated by adding factory noise from the Noisex92 database at various SNRs. The numbers in* **bold** *show the best performance and * indicates that the difference in performance between the two systems is significant.*

feature vector[1] performs better than the spectral variance feature vector.

[1]These were preliminary results. In Chapter 5, we show the results for spectral entropy features extracted from the STFT of the signal. The results obtained on the STFT spectrum are better than the ones obtained on the PLP spectrum. The higher frequency resolution of the STFT spectrum could be the reason for those better results.

Appendix D

Oracle Combination of PLP, CJRASTA-PLP and Spectral Entropy Features

In Table D.1, the results are presented for an oracle test when CJRASTA-PLP and spectral entropy features are used individually with PLP features. The setup of an FCMS hybrid system was used for carrying out the experiments. Recalling that oracle can indicate the complementarity of feature streams (Section 6.2.2), the table shows that spectral entropy features carry more complementary information than CJRASTA-PLP features, when each of the features are used with PLP features.

Features	Clean	Factory Noise			Car Noise		
		SNR12	SNR6	SNR0	SNR12	SNR6	SNR0
FCMS Oracle (PLP, CJRASTA)	6.6	9.8	16.7	32.1	8.9	12.1	21.8
FCMS Oracle (PLP, 24-Mel)	**6.2**	**8.4**	**14.7**	**27.9**	**8.7**	**12.0**	**19.7**

Table D.1. *WERs in % for CJRASTA-PLP features and 24 Mel-band spectral entropy feature with its time derivatives (24-Mel) along with PLP features in FCMS for oracle selection. Results are for a hybrid system on the Numbers95 database. The noise conditions are simulated by adding the factory and car noises from the Noisex92 database at various SNRs. The numbers in* **bold** *show the best performance.*

The trend observed in Table D.1 is reflected in Table 5.6 (Page 80) as well. In Table 5.6, we observed that most often combining spectral entropy features with PLP features yielded better performance than combining CJRASTA-PLP features with PLP features.

Bibliography

Ajmera, J., McCowan, I., and Bourlard, H. (2003). Speech/music discrimination using entropy and dynamism features in a HMM classification framework. *Speech Communication*, 40:351–363.

Allen, J. B. (1994). How do humans process and recognize speech? *IEEE Trans. Speech, Audio Processing*, 2(4):567–577.

Antoniou, C. and Reynolds, T. (2000). Acoustic modeling using modular/ensemble combination of heterogeneous neural networks. In *Proceedings of International Conference on Spoken Language Processing*, pages 282–285.

Arons, B. (1992). A review of the cocktail party effect. *Journal of the American Voice I/O Society*, 12:35–50.

Atal, B. S. (1976). Automatic recognition of speakers from their voices. *Proc. IEEE*, 64(4):460–475.

Bagshaw, P. C., Hiller, S. M., and Jack, M. A. (1993). Enhanced pitch tracking and the processing of F0 contours for computer aided intonation teaching. In *Proceedings of European Conference on Speech Communication and Technology*, pages 1003–1006.

Bahl, L. R., Jelinek, F., and Mercer, R. L. (1983). A maximum-likelihood approach to continuous speech recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 5(2):179–190.

Baum, L. E., Petrie, T., Souled, G., and Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Ann. Math. Statist.*, 41(1):164–171.

Bengio, S. (2003). An asynchronous hidden Markov model for audio-visual speech recognition. In *Advances in Neural Information Processing Systems (NIPS)*, volume 15, pages 1237–1244.

Berouti, M., Schwartz, R., and Makhoul, J. (1979). Enhancement of speech corrupted by acoustic noise. In *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing*, pages 208–211.

Berthommier, F. and Glotin, H. (1999). A measure of speech and pitch reliability from voicing. In *International Joint Conference on Artificial Intelligence (IJCAI), Computational Auditory Scene Analysis (CASA) Workshop*, pages 61–70, Stockholm, Sweden.

Bilmes, J. A. (1998). A gentle tutorial of the EM algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. ICSI-TR-97-021, University of Berkeley, ICSI.

Bisani, M. and Ney, H. (2004). Bootstrap estimates for confidence intervals in ASR performance evaluation. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, Montreal, Canada.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*, chapter 2, pages 33–34. Clarendon Press, Oxford.

Bishop, C. M. (1999). *Neural Networks for Pattern Recognition*, chapter 9. Oxford University Press.

Boll, S. F. (1979). Supression of acoustic noise in speech using spectral subtraction. *IEEE Trans. Acoust., Speech, Signal Processing*, 27:113–120.

Bourlard, H. (1999). Non-stationary multi-channel (multi-stream) processing towards robust and adaptive ASR. In *Proceedings of the ISCA Workshop on Robust Methods for Speech Recognition in Adverse Conditions*, pages 1–9.

Bourlard, H. and Dupont, S. (1996). A new ASR approach based on independent processing and recombination of partial frequency bands. In *Proceedings of International Conference on Spoken Language Processing*, pages 426–429, Philadelphia, PA, USA.

Bourlard, H., Dupont, S., and Ris, C. (1996). Multi-stream speech recognition. IDIAP-RR 7, IDIAP.

Bourlard, H. and Morgan, N. (1994). *Connectionist Speech Recognition - A Hybrid Approach*. Kluwer Academic Press, 101 Philip Drive, Assinippi Park, Norwell, Massachusetts 02061 USA.

Bourlard, H. and Wellekens, C. J. (1989). Speech pattern discrimination and multi-layered perceptrons. *Computer, Speech and Language*, pages 1–19.

Buxton, B. F. and Langdon, W. B. (2001). Data fusion by intelligent classifier combination. *Measurement and Control*, 34(8):229–234.

Cerisara, C. (1999). *Contribution de l'approche multi-bande á la reconnaissance automatique de la parole*. PhD thesis, Institut National Polytechnique de Lorraine, Nancy, France.

Cerisara, C., Fohr, D., and Haton, J.-P. (2000). Asynchrony in multi-band speech recognition. In *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing*, pages 1121–1124, Istanbul, Turkey.

Chen, B., Zhu, Q., and Morgan, N. (2004). Learning long-term temporal features in lvcsr using neural networks. In *Proceedings of International Conference on Spoken Language Processing*, Jeju Island, South Korea.

Chen, T. (2001). Audiovisual speech processing. *IEEE Signal Processing Magazine*, 18:9–21.

Cherry, E. C. (1953). Some experiments on the recognition of speech, with one and two ears. *J. Acoust. Soc. Amer.*, 25(5):975–979.

Cherry, E. C. and Taylor, W. K. (1954). Some further experiments on the recognition of speech, with one and two ears. *J. Acoust. Soc. Amer.*, 26(4):554–559.

Christensen, H., Lindberg, B., and Anderson, O. (2000). Employing heterogeneous information in a multi-stream framework. In *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing*, pages 1571–1574.

Clarkson, P. and Rosenfeld, R. (1997). Statistical language modelling using the CMU-Cambridge toolkit. In *Proceedings of European Conference on Speech Communication and Technology*, Rhodes, Greece.

Cohen, J., Kamm, T., and Andreou, A. G. (1995). Vocal tract normalization in speech recognition: Compensating for systematic speaker variability. *J. Acoust. Soc. Amer.*, 97(5):3246–3247.

Cole, R., Noel, M., Lander, T., and Durham, T. (1995). New telephone speech corpora at CSLU. In *Proceedings of European Conference on Speech Communication and Technology*, volume 1, pages 821–824.

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

Davis, S. B. and Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Trans. Acoust., Speech, Signal Processing*, pages 357–366.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *J. Roy. Statist. Soc. Ser. B*, (39):1–38.

Doss, M. M. (2005). *Using auxilliary sources of knowledge for automatic speech recognition*. PhD thesis, École Polytechnique Fédérale de Lausanne.

Doss, M. M., Stephenson, T. A., and Bourlard, H. (2003). In *Proceedings of European Conference on Speech Communication and Technology*, Geneva, Switzerland.

Dupont, S. (2000). *Études et Développement de nouveaux paradigmes pour la reconnaissance robuste de la parole*. PhD thesis, Laboratoire TCTS, Université de Mons, Belgium.

Dupont, S. and Bourlard, H. (1997). Using multiple time scales in a multi-stream speech recognition system. In *Proceedings of European Conference on Speech Communication and Technology*, Rhodes, Greece.

Dupont, S. and Luettin, J. (1998). Using the multi-stream approach for continuous audio-visual speech recognition: Experiments on the M2VTS database. In *Proceedings of International Conference on Spoken Language Processing*, pages 1283–1286.

Ellis, D. P. W. and Bilmes, J. A. (2000). Stream combination before and/or after the acoustic model. In *Proceedings of International Conference on Spoken Language Processing*, Beijing, China.

Ellis, D. P. W., Singh, R., and Sivadas, S. (2001). Tandem acoustic modeling in large-vocabulary recognition. In *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing*, pages 517–520, Salt Lake City, U.S.A.

Fiscus, J. G. (1997). A post-processing system to yield reduced word error rates: Recognizer Ouput Voting Error Reduction (ROVER). In *IEEE-ASRU*, pages 347–354, Santa Barbara, C.A.

Fletcher, H. (1953). *Speech and Hearing in Communication*. Robert E. Krieger Publishing Company, Huntington, New York.

Forney, G. D. (1973). The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278.

Fujinaga, K., Nakai, M., Shimodaira, H., and Sagayama, S. (2001). Multiple-regression hidden Markov model. In *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing*, pages 513–516, Salt Lake City, Utah, U.S.A.

Furui, S. (1981). Cepstral analysis technique for automatic speaker verification. *IEEE Trans. Acoust., Speech, Signal Processing*, 29(2):254–272.

Furui, S. (1986). Speaker-independent isolated word recognition using dynamic features of speech spectrum. pages 52–59.

Glotin, H. (2000). *Élaboration et comparaison de systèmes adaptatifs multi-flux de reconnaissance de la parole: Incorporation des indices d'harmonicité et de localisation*. PhD dissertation, Institut National Polytechnique de Grenoble, Grenoble, France.

Glotin, H. and Berthommier, F. (2000). Test of several external posterior weighting functions for multi-band full combination ASR. In *Proceedings of International Conference on Spoken Language Processing*, Beijing-China.

Glotin, H., Berthommier, F., and Tesier, E. (1999). A casa-labelling model using the localisation cue for robust cocktail-party speech recognition. In *Proceedings of European Conference on Speech Communication and Technology*.

Gong, Y. (1995). Speech recognition in noisy environments: A survey. *Speech Communication*, 16(3):261–291.

Greenberg, S. and Kingsbury, B. E. D. (1997). The modulation spectrogram: In pursuit of an invariant representation of speech. In *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing*, pages 1647–1650.

Hagen, A. (2001). *Robust speech recognition based on multi-stream processing*. PhD dissertation, École Polytechnique Fédérale de Lausanne, Départment d'Informatique, EPFL, Lausanne, Switzerland.

Hagen, A. and Bourlard, H. (2000). Using multiple time scales in the framework of multi-stream speech recognition. In *Proceedings of International Conference on Spoken Language Processing*, volume 1, pages 349–352, Beijing, China.

Hagen, A. and Morris, A. (2005). Recent advances in the multi-stream HMM/ANN hybrid approach to noise robust ASR. *Computer, Speech and Language*, (19):3–30.

Hagen, A. and Morris, A. C. (2000). Comparison of HMM experts with MLP experts in the full combination multi-band approach to robust ASR. In *Proceedings of International Conference on Spoken Language Processing*, Beijing, China.

Hagen, A., Morris, A. C., and Bourlard, H. (2000). From multi-band full combination approach to multi-stream full combination processing in robust ASR. In *ISCA ITRW Workshop on Automatic Speech Recognition: Challenges for the new Millennium (ASR2000)*, pages 175–180, Paris, France.

Heckmann, M., Berthommier, F., and Kroschel, K. (2001). Optimal weighting of posteriors for audio-visual speech recognition. In *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing*, volume 1, pages 161–164.

Heckmann, M., Berthommier, F., and Kroschel, K. (2002). Noise adaptive stream weighting in audio-visual speech recognition. *Journal on Applied Signal Processing (special issue on Audio-Visual Processing)*, 2(11):1260–1273.

Hermansky, H. (1990). Perceptual linear predictive (PLP) analysis of speech. *J. Acoust. Soc. Amer.*, 87(4):1738–1752.

Hermansky, H., Ellis, D. P. W., and Sharma, S. (2000). TANDEM connectionist feature extraction for conventional HMM systems. In *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing*, pages 1635–1638, Istanbul, Turkey.

Hermansky, H. and Morgan, N. (1994). RASTA processing of speech. *IEEE Trans. Speech, Audio Processing*, 2(4):578–589.

Hermansky, H. and Sharma, S. (1998). TRAPs: Classifiers of temporal patterns. In *Proceedings of International Conference on Spoken Language Processing*, pages 1003–1006, Sydney, Australia.

Hermansky, H., Sharma, S., and Jain, P. (1999). Data-drived non-linear mapping for feature extraction in HMM. Keystone, CO.

Hermansky, H., Tibrewala, S., and Pavel, M. (1996). Towards ASR on partially corrupted speech. In *Proceedings of International Conference on Spoken Language Processing*, volume 1, pages 462–465, Philadephia, U.S.A.

Hirsch, H.-G. and Pearce, D. (2000). The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions. In *ISCA ITRW Workshop on*

Automatic Speech Recognition: Challenges for the new Millennium (ASR2000), pages 181–188, Paris, France.

Huang, L. and Yang, C. (2000). A novel approach to robust speech endpoint detection in car environments. In *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing*, Istanbul, Turkey.

Ikbal, S. (2004). *Nonlinear feature transformations for noise robust speech recognitio*. PhD thesis, École Polytechnique Fédérale de Lausanne.

Ikbal, S., Doss, M. M., Misra, H., and Bourlard, H. (2004a). Spectro-temporal activity pattern (STAP) features for noise robust ASR. In *Proceedings of International Conference on Spoken Language Processing*, Jeju Island, South Korea.

Ikbal, S., Hermansky, H., and Bourlard, H. (2003a). Nonlinear spectral transformations for robust speech recognition. In *IEEE-ASRU*, St. Thomas, Virgin Islands, U.S.A.

Ikbal, S., Misra, H., and Bourlard, H. (2003b). Phase autocorrelation (PAC) derived robust speech features. In *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing*, Hong Kong.

Ikbal, S., Misra, H., Bourlard, H., and Hermansky, H. (2004b). Phase autocorrelation (PAC) features in entropy based multi-stream for robust speech recognition. In *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing*, Montreal, Canada.

Ikbal, S., Misra, H., Sivadas, S., Hermansky, H., and Bourlard, H. (2004c). Entropy based combination of Tandem representations for noise robust ASR. In *Proceedings of International Conference on Spoken Language Processing*, Jeju Island, South Korea.

Jain, P. and Hermansky, H. (2001). Improved mean and variance normalization for robust speech recognition. In *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing*.

Janin, A., Ellis, D., and Morgan, N. (1999). Multi-stream speech recognition: Ready for prime time? In *Proceedings of European Conference on Speech Communication and Technology*, pages 591–594.

Jelinek, F. (1976). Continuous speech recognition by statistical methods. *Proc. IEEE*, 64(4):532–556.

Katz, S. M. (1987). Estimation of probabilities from sparse data for the language model component of a speech recognizer. 35(3):400–401.

Ketabdar, H., Bourlard, H., and Bengio, S. (2005). Hierarchical multi-stream posterior based speech secognition system. In *Proceedings MLMI workshop*, Edinburgh, U.K.

Kirchhoff, K. (1998). Combining articulatory and acoustic information for speech recognition in noisy and reverberation environments.

Kirchhoff, K. and Bilmes, J. A. (2000). Combination and joint training of acoustic classifiers for speech recognition. In *ISCA ITRW Workshop on Automatic Speech Recognition: Challenges for the new Millennium (ASR2000)*, Paris, France.

Kirchhoff, K., Fink, G. A., and Sagerer, G. (2000). Conversational speech recognition using acoustic and articulatory input. In *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing*, Istanbul, Turkey.

Kittler, J., Hatef, M., Duin, R. P. W., and Matas, J. (1998). On combining classifiers. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(3).

Kock, W. E. (1950). Binaural localization and masking. *J. Acoust. Soc. Amer.*, 22(6):801–804.

Koenig, W. (1950). Subjective effects in binaural hearing. *J. Acoust. Soc. Amer.*, 22(1):61–62.

Konig, Y., Mogan, N., and Chandra, C. (1991). GDNN: A gender-dependent neural network for continuous speech recognition. TR 91-071, ICSI, Berkeley, California, U.S.A.

Kuncheva, L. I. (2002). A theoretical study on six classifier fusion strategies. *IEEE Trans. Pattern Analysis and Machine Intelligence*, (2):281–286.

Kuncheva, L. I. (2005). Using diversity measures for generating error-correcting output codes in classifier ensembles. *Pattern Recognition Letters*, 26:83–90.

Lee, L. and Rose, R. C. (1996). Speaker normalization using efficient frequency warping procedures. In *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing*, pages 353–356, Atlanta, Georgia, U.S.A.

Lee, L. and Rose, R. C. (1998). A frequency warping approach to speaker normalization. *IEEE Trans. Speech, Audio Processing*, 6:49–60.

Leggetter, C. J. and Woodland, P. C. (1995). Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. *Computer, Speech and Language*, 9:171–185.

Lei, X., Hwang, M., and Ostendorf, M. (2005). Incorporating tone-related MLP posteriors in the feature representation for Mandarin ASR. In *Proceedings of European Conference on Speech Communication and Technology*, Lisbon, Portugal.

Lippmann, R. P. (1987). An introduction to computing with neural nets. *IEEE ASSP Magazine*, pages 4–22.

Lockwood, P. and Boudy, J. (1992). Experiments with a non-linear spectral subtractor (nss), hidden Markov model and the projection, for robust speech recognition in cars. *Speech Communication*, 11(2–3):215–228.

Lucassen, J. M. and Mercer, R. L. (1984). An information theoretic approach to automatic determination of phonemic baseforms. In *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing*, pages 42.5.1–42.5.4, San Diego, California, USA.

Makhoul, J. (1975). Linear prediction: A tutorial review. *Proc. IEEE*, 63:561–580.

Markel, J. D. (1972). The SIFT algorithm for fundamental frequency estimation. *IEEE Transcations on Audio Electroacoustics*, 20:367–377.

McCandless, S. S. (1974). An algorithm for automatic formant extraction using linear prediction spectra. *IEEE Trans. Acoust., Speech, Signal Processing*, pages 135–141.

McClellan, S. and Gibson, J. D. (1997). Variable-rate CELP based on subband flatness. *IEEE Trans. Speech, Audio Processing*, 5(2):120–130.

McClelland, T. L., Rumelhat, D. E., and the PDP Research Group (1986). *Parallel Distributed Processing*. MIT Press, Cambridge, MA.

McGurk, H. and McDonald, J. (1976). Hearing lips and seeing voices. *Nature*, 276:746–748.

Mirghafori, N. (1999). *A multi-band approach to automatic speech recognition*. PhD dissertation, The International Computer Science Institute (ICSI), Berkeley, California.

Mirghafori, N. and Morgan, N. (1997). Transmissions and transitions: A study of two common assumptions in multi-band ASR. In *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing*, Munich, Germany.

Misra, H. and Bourlard, H. (2005). Spectral entropy feature in full-combination multi-stream system for robust ASR. In *Proceedings of European Conference on Speech Communication and Technology*, Lisbon, Portugal.

Misra, H., Bourlard, H., and Tyagi, V. (2002). Entropy-based multi-stream combination. IDIAP-RR 31, IDIAP, Martigny, Switzerland.

Misra, H., Bourlard, H., and Tyagi, V. (2003). New entropy based combination rules in HMM/ANN multi-stream ASR. In *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing*, Hong Kong.

Misra, H., Ikbal, S., Bourlard, H., and Hermansky, H. (2004). Spectral entropy based feature for robust ASR. In *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing*, Montreal, Canada.

Misra, H., Ikbal, S., Sivadas, S., and Bourlard, H. (2005a). Multi-resolution spectral entropy feature for robust ASR. In *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing*, Philadelphia, U.S.A.

Misra, H., Vepa, J., and Bourlard, H. (2005b). Multi-stream ASR: Oracle test and embedded training. IDIAP-RR 62, IDIAP, Martigny, Switzerland.

Molau, S., Hilger, F., and Ney, H. (2003). Feature space normalization in adverse acoustic conditions. In *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing*, Hong Kong.

Molau, S., Pitz, M., and Ney, H. (2001). Histogram based normalization in the acoustic feature space. In *IEEE-ASRU*.

Moore, B. C. J. (1997). *An Introduction to the Psychology of Hearing*. Academic Press, New York.

Morgan, N., Bourlard, H., and Hermansky, H. (1998). Automatic speech recognition: An auditory perspective. IDIAP-RR 17, IDIAP, Martigny, Switzerland.

Morgan, N., Chen, B. Y., Zhu, Q., and Stolckle, A. (2004). Trapping conversational speech: Extending trap/tandem approaches to conversational telephone speech recognition. In *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing*, Montreal, Canada.

Morris, A. C., Hagen, A., Glotin, H., and Bourlard, H. (2001). Multi-stream adaptive evidence combination for noise robust ASR. *Speech Communication*, 34(1–2):25–40.

Neti, C., Potamianos, G., Luettin, J., Matthews, I., Glotin, H., and Vergyri, D. (2001). Large-vocabulary audio-visual speech recognition: A summary of the Johns Hopkins summer 2000 workshop. In *Proceedings of IEEE Signal Processing Society 2001 Workshop on Multimedia Signal Processing*, pages 619–624.

Niyogi, P. and Sondhi, M. M. (2002). Detecting stop consonants in continuous speech. *J. Acoust. Soc. Amer.*, 111(2):1063–1076.

Okawa, S., Bocchieri, E., and Potamianos, A. (1998). Multi-band speech recognition in noisy environments. In *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing*, pages 641–644, Seattle, Washington.

Okawa, S., Nakajima, T., and Shirai, K. (1999). A recombination strategy for multi-band speech recognition based on mutual information criterion. In *Proceedings of European Conference on Speech Communication and Technology*, pages 603–606, Budapest, Hungary.

Oppenheim, A. V. and Schafer, R. W. (1975). *Digital Signal Processing*. Prentice Hall, Englewood Cliffs, New Jersey.

Padmanabhan, M. (2000). Spectral peak tracking and its use in speech recognition. In *Proceedings of International Conference on Spoken Language Processing*, Beijing, China.

Papoulis, A. (1991). *Probability, Random Variables, and Stochastic Processes*. McGraw–Hill, New York, third edition.

Pickles, J. O. (1998). *An Introduction to the Physiology of Hearing*. Academic Press, New York.

Poh, N. and Bengio, S. (2005). How do correlation and variance of base-experts affect fusion in biometric authentication tasks? *IEEE Trans. Signal Processing*.

Rabiner, L. and Juang, B.-H. (1993). *Fundamentals of Speech Recognition*. Prentice Hall, Englewood Cliffs, New Jersey.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Renals, S., Morgan, N., Bourlard, H., Cohen, M., and Franco, H. (1994). Connectionists probability estimators in HMM speech recognition. *IEEE Trans. Speech, Audio Processing*, 2(1):161–174.

Richard, M. D. and Lippmann, R. P. (1991). Neural network classifiers estimate Bayesian a-posteriori probabilities. *Neural Computation*, 4(3):461–483.

Ris, C. and Dupont, S. (2001). Assessing local noise level estimation methods: Application to noise robust ASR. 34(1–2):141–158.

Roark, B., Saraclar, M., and Collins, M. (1994). Corrective language modelling for large vocabulary ASR with the perceptron algorithm. In *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing*.

Robinson, A. J., Cook, G. D., Ellis, D. P. W., Fosler-Lussier, E., Renals, S. J., and Williams, D. A. G. (2002). Connectionist speech recognition of Broadcast News. *Speech Communication*, 37:27–45.

Robinson, T., Hochberg, M., and Renals, S. (1996). *Automatic Speech and Speaker Recognition - Advanced Topics*. Kluwer Academic Publishers.

Rogozan, A. and Deléglise, P. (1998). Adaptive fusion of acoustic and visual sources for automatic speech recognition. *Speech Communication*, pages 149–161.

Schwenk, H. and Gauvain, J.-L. (2000). Improved rover using language model information. In *ISCA ITRW Workshop on Automatic Speech Recognition: Challenges for the new Millennium (ASR2000)*, pages 47–52, Paris, France.

Shannon, C. E. and Weaver, W. (1949). *The Mathematical Theory of Communication*. University of Illinois Press, Urbana.

Sharma, S. (1999). *Multi-stream approach to robust speech recognition*. PhD thesis, Oregon Graduate School of Science and Technology, Oregon, Portland, U.S.A.

Sharma, S., Ellis, D. P. W., Kajarekar, S., Jain, P., and Hermansky, H. (2000). Feature extraction using non-linear transformation for robust speech recognition on the Aurora database. In *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing*, volume 2, pages 1117–1120, Istanbul, Turkey.

Shen, J., Hung, J., and Lee, L. (1998). Robust entropy-based endpoint detection for speech recognition in noisy environments. In *Proceedings of International Conference on Spoken Language Processing*, Sydney, Australia.

Shire, M. L. (2001). Multi-stream ASR trained with heterogeneous reverberant envirnoments. In *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing*, Salt Lake City, U.S.A.

Shire, M. L. and Chen, B. Y. (2000). Data-driven RASTA filters in reverberation. In *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing*, Istanbul, Turkey.

Sivadas, S. and Hermansky, H. (2004). On use of task independent training data in tandem feature extraction. In *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing*, Montreal, Canada.

Strope, B. and Alwan, A. (1998). Robust word recognition using threaded spectral peaks. In *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing*, pages 625–628.

Subramanya, A., Bilmes, J., and Chen, C.-P. (2005). Focused word segmentation for ASR. In *Proceedings of European Conference on Speech Communication and Technology*, Lisbon, Portugal.

Teissier, P., Robert-Ribes, J., Schwartz, J.-L., and Guérin-Dugué, A. (1999). Comparing models for audiovisual fusion in a noisy-vowel recognition task. *IEEE Trans. Speech, Audio Processing*, 7:629–642.

Tibrewala, S. and Hermansky, H. (1997). Sub-band based recognition of noisy speech. In *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing*, pages 1255–1258, Munich, Germany.

Tomlinson, M. J., Russell, M. J., and Brooke, N. M. (1996). Integrating audio and visual information to provide highly robust speech recognition. In *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing*, pages 821–824.

Tyagi, V., McCowan, I., Bourlard, H., and Misra, H. (2003). On factorizing spectral dynamics for robust speech recognition. In *Proceedings of European Conference on Speech Communication and Technology*, Geneva, Switzerland.

Varga, A., Steeneken, H., Tomlinson, M., and Jones, D. (1992). The NOISEX-92 study on the affect of additive noise on automatic speech recognition. Technical report, DRA Speech Research Unit, Malvern, England.

Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. 13(2):260–267.

Wakita, H. (1977). Normalization of vowels by vocal-tract length and its application to vowel identification. *IEEE Trans. Acoust., Speech, Signal Processing*, 25:183–192.

Weber, K., Ikbal, S., Bengio, S., and Bourlard, H. (2003). Robust speech recognition and feature extraction using HMM2. *Computer, Speech and Language*, 17(2-3):195–221.

Welling, L. and Ney, H. (1998). Formant estimation for speech recognition. *IEEE Trans. Speech, Audio Processing*, 6(1):36–48.

Wu, S., Kingsbury, B., Morgan, N., and Greenberg, S. (1998a). Incorporating information from syllable-length time scales into automatic speech recognition. In *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing*, pages 721–724.

Wu, S.-L., Kingsbury, B., Morgan, N., and Greenberg, S. (1998b). Performance improvements through combining phone- and syllable-scale information in automatic speech recognition. In *Proceedings of International Conference on Spoken Language Processing*, pages 459–462.

Yang, C., Soong, F. K., and Lee, T. (2005). Static and dynamic spectral features: Their noise robustness and optimal weights for ASR. In *Proceedings of European Conference on Speech Communication and Technology*, Philadelphia, U.S.A.

Yegnanarayana, B. (1999). *Artificial Neural Networks*. Prentice–Hall India, Connaught Circus, New Delhi.

Young, S., Odell, J., Ollason, D., Valtchev, V., and Woodland, P. (1997). *HTK: Hidden Markov Model Toolkit V2.1*. Entropic Cambridge Research Laboratory, U.K.

Zhu, Q., Chen, B., Grezl, F., and Morgan, N. (2005a). Improved MLP structures for data-driven feature extraction for ASR. In *Proceedings of European Conference on Speech Communication and Technology*, Lisbon, Portugal.

Zhu, Q., Chen, B., Morgan, N., and Stolcke, A. (2004). On using MLP features in LVCSR. In *Proceedings of International Conference on Spoken Language Processing*, Jeju Island, South Korea.

Zhu, Q., Stolcke, A., Chen, B. Y., and Morgan, N. (2005b). Using MLP features in SRI's conversational speech recognition system. In *Proceedings of European Conference on Speech Communication and Technology*, Lisbon, Portugal.

Zweig, G., Bilmes, J., Richardson, T., Filali, K., Livescu, K., Xu, P., Jackson, K., Brandman, Y., Sandness, E., Holtz, E., Torres, J., and Byrne, B. (2002). Structurally discriminative graphical models for automatic speech recognition: Results from the 2001 Johns Hopkins summer workshop. In *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing*, pages I:93–96, Orlando, Florida, USA.

Curriculum Vitae

Personal Profile:

Name:	Hemant Misra
Nationality:	Indian
Present Occupation:	Pursuing Ph.D. at IDIAP, Martigny, Switzerland Swiss Federal Institute of Technology, Lausanne, Switzerland
Last Degree:	Master of Science (by Research), Electrical Engineering Department, Indian Institute of Technology, Madras, India
Contact Address:	1, Rue des Finettes, 1920 Martigny, Switzerland. Phone: +41 27 722 74 81 (H) +41 27 721 77 57 (O) +41 77 406 64 71 (M)
Permanent Address:	C - 27, Sector I, Aliganj Colony, Lucknow - 226 020, India. Phone: +91 522 233 23 14
E-mail:	misra@idiap.ch hemant_misra@yahoo.com
Web-page:	http://www.idiap.ch/~misra

Educational Qualifications:

Exam	Year	Institute	%/CGPA
Ph.D. School of Engineering	2006*	EPFL, Lausanne	- -
Master of Science (by Research) Electrical Engineering	2000	IIT Madras	9.83/10
Bachelor of Engineering Electronics and Communication	1995	SVNIT Surat (former REC, Surat)	72.00
A.I.S.S.C.E.	1989	St. Gabriels Academy Roorkee	74.60
High School	1987	Holy Cross H.S.S., Raipur	69.23

* March, 2006.

Ph.D. Details: Since August 2001, I am doing my Ph.D. on speech recognition at IDIAP Research Institute, Switzerland and Prof. Hervé Bourlard is my thesis supervisor. IDIAP is affiliated with Swiss Federal Institute of Technology, Lausanne, Switzerland (EPFL). My interest area is robustness issues in speech recognition and specifically I am working on multi-stream combination to address this issue. We have investigated few weighting techniques and features that helped us in making the ASR system more robust to different kinds of noises at various SNRs. We studied the relationship between entropy at the output of an MLP and its accuracy. Based on the inverse linear relationship between the two, we suggested inverse entropy weighting methods which reduce the entropy and improve the ASR performance. Further, we suggested spectral entropy features which are obtained from the spectrum. The new feature tries to capture the peaks of the spectrum and is found to be robust to wide-band noises. Subsequently, we applied the two techniques discussed above to Tandem system which is a combination of hybrid HMM/ANN and standard HMM/GMM systems. The performance of multi-stream Tandem system thus obtained has a significant advantage over the best baseline systems under clean as well as various noises conditions investigated in the work.

We worked on two kinds of databases, Numbers95 connected digit database and DARPA large vocabulary conversational telephone speech database. The above mentioned techniques give an improvement for both the databases and improvement is more significant for high noise cases. While investigating these methods, I worked on *Respite* (http://www.dcs.shef.ac.uk/research/groups/spandh/projects/respite/), *DARPA EARS* (http://www.darpa.mil/ipto/programs/ears/) and *IM2.SP* (http://www.im2.ch/).

During the first two years of my Ph.D., I was a *teaching assistant* (TA) for the lab of *speech processing* taken by Prof. Hervé Bourlard at EPFL. Also, I am a reviewer for *IEEE Transactions on Speech and Audio Processing* and *IEEE Signal Processing Letters*.

Master of Science (M.S.) Details: I did my M.S. under the guidance of *Prof. B. Yegnanarayana* (Computer Science and Engineering Department, IIT Madras) and *Prof. K. R. K. Rao* (Electrical Engineering Department, IIT Madras). I worked on the problem of *automatic speaker recognition*. We developed a *mapping approach* using artificial neural networks to capture speaker specific information. We obtained an *equal error rate* (EER) of 5.45% on a subset of *NTIMIT* database and 0.45% on all the 630 speakers of *TIMIT* database.

Apart from the research work, I pursued five courses to help me in my research work. The courses were: Digital Filter Design, Architectures of Artificial Neural Networks and their Applications, Speech Processing, Communication Theory and Computer Organization. During the M.S. program, I was also a TA for the bachelor courses of *DSP*, *communication systems* and *speech processing*.

Bachelor of Engineering (B.E.) Details: The B.E. program at SVNIT Surat (earlier known as REC Surat) has a four year curriculum. Following are few of the important subjects I studied during my four years at SVNIT Surat: Electronics, Communication, Analog and Digital Integrated Circuits, Microprocessors, Power Electronics, Instrumentation, Electrical Machines, Control Theory, Power Systems, and Electrical Networks. As part of the curriculum, I worked on two projects in the final year (i) I gave a seminar on *Image Processing* and implemented a software for image enhancement using histogram equalization technique. (ii) My team of 4 students developed a software tool for designing digital filters. Our project was awarded the highest rank in the class.

Work Experience:

1. *Organization:* Speech and Software Technologies (India) Private Limited, Chennai (www.sstil.com)
   Designation: Software Analyst
   Work Profile: I developed a speaker recognition system based on my M.S. work. Additionally, I coordinated with other speech group members in the implementation of its various components (feature extraction, TAPI etc). In the later stages of my work, I worked on a limited vocabulary connected speech recognition system for telephone quality speech. All the systems developed were application oriented and real-time. I was also in-charge of evaluating the 'C' coding standards followed in our company.
   Duration: August 1999 - July 2001

2. *Organization:* Motorola India Electronics Limited, Bangalore (www.motorola.com/in/)
   Designation: Software Engineer
   Work Profile: I implemented and ported the existing DSP codes written for one processor to different processors for our off-shore clients.
   Duration: May 1999 - July 1999

3. *Organization:* Petrofils Co-operative Limited, Ankleshwar, India
   Designation: Graduate Engineer Trainee
   Work Profile: Maintenance of electronic and mechanical instruments in the polymerization unit of the plant.
   Duration: September 1995 - June 1996

Computer Skills:

Programming Languages: C and Visual C
Script Languages: Tcl/Tk
Software Tools: HTK, Matlab and CSLU
Packages for Documentation: Latex and MS-Word
Operating Systems: Unix, Linux and Windows 95/NT/XP

Professional Affiliations

1. A reviewer for *IEEE Transactions on Speech and Audio Processing* and *IEEE Signal Processing Letters*.

2. Student member of Institute of Electrical & Electronic Engineers (IEEE) and International Speech Communication Association (ISCA).

Other Education Related Performances:

1. Scored 88% in *Mathematics Olympiad*, an event organized at all India level (1989).

2. Scored 97.20 percentile in *Graduate Aptitude Test in Engineering* (GATE) in Electronics and Communication paper (1996).

3. Selected for Ph.D. program in Computer Science and Communication Group of *Tata Institute of Fundamental Research, Mumbai* (1996).

4. Obtained a score of 263 out 300 in *Computer Based TOEFL* (2001).

Extra Curricular Activities:

1. Coordinator *MINDBEND'95*, an All India level technical symposium for undergraduates organized at SVNIT Surat (former REC Surat) (1995).
2. Chairman *Indian Society for Technical Education*, Student Chapter, REC Surat (1994-1995).
3. Research Affairs Secretary, Indian Institute of Technology, Madras (1998-1999).
4. A good sportsman, with athletics and football as special interests, mainly for recreational purposes.

Publications:

On Automatic Speech Recognition

[1]. Hemant Misra, and Herve Bourlard, "Spectral entropy feature in full-combination multi-stream for robust ASR," in *Proceedings of Eurospeech*, Lisbon, Portugal, 2005.
[2]. Hemant Misra, Shajith Ikbal, Sunil Sivadas, and Herve Bourlard, "Multi-resolution spectral entropy based feature for robust ASR," in *Proceedings of ICASSP*, Philadelphia, U.S.A., 2005.
[3]. Shajith Ikbal, Hemant Misra, Sunil Sivadas, Hynek Hermansky, and Herve Bourlard, "Entropy based combination of TANDEM representations for noise robust ASR," in *Proceedings of ICSLP*, Jeju Island, South Korea, 2004.
[4]. Shajith Ikbal, Mathew Magimai Doss, Hemant Misra, and Herve Bourlard, "Spectro-Temporal Activity Pattern (STAP) features for noise robust ASR," in *Proceedings of ICSLP*, Jeju Island, South Korea, 2004.
[5]. Hemant Misra, Shajith Ikbal, Herve Bourlard, and Hynek Hermansky, "Spectral entropy based feature for robust ASR," in *Proceedings of ICASSP*, Montreal, Canada, 2004.
[6]. Shajith Ikbal, Hemant Misra, Herve Bourlard, and Hynek Hermansky, "Phase autocorrelation (PAC) features in entropy based multi-stream for robust speech recognition," in *Proceedings of ICASSP*, Montreal, Canada, 2004.
[7]. Vivek Tyagi, Iain McCowan, Herve Bourlard, and Hemant Misra, "Mel-Cepstrum Modulation Spectrum (MCMS) features for robust ASR," in *Proceedings of IEEE ASRU*, St. Thomas, Virgin Islands, U.S.A, 2003.
[8]. Hemant Misra, and Andrew C. Morris, "Confusion matrix based entropy correction in multi-stream combination," in *Proceedings of Eurospeech*, Geneva, Switzerland, 2003.
[9]. Vivek Tyagi, Iain McCowan, Herve Bourlard, and Hemant Misra, "On factorizing spectral dynamics for robust speech recognition," in *Proceedings of Eurospeech*, Geneva, Switzerland, 2003.
[10]. Hemant Misra, Herve Bourlard, and Vivek Tyagi, "New entropy based combination rules in HMM/ANN multi-stream ASR," in *Proceedings of ICASSP*, Hong Kong, 2003.
[11]. Shajith Ikbal, Hemant Misra, and Herve Bourlard, "Phase autocorrelation (PAC) derived robust speech features," in *Proceedings of ICASSP*, Hong Kong, 2003.

On Automatic Speaker Recognition

[1]. Hemant Misra, Shajith Ikbal, B. Yegnanarayana, "Speaker-specific mapping for text-independent speaker recognition," *Speech Communication*, vol. 39, no. 3-4, pp. 301-310, February 2003.
[2]. M. S. Ikbal, H. Misra and B. Yegnanarayana, "Analysis of autoassociative mapping neural networks," in *Proceedings of IJCNN*, Washington, USA, July 12-16, 1999.
[3]. Hemant Misra, Shajith Ikbal and B. Yegnanarayana, "Spectral mapping as a feature for speaker recognition," in *Proceedings of National Conference on Communication*, Kharagpur, India, pp. 151-156, January 29-31, 1999.

Unpublished Reports

[1]. Hemant Misra, Jithendra Vepa and Hervé Bourlard, "Multi-stream ASR: Oracle Test and Embedded Training", IDIAP-RR 05-62, 2005.
[2]. Hemant Misra and Hervé Bourlard, "Spectral Entropy Feature in Multi-Stream for Robust ASR," IDIAP-RR 05-45, 2005.