

# PHYSIOLOGICALLY MOTIVATED METHODS FOR AUDIO PATTERN CLASSIFICATION

A Dissertation  
Presented to  
The Academic Faculty

By

Sourabh Ravindran

In Partial Fulfillment  
of the Requirements for the Degree  
Doctor of Philosophy in Electrical Engineering



School of Electrical and Computer Engineering  
Georgia Institute of Technology  
December 2006

Copyright © 2006 by Sourabh Ravindran

# PHYSIOLOGICALLY MOTIVATED METHODS FOR AUDIO PATTERN CLASSIFICATION

Approved by:

Dr. Chin-Hui Lee, Committee Chair  
*Professor, School of ECE*  
*Georgia Institute of Technology*

Dr. James M. Rehg  
*Professor, College of Computing*  
*Georgia Institute of Technology*

Dr. David V. Anderson, Advisor  
*Professor, School of ECE*  
*Georgia Institute of Technology*

Dr. Paul E. Hasler  
*Professor, School of ECE*  
*Georgia Institute of Technology*

Dr. Yucel Altunbasak  
*Professor, School of ECE*  
*Georgia Institute of Technology*

Date Approved: October 31, 2006

## DEDICATION

*To my parents and to Parag,  
for their support, faith, and selfless love*

## ACKNOWLEDGMENT

First and foremost, I would like to thank my parents for everything they have done for me. My gratitude for their kindness and love cannot be expressed in words. My journey as a graduate student would not have begun without the constant encouragement and inspiration from my brother, Dr. Parag Ravindran. He has often been the calming influence during the frustrations of missed deadlines and failed experiments. I am forever indebted to him.

I would like to express my deepest thanks to my thesis Advisor, Prof. David Anderson, for his guidance, patience, and support. His wonderful ability to balance guidance and exploratory learning has made this journey a valuable experience. I would also like to express my gratitude to my thesis committee members, Dr. Chin-Hui Lee, Dr. Paul Hasler, Dr. James Rehg and Dr. Yucel Altunbasak for their useful comments, suggestions, and readiness to help every time I approached them. I would like to express my heartfelt gratitude to Dr. Malcolm Slaney, for his guidance and advice. His ability to catch glitches in papers and his drive for improving a paper have been the cause of many sleepless nights, but also a wonderful learning experience that I would not trade for anything.

The work environment often shapes a person and I would like to acknowledge the contribution of the excellent research and social atmosphere of the *ESP* lab. Time spent with my colleague and good friend, Dr. Tyson Hall, has been educational and enriching. I would like to thank Sunil and Sanmati Kamath for their counsel and for their friendship. I would also like to thank the other research group members for all their support and companionship, they made my stay at Georgia Tech an enjoyable one.

I would like to thank the many faculty members who have impacted my life during my undergraduate and graduate studies. In particular, I would like to thank Prof. Narendra, for his ability to inspire students to dream the impossible. His love for signal processing is infectious. Last but certainly not the least, I would like to thank Pam Halverson and Janet Myrick for their great administrative support.

# TABLE OF CONTENTS

<b>DEDICATION</b> . . . . .	iii
<b>ACKNOWLEDGMENT</b> . . . . .	iv
<b>LIST OF TABLES</b> . . . . .	vii
<b>LIST OF FIGURES</b> . . . . .	x
<b>SUMMARY</b> . . . . .	xv
<b>CHAPTER 1 INTRODUCTION</b> . . . . .	1
1.1 Background . . . . .	3
1.1.1 Early Auditory System . . . . .	3
1.1.2 Mathematical Model of the Auditory System . . . . .	5
1.2 Review of Filter-Bank Features for Speech Recognition . . . . .	7
1.3 Review of Previous Audio Classification work . . . . .	10
1.4 Contributions of this Research . . . . .	12
<b>CHAPTER 2 IMPROVING NOISE ROBUSTNESS OF PRIMARY FEATURES</b> . . . . .	15
2.1 Issues with Mel-Frequency Cepstral Coefficients . . . . .	15
2.2 Noise-Robust Auditory Features (NRAF) . . . . .	17
2.2.1 Motivation for using BPF . . . . .	18
2.2.2 Noise Robustness of NRAF . . . . .	20
2.2.3 Evaluation of Noise Robustness of NRAF features . . . . .	21
2.2.4 Information-Theoretic Clustering Validity Measure . . . . .	24
2.3 Experimental Performance Comparison of MFCC and NRAF . . . . .	28
2.3.1 Speech Versus Non-Speech Discrimination . . . . .	28
2.3.2 Audio Classification . . . . .	31
2.3.3 Speech Recognition . . . . .	31
2.4 Varying Time Constants in Feature Extraction . . . . .	33
2.5 Gain Adaptation . . . . .	36
2.5.1 Effect of Compression on Noise Robustness . . . . .	37
2.5.2 Adaptive Gain Control . . . . .	41
2.6 Design Notes . . . . .	46
2.7 Summary . . . . .	47
<b>CHAPTER 3 PROCESSING SECONDARY FEATURES</b> . . . . .	49
3.1 AdaBoost . . . . .	50
3.2 Generative AdaBoost . . . . .	57
3.2.1 Boosting Density Estimation . . . . .	58
3.2.2 Minimizing $L_2$ norm . . . . .	60
3.2.3 KL divergence-based Approach . . . . .	62
3.2.4 Experimental Validation . . . . .	63
3.2.5 Results and Discussion . . . . .	63
3.3 Cascade Jump SVMs . . . . .	64

3.4	Dimensionality Reduction Using AdaBoost . . . . .	73
3.5	Design Notes . . . . .	76
3.6	Summary . . . . .	78
<b>CHAPTER 4</b>	<b>APPLICATIONS AND FUTURE WORK . . . . .</b>	<b>80</b>
4.1	Digital Hardware Implementation . . . . .	82
4.1.1	Feature Extraction . . . . .	82
4.1.2	Implementation of the classifier . . . . .	83
4.1.3	Power . . . . .	84
4.2	<i>CADSP</i> Implementation . . . . .	84
4.2.1	Feature Extraction . . . . .	84
4.3	Summary . . . . .	85
4.4	Future Work . . . . .	87
<b>APPENDIX A</b>	<b>ADAPTIVE STANDARDIZATION . . . . .</b>	<b>89</b>
<b>APPENDIX B</b>	<b>DEADLOCK RESOLUTION USING A NORMALIZED MEASURE OF MARGIN . . . . .</b>	<b>94</b>
<b>REFERENCES</b>	<b>. . . . .</b>	<b>95</b>

## LIST OF TABLES

Table 1	Empirical conditional entropy measures for MFCC and NRAF for a 4-class, 4-cluster case. It is seen that NRAF has better class discrimination ability.	30
Table 2	Comparison between root compressed MFCC and NRAF. Since the added noise is white, mean and variance normalization removes most of the noise, making the performance of the two features similar. A 15 mixture GMM and 12 features were used. . . . .	32
Table 3	Comparison between root compressed MFCC and NRAF. Pink noise was synthetically added. A 15 mixture GMM and 12 features were used. . . .	32
Table 4	Table showing that spatial derivative is useful in clean and low noise conditions but in high noise cases spatial derivative can hurt the robustness of the features. A 15 mixture GMM and 12 NRAF features were used. Pink noise was synthetically added. . . . .	32
Table 5	Table showing performance of MFCCs and NRAFs for a four-class audio classification problem. . . . .	33
Table 6	Six Gaussian components per mixture was used for every state, except silence, which was modelled using 12 components. Training was carried out in clean condition. . . . .	33
Table 7	Table showing the significance of the improvements afforded by NRAF features. The improvement is relative to MFCC features. It is seen that at low SNRs there is significant improvement. . . . .	34
Table 8	Six Gaussian components per mixture was used for every state, except silence, which was modelled using 12 components. The entire training and test data was used. The increased modeling ability of the backend enables it to better fit the extra information encoded by the NRAF representation.	36
Table 9	Table showing the significance of the improvements afforded by varying time constants. The improvement is relative to NRAF features. It is seen that at medium SNRs the improvement is significant. . . . .	39
Table 10	Table showing that root compression is better than log compression for noise robustness. A 15 mixture GMM was used and pink noise was synthetically added. The first 12 MFCC features were used. . . . .	39
Table 11	With more compression, between-class distance of the features decrease. .	41
Table 12	Table showing that smaller $\alpha$ yields greater discrimination in clean conditions. However, in noisy conditions larger $\alpha$ yields better class discrimination.	41
Table 13	Table showing improvement in noise robustness of features with gain adaptation. Pink noise was synthetically added to generate different noise conditions. . . . .	46

Table 14	Affect of AGC (with different values of K) on the noise robustness of features. White noise was synthetically added to obtain different SNRs. . . .	46
Table 15	The AdaBoost algorithm. . . . .	52
Table 16	Table showing performance of a single stage “one versus one” AdaBoost classifier and “one versus rest” AdaBoost classifier using SF and NRAF features. 1vs1-b refers to the case where GMM is used to break the deadlock.	55
Table 17	Table showing performance of single stage AdaBoost and cascade AdaBoost using SF and NRAF features. . . . .	55
Table 18	Table showing performance of single stage AdaBoost and cascade AdaBoost using SF, NRAF and STRF features. . . . .	56
Table 19	AdaBoost based algorithm for boosting density estimates, as proposed by Rosset et al. [62]. . . . .	58
Table 20	KL divergence-based approach. . . . .	63
Table 21	Classification between social and office auditory scenes. 13 PCA transformed MFCCs were used. For the RBF-SVM $C = 100$ and $\gamma = 0.09$ . The same parameters were also used for the final stage of CJSVM. The first two stages were linear SVMs. . . . .	72
Table 22	Classification between social and industrial auditory scenes. 13 PCA transformed MFCCs were used. For the RBF-SVM $C = 100$ and $\gamma = 0.4$ . The same parameters were also used for the final stage of CJSVM. The first two stages were linear SVMs. . . . .	77
Table 23	Results using the difference of proportion significance tests for each of the experiments. It is seen that the CJSVM gives a significant improvement over SVM . . . . .	77
Table 24	Dimensionality reduction using AdaBoost . . . . .	78
Table 25	Table showing results for AdaBoost-based classifier using 1 second data segments for training and testing on the <i>Phonak</i> database. Overall accuracy was 87.96%. . . . .	81
Table 26	Table showing results for AdaBoost-based classifier using 30 second data segments (the outputs of the 1 second case were combined by majority voting) on the <i>Phonak</i> database. Overall accuracy was 97.91%. . . . .	82
Table 27	Table showing results for the <i>Phonak</i> database using simulation of CADSP implementation. Overall accuracy was 82.79 %. . . . .	84
Table 28	The mean and variance of the test data is adaptively learned using a Kalman filter. A 4-mixture GMM was used for classification. MFCC features were used and white noise was synthetically added to generate the different SNR conditions. . . . .	93



Table 29	The mean and variance of the test data is adaptively learned using a Kalman filter. A 4-mixture GMM was used for classification. MFCC features were used and pink noise was synthetically added to generate the different SNR conditions. . . . .	93
----------	---	----

## LIST OF FIGURES

Figure 1	Block diagram showing the organization of this thesis and its impact on various stages of the audio processing pathway for classification systems. An audio classification system can be broadly considered as consisting of three stages, feature extraction, data processing and classification algorithms. This thesis contributes to all three of these stages and also presents a practical application (scene recognition for hearing aids) of the techniques developed in this work. . . . .	2
Figure 2	A cross section of the human cochlea. Within the bone are three fluid-filled chambers that are separated by two membranes. The input to the cochlea is in the scala vestibuli, which is connected at the apical end to the scala tympani. Pressure differences between these two chambers leads to movement in the basilar membrane. . . . .	4
Figure 3	Mathematical model of the early auditory system consisting of filtering in the cochlea (analysis stage), conversion of mechanical displacement into electrical activity in the IHC (transduction stage) and the lateral inhibitory network in the cochlear nucleus(reduction stage) [3]. . . . .	6
Figure 4	Schematic of the cortical model. It is proposed in [9] that the response fields of neurons in the primary auditory cortex are arranged along three mutually perpendicular axes. The tonotopic axis, the bandwidth or scale axis and the symmetry or phase axis. . . . .	7
Figure 5	Figure showing the extraction of MFCC. Frequency decomposition is accomplished using the FFT and the critical bands are modeled using triangular filters. The logarithm provides static compression and decorrelation is achieved using the discrete cosine transform (DCT). . . . .	17
Figure 6	Figure showing the speech spectrum using MFCC representation. (a) shows the clean speech spectrum, (b) shows the clean speech spectrum with mean subtraction (c) shows the noisy speech spectrum (d) shows the noisy speech spectrum with mean subtraction. It is clear that even with mean subtraction, noise affects the MFCC representation . . . . .	18
Figure 7	The bandpass filtered version of the input is subjected to a spatial derivative (approximated by a difference operation). The half-wave rectification followed by the smoothing filter is used for envelope detection. AGC represents amplitude compression, which is followed by DCT to decorrelate the signal. . . . .	19

Figure 8	Figure showing the comparison of clean and noisy speech spectrums for the MFCC and NRAF representations. (a) clean speech spectrum using the MFCC representation, (b) clean speech spectrum using the NRAF representation, (c) noisy speech spectrum using the MFCC representation, (d) noisy speech spectrum using the NRAF representation. It is quite evident that NRAF representation is able to retain most of the speech information even in the presence of noise. Babble noise was synthetically added. . . . .	22
Figure 9	Figure showing the comparison of mean subtracted clean and noisy speech spectrums for the MFCC and NRAF representations. (a) clean speech spectrum using the MFCC representation, (b) clean speech spectrum using the NRAF representation, (c) noisy speech spectrum using the MFCC representation, (d) noisy speech spectrum using the NRAF representation. As is evident, with mean subtraction the NRAF feature is able to keep out most of the noise while retaining the speech information while the MFCC representation still suffers from the effects of noise. . . . .	23
Figure 10	Figure showing the per channel SNR of MFCC and NRAF. Input was noisy speech with white noise synthetically added. It can be seen that NRAF yields higher per channel SNR. The mean of the SNR for MFCC representation is 23.94 and the standard deviation is 9, while the mean for the NRAF representation is 29.04 and the standard deviation is 10.4. . .	24
Figure 11	Figure showing effect of spatial derivative. Plots on the left are the original auditory spectrums and those on the right are the auditory spectrums with 4 <sup>th</sup> order BPFs. The plots on top were generated with spatial derivative and those at the bottom did not use spatial derivative. It is clear that using 4 <sup>th</sup> order filters limits the frequency spreading. However, the spatial derivative stage is still useful in clean and high SNR conditions where changes across the spectral profile are enhanced by the difference operation.	25
Figure 12	Comparison of envelopes in a particular channel ( $\approx 200\text{Hz}$ ) for the MFCC and NRAF front-ends a) Speech input at different SNRs (clean, 20 dB, 10 dB, 5 dB and 0 dB) b) Envelopes using the MFCC front-end c) Envelopes using the NRAF front-end. It is seen that, even with addition of a small amount of noise, the MFCC representation is not very smooth. The NRAF representation is able to maintain the spectral peaks even at very low SNRs.	26
Figure 13	Comparison of envelopes in a particular channel ( $\approx 800\text{Hz}$ ) for the MFCC and NRAF front-ends a) Speech input at different SNRs (clean, 20 dB, 10 dB, 5 dB and 0 dB) b) Envelopes using the MFCC front-end c) Envelopes using the NRAF front-end. As in the previous case, the NRAF representation is much more robust to noise compared to the MFCC representation.	27

Figure 14	Comparison of modulation spectrograms of the MFCC and NRAF front-ends a) MFCC-based modulation spectrogram for clean speech b) NRAF-based modulation spectrogram for clean speech c) MFCC-based modulation spectrogram for noisy speech d) NRAF-based modulation spectrogram for noisy speech. As is evident, the NRAF representation is able to mask the noise modulations much better than the MFCC representation. . . . .	28
Figure 15	Figure showing the empirical conditional entropy measures for MFCC and NRAF for a 2-class, 2-cluster case. It is seen that NRAFs cluster better than MFCCs. White noise was synthetically added. . . . .	29
Figure 16	Figure showing the empirical conditional entropy measures for MFCC and NRAF for a 2-class, 2-cluster case. It is seen that for the task considered NRAFs are better features than MFCCs. Pink noise was synthetically added. . . . .	29
Figure 17	Figure showing the variation of the time constants with the center frequency of each channel . . . . .	36
Figure 18	Speech spectrum in clean condition with a) same time constant in each channel b) varying time constants in each channel. Spectrum in noisy conditions with c) same time constant in each channel d) varying time constants in each channel. As is clear, varying time constants helps reduce the effect on noise on the speech spectrum. . . . .	37
Figure 19	Figure showing the comparative performance of MFCC, NRAF, and NRAF-TC for the speech versus non-speech classification task. Different SNRs were obtained by synthetically adding pink noise. Root compression was used for all the features. . . . .	38
Figure 20	Figure showing the performance of MFCC, NRAF, and NRAF-TC on the Aurora 2 task. Six Gaussian mixtures were used for each state and silence was modelled using 12 component mixture. . . . .	38
Figure 21	Figure showing that root compression followed by DCT leads to better compaction of energy. Reconstruction error is plotted as a function of number of coefficients used for the reconstruction. . . . .	40
Figure 22	Figure showing the effect of AGC (with different values of $K$ ) on clean speech. It is seen that $K < 1$ , results in some loss of information in clean conditions, while $K > 1$ , enhances the low energy parts of the signal. . . . .	44
Figure 23	Figure showing the effect of AGC (with different values of $K$ ) on noisy speech. It is seen that $K < 1$ , suppresses the noise in the signal, smaller value of $K$ leads to more suppression. $K > 1$ , on the other hand, tends to amplify the noisy. . . . .	45
Figure 24	Figure showing the concept of AdaBoost. Although the decision function is linear it takes advantage of the fact that the mapping to a suitable hypothesis space makes the data linearly separable (to a large extent). . . . .	51

Figure 25	Figure showing the improvement in performance of single stage AdaBoost-based classifier with the addition of cortical features (STRF). . . . .	56
Figure 26	Plot showing the improvement in performance of the speech music classifier due to boosting. . . . .	64
Figure 27	Plot showing the improvement in performance of the speech noise classifier due to boosting. . . . .	65
Figure 28	Figure showing a SVM classifier. The concept is to find a hyperplane that maximizes the margin between the two classes. . . . .	67
Figure 29	Figure showing the concept of cascade jump SVMs. The easily separable data points are removed before presenting the rest of the data points to the next classifier in the cascade. . . . .	68
Figure 30	Plot showing the linear hyperplane for the first stage. . . . .	71
Figure 31	Plot showing classification of points for first stage. Data points not lying between the hyperplanes are classified as belonging to the positive or negative class. . . . .	72
Figure 32	Plot showing the linear hyperplane for the second stage. . . . .	73
Figure 33	Plot showing classification of points for second stage. Data points not lying between the hyperplanes are classified as belonging to the positive or negative class. . . . .	74
Figure 34	Plot showing classification of points using a modified proximal SVM with sigmoid kernel. . . . .	75
Figure 35	Plot showing classification of points using a modified proximal SVM with polynomial kernel. . . . .	76
Figure 36	(a) Plot showing the performance curves of the classification system after dimensionality reduction by PCA and cAda for data-set 1 (b) shows the same plot for data-set 2 . . . . .	79
Figure 37	Performance of the AdaBoost-based classifier with rounds of boosting. . .	81
Figure 38	Feature extraction process for the hearing-aid front-end as implemented on the <i>c5510</i> fixed point processor. . . . .	83
Figure 39	Block diagram showing the proposed implementation of the feature extraction process on a <i>CADSP</i> platform. A 20 channel implementation consumes about $5.2 \mu\text{W}$ of power for the analog part. The DCT and temporal filtering are performed in the digital domain. . . . .	86
Figure 40	a) Shows the true mean of the test set (—) and the mean learned by the Kalman filter (— * —) b) Shows the true variance of the test set (—) and the variance learned by the Kalman filter (— * —). For the purposes of illustration only 9 features were chosen to do the adaptive standardization	91

Figure 41 a) Shows tracking of the true mean for 3 different features as a function of segments. b) shows tracking of the true variance for 3 different features. Blue indicates the true value and red indicates the learned value. . . . 92

## SUMMARY

Human-like performance by machines in tasks of speech and audio processing has remained an elusive goal. In an attempt to bridge the gap in performance between humans and machines there has been an increased effort to study and model physiological processes. However, the widespread use of biologically inspired features proposed in the past have been hampered mainly by either the lack of robustness across a range of signal-to-noise ratios or the formidable computational costs. It is possible that the biologically inspired features proposed in the past have been unsuccessful because the classifiers that employed them were not well suited to the characteristics of these features. In physiological systems, sensor processing occurs in several stages. It is likely the case that signal features and biological processing techniques evolved together and are complementary or well matched. It is precisely for this reason that modeling the feature extraction processes should go hand in hand with modeling of the processes that use these features. This research presents a front-end feature extraction method for audio signals inspired by the human peripheral auditory system. It is shown that the noise robustness issues of current state-of-the-art features, specifically, mel-frequency cepstral coefficients (MFCCs) can be addressed by paying closer attention to peripheral auditory processing. Features based on modeling processing in the primary auditory cortex have a distinctly different flavor and classifiers such as Gaussian mixture models (GMMs) cannot fully exploit the potential of these features. New developments in the field of machine learning are leveraged to build classifiers to exploit the performance gains afforded by the features based on advanced models of the human auditory system. Further, a classification structure similar to what might be expected in physiological processing is used to demonstrate the clear advantage of incorporating biologically inspired features into mainstream audio processing. The feature extraction and classification system can be efficiently implemented using the low-power cooperative analog-digital signal processing platform. The usefulness of the features are demonstrated for tasks of audio classification, speech versus non-speech discrimination, and speech recognition. The low-power nature of the classification system makes it ideal for use in applications such as hearing aids, hand-held devices, and surveillance through acoustic scene monitoring. It is

clear that biologically inspired features have huge potential with respect to advancing the state-of-the-art in audio signal processing. There is a clear need to address the issue of how best to use these features. This thesis strives to demonstrate the possible advantages to be gained by using biologically inspired features and also suggests ways to incorporate these features into current classification methods, thereby opening the door to exciting research possibilities.



# CHAPTER 1

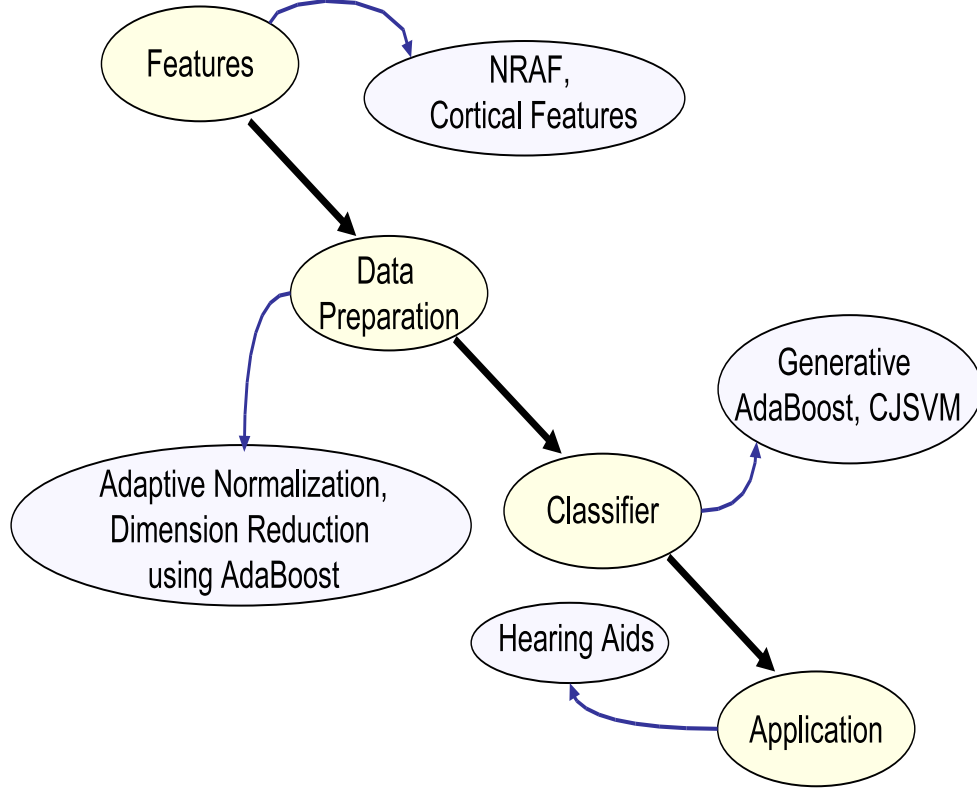
## INTRODUCTION

Audio enabled applications have become ubiquitous, be it voice activated commands for automobiles, voice-based identity verification or audio-centric monitoring and surveillance. Underlying these applications are audio processing techniques such as speech recognition, audio classification, and speaker identification, to name a few. Most audio processing tasks can be considered as consisting of three broad stages, namely, feature extraction, post-processing of data, and back-end classification algorithms. This research touches upon all three of these aspects of the audio processing pathway as shown in Fig. 1.

Humans are much more effective at audio understanding than machines. They can distinguish subtle changes in speech or a variety of other sounds that are difficult to quantify for a computer. Pattern recognition has come a long way, yet the difference in performance between a human and a computer in audio processing tasks is telling. One of the reasons for this performance gap is the feature set used in audio signal processing. In the past researchers have proposed a variety of features based on the human auditory system, however, none of these features have been able to replace mel-frequency cepstral coefficients (MFCCs) as the preferred features for audio processing. The biologically motivated features presented in the past have failed not necessarily because they are poor features but perhaps because they were not well suited to the methods that employed them. Lazzaro et al. [1] cited this “representation-recognizer” gap to be a major hurdle in using physiological motivated features for speech recognition.

Apart from their good performance, MFCCs claim to fame is their efficiency in terms of computation and ease of implementation. The challenge is to improve the performance of MFCCs without significantly adding to the computational overhead. With recent advances in analog VLSI and in low-power implementation of bandpass filters [2], it is perhaps time to revisit physiological processing as a means to improving the performance of MFCCs over a wide range of signal-to-noise ratios (SNRs). Herein new features derived from a model of the early auditory system are presented that outperform MFCCs in tasks of speech recognition and audio classification. These features not only possess superior noise robustness but also

have greater class discrimination ability. The new features can be viewed as physiologically motivated modifications to MFCCs. They share characteristics similar to that of MFCCs and can be used with current popular classification algorithms. In this work, features based on the peripheral auditory system are referred to as “primary” features.



**Figure 1.** Block diagram showing the organization of this thesis and its impact on various stages of the audio processing pathway for classification systems. An audio classification system can be broadly considered as consisting of three stages, feature extraction, data processing and classification algorithms. This thesis contributes to all three of these stages and also presents a practical application (scene recognition for hearing aids) of the techniques developed in this work.

In physiological processing feature extraction is a multi-layered process and in modeling the higher stages of the auditory pathway for feature extraction, new methods of processing these features have to be developed that are capable of working with the sparse nature and high dimensionality of such features. Herein classification algorithms are developed that can work effectively with all kinds of features and not be restricted to a particular class or kind of feature. In particular, spectro-temporal modulation features [3] that are rich but sparse and redundant in terms of information representation are used as an example feature

set to demonstrate the feasibility of the new algorithms. Features based on modeling latter stages of the auditory system are referred to as “secondary” features.

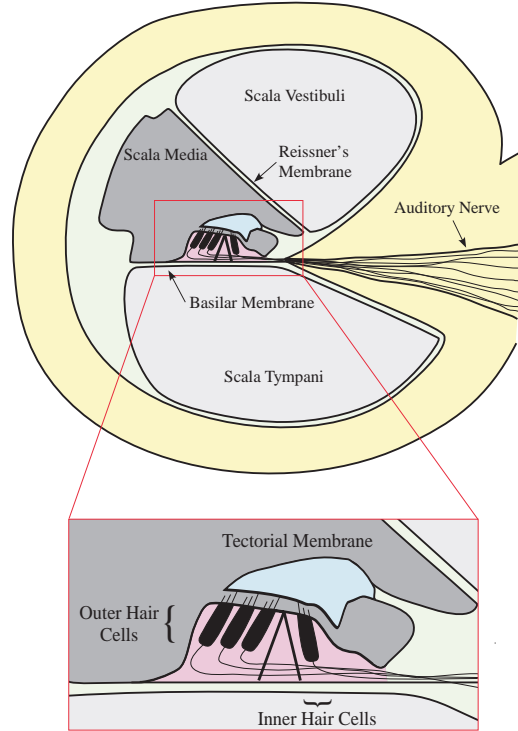
This chapter deals with understanding the functionalities of the human auditory system in hopes of incorporating some of the functionalities into state-of-the-art features to improve their performance in the presence of noise without compromising the performance in clean conditions. The organization of the chapter is as follows, Section 1.1 presents a brief description of the functioning of the human auditory system and presents a mathematical model of the early auditory system as well as a model for the processing in the primary auditory cortex. The model for the early auditory system is used to improve the performance of existing features (explained in further detail in Chapter 2). The cortical model has been used to extract features that are shown to be very robust to noise [4]. Unfortunately these features exist in a high dimensional space and cannot be efficiently utilized with conventional methods such as GMM-based classifiers. These features are used as a motivation for developing some of the algorithms presented in Chapter 3. Section 1.2 reviews some of the features previously proposed that are pertinent to the work presented here. Section 1.3 briefly recalls some of the previous work in audio classification and Section 1.4 outlines the salient contributions of this thesis.

## **1.1 Background**

### **1.1.1 Early Auditory System**

From a signal processing perspective, signals reach the middle ear relatively unchanged. The middle ear is composed of three small bones, or ossicles, which provide gain control and impedance matching between the outer and the inner ear. The middle ear couples the sound energy in the auditory canal to the inner ear or the cochlea, which is a snail-shaped bone.

Figure 2 shows a cross sectional view of the cochlea. The input to the cochlea is through the oval window. The oval window leads to one of three fluid-filled compartments within the Cochlea. These chambers, called scala vestibuli, scala media, and scala tympani, are separated by flexible membranes. The Reissner’s membrane separates the scala vestibuli from the scala media, and the basilar membrane separates the scala tympani from the scala



**Figure 2.** A cross section of the human cochlea. Within the bone are three fluid-filled chambers that are separated by two membranes. The input to the cochlea is in the scala vestibuli, which is connected at the apical end to the scala tympani. Pressure differences between these two chambers leads to movement in the basilar membrane.

media [5]-[6].

As the oval window is pushed in and out as a result of incident sound waves, pressure waves enter the cochlea in the scala vestibuli and then propagate down the length of the cochlea. Since the scala vestibuli and the scala tympani are connected, the increased pressure propagates back down the length of the cochlea through the scala tympani to the basal end. When the pressure wave hits the basal end, it causes a small window, called the round window, to bow outward to absorb the increased pressure. During this process, the two membrane dividers bend and bow in response to the changes in pressure [7], giving rise to a traveling wave in the basilar membrane.

At the basal end, the basilar membrane is very narrow but gets wider toward the apical end. Further, the stiffness of the basilar membrane decreases down its length from the base to the apex. Because of these variations along its length, different parts of the basilar membrane resonate at different frequencies, and the frequencies at which they resonate is

highly dependent upon the location within the cochlea. The traveling wave that develops inside the cochlea propagates down the length of the cochlea until it reaches the point where the basilar membrane resonates with the same frequency as the input signal. The wave will essentially die out after the point where resonance occurs because the basilar membrane will no longer support the propagation. It has been observed that the lower frequencies travel further than the higher frequencies. Also, the basilar membrane has exponential changes in the resonant frequency for linear distances down the length of the cochlea.

The basilar membrane is also attached to what is known as the Organ of Corti. One important feature of the Organ of Corti is that it has sensory cells called inner hair cells (IHC) that sense motion of the basilar membrane. As the basilar membrane moves up and down in response to the pressure waves, it causes local movement of the cochlear fluid. The viscous drag of the fluid bends the cilia attached to the IHC. The bending of the cilia controls the ionic flow into the hair cells through a nonlinear channel. Because of this ionic current flow, charge builds up across the hair cell membrane. This mechanism converts the mechanical displacement of the basilar membrane into electrical activity. Once the potential builds up above a certain threshold, the hair cell fires. This neural spike is carried to the cochlear nucleus by the auditory nerve fiber. The neurons in the cochlear nucleus (CN) exhibit inhibition characteristics and it is believed that lateral inhibition exists in the cochlear nucleus. The lateral interaction of the neurons is spatially limited, i.e., as the distance between the neurons increases the interaction decreases [3].

### **1.1.2 Mathematical Model of the Auditory System**

#### *1.1.2.1 Model of the Peripheral Auditory System*

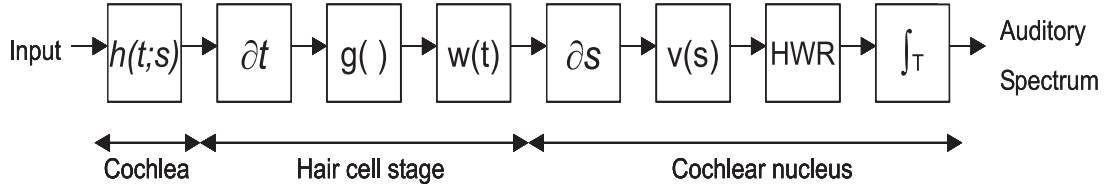
Yang *et al.* [8] have presented a biophysically defensible mathematical model of the early auditory system. The model is shown in Fig. 3 and described below.

When viewing the way the cochlea acts on signals of different frequencies from an engineering perspective, it can be seen that the cochlea has bandpass frequency responses for each location. An accurate but computationally prohibitive, model would have a bank of bandpass filters with center frequencies corresponding to the resonant frequency of every point along the cochlea—the cochlea has about 3000 inner hair cells acting as transduction

points. In practice 10-20 filters per octave are considered an adequate approximation. The cochlear filters,  $h(t; s)$ , typically have 20 dB/decade roll-offs on the low-frequency side and a very sharp roll-off on the high-frequency side.

The coupling of the cochlear fluid and the inner hair cells is modeled by a time derivative ( $\partial t$ ). This can be justified since the extent of IHC cilia deflection depends on the viscous drag of the cochlear fluid and the drag is directly dependent on the velocity of motion. The nonlinearity of the ionic channel is modeled by a sigmoid-like function,  $g(\cdot)$ , and the leakiness of the cell membrane is modeled by a lowpass filter,  $w(t)$ .

Lateral inhibition in the cochlear nucleus is modeled by a spatial derivative ( $\partial s$ ). The spatial derivative is leaky in the sense that it is accompanied by a local smoothing that reflects the limited spatial extent of the interactions of the CN neurons. Thus, the spatial derivative is often modeled along with a spatial lowpass filter,  $v(s)$ . The nonlinearity of the CN neurons is modeled by a half-wave rectifier (HWR) and the inability of the central auditory neurons to react to rapid temporal changes is modeled by temporal integration ( $\int_T$ ). The output of this model is referred to as the auditory spectrum and it has been shown that this representation is more robust to noise compared to the normal power spectrum [3].

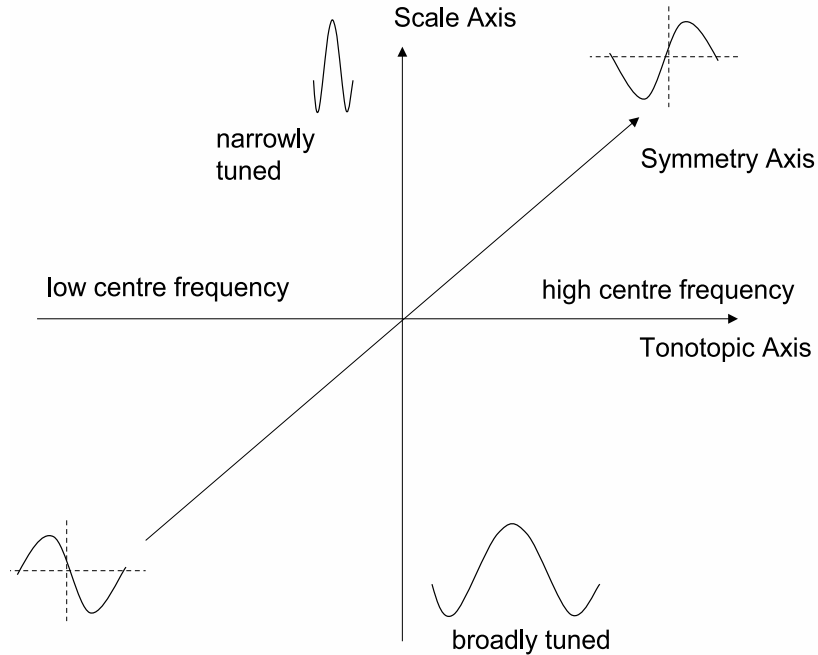


**Figure 3.** Mathematical model of the early auditory system consisting of filtering in the cochlea (analysis stage), conversion of mechanical displacement into electrical activity in the IHC (transduction stage) and the lateral inhibitory network in the cochlear nucleus(reduction stage) [3].

#### 1.1.2.2 Cortical Model

Wang and Shamma [9] have proposed a model of the spectral shape analysis in the primary auditory cortex. The schematic of the model is shown in Fig. 4. According to this model neurons in the primary auditory cortex (A1) are organized along three mutually perpendicular axes. The response field of neurons lined along the tonotopic axis are tuned to different

center frequencies. The bandwidth of the response field of neurons lined along the scale axis monotonically decreases along that axis. At the center of A1, the response field has an excitatory center, surrounded by inhibitory side bands. The response field tends to be more asymmetrical with increasing distance from the center of A1. It has been argued that the tonotopic axis is akin to a Fourier transform and the presence of different scales over which this transform is performed leads to a multi-scale Fourier transform. It has been shown that performing such an operation on the auditory spectrum leads to the extraction of spatial and temporal modulation information [10].



**Figure 4. Schematic of the cortical model.** It is proposed in [9] that the response fields of neurons in the primary auditory cortex are arranged along three mutually perpendicular axes. The tonotopic axis, the bandwidth or scale axis and the symmetry or phase axis.

## 1.2 Review of Filter-Bank Features for Speech Recognition

In this section previous work using filter-bank features is briefly reviewed. White and Neely [11] compared filter-bank energy with linear predictive coding (LPC) for speech recognition tasks. Dynamic programming was used as the back-end (for time alignment). Twenty

one-third octave filters were used for the frequency decomposition. Spectral shaping was achieved by adjusting the gain of each filter. Output of each channel was energy smoothed, noise subtracted (achieved by rectification, subtraction by a constant value and summing over 10 msec) and subjected to log amplitude scaling. This signal was sampled at 100 Hz and fed to the recognizer. For comparison, fourteen LPC coefficients were calculated every 12.8 msec (using autocorrelation). Hamming window of length 25.6 msec was used. It was reported that both LPC and the filter-bank method performed comparably. However, noise-robustness of the features was not tested. Since the recognizer did not use Gaussian mixture models with diagonal covariance no decorrelation was done. Searle et al. [12] designed a phoneme detector using filter-bank energy. A 16-channel, one-third octave bandpass filter (BPF) filter-bank was used. High speech, wide dynamic range envelope detectors were used at the output of each channel, followed by a logarithmic amplifier. They were able to capture temporal information, including voice onset time (VOT) and also some spectral information. Further, the outputs of the channels were sampled at 625 Hz and plotting the output of each 1.6 msec time slice side by side for about 100 msec gave enhanced information about the spectral aspects of the signal. In this representation voicing was seen by an increase in energy at low frequencies and by periodic bunching of the spectra above 1 kHz. A running average of the time slices over 5-10 slices for various speakers was used to suppress interspeaker variability. The features used were, VOT, frequency, amplitude, curvature of the energy peaks (formant tracks) with regard to the frequency at the burst, and 20, 45 and 90 msec after the burst. Kimberley and Searle [13] also implemented a similar classifier for fricative discrimination. Dautrich et al. [14] studied different filter design choices (number of channels, type of filter, filter spacing, overlapping or not) and filter-output processing choices on performance of the recognizer. Eight uniform and five non-uniform filters with varying amounts of overlap were considered. It was reported that a 15-channel uniform filter-bank and a highly-overlapping 13-channel non-uniform critical band filter-bank performed best on a 39-word alphadigit vocabulary. It was also reported that an 8<sup>th</sup> order LPC-based recognizer performed better than the filter-banks. However, it is interesting to note that the performance of the LPC-based recognizer deteriorated faster than that of filter-bank implementation, LPC was better for SNR greater than or equal to 6



dB. Speech signal was bandlimited to 3200 Hz and sampled at 6.67 kHz. The preprocessed (spectral shaping operation to correct the 6 dB per octave spectral tilt) signal was then passed through a filter-bank, a non-linearity (full-wave rectifier), a lowpass filter (cutoff of 30 Hz), a sampler (rate of 67 Hz) and a logarithmic compressor. Post-processing considered were, threshold and energy normalization, and temporal and spectral smoothing. Thresholding clamps low level noise signal and energy normalization is done to remove variations from utterance to utterance. Mean and peak energy normalization were done (i.e. either peak energy or mean energy was subtracted from each channel output on per frame basis). Smoothing was performed in order to remove channel variations. This could result in loss of spectral and temporal resolution. None of these post-processes substantially improved the performance. A dynamic time warping based recognizer was used as the back-end. Ghitza [15] conducted recognition tests on the Ensemble Interval Histogram (EIH) setup but with the cochlear filters replaced with uniformly shaped Hamming filters (linear scale). It was shown that recognition rate was better than when Fourier power spectrum measurement (short term power at output of each filter) was used in both clean and noisy cases. Further, the recognition was also better than when the cochlear filters were used. The author concludes that it is not the shape of the filter but the timing-synchrony analyzer that leads to noise robustness. In the original EIH setup 85 cochlear filters equally spaced on the log-frequency scale from 200-3200 Hz were used. Level crossing detectors with different positive thresholds equally spaced on log scale were used to produce a “spectrum.” However it should be noted that for the control case, too many filters (85) were used in the filter-bank, this reduces recognition rate (see [14]). Nadeu et al. [16] did a comparison of decorrelated filter-bank energy (FBE) with MFCC. The decorrelated FBE are obtained by filtering the log FBEs to equalize the variance of the cepstral coefficients. The filter used (1<sup>st</sup> order high-pass filter) provides both equalization and decorrelation. The authors also mention that the output of the HPF (derivative type filter) is a spectral slope measure and is a perceptually relevant characteristic for phonetic distance. Continuous observation density HMM was used for recognition. It was reported that doing Karhunen-Loeve transform (KLT) for decorrelation (of the average subtracted log FBE) did not perform as well as high-pass filtering. In the control case 20 channels and 8 MFCCs were used, the authors claimed this

to be the empirical optimum number for MFCCs. Further, no delta or acceleration features were used. Nadeu et al. [17] considered 2-D log FBE as features for speech recognition. The authors designed with a Quadrature mirror filter (QMF) representation, which is obtained by taking an inverse DFT along the frequency axis and then taking another Fourier transform along the time axis to obtain a modulation spectrogram. They contend that since weighting of the cepstrum does not improve recognition using continuous observation Gaussian density HMM (due the variance normalization of the Gaussian pdfs), it is better to perform filtering in the frequency (filtering in the frequency domain leads to implicit weighting in cepstral domain) and time domains and not make a transition to the quefrequency domain. The features used here were obtained by, mean subtraction, variance equalization by 1<sup>st</sup> order HPF, and lowpass filtering for shaping the equalized bands. Performance compared to MFCC was better only when energy was not used as a feature. In case of discrete HMMs when 12 MFCCs, energy, and delta's were used along with cepstral mean subtraction (CMS), filtered log FBEs did not provide better performance. Paliwal et al. [18] use a linear predictor for the decorrelation of log FBE coefficients. An FIR highpass filter (HPF) was used to lifter the log FBE features. It was reported that log FBEs perform better than MFCCs in clean and noisy conditions (delta features were incorporated for both). Energy was not used with MFCC as is usually done. Mantha et al. [19] combined perceptual linear predictor (PLP), filter-bank amplitudes (FBA) and MFCC and delta features for HMM based speech recognition. While computing FBA linear phase, critical band filters were used. The main objective however was in comparison of the recognition back-ends and no insight into feature performance was given.

### 1.3 Review of Previous Audio Classification work

In the audio classification literature, many features and various classifiers have been tested with varying degrees of success. Zhang and Kuo [20] developed a hierarchical system for audio classification and reported that using temporal curves of energy, average zero-crossing rate, and fundamental frequency they were able to achieve over 90% accuracy while classifying sound into speech, music, noise, and silence using a rule-based heuristic procedure. They also performed the “fine classification” of sounds into further subgroups using timbre

and rhythm as features and Gaussian mixture models (GMMs) and hidden Markov models (HMMs) as classifiers. For a 10-cluster noise subgroup they reported 80% accuracy. Gaunard et al. [21] describe a system for noise classification. They use LPC-cepstral features with a discrete HMM as the classifier. The categories considered were “car,” “truck,” “aircraft,” “moped,” and “train.” They report that the best result obtained was 95.3% accuracy. However, the database used was small and the variability in the categories was limited. Goldhor [22] presented a system for classifying different environmental sounds such as different bells, running water, drill, fan, car engine, etc. Two-dimensional cepstral coefficients were used as the features and clustering was performed to obtain the classification results. He reported very high classification accuracy when 12 or more cepstral coefficients were used. Kates [23] presented a noise classification system that would enable the automatic adjustment of electroacoustic response of a hearing aid. He extracted envelope fluctuation mean-to-standard deviation ratio, mean of frequency, low-frequency slope, and high-frequency slope from sound samples and performed cluster analysis for the classification. Using 2 seconds of data he was able to obtain above 90% accuracy for seven or fewer clusters. Allegro et al. [24] describe a system to distinguish between speech, music, noise, and speech in noise that was specifically designed for automatic switching in hearing aids. Their feature set includes width extracted from an amplitude histogram [25], frequency centroid, fluctuation of frequency centroid, tonality, and pitch variance. The classification is performed using a HMM-based classifier with majority voting as a post-processor. They reported over 90% accuracy in classifying speech, and over 80% accuracy for each of music and speech in noise, and 65% for noise. They also reported low false positive rates, between 7.8% and 10%. However, 30 seconds of data is used for the classification. Peltonen [26] et al. considered the problem of recognizing 17 different scenes. They reported that the best results were obtained with MFCCs as features and a GMM-based classifier. The mean and variance of MFCC features over a segment were concatenated to form the feature vector. However, their classification results are also based on 30 seconds of data.

## 1.4 Contributions of this Research

The contributions of this thesis are in two main areas, namely, incorporating some of the functionalities of the peripheral auditory system into state-of-the-art features to improve their performance in clean and noisy conditions, and developing data processing and classification algorithms based on ideas from the machine learning community that are geared towards working directly with features derived from advanced models of the auditory system (which are usually sparse and high-dimensional). Some of the algorithms presented can also work with various types of features, thus allowing us to combine the benefits afforded by different feature representations.

1. Developed an understanding of noise robustness issues with the MFCC representation.

The MFCC features were studied in detail and new insights into some of the failings of these features were developed. The triangular filtering in the MFCC processing is sensitive to small changes in frequency which leads to a representation in each channel that is not very smooth. Further, ignoring the phase information and downsampling in the frequency domain discards information that is not exactly quantifiable and hence one cannot guarantee that there is no aliasing of information or that useful information is not being masked.

2. Developed features which have been shown to be better than MFCCs in noisy conditions, but more importantly these features do not degrade the performance in clean conditions.

- Showed that varying time constants in the feature extraction process has an important role in noise-robustness. Varying time constants to suit speech modulations helps to mask noise modulations. In the modulation domain, noise can be represented as consisting of a DC component and modulation terms. The DC component is readily removed using mean subtraction and thus, filtering out the modulation terms leads to a relatively cleaner representation.
- Incorporated a new gain adaptation technique into the feature extraction process to improve the performance of auditory features in clean conditions. The gain

adaptation technique demands very little computational overhead by exploiting the fact that the feature extraction process inherently extracts the signal envelope in each channel.

- Developed a method for SNR-based gain adaptation for feature extraction. Showed that the amount of compression should be a function of the SNR and varying the amount of compression based on the SNR leads to improved performance in all noise conditions. Further, it is shown that the new gain adaptation method developed has links to the Wiener gain function and can be used for noise suppression applications. The presented method however, does not depend on very accurate estimates of SNR, thereby addressing one of the main concerns of Wiener filtering.
3. Developed a multi-class AdaBoost-based classifier which is a collection of binary classifiers wherein a confidence measure based majority voting is used to combine the classifiers.
  4. Developed an AdaBoost-based dimensionality reduction technique by constraining the AdaBoost algorithm to pick a different feature at each iteration. This leads to a dimensionality reduction technique which does not transform the feature space. This has applications in feature selection and merging of different classifier outputs.
  5. Developed the cascade jump support vector machine (SVM) classifier which has better generalization ability as compared to a single kernel SVM and is computationally less expensive. The discrimination ability afforded by different kernels is exploited to build a classifier that not only improves the accuracy but avoids over-fitting compared to a traditional SVM.
  6. Developed a generative AdaBoost classifier that scales well with large number of classes. The concept of boosting density estimates is used to build an AdaBoost-based classifier that provides a likelihood measure for each class and thus scales well to large number of classes. Three different approaches to computing the mixing weights are presented. A technique for improving the estimate of a single base estimator is also

presented. The performance of a single base estimator is improved by combining its estimates on different transformations of the input data.

7. Developed an adaptive normalization technique that learns the normalization parameters adaptively and is shown to improve the performance over segment-based normalization techniques. A Kalman filter is used to learn the mean and variance of the feature set in an adaptive manner. The advantage of such a technique is that it is able to adapt to changes in the recording environment or transmission channel.

The objective of this thesis is to bring together ideas from physiological processing and machine learning. On one hand it strives to use machine learning techniques to harness the benefits afforded by features based on modeling physiological processes and on the other hand, it strives to show the links between physiological processing and popular ideas in the machine learning community. In the effort to bring together these two diverse fields, this work serves to create promising research avenues that could advance audio and speech processing beyond mere incremental improvements.

## CHAPTER 2

### IMPROVING NOISE ROBUSTNESS OF PRIMARY FEATURES

The research presented in this chapter addresses the issue of robust feature extraction for speech and audio processing. Mel-frequency cepstral coefficients (MFCCs) [27], the current state-of-the-art features in audio processing, are known to perform poorly in the presence of noise [28]. MFCCs are loosely modeled on physiological processing and herein, modifications to MFCCs are suggested based on a more detailed model of the peripheral auditory system. The new features are compared with MFCCs for audio classification, speech recognition, and speech versus non-speech discrimination. It is shown that the proposed features are more robust to noise and have better class discrimination ability. Analysis of the noise robustness of these features is also presented. The organization of the chapter is as follows, Section 2.1 deals with some of the issues concerning the MFCC features, Section 2.2 presents the modifications to MFCCs that lead to the new feature representation. This section also studies the noise robustness of the new features and evaluates the noise performance both quantitatively and qualitatively. Section 2.3 compares the performance of the two feature sets on various audio processing tasks, Section 2.4 introduces further changes to the feature extraction process that effectively filters out noise modulations to improve the performance of the features in noisy conditions, Section 2.5 studies the effect of varying degrees of compression on the noise robustness of features and presents a new gain adaptation technique, Section 2.6 presents some design insights, and Section 2.7 summarizes the findings presented in the chapter.

#### 2.1 Issues with Mel-Frequency Cepstral Coefficients

MFCC's are very useful features for audio processing in clean conditions. However, performance using MFCC features deteriorates in the presence of noise. There has been an increased effort in recent times to find new features that are more noise robust compared to MFCCs. Features such as, spectro-temporal modulation features [4] are more robust to noise but are computationally expensive. Skowronski and Harris [29] suggested modification of MFCC that uses the known relationship between center frequency and critical bandwidth.

They also studied the effects of wider filter bandwidth on noise robustness. Herein, more fundamental issues with MFCCs relating to time-frequency trade-off and masking of relevant information are addressed.

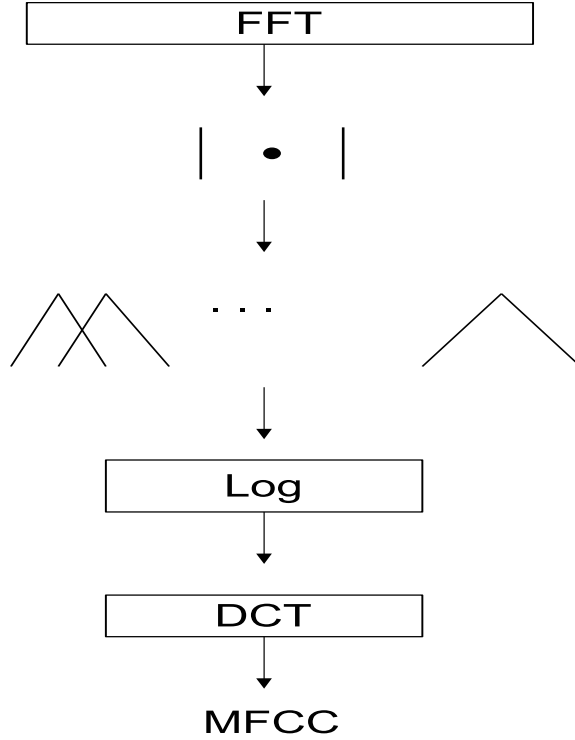
MFCC features approximate the frequency decomposition along the basilar membrane by a short-time Fourier Transform. The auditory critical bands are modeled using triangular filters, compression is expressed as a log function and a discrete cosine transform (DCT) is used to decorrelate the features [27]. MFCC feature extraction is shown in Fig. 5.

In most audio feature extraction processes the number of samples used to represent each frame is small compared to the original sampled waveform. Given that there will be some loss of information in building a compact representation of the audio signal, the key to generating better representations is to discard information that is least significant. In case of MFCCs, the FFT followed by grouping into critical bands using triangular filters leads to discarding of information that is not easily quantifiable. The temporal information in the signal is distributed in the magnitude and phase of the multiple frequency bins and combining them could lead to masking of pertinent information. As is explained in the next section, it is possible to discard information in a way that guarantees that perceptually relevant information is not lost.

The MFCC front-end due to its dependence on block processing and combination of frequency bins leads to a representation that has low time and frequency resolutions. In the human auditory system the asymmetrical shape of the cochlear filters allows for good time resolution (due to its gradual roll-off on the low frequency side) and good frequency resolution (due to the sharp cut-off on the high frequency side) [30]. But even without the asymmetrical shape, bandpass filtering is desirable since it avoids the widowing effects due to block processing and provides better temporal resolution compared to the short-time Fourier transform (wherein temporal resolution is restricted by the size of the analysis window and frame rate). Also, the use of triangular filters for critical band filtering leads to large changes in gain for small changes in the frequency [31] leading to a representation that is not as smooth as that obtained using BPFs. Thus even relatively small amount of noise tends to distort the MFCC representation. Figure 6 shows the distortion in the MFCC-based spectrum when noise (babble noise at 10 dB SNR) is added to the input



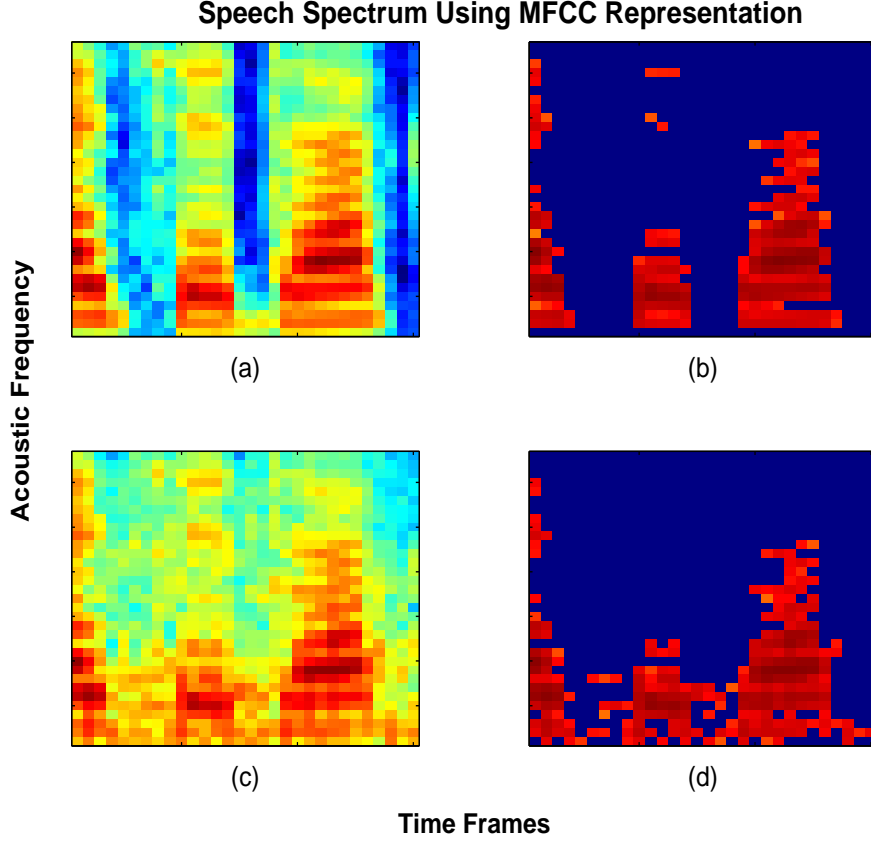
speech signal. As can be seen from Fig. 6(c), some speech information is lost due to the addition of noise. In most classification systems, in order to counter the effect of noise, mean subtraction is done as a post processing step, Figs. 6(b) and 6(d) show the mean subtracted spectrum. As is clear even with mean subtraction there is considerable distortion in the MFCC representation.



**Figure 5.** Figure showing the extraction of MFCC. Frequency decomposition is accomplished using the FFT and the critical bands are modeled using triangular filters. The logarithm provides static compression and decorrelation is achieved using the discrete cosine transform (DCT).

## 2.2 Noise-Robust Auditory Features (NRAF)

The NRAF features are derived from a model of the early auditory system [3]. The input signal is passed through a bandpass filter-bank. The filter-bank output is subjected to a spatial derivative. This is followed by a half-wave rectification and a smoothing filter. The half-wave rectification followed by the smoothing can be thought of as an envelope follower. The output at this stage is referred to as the auditory spectrum [3]. The auditory spectrum is subjected to amplitude compression and a discrete cosine transform (DCT) to obtain the



**Figure 6.** Figure showing the speech spectrum using MFCC representation. (a) shows the clean speech spectrum, (b) shows the clean speech spectrum with mean subtraction (c) shows the noisy speech spectrum (d) shows the noisy speech spectrum with mean subtraction. It is clear that even with mean subtraction, noise affects the MFCC representation

NRAFs. The feature extraction process is shown in Fig. 7.

### 2.2.1 Motivation for using BPF

As noted above, the asymmetrical shape of the cochlear filters allows for good time and frequency resolution. From a mathematical standpoint we can argue the case for BPF using the uncertainty principle. It is well known (see [32] and references therein) that for any two quantities represented by operators which do not commute, there exists an uncertainty principle, i.e. for quantities  $a$  and  $b$  represented by operators  $A$  and  $B$ ,

$$\Delta a \Delta b \geq \frac{1}{2} |[A, B]|$$

where  $\Delta a$  and  $\Delta b$  are the uncertainty (defined as mean-square deviation) in quantities

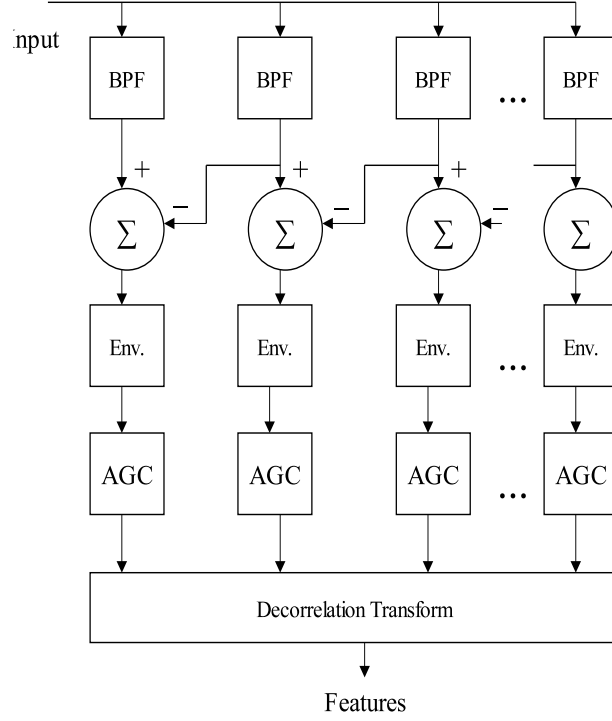


Figure 7. The bandpass filtered version of the input is subjected to a spatial derivative (approximated by a difference operation). The half-wave rectification followed by the smoothing filter is used for envelope detection. AGC represents amplitude compression, which is followed by DCT to decorrelate the signal.

$a$  and  $b$ . For time and frequency, the operators can be defined as:

$$A = t$$

and,

$$B = -j \frac{d}{dt}$$

It is easy to see that  $A$  and  $B$  do not commute, i.e.

$$[A, B] = AB - BA = -j \quad (1)$$

For a signal  $s(t)$ ,  $\Delta t$  and  $\Delta \omega$ , are defined as

$$(\Delta t)^2 = \int (t - \langle t \rangle)^2 |s(t)|^2 dt$$

$$(\Delta \omega)^2 = \int (\omega - \langle \omega \rangle)^2 |\hat{s}(\omega)|^2 d\omega$$

where  $\hat{s}(\omega)$  is the Fourier transform of  $s(t)$  and  $\langle \cdot \rangle$  represents the expectation. The uncertainty relation is simply,

$$\Delta t \Delta \omega \geq \frac{1}{2} \quad (2)$$

Thus, there is a trade-off between time and frequency resolutions and the representation that is closer to the lower bound on the above equation would be the better representation. From psychoacoustic experiments we know that humans only need limited frequency resolution to process audio signals [33]. If we assume that the frequency resolution of the bandpass filter method and grouping of FFT bins using triangular filters (as in MFCCs) is the same then the improved temporal resolution of the BPF leads to a better representation in terms of the time-frequency resolution trade-off. The problem with MFCCs is that temporal resolution is sacrificed to obtain a relatively high frequency resolution representation and then the frequency resolution is reduced by combining into critical bands. Thus the final representation has relatively low time and frequency resolution.

### 2.2.2 Noise Robustness of NRAF

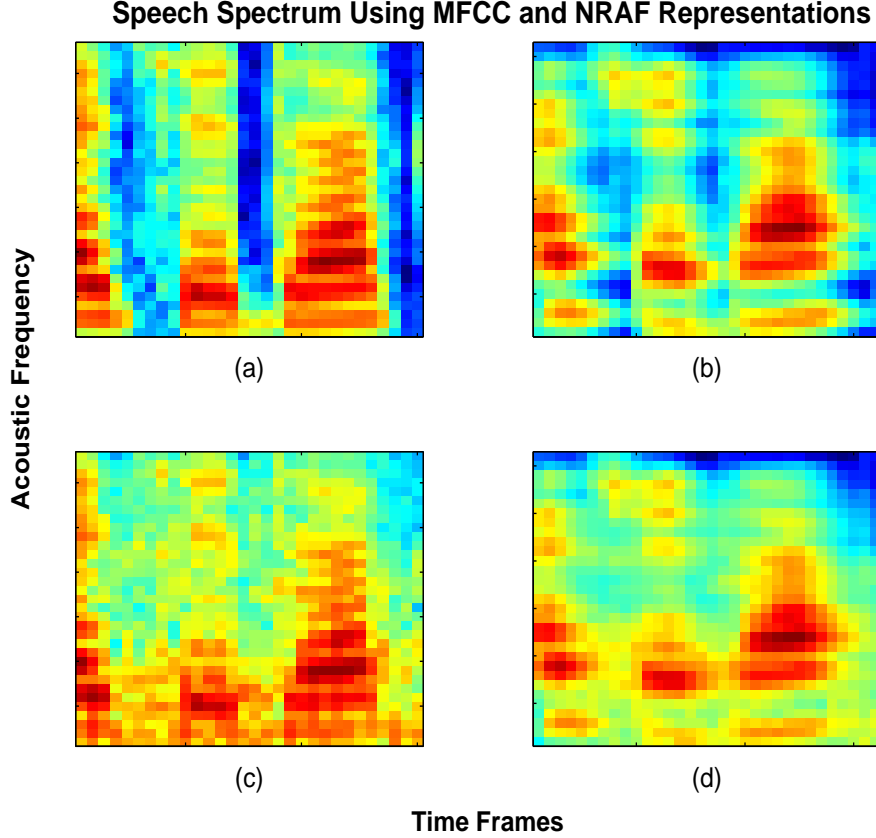
As mentioned in the previous section, the use of triangular filters for grouping frequency bins leads to a representation in each channel that is sensitive to small changes in frequency and thus even small amounts of noise tend to affect the representation. The energy estimation in each channel is smoother if frequency decomposition is performed using exponentially spaced bandpass filters and the signal strength in each channel is estimated using an envelope detector (implemented using a rectifier and a lowpass filter). Lowpass filtering before downsampling ensures that there is no temporal aliasing. The lowpass filter does not discard perceptually relevant information since we know that the central auditory neurons cannot respond to very fast temporal modulations [8]. The fast temporal variations that

are smoothed out are most likely perceptually insignificant. Further, envelope extraction following bandpass filtering allows us the opportunity to filter out the noise modulations in each channel to some extent (explained in further detail in Section 2.4).

### 2.2.3 Evaluation of Noise Robustness of NRAF features

In this section we compare the noise robustness of MFCC's and NRAF's. Figure 8 shows the effect of noise on the speech spectrum obtained using the MFCC and NRAF representations. It is clear that the NRAF representation is able to retain most of the speech information even in the presence of noise. Figure 9 shows the mean subtracted spectrums using the MFCC and NRAF representations. Again, it is seen that with mean subtraction, which gets rid of the noise DC component, NRAF is able to better preserve the speech modulation information. Figure 10 shows the per channel SNR for MFCC and NRAF (before the compression stage) for a noisy speech input. It is clearly seen that the NRAF representation has a better SNR per channel (on average) compared to the MFCC representation. The improvement in SNR is due to the fact that the spatial derivative gets rid of most of the wide-band noise. Figure 11 shows that using 4<sup>th</sup> order BPF instead of the cochlear filters proposed in [3], the frequency spreading can be limited. It is seen that removing the spatial derivative stage improves the noise performance of the features in very low SNR cases (see Table 4). The reason being that in high noise cases the spatial derivative (which is approximated by a difference between adjacent channels) removes the signal from those channels whose adjacent higher channels are noisy. In other words, the noise signal consists of a bias component and a variance term, in high SNR cases where the noise variance is low compared to the signal variance, spatial derivative amounts to removing the bias component of noise. In very low SNR cases where the noise variance is equal to or greater than signal variance, spatial derivative results in some loss of signal component along with noise removal. However, the spatial derivative stage is still useful in clean and high SNR conditions where changes across the spectral profile are enhanced by the difference operation. This can be looked upon as an edge detection operation common in image processing, although the effect in audio is less dramatic due to lack of abrupt changes across frequency channels. NRAF without the spatial derivative can be looked upon as continuous-time MFCC extraction [34], [35]. From

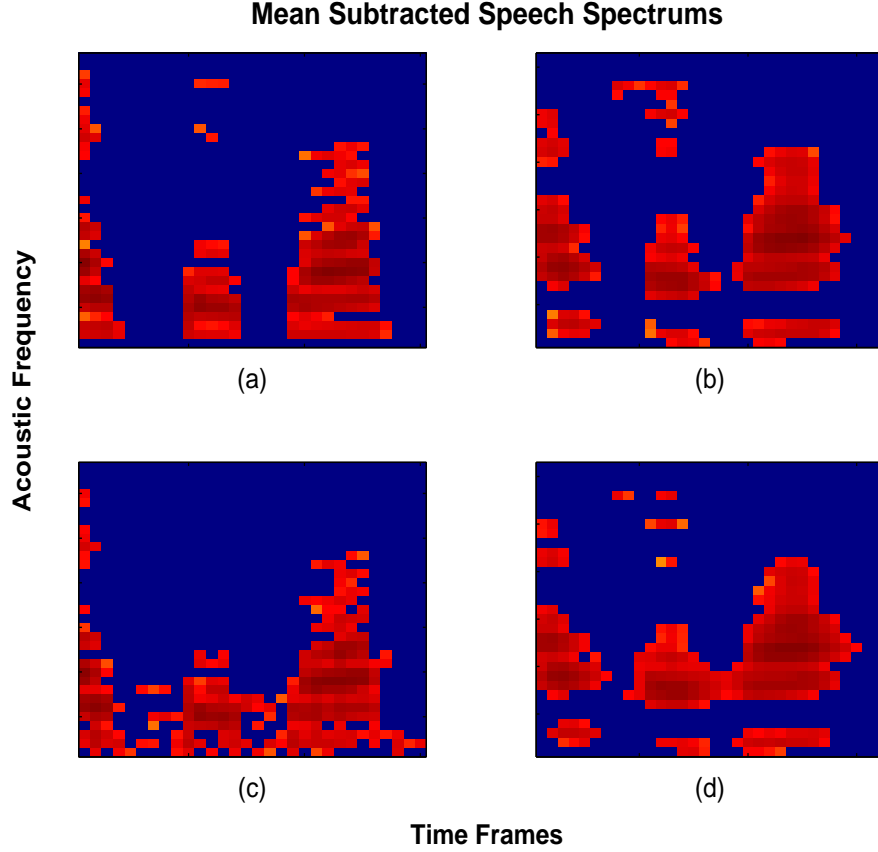
a signal processing perspective it can be argued that the spatial derivative only tightens up the filter response and using an appropriate filter would accomplish the same. The issue however lies in the design of the appropriate filter. Spatial derivative circumvents this issue and allows for obtaining a high  $Q$  filter using lower order filters.



**Figure 8.** Figure showing the comparison of clean and noisy speech spectrums for the MFCC and NRAF representations. (a) clean speech spectrum using the MFCC representation, (b) clean speech spectrum using the NRAF representation, (c) noisy speech spectrum using the MFCC representation, (d) noisy speech spectrum using the NRAF representation. It is quite evident that NRAF representation is able to retain most of the speech information even in the presence of noise. Babble noise was synthetically added.

Visual examples of the noise-robustness of the NRAF front-end are shown in Figs. 12 and 13. Figure 12 shows the envelope in a particular frequency channel ( $\approx 200$  Hz) for the MFCC and NRAF front-ends for a noisy speech input. It is evident that the MFCC representation deteriorates faster than the NRAF representation. Figure 13 shows the same effect for a frequency channel with center frequency close to 800 Hz.

A spectrogram is a representation that presents the input signal as a plot of acoustic



**Figure 9.** Figure showing the comparison of mean subtracted clean and noisy speech spectrums for the MFCC and NRAF representations. (a) clean speech spectrum using the MFCC representation, (b) clean speech spectrum using the NRAF representation, (c) noisy speech spectrum using the MFCC representation, (d) noisy speech spectrum using the NRAF representation. As is evident, with mean subtraction the NRAF feature is able to keep out most of the noise while retaining the speech information while the MFCC representation still suffers from the effects of noise.

frequency versus time. By performing a Fourier transform across the time axis one can obtain a representation which is acoustic frequency versus modulation frequency. This representation is referred to as the modulation spectrogram. Modulation spectrograms are very useful in studying the different modulating signals present in an acoustic signal. The modulation spectrogram is used to portray the noise masking abilities of the NRAF representation. Figure 14 shows the modulation spectrogram of the MFCC and NRAF front-ends in clean and noisy conditions. It is clear from Fig. 14(a) and Fig. 14(c) that addition of noise leads to loss of speech modulations and introduction of undesirable noise modulations in the modulation spectrogram of the MFCC front-end. However, as evidenced

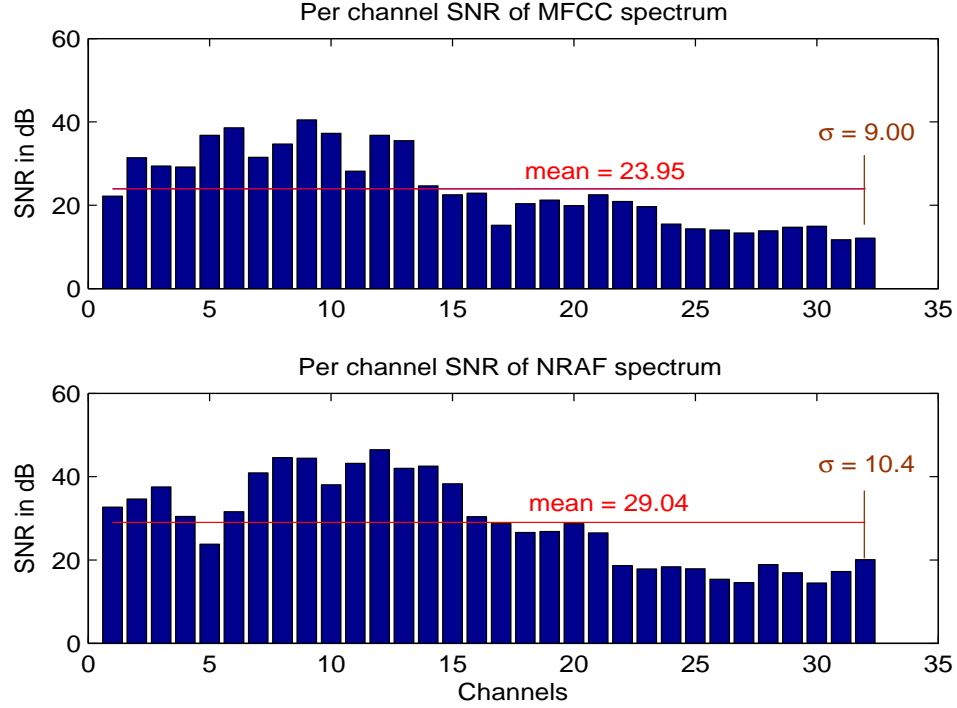


Figure 10. Figure showing the per channel SNR of MFCC and NRAF. Input was noisy speech with white noise synthetically added. It can be seen that NRAF yields higher per channel SNR. The mean of the SNR for MFCC representation is 23.94 and the standard deviation is 9, while the mean for the NRAF representation is 29.04 and the standard deviation is 10.4.

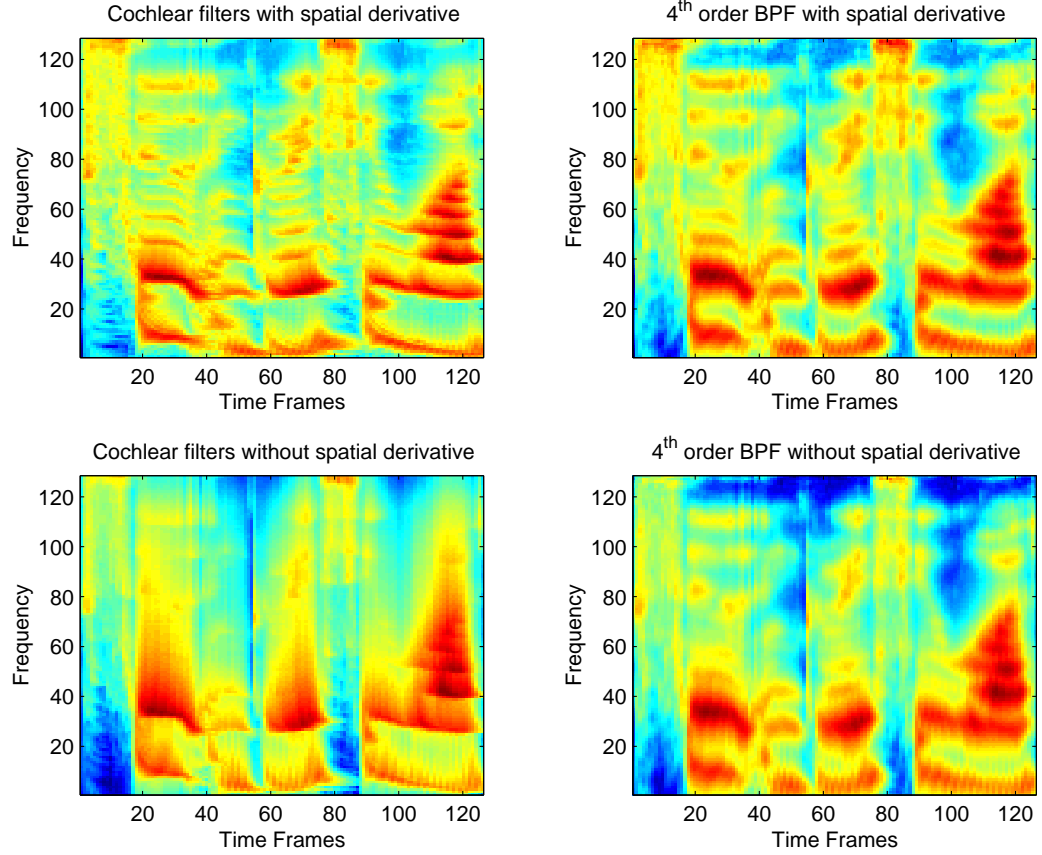
from Fig. 14(b) and Fig. 14(d), the NRAF representation not only preserves most of the speech information but also masks the noise modulations.

#### 2.2.4 Information-Theoretic Clustering Validity Measure

In this section we use an information theoretic measure of clustering to substantiate the argument that NRAFs are better than the original MFCCs not only in terms of noise-robustness but also in terms of class discrimination ability. Conditional entropy has been used as a criterion for evaluating the clustering validity of clustering algorithms [36]. By using a very “naive” clustering algorithm the clustering properties of the underlying attributes can be studied. Mahalanobis distance from the mean of the two clusters is used as the clustering algorithm to study the effect of synthetically added noise on the clustering properties of MFCC and NRAF.

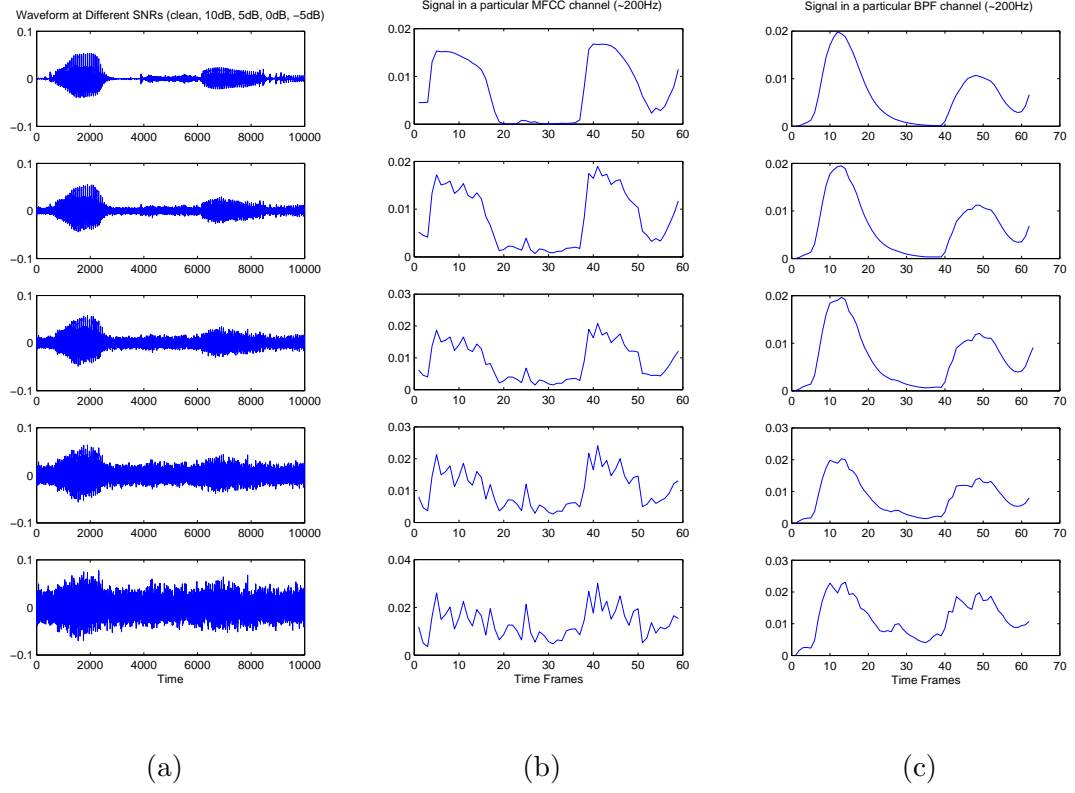
Given a set of class labels  $c \in C$  and clusters  $k \in K$ , it can be assumed that the class





**Figure 11.** Figure showing effect of spatial derivative. Plots on the left are the original auditory spectrums and those on the right are the auditory spectrums with 4<sup>th</sup> order BPFs. The plots on top were generated with spatial derivative and those at the bottom did not use spatial derivative. It is clear that using 4<sup>th</sup> order filters limits the frequency spreading. However, the spatial derivative stage is still useful in clean and high SNR conditions where changes across the spectral profile are enhanced by the difference operation.

labels and cluster labels are drawn from some distribution  $p(c)$  and  $p(k)$  respectively. And that each pair  $(c_i, k_i)$  associated with  $C$  and  $K$  are drawn from a distribution  $p(c, k)$ . The conditional entropy,  $H(C|K)$  can be approximated by empirical conditional entropy,  $H^e(C|K)$  given by,

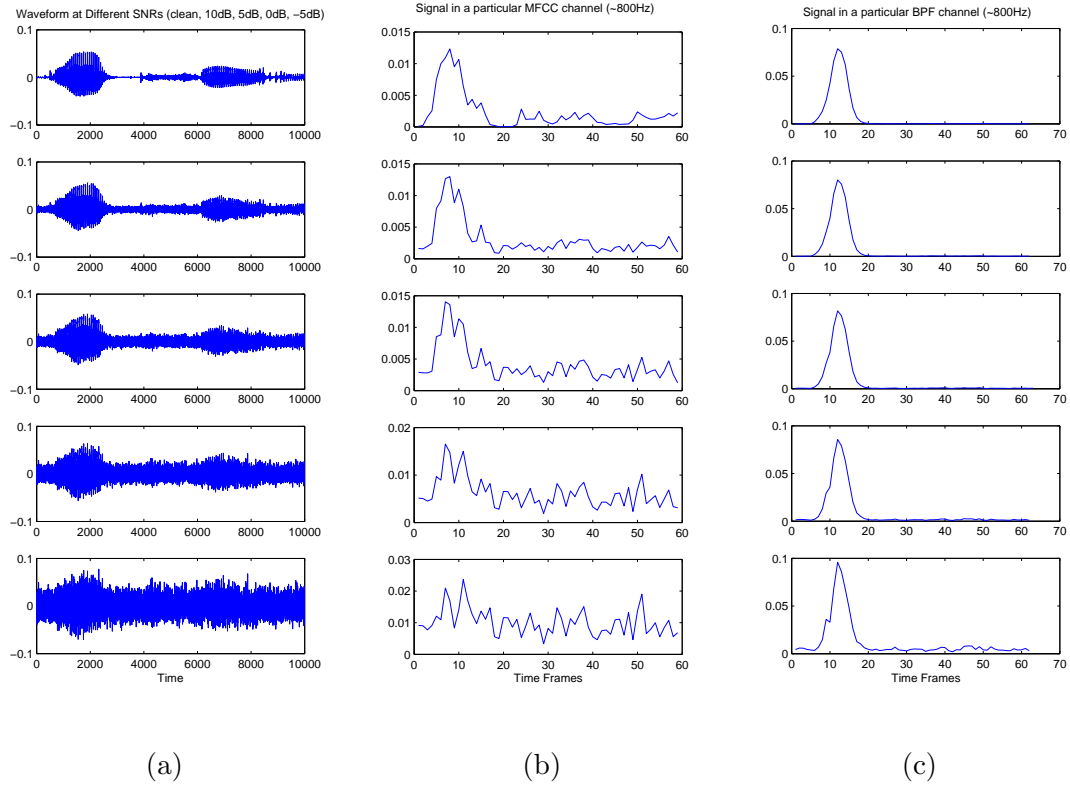


**Figure 12. Comparison of envelopes in a particular channel ( $\approx 200\text{Hz}$ ) for the MFCC and NRAF front-ends a) Speech input at different SNRs (clean, 20 dB, 10 dB, 5 dB and 0 dB) b) Envelopes using the MFCC front-end c) Envelopes using the NRAF front-end. It is seen that, even with addition of a small amount of noise, the MFCC representation is not very smooth. The NRAF representation is able to maintain the spectral peaks even at very low SNRs.**

$$\begin{aligned}
H^e(C|K) &= - \sum_{c=1}^{|C|} \sum_{k=1}^{|K|} \frac{h(c,k)}{n} \log \frac{h(c,k)}{h(k)} \\
&= - \sum_{c=1}^{|C|} \sum_{k=1}^{|K|} \frac{h(c,k)}{n} \log \frac{h(c,k)}{n} \\
&\quad + \sum_{k=1}^{|K|} \frac{h(k)}{n} \log \frac{h(k)}{n} \\
&= H^e(C, K) - H^e(K)
\end{aligned} \tag{3}$$

$$\tag{4}$$

where,  $h(c,k)$  is the number of examples of class  $c$  assigned to cluster  $k$ ,  $n$  is the total number of examples and  $h(k)$  is the associated marginal. Conditional entropy is a good measure of the validity of clustering since it gives the number of bits needed to encode the class labels given the model  $\Pi \equiv \{h(c,k)\}$  and the cluster labels  $\{k_i\}$ . In the best case scenario, the conditional entropy would be zero, indicating that the cluster labels had all



**Figure 13. Comparison of envelopes in a particular channel ( $\approx 800\text{Hz}$ ) for the MFCC and NRAF front-ends** a) Speech input at different SNRs (clean, 20 dB, 10 dB, 5 dB and 0 dB) b) Envelopes using the MFCC front-end c) Envelopes using the NRAF front-end. As in the previous case, the NRAF representation is much more robust to noise compared to the MFCC representation.

the information about the class labels. A lower value of conditional entropy indicates that the cluster labels are good indicators of the class labels.

The first evaluation task was chosen to be that of separating the speech class from the non-speech class. This can be looked upon as a two-cluster, two-class problem where speech forms one class and music, noise, and animal vocalizations form the second class. For this case, the empirical conditional entropy measure is shown in Figs. 15-16. As is evident, NRAF has a lower value of conditional entropy, demonstrating its robustness to noise and also its ability to separate the speech class from the other three classes. However another question remains, namely, how good are the NRAF features in separating all the classes. In other words, we want to test the discrimination ability of the NRAF features for all the four classes. This is easily tested by looking at the speech versus non-speech classification problem as a four class audio classification task. The empirical conditional entropy measure

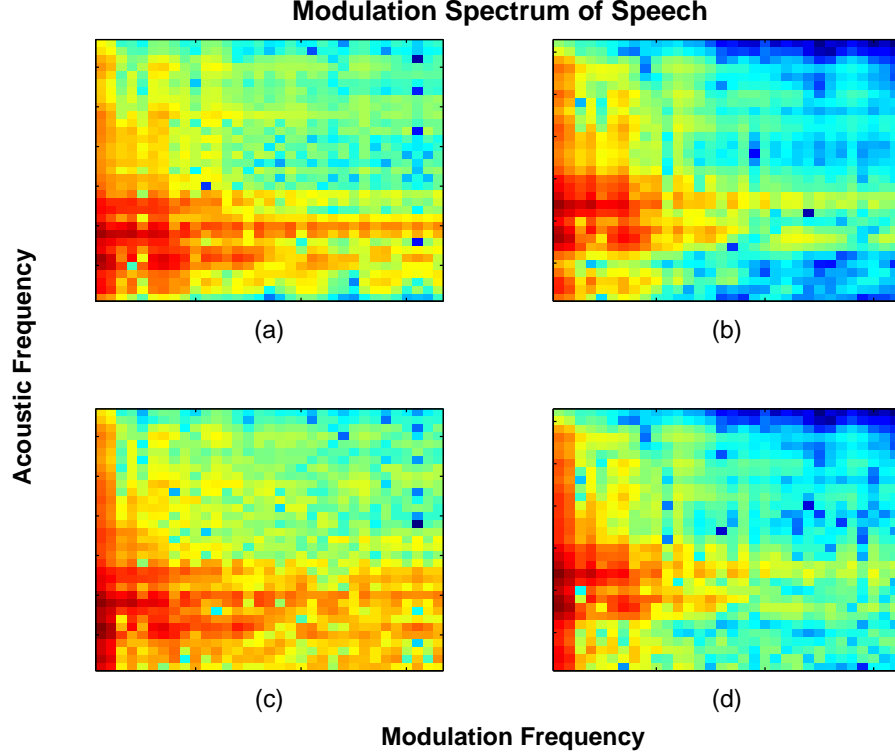


Figure 14. Comparison of modulation spectrograms of the MFCC and NRAF front-ends a) MFCC-based modulation spectrogram for clean speech b) NRAF-based modulation spectrogram for clean speech c) MFCC-based modulation spectrogram for noisy speech d) NRAF-based modulation spectrogram for noisy speech. As is evident, the NRAF representation is able to mask the noise modulations much better than the MFCC representation.

for the four-class, four-cluster case is shown in Table. 1. It is evident that NRAF is not only more robust to noise as compared to MFCCs but also has better class discrimination ability.

## 2.3 Experimental Performance Comparison of MFCC and NRAF

In this section NRAF and MFCC features are evaluated for speech versus non-speech discrimination, audio classification and speech recognition tasks.

### 2.3.1 Speech Versus Non-Speech Discrimination

The speech versus non-speech discrimination task [37], [4] consisted of identifying whether a given one second segment was speech or non-speech. Different SNR conditions were created by synthetically adding pink and white noises. The audio database was constituted

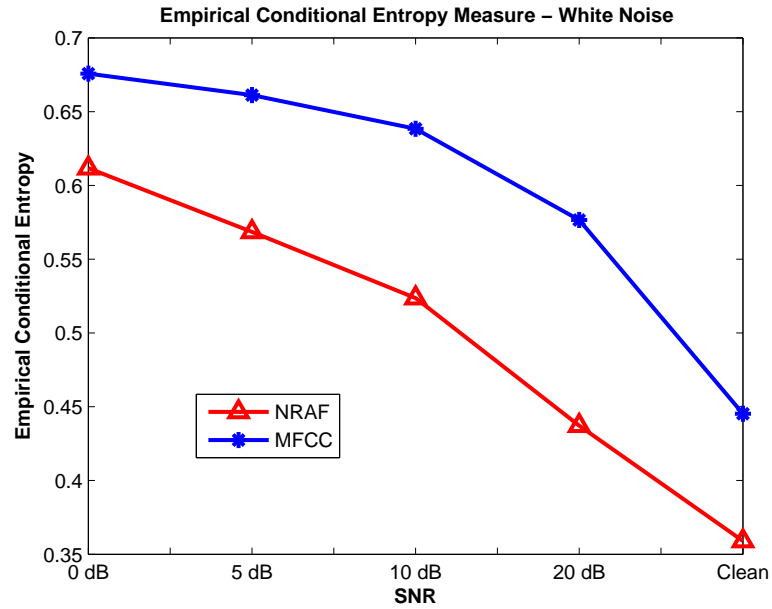


Figure 15. Figure showing the empirical conditional entropy measures for MFCC and NRAF for a 2-class, 2-cluster case. It is seen that NRAFs cluster better than MFCCs. White noise was synthetically added.

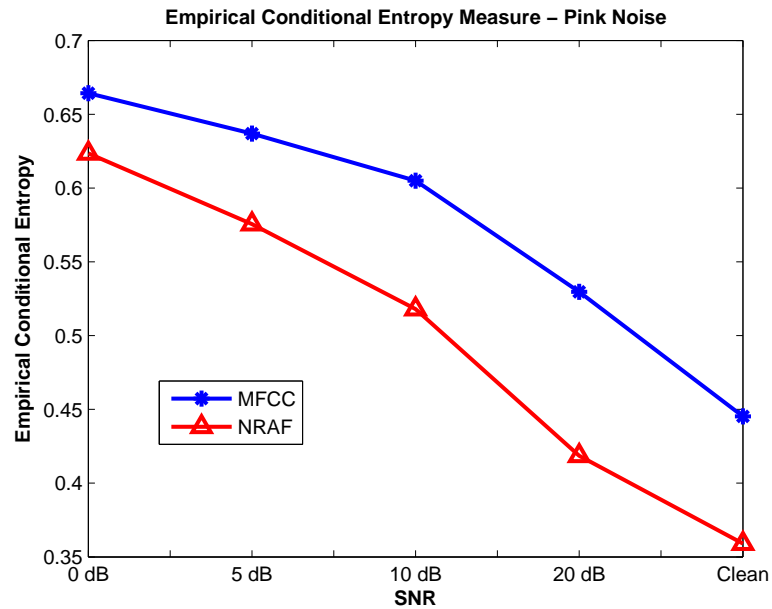


Figure 16. Figure showing the empirical conditional entropy measures for MFCC and NRAF for a 2-class, 2-cluster case. It is seen that for the task considered NRAFs are better features than MFCCs. Pink noise was synthetically added.

**Table 1. Empirical conditional entropy measures for MFCC and NRAF for a 4-class, 4-cluster case. It is seen that NRAF has better class discrimination ability.**

Empirical Conditional Entropy				
	White Noise		Pink Noise	
SNR	MFCC	NRAF	MFCC	NRAF
Clean	0.7180	0.6442	0.7180	0.6442
20 dB	0.7520	0.7288	0.7661	0.7073
10 dB	0.7929	0.7803	0.8111	0.7710
5 dB	0.8258	0.8120	0.8428	0.8080
0 dB	0.8672	0.8447	0.8755	0.8412

from five publicly available corpora. Speech samples were taken from TIMIT Acoustic-Phonetic Continuous Speech Corpus. The training set consisted of 300 segments from TIMITs training subset and the test set consisted of 150 different sentences spoken by 50 different speakers (25 male and 25 female) from TIMITs test subset. Sentences and speakers in training and test sets were different. The non-speech class consisted of 450 (300 training and 150 test) segments which included animal vocalization from BBC Sound Effects audio CD collection, music samples from RWC Genre Database [38] and Environmental sounds from Noisex and Aurora databases. Segments of one second duration were selected for training and testing. Pink and white noise were synthetically added to generate different SNRs.

Twelve MFCC's were extracted from each frame. Frame length was chosen to be 25.625 msec with frame rate of 100 Hz. For MFCC extraction, 13 linear and 27 log filters were used to group the FFT bins. The lowest frequency was chosen to be 133.33 Hz, a linear spacing of 66.66 Hz and log spacing of 1.07 was used. The code is publicly available in Slaney's *Auditory Toolbox* [39]. For NRAFs, 40 fourth-order bandpass filters (spanning the same frequency range as MFCCs) were used (but only the first 12 coefficients were used for the experiments). A Gaussian mixture model based classifier was used to predict the log-likelihood of each frame belonging to a particular class. The log-likelihoods of all frames in a segment belonging to different classes were added to form the final decision. Each segment was variance normalized and mean subtracted [40].

Tables 2-3 compare the performance of MFCC's and NRAF's. It is seen that when the added noise is white, mean and variance normalization helps in removing most of the

noise, there by resulting in similar performance for the two features (although MFCC's have higher false alarms). But in the case where pink noise is synthetically added we see that the performance of MFCC drops sharply at low SNRs while the degradation of NRAF is more graceful (see Table 3). Table 4 shows the comparison of NRAF with and without spatial derivative. As predicted the performance of NRAF with spatial derivative is better in clean conditions, but in very low SNR cases, NRAF without spatial derivative does better.

### 2.3.2 Audio Classification

Performance comparison of NRAF and MFCC for an audio classification task is presented. For the audio classification experiment, the *Phonak* database was used. It consisted of four classes: speech, music, noise, and speech in noise. In total there were 287 recordings, each of 30 second duration. There were 60 speech, 73 music, 80 noise and 74 noisy speech recordings. Speech class consisted of children, male and female speakers. It also had reverberated, fast, raised effort, dialogue and compressed speech. The different types of music consisted of classic, pop/rock, single instrument, singing and a few others such as Jazz and folk music. Noise class consisted of babble, in-car noise, traffic, industrial and other noises such as blender, shower, sink, office noises etc. Noisy speech class consisted of speech by 5 different speakers in different kinds of noise. Each recording was segmented into 1 second segments that were used for classification, there were no overlaps between training and test recordings. The mean and variance of MFCC and NRAF features over all frames of the audio segment were concatenated to form the feature vector. It has been shown that for audio classification this is a better approach as opposed to the conventional method of using MFCCs on a per frame basis [41], [26]. The results are tabulated in Table 5. It is seen that NRAFs perform better than MFCCs.

### 2.3.3 Speech Recognition

The speech recognition results for the Aurora 2 task in clean training condition are presented in Table 6 below. MFCCs were used as the baseline and were extracted using code based on the HTK toolkit frontend [42], 23 channels were used. The NRAF features were also extracted in a similar way. Thirty-two one-sixth octave filters were used for the filter-bank

**Table 2. Comparison between root compressed MFCC and NRAF. Since the added noise is white, mean and variance normalization removes most of the noise, making the performance of the two features similar. A 15 mixture GMM and 12 features were used.**

Table showing the comparison between NRAF and MFCC.		
	NRAF (root compression)	MFCC (root compression)
SNR	Overall Accuracy	Overall Accuracy
Clean	<b>99.00</b> %	98.67 %
20 dB	<b>98.67</b> %	98.33 %
10 dB	<b>98.00</b> %	97.33 %
5 dB	<b>98.00</b> %	95.33 %
0 dB	<b>95.33</b> %	92.00 %

**Table 3. Comparison between root compressed MFCC and NRAF. Pink noise was synthetically added. A 15 mixture GMM and 12 features were used.**

Table showing the comparison between NRAF and MFCC.		
	NRAF (root compression)	MFCC (root compression)
SNR	Overall Accuracy	Overall Accuracy
Clean	<b>99.00</b> %	98.67 %
20 dB	<b>98.67</b> %	98.00 %
10 dB	<b>98.33</b> %	97.67 %
5 dB	<b>97.33</b> %	94.67 %
0 dB	<b>94.67</b> %	83.33 %

**Table 4. Table showing that spatial derivative is useful in clean and low noise conditions but in high noise cases spatial derivative can hurt the robustness of the features. A 15 mixture GMM and 12 NRAF features were used. Pink noise was synthetically added.**

Table showing the comparison between NRAF with and without spatial derivative.		
	NRAF (root compression)	NRAF (root compression) (no spatial derivative)
SNR	Overall Accuracy	Overall Accuracy
Clean	<b>99.00</b> %	98.33 %
20 dB	<b>98.67</b> %	98.33 %
10 dB	<b>98.33</b> %	97.67 %
5 dB	<b>97.33</b> %	97.00 %
0 dB	94.67 %	<b>96.33</b> %



**Table 5.** Table showing performance of MFCCs and NRAFs for a four-class audio classification problem.

Performance of MFCCs and NRAFs		
Method	MFCC	NRAF
Mean-variance	85.45 %	87.57 %

implementation. The first 13 coefficients (including the zeroth coefficient) were mean subtracted and variance normalized (MVA) and delta and acceleration features were computed to form a 39-dimensional feature vector. The features were MVA processed as suggested by Chen et al. [43]. Delta and acceleration coefficients were extracted from the MVA processed static features. The zeroth coefficient was used since it is shown to respond better to MVA processing than using the log energy. Logarithmic compression was used for both feature sets. The HMM was trained with 6 Gaussian components per mixture for every state other than silence which was modeled using 12 components. The significance of the improvements is measured using a difference of proportion test and the results are tabulated in Table 7. As is evident NRAF produces significant improvements over MFCC, especially at low SNRs.

**Table 6.** Six Gaussian components per mixture was used for every state, except silence, which was modelled using 12 components. Training was carried out in clean condition.

Recognition results on Aurora 2 (clean condition)								
	Set A		Set B		Set C		Average	
	MFCC	NRAF	MFCC	NRAF	MFCC	NRAF	MFCC	NRAF
Clean	99.41	99.36	99.41	99.36	99.44	99.42	<b>99.42</b>	99.38
20 dB	97.53	97.72	97.89	97.79	97.49	97.78	97.64	<b>97.76</b>
15 dB	95.02	95.45	95.47	95.44	95.19	95.37	95.23	<b>95.42</b>
10 dB	88.55	90.16	89.83	89.80	89.15	90.90	89.17	<b>90.29</b>
5 dB	74.47	77.75	75.18	76.60	75.63	79.20	75.09	<b>77.85</b>
0 dB	47.94	53.89	48.28	51.02	48.55	56.41	48.26	<b>53.77</b>
-5 dB	20.94	24.56	21.10	22.33	22.01	26.51	21.35	<b>24.47</b>

## 2.4 Varying Time Constants in Feature Extraction

In this section, modifications are suggested to the NRAF features that improve its overall performance. In the auditory model proposed by Yang et. al [8] the lowpass filters in each channel model the inability of neurons in the central auditory system to respond to fast

**Table 7.** Table showing the significance of the improvements afforded by NRAF features. The improvement is relative to MFCC features. It is seen that at low SNRs there is significant improvement.

Difference of proportions test		
Condition	Significance Level	Differences
Clean	Not significant	-
20 dB	0.4	3.70
15 dB	0.2	4.32
10 dB	0.1	31.39
5 dB	0.005	88.14
0 dB	0.0001	149.82
-5 dB	0.002	97.56

temporal fluctuations. The time constants for all the lowpass filters are the same. However, from a signal processing perspective the combination of the rectifier and the lowpass filter is an envelope detector and it makes sense to have detectors with varying time constants in different frequency channels. Moreover, it is known that the human auditory system has greater time resolution at higher frequencies and greater frequency resolution at lower frequencies. Thus, the time constants (in *msec*) are set according to,

$$tc(i) = \frac{k_1}{f_s} \left( \frac{f_s}{2} - f_c(i) \right) + k_2 \quad (5)$$

where  $f_c(i)$  is the center frequency of the  $i^{th}$  channel and  $k_1$  and  $k_2$  are parameters used to set the range of time constants. For the speech recognition tests we set  $k_1 = 18.4$  and  $k_2 = 31$ . Figure 17 shows the variation of the time constants with center frequency. The range of the time constants was chosen empirically and is not the optimal operating point. The range was chosen because it gave good performance at all SNRs (including clean condition). There is a tradeoff between a shorter time constant, which gives better temporal resolution and better performance at low noise levels, and longer time constants that give better noise robustness. Choosing the time constants based on the ambient SNR would provide the best overall improvement (i.e. in all SNR conditions). But this requires the use of a noise estimation algorithm. Further, it should be noted that the time constants also depend on the type of audio. In a physiological system the time constants for speech are restricted both by the production and the hearing mechanism, while for music or noise this may not

be the case.

The improved noise robustness can be explained as follows, let the noisy speech signal be represented as,

$$x(t) = s(t) + n(t) \quad (6)$$

where  $s(t)$  is the speech signal and  $n(t)$  is the additive noise. For the  $i^{th}$  channel, output after the spatial derivative is,

$$(s_i(t) + n_i(t)) - (s_{i+1}(t) + n_{i+1}(t)) \quad (7)$$

assuming an acoustic signal can be expressed as  $x(t) = \sum_i e_{x_i}(t)v_{x_i}(t)$ , where  $v_{x_i}(t)$  is the modulated signal and  $e_{x_i}(t)$  is the modulating signal in the  $i^{th}$  channel, the above equation can be rewritten as,

$$(e_{s_i}(t) * v_{s_i}(t) - e_{s_{i+1}}(t) * v_{s_{i+1}}(t)) + (n_i(t) - n_{i+1}(t)) \quad (8)$$

where  $v_i(t)$  is the speech excitation and  $e_{s_i}(t)$  is the speech envelope in the  $i^{th}$  channel. The output after the envelope detector is given by,

$$(e_{s_i}(t) - e_{s_{i+1}}(t)) + (e_{n_i}(t) - e_{n_{i+1}}(t)) \quad (9)$$

where  $e_{n_i}(t)$  is the noise modulation in the  $i^{th}$  channel. If we assume the noise spectrum to be approximately flat, the noise term is dominated by the signal term. But in the general, it is possible to adjust the time constants in each channel to selectively extract the speech modulation and hence weed out the noise component, making the representation more robust to noise. Figure 18 shows the advantage of using varying time constants. As can be seen clearly, varying the time constants to suit the speech modulation reduces the effect of noise. The features extracted with varying time constants are referred to as NRAF-TC. Comparison of NRAF and NRAF-TC for the speech versus non-speech discrimination task is shown in Fig. 19.

Varying time constants improves the speech recognition performance as seen in Table 8

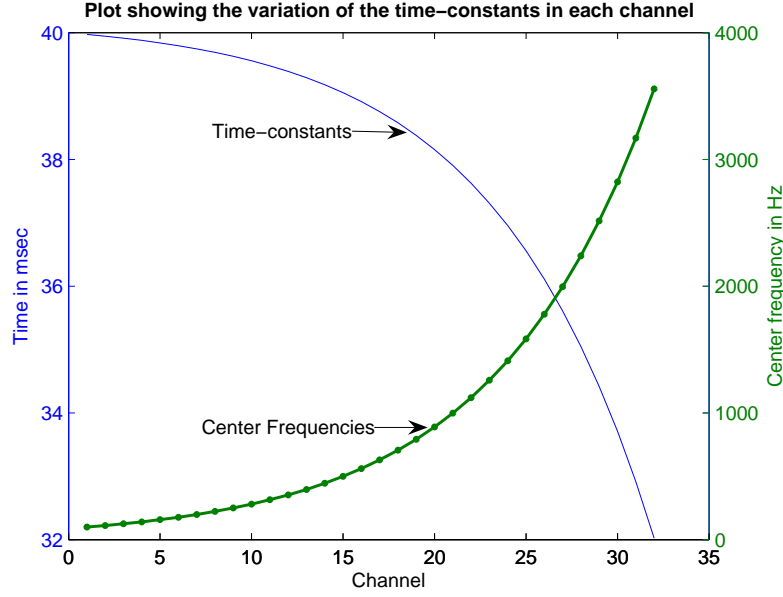


Figure 17. Figure showing the variation of the time constants with the center frequency of each channel

and Fig. 20. The significance test results are shown in Table 9. It is seen that the improvements are significant especially at medium SNRs (close to 10 dB SNR).

Table 8. Six Gaussian components per mixture was used for every state, except silence, which was modelled using 12 components. The entire training and test data was used. The increased modeling ability of the backend enables it to better fit the extra information encoded by the NRAF representation.

Recognition results on Aurora 2 (clean condition)								
	Set A		Set B		Set C		Average	
	NRAF-TC	NRAF	NRAF-TC	NRAF	NRAF-TC	NRAF	NRAF-TC	NRAF
Clean	99.38	99.36	99.38	99.36	99.36	99.42	99.37	<b>99.38</b>
20 dB	98.06	97.72	98.09	97.79	97.96	97.78	<b>98.03</b>	97.76
15 dB	96.00	95.45	96.16	95.44	95.98	95.37	<b>96.05</b>	95.42
10 dB	91.08	90.16	90.92	89.80	92.50	90.90	<b>91.50</b>	90.29
5 dB	79.07	77.75	78.31	76.60	80.76	79.20	<b>79.38</b>	77.85
0 dB	55.37	53.89	52.90	51.02	58.16	56.41	<b>55.47</b>	53.77
-5 dB	25.76	24.56	23.84	22.33	27.45	26.51	<b>25.68</b>	24.47

## 2.5 Gain Adaptation

The type and degree of compression plays an important role in the performance of MFCC and NRAF features. In this section we present a gain adaptation technique that not only improves the performance in clean conditions but also provides noise suppression in low

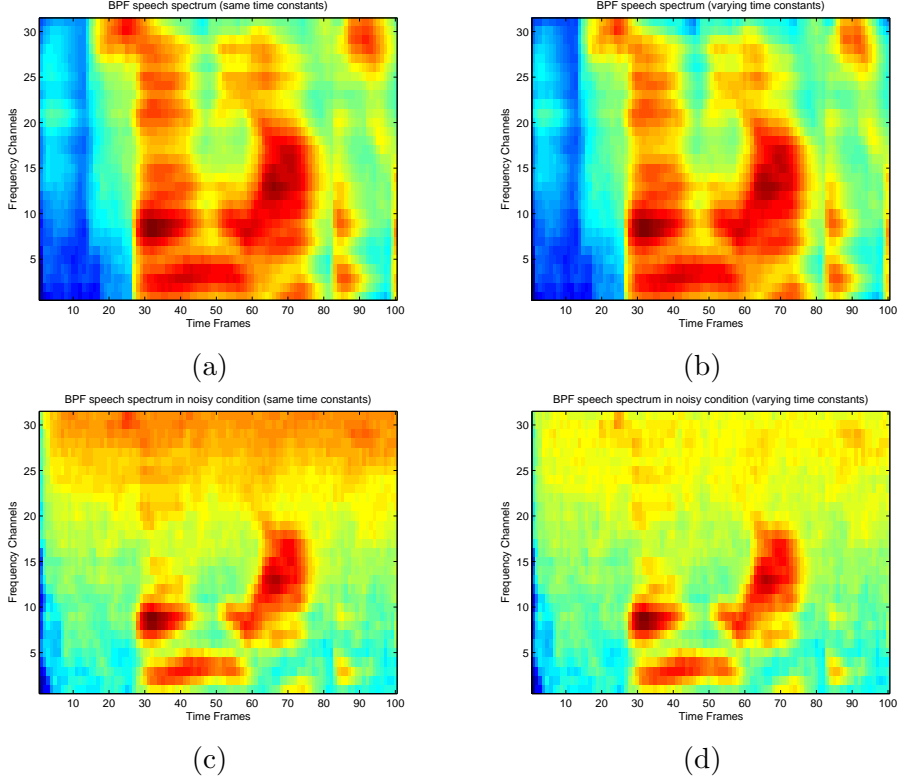


Figure 18. Speech spectrum in clean condition with a) same time constant in each channel b) varying time constants in each channel. Spectrum in noisy conditions with c) same time constant in each channel d) varying time constants in each channel. As is clear, varying time constants helps reduce the effect on noise on the speech spectrum.

SNR conditions.

### 2.5.1 Effect of Compression on Noise Robustness

One of the reasons for MFCC's poor performance is attributed to log compression [44], [45], [46]. The large negative excursions of the log function for values close to zero leads to a splattering of energy after DCT whereas root compression (expressed as  $(\cdot)^\alpha$ , with  $0 < \alpha < 1$ ) followed by DCT leads to more compaction of energy. A simple experiment was devised to show that root compression followed by DCT leads to better compaction, a full-wave rectified speech segment was amplitude compressed and transformed using DCT. Varying number of transformed coefficients were used to reconstruct the amplitude compressed signal and reconstruction error was calculated. Figure 21 shows the plot of errors for log and root compression and Table 10 shows the improvement in performance of MFCC's with root compression for the speech versus non-speech discrimination task.

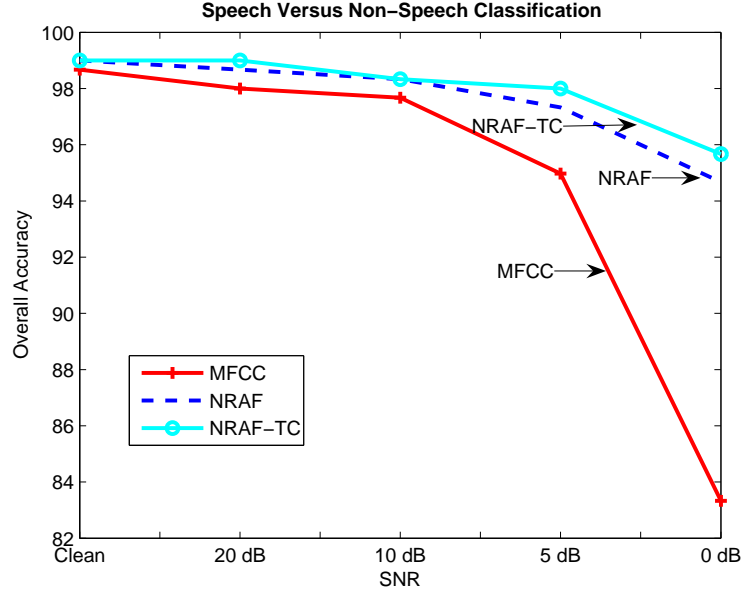


Figure 19. Figure showing the comparative performance of MFCC, NRAF, and NRAF-TC for the speech versus non-speech classification task. Different SNRs were obtained by synthetically adding pink noise. Root compression was used for all the features.

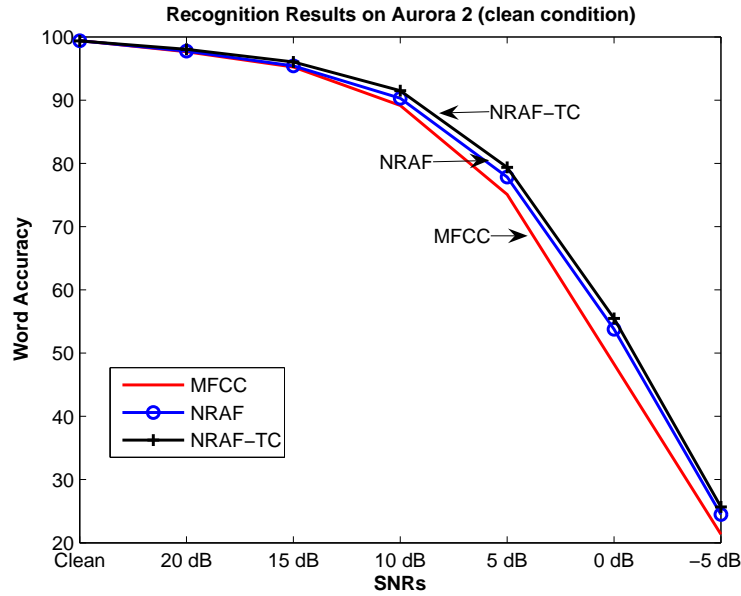


Figure 20. Figure showing the performance of MFCC, NRAF, and NRAF-TC on the Aurora 2 task. Six Gaussian mixtures were used for each state and silence was modelled using 12 component mixture.

**Table 9.** Table showing the significance of the improvements afforded by varying time constants. The improvement is relative to NRAF features. It is seen that at medium SNRs the improvement is significant.

Difference of proportions test		
Condition	Significance Level	Differences
Clean	Not significant	-
20 dB	0.4	2.94
15 dB	0.2	13.72
10 dB	0.05	38.18
5 dB	0.1	42.40
0 dB	0.1	51.49
-5 dB	0.2	29.44

**Table 10.** Table showing that root compression is better than log compression for noise robustness. A 15 mixture GMM was used and pink noise was synthetically added. The first 12 MFCC features were used.

Table showing the comparison between MFCC using root compression and MFCC using log compression.		
	MFCC (log compression)	MFCC (root compression)
SNR	Overall Accuracy	Overall Accuracy
Clean	<b>99.33</b> %	98.67 %
20 dB	95.00 %	<b>98.00</b> %
10 dB	77.67 %	<b>97.67</b> %
5 dB	65.00 %	<b>94.67</b> %
0 dB	53.00 %	<b>83.33</b> %

It is known that the type of compression performed has an effect on the robustness of features used for speech processing [45], [44], [46]. Tchorz and Kollmeier [47] have shown that adaptive gain control via the use of successive compressive loops helps noise robustness of features. A set of simple experiments were performed to study the effect of degrees of static compression on noise robustness of features. For a two-class classification problem a measure of the between-class distance and within-class scatter is studied, and it is shown that in clean conditions more compression gives better results (due to better clustering of data) but with the introduction of noise more compression leads to a drop in performance.

Between-class distance is defined as the distance between the mean of the two classes and within-class scatter is defined as the average of the distances of each data point from the mean of its class. As seen from Table 11, with increasing amount of compression the

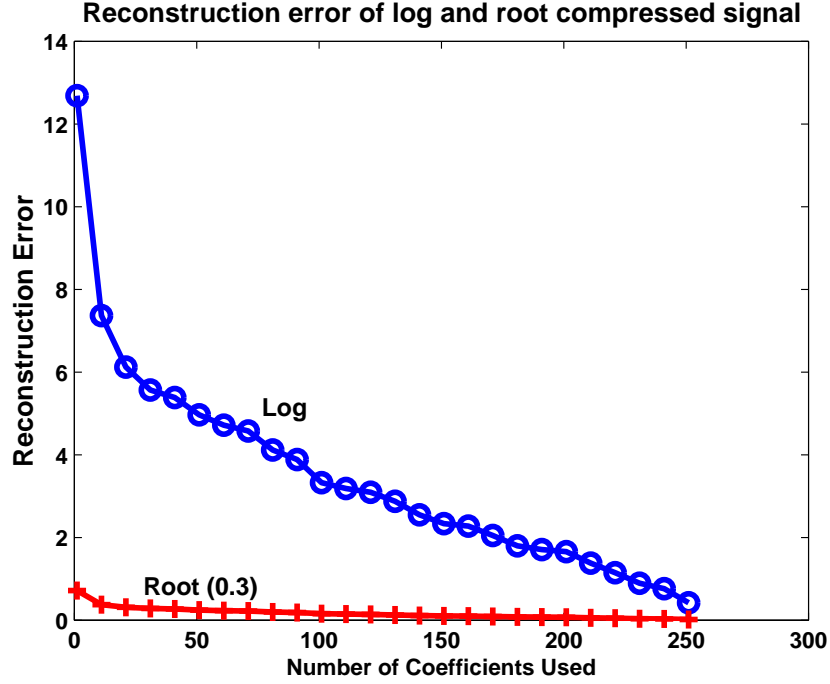


Figure 21. Figure showing that root compression followed by DCT leads to better compaction of energy. Reconstruction error is plotted as a function of number of coefficients used for the reconstruction.

between-class distance decreases. With increasing compression, the within-class scatter also decreases (this is expected since the variability in the features is reduced). In clean condition, the decrease in within-class scatter is sufficient to offset the decrease in between-class scatter but when noise is present, the decrease in within-class scatter is not enough to offset the reduction in between-class distance and as a result the overall discrimination ability (defined as the the ratio of between-class distance to within-class scatter) of the features decreases, as seen from Table 12. It is clear that a compression scheme that changes with SNR will yield the best over all performance.

It is well understood that in physiological systems outer hair cells play a very important roll in automatic gain control (AGC). AGC boosts weak signals and compresses strong signals adaptively to achieve the desired dynamic range. It seems plausible that replacing the static compression in the feature extraction process with AGC will lead to a better representation. Herein, both functionalities are combined and a gain adaptation is presented that provides time varying gain not only relative to the signal strength but is also based on



the SNR. It is shown that this kind of formulation takes a form very similar to the Wiener gain.

**Table 11. With more compression, between-class distance of the features decrease.**

Table showing degradation of between class distance				
SNR	No comp.	Root comp. (0.7)	Root comp. (0.3)	Root comp. (0.07)
Clean	3.72	3.31	2.03	0.83
10 dB	3.24	2.42	0.92	0.27

**Table 12. Table showing that smaller  $\alpha$  yields greater discrimination in clean conditions. However, in noisy conditions larger  $\alpha$  yields better class discrimination.**

Measure of discrimination ability for log and various degrees of root compression		
Compression	Clean	Noisy
Log	0.4364	0.2372
Root ( $\alpha = 0.07$ )	0.4182	0.2387
Root ( $\alpha = 0.3$ )	0.3645	0.2473
Root ( $\alpha = 0.7$ )	0.3196	0.2719

### 2.5.2 Adaptive Gain Control

The approach suggested by Anderson et al. [48], [49] is followed in non-linearly compressing the envelope in each channel during the feature extraction process.

An acoustic signal can be expressed as,

$$s(t) = \sum_i e_{s_i}(t) v_i(t) \quad (10)$$

where  $v_i(t)$  is the speech excitation and  $e_{s_i}(t)$  is the speech envelope in the  $i^{th}$  channel. The relationship between the non-linearly compressed envelope and the original envelope can be expressed as,

$$\hat{e}_{s_i}(t) = \beta e_{s_i}^\alpha(t) \quad (11)$$

$$\hat{e}_{s_i}(t) = G e_{s_i}(t) \quad (12)$$

where  $G = \beta e_{s_i}^{\alpha-1}(t)$ . Equation 11 can be re-written as,

$$\log \hat{e}_{s_i}(t) = \alpha \log e_{s_i}(t) + \log \beta \quad (13)$$

$\alpha$  and  $\beta$  are computed based on the desired range of compressed envelope. The gain function should be designed such that the maximum of the input corresponds to unity gain. The minimum of the compressed envelope is chosen to be a scaled version of the minimum of the input envelope. Thus we have:

$$\hat{e}_{s_{i\max}} = e_{s_{i\max}} \quad (14)$$

$$\hat{e}_{s_{i\min}} = K e_{s_{i\min}} \quad (15)$$

where  $K$  is a positive scaling factor. Using Eqn.(14) in Eqn.(13),

$$\log(e_{s_{i\max}}) = \alpha \log(e_{s_{i\max}}) + \log(\beta)$$

$$\log(\beta) = (1 - \alpha) \log(e_{s_{i\max}}) \quad (16)$$

$$\beta = e_{s_{i\max}}^{(1-\alpha)} \quad (17)$$

and using Eqn.(15) in Eqn.(13) and substituting for  $\log(\beta)$ ,

$$\log(K e_{s_{i\min}}) = \alpha \log(e_{s_{i\min}}) + \log(\beta)$$

$$(1 - \alpha) \log(e_{s_{i\min}}) + \log(K) = (1 - \alpha) \log(e_{s_{i\max}})$$

$$(1 - \alpha) \log\left(\frac{e_{s_{i\min}}}{e_{s_{i\max}}}\right) = -\log(K)$$

$$\alpha = 1 - \frac{\log(K)}{\log(M)} \quad (18)$$

where,  $M = \frac{e_{s_{i\max}}}{e_{s_{i\min}}}$

The gain function multiplying the signal is given by,

$$\begin{aligned}
G &= \beta e_{s_i}^{(\alpha-1)} \\
&= e_{s_{i\max}}^P e^{-P} \\
&= \left( \frac{e_{s_{i\max}}}{e_{s_i}} \right)^P
\end{aligned} \tag{19}$$

where,  $P = \frac{\log(K)}{\log(M)}$ . Since  $M \geq 1$  we have,

$$G \geq 1 \text{ when } K \geq 1 \text{ and,}$$

$$G < 1 \text{ when } 0 \leq K < 1$$

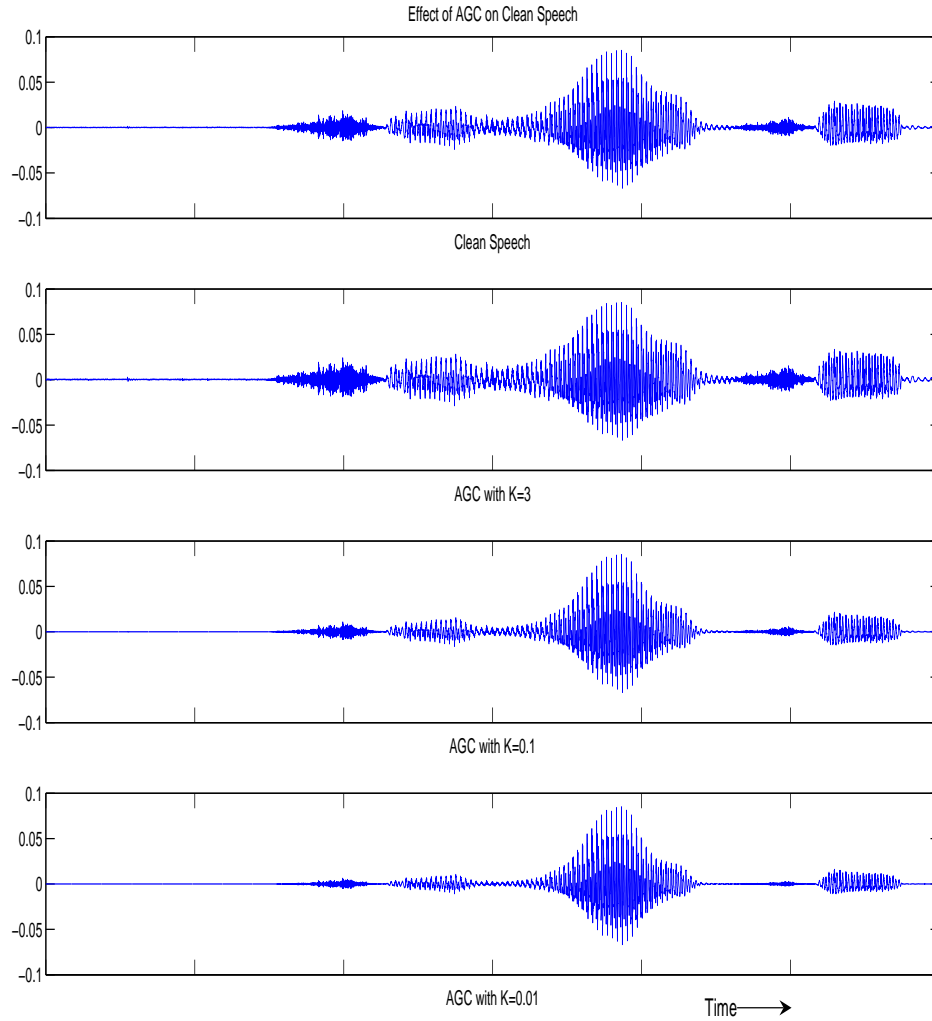
It is easily seen that by appropriately setting  $G$ , either noisy portions can be suppressed or clean portions of the signal can be boosted. Figures 22 and 23 show the effect of AGC with different values of  $K$ , on clean and noisy speech. As can be seen,  $K < 1$  suppresses most of the noise. The advantage of using this approach is that only a very rough estimate of the SNR is needed to set the value of  $K$  and hence, this technique is less sensitive to errors in estimation of the SNR.

#### 2.5.2.1 Speech Discrimination Results

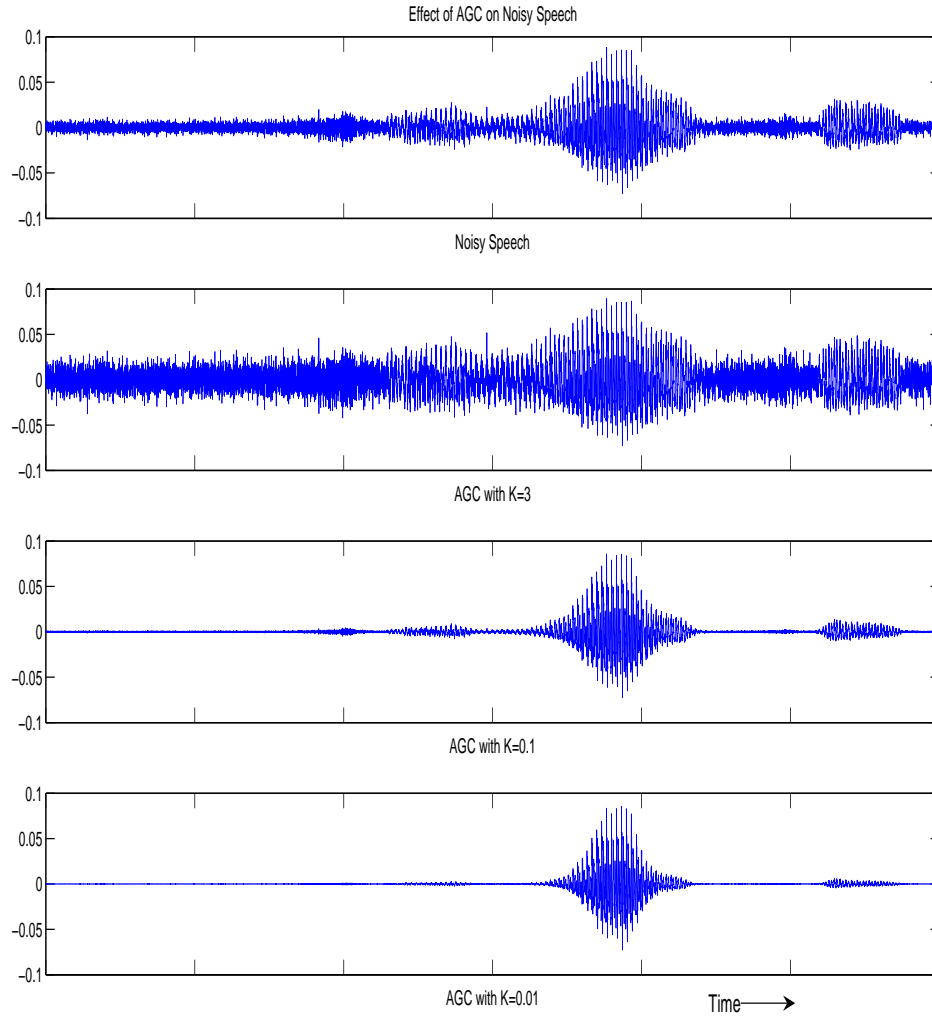
As is seen from Table 13, with  $K=0.1$  the performance of the features in noisy conditions can be improved. For  $K=0.01$ , the noise robustness of the features can be further improved but at the cost of reduced performance in high-SNR conditions. NRAF by itself performs very well at this task and hence the gains afforded by AGC is not as pronounced as in the speech recognition case.

#### 2.5.2.2 Speech Recognition Results

The speech recognition results (on a subset of the TIDIGITS database) are presented in Table 14. The training was in clean condition and different SNRs were obtained by synthetically adding white noise. White noise was used (instead of the standard Aurora 2 noise samples) since it allows the setting of a universal value of  $K$  for all channels, thus enabling one to explicitly study the effects of varying the parameter,  $K$ . HTK toolkit was used to



**Figure 22.** Figure showing the effect of AGC (with different values of  $K$ ) on clean speech. It is seen that  $K < 1$ , results in some loss of information in clean conditions, while  $K > 1$ , enhances the low energy parts of the signal.



**Figure 23.** Figure showing the effect of AGC (with different values of  $K$ ) on noisy speech. It is seen that  $K < 1$ , suppresses the noise in the signal, smaller value of  $K$  leads to more suppression.  $K > 1$ , on the other hand, tends to amplify the noisy.

**Table 13. Table showing improvement in noise robustness of features with gain adaptation. Pink noise was synthetically added to generate different noise conditions.**

Affect of AGC on speech versus non-speech classification			
Condition	NRAF (no AGC)	NRAF (with AGC, K=0.1)	NRAF (with AGC, K=0.01)
Clean	99.00	99.00	99.00
20 dB	98.67	99.00	98.67
10 dB	98.33	98.33	98.33
5 dB	97.33	97.67	97.67
0 dB	94.67	96.00	96.33

perform the training and evaluation. Twelve components per mixture for the silence model and 6 components for every other state were used. As seen with the previous task,  $K < 1$  improves the performance of the features in low SNR conditions. The performance in clean can be improved by using a value of  $K > 1$ . It is apparent that using an algorithm to compute the SNR in each channel to set the value of K would provide better performance in all conditions.

**Table 14. Affect of AGC (with different values of K) on the noise robustness of features. White noise was synthetically added to obtain different SNRs.**

Recognition results				
	NRAF (no AGC)	NRAF (with AGC, K=0.05)	NRAF (with AGC, K=0.01)	NRAF (with AGC, K=1.5)
Clean	99.51	99.48	99.42	99.54
20 dB	97.73	98.13	98.10	97.67
15 dB	95.73	96.50	96.56	95.61
10 dB	90.76	92.39	92.54	90.70
5 dB	79.71	83.02	83.79	79.09
0 dB	59.69	64.54	65.67	58.21
-5 dB	37.80	41.51	42.19	37.21

## 2.6 Design Notes

There is some flexibility in the design choices pertaining to the NRAF features, for example, the number of bandpass filters to use, the order of the filters, filter spacing, amount of smoothing, and decimation factor. Great care has to be taken while choosing the filter order and the filter spacing, since the spatial derivative stage tends to tighten the filter response.

These design choices also greatly depend on the task at hand. For audio classification tasks, where very fine granularity is not required (as opposed to speech recognition) better robustness can be achieved in high noise conditions by dropping the spatial derivative stage and using a higher order bandpass filter (to limit frequency spreading). The amount of smoothing is also dependent on the task. For the speech recognition task, it was found that smoothing for 32 msec worked well. However, the amount of smoothing is also a function of the SNR, in general more smoothing leads to more robustness but reduced performance in clean condition. A possible avenue for further research could be non-uniform sampling of frames. It would make sense to have less frequent sampling for channels (and temporal instances) where the smoothing time constants are large.

## 2.7 Summary

In this chapter a new feature extraction method was presented that improves the noise robustness of MFCCs. In the past features based on detailed models of the human auditory system have not been able to outperform MFCCs in all SNR conditions, this could be attributed in part to the limitations of the back-end used for the recognition. The approach taken in this work was to learn from biology and incorporate the principles of physiological processing as modifications to the MFCC feature extraction process. These modifications were done keeping in mind the computational overhead as well as the limitations of the back-end recognizer. For instance, much like the MFCC features, the NRAF features were made to be asymptotically uncorrelated, to take advantage of the diagonal covariances used in the state-of-the-art speech recognition back-ends. One of the challenges arising while comparing features is that, in order to ensure fairness of comparison, it is desired that the back-end recognizer be similar in both cases, but this leads to issues when the compared feature sets have different characteristics. As an example filter-bank energy features, which are correlated, will not perform well with back-ends employing GMM's with diagonal covariances. Thus a fair comparison can be achieved only by designing a back-end processor suited to the features, much like HMM-based recognizers are to MFCC's. In this work this issue was avoided to a large extent by consciously modeling the new features to have characteristics similar to MFCCs. The potential of physiologically inspired features is huge, but

in order reap the benefits of these features, the feature characteristics have to be studied carefully and a back-end recognizer built to fit the characteristics well. It is speculated that classifiers that can work directly with high dimensional and sparse representations might lead to better exploitation of biologically inspired features. The next chapter is concerned with presenting a few ideas that could be used to exploit the benefits afforded by the high-dimensional cortical representation presented in Chapter 1.



## CHAPTER 3

### PROCESSING SECONDARY FEATURES

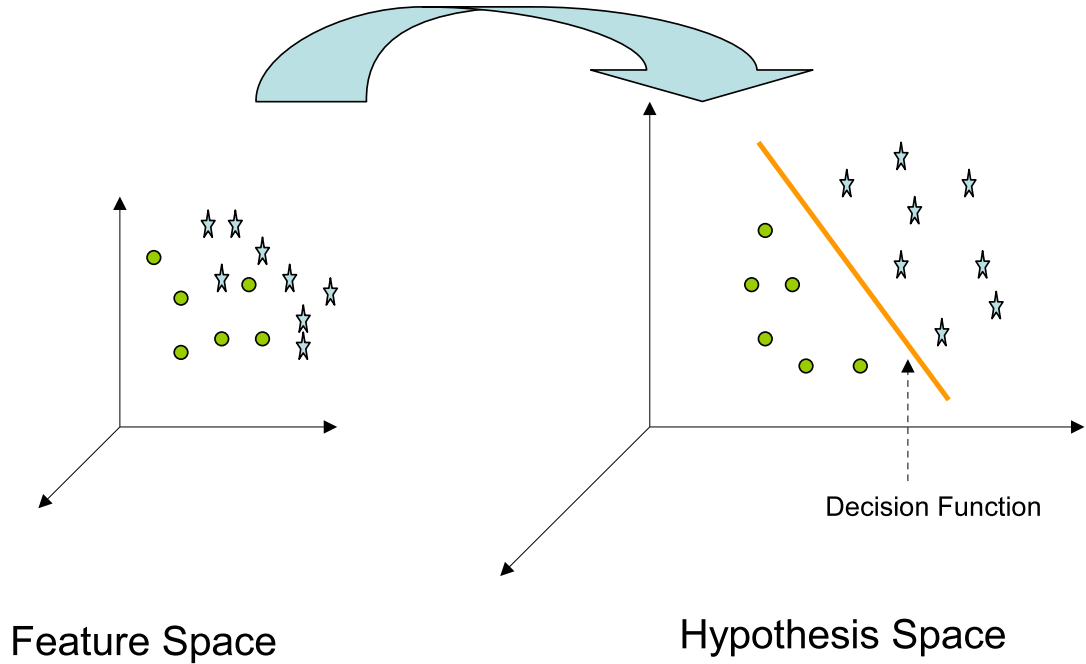
Features derived from modeling the auditory pathway beyond the peripheral auditory system tend to have characteristics different from that of the “primary” features. Feature sets such as the spectro-temporal modulation features (STRF) [41], [4] extracted from a model of the human primary auditory cortex exist in a high dimensional feature space and efficient methods have to be devised to process them. Herein, several approaches are suggested that use existing tools from the machine learning community to exploit the “dimension expanded” representation. The methods used here were chosen due to two primary reasons: they could be easily modified to suit the objective of utilizing the “secondary” features effectively, and it is possible to draw parallels between the chosen methods and biological processing. Two approaches are possible to process these features, namely, reducing these features dimensionally and making them more compact to allow their incorporation into standard classifiers (such as GMM-based classifiers), or devising methods to work with these features directly. Nima et al. [4] reduce the STRF features using a multi-linear SVD. Jeon et al. [50] propose a feature set similar to STRFs (based on the model proposed by Wang et al. [3]) and use “low-variance” and “high-activation” filters to weed out uninteresting features, they then perform a “neuronal reduction” (which is simply finding the mean feature of local clusters) followed by principal component analysis (PCA) for the dimensionality reduction. In this work, a dimensionality reduction technique is presented that learns the input feature space and provides dimensionality reduction without transforming the features. Kleinschmidt [51], proposed a method for extracting spectro-temporal modulation features and used feature finding neural networks [52] to perform the back-end recognition. Herein, methods based on boosting and SVM are presented that can directly work with the high-dimensional features. The methods presented here are only the first step towards using machine learning techniques for processing “secondary” features, but the results certainly warrant a closer examination. The rest of the chapter is organized as follows, Section 3.1 describes the AdaBoost classifier and presents a multi-class AdaBoost classifier that uses a measure of “margin” in combining the outputs of multiple binary classifiers. Section 3.2

presents a generative AdaBoost classifier that relies on boosting density estimates to return a likelihood estimate for each class, thus making it easily scalable to large number of classes. Section 3.3 presents a new multi-kernel SVM that improves both the accuracy and the generalization of a single kernel SVM. Section 3.4 presents the new dimensionality reduction technique and Section 3.5 outlines some of the design issues pertaining to the presented algorithms, followed by a summary of the chapter.

### 3.1 AdaBoost

It is well known that humans represent audio information in a sparse manner and “task-based” relevant information is extracted for back-end processing. In this section we present a machine learning technique that performs feature selection based on the target class. Boosting is a method of improving the accuracy of a learning algorithm. It can be thought of as a method to combine multiple simple rules to form a strong rule. In machine learning terms, the entity producing the simple rules is called the “weak” or “base” learner. The boosting algorithm repeatedly calls the weak learner, each time feeding the weak learner a subset of training examples [53]. One of the questions that needs to be answered pertains to the choice of base learners to be used. Popular choices include decision stumps, decision trees, and nearest neighbor classifiers. Many flavors of boosting exist and AdaBoost [54] is one of the most popular algorithms. The AdaBoost algorithm calls a weak learning algorithm repeatedly for a specified number of iterations and at each iteration it increases the “cost” of mistakes made at the previous iteration thereby forcing the weak learner to focus on the mistakes made in the previous iteration. It was modified by Tieu and Viola [55] to work as a feature selector by restricting the weak learner to work with just one feature at a time. AdaBoost can be thought of as transforming the feature space to a hypothesis space by the use of weak learners and performing the classification in the hypothesis space (see Fig. 24).

The binary AdaBoost algorithm is shown in Table 15. Consider a set of inputs  $(x_i, y_i)$  where  $x_i$  is the feature vector and  $y_i$  is the true class label. For the purposes of the algorithm development in this section, it is assumed that  $y_i \in \{1,0\}$ ,  $y_i=1$  if the example  $x_i$  belongs to Class 1 and  $y_i=0$  if example  $x_i$  belongs to Class 0. If the weak learner is restricted to work



**Figure 24.** Figure showing the concept of AdaBoost. Although the decision function is linear it takes advantage of the fact that the mapping to a suitable hypothesis space makes the data linearly separable (to a large extent).

with one feature, then at each iteration the AdaBoost algorithm picks a different feature (in reality there is no guarantee that the picked feature will be a new feature, but it will be guaranteed to be the one that produces the minimum weighted error). The prediction returned by feature  $j$  for the example  $x_i$ , is referred to as the “hypothesis”,  $h_{i,j}$ . Feeding the weak learner with a different subset of training examples at each iteration can be equivalently accomplished by maintaining a distribution over the training set that changes with every iteration. The initial distribution is set to be uniform over the entire training set. At the first iteration the AdaBoost algorithm finds the feature that makes the smallest weighted error over the training set. If the hypothesis is the same as the true class label the error is 0, otherwise it is 1. The error made on an example is weighted by the weight given to that example (determined according to the distribution). The algorithm then reweights the examples, giving a higher weight to examples misclassified by the previous feature. Thus the next feature picked learns the mistakes of the previous feature and so on. At each iteration the feature that is selected is assigned a confidence measure based on its weighted

error. The final classification is given by:

$$H(x) = \begin{cases} 1, & \text{if } \sum_1^T h_t \alpha_t \geq \frac{1}{2} \sum_1^T \alpha_t, \\ 0, & \text{otherwise} \end{cases}$$

where the confidence measure,  $\alpha_t = \log \frac{1}{\beta_t}$ .

In this research the a very simple distance metric was used as the weak learner. This was done partly, keeping the low-power hardware implementation in mind (treated in detail in Chapter 4). Thus, the hypothesis is defined as:

$$h_{ij} = 2 - \operatorname{argmin}_{k \in \{1,2\}} (|x_i(j) - \mu_k(j)|)$$

where  $\mu_k(j)$  is the mean of feature  $j$  for class  $k$ .

**Table 15. The AdaBoost algorithm.**

<p><b>Input</b></p> <ol style="list-style-type: none"> <li>1. <math>N</math> training examples  <math>(x_1, y_1), \dots, (x_N, y_N)</math>  with <math>y_i = 1</math> or <math>0</math>, the true class of <math>x_i</math>,</li> <li>2. <math>T</math>, the number of iterations</li> <li>3. <math>h_{i,j}</math>, hypothesis for example <math>x_i</math> based on feature <math>j</math></li> </ol> <p><b>Initialize</b> the weights <math>w_{1,i} = 1/N</math>  <b>Do for</b> <math>t = 1 : T</math></p> <ol style="list-style-type: none"> <li>1. For all features calculate the error,  <math display="block">\epsilon_j = \sum_{i=1}^N w_{t,i} D</math> where <math>D = 0</math> if <math>h_{i,j} = \text{true class of } x_i</math>, else <math>D = 1</math></li> <li>2. Choose the feature which corresponds to the minimum error <math>\epsilon_t</math>. <math>h_t</math> is the hypothesis that corresponds to <math>\epsilon_t</math>.</li> <li>3. Update weights,  <math display="block">w_{t+1,i} = w_{t,i} \beta_t^{e_i},</math> where <math>e_i = 1</math> or <math>0</math> for <math>x_i</math> classified correctly or incorrectly respectively, and  <math display="block">\beta_t = \frac{\epsilon_t}{1 - \epsilon_t}</math></li> <li>4. Normalize the weights,  <math display="block">w_{t+1,i} \leftarrow \frac{w_{t+1,i}}{\sum_{k=1}^N w_{t+1,k}}</math></li> </ol>
--

There has been previous work on using AdaBoost for a multi-class problem, the AdaBoost.M1 [56] is a simple extension of the binary AdaBoost with the hypothesis returning

one of the  $k$  labels. However this algorithm requires that the error of the weak learner be less than 0.5 which is not easy to attain when the number of classes,  $k > 2$ . The AdaBoost.M2 [56] requires the weak hypothesis to return a vector that indicates its belief in the example belonging to a particular label. Each component belongs to  $[0,1]$  and a number close to 1 indicates high degree of belief. The weak learner is required to do well with respect to a pseudo-loss function that is supplied to it by the boosting algorithm. The loss function varies from example to example and allows the boosting algorithm to force the weak learner to concentrate on harder examples. The multi-class AdaBoost classifier presented here uses multiple 1-versus-1 binary AdaBoost classifiers, and combines their outputs using majority voting. Majority voting usually results in “deadlocks” i.e. two or more classes get the same number of votes. Two methods are proposed to break the deadlocks, in the first method, the class with the highest normalized measure of margin (see Appendix B for details) is declared the winner. The second approach uses a GMM-based classifier to break any deadlocks resulting from majority voting. Further, a multi-class AdaBoost classifier consisting of 1-versus-rest binary AdaBoost classifiers is presented, the deadlocks in this case are resolved using a GMM-based classifier. The advantage of multi-class AdaBoost classifiers of the type presented in this section is that they allow the use of very simple weak learners.

Viola et al. [57] showed the benefit of using a cascade structure for a face recognition problem. The basic idea is to have simple classifiers upfront that remove the “easy” examples and pass on the “difficult” examples to the next classifier in the cascade, which is more complex than the previous one. Here a similar approach was used to build a cascade of AdaBoost-based classifiers for audio classification. The first stage is a simple classifier that only rejects examples that are not even close to the target class. The surviving examples are passed on to the next stage that uses more number of features to form a more complex classifier and so on. The advantage of such a structure is that more computational power (by use of complex classifiers) is spent on harder examples and the generalization is improved since the complex classifier works on a smaller subset of the training data.

A set of experiments were designed to show the usefulness of AdaBoost in handling high-dimensional features, and to demonstrate the advantages of the cascade structure. In

the following experiments, in addition to the NRAF and STRF features described earlier, 6 simple features (SF) are also used (this was mainly to allow the construction of the initial stage of the cascade). The six SF were: volume, defined as the root mean square of the sample values in the frame, volume distribution ratio (VDR) [58] which is the difference of maximum and minimum values of the volume relative to the maximum value, fluctuation, which is the ratio of variance to the mean of the signal envelope, and the width, symmetry and skewness extracted from the amplitude histogram of the signal [59]. In these set of experiments 128 BPFs were used for the frequency decomposition while extracting the NRAF features. This was done mainly to use the same auditory spectrum for computing NRAF and STRF features (the original implementation of STRF features [4] used 128 filters).

In the first set of experiments the SF and NRAF features were used to test 3 different versions of the multi-class AdaBoost algorithm. The multi-class AdaBoost algorithm using 1-versus-1 binary classifiers and normalized measure of margin to resolve deadlocks is referred to as 1vs1-a, while the version using a GMM-based classifier to resolve deadlocks is referred to as 1vs1-b. The classifier that works by dividing the multi-class problem into binary problems of 1-versus-rest type is referred to as 1vsRest (again the deadlocks are resolved using GMM-based classifiers). Euclidean distance from mean was used as the weak hypothesis and the implementation details can be found in [60]. The results are tabulated in Table 16. In the 1vs1-a algorithm the number of deadlocks were 16 and only 7 of these were correctly resolved. The use of GMMs in 1vs1-b to resolve the deadlocks improves the accuracy by increasing the number of deadlocks resolved to 14. In the 1vsRest algorithm there are 215 deadlocks and GMM is able to resolve 171 of them. A better deadlock resolving method could further improve the accuracy of the 1vsRest approach.

The second experiment was aimed at showing that the AdaBoost classifier can be improved by using the cascade structure. The 6 simple features were used to construct a threshold based initial classifier. The initial classifier was designed to have nearly 50 % false positive and very low false negative. Two additional stages were designed. The second stage consisted of 30 features that were selected using AdaBoost. The last stage consisted of 60 features picked by AdaBoost and was trained on examples that got through the second

**Table 16.** Table showing performance of a single stage “one versus one” AdaBoost classifier and “one versus rest” AdaBoost classifier using SF and NRAF features. 1vs1-b refers to the case where GMM is used to break the deadlock.

Performance Comparison different multi-class AdaBoost-based classifiers	
Classifier	Percentage Correct
1vs1-a	93.06 %
1vs1-b	93.68 %
1vsRest	93.95 %

stage as well as some negative examples from the initial mix. As can be seen from Table 17 the cascade structure provides substantial improvement over the single stage AdaBoost-based classifier. It is interesting to note that the cascade structure not only increases the accuracy but also gives a huge gain in terms of computation.

**Table 17.** Table showing performance of single stage AdaBoost and cascade AdaBoost using SF and NRAF features.

Performance Comparison of GMM and AdaBoost based classifiers	
Classifier	Percentage Correct
Single stage AdaBoost	93.06 %
Cascade AdaBoost	95.55 %

The third set of experiments were aimed at improving the accuracy of the classifiers by incorporating the cortical features (i.e. STRF). STRF was collapsed across the time axis to yield a 6144-dimensional feature vector. A single stage AdaBoost-based classifier was used to work with the entire set of features (SF + NRAF + STRF). The classifier was boosted for 200 rounds and the best accuracy was achieved at round 142. Next, a two-stage cascade classifier was designed to work with the entire feature set. The first stage was the initial classifier used in the second experiment and the second stage used the 142 features picked by AdaBoost. Figure 25 shows the improvement in accuracy of a single-stage AdaBoost-based classifier due to the addition of STRF features, as a function of rounds of boosting. As can be seen from Table 18 the STRF features increase the accuracy of both the single-stage and cascade classifier. The cascade classifier does not provide a significant improvement

over the single-stage classifier in accuracy due to the fact that the maximum discrimination ability of the features for the given database was more or less reached.

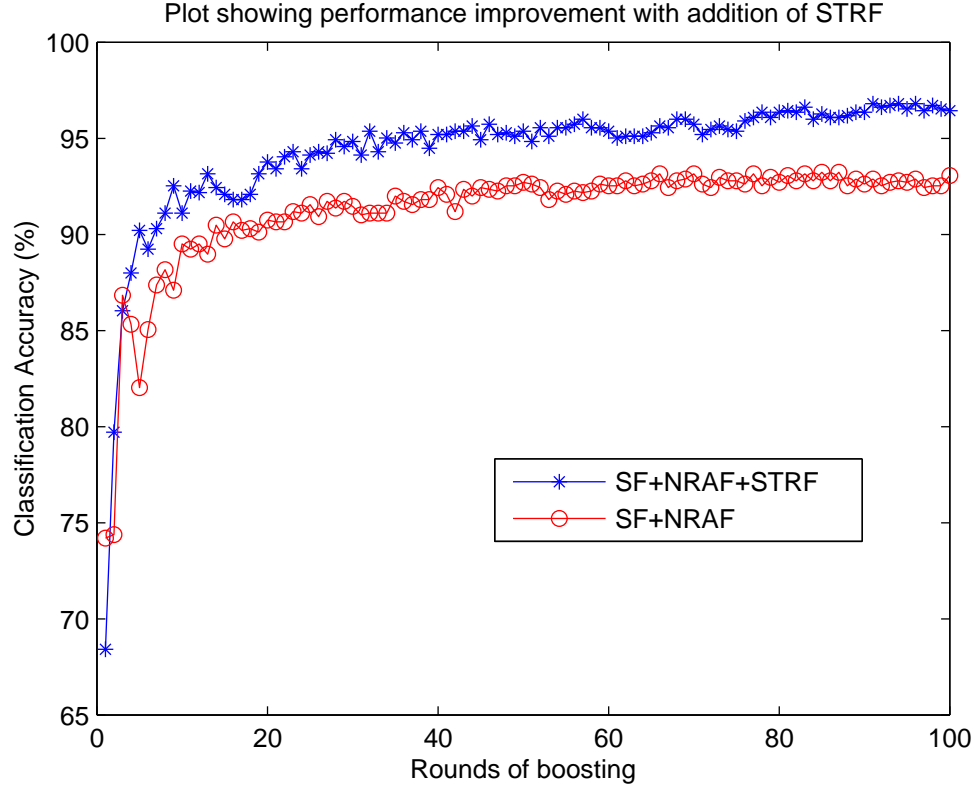


Figure 25. Figure showing the improvement in performance of single stage AdaBoost-based classifier with the addition of cortical features (STRF).

Table 18. Table showing performance of single stage AdaBoost and cascade AdaBoost using SF, NRAF and STRF features.

Performance Comparison of AdaBoost-based classifiers	
Classifier	Percentage Correct
Single Stage AdaBoost	97.68 %
Cascade AdaBoost	97.77 %

The above experiments serve to show that AdaBoost-based classifiers can be used to process the “secondary” features efficiently. These classifiers are also very effective in working with different feature sets and are not tied down by feature specific characteristics. The experiments also demonstrate the usefulness of the cascade-of-classifiers approach.



### 3.2 Generative AdaBoost

Binary classifiers such as AdaBoost and SVMs are able to effectively work with high dimensional data but their usefulness is restricted by the fact that these methods being discriminative in nature do not scale well to large number of classes. Further, it is likely that physiological systems make “soft” decisions as opposed to “hard” decisions, allowing for the possibility of a multi-label scenario. This motivates the need to have a classifier that provides likelihood measures for each class. Also, state-of-the-art methods for speech recognition rely on a likelihood measure of the observation, which cannot be provided directly by AdaBoost or SVM. Although methods have been suggested to convert the “margins” provided by these classifiers to probability values [61], it would be desirable to have these classifiers provide a likelihood measure directly.

In this section the notion of boosting density estimates is used to build a generative AdaBoost classifier that can return a likelihood measure for each class model, thus making it easily scalable to a large number of classes. Given a current density estimate  $F_t$ , boosting density is concerned with finding an estimate  $h_t$  and mixing weight  $\alpha$ , to form a new estimate  $F_{t+1} = (1 - \alpha)F_t + \alpha h_t$ . In general, the improvement need not be monotonic.

Previous work in boosting density estimate has been reported in [63], [62] and [64] and references therein. However this work differs in the way the data is transformed before re-estimation of density at each stage. Modifications to the methods suggested in [62] are explored from an information theoretic viewpoint. Further, the stagewise risk minimization suggested in [64] is used to obtain a method for computing the mixing weights based on minimizing the  $L_2$  norm. Finally an approach based on the symmetric Kullback-Liebler (KL) divergence is proposed that seems to work well in practical situations.

Boosting density estimates usually refers to the method of combining different base estimators (trained on differently weighted data). Herein, attention is focused on improving the estimate of a single base estimator by proper transformation of the data. This would lead to a solution with lower variance. Data transformation is achieved by using principal component analysis (PCA) on the weighted data. Essentially, data points not modeled well at the previous iteration get a higher weight at the current iteration.

### 3.2.1 Boosting Density Estimation

Given a weak learner,  $h(t)$  that gives an estimate of the underlying density, the boosting algorithm proposed by Rosset and Segal [62] is shown in Table 19.

**Table 19. AdaBoost based algorithm for boosting density estimates, as proposed by Rosset et al. [62].**

1. Set $F_0(z)$ to uniform over the example domain
2. For $t = 1:T$
a) Set $w_t = \frac{1}{F_{t-1}(z)}$
b) Find $h_t$ to maximize $\sum_i w_i \log(h_t(z_i))$
c) If $\sum_i w_i h_t(z_i) \leq n$ , break
d) Find $\alpha_t = \operatorname{argmin}_{\alpha} \sum_i -\log((1 - \alpha)F_{t-1}(z_i) + \alpha h_t(z_i))$
e) Set $F_t = (1 - \alpha_t)F_{t-1}(z_i) + \alpha h_t(z_i)$
3. Output the final $F_T$

In step 2(b) the weak learner is chosen to maximize the sum of weighted log probabilities instead of sum of weighted probabilities because the latter case amounts to maximization subject to a  $L_1$  constraint (since the pdf's sum to 1). This would lead to the weak learner concentrating all the probability mass on the example with the highest weight. Maximizing the weighted log probabilities is equivalent to maximizing the weighted probabilities subject to the  $L_2$  constraint [62]. In the AdaBoost classifier algorithm this issue is avoided since calculating the 1-0 hypothesis entails using the Chebychev norm. In the following sections, different methods to compute the mixing weights are explored.

#### 3.2.1.1 Minimizing Inefficiency

From information theory we know that, for a random variable  $X$  drawn from distributions  $P$  or  $Q$ , that have probabilities  $\lambda$  and  $(1 - \lambda)$  respectively, the information gain by knowing which distribution  $X$  came from is given by the so called  $\lambda$  divergence,

$$\begin{aligned}
 D_{\lambda} = & \lambda D_{KL}(P || (\lambda P + (1 - \lambda)Q)) \\
 & + (1 - \lambda) D_{KL}(Q || (\lambda P + (1 - \lambda)Q))
 \end{aligned} \tag{20}$$

where,  $D_{KL}(P || Q)$  is the KL divergence,

$$D_{KL}(P||Q) = \sum_i P(x_i) \log \frac{P(x_i)}{Q(x_i)} \quad (21)$$

KL divergence can be thought of as the inefficiency in using  $Q$  as the distribution when the true distribution is  $P$ . The  $\lambda$  divergence provides a natural way of computing the mixing weights. If  $h_t$  is the new “weak” estimate and  $F_{t-1}$  is the current estimate, then it could be claimed that the mixing weight should be the one that minimizes the inefficiency,

$$\alpha_{opt} = \arg \min_{\alpha} (D_{\lambda}) \quad (22)$$

Minimizing  $D_{\lambda}$  is equivalent to maximizing a weighted log-likelihood function. This can be proved as follows, Eqn. 20 is the same as,

$$D_{\lambda} = \lambda D_{KL}(P||Z) + (1 - \lambda) D_{KL}(Q||Z) \quad (23)$$

where  $Z = \lambda P + (1 - \lambda)Q$ . Equation 23 can be re-written as,

$$\begin{aligned} D_{\lambda} &= \lambda (E_P[\log P] - E_P[\log Z]) \\ &\quad + (1 - \lambda) (E_Q[\log Q] - E_Q[\log Z]) \end{aligned} \quad (24)$$

where  $E_X[Y]$  is the expectation of  $Y$  w.r.t the distribution  $X$ . Approximating the expectations with averages,

$$\begin{aligned} D_{\lambda} &\approx \lambda \left( \frac{1}{n} \sum_i \log P(x_i) - \frac{1}{n} \sum_i \log Z(x_i) \right) \\ &\quad + (1 - \lambda) \left( \frac{1}{n} \sum_i \log Q(x_i) - \frac{1}{n} \sum_i \log Z(x_i) \right) \\ D_{\lambda} &\approx \lambda \left( \frac{1}{n} \sum_i \log \frac{P(x_i)}{Q(x_i)} \right) \\ &\quad + \frac{1}{n} \sum_i \log Q(x_i) - \frac{1}{n} \sum_i \log Z(x_i) \end{aligned} \quad (25)$$

From Eqn. 25, minimizing  $D_\lambda$  is the same as minimizing,

$$-\left(\frac{\lambda}{n} \sum_i \log \frac{Q(x_i)}{P(x_i)} + \frac{1}{n} \sum_i \log Z(x_i)\right) \quad (26)$$

which is equivalent to,

$$\max \sum_i \log(k_i Z(x_i)) \quad (27)$$

where  $k_i = (\frac{Q(x_i)}{P(x_i)})^\lambda$ . Thus, minimizing  $D_\lambda$  is equivalent to maximizing a weighted log-likelihood function.

### 3.2.1.2 Stopping Criterion

In order to find a stopping criterion for the algorithm, the approach presented in [62] is pursued. The algorithm has to terminate when the loss function ceases to decrease further with the addition of a new estimate i.e. the derivative of  $-\sum_i \log(k_i Z(x_i))$  with respect to  $\alpha$  ceases to be negative:

$$\begin{aligned} \frac{\partial \sum_i -\log(k_i((1-\alpha)F_{t-1}(x_i) + \alpha h(x_i)))}{\partial \alpha} \Big|_{\alpha=0} \\ = n - \sum_i \frac{h(x_i)}{F_{t-1}(x_i)} \geq 0 \end{aligned} \quad (28)$$

Thus, the algorithm terminates when  $\sum_i \frac{h(x_i)}{F_{t-1}(x_i)} \leq n$ .

## 3.2.2 Minimizing $L_2$ norm

Instead of maximizing the log-likelihood as in [62], Klemela [64] suggested minimizing the  $L_2$  norm. He uses fixed mixing weights and the problem is setup as a stagewise minimization of empirical risk (as opposed to the boosting problem that minimizes both with respect to the estimator and the mixing weights). In the formulation proposed in this work, since only one estimator is used, the problem is essentially that of finding the mixing weights alone. Thus the problem could be solved as a stagewise minimization of empirical risk with respect to the mixing weights. A second order approximation is used in order to solve for the weights.

Let  $f$  be the true density and  $\hat{f}_t$ , the estimate at iteration  $t$ . The aim is to minimize the  $L_2$  distance between  $f$  and  $\hat{f}_t$ :

$$\min \quad ||f - \hat{f}_t||_2^2$$

This can be equivalently written as:

$$\min \quad ||f - \hat{f}_t||_2^2 \quad - \quad ||f||_2^2$$

or,

$$\min \quad \frac{-2}{n} \sum_i \hat{f}_t(x_i) + ||\hat{f}_t||_2^2 \quad (29)$$

If the loss function is defined to be  $\gamma(k, x) = -2k(x) + ||k||_2^2$ . Then the above minimization problem (Eqn. 29) becomes an empirical risk minimization problem. This is essentially the formulation in [64]. Approximating the empirical risk to the second order using Taylor expansion around  $\hat{f}_{t-1}$ ,

$$\begin{aligned} \gamma_n(\hat{f}_t) &= \gamma_n(\hat{f}_{t-1}) + \frac{\partial}{\partial \hat{f}_{t-1}}(\gamma(\hat{f}_{t-1}))(\alpha(h_t - \hat{f}_{t-1})) \\ &\quad + \frac{\partial^2}{\partial \hat{f}_{t-1}^2}(\gamma_n(\hat{f}_{t-1}))(\alpha(h_t - \hat{f}_{t-1}))^2 \\ &= \gamma_n(\hat{f}_{t-1}) - \frac{2}{n} \sum_i (h_t(x_i) - \hat{f}_{t-1}(x_i)) \\ &\quad + 2\alpha \sum_i (\hat{f}_{t-1}(x_i)(h_t(x_i) - \hat{f}_{t-1}(x_i))) \\ &\quad + 2\alpha^2 \sum_i (h_t(x_i) - \hat{f}_{t-1}(x_i))^2 \end{aligned} \quad (30)$$

The  $\alpha$  that minimizes the empirical risk can be found by differentiating Eqn. 30 w.r.t  $\alpha$  and equating to zero. Omitting  $x_i$  for convenience of writing, the mixing weight is given by,

$$\begin{aligned} -\frac{2}{n} \sum (h_t - \hat{f}_{t-1}) + 2 \sum \hat{f}_t(h_t - \hat{f}_{t-1}) \\ + 4\alpha \sum (h_t - \hat{f}_{t-1}) = 0 \end{aligned} \quad (31)$$

or,

$$\alpha_{opt} = \frac{\frac{1}{n} \sum (h_t - \hat{f}_{t-1}) - \sum \hat{f}_t (h_t - \hat{f}_{t-1})}{2 \sum (h_t - \hat{f}_{t-1})^2} \quad (32)$$

This is yet another solution to the problem of boosting density estimates. However, it is seen that in practice the above discussed methods do not always perform satisfactorily (perhaps due to the type of base estimators used). In the next section an approach is introduced that uses a divergence measure between the current estimate and the new “weak” estimate directly to compute the mixing weights. It is shown that this method performs well in practice.

### 3.2.3 KL divergence-based Approach

From preceding discussion it is seen that the selection of the best estimator at each step depends on minimizing the mutual information between the current estimate and the chosen estimate. This would suggest that a distance measure between the two distributions could be used to predict the mixing weight,  $\alpha$ . The symmetric KL divergence is used to compute  $\alpha$ . The basic premise is that, if the divergence between the current and chosen density estimates is large, indicating that the chosen density has high information content then  $\alpha$  needs to be higher. But too high a divergence should be penalized to allow a more smooth update of the estimate. This is accomplished by using a Gaussian function for drawing the values of  $\alpha$ . The mean of the Gaussian is set to 1 (this can be done without loss of generality by the use of a proper normalization constant). Also, a high divergence at latter iterations could imply that the chosen estimator is modeling outliers and hence the divergence measure at latter iterations should be given a smaller weight. A damping function of the form  $C^{-T+1}$  is used to accomplish this. The proposed algorithm is given in Table 20. The parameters  $V$  (variance of the Gaussian) and  $C$  are found by cross-validation. The normalization constant is chosen to be the divergence measure from the first iteration. Typically,  $0.16 < V < 5$  and  $1 < C < 1.01$ . At each iteration the data is weighted based on the base estimator’s ability to model the data at the previous iteration. The weighted data is then transformed using PCA and a GMM (although in general, any weak learner can be used) is used to fit the data to obtain the new estimate.

**Table 20. KL divergence-based approach.**

1. Set  $F_0(z)$  to uniform over the example domain
2. Let  $D$  be the data matrix
3. Let  $d$  be the dimension of the Gaussian
4. For  $t = 1:T$ 
  - a) Set  $w_t = \frac{1}{F_{t-1}(z)}$
  - b)  $D_1 = \text{PCA}(w_t * D, d)$
  - c) Use weak learner with  $D_1$  and obtain density estimate,  $h_t$ .
  - d) Let  $A(t) = \sum_i F_{t-1} \log(\frac{F_{t-1}}{h_t}) + \sum_i h_t \log(\frac{h_t}{F_{t-1}})$
  - e)  $\alpha_t = \text{Gauss}(1, V, A(t)/A(1)) * C^{(1-t)}$ , where  $V$  is variance and  $C^{(1-t)}$  is the damping function.
  - f) Set  $F_t = (1 - \alpha_t)F_{t-1}(z_i) + \alpha h_t(z_i)$
5. Output the final  $F_T$

### 3.2.4 Experimental Validation

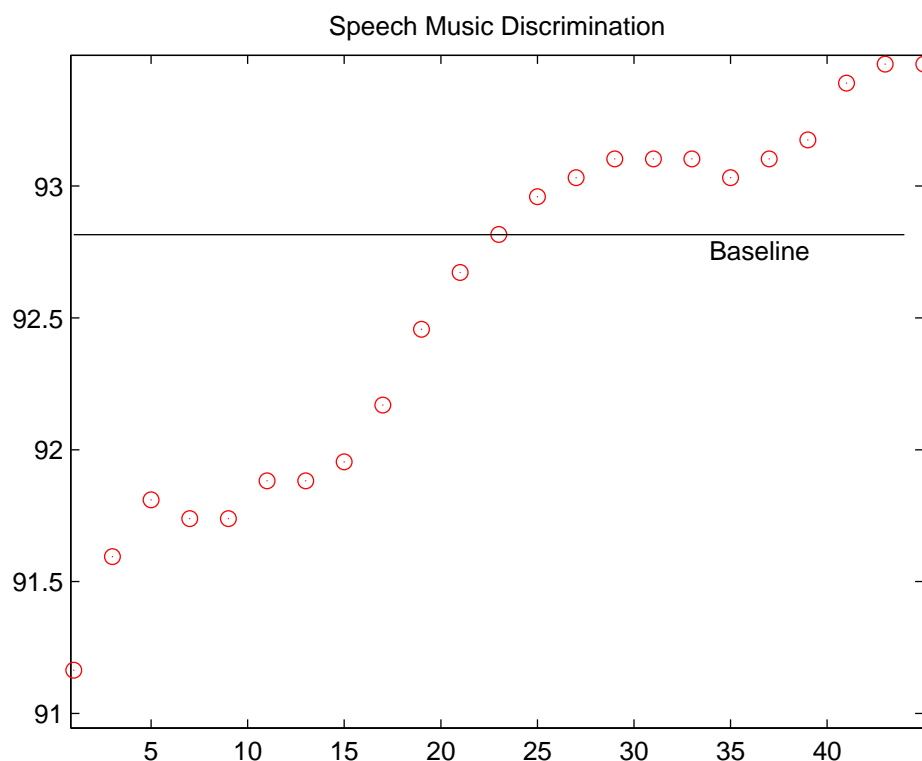
Speech versus noise classification and speech versus music classification tasks were used to evaluate the proposed method. The audio database was constructed from the *Phonak* database. Each 30 second file was divided into one second segment and used as an example for training and testing. There was no overlap between the recordings used for training and testing. The task consisted of predicting whether a given one second test segment belongs to one class or the other.

A four mixture GMM was chosen as the baseline since empirically it gave the best result for the considered audio classification tasks. The base estimator was also a four mixture GMM. In general, the KL divergence based method is not constrained by the type of base estimator and can be used to improve the estimate of any base estimator. NRAF features [37] were used for the classification. After PCA only the first 12 dimensions were used.

### 3.2.5 Results and Discussion

The first two methods discussed did not always perform well for the different tasks on which they were evaluated and hence were not used for comparison with the baseline. As is evidenced from Figs. 26 and 27, the KL divergence based algorithm improves the accuracy of the GMM-based classifier. Restricting the algorithm to work with one estimator results

in removal of one of the optimization steps leading to reduced computational cost. This also results in reduction in variance of the estimator. The proposed method could be generalized by allowing the weak learner to search a larger estimator space (parameterized by number of mixtures and dimensionality of the Gaussian), this would involve an additional optimization step in selecting the best estimator. This classifier can be used to process “secondary” features by letting GMMs model local clusters in the high dimensional feature space. The problem would then no longer be that of improving the accuracy of one base estimator but of using many base estimators and optimizing with respect to both, the mixing weights and the base estimator.

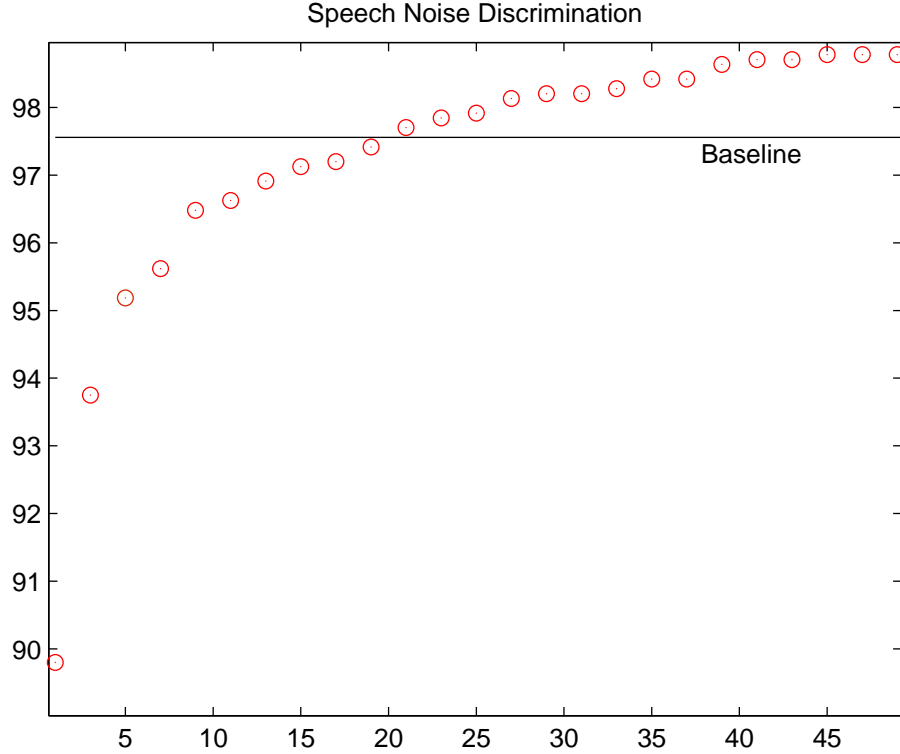


**Figure 26.** Plot showing the improvement in performance of the speech music classifier due to boosting.

### 3.3 Cascade Jump SVMs

It is believed that in the cortex (both visual and auditory) the classification is a stagewise process, different filters are used to progressively extract different aspects of the input





**Figure 27.** Plot showing the improvement in performance of the speech noise classifier due to boosting.

stimuli. Viola and Jones [57], showed the computational advantage of following a cascade-of-classifiers approach for a face recognition problem. In this section we look to leverage the benefits afforded by SVMs and the cascade structure, and propose a SVM-based cascade-of-classifiers structure. SVM classifiers aim to find a hyperplane in a high dimensional feature space that optimizes the generalization bounds. The idea of SVMs was conceived by Vapnik [65] more than three decades ago, but has come to receive widespread attention only recently. A common formulation of the SVM classifier, given two classes, is to find a hyperplane that maximizes the margin between the two classes. The idea is illustrated in Fig. 28. Let the training data be represented by the set:  $\{ \bar{x}_i, y_i \}$ , where  $\bar{x}_i$  is the feature vector for example  $i$ , and  $y_i$  is the true class label and is either 1 or -1. Each  $\bar{x}_i$  is scaled to lie in  $[0,1]$  or  $[-1,1]$ , this is important to prevent features with large variances from dominating the classification. If  $\bar{w}$  is the normal to the “optimal” hyperplane (optimal from

the perspective of maximizing the margin) separating the 2 classes, then the hyperplane can be expressed as:

$$\bar{w}' \cdot \bar{x} - b = 0 \quad (33)$$

the boundaries of the two classes can be expressed as,

$$\bar{w}' \cdot \bar{x} - b = 1 \quad (34)$$

$$\bar{w}' \cdot \bar{x} - b = -1 \quad (35)$$

Ideally we would expect the examples corresponding to  $y_i=+1$  to lie to the right of  $\bar{w}' \cdot \bar{x} - b = 1$  and the examples corresponding to  $y_i=-1$  to lie to the left of  $\bar{w}' \cdot \bar{x} - b = -1$ , i.e. to say the training data is separable. This can be combined into one constraint as,  $y_i(\bar{w}' \cdot \bar{x} - b) - 1 \geq 0 \forall i$ . It is trivial to show that the margin between the two boundaries is given by,  $\frac{2}{\|\bar{w}\|}$ , where  $\|\bar{w}\|$  is the norm of the normal  $\bar{w}$ . Thus, the SVM problem can be formulated as a quadratic programming problem given by,

$$\min \quad \frac{\|\bar{w}\|^2}{2} \quad (36)$$

subject to:

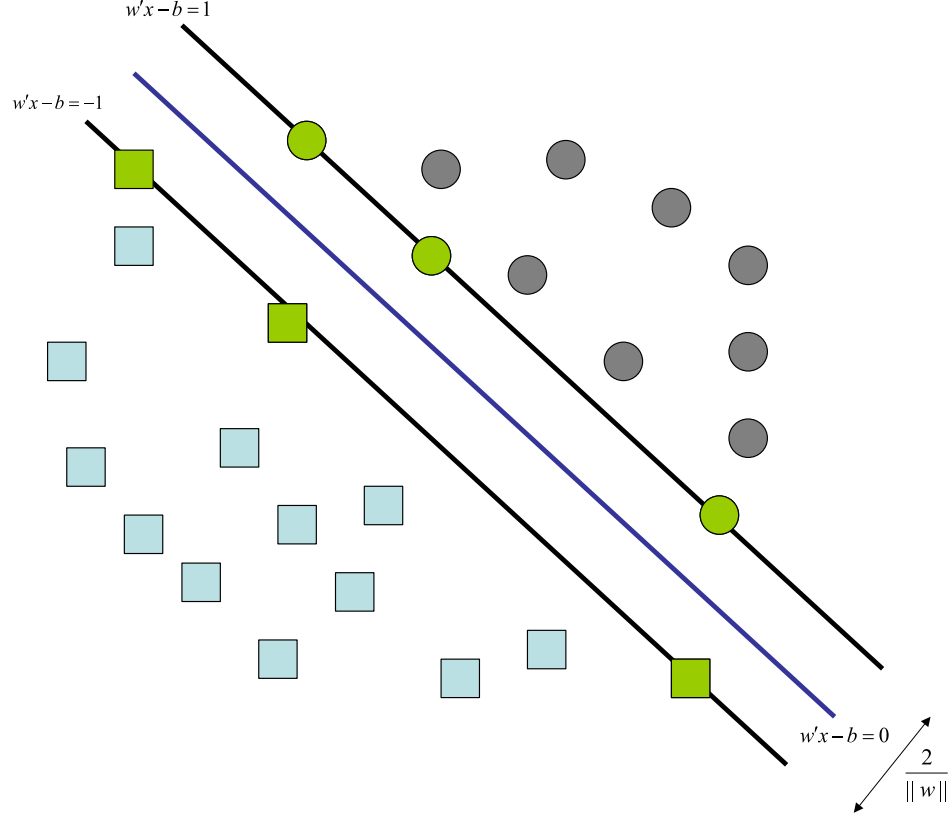
$$y_i(\bar{w}' \cdot \bar{x} - b) - 1 \geq 0 \quad (37)$$

For positive Lagrange multipliers  $\alpha_i$ , the Lagrangian for the above equation can be expressed as,

$$L_p \equiv \frac{\|\bar{w}\|^2}{2} - \sum_{i=1}^N \alpha_i y_i (\bar{w}' \cdot \bar{x}_i - b) + \sum_{i=1}^N \alpha_i \quad (38)$$

Thus, the problem is that of minimizing  $L_p$  w.r.t.  $\bar{w}$ ,  $b$  and also require that the derivatives of  $L_p$  w.r.t.  $\alpha_i$  vanish, all subject to the constraints that  $\alpha_i \geq 0$ . Since this is a convex

optimization problem, the equivalent Wolfe dual problem [66] can be solved. The dual problem can be stated as maximizing  $L_p$ , subject to the constraints that its gradient w.r.t.  $\bar{w}$ ,  $b$  vanish, and also subject to the constraints that  $\alpha_i \geq 0$ .

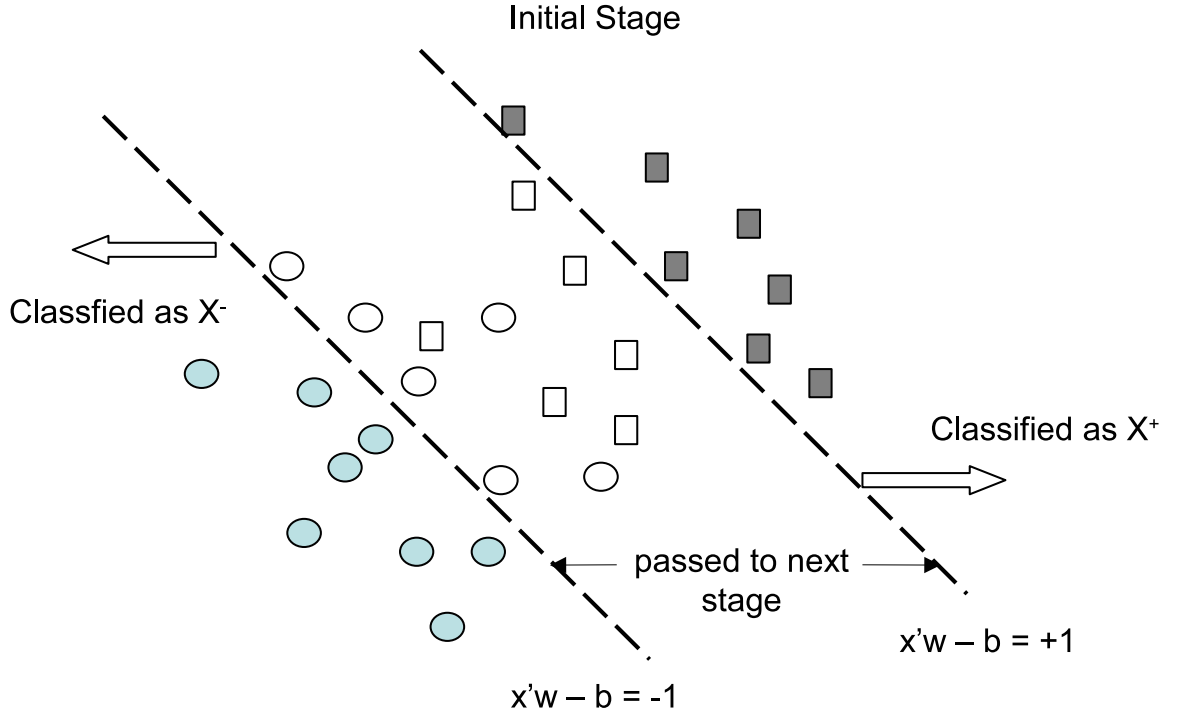


**Figure 28.** Figure showing a SVM classifier. The concept is to find a hyperplane that maximizes the margin between the two classes.

Different SVM algorithms exist based on different optimizations such as optimizing the maximal margin, the number of support vectors etc. However all the algorithms have an inherent tradeoff between accuracy (the ability of the classifier to learn a given training set) and capacity (the ability to learn any training set without error) [67]. Improving the accuracy on the training set by using complex SVMs leads to poor generalization. Herein, the problem of improving the generalization is approached by suggesting the use of complex SVMs on a reduced number of data points and attempting to remove the noisy support vectors in earlier stages, thereby limiting the effects of overfitting [68]. The notation used herein is as follows: matrices are represented by upper-case letters, vectors are represented

by lower-case letters with a bar above and plain lower-case letters represent scalars.

The basic idea of cascade jump SVMs (CJSVMs) is to have a cascade of SVM classifiers with low false negatives upfront and complex classifiers later on in the cascade. The cascade structure presented here is, in spirit, similar to the work presented by Viola and Jones using AdaBoost based classifiers [57]. To implement the low false negative classifiers a modified proximal SVM [69] is used to remove the “peripheral” examples, and a standard SVM with RBF kernel is used to perform the final classification. As seen from Fig. 29, the data points that do not lie between the two proximal support vectors are considered as easy examples and readily classified. The data points between the two hyperplanes are passed on to the next stage for further processing.



**Figure 29.** Figure showing the concept of cascade jump SVMs. The easily separable data points are removed before presenting the rest of the data points to the next classifier in the cascade.

The SVM formulation for classifying  $m$  points in  $n$ -dimensional real space  $R^n$  is given by:

$$\begin{aligned}
& \min_{(\bar{w}, b)} \quad \frac{1}{2} \bar{w}' \bar{w} \\
\text{s.t.} \quad & Y(X\bar{w} - \bar{u}b) \geq \bar{u}
\end{aligned} \tag{39}$$

$\bar{x}$  represents the data point,  $y$  is the associated class label (+1 or -1) and  $\bar{u}$  is a unit vector.  $X$  and  $Y$  are defined as,

$$X = \begin{bmatrix} \bar{x}'_1 \\ \bar{x}'_2 \\ \vdots \\ \bar{x}'_m \end{bmatrix}$$

$$Y = \begin{bmatrix} y_1 & & & \\ & y_2 & & \\ & & \ddots & \\ & & & y_m \end{bmatrix}$$

$\bar{w}'$  is the normal the planes given by,

$$\bar{x}'\bar{w} - b = +1 \tag{40}$$

$$\bar{x}'\bar{w} - b = -1 \tag{41}$$

Fung and Magasarian [69] developed a SVM classifier (proximal SVMs) that found hyperplanes around which data is clustered and assigned data points to the closest hyperplane. The SVM formulation can be modified to give the proximal SVM formulation as below,

$$\begin{aligned}
& \min_{(\bar{w}, b)} \quad \frac{1}{2} (\bar{w}'\bar{w} + b^2) \\
\text{s.t.} \quad & Y(X\bar{w} - \bar{u}b) = \bar{u}
\end{aligned} \tag{42}$$

This modification leads to an explicit exact solution to the classification problem. The planes  $\bar{x}'\bar{w} - b = \pm 1$  can be thought of as planes around which the data is clustered. For the CJSVM the classification function was changed to be,

$$\bar{x}'\bar{w} - b = \begin{cases} \geq +1, & \text{then } \bar{x} \in X^+, \\ \leq -1, & \text{then } \bar{x} \in X^- \end{cases}$$

The change amounts to removing all the points to the right of the plane  $\bar{x}'\bar{w} - b = +1$  and left of the plane  $\bar{x}'\bar{w} - b = -1$ . This kind of classification would result in very low false negatives. The data points between the planes are then passed on to the next classifier in the cascade. As data points traverse the cascade they jump from one feature space to another due to application of different kernels. The Lagrangian for the classification problem can be written as:

$$L(\bar{w}, b, \bar{\lambda}) = \frac{1}{2}(\bar{w}'\bar{w} + b^2) - \bar{\lambda}'[Y(X\bar{w} - \bar{u}b) - \bar{u}] \quad (43)$$

Differentiating L w.r.t  $\bar{w}$ ,  $b$ ,  $\bar{\lambda}$  and setting to zero,

$$\bar{w} = X'Y\bar{\lambda} \quad (44)$$

$$b = -\bar{u}'Y\bar{\lambda} \quad (45)$$

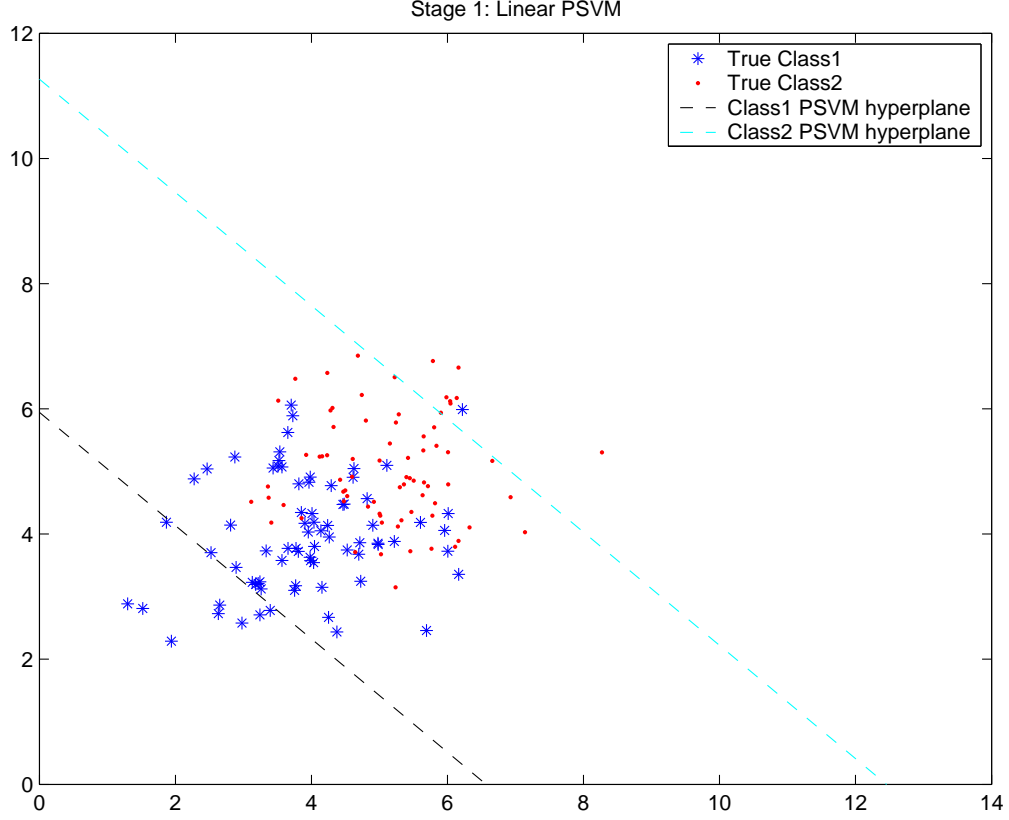
$$Y(X\bar{w} - \bar{u}b) - \bar{u} = 0 \quad (46)$$

Substituting  $\bar{w}$  and  $b$  in Eqn. 46 above,

$$\bar{\lambda} = [Y(XX' - \bar{u}\bar{u}')Y]^{-1}\bar{u} \quad (47)$$

It has been shown [69] that Eqn. 47 can be very efficiently implemented to perform the classification much faster than the standard SVM. Typically, the proximal SVM is an order of magnitude faster than the standard SVM. The working of CJSVM for synthetic 2-D data is shown in Figs. 30-35. Figures 30 and 31 show the first stage of the cascade which is a proximal SVM with linear kernel. Figure 30 shows the hyperplanes around which the data is clustered and Fig. 31 shows the classification of the data points that are not between the hyperplanes. Figures 32 and 33 shows the working of the second stage which is also a proximal SVM with linear kernel, the reason for having a second linear stage is that the presence of, say, a negative outlier in the positive class prevents the hyperplane from

removing all the positive peripheral points it can if the outlier were not present. Figures 34 and 35 show the classification of peripheral points by a proximal SVM using sigmoid and polynomial kernels.



**Figure 30.** Plot showing the linear hyperplane for the first stage.

The performance improvement due to CJSVM is shown for an auditory scene recognition task. Each one second segment was divided into 20 msec frames and 12 MFCCs were extracted from each frame. The mean and variance of these features over the one second segment were computed to yield a 24-dimensional feature vector. The features were reduced to a dimension of 13 using PCA.

For the purpose of comparison a standard SVM with RBF kernel was used as the baseline. For each of the experiments the best parameters for the RBF-SVM were found by repeated trials. Software provided by Ma et al. [70] was used to implement the standard SVM classifier. The CJSVM consisted of 2 stages of linear proximal SVMs followed by the RBF-SVM with same parameters as in the baseline case. The statistical significance of the

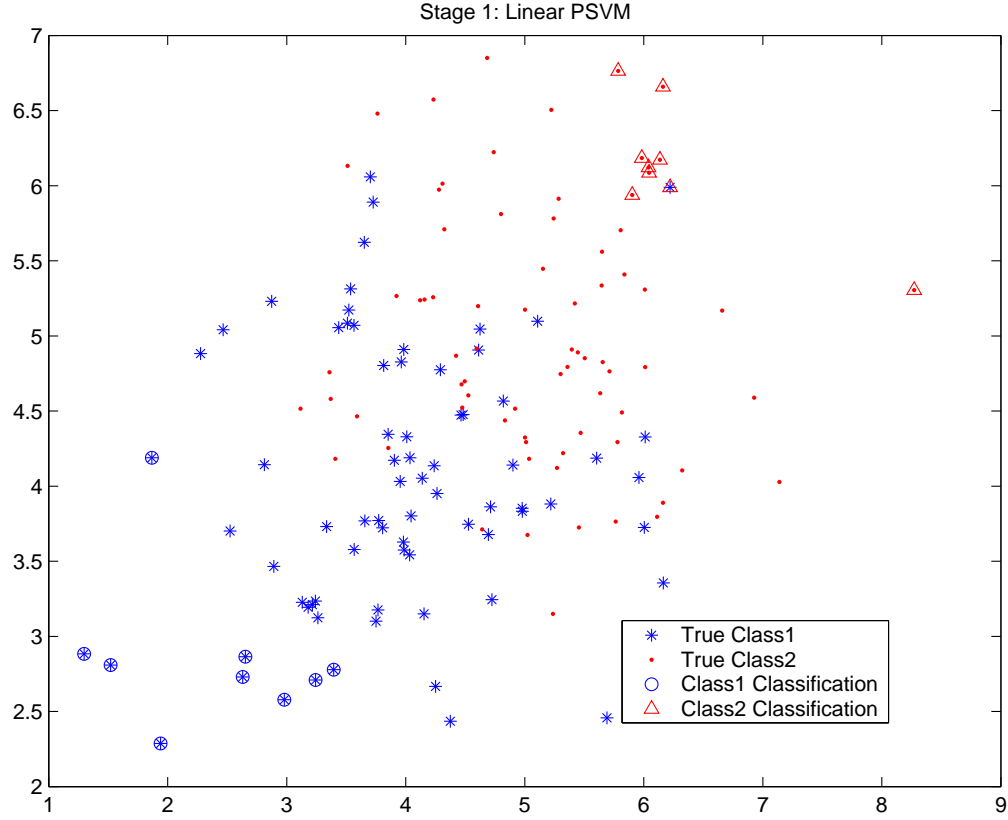


Figure 31. Plot showing classification of points for first stage. Data points not lying between the hyperplanes are classified as belonging to the positive or negative class.

improvement was tested using a difference of proportions significance test (see Table 23). It is seen that the new approach achieves significant improvement over the conventional SVM. Since SVMs can handle high-dimensional data, the application of CJSVM to a high-dimensional feature set is straight forward. However, using a conventional SVM (with high-dimensional feature sets) would be computationally expensive and CJSVM provides a way of circumventing this issue.

Table 21. Classification between social and office auditory scenes. 13 PCA transformed MFCCs were used. For the RBF-SVM  $C = 100$  and  $\gamma = 0.09$ . The same parameters were also used for the final stage of CJSVM. The first two stages were linear SVMs.

Scene Recognition Using MFCC	
Method	Percentage Correct
SVM	95.17 %
CJSVM	97.22 %



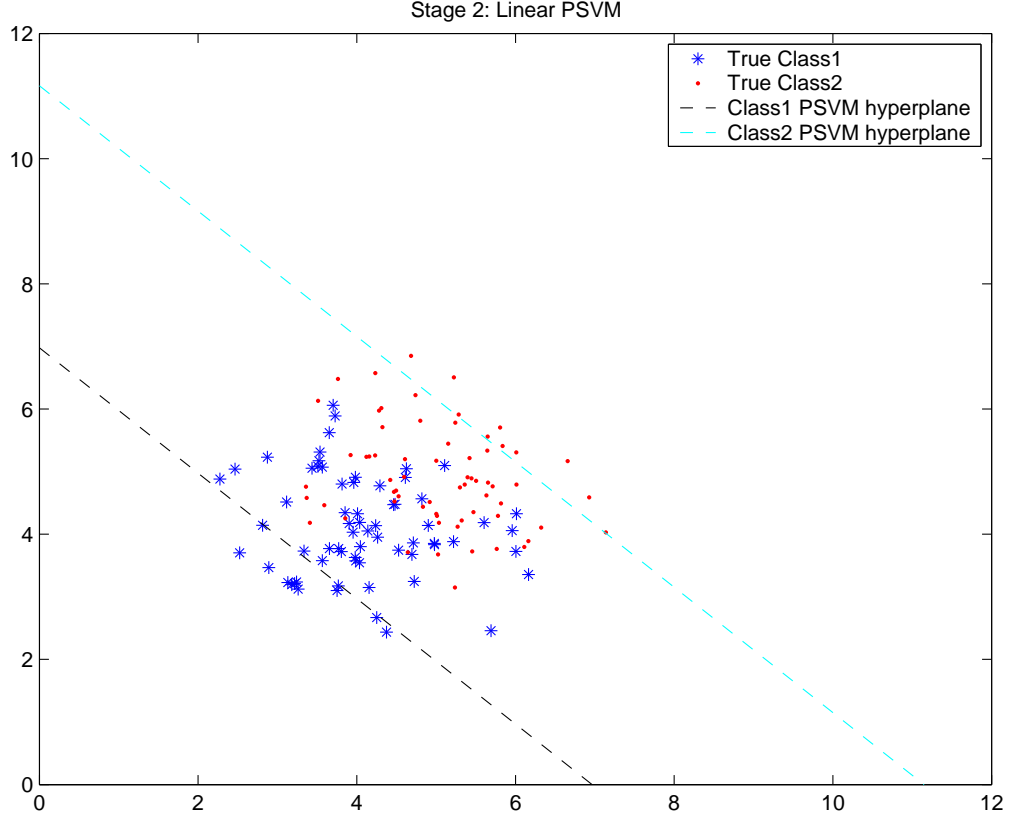
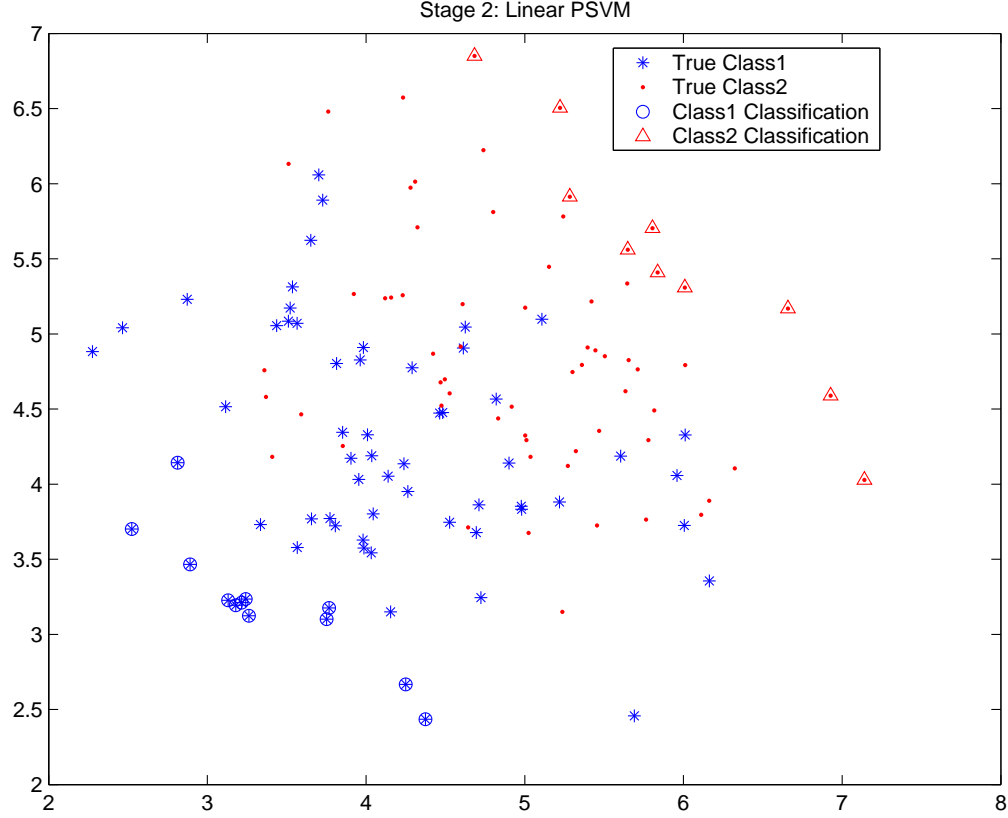


Figure 32. Plot showing the linear hyperplane for the second stage.

### 3.4 Dimensionality Reduction Using AdaBoost

In pattern classification problems it is often desirable to find a relevant subspace in order to reduce the dimensions of the data before clustering. PCA, which finds a set of axis of maximum variance, is often used to accomplish this. But PCA makes an implicit Gaussian assumption about the underlying data distribution that need not always be true. The method presented here is based on AdaBoost and does not transform the feature space. The relevant features are selected based on a minimum error criterion. This technique is particularly useful in selecting the best features from a collection of different feature sets, where techniques such as PCA might not work well since the features are not from the same feature space.

The new algorithm [71] does not make any assumptions about the underlying distribution of the data points nor does it transform the feature space in order to obtain new features. The algorithm is similar to the one proposed by Tieu and Viola [55] except that an



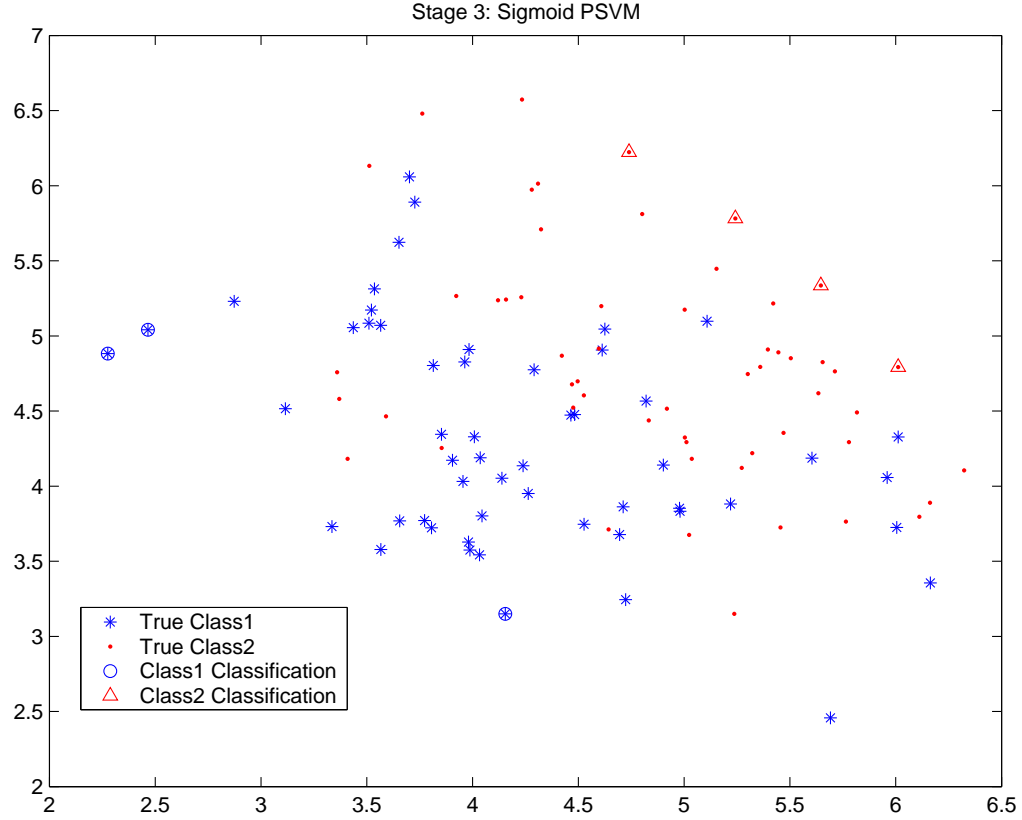
**Figure 33.** Plot showing classification of points for second stage. Data points not lying between the hyperplanes are classified as belonging to the positive or negative class.

additional constraint is imposed to prevent AdaBoost from picking the same feature twice. The modified AdaBoost algorithm referred to as constrained AdaBoost (cAda) is shown in Table 24.

The PCA and cAda algorithms were compared in a four way audio classification problem described previously. NRAF features were extracted from each sound example and were reduced to different dimensions. A Gaussian Mixture Model (GMM) based classifier was used to perform classification on the reduced set. The hypothesis for AdaBoost was done by calculating the minimum of the absolute distance between a feature and the mean value of that feature for the four classes. Hypothesis for example  $x_i$  using feature  $j$  ,

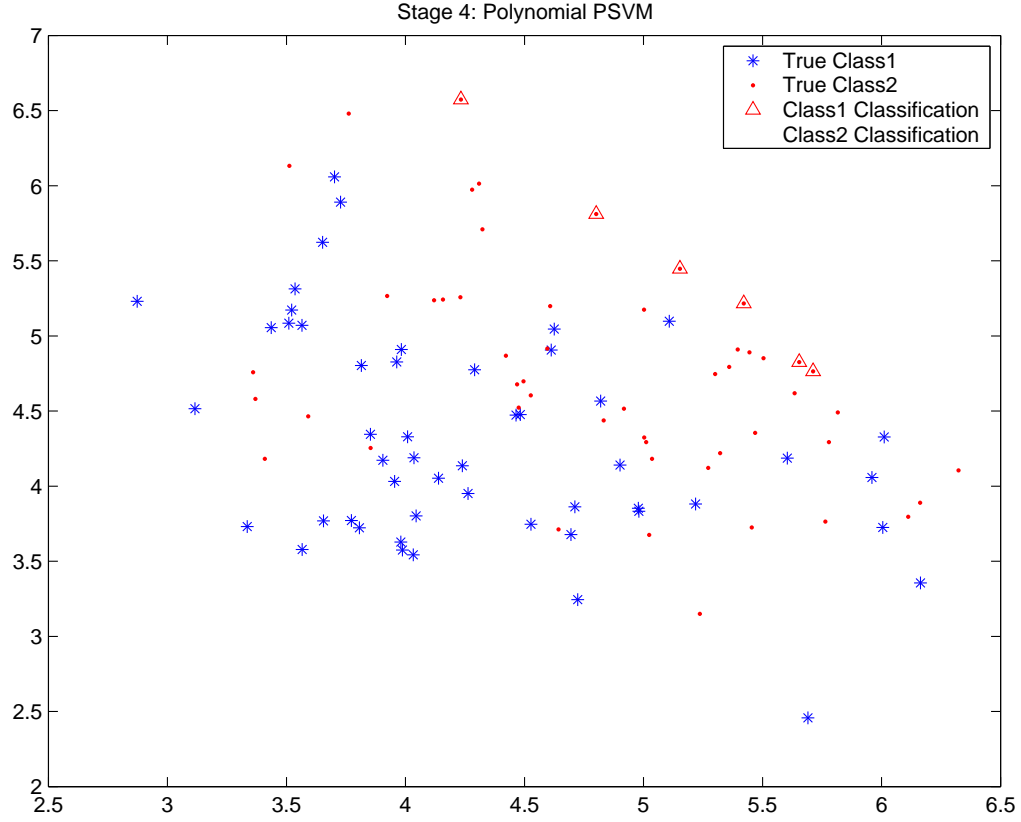
$$h_{ij} = \underset{k \in \{1,2,\dots,4\}}{\operatorname{argmin}} (|x_i(j) - \mu_k(j)|) \quad (48)$$

The input data was reduced to 10, 20, 30, 40, 50 and 60 dimensions using PCA and



**Figure 34.** Plot showing classification of points using a modified proximal SVM with sigmoid kernel.

cAda. The classification results are shown in Fig. 36. PCA performs better with few features because it is capable of generating new features using a projection from multiple features, thus when we go down to a smaller dimension, it can take into account more information from the original feature set. On the other hand, the cAda algorithm needs a few iterations (at each iteration it selects a new feature) to build a collection of features that can learn the feature space well. The cAda algorithm learns more about the feature space with every iteration and the error generally decreases, though not necessarily monotonically. The decrease in performance in data-set 2 (see Fig. ??(b)) while increasing the dimension from 10 to 20 using PCA can be explained by the fact that PCA creates new features based on variance information and this new information need not necessarily reflect the true class models. However, with more information available the GMM is able to model the classes better. It should also be noted that since GMMs with diagonal covariance were used, the results are probably skewed in favor of PCA. Further, the cAda algorithm can be used to



**Figure 35.** Plot showing classification of points using a modified proximal SVM with polynomial kernel.

work with large dimensional feature sets, which may not be possible with PCA.

### 3.5 Design Notes

The guiding design principle of the AdaBoost-based classifiers presented here was to develop classifiers that were easy to implement in low-power hardware. It is mainly due to this reason that the weak learners in almost all AdaBoost implementations were simple distance based classifiers. It is easy to envision greater performance by the use of better weak learners. In the AdaBoost setting, a better weak learner would be one which is able to capture different characteristics of the underlying feature space. In other words, the hypothesis space should be diverse and rich, using classifiers that individually give very good accuracy but model the same characteristic of the data set will not be a good choice for weak learners.

The main concept of the CJSVM is to utilize the different feature spaces afforded by the different kernels to maximize the separability of the data. The type of kernels to use

**Table 22. Classification between social and industrial auditory scenes. 13 PCA transformed MFCCs were used. For the RBF-SVM  $C = 100$  and  $\gamma = 0.4$ . The same parameters were also used for the final stage of CJSVM. The first two stages were linear SVMs.**

Scene Recognition Using MFCC	
Method	Percentage Correct
SVM	87.93 %
CJSVM	91.74 %

**Table 23. Results using the difference of proportion significance tests for each of the experiments. It is seen that the CJSVM gives a significant improvement over SVM**

Significance Tests		
Experiment	Significance Level	Differences
1	0.1	6.50
2	0.05	16.13

and the number of stages have to be determined empirically and will change depending on the task at hand. However, a simple rule to remember is that the earlier stages should be “simpler” in order to avoid overfitting and also the minimize processing time. For instance, it would make sense to use linear kernels upfront as opposed to say, radial basis function (RBF) kernels. The length of the cascade is decided by the trade-off between accuracy and computation. At some stage the improvement in performance is negligible compared to the increase in computational cost. In the implementation presented here all data points lying between the two hyperplanes were considered as “difficult” examples. Perhaps a better approach would be to define a distance measure around the hyperplane to differentiate “easy” and “difficult” examples.

Three different approaches to designing the generative AdaBoost classifier were presented. The approaches based on minimizing inefficiency and  $L_2$  norm, though theoretically sound, do not perform very well in all cases. This is rather surprising, and might be due to the type of base classifiers used. The KL divergence-based method worked well in the tasks it was evaluated on. The damping function plays an important role in the functioning of the algorithm, and the parameter  $C$  has to be computed empirically to best suit the considered task. It should be noted that the damping function need not be  $C^{1-T}$ , it can be of the form  $C^{n-T}$ , or belong to an altogether different family. The KL divergence-based method can be easily extended to include more than one base estimator. An important fact to remember is that different base estimators will give divergence values that are not in the

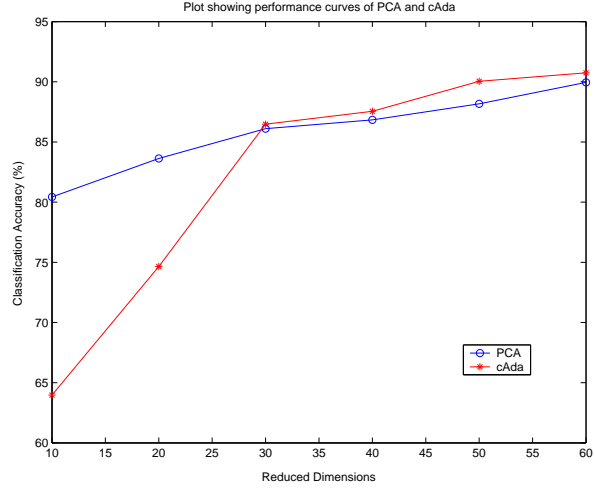
**Table 24. Dimensionality reduction using AdaBoost**

<p><b>Input</b></p> <ol style="list-style-type: none"> <li>1. <math>N</math> training examples  <math>(x_1, y_1), \dots, (x_N, y_N)</math>  with <math>y_i = 1, 2, 3</math> or 4, the true class of <math>x_i</math>,</li> <li>2. <math>T</math>, the number of iterations</li> <li>3. <math>h_{i,j}</math>, hypothesis for example <math>x_i</math> based on feature <math>j</math></li> </ol> <p><b>Initialize</b> the weights <math>w_{1,i} = 1/N</math></p> <p><b>Do for</b> <math>t = 1 : T</math></p> <ol style="list-style-type: none"> <li>1. For all features calculate the error,  <math display="block">\epsilon_j = \sum_{i=1}^N w_{t,i} D</math> where <math>D = 0</math> if <math>h_{i,j} = \text{true class of } x_i</math>, else <math>D = 1</math></li> <li>2. Choose the feature which corresponds to the minimum error <math>\epsilon_t</math> amongst the features that were not selected before.</li> <li>3. Update weights,  <math display="block">w_{t+1,i} = w_{t,i} \beta_t^{e_i},</math> where <math>e_i = 1</math> or 0 for <math>x_i</math> classified correctly or incorrectly respectively, and  <math>\beta_t = \frac{\epsilon_t}{1-\epsilon_t}</math></li> <li>4. Normalize the weights,  <math display="block">w_{t+1,i} \leftarrow \frac{w_{t+1,i}}{\sum_{k=1}^N w_{t+1,k}}</math></li> </ol>
---

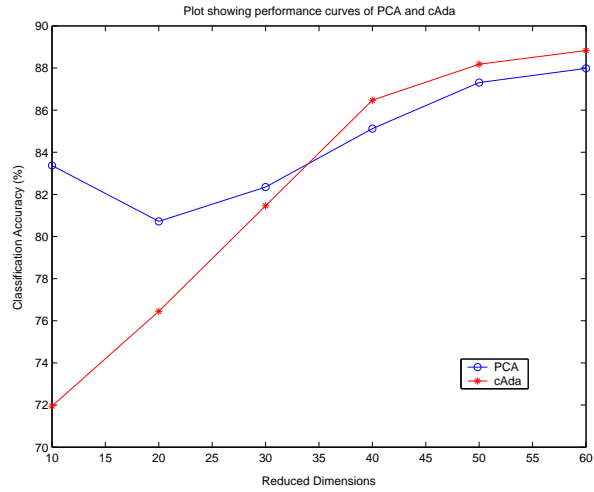
same range. Hence an additional normalization step has to be performed to ensure that all divergences are in the same range. The use of a Gaussian is more of a convenience and a different function could be used in place of the Gaussian.

### 3.6 Summary

This chapter presented classification algorithms, capable of working directly with high-dimensional data. The cascade jump SVM takes advantage of the discrimination afforded by different feature spaces (due to the use of different kernels) to improve the accuracy of the traditional single kernel SVM. The generative AdaBoost classifier is able to provide a likelihood measure for each class and scales well to large number of classes. It can be used to incorporate sparse, high-dimensional data into a traditional HMM for speech recognition.



(a)



(b)

**Figure 36.** (a) Plot showing the performance curves of the classification system after dimensionality reduction by PCA and cAda for data-set 1 (b) shows the same plot for data-set 2

Further, a dimensionality reduction technique was presented that achieves dimensionality reduction without feature transformation. This property of the algorithm can be exploited for feature selection and classifier combination.

## CHAPTER 4

### APPLICATIONS AND FUTURE WORK

The emphasis of this thesis has been on developing biologically inspired robust classification systems that can be implemented on a low-power platform. In this chapter we focus on building a front-end sound classifier for electro-acoustic tuning of a hearing-aid. A hearing-aid consists of various algorithms each suited for different conditions. A classification system that can accurately identify the ambient environment of the user can help in automatically switching to the most appropriate algorithm. One of the major requirements of such an application, apart from robustness to various noise conditions, is low power.

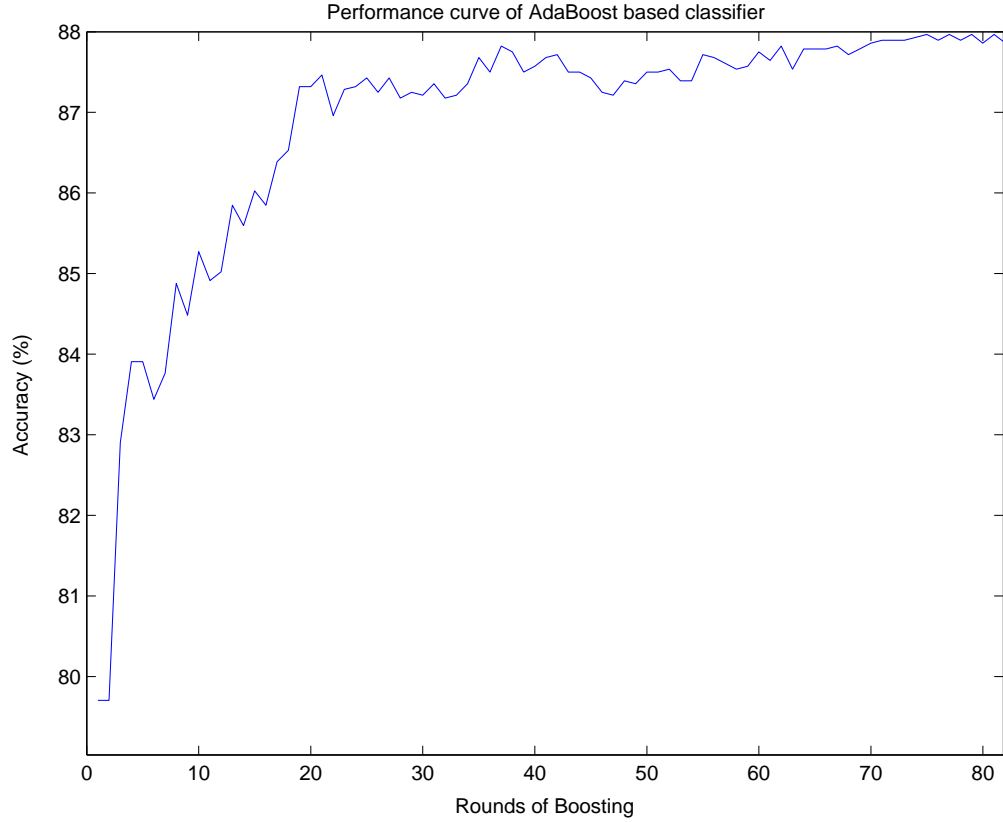
An AdaBoost-based classification system using the previously described features is used as a front-end sound classifier to a hearing-aid. Apart from demonstrating the high accuracy of the system, the generalization ability of the system on a completely different database is also shown. For the audio classification experiment, the *Phonak* database was used. It consisted of four classes namely speech, music, noise and speech in noise.

The files were segmented into one second segments and each segment was treated as a separate example. The audio classification experiment was aimed at classifying the input sound signal as one of the four categories (music, speech, noise or noisy speech). Eighty percent of the files in each class were randomly chosen as the training set and the rest as test set. In no case was there any overlap between the training and test sets.

NRAF and STRF were used as the features for classification. The STRF features for each 1 second segment were collapsed (averaged) across the time axis and aligned to form a single vector. The mean and variance of the NRAF features over the 1 second segment were computed to form a single feature vector. The two feature vectors were concatenated to form a super vector to represent each 1 second segment. The features were fed to the AdaBoost-based classifier and boosted for 200 rounds. It was found that training more or less saturates after round 75 as shown in Fig. 37. Thus the final classifier was built using these 75 features.

The results for the audio classification experiment are shown in Table 25. The AdaBoost-based classifier was able to learn the noise class well. Most of the errors made in the music





**Figure 37. Performance of the AdaBoost-based classifier with rounds of boosting.**

class were misclassifications as speech, this might be due to the fact that for 1 second segments it is difficult to distinguish between speech and vocals. Maximum number of errors in the noisy speech class were misclassifications as noise, this is understandable because it is possible that over certain one second segments speech is dominated by noise. Most of the errors in the speech class were misclassifications as noisy speech. The outputs of the 1 second segments were combined over 30 seconds by a simple majority voting and the results are as shown in Table 26. It is seen that the overall accuracy increases to 97.91 %.

**Table 25. Table showing results for AdaBoost-based classifier using 1 second data segments for training and testing on the *Phonak* database. Overall accuracy was 87.96%.**

Category	Hit Rate	False Alarm
Music	85.63 %	2.87 %
Noise	91.81 %	4.17 %
Speech	87.78 %	3.54 %
Speech in Noise	86.63 %	5.45 %

**Table 26.** Table showing results for AdaBoost-based classifier using 30 second data segments (the outputs of the 1 second case were combined by majority voting) on the *Phonak* database. Overall accuracy was 97.91%.

Category	Hit Rate	False Alarm
Music	91.66 %	0 %
Noise	100 %	2.7 %
Speech	100 %	0 %
Speech in Noise	100 %	0 %
Overall	97.91 %	

## 4.1 Digital Hardware Implementation

In this section the design compromises required for a real-time digital hardware implementation of the hearing-aid front-end system are discussed and results from the hardware implementation are presented.

### 4.1.1 Feature Extraction

*TI c5510* fixed-point processor was chosen as the digital implementation platform due to its low-power capabilities. In order to achieve real-time performance and reduce power consumption the sampling rate was reduced to 8kHz and 20 channels were used instead of the original 128. Since the fast Fourier transform (FFT) was provided as part of the DSP library, FFT was used to accomplish the frequency decomposition. The modified feature extraction process is shown in Fig. 38. The FFT bins are combined into 20 channels. Each channel is then root compressed with a compression factor of  $\alpha=0.3$  and DCT is used to obtain the first set of features. A copy of the spectrum is subjected to a multi-scale transform with 4 rate filters and 6 scale filters, to obtain the cortical features (STRFs). The results of hardware implementation are presented in Table 27. Simulation results of a 20-filter software implementation are also presented to quantify the effects of fixed point implementation. It is seen that reducing the spectral resolution and using reduced number of temporal and spectral filters results in a drop in performance from 87.96 % to 82.79 %. Using the FFT and fixed-point precision further drops the performance to 78.73 %.

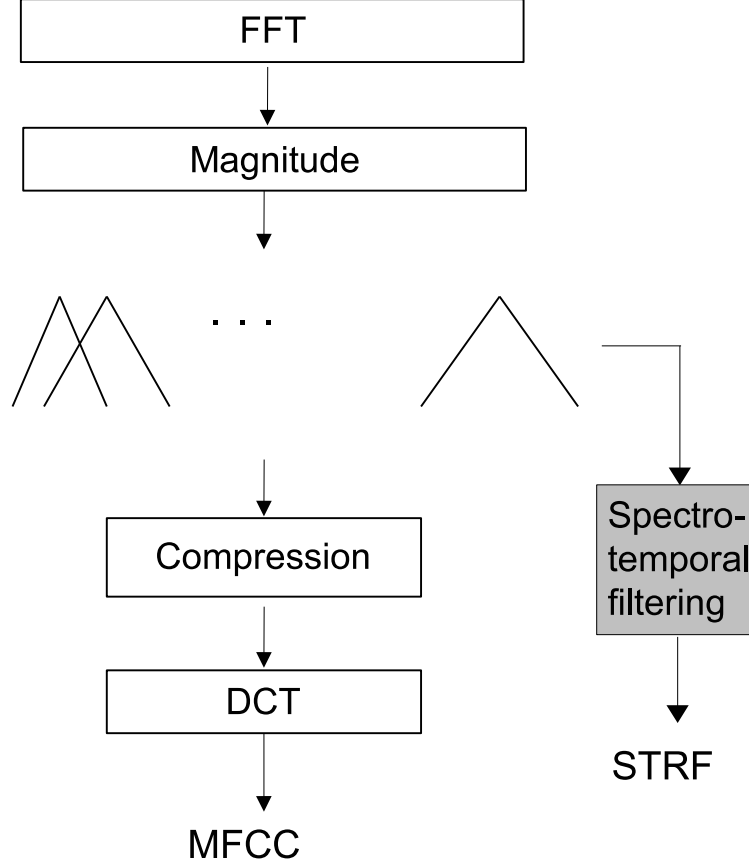


Figure 38. Feature extraction process for the hearing-aid front-end as implemented on the *c5510* fixed point processor.

#### 4.1.2 Implementation of the classifier

Once the system is trained offline, the runtime classification rule is simply a weighted sum and compare operation. Four 1-versus-rest classifiers were used for the four classes. The Euclidean distance measure between a feature and class mean of that feature is used as the hypothesis (weak learner) for AdaBoost. During the test phase the features picked by the AdaBoost classifier during the training phase are extracted from the test examples and their corresponding hypotheses are computed. The hypotheses are then multiplied with the pre-assigned weights (computed during training). The final decision rule is given by:

$$Decision = \begin{cases} 1 & \text{if } \sum_{j=1}^N \alpha_j h_j(x_i) > \frac{1}{2} \sum \alpha_j \\ 0 & \text{else} \end{cases}$$

**Table 27.** Table showing results for the *Phonak* database using simulation of CADSP implementation. Overall accuracy was **82.79 %**.

Category	Simulation (20 channel)	Fixed-Point
Music	86.21 %	88.50 %
Noise	76.0 %	66.50 %
Speech	87.35 %	83.04 %
Speech in Noise	72.55 %	76.86 %
<b>Overall</b>	<b>82.79 %</b>	<b>78.73 %</b>

#### 4.1.3 Power

The entire code size was about 8kbytes and the data memory utilized (for processing 1 second segment) was about 62 kbytes. The core power consumption was about 2 mW. For an application specific integrated circuit (ASIC) design, a good rule of thumb is to double the core power consumption. Thus the entire digital implementation would consume about 4 mW of power for processing one second of data.

## 4.2 CADSP Implementation

In this section the implementation of the auditory model based feature extraction and the AdaBoost-based classification process using the cooperative analog-digital signal processing (*CADSP*) [72] approach is proposed. Although, the analog VLSI circuitry needed for the implementation has already been developed [34], [2], [73], [74], [75], the actual *CADSP* implementation of the complete system is left as part of future work. Herein, the high level block diagram of the implementation is discussed along with power advantages that could be gained by using the *CADSP* approach.

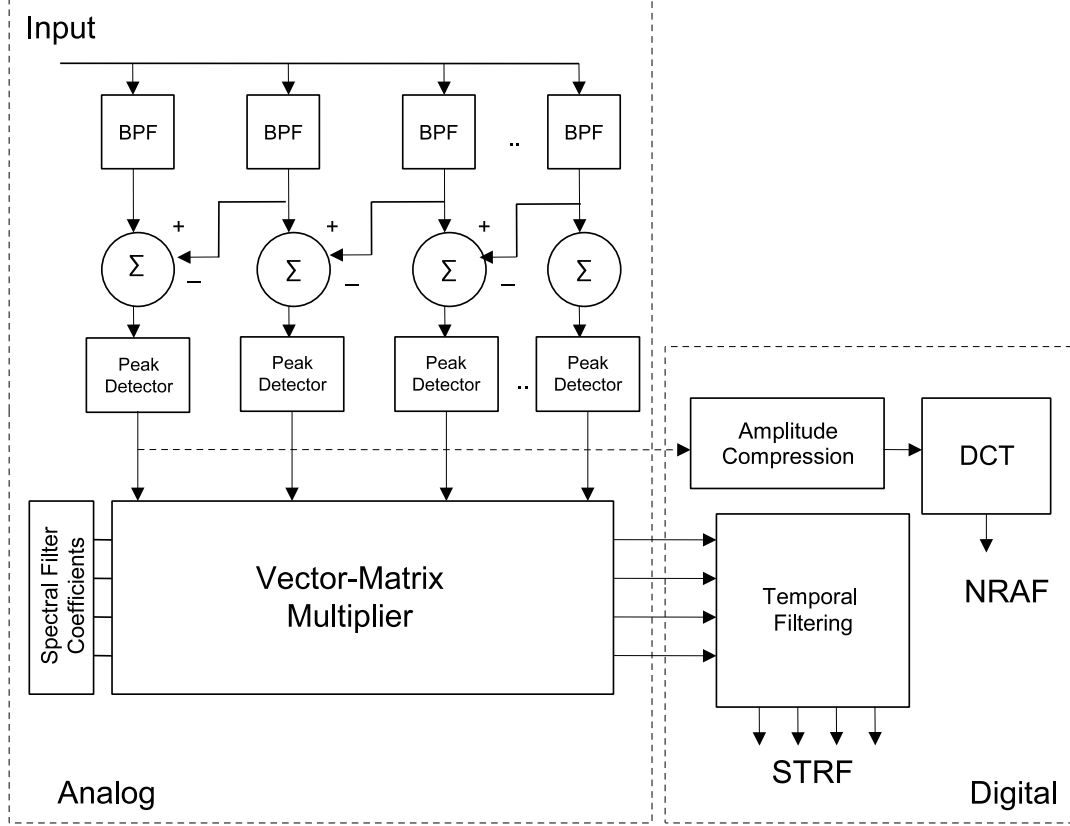
### 4.2.1 Feature Extraction

Figure 39 shows a block diagram of how the entire feature extraction can be implemented using the *CADSP* platform. While working with low-power analog design, noise and resolution issues have to be carefully considered. As in the case of the digital implementation, the number of filters in the bandpass filter bank need to be reduced (a 20 channel filter-bank is suggested but filterbanks with upto 32 filters have been implemented successfully [76]). Even though the reduction in the number of filters affects the accuracy of the classifier, the

performance (in simulation) is still better than the control case (using MFCCs and GMM-based classifiers). Each of the bandpass filters can be built with a Capacitively Coupled Current Conveyor Second-Order Section ( $C^4$  SOS) [2]. The entire filterbank would consume about  $4 \mu\text{W}$  of power. The half-wave rectifier followed by the lowpass filter can be implemented as a peak detector. The difference operation between adjacent frequency channels as well as the peak detector consumes about  $1 \mu\text{W}$  of power [75]. A copy of the signal at this point is routed to a DSP to perform the amplitude compression and decorrelation using DCT, to yield the NRAF features. Due to the difference operation between adjacent frequency channels there are only 19 outputs, that are fed to a vector-matrix multiplier (VMM) [75] to perform the spatial filtering. The advantage of using this circuit is that all the 6 spatial filtering operations can be performed in parallel. The VMM is capable of performing  $5 \text{ MMACs}/\mu\text{W}$ , hence it would consume about  $0.2 \mu\text{W}$  of power for processing one second of data (assuming 8 kHz sampling rate). The output of the VMM is fed to a DSP to perform the temporal filtering and obtain the STRF features. The entire analog part can be implemented in less than  $6 \mu\text{W}$  of power, thus yielding a power reduction of close to a factor of 1000.

### 4.3 Summary

This thesis highlights the usefulness of incorporating physiological processing functionalities into audio signal processing. On the feature front, modifications were suggested to the mel-frequency cepstral coefficients that improve the noise-robustness of these features without adversely affecting the performance in clean conditions. The improvements afforded by these modifications are demonstrated for connected digit recognition and audio classification tasks. Classification algorithms based on the ideas from the machine learning community are presented that are able to work directly with high dimensional data. The AdaBoost-based classifiers presented are able to work with a wide variety features, thus efficiently incorporating the discrimination afforded by different feature sets into the classification process. The Generative AdaBoost classifier presented, is able to boost estimates of density to provide a more accurate measure of the underlying data distribution. This enables it to be used in tasks such as speech recognition and also overcome the scalability issues



**Figure 39.** Block diagram showing the proposed implementation of the feature extraction process on a *CADSP* platform. A 20 channel implementation consumes about  $5.2 \mu\text{W}$  of power for the analog part. The DCT and temporal filtering are performed in the digital domain.

associated with the binary AdaBoost classifier. A normalization technique (see Appendix A) is presented which updates its estimate of the mean and variance of the data, adaptively. The updated mean and variance estimates can then be used for mean subtraction and variance normalization of the data. It is shown that, this improves the noise-robustness of the features. The major contribution of the thesis has been in bringing together feature extraction based physiological modeling and classification algorithms based on machine learning techniques. Functionalities from physiological processing are borrowed to improve the performance of state-of-the-art features in audio processing. In the past features based on auditory models have been limited by the classification algorithms available to process them. Herein, new developments in machine learning are leveraged to overcome this hurdle.

## 4.4 Future Work

The performance of new features have shown the value of the modifications introduced, in particular, varying the time constants and gain adaptation. Further work in this area would involve designing the optimal function for setting the time constants. Further, initial results have shown the usefulness of varying the time constants based on the SNR. This could be looked upon as extracting information at different levels of detail based on how reliably the detail can be extracted. For instance, humans are known to interpolate to complete partially heard words based on auxiliary information. An ASR parallel to this would be, extraction of information at a lower level of detail (presumably since the minute details would be smudged by noise) and letting the language model and pattern recognition interpolate the finer details.

The similarity between the gain function derived in this thesis and the Wiener gain was shown. This aspect of the work could be pursued further to explore the possibility of a more concrete relationship between Wiener filtering and automatic gain adaptation. The time varying aspect of the Wiener gain comes from the fact that instantaneous value of SNRs are used to compute the gain function. But in the gain function derived in this work, the time varying aspect is the result of the strength of the signal relative to the maximum signal strength. The fact that only a very rough estimate of the SNR is required to compute the gain function in the latter case is very appealing and certainly deserves more attention. Treating gain adaptation as a way of removing noise could lead to interesting findings relevant to speech enhancement and noise suppression.

The generative AdaBoost classifier, is very appealing due to its ability to provide a likelihood measure for each class. Further work in this regard would involve applying the classifier to the speech recognition problem in improving the observation probabilities. A robust theoretical framework for the KL divergence-based approach would also be part of the future work.

Some avenues that were not explored but seem very promising are cortical transform based SVMs and variance weighted AdaBoost. The idea of the former is to use the various modulation filters in the cortical transform as the feature mapping for a SVM. The CJSVM

would be the ideal vehicle to integrate the feature extraction and classification process into a single process.

The AdaBoost algorithm reweights the examples at each iteration, however there is no way to encode prior information about the relative reliability of features. As an example, for PCA transformed features one would expect the first component to be more reliable than, say, the tenth component. The integration of some measure of confidence about the features (i.e. enabling the algorithm to weight both the examples and the features) into the AdaBoost algorithm would lead to a more robust classifier.



## APPENDIX A

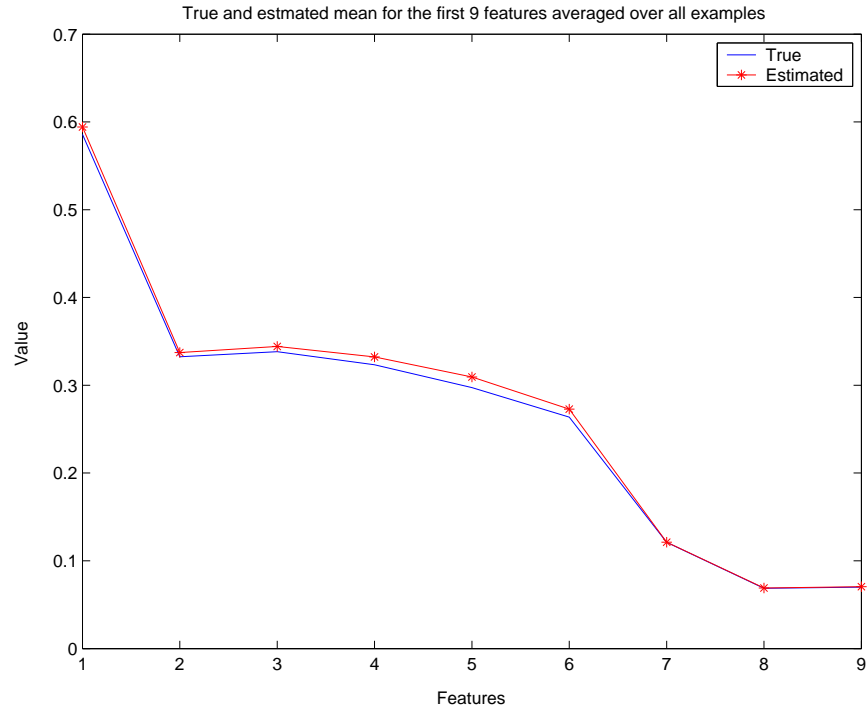
### ADAPTIVE STANDARDIZATION

Humans are very effective in learning the acoustic background and factoring it out of the audio stream of interest. This idea has been incorporated into pattern recognition by the introduction of “standardization” (i.e. mean subtraction and variance normalization) of input data. But most present standardization techniques are “static”, in that, they learn parameters from the training set and apply them to the test set. Taking cue from physiological processing (with respect to learning the changes in the background), we propose a new adaptive standardization technique to negate the effect of channel and environmental noise.

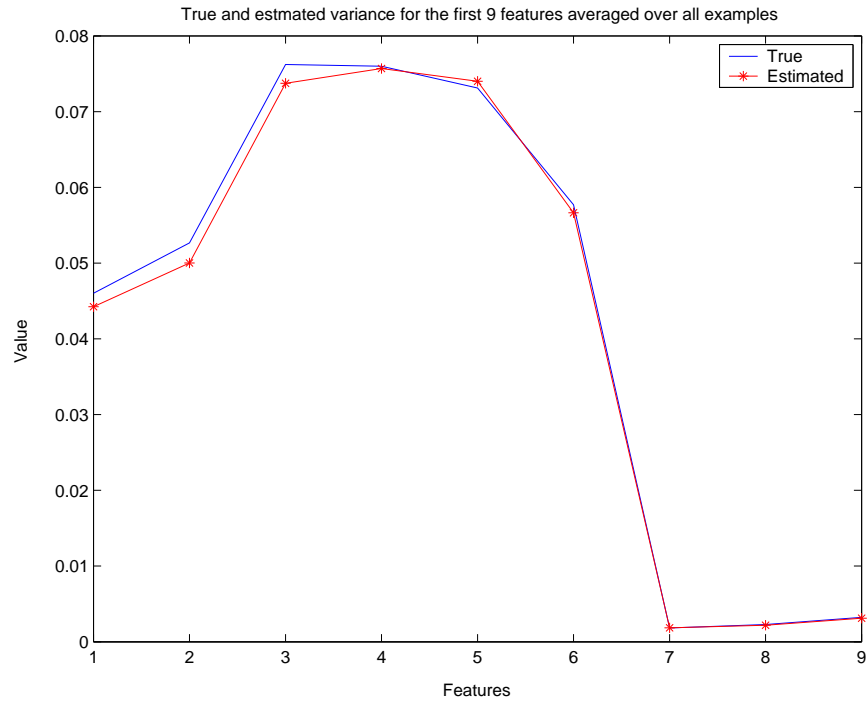
Given a data matrix (where each row is an example and each column is a variable), two kinds of standardization are common in pattern recognition. Per-example standardization amounts to centering (making zero-mean) and variance normalizing every row of the data matrix. Per-feature standardization amounts to centering and variance normalizing each column of the data matrix. In the speech recognition community it is common to perform such mean subtraction and variance normalization on a per utterance or segment basis i.e. over all the frames of the segment [43]. However in the audio classification domain where it is advantageous to form a feature vector for every segment by concatenating the mean and variance of the features over all frames of the segment [41], [26], it is not possible to do per-feature standardization on a segment basis. Hence, mean and variance extracted from the training set needs be used to standardize the test set. However, if the mismatch between train and test conditions is large then the standardization parameters obtained from the training set is no longer valid for the test set. Hence, these parameters need to be learnt adaptively from the test set. This is done by setting it up as an estimation problem and using a Kalman filter to estimate the parameters.

The process that generates the mean and variance is considered to be noisy. Fifty examples are used to calculate the mean and variance, and the difference between these values and the actual parameters (of the training set) is considered as the noise of the generation process. The noise estimated over different sets of 50 examples is used to calculate

the process noise variance. To estimate the measurement noise variance we calculate the mean and variance from all the data seen thus far (i.e. starting with one example and adding more examples, one at a time) and difference between these and the actual parameters is considered the measurement noise. Training data is used for system identification i.e. to find the covariance of the posterior error and to obtain an initial estimate of the parameters for the test case. During training the Kalman filter is initialized with random estimates of the initial mean and variance, and the filter improves its estimates with every iteration. Ten past examples are used to predict the prior estimate of the next value. When the first test data is presented to the Kalman filter it uses the parameters learned from the training phase as the initial estimate. The comparison between the true and estimated mean and variance is shown in Figs. 40(a) and 40(b). As the filter sees more and more data, the estimate gets better as seen from Fig. 41. The results of adaptive standardization for the speech versus non-speech discrimination task are shown in Tables 28-29. MFCC features and GMM-based classifiers were used. It is seen that adaptive standardization outperforms per-feature standardization.

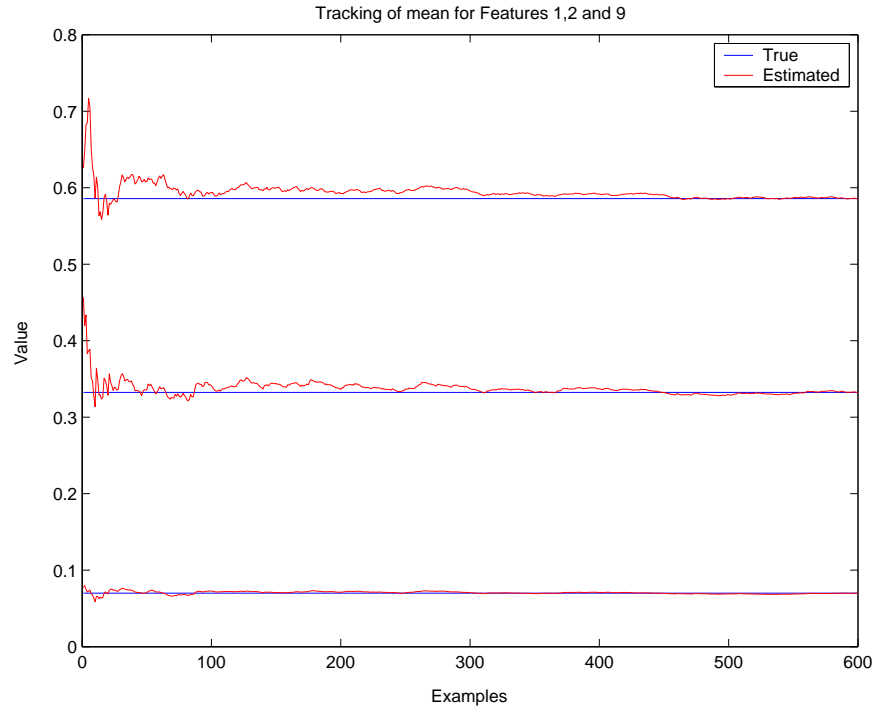


(a)

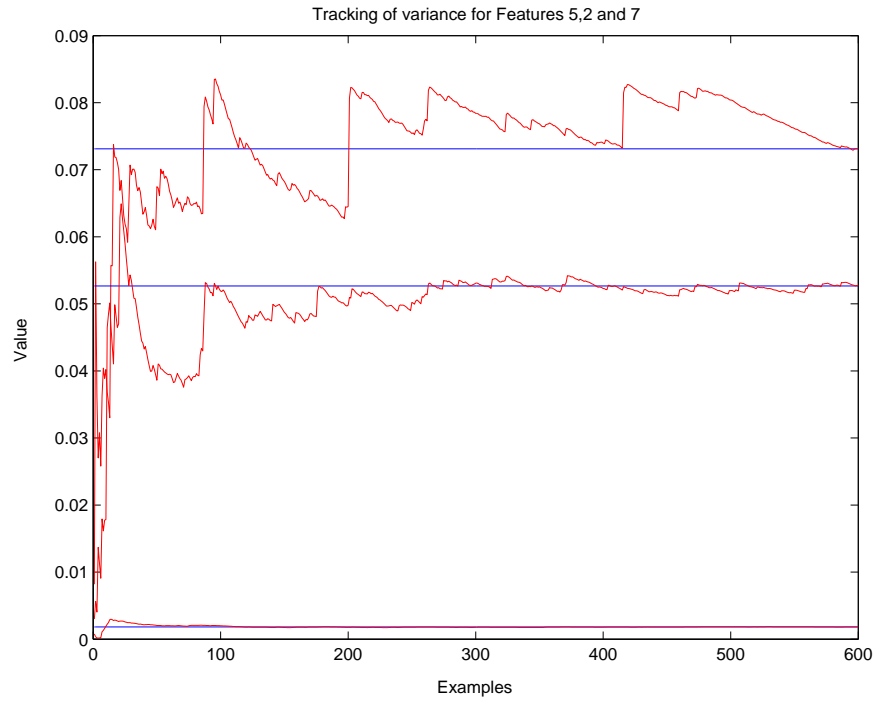


(b)

Figure 40. a) Shows the true mean of the test set (—) and the mean learned by the Kalman filter (— \* —) b) Shows the true variance of the test set (—) and the variance learned by the Kalman filter (— \* —). For the purposes of illustration only 9 features were chosen to do the adaptive standardization



(a)



(b)

**Figure 41.** a) Shows tracking of the true mean for 3 different features as a function of segments. b) shows tracking of the true variance for 3 different features. Blue indicates the true value and red indicates the learned value.

**Table 28.** The mean and variance of the test data is adaptively learned using a Kalman filter. A 4-mixture GMM was used for classification. MFCC features were used and white noise was synthetically added to generate the different SNR conditions.

Speech Discrimination using MFCC (with log compression)				
	Adaptive standardization		Per-feature standardization	
SNR	Hit Rate	Overall	Hit Rate	Overall
Clean	98.67 %	99.33 %	98.00 %	98.67
20 dB	98.67 %	99.00 %	98.00 %	98.33
10 dB	98.00 %	98.00 %	95.33 %	97.33
5 dB	98.00 %	98.00 %	91.33 %	95.33
0 dB	98.00 %	98.00 %	84.00 %	92.00

**Table 29.** The mean and variance of the test data is adaptively learned using a Kalman filter. A 4-mixture GMM was used for classification. MFCC features were used and pink noise was synthetically added to generate the different SNR conditions.

Speech Discrimination using MFCC (with log compression)				
	Adaptive standardization		Per-feature standardization	
SNR	Hit Rate	Overall	Hit Rate	Overall
Clean	98.67 %	99.33 %	98.00 %	98.67 %
20 dB	98.67 %	98.67 %	96.67 %	98.00 %
10 dB	98.00 %	98.00 %	96.00 %	97.67 %
5 dB	97.33 %	97.67 %	90.00 %	94.67 %
0 dB	94.67 %	96.33 %	66.67 %	83.33 %

## APPENDIX B

### DEADLOCK RESOLUTION USING A NORMALIZED MEASURE OF MARGIN

In the AdaBoost formulation presented in Table 15, the final decision is given by:

$$H(x) = \begin{cases} 1, & \text{if } \sum_1^T h_t \alpha_t \geq \frac{1}{2} \sum_1^T \alpha_t, \\ 0, & \text{otherwise} \end{cases}$$

where  $\alpha_t = \log \frac{1}{\beta_t}$ . Normalizing both sides by  $\sum_1^T \alpha_t$  it is easy to see that the decision boundary is at 0.5. If  $\frac{\sum_1^T h_t \alpha_t}{\sum_1^T \alpha_t} > \frac{1}{2}$  then the example belongs to class 1 and  $\frac{\sum_1^T h_t \alpha_t}{\sum_1^T \alpha_t} - \frac{1}{2}$  can be considered as the belief in the decision made. Similarly, if  $\frac{\sum_1^T h_t \alpha_t}{\sum_1^T \alpha_t} < \frac{1}{2}$ , then the example belongs to class 0 and  $\frac{1}{2} - \frac{\sum_1^T h_t \alpha_t}{\sum_1^T \alpha_t}$  can be considered as the confidence in the decision. Thus the margin of a decision can be expressed as:

$$M = \left| \frac{\sum_1^T h_t \alpha_t}{\sum_1^T \alpha_t} - \frac{1}{2} \right|$$

For the multi-class AdaBoost classifier that combines binary classifiers of the 1-versus-1 type, two way deadlocks (i.e. deadlocks where two classes get the same vote) can simply be resolved by using the head-to-head result between the two classes. For deadlocks involving more than two classes, the class with the highest margin is chosen as the winner.

## REFERENCES

- [1] J. Lazzaro and J. Wawrzynek, "Speech recognition experiments with silicon auditory models," 1997.
- [2] D. Graham and P. Hasler, "Capacitively-coupled current conveyer second-order section for continuous-time bandpass filtering and cochlea modeling," in *International Conference on Systems and Circuits*, vol. V, pp. 485–488, 2002.
- [3] K. Wang and S. Shamma, "Self-normalization and noise-robustness in early auditory representations," *IEEE Transactions on Speech and Audio Processing*, vol. 2, pp. 421–435, July 1994.
- [4] N. Mesgarani, S. Shamma, and M. Slaney, "Speech discrimination based on multi-scale spectro-temporal modulations," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, (Montreal, Canada), May 2004.
- [5] E. Kandel, J. Schwartz, and T. Jessel, *Principles of Neural Science*. New York: McGraw-Hill, 4th ed., 2000.
- [6] R. Berne and M. Levy, *Physiology*. New York: Mosby, 4th ed., 1998.
- [7] P. Denes and E. Pinson, *The Speech Chain*. Freeman, 2nd ed., 1993.
- [8] X. Yang, K. Wang, and S. Shamma, "Auditory representations of acoustic signals," *IEEE Transactions on Information Theory*, vol. 38, pp. 824–839, March 1992.
- [9] K. Wang and S. Shamma, "Spectral shape analysis in the central auditory system," *IEEE Transactions on Speech and Audio Processing*, vol. 3, pp. 382–395, Sept 1995.
- [10] L. Atlas and S. A. Shamma, "Joint acoustic and modulation frequency," *Eurasip Journal on Applied Signal Processing*, vol. 2003, pp. 668–675, June 2003.
- [11] G. White and R. Neely, "Speech recognition experiments with linear prediction, bandpass filtering, and dynamic programming," in *International Conference on Acoustics, Speech and Signal Processing*, vol. 24, pp. 183–188, Apr 1976.
- [12] C. Searle, J. Jacobson, and S. Rayment, "A phoneme recognition system based on human audition," in *International Conference on Acoustics, Speech and Signal Processing*, vol. 3, pp. 557–560, Apr 1978.
- [13] B. Kimberley and C. Searle, "Automatic discrimination of fricative consonants based on human audition," in *International Conference on Acoustics, Speech and Signal Processing*, vol. 4, pp. 89–92, Apr 1979.
- [14] B. A. Dautrich, L. R. Rabiner, and T. B. Martin, "On the use of filter bank features for isolated word recognition," in *International Conference on Acoustics, Speech and Signal Processing*, vol. 8, pp. 1061–1064, May 1983.

- [15] O. Ghitza, “Robustness against noise: The role of timing-synchrony measurement,” in *International Conference on Acoustics, Speech and Signal Processing*, vol. 12, pp. 2372–2375, Apr 1987.
- [16] C. Nadeu, J. Hernando, and M. Gorricho, “On decorrelation of filter-bank energies in speech recognition,” in *Proceedings of Eurospeech’95*, vol. 1, pp. 1381–1384, Sept 1995.
- [17] C. Nadeu, J. Marino, J. Hernando, and A. Nogueiras, “Frequency and time filtering of filter-bank energies for hmm speech recognition,” in *Fourth International Conference on Spoken Language*, vol. 1, pp. 430–433, Oct 1996.
- [18] K. K. Paliwal, “On the use of filter-bank energies as features for robust speech recognition,” in *International Symposium on Signal Processing and its Applications*, vol. 2, (Brisbane, Australia), pp. 641–644, Aug 1999.
- [19] V. Mantha, R. Duncun, Y. Wu, J. Zhao, A. Ganapathiraju, and J. Picone, “Implementation and analysis of speech recognition front-ends,” in *Proceedings of IEEE SoutheastCon*, pp. 32–35, Mar 1999.
- [20] T. Zhang and C. C. J. Kuo, “Hierarchical system for content-based audio classification and retrieval,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 3001–3004, 1999.
- [21] P. Gaunard, C. Mubikangiey, C. Couvreur, and V. Fontaine, “Automatic classification of environmental noise events by hidden markov models,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 3609–3612, 1998.
- [22] R. S. Goldhor, “Recognition of environmental sounds,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 149–152, 1993.
- [23] J. M. Kates, “Classification of background noises for hearing-aid applications,” *Journal of the Acoustical Society of America*, vol. 97, pp. 461–470, Jan. 1995.
- [24] S. Allegro, M. Buchler, and S. Launer, “Automatic sound classification inspired by auditory scene analysis,” in *Consistent and Reliable Acoustic Cues for Sound Analysis*, (Aalborg, Denmark), September 2001.
- [25] C. Ludvigsen, “Schaltungsanordnung für eine automatische regelung von hörhilfsgeräten,” in *Deutsches Patent Nr. DE 43 40 817 A1*, 1993.
- [26] V. Peltonen, J. Tuomi, A. Klapuri, J. Huopaniemi, and T. Sorsa, “Computational auditory scene recognition,” in *International Conference on Acoustics, Speech and Signal Processing*, (Orlando, Florida), May 2002.
- [27] M. Hunt, M. Lenning, and P. Mermelstein, “Experiments in syllable-based recognition of continuous speech,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, (Denver, CO), Apr. 1980.
- [28] S. Ravindran, D. V. Anderson, and M. Slaney, “Low-power audio classification for ubiquitous sensor networks,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, (Montreal, Canada), May 2004.



- [29] M. D. Skowronski and J. G. Harris, “Exploiting independent filter bandwidth of human factor cepstral coefficients in automatic speech recognition,” *Journal of Acoustical Society of America*, vol. 116, pp. 1774–1780, sept 2004.
- [30] R. Lyon, “A computational model of filtering, detection, and compression in the cochlea,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, (Paris, France), May 1982.
- [31] S. Beet, “Email contribution to auditory mailing list, Nov, 2004. <http://www.auditory.org/postings/2004/833.html>,”
- [32] L. Cohen, “The scale representation,” *IEEE Transactions on Signal Processing*, vol. 41, pp. 3275–3292, Dec 1993.
- [33] W. A. Yost, *Fundamentals of Hearing: An Introduction*. Academic Press, 2000.
- [34] P. Smith, M. Kucic, R. Ellis, P. Hasler, and D. V. Anderson, “Cepstrum frequency encoding in analog floating-gate circuitry,” in *Proceedings of the IEEE International Symposium on Circuits and Systems*, vol. IV, (Phoenix, AZ), pp. 671–674, May 2002.
- [35] S. Ravindran, C. Demiroglu, and D. Anderson, “Speech recognition using filter-bank features,” in *Asilomar Conference on Signals and Systems*, (Pacific Grove, CA), Nov. 2003.
- [36] B. E. Dom, “An information-theoretic external cluster-validity measure,” in *IBM Research Report RJ 10219*, May 2001.
- [37] S. Ravindran, D. V. Anderson, and M. Slaney, “Improving the noise robustness of mel-frequency cepstral coefficients,” in *IEEE Workshop on Statistical and Perceptual Audition*, (Pittsburgh), Sept. 2006.
- [38] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, “RWC music database: Music genre database and musical instrument sound database,” in *Proceedings of the 4th International Conference on Music Information Retrieval*, pp. 229–230, oct 2003.
- [39] M. Slaney, “Auditory toolbox, <http://www.slaney.org/malcolm/pubs.html>,”
- [40] C. P. Chen, K. Filali, and J. A. Bilmes, “Frontend post-processing and backend model enhancement on aurora 2.0/3.0 databases,” in *International Conference on Speech and Language Processing*, pp. 241–244, 2002.
- [41] S. Ravindran, K. Schlemmer, and D. V. Anderson, “A physiologically inspired method for audio classification,” *Eurasip Journal for Applied Signal Processing*, vol. 2005, July 2005.
- [42] E. E. . . v1.1.3 (2003-09), “Speech Processing, Transmission and Quality Aspects (STQ); Distributed speech recognition; Front-end feature extraction algorithm; Compression algorithms,”
- [43] C.-P. Chen, K. Filali, and J. A. Bilmes, “Frontend post-processing and backend model enhancement on the aurora 2.0/3.0 databases,” in *International Conference on Speech and Language Processing*, pp. 241–244, 2002.

- [44] P. Alexandre and P. Lockwood, "Root cepstral analysis: A unified view," *Speech Communication*, vol. 3, pp. 277–288, 1993.
- [45] J. Lim, "Spectral root homomorphic deconvolution system," *IEEE Trans. ASSP*, vol. 27, no. 3, pp. 223–233, 1979.
- [46] J. H. H. Ruhi Sarikaya, "Analysis of the root-cepstrum for acoustic modeling and fast decoding in speech recognition," in *Eurospeech*, (Aalborg, Denmark), Sept 2001.
- [47] T. J. and K. B., "A model of auditory perception as front end for automatic speech recognition," *Journal of Acoustical Society of America*, vol. 106, pp. 2040–2050, 1999.
- [48] D. V. Anderson, "Model based development of a hearing aid," in *M.S. Thesis, Brigham Young University*, (Provo, Utah), 1994.
- [49] D. M. Chabries, D. V. Anderson, T. G. S. Jr., and R. W. Christiansen, "Application of a human auditory model to loudness perception and hearing," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, May 1995.
- [50] W. Jeon, "Speech analysis and cognition using category-dependent features in a model of the central auditory system," 2006. PhD Thesis, Georgia Institute of Technology.
- [51] M. Kleinschmidt, "Methods for capturing spectro-temporal modulations in automatic speech recognition," in *Acustica united with acta acustica*, vol. 88(3), pp. 416–422, 2002.
- [52] T. Gramms and H. W. Strube, "Recognition of isolated words based on psychoacoustics and neurobiology," *Speech Commun.*, vol. 9, no. 1, pp. 35–40, 1990.
- [53] R. E. Schapire, "The boosting approach to machine learning: An overview," in *MSRI Workshop on Nonlinear Estimation and Classification*, 2002.
- [54] Y. Freund and R. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *Computational Learning Theory: Eurocolt '95*, pp. 23–37, 1995.
- [55] K. Tieu and P. Viola, "Boosting image retrieval," in *Proceedings of Computer Vision and Pattern Recognition*, vol. 1, pp. 228–235, 2000.
- [56] Y. Freund and R. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, pp. 119–139, Aug. 1997.
- [57] P. Viola and K. Tieu, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of Computer Vision and Pattern Recognition*, pp. 511–518, 2001.
- [58] M. Liu and C. Wan, "A study on content based classification and retrieval of audio database," in *IEEE International Database Engineering Application Symposium*, pp. 0339–03315, 2001.
- [59] M. C. Buchler, "Algorithms for sound classification in hearing instruments," PhD Thesis, 2002.

- [60] S. Ravindran and D. V. Anderson, "Cascade classifiers for audio classification," in *IEEE DSP Workshop*, (Toas Ski Valley, NM), Aug. 2004.
- [61] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances in Large Margin Classifiers*, P.B. Alexander, J. Smola, Bernhard Scholkopf and Dale Schuurmans, editors, 1999.
- [62] S. Rosset and E. Segal, "Boosting density estimation," in *Neural Information Processing Systems*, 2002.
- [63] G. Ridgeway, "Looking for lumps: Boosting and bagging for density estimation," in *Computational Statistics and Data Analysis*, vol. 38, pp. 379–392, Feb. 2002.
- [64] J. Klemela, "Density estimation with stagewise optimization of the empirical risk," (<http://www.rni.helsinki.fi/~jsk/ps/kitera.pdf>), 2005.
- [65] V. Vapnik, "Estimation of dependencies based on empirical data [*in Russian*]," 1979. English translation: Springer Verlag, New York, 1982.
- [66] R. Fletcher, *Practical Methods of Optimization*. John Wiley and Sons, 2nd ed., 1987.
- [67] C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.
- [68] S. Ravindran, D. V. Anderson, and J. Reh, "Cascade jump support vector machine classifiers," in *IEEE Machine Learning for Signal Processing*, (Mytsic, CT), Sept. 2005.
- [69] G. Fung and O. L. Mangasarian, "Proximal support vector machine classifiers," in *Knowledge Discovery and Data Mining*, (San Fransisco, CA), pp. 77–86, 2001.
- [70] J. Ma, Y. Zhao, and S. Ahalt, "OSU SVM classifier matlab toolbox (ver 3.00)," [http://www.ece.osu.edu/~maj/osu\\_svm/](http://www.ece.osu.edu/~maj/osu_svm/).
- [71] S. Ravindran and D. V. Anderson, "Boosting as a dimensionality reduction tool for audio classification," in *IEEE International Symposium on Circuits and Systems*, (Vancouver, Canada), May 2004.
- [72] P. Hasler and D. Anderson, "Cooperative analog-digital signal processing," in *IEEE International Conference on Accoustics, Speech, and Signal Processing*, (Orlando, FL), May 2002.
- [73] V. Srinivas, D. Graham, and P. Hasler, "Floating gate transistors for precision analog circuit design: an overview," in *48th Midwest Symposium on Circuits and Systems*, 2005.
- [74] V. Srinivas, G. Rosen, and P. Hasler, "Low-power realization of FIR filters using current-mode analog design techniques," in *Asilomar Conference on Signals, Sytems and Computers*, 2004.
- [75] P. Hasler, "Low-power programmable signal processing," in *Fifth International Workshop on System-on-Chip for Real-Time Applications*, 2005.
- [76] D. Graham, "Capacitively-coupled current conveyer second-order section for continuous-time bandpass filtering and cochlea modeling," 2006. PhD Thesis, Georgia Institute of Technology.