Summer 2005

Non-Linear and Linear Transformations of Features for Robust Speech Recognition and Speaker Identification

Saurabh Prasad
Old Dominion University

NON-LINEAR AND LINEAR TRANSFORMATIONS OF FEATURES FOR ROBUST SPEECH RECOGNITION AND SPEAKER IDENTIFICATION

by

Saurabh Prasad
B.TECH(EE). June 2003, J.M.I., New Delhi, INDIA

A Thesis Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirement for the Degree of

MASTER OF SCIENCE

ELECTRICAL AND COMPUTER ENGINEERING

OLD DOMINION UNIVERSITY
August 2005

Approved by:

Stephen A. Zahorian (Director)

W. Steven Gray (Member)

Vijayan K. Asari (Member)

ABSTRACT

NON-LINEAR AND LINEAR TRANSFORMATIONS OF SPEECH FEATURES FOR ROBUST RECOGNITION AND SPEAKER IDENTIFICATION

Saurabh Prasad
Old Dominion University, 2005
Director: Dr. Stephen A. Zahorian

Automatic speech recognizers perform poorly when training and test data are systematically different in terms of noise and channel characteristics. One manifestation of such differences is variations in the probability density functions (pdfs) between training and test features. Consequently, both automatic speech recognition and automatic speaker identification may be severely degraded. Previous attempts to minimize this problem include Cepstral Mean and Variance Normalization and transforming all speech features to a uni-variate Gaussian pdf. In this thesis, two techniques are presented for non-linearly scaling speech features to fit them to a target pdf – the first is based on the principles of Histogram matching (a commonly employed algorithm in image contrast enhancement applications) and the second is based on principles of quantile based Cumulative Density Function (CDF) matching for data drawn from different distributions. These methods can be used to compensate for the systematic marginal (i.e. each feature considered individually) differences between training and test features. For a more complete, multi-dimensional restoration of feature statistics, a linear (matrix) transformation is proposed, mapping the noisy feature space to the corresponding clean space. The matrix used for this global transformation is learned in a least squares sense from stereo training data – comprised of speech recorded simultaneously in clean and noisy conditions. We further propose a linear covariance

normalization technique to compensate for differences in covariance properties between training and test data. Experimental results are given that illustrate the benefits of these algorithms for speech recognition and automatic speaker identification.

This thesis is dedicated to my parents and my brother.

ACKNOWLEDGEMENT

This thesis would not have been possible without the support and encouragement of my advisor, Dr. Stephen A. Zahorian. I joined the speech communications Laboratory as a research assistant with much enthusiasm in pursuing graduate research in the area of digital and statistical signal processing but little background in speech recognition. Dr. Zahorian welcomed me into his group and gave me an opportunity to grow as a researcher and learn along the way.

I would also like to thank members of the Speech Communications Laboratory – the two years I spent with the group taught me much.

I am also grateful to the members of the faculty – Dr. W. Steven Gray and Dr. Vijayan K. Asari for giving their consent to be on the thesis advisory committee.

Last but not least, I would like to thank my parents without whom none of this would be possible. They have always been a bountiful source of inspiration for me. I thank them for instilling in me love of knowledge. Their boundless love and support has been a pillar to me.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Automatic Speech Recognition (ASR) is a classical pattern recognition problem. The first step towards building a speech-recognizer involves extracting features from spoken data that possess reasonably discriminating information with respect to the basic units of speech being recognized, and are also least-sensitive to noise and channel effects. This is followed by a learning phase, in which labeled data is used to build a statistical model for each category (basic speech unit to be recognized in our case). This statistical model provides us with a framework to make a labeling decision on a new unlabeled speech utterance based on some metric (the posterior probability, in most cases). Average vocabulary sizes for a large unconstrained continuous speech recognition system now exceed 65,000 words. Fast recognition algorithms allow continuous speech recognition systems to work in real time.

However, a big fundamental issue still remains for ASR – successful commercial recognizers still work primarily in a restricted application-specific environment. Humans still greatly outperform computer-based recognition in their ability to successfully recognize spoken language, especially from noisy and/or barely intelligible speech. Most current research in speech recognition focuses on building recognition systems that have the ability to recognize speech in the presence of channel distortion, noise and speaker variability. It is important to have a good recognizer that is robust to sources of error and a good front-end (feature extraction and preprocessing) that removes the channel and noise effects from distorted speech to provide clean data to the back-end

---

The journal model used is *IEEE Transactions on Speech and Audio Processing.*

for recognition. Large vocabulary recognition systems achieve word error rates of less than 10% for clean speech. Machine recognition, however degrades dramatically in noisy conditions. On the other hand, human recognition results obtained with channel variability and noise show that people can easily recognize speech with normally occurring degradations.

A Robust Speech Recognition system consists of a classifier that gives consistently good recognition performance in different acoustical environments. As an illustration, consider an ASR system that was trained on clean studio speech. It will work well when it is used for recognizing any speech recorded in a clean environment. Whether it will work well for speech recorded with additive noise and channel distortion (e.g. telephone quality) is what determines whether the classifier is robust or not. As the basic speech recognition algorithms and frameworks reach maturity, the next important issue at hand for researchers is robustness. "Robustness" as interpreted in the speech recognition literature has come a long way from 'Recognition in a quiet room using desktop microphones' in the mid 1980's to 'Recognition over a cell phone in a car being driven at highway speeds, with the windows rolled down and the radio playing' in the 2000's [1].

A speech recognizer trained and tested with speech at the same SNR typically performs well. However, situations where the recognizer is trained with clean speech and used for recognizing noisy speech are commonly encountered and generally result in greatly degraded performance or lack of robustness. Further, the noise that degrades the speech quality is typically non-stationary. This makes the modeling of the environmental degradation function difficult. It will be shown in chapter 2 that even a simple distortion

model comprised of stationary noise and linear channel distortion (i.e., filtering) transforms the feature space in a highly non-linear way. This makes the process of restoring clean speech features from noisy features mathematically intractable. In recent years, several pre-processing front-ends have been proposed that normalize the feature space to partially compensate for the non-linear effects of noise and channel distortion.

This thesis documents work done with the objective of designing algorithms that restore the statistics of the features extracted from noisy speech by appropriate pre-processing.

1.1 Outline of work

Our contribution towards robust speech (/speaker) recognition consists of using a combination of multi-dimensional linear (matrix) and one-dimensional non-linear (scalar) transformations to obtain a normalization of the distorted feature space to a defined / learned target density. Unlike previous work in the area of cepstral normalization, in the work presented in this thesis, marginal non-linear transformations are combined with a linear transformation that results in a multi-dimensional normalization. Some specific objectives of our work are as follows:

- A simple marginal transformation technique (**Q**uantile based **C**DF **M**atching - QCM) is proposed, allowing us to transform the incoming speech features to have a previously learned or defined probability density function (pdf). The benefit of learning the pdfs of clean speech features from training data and matching the features of the test data to these learned pdfs (instead of restricting the normalization to a global single-mode Gaussian) is illustrated. The effect of

incorporating PCA (allowing us to de-correlate the feature space) with marginal normalization is also illustrated - this leads to uncorrelated features during training and testing.

- As a further refinement, a linear (matrix) transformation (**C**ross-feature **L**inear **N**ormalization - CLN) is proposed which forces the collected data vectors to possess a specific covariance matrix (given by the covariance matrix of the clean speech features from the training data-set). This leads to a complete multi-dimensional statistical conditioning – not only restoring the uni-variate statistics of each feature, but also compensating for distortions and rotations in the equi-probable contours of the categories being recognized.
- The idea of using a simple linear Least Squares based estimator is also explored for determining a transformation matrix mapping the noisy feature space to the corresponding clean space for removing noise and channel mismatch and test this technique separately and in conjunction with QCM.
- The above mentioned algorithms were tested on a variety of speech classification tasks including phonetic classification and speaker identification. The effects of the amount of training used for computing the above transformations on the quality of restoration provided by them were tested experimentally. An analysis of the best possible ways to combine these transformations for obtaining the best possible multi-dimensional normalization is given.

Chapter 2 describes the effect of a linear degradation model on speech features and motivates the need for robustness in speech recognition front-ends. It also summarizes previous related work of feature transformations intended to improve robustness of

classifiers. In Chapter 3, the front-end used in this work is described and the derivation is given for the algorithms for non-linearly scaling each feature individually. This derivation is followed by a derivation of the method for linearly rotating and warping the feature space, with the overall goal of compensating for statistical mismatches between training and test conditions. Chapter 4 describes the various experiments performed to test the classifier in various test conditions, using different classifiers. Chapter 5 concludes this thesis with a discussion on future work and contributions that can be appended to the work described in this thesis.

CHAPTER II

NEED FOR ROBUSTNESS – EFFECTS OF NOISE AND CHANNEL DISTORTION ON SPEECH FEATURES

The effects of noise and linear channel distortions on robust speech recognition have been documented in various studies in the past [2], [3]. It has been shown that the feature space is non-linearly distorted by a simple environmental model consisting of additive noise and linear distortion. In the mismatched scenario, where the classifier is trained with clean speech and tested on noisy speech, the parameters of the trained system are not representative of the noisy speech. It has been shown that compensation for the effects of noise on the first two moments of the corrupted speech improves the recognizer's performance [4], [5].

2.1 Effects of noise and linear channel distortion on MFCCs

The environment model used for describing the effects of speech and channel distortion is shown in fig. 2.1.

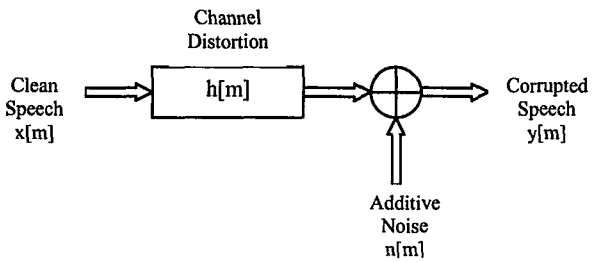


Fig. 2.1: The Environment Model

In the time domain, linear filtering and additive noise leads to

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

In the frequency domain, this can be expressed as

$$|Y(f_i)|^2 = |X(f_i)|^2 |H(f_i)|^2 + |N(f_i)|^2 + 2\mathrm{Re}\{X(f_i)H(f_i)N^*(f_i)\} \tag{2.2}$$

Assuming that the noise and speech are uncorrelated, when using a filter-bank, then

$$|Y(f_k)|^2 \approx |X(f_k)|^2 |H(f_k)|^2 + |N(f_k)|^2 \tag{2.3}$$

where $Y(f_k)$, k=0,1,2..., K represents the energy at each of the K filters [4]

To generate the log spectral coefficients from this representation, take the logarithm of both sides, resulting in

$$10\log_{10}\{|Y(f_k)|^2\} = 10\log_{10}\{|X(f_k)|^2 |H(f_k)|^2 + |N(f_k)|^2\} \tag{2.4}$$

For convenience, denote these log-energy coefficients for clean speech, noise and channel impulse response as

$$\begin{aligned} x[k] &= 10\log_{10}\{|X(f_k)|^2\} \\ y[k] &= 10\log_{10}\{|Y(f_k)|^2\} \\ n[k] &= 10\log_{10}\{|N(f_k)|^2\} \\ h[k] &= 10\log_{10}\{|H(f_k)|^2\} \end{aligned} \tag{2.5}$$

Using these in equation 2.4 results in

$$10\log_{10}\{|Y(f_k)|^2\} = 10\log_{10}\{10^{\frac{x[k]+h[k]}{10}} + 10^{\frac{n[k]}{10}}\} \tag{2.6}$$

$$y[k] = x[k] + h[k] + 10\log_{10}\{1 + 10^{\frac{n[k]-x[k]-h[k]}{10}}\} \tag{2.7}$$

This can also be expressed in the vector form as

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{h} + \boldsymbol{g}(\boldsymbol{n} - \boldsymbol{x} - \boldsymbol{h}) \tag{2.8}$$

where $\boldsymbol{g}(\boldsymbol{n} - \boldsymbol{x} - \boldsymbol{h})$ represents the non-linear function $10\log_{10}\{1 + 10^{\frac{n[k]-x[k]-h[k]}{10}}\}$.

This illustrates that our relatively simple model of additive noise and linear channel distortion non-linearly distorts the feature space.

To visually comprehend the effect of this distortion, consider a one-dimensional feature space. The effect of simply adding noise to this representation is illustrated in the following figures.

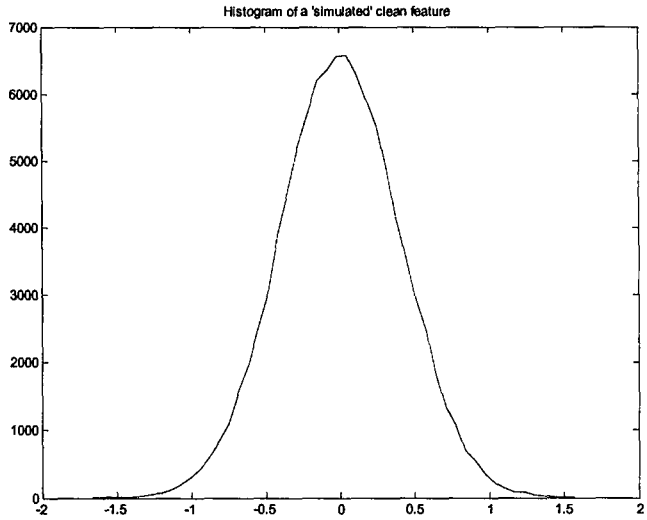


Fig. 2.2: Histogram representation of a 'simulated' log-energy feature of a clean signal (with a Gaussian density with a mean of 0 and standard deviation of 1)

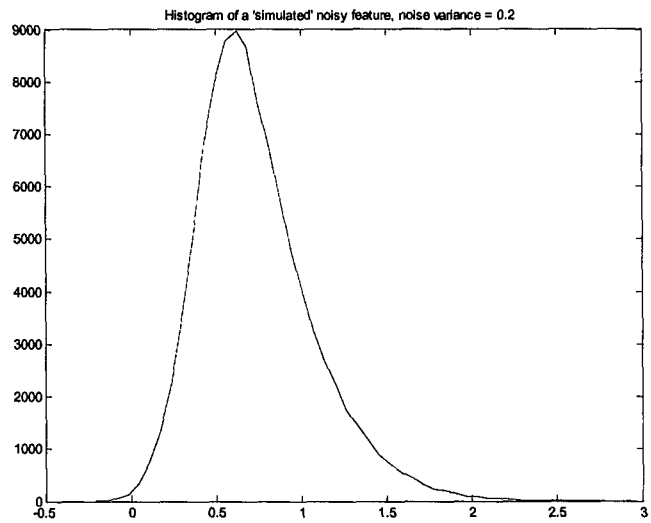


Fig. 2.3: Histogram envelope of same feature as shown in Fig. 2.2, except the effect of additive noise of mean 0 and variance 0.2 mixed with the input signal is included in the simulation.

Figures 2.2 and 2.3 illustrate the effect of noise on a single feature (assuming the log-spectral vectors representing clean speech follow a Gaussian density). The extent of this distortion increases significantly as the noise variance is increased and linear filtering (e.g. a telephone line's impulse response) is introduced.

2.2 Modeling the effects of noise and channel distortion on probability models

Given the non-linear relation between the clean and noisy speech vectors (equation 2.8), restoring the statistics of the clean speech from the noisy version is itself not a trivial problem. To add to this, the impulse response of the environment model is almost never known thus precluding the analytic restoration approach.

Let us assume that the speech feature vectors are modeled by a multivariate Gaussian density $N_x(\mu_x, \sum_x)$. The discussion can be extended to a more general, Gaussian Mixture Model (GMM). Also assume that the additive noise is stationary and uncorrelated with the speech data. Using the basic concepts of transformation of random variables [9] the pdf of the noisy speech vectors y can be related to the clean speech pdf as [3]

$$p(\boldsymbol{y}\,|\,\boldsymbol{x},\boldsymbol{n},\boldsymbol{h}) = \{(2\pi)|\boldsymbol{\Sigma}_x|^{L/2}\left|(\mathbf{I}-10^{\frac{\boldsymbol{n}-\boldsymbol{y}}{10}})\right|\}^{-1}\exp\left(-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{h}-\boldsymbol{\mu}_x+10\log_{10}(\boldsymbol{i}-10^{\frac{\boldsymbol{n}-\boldsymbol{y}}{10}}))^T\boldsymbol{\Sigma}_x^{-1}(\boldsymbol{y}-\boldsymbol{h}-\boldsymbol{\mu}_x+10\log_{10}(\boldsymbol{i}-10^{\frac{\boldsymbol{n}-\boldsymbol{y}}{10}})\right) \tag{2.9}$$

The desired density is clearly non-Gaussian (as exemplified by figure 2.3). This also shows that a restoration algorithm that tries to restore the statistics of the noisy, distorted speech will not be mathematically tractable. For a simpler analysis, instead of studying the effect of noise and distortion on the pdf, let us analyze the effect on just the first moment and the second central moments.

The expression for the mean vector will then be given by

$$\begin{aligned}\boldsymbol{\mu}_y &\triangleq E[\boldsymbol{y}] \\ &= E[\boldsymbol{x}+\boldsymbol{h}+\boldsymbol{g}(\boldsymbol{n}-\boldsymbol{h}-\boldsymbol{x})] \\ &= \boldsymbol{\mu}_x + \int_X \boldsymbol{g}(\boldsymbol{n}-\boldsymbol{h}-\boldsymbol{x}) N_x(\boldsymbol{\mu}_x, \Sigma_x) d\boldsymbol{x}\end{aligned} \tag{2.10}$$

Similarly, the covariance matrix of the noisy speech vector can be expressed as

$$\begin{aligned}\Sigma_y &= E[(\boldsymbol{x}+\boldsymbol{g}(\boldsymbol{x},\boldsymbol{h},\boldsymbol{n}))(\boldsymbol{x}+\boldsymbol{g}(\boldsymbol{x},\boldsymbol{h},\boldsymbol{n}))^T] - \boldsymbol{\mu}_y\boldsymbol{\mu}_y^T \\ &= \Sigma_x + \boldsymbol{\mu}_x\boldsymbol{\mu}_x^T - \boldsymbol{\mu}_y\boldsymbol{\mu}_y^T + \int_X \boldsymbol{g}(\boldsymbol{n}-\boldsymbol{h}-\boldsymbol{x})\boldsymbol{g}(\boldsymbol{n}-\boldsymbol{h}-\boldsymbol{x})^T N_x(\boldsymbol{\mu}_x, \Sigma_x) d\boldsymbol{x} \\ &+2\int_X \boldsymbol{x}.\boldsymbol{g}(\boldsymbol{n}-\boldsymbol{h}-\boldsymbol{x})^T N_x(\boldsymbol{\mu}_x, \Sigma_x) d\boldsymbol{x}\end{aligned} \tag{2.11}$$

The integrals in equations 2.10 and 2.11 do not have a closed form expression. Numerical methods are needed to solve the above equations to estimate the new mean vector and covariance matrix. Hence, any environmental compensation technique using the above equations requires a good model for the noise and channel distortion and then numerical methods.

2.3 Robust Classification

The changes in the statistics of clean speech caused by environmental degradation has an adverse effect on the performance of the pattern recognition backend of a speech recognition system, especially in cases where the classifier makes implicit assumptions about the feature pdfs, based on clean speech features. To illustrate this issue, consider fig. 2.4, representing a classifier that has two categories and a two-dimensional feature space. Further assume that each category is best represented by a Gaussian density of a certain mean vector and an identity covariance matrix. Based on the appropriate distance measure, the classifier learns the boundary from the clean training samples. However,

consider the case when this classifier is used for classifying noisy and distorted data. A quick glance at equations 2.10 and 2.11 show that the mean vectors and the covariance matrices of the data being recognized will no longer be equal to what the classifier's boundaries were learned from (In fact, the means will shift and the covariance matrix will no longer be identity, which represent shifts and rotations of the class-distributions). This obvious mismatch will introduce classification errors in the recognition system because the decision boundaries are no longer representative of the data that is being classified.

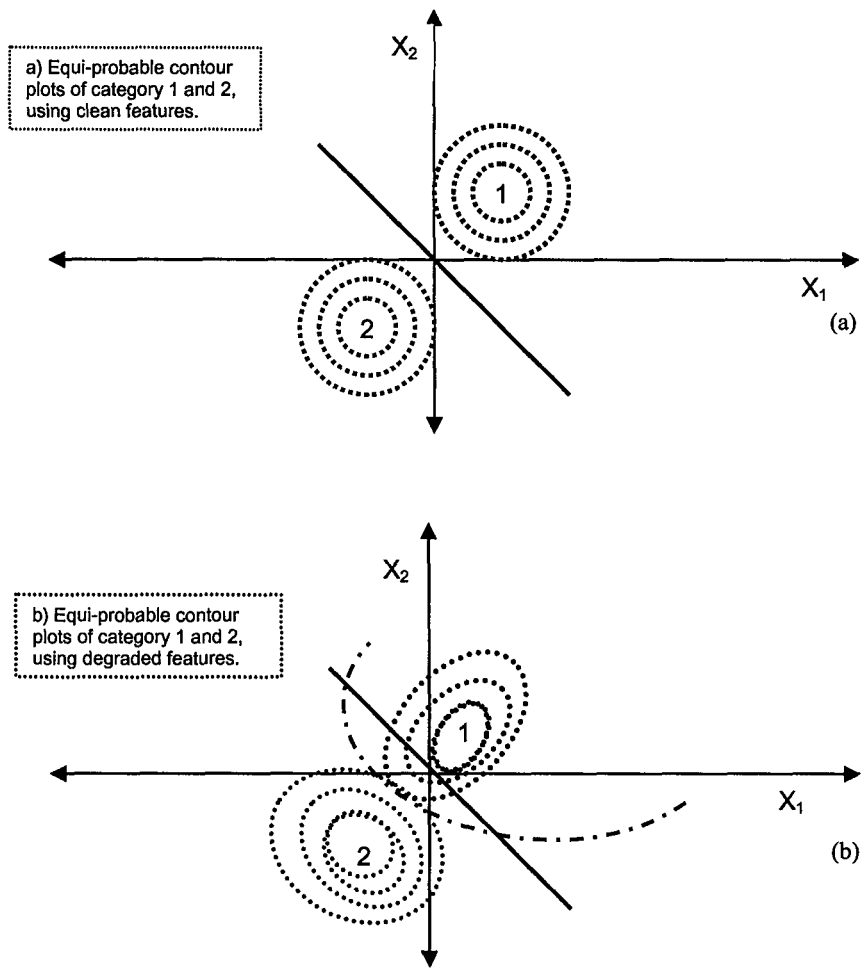


Fig. 2.4: A stylistic illustration of a binary classification example - Equi-probable contour plots using (a) Clean features, (b) Distorted features. (Solid Red Line: Optimal boundary for clean features; Dashed Blue Line: Optimal boundary for distorted features)

2.4 Previous work

The basic problem of reducing noise and channel effects for robust recognition can broadly be solved using two techniques – restoration and normalization.

Normalization deals with conditioning the statistics of the data as a whole before training the classifier and before using it for recognizing test data. It has been shown that compensation for the effects of noise on the first two moments of the corrupted speech improves the recognizer's performance [4]. As shown by equations 2.10 and 2.11, channel distortion and noise change the overall mean and covariance matrix of the data. Cesptral Mean Normalization (CMN) consists of subtracting the global mean of the feature space from each feature vector. It has been found [6] that this mean subtraction compensates significantly the mismatch induced by the channel / handset distortion and noise. With a similar motivation, Cepstral Variance Normalization (CVN) consists of normalizing each feature to a standard variance by an appropriate scaling factor (learned from training data). The most recent work in this area [5], [7] involves global Gaussianization. In this work, an algorithm has been proposed that equalizes the histogram of the speech features to have a Gaussian density for each feature, over the entire data-base – both in the training and test phase. Using an appropriate non-linearity, each feature has been transformed to have a Gaussian histogram with a zero mean and unit standard deviation over each sentence (the histogram used for obtaining the non-linear transformation was built with data from 1 sentence). The ASR performance after this transformation has been compared to linear transformations involving the first two moments of the features – Cepstral Mean and Variance Normalization (CMVN). In all the experiments, histogram equalization performed better than the mean and variance

normalization, except at high SNRs where all the front-ends gave reasonably similar performance. Though the results of this work were promising, there is still significant room for improvement. The authors acknowledged the limitations of performing marginal Gaussian transformations over the entire sentence. In [8] a modification of the above approach is proposed, wherein a global linear transformation of the data is followed by a non-linear transformation (Gaussianization), using only a short-time interval for computing the transformation.

Restoration on the other hand deals with using Minimum Mean Squared Error (MMSE) techniques for estimating clean speech features given noisy speech features. Researchers at Carnegie Mellon University have worked intensively in this area [3]. The basic approach in this direction involves taking a MMSE estimate of the clean speech given the degraded speech. This involves finding an estimate $\hat{x}$ minimizing $E[(x-\hat{x})^2 \mid z]$, where $\hat{x}$ is the estimate of the clean speech feature given the noisy speech feature z. Their algorithms begin with an initial guess of the channel noise and degradation impulse response and run the Expectation Maximization (EM) algorithm to obtain an accurate estimate of x in the mean squared sense. It has been shown that a linear model serves as a good approximation of the environmental degradation function. This allows using a simpler realization of the algorithm, with fewer parameters to solve for.

It is also important at this point to make a distinction between our non-linear and linear statistical restoration technique from previously proposed speech enhancement systems based on Wiener filtering [4]. Speech enhancement systems based on Wiener filtering require prior knowledge of the power spectral density of clean speech and the

corrupting process. This is a severe limitation especially when the corrupting noise or distortion is unknown. The non-linear and linear statistical restoration techniques discussed in this thesis do not require a prior knowledge of the corrupting process. In-fact, the algorithms discussed in the next chapter learn the appropriate restoration transformation from a small amount of noisy training data. This can be provided for in real-time systems by making the speaker speak a training sentence to measure the statistics of the speech from the speaker in the current environment / handset conditions.

In Chapter 3, a discussion is presented on the algorithms proposed by us for the above mentioned pre-processing. Two different techniques to non-linearly process data for pdf-matching are described, followed by two different linear (matrix multiply) transformations for restoring the cross-feature statistical properties. A discussion on various algorithm implementation issues encountered is also provided where appropriate.

CHAPTER III

ALGORITHMS FOR LINEAR AND NON-LINEAR TRANSFORMATIONS OF SPEECH FEATURES

In this chapter, the various pre-processing steps used in the design and implementation of our proposed front-end that equalizes the feature space to a 'learned' or 'pre-defined' pdf are presented. First, single feature non-linearities that transform speech features marginally to have desired pdfs are presented. To accomplish this, two algorithms are presented– 'histogram matching' and 'CDF matching.' The next step towards achieving multi-variate pdf normalization is to perform a linear transformation on these speech features that restores cross-feature statistical properties. To accomplish this multivariate normalization, two separate linear transformations are proposed. Experimental results illustrating the effects of these non-linear and linear transformations on the recognition performance of various speech recognizers under different test conditions are presented and discussed in the next chapter.

3.1 Histogram matching – A 'transformation of random variables' problem

The non-linear transformation used to make the pdfs of the features Gaussian is based on the principle of histogram matching. This is a popular technique used in image contrast enhancement applications, mapping probability densities of pixel intensities to a reference shape. In this section, concepts leading to this technique are discussed, and an efficient way to implement this algorithm is proposed without using elaborate look-up tables.

The underlying theory is based on the transformation of random variables – given a random variable x with a pdf $p_x(x)$, an appropriate transformation $z = T(x)$, such that z has a desired pdf $p_z(z)$, must be found. The pdfs of x and z in this case can be related [9] by

$$p_z(z) = p_x(x)\left|\frac{dx}{dz}\right| \tag{3.1}$$

For the case where $p_z(z)$ is uniform (Histogram Equalization), and assuming the original density $p_x(x) = 0$ for $x < 0$), the transformation T(x) is given by

$$z = T(x) = \int_0^x p_x(w)dw \tag{3.2}$$

For the case of feature data, a transformation algorithm can be derived as follows:

1. 'Equalize' the levels of the input feature-data using the transformation described above.

   $$u = T(x) = \int_0^x p_x(w)dw \tag{3.3}$$

   The random variable 'u' is then uniformly distributed and is used in step 3. In a practical implementation, since $p_x(w)$ is not known, a normalized histogram of the data is used to approximate the pdf $p_x(w)$. Further, the integral in Eq 3.3 can be approximated by a summation, using the normalized histogram, and therefore a stepwise continuous approximation of T can be obtained. This transformation can be stored in a lookup table 'u' vs. 'x' and then used in step 3 below.

2. Specify the desired density function $p_z(z)$ and obtain the transformation function

$$v = G(z) = \int_0^z p_z(w)dw \tag{3.4}$$

such that the random variable, 'v' is also uniformly distributed. Once again, if $p_z(z)$ is not known as a function, this transformation can be derived with discrete data by replacing the integration with a summation of the normalized histogram over the bins. This transformation can be stored in a lookup table 'v' vs. 'z'.

3. Apply the inverse transformation function $z = G^{-1}(u) = G^{-1}(T(x))$ to the values obtained in step 1. The resulting random variable z will have the desired density function $p_z(z)$. With discrete data, this will successfully 'match' the target histogram to the data

It is straightforward to derive an analytic expression for the transformation for cases when the desired density function is Exponential, Laplacian, Uniform etc. However, when the desired density function is Gaussian, the transformation in step 2 of the above algorithm cannot be analytically expressed. A good approximation to the function G^{-1} for the case of a Gaussian desired pdf is given by Marsaglia [10]. Using Polya's approximation to the normal he proposed the following analytic expression for generating a standard normal random variable **z** (density $ce^{-z^2/2}, 0 < z < 1$) from a uniform random variable:

$$\mathbf{z} = -[1.553\ln(1-\mathbf{u}^2)]^{1/2} \tag{3.5}$$

where **u** is uniform between 0 and 1. This approximation is used for performing step 3 of the non-linear transformation. Previous implementations of the histogram matching techniques for cepstral normalization employ elaborate look-up tables and

computationally costly Monte Carlo simulations. Marsaglia's result allows us to perform histogram matching in two easy steps

- Equalizing (to a uniform density) the histogram of the data to be transformed,
- Applying equation 3.5 on the uniformly distributed data.

For any target pdf, the corresponding transformation can be obtained using histogram matching by following the series of transformations given by steps one through three described above.

It is important to note that the quality of the Gaussian density function achieved by applying Polya's approximation to uniformly distributed data is sensitive to the deviation of the equalized data from the uniform density, in the first step. Thus, if the equalized data is not highly uniform, the resultant histogram will show significant deviation from Gaussian form. Hence, the uniform equalization must be done with a large number of bins to ensure a smooth Gaussian pdf of the overall transformed data.

Figure 3.1 illustrates this procedure for a case when the original data is approximately uniform, and the target density is Gaussian. For this example, the transformation T (equation 3.3) was not needed, since the original feature was assumed to be uniform. Fig. 3.1a shows the original histogram, fig.3.1b shows the non-linearity used based on Polya's approximation and fig. 3.1c shows the histogram of the data after this transformation.

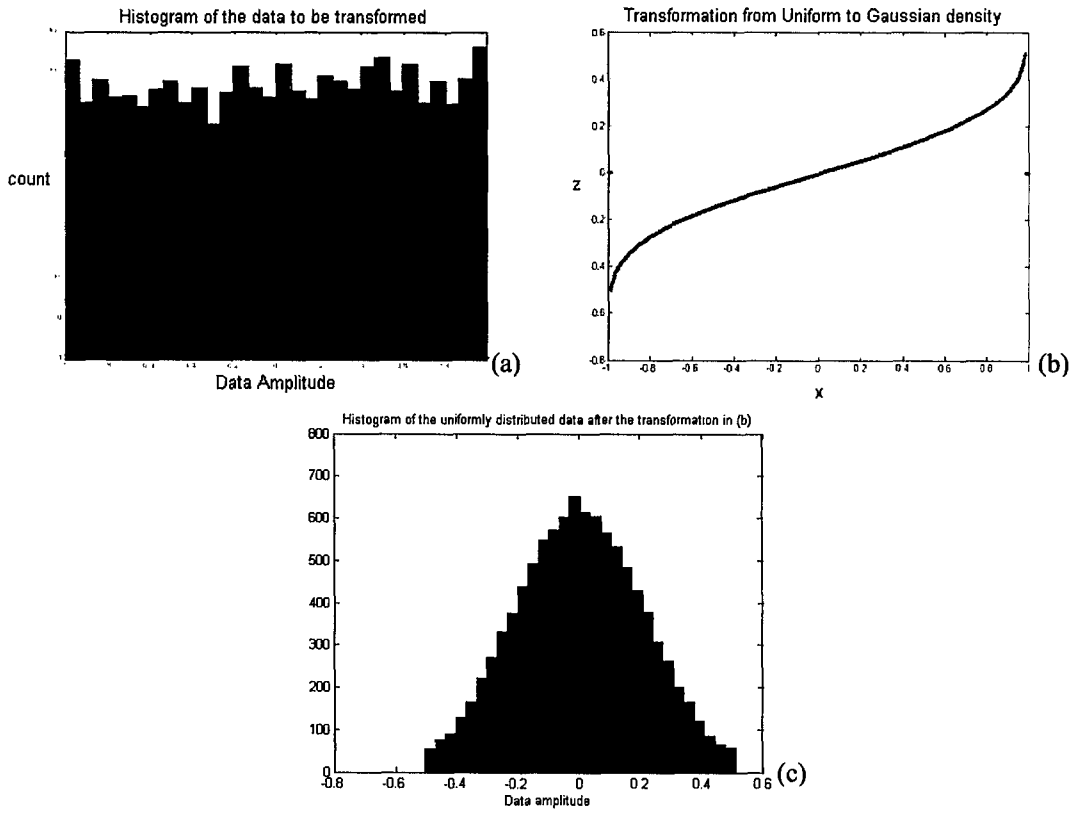


Fig. 3.1: Transformation from uniformly-distributed in (a) to a Gaussian distributed data in (c) using the non-linearity in (b) derived using histogram matching.

3.2 Quantile based Cumulative Density Function matching (QCM)

Another approach for cepstral density normalization uses a Cumulative Density Function (CDF) matching technique. In this technique, a target (reference) CDF is created, and a transformation is found that forces the data to be distributed according to this target CDF. The algorithm can be described as follows.

- Break up the target CDF into N equi-probable bins (quantiles), i.e., bin sizes for which the probabilities of the random variable taking on values in each bin are equal. To achieve this, simply break up the target CDF into bins corresponding to equal differences on the CDF-axis, resulting in bins that are non-uniformly spaced along the data axis. An example illustrating this process is shown in Fig. 3.2.

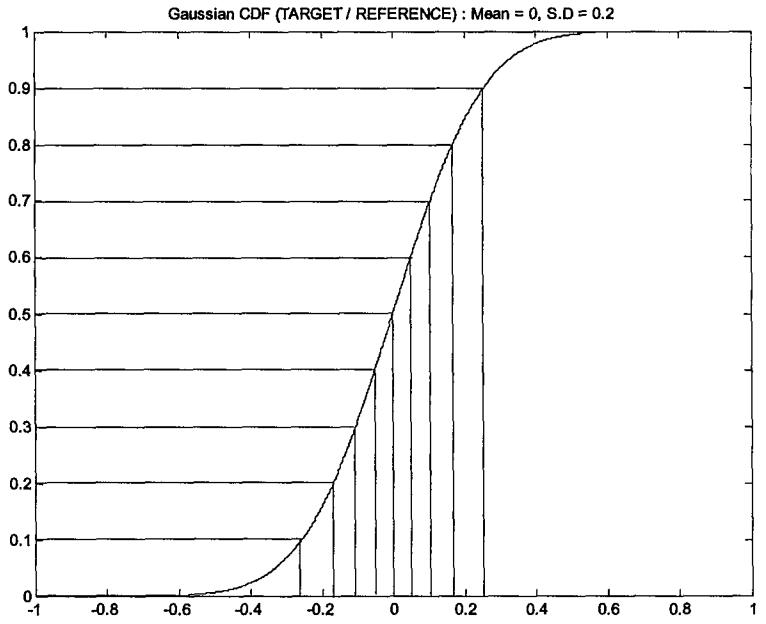


Fig. 3.2: A sample target CDF (Gaussian, in this case) is broken up into 10 equi-probable bins. That is, if data were distributed according to this CDF, the expected value of the number of data samples would be the same for each bin.

As can be seen from Fig. 3.2, the bins are very narrow around the mean and become wider towards the tails of the Gaussian distribution. This essentially emphasizes the fact that Gaussian distributed data will have larger probabilities around the mean and smaller probabilities towards the tails (and hence the variation in the bin-widths). Next, the mean of each such bin (μ_i^{target}) is found, and stored in an array. Note that it is assumed that either there is a known functional form for the target CDF or there is target data. For the case of a known target CDF, the calculations mentioned in this paragraph are straightforward in principle. For the case of target data, μ_i^{target} can be determined using the method described below.

- Next, sort the data to be transformed in descending order and define bins along the data axis such that each bin holds an equal number of data points

(note: this is again effectively breaking up the data sample-CDF into equi-probable bins). Then find the mean of each such bin (μ_i^{data}).

- The required transformation is simply the functional relationship between μ_i^{data} and μ_i^{target}, i = 1 to N.
- This non-linearity is approximated by a k'th order polynomial using least squares polynomial fitting, where the order 'k' is appropriately chosen – typically increasing with the increase in degradation of the speech signal.

For the sake of completeness, the problem of fitting a non-linearity onto a k^{th} order polynomial using least squares is formulated as follows:

Given 'n' data points (x,y), find the coefficients a_0, a_1, a_2,...a_k which satisfy $y = a_0 + a_1x + a_2x^2...+ a_kx^k$ in a least squares sense. The corresponding cost function is hence given by:

$$E^2 = \sum_{i=1}^{n}[y_i - (a_0 + a_1x_i + a_2x_i^2...+ a_kx_i^k)]^2 \tag{3.6}$$

and the coefficients a_0, a_1, a_2,...a_k are obtained by solving the equations

$$\frac{\partial E^2}{\partial a_j} = 0, j = 1:k \tag{3.7}$$

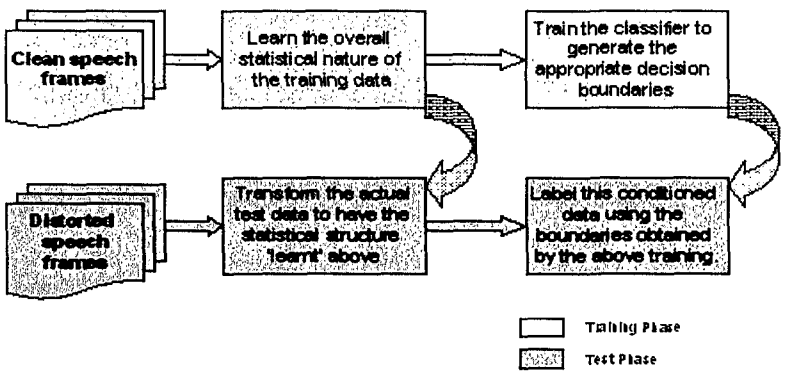


Fig. 3.3: Illustrating the proposed implementation of the QCM algorithm for automatic speech-recognition.

Fig. 3.3 illustrates the use of the QCM algorithm for typical speech recognition / speaker identification applications. In the training phase, the front-end learns the histogram of each speech feature representing the clean training speech and the back-end learns the appropriate decision boundaries for the classification task. During the testing phase, the front-end non-linearly scales the incoming speech features to fit the learned density function using QCM. The classifier makes a labeling decision on this conditioned un-labeled data. In a practical implementation, the model in Fig. 3.3 is used to learn the polynomial coefficients of the non-linearity transforming features from the data at the test SNR to have the same pdfs as those of clean training features. These polynomial transformations are then applied to features from the entire test data. An illustration of this algorithm is provided with real speech data in the next chapter.

3.3 Cross feature Linear Normalization (CLN)

Since in most speech/speaker recognition applications, class conditional density functions are Gaussians or their mixtures, covariance matrices have a direct influence on the shape of the equi-probable contours and hence the decision boundaries learned by the classifier. It can hence be inferred that a classifier that learns its boundaries from data having a covariance matrix $\mathbf{R}_i^{clean}$ for the i'th category will not perform well when the test data has a covariance matrix $\mathbf{R}_i^{noisy}$ (e.g. – see fig. 2.4 in chapter 2). It is desired to restore the shape of the equi-probable contours before making a labeling decision. In this section, one method is described for combining the CDF matching restoration with a linear transformation, which can be used to make the second order statistics (covariance matrix)

of test data match that of training data. In particular, consider the following proposed algorithm:

- Estimate the covariance matrix $\mathbf{R}_i^{clean}$ of data for each category from the clean speech frames.
- Collect a suitable number of test data points ($\mathbf{y}$), and compute the sample covariance matrix, $\mathbf{R}^{test}$.
- 'Diagonalize' and then 'whiten' the collected test data to have an identity covariance matrix:

$$\tilde{\boldsymbol{y}} = \Lambda_y^{-1/2} \mathbf{U}_y^T \boldsymbol{y} \tag{3.8}$$

Where $\mathbf{U}_y$ is a matrix whose columns contain the eigen-vectors of $\mathbf{R}^{test}$, and Λ_y contains the corresponding eigen-values on the diagonal.

- Perform another transformation as :

$$\hat{\boldsymbol{y}} = (\mathbf{R}_i^{clean})^{1/2} \tilde{\boldsymbol{y}} \tag{3.9}$$

It can be easily shown [A.1] that $E\{\hat{\boldsymbol{y}}\hat{\boldsymbol{y}}^T\} = \mathbf{R}_i^{clean}$ assuming that the original random variable $\mathbf{y}$ has zero mean. The mean vector is hence subtracted before this transformation and added back later. These steps will ensure that the collected test-data now has a covariance matrix almost equal to the clean data corresponding to that category. It is hoped that this together with marginal pdf normalization would lead to a better matching of the training and test feature spaces.

In an actual test-situation, the category label is not known and hence the target matrix $\mathbf{R}_i^{clean}$ is not known. This is the problem which needs to be addressed properly for

this approach to become really useful. However, to experimentally illustrate the theory, the following approach was used:

- Evaluate the sample covariance matrix $\mathbf{R}^{test}$ over a sufficient number of frames from the category being recognized.
- Among the possible target covariance matrices $\mathbf{R}_i^{clean}$ ($i \in [1:c]$, where 'c' is the number of categories), choose the one such that the matrix norm $\|\mathbf{R}^{test} - \mathbf{R}_i^{clean}\|$ is minimum.

The above proposition can be easily extended to the case of isolated word recognition or speaker ID problems, where each category is modeled by a mixture of Gaussians. In this case, a Gaussian mixture model of the test frames must be built, and the above technique should be repeated to each component in the mixture.

3.4 Linear Least Squares based compensation of training-test mismatch

It is proposed that for a given mismatch in the training and test conditions, the effects of noise and channel filtering can be compensated to a certain extent by a matrix transformation which can be learned from "stereo" training data using Least Squares estimation. If $(\boldsymbol{x}, \boldsymbol{y})$ represents paired feature vectors from the clean and noisy feature space respectively over all the categories obtained from stereo training data, then it is desired to find a transformation of the feature space, **T**, such that:

$$\boldsymbol{x} = \mathbf{T}\boldsymbol{y} \quad \forall(\boldsymbol{x}, \boldsymbol{y}) \tag{3.10}$$

Hence, **T** can be computed in a Least Square sense as follows:

$$\arg\min_{\mathbf{T}} \{\|\boldsymbol{X} - \mathbf{T}\boldsymbol{Y}\|^2\} \tag{3.11}$$

The least squares solution [A.2] is given by:

$$\mathbf{T} = \mathbf{X}.\mathbf{Y}^{T}.(\mathbf{Y}.\mathbf{Y}^{T})^{-1} \tag{3.12}$$

where $\mathbf{X}$ and $\mathbf{Y}$ represent matrices containing the vectors (x, y) over the entire training data (dimension of $\mathbf{X}$ and $\mathbf{Y}$ equals number of features by number of training data points).

Fig. 3.4 illustrates the extent of restoration accomplished using the proposed Linear Least Squares approach. Fig. 3.4(a) and Fig. 3.4(b) represent paired training data from clean and noisy feature space respectively for two vowels. It can be seen from Fig. 3.4(b) that noise and channel distortion shifts the mean vectors and changes the overall shape of each category (vowel). A matrix mapping data from Fig. 3.4(b) to Fig. 3.4(a) is computed using Least Squares estimation and is applied to other noisy test data. It can be seen from Fig. 3.4(c) that the matrix transformation of noisy data forces it to resemble the clean data in Fig. 3.4(a) closely - thereby compensating to some extent the effect of noise and distortion on clean speech. Experimental results in the next chapter illustrate the usefulness of this pre-processing in speech recognition, and also illustrate that a very small amount of training data is sufficient to compute a reasonably accurate transformation mapping the noisy space to the clean space.

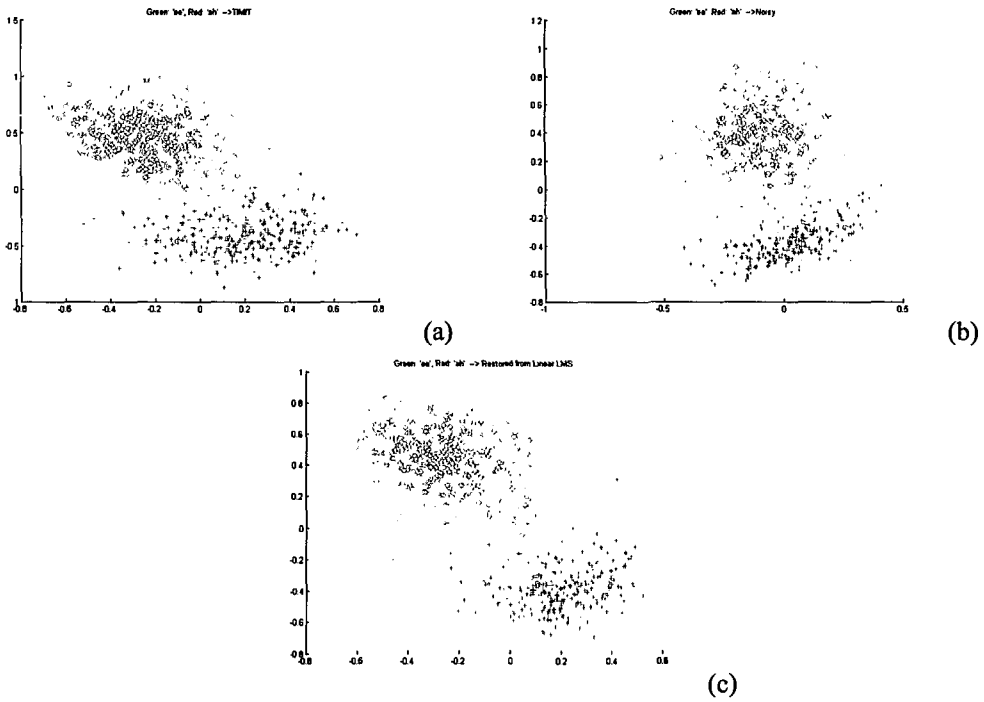


Fig. 3.4: DCT-1 vs. DCT-2 of vowels 'ah'(red) and 'ee'(green) for (a) Clean Speech (TIMIT), (b) Noisy Speech (NTIMIT), (c) Linear Least Squares estimate of clean speech from noisy speech.

The above proposed pre-processing is in-fact a simpler implementation of the "Vector Taylor Series (VTS)" approach proposed by [11] in which they approximate the environment degradation function by a 0'th and 1'st order Vector Taylor Series and compute the coefficients of the series empirically using the iterative Expectation Maximization (EM) technique. These are then used to compute an estimate of the clean speech features given the noisy ones in a minimum mean squared sense. For a mathematically tractable expansion of the Vector Taylor series, it was assumed that the speech features are uncorrelated and the covariance matrix is diagonal. The linear least squares method proposed in this section does not make any assumptions on the statistical structure of the data and only requires a small amount of stereo training data (i.e. same speech data recorded in clean and noisy conditions).

In the next chapter, the various algorithms proposed thus far separately and in combination are evaluated in the context of automatic speech recognition. Experimental results illustrate the extent to which these non-linear and linear transformations 'clean-up' the features.

CHAPTER IV

EXPERIMENTS

In the previous chapter various algorithms were discussed for non-linearly scaling each feature individually and linearly transforming the feature space for compensating for the effects of noise and channel distortion towards the goal of improving the robustness of speech recognition and/or speaker-identification systems. We now evaluate the improvement in performance of various recognition systems using these non-linear and linear transformation algorithms. Unless otherwise mentioned, the TIMIT [A.3] database (which contains speech recorded in a clean studio environment) was used for clean speech and NTIMIT [A.4] database (speech corrupted by telephone noise and channel distortion) was used for noisy speech. In all experiments described in this chapter, the sampling rate was 16 KHz and a 1024 point FFT was used for spectral processing (except for the SPIDRE database for which the sampling rate was 8 KHz). DCTCs, very similar to MFCCs but computed somewhat differently [12] were used as the features.

We first give four examples which illustrate that the implementations of the algorithms presented in Chapter 3 in fact do give the desired changes in probability density functions. These examples were done with both simulated and actual speech data. These examples are followed by an experimental evaluation of the effects of the algorithms on accuracy for speech recognition and speaker identification tasks.

4.1 Example 1 – Histogram matching of speech features to a Gaussian pdf using Polya's approximation

This example demonstrates the use of histogram matching for 'Gaussianizing' speech features marginally (i.e., transforming each feature individually to have a global Gaussian pdf with zero mean and 0.2 standard deviation). Fig. 4.1a shows the histogram of a feature extracted from NTIMIT (DCTC #2 based on all the vowel tokens). Fig. 4.1b illustrates the nonlinearity obtained for transforming this feature so that it would have a Gaussian pdf using the algorithm proposed in section 3.1 of chapter 3. Fig. 4.1c illustrates the histogram of the feature after applying the nonlinearity in Fig. 4.1b to the original feature. For this example, there were 3424 feature vectors and 100 bins were used to approximate Eq 3.3.

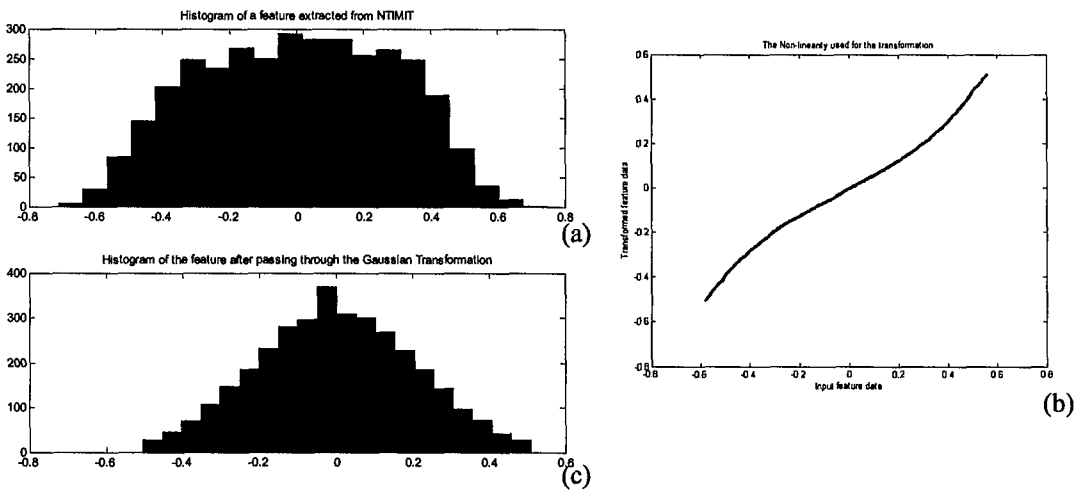


Fig. 4.1: (a) Histogram of a feature from speech utterances from NTIMIT, (b) the non-linearity obtained from Polya's approximation, (c) the histogram of the data after transformation.

4.2 Example 2 – Quantile based CDF matching (QCM)

In this example we use simulated data to illustrate the capability of the quantile-based Cumulative Density Function (CDF) matching technique for data drawn from different distributions, as presented in Chapter 3, Section 3.2. Fig. 4.2 illustrates the use of QCM

for restoration of uni-variate statistics of simulated data. Note that our implementation is no longer restricted to a single-mode Gaussian. In Fig. 4.2, the distorted data is simulated to have the 'black' pdf, the target data is defined to have the 'red' pdf, and the 'blue' pdf corresponds to the histogram of the data after the non-linear restoration on the distorted data using QCM. To obtain smooth histograms for illustration, we chose 10^5 data points drawn from different bi-modal Gaussian mixture distributions to derive the distorted and target pdfs. In this experiment, 200 quantiles were used for CDF matching and a 7^{th} order polynomial was used to approximate the non-linearity. Typically, the number of quantiles that can be used is directly proportional to the size of the available data-set. When a large training data-set is available for computing the transformation, a larger number of quantiles can be chosen for better resolution.

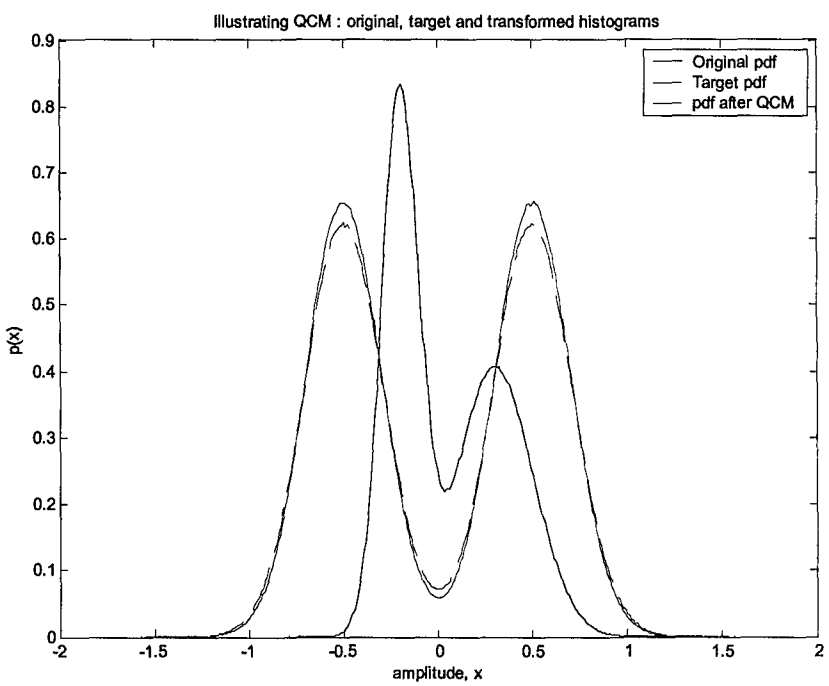


Fig. 4.2: Illustrating the use of QCM to transform simulated data from 'distorted' CDF to a target 'clean' CDF - the corresponding pdf-restoration attained.

Similarly in a speech recognition implementation, the order of the polynomial should be a function of the SNR of the test signal that is being transformed, e.g. at low SNRs there will be a significant deviation of the pdf obtained from noisy speech versus the pdf obtained from clean speech, and hence the transformation will be better approximated by a higher order polynomial.

4.3 Example 3 – QCM: Effect of polynomial order on the transformation quality

For the purpose of illustration, consider the task of transforming uniformly distributed data to a single-mode Gaussian distributed data. The following experiment illustrates the deviation of the pdf of the transformed data from a Gaussian pdf as a function of the polynomial order used for the transformation.

A zero mean and 0.2 standard deviation Gaussian pdf was used as the target pdf. 10^3 points were drawn from a uniform distribution for simulating the data to be transformed. 25 bins were used for the histograms used to approximate the data pdf. Least squares polynomial fitting was used to approximate the polynomial coefficients representing the non-linearity. The order of the polynomial was varied from 3 to 15 and with each unit increase in the order, the smoothness of the resulting Gaussian improved – up to 12 at which point the polynomial approximation produced best results. Fig. 4.3 illustrates visually the quality of transformation attained using polynomial orders 6 and 15.

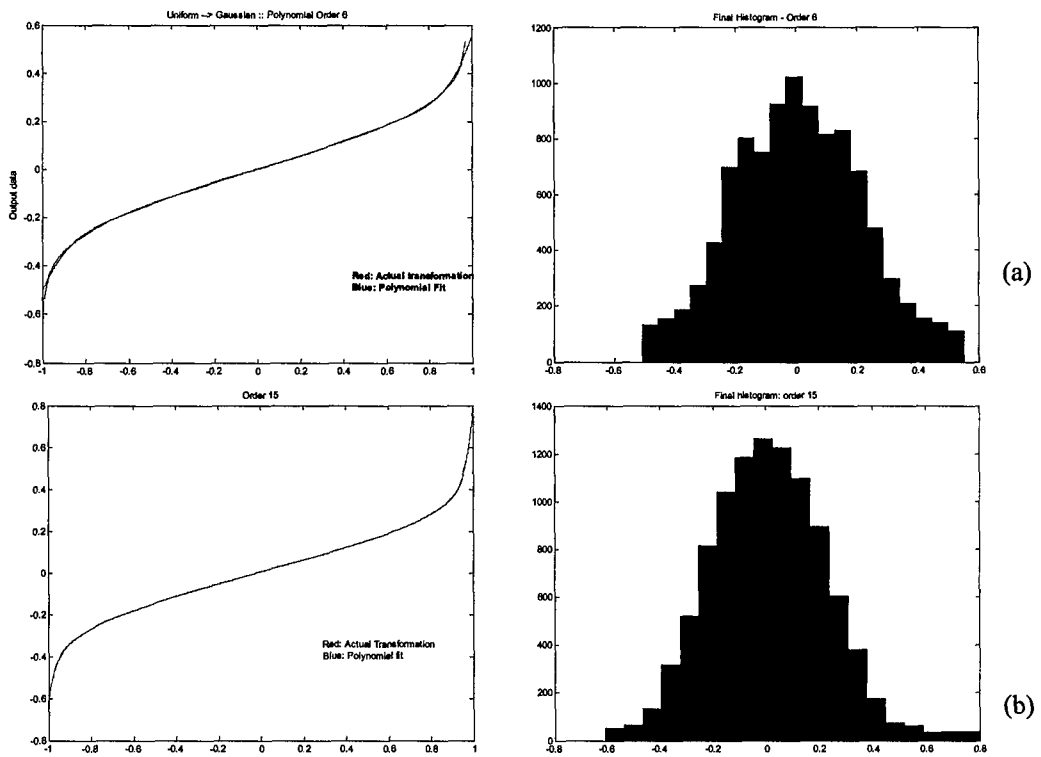


Fig. 4.3: (a) Using a 6'th order polynomial approximation, and the corresponding histogram of the transformed data, (b) Using a 15'th order polynomial and the corresponding histogram of the transformed data.

4.4 Example 4 – Using QCM to force noisy speech features to Gaussian pdfs

This example illustrates the use of QCM for 'Gaussianizing' speech features marginally to have a global Gaussian pdf with zero mean and 0.2 standard deviation. Fig. 4.4a shows the histogram of a feature (DCTC #2) extracted from vowel utterances in the NTIMIT database. 3424 feature vectors were used for computing the non-linearity and 100 quantiles were used for the CDF matching. 25 bins were used for the display-histograms used to approximate the data pdfs.

Fig. 4.4b illustrates the nonlinearity used for the transformation to a Gaussian pdf using the QCM algorithm proposed in Section 3.2 in Chapter 3. It also illustrates the

corresponding polynomial fit to this nonlinearity – computed using least squares fitting. This polynomial can be used for transforming future data points of the feature extracted in similar environmental conditions. Fig. 4.4c illustrates the histogram of the feature after applying the nonlinearity in Fig. 4.4b to the distorted feature.

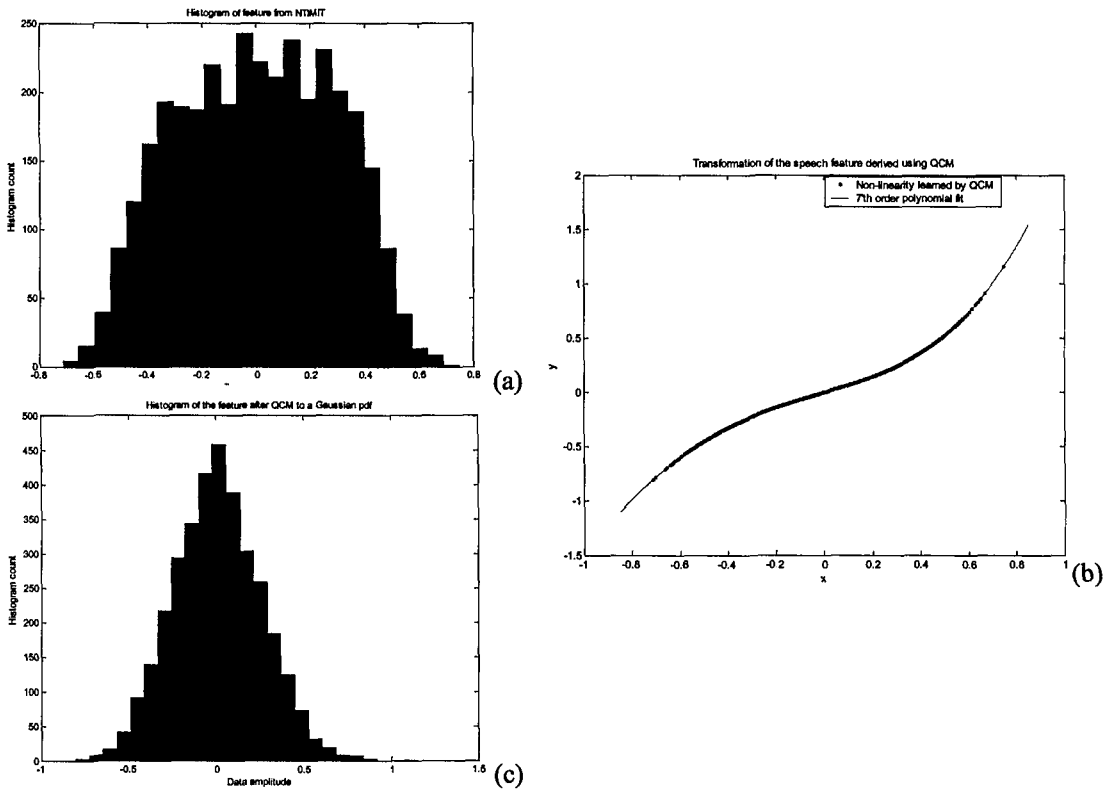


Fig. 4.4: (a) Histogram of a feature from speech utterances from NTIMIT, (b) the non-linearity used based on QCM and the corresponding 7'th order polynomial fit, (c) the histogram of the data after transformation.

Sections 4.5 through 4.9 illustrate the benefit of the algorithms proposed in chapter 3 by experimental evaluation on speech recognition and speaker identification tasks.

4.5 Experiment 1 – Phonetic classification incorporating histogram matching to a Gaussian pdf

This experiment illustrates the improvement in recognition performance using marginal histogram matching (using the algorithm described in section 3.1) of the feature space to Gaussian pdfs with a zero mean and standard deviation of 0.2. Note that this experiment is an example of normalization rather than restoration, since all features for both training and test are transformed to a fixed specified density (Gaussian). Thus the features are globally normalized to have the same statistical properties for both the training and test utterances. It is hoped that this normalization reduces the statistical mismatch that the classifier faces between the training and test conditions and thus will result in improved classifier accuracy. 3424 noisy training feature vectors and 100 bins were used to approximate Eq 3.3 for histogram matching. As the baseline experiment (illustrated in fig. 4.5) we simulate a training-test mismatch by training a classifier on speech from the TIMIT database. This classifier was tested with speech from the NTIMIT database.

The classifier used for the recognition task (as well as for most of the other experiments reported in this section) was a minimum error rate Bayesian classifier [16]. This classifier is minimum error rate only if the features are multivariate Gaussian. The computer program used to implement this classifier had three basic modes of operation, referred to as MXL-1, MXL-2, and MXL-3. For MXL-1 the features are assumed to be uncorrelated and to have equal variances. Thus the decision rule is based on the minimum Euclidean distance to class centroids, plus a term to incorporate the apriori probabilities of the classes. For the case of MXL-3, a common covariance matrix is assumed among all classes. Thus the decision is based on the minimum Mahalanobis distance, plus a term to incorporate apriori probabilities. For MXL-2, no assumptions are made other than

multivariate Gaussian, and thus covariance matrices are computed separately for each class and used in the decision process.

Figure 4.5 depicts classification results obtained for this experiment, as a function of number of features used, type of normalization, and classification mode. Results are only given for MXL-1 (simplest classifier, but with lowest overall accuracy), and MXL-3, for which the highest overall accuracy was obtained for this experiment. As expected, the baseline recognition results in Fig. 4.5 for this experiment are extremely low since the boundaries learned by the classifier are not representative of the noisy and distorted speech.

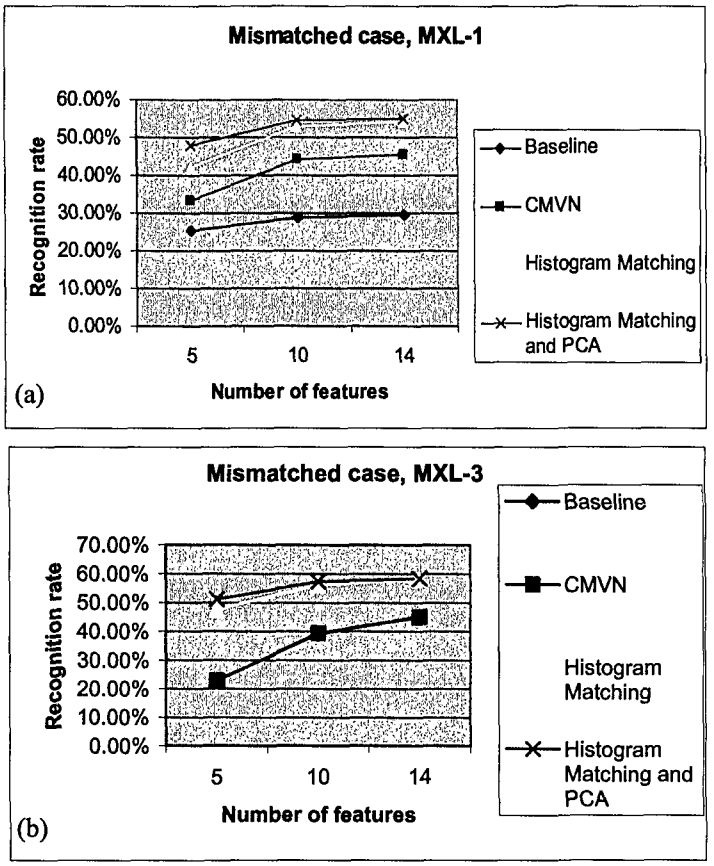


Fig. 4.5: Phonetic classification results using the (a) Euclidean distance measure, (b) Mahalanobis distance measure.

The Mahalanobis distance measure is a better metric for vowel classification tests and hence the general results with Mahalanobis distance measure (MXL-3) are better than those with the Euclidean distance measure (MXL-1). Restoring the first two moments (Cepstral Mean and Variance Normalization – CMVN) of the noisy test data improves the test-recognition performance with MXL-1. However, normalizing both the training and test data to a single-mode Gaussian (using histogram matching discussed in Section 4.1 and illustrated in Fig. 4.1) with a zero mean and 0.2 standard deviation improves the test recognition results with both distance measures significantly.

As a further refinement we propose that in addition to marginally 'Gaussianizing' the features, if we also de-correlate the feature space in both the training and test conditions, we will compensate for the mismatch in the global cross-feature information. It is hoped that by forcing both the training and test-features to be un-correlated we will compensate for changes in the cross-feature properties introduced by channel noise and distortion. To this effect, we performed Principal Component Analysis (PCA) in conjunction with marginal histogram matching to de-correlate the feature space because PCA diagonalizes the sample covariance matrix of the data it transforms. Experimental results in Fig. 4.5 depict the slight improvement in recognition performance by adding PCA to the histogram matching based normalization. The improvement is significant when a small number of features are used. In recognition using higher dimensional feature spaces, the improvement in performance is small. The basis vectors used for performing PCA on the data were learnt from a training set of data at the same SNR as the test data.

4.6 Experiment 2 – Generalized multi-modal marginal normalization vs. single-mode marginal Gaussianization

It has been shown [13] that some features do not necessarily exhibit a single mode Gaussian pdf even for speech utterances collected in clean conditions. It was hence hoped that instead of transforming both the training and test data to a Gaussian pdf, it would be beneficial to learn the 'target' pdf of each feature from clean training speech data and transform the noisy speech features to the corresponding 'target' pdf. To test this idea, the reference CDFs for QCM were learned from TIMIT (clean speech), and 7^{th} order polynomials were used to transform the incoming test features from NTIMIT to have the same pdfs as those of the features extracted from TIMIT. The baseline experiment is similar to the one discussed in Section 4.5 (training the classifier with clean data and testing it on noisy data). Results in Fig. 4.6 show that instead of forcing the target pdf of each feature to be a single mode Gaussian (QCM-1), if we force the test data to have the learned clean pdf, there is a further (small) improvement in recognition performance.

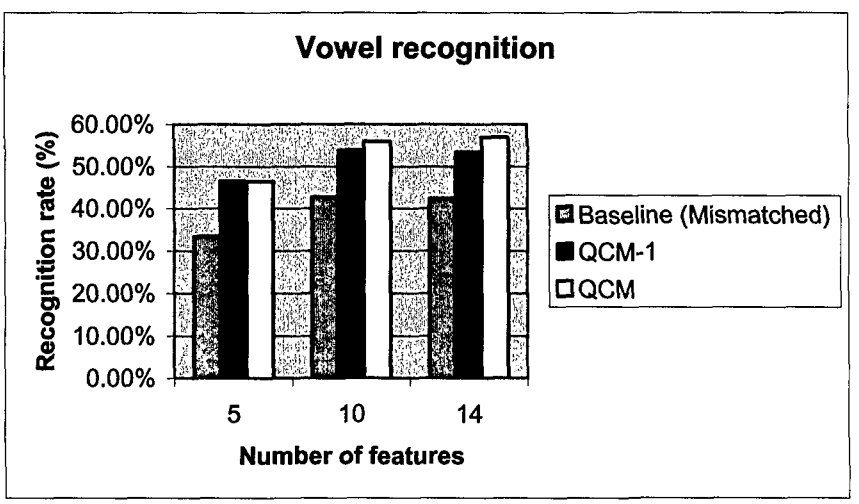


Fig. 4.6: Illustrating the benefit of using target pdfs learned from clean speech rather using single-mode Gaussian pdfs for both training and test data.

4.7 Experiment 3 – Using QCM for robust Speaker Identification

This section illustrates the use of QCM for robust speaker identification. The experimental setup consists of a binary-pair partitioned neural network [14] for recognizing 50 speakers. The NTIMIT database was used (for both training and testing), containing speech degraded by telephone distortion and noise. Seven sentences from each speaker were used for training the neural network and three were used for the identification task. Twenty five static features were extracted for recognition. Both the training and test utterances were forced to belong to a Gaussian density with a mean of zero and standard deviation of 0.2. The non-linearity required to transform the data to a Gaussian pdf was derived from the histogram of a single training sentence and was applied to all the training and test data. Approximately 1000 feature vectors were used for computing the non-linearity and 100 quantiles were used for the CDF matching. A 7^{th} order polynomial was used to approximate the non-linearity. In fact, the process can be viewed as adding an extra scaling layer to the neural network whose nodes have a non-linearity determined using QCM. This process can also be viewed more as normalization rather than restoration (Section 2.4). Once this non-linearity is determined for each node (feature), it is applied to all the data – during the training and testing phase. Table 4.1 illustrates the general improvement in speaker identification performance by incorporating QCM-based non-linear scaling of training and test data. The speaker ID recognition accuracy improves by approximately 4-6% with the NTIMIT database by incorporating the QCM based non-linear feature scaling described above. Additive white Gaussian noise was added to speech from NTIMIT to control the SNR. The improvement

in recognition performance is consistent down to a 0dB SNR, at which point, the improvement is only slight.

Table 4.1: Illustrating the improvement in speaker identification performance using QCM with NTIMIT and SPIDRE databases.

NTIMIT							SPIDRE		
Utterance (Sec.)	Baseline (0dB)	QCM (0dB)	Baseline (20dB)	QCM (20dB)	Baseline (50dB)	QCM (50dB)	Utterance (Sec.)	Baseline	QCM
0.375	21.3%	21.5%	25.1%	30.4%	24.0%	29.5%	3.4	57.5%	59.8%
0.75	24.3%	24.8%	34.0%	41.3%	33.3%	38.0%	6.7	58.8%	62.2%
1.5	28.5%	30.0%	45.5%	50.0%	47.5%	51.0%	13.5	60.0%	62.2%
3	38.0%	40.0%	57.0%	61.0%	60.0%	66.0%	26.9	64.4%	64.4%
6	41.0%	42.0%	70.0%	72.0%	74.0%	78.0%	-	-	-

A similar experiment was also performed on the SPIDRE database – which has forty five speakers; four conversations from each speaker are included, of which two are from the same handset [15]. The baseline experiment consists of a handset mismatch where the two sentences from one handset were used for training and 1 sentence from a different handset was used for testing (to simulate a harsh handset mismatch). The classifier was again a binary-pair partitioned neural network and twenty five features were used for recognition. Table 4.1 illustrates that even for these difficult conditions, incorporating QCM in the configuration discussed above is slightly beneficial, especially in recognition with shorter test utterance lengths.

The details of the binary-paired partitioned classifier are described in [14], and not really important for the sake of the work described in this thesis, except for one critical point. That is, the neural network makes no assumptions about the underlying density functions of the data and is thus considered a non-parametric classifier. Ideally, it simply determines optimum decision boundaries for any feature distributions. Thus it is quite remarkable that the classification accuracy improved for the experiment described in this section, since a fixed normalization was applied to both training and test data.

Although not experimentally verified, we speculate that the Gaussian features have better generalization from training to test than do the original features.

4.8 Experiment 4 – Using QCM in conjunction with CLN for phonetic classification

With the motivation of attaining a more general multi-dimensional statistical restoration, we study the effect of incorporating the linear matrix transformation (CLN) with the marginal non-linear transformation (QCM). This experimental evaluation was carried out for vowel classification experiments using a setup similar to the one described in Section 4.5 – the classifier was trained on clean speech and tested on noisy speech. Fig. 4.8 illustrates the front-end incorporating both the QCM and CLN algorithms for reducing the training-test mismatch in speech recognition / speaker-identification systems.

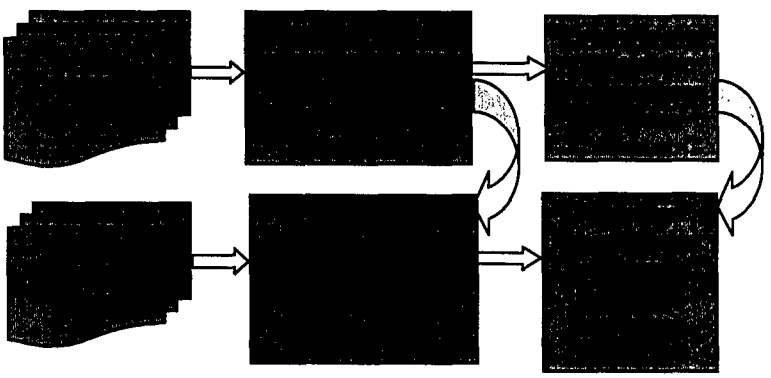


Fig. 4.7: Illustrating the proposed implementation of the QCM and CLN algorithms

During training at high SNR, the front-end learns the covariance matrices of the categories and the marginal CDFs of the features and the back-end learns the optimal decision boundaries for the labeled training vectors. During testing (at any SNR), the CLN unit reduces the distortion and rotation effects on the test data. The front-end then conditions the feature vectors to have marginal CDFs close to those learned from the

clean speech frames. After this conditioning, the back-end makes the classification decision.

Fig. 4.9 shows the significant improvement attained by using both CLN and QCM together. The baseline results in this case correspond to the recognition performance attained by performing Cepstral Mean and Variance Normalization on the raw feature space. A note of caution is required here in that the experimental implementation of CLN in this vowel classification setup is not a practical implementation for a real-time system because the CLN module linearly transforms a 'chunk' of data belonging to some category to the right shape / orientation.

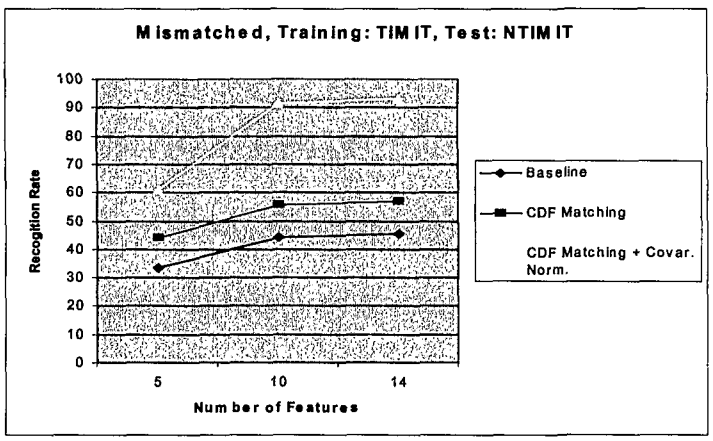


Fig. 4.8: Illustrating the improvement in the overall vowel recognition performance by using CLN in conjunction with QCM using the Euclidean distance.

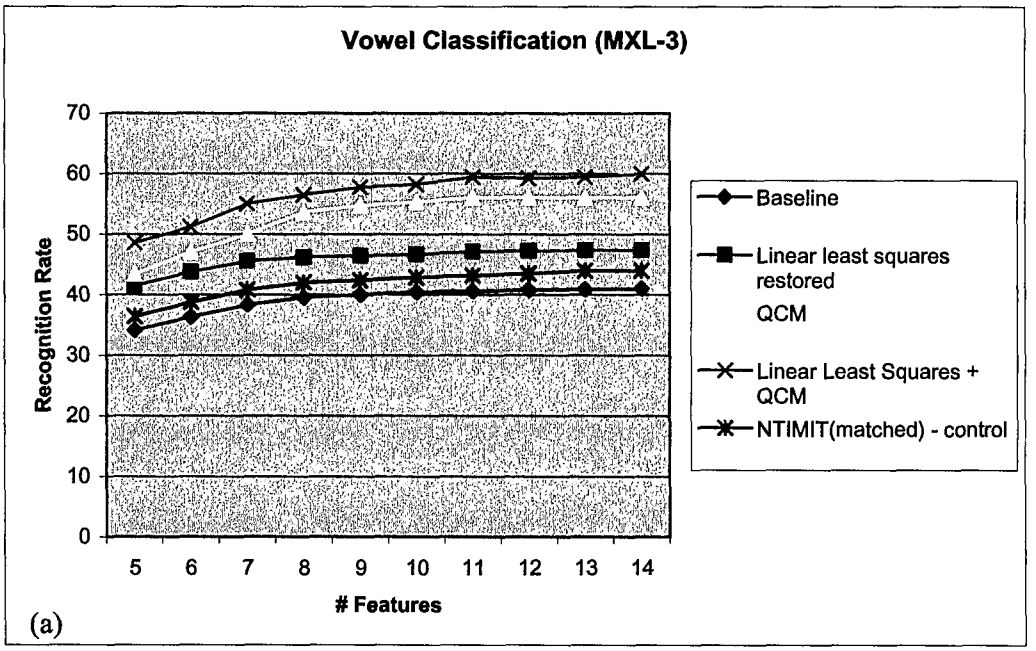
The experimental evaluation of CLN as illustrated here hence requires that we collect all the vectors corresponding to the noisy speech vectors for each vowel separately and transform them in 'chunks'. Although this approach is not possible for an actual speech recognition system (categories must be known in advance), it does demonstrate the extent of 'cleaning up' possible by such a linear transformation. It is further hoped that a variation on this method must be feasible for isolated-word recognition or speaker identification. The proposed CLN technique can be easily

extended for these recognition tasks - where each category is modeled by a mixture of Gaussians. In this case, we need to build the Gaussian mixture model of the test frames, and repeat the above technique to each component in the mixture.

4.9 Experiment 5 – Linear Least Squares based compensation of training-test mismatches

In this experiment, we explore the idea of incorporating a linear (matrix) 'least squares based' transformation as presented in Section 3.4 in Chapter 3. Fig. 4.10 illustrates the improvement in performance for vowel recognition (using two different classifiers – a minimum error rate based Bayesian classifier and a neural network based classifier) by using this linear transformation alone and in conjunction with QCM.

Once again, the baseline setup for this experiment is similar to the one described in section 4.5 – the classifier was trained on clean speech and tested on noisy speech. The matrix for the linear transformation was learned using equation 3.12 (chapter 3) from stereo training data consisting of speech data recorded simultaneously in clean and noisy conditions. The classifier in Fig. 4.10a uses a distance measure based on common covariance matrices for all categories [16] while Fig. 4.10b corresponds to a neural network based classifier (comprising of 1 hidden layer and 25 nodes and using the back-propagation algorithm for learning the network weights). We consider the NTIMIT (matched) classification to be the control since in this situation the training and test data come from similar environmental conditions.



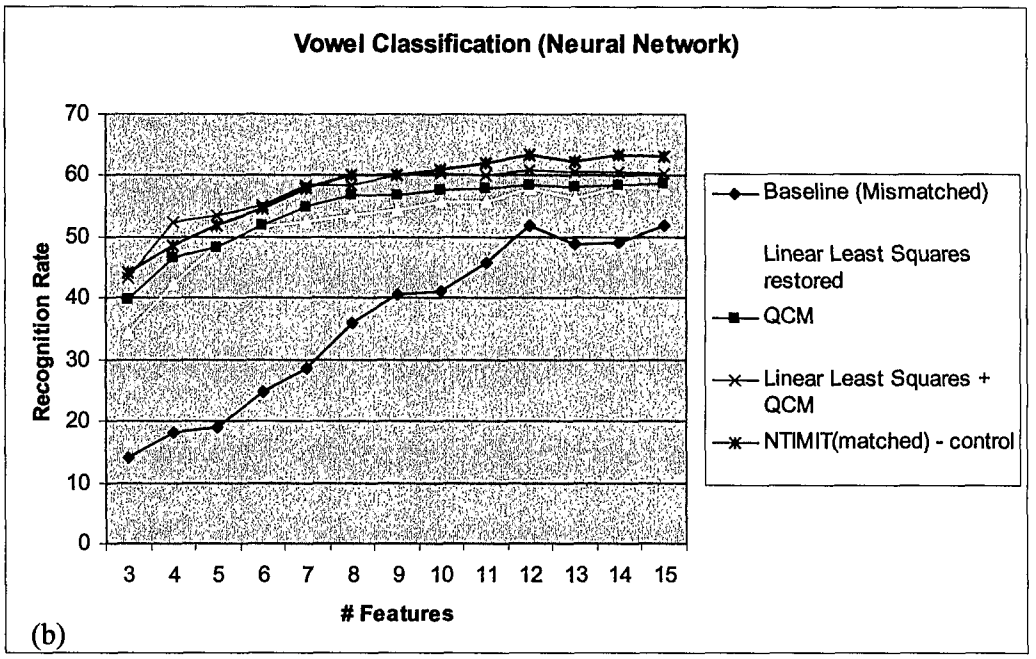


Fig. 4.9: Illustrating the improvement in overall vowel recognition performance by using Linear Least Squares based compensation in conjunction with QCM for a vowel classification experiment using (a) a minimum error rate Bayesian classifier based on a distance measure using common covariance matrix for each category (MXL-3), (b) a neural network based classifier.

The classification results shown in Fig. 4.10a (MXL-2) show that we attain a better performance than both the baseline mismatched results and the control matched

results by using QCM and the linear transformation individually and together. Fig. 4.10b gives recognition results of the same experiment with a different classifier (neural network). In this setup also, the proposed linear and nonlinear transformations improve the baseline mismatched recognition performance significantly. QCM in conjunction with the linear least square transformation performs slightly better than either of them individually. The combination of QCM and linear transformation performs better than the matched control recognition when a smaller number of features are used for recognition. When a larger number of features are used, the matched control recognition results are slightly better than the results corresponding to the combined QCM and linear transformations.

Fig. 4.11 illustrates the quality of the linear least squares based transformation (represented by the improvement in recognition performance of various classifiers) as a function of the number of stereo (i.e., having speech recorder simultaneously in clean and distorted conditions) training vector pairs used for learning the transformation matrix **T**. It can be seen from the figure that the transformation matrix **T** can be optimally learned from as little as 400 training vector pairs for various classifiers. We can hence conclude from this experiment that though for some classifiers, optimal decision boundaries are learned when the training and test data are from speech at the same SNR (matched case), in the situation where enough training data is not available for learning the optimal decision boundaries, it is beneficial to perform such a linear transformation (which can be learned using a small amount of training data). This coupled with marginal non-linear transformation further leads to a more general multi-dimensional statistical normalization – giving the best results in this mismatched scenario.

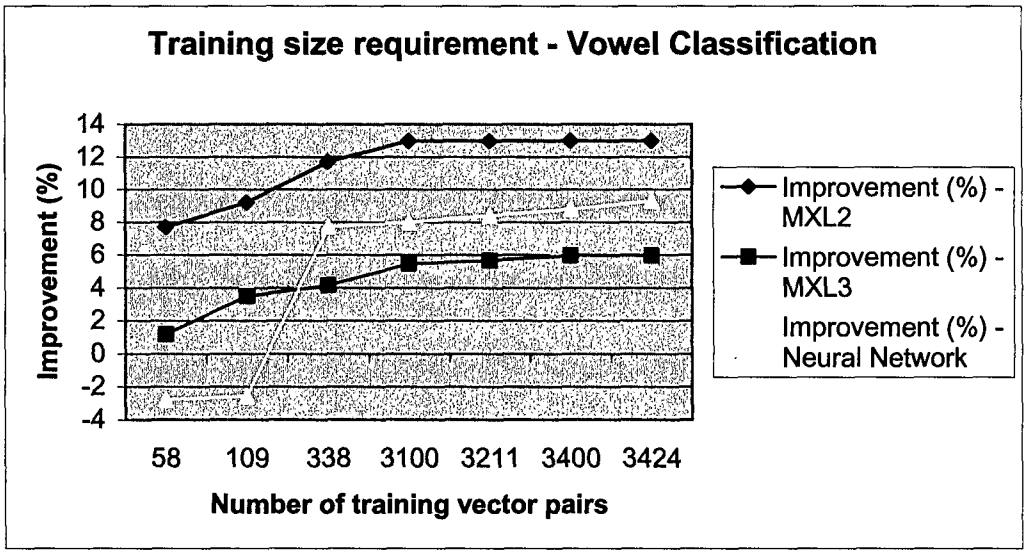


Fig. 4.10: Illustrating the improvement in vowel recognition results using the proposed linear transformation as a function of the number of vector pairs used for learning the transformation matrix **T**.

CHAPTER V

CONCLUSIONS AND FUTURE WORK

In this thesis, the potential of "cleaning up" statistics of speech features distorted by noise and channel effects is demonstrated, using linear and non-linear transformations. The non-linear transformation was derived using two techniques – Histogram matching and Quantile based CDF matching followed by least-squares polynomial fitting. The transformations can be determined from a small number of speech frames at the same SNR as the test speech and can then be applied to all test frames. The advantage of the non-linear transformation was illustrated both for speaker identification (using a binary-pair partitioned neural network classifier) and vowel classification (using separately a neural network classifier and a minimum error rate Bayesian classifier). The histogram of the features after this transformation will be closer to the target pdf if a suitably high number of bins are chosen for computing the transformation and a high enough polynomial order is chosen for polynomial fitting. However, as a general rule of thumb – a higher number of bins should be chosen only if there are sufficient training data points from which the non-linearity is computed (and vice-versa, when there are fewer data points available for training, fewer bins should be used). Similarly, the order used for polynomial fitting should be proportional to the extent of mismatch (since the greater the extent of mismatch, the more non-linear the required transformation generally is).

A summary of the experimental results obtained in this work is as follows. For the case of vowel classification with NTIMIT data (10 vowels), the best classification result obtained with no normalizing transformations, using a classifier trained with TIMIT data, is 52%. The highest classification accuracy using the normalizing transformations (QCM,

followed by the linear least squares transformation using a neural network classifier is 58%. However, the highest accuracy obtained using NTIMIT data for training was 62%. For the case of speaker identification with 50 speakers from NTIMIT, using QCM to globally Gaussianize all features, the best accuracy obtained was 78%, as opposed to 74% without any normalization.

Although difficult (and beyond the scope of this thesis) to quantify the above mentioned relationships among number of feature vectors, number of histogram bins, and polynomial order, the values used for the experiments reported in this thesis can be summarized. For example, for experiments involving QCM based transformation on a vowel classifier with the NTIMIT database (3478 feature vectors), 100 CDF quantiles were used for matching and a 7^{th} order polynomial approximation was used. Thus, on the average, there would have been approximately 35 data samples per histogram bin. Similarly, for speaker-identification experiments with SPIDRE, 200 CDF quantiles and a 7^{th} order polynomial were used. For this case, there were typically 5000 data points available per sentence, or approximately 25 data points per histogram bin.

Further proposed was the idea of deriving a linear (matrix) transformation mapping the noisy feature space to the corresponding clean feature space which is computed from stereo training data in a least squares sense. Experimental evaluation with vowel classification experiments reveals that this transformation significantly improves recognition accuracy in mismatched situations. It was also found that combining this linear transformation of speech features with a marginal non-linear transformation derived using the QCM algorithm produced further improvement in the recognition accuracy.

The concept of compensating for cross-feature distortion and rotation by restoring the covariance matrix of the test data using a linear transformation (Cross feature Linear Normalization) was introduced. This technique restores the covariance matrix of a chunk of distorted test data. Experimental evaluation results for a vowel classification paradigm were provided to demonstrate the potential of this technique to reduce mismatch between training and test data.

Combining linear (matrix) transformations with marginal non-linear transformations can be viewed as a more general multi-dimensional statistical restoration – as opposed to only marginal non-linear transformations. This forces the test data to have similar global statistics as the training data – from which the classifier learned its boundaries. This is demonstrated by the fact that in all these techniques, best recognition performance was obtained when both linear (matrix) and marginal non-linear techniques were used in conjunction. The two operations are not commutative, and the order of these two transformations was found to have a small effect on recognition accuracy. It was found that it was beneficial to first perform the marginal non-linear transformations and then the linear (matrix) transformation rather than the other way. However, transformations in the other order (linear first and then nonlinear) were still better than either transformation alone. Note that for either order of these transformations, the second transformation has the potential to interfere with the intended goal of the first transformation. A more complete analysis of these issues is left for future work.

It is worth mentioning here that ideally a single high order multi-dimensional nonlinear transformation capable of restoring multi-variate statistical information of distorted features should be used. However it may be very difficult to derive a

mathematically tractable solution. Further, for a multi-variate transformation, the training requirement will grow exponentially with the dimensionality of the feature space because of "the curse of dimensionality." The techniques proposed in this thesis thus are only sub-optimal in this sense. One possible approach to obtaining a single nonlinear transformation would be to train a multilayer neural network using stereo training data. That is, the network could be trained with noisy data features as input and clean data features as targets. With sufficient such data pairs, the network could learn the multi-dimensional nonlinearity that best maps noisy data to clean data.

It is also important to point out that for most classifiers, if there are enough training data available having the same statistical properties as the test data, the decision boundaries learned by the classifiers will be optimal. However, there are many situations where this may not be the case – e.g. when recognizing speech over a telephone network, the user may use a different hand-set than the one on which the system was trained on or the channel distortion affecting the speech may be different due to different network conditions etc. In these situations, a statistical normalization is beneficial for speech recognition and speaker identification.

As a natural extension to this work, it would be worthwhile to investigate the improvement in recognition performance of continuous speech recognition systems by similar pre-processing. Further, the CLN algorithm proposed in this thesis can be extended to a system for automatic speaker identification. One idea in that direction is to use a classifier that not only returns a category label, but also a confidence measure in the decision it makes. In this case, the CLN algorithm can be used to convert a single classification decision of a speaker's identity into 'c' classification decisions by forcing

the data from the speaker's distorted test sentence to have the covariance matrix of all the 'c' speakers learned from clean training sentences. A classification would then be performed on each of these transformed sets of features and the label for the one for which the classifier gave the highest confidence measure would be used.

REFERENCES

[1] Richard Stern. *Robust Signal Representations for Automatic Speech Recognition*, Presentation to 'Institute for Mathematics and its Applications, University of Minnesota' – September 19, 2000, ECE Dept., Carnegie Melon University.

[2] Alejandro Acero. *Acoustical and Environmental Robustness in Automatic Speech Recognition*, PhD Thesis, ECE Dept., Carnegie Mellon University, 1990.

[3] Pedro J Moreno. *Speech Recognition in Noisy Environments*, PhD Thesis, ECE Dept., Carnegie Mellon University, 1996.

[4] Xuedong Huang, Alex Acero, Hsiao-Wuen Hon, *Spoken Language Processing*, Prentice Hall, 2001.

[5] Angel de la Torre, Jose C Segura, Carmen Benitez, Antonio M Peinado, Antonio J Rubio. "Non-Linear Transformations of the feature space for robust speech recognition", *Proceedings of ICASSP-2002*, pp. I-401 – I-404.

[6] B.S. Atal, "Effectiveness of Linear Prediction Characteristics of the Speech Wave for Automatic Speaker Identification and Verification", *Journal of the Acoustical Society of America*, 1974, 55(6) pp. 1304-1312.

[7] Angel de la Torre, Antonio M Peinado, Jose C Segura, Jose L Perez Corodoba, Ma Carmen Benitez, Antonio J Rubio. "Histogram Equalization of Speech Representation for Robust Speech Recognition", *IEEE Transactions on Speech and Audio Processing*, Vol. 13, No. 3, May 2005, pp. 355-366.

[8] Bing Xiang, Upendra V. Chaudhari, Jirı Navratil, Ganesh N. Ramaswamy, Ramesh A. Gopinath, "Short-Time Gaussianization For Robust Speaker Verification", *Proceedings of ICASSP-2002,* Vol. 1, pp. 681–684.

[9] Athanasios Papoulis, S. Unnikrishna Pillai. *Probability, Random Variables and Stochastic Processes*, McGraw Hill.

[10] George Marsaglia, "The Exact Approximation method for generating Random Variables in a Computer", *Journal of the American Statistical Association*, Vol. 79, No. 385 (Mar. 1984), pp. 218-221.

[11] Pedro J. Moreno, Bhiksha Raj and Richard M. Stern, "A Vector Taylor Series Approach For Environment-Independent Speech Recognition", *IEEE International Conference on Acoustic, Speech and Signal Processing*, Vol. 2 (May 1996), pp. 733-736.

[12] S. A. Zahorian and Z. B. Nossair, "A Partitioned Neural Network Approach for Vowel Classification Using Smoothed Time/Frequency Features," *IEEE Transactions on Speech and Audio Processing*, Vol. 7, No. 4, pp. 414-425, July 1999.

[13] H. Yang; S. Van Vuuren; H. Hermansky, "Relevancy of time-frequency features for phonetic classification measured by mutual information", *IEEE International Conference on Acoustics, Speech, and Signal Processing, 1999 - ICASSP '99*, Vol. 1, pp. 225-228.

[14] L. Rudasi and S. A. Zahorian (1991), "Text-independent Talker Identification using Binary-pair Partitioned Neural Networks", *Proc. IJCNN-92*, pp. IV: 679-684.

[15] Catalogue of various databases maintained by the linguistic data consortium at http://www.LDC.upenn.edu/Catalog/, 15'th July 1992.

[16] Richard O. Duda; Peter E. Hart and David G. Stork. *Pattern Classification,* Second Edition. Wiley Interscience, 2000.

Appendix

A.1 Proof of the algorithm for covariance normalization proposed in section 3.3, chapter 3 (CLN)

Problem statement: To linearly transform a collection of data-vectors (denoted by $\boldsymbol{y}$) having covariance matrix $\mathbf{R}^{\text{test}}$ such that the sample covariance matrix of the data vectors after transformation (denoted by $\tilde{\boldsymbol{y}}$) is $\mathbf{R}^{\text{target}}$.

Proposition: Assuming that vectors being transformed are zero mean, the transformation attaining the above mentioned affect is given by:

$$\tilde{\boldsymbol{y}} = \boldsymbol{R}^{target} \Lambda_y^{-1/2} \boldsymbol{U}_y^T \boldsymbol{y} \tag{A.1}$$

The assumption of dealing with zero mean vectors is not a strong assumption because the mean vector of the chunk of data vectors can be easily subtracted before this transformation and added back to the transformed vectors.

Proof:

Consider first the whitening transformation

$$\boldsymbol{y}^* = \Lambda_y^{-1/2} \boldsymbol{U}_y^T \boldsymbol{y} \tag{A.2}$$

giving

$$\begin{aligned} \text{cov}[\boldsymbol{y}^*] &= E[\boldsymbol{y}^* . \boldsymbol{y}^{*T}] \\ &= \boldsymbol{I} \end{aligned} \tag{A.3}$$

The relation between $\tilde{\boldsymbol{y}}$ and $\boldsymbol{y}^*$ is given by

$$\tilde{\boldsymbol{y}} = (\boldsymbol{R}^{target})^{1/2} \boldsymbol{y}^* \tag{A.4}$$

giving

$$
\begin{aligned}
&\mathrm{cov}[\bar{\boldsymbol{y}}] = E[\bar{\boldsymbol{y}}.\bar{\boldsymbol{y}}^{T}] \\
&= E[(\boldsymbol{R}^{target})^{1/2}\boldsymbol{y}^{*}\boldsymbol{y}^{*T}(\boldsymbol{R}^{target})^{1/2}] \\
&= (\boldsymbol{R}^{target})^{1/2}.E[\boldsymbol{y}^{*}\boldsymbol{y}^{*T}].(\boldsymbol{R}^{target})^{1/2} \\
&= (\boldsymbol{R}^{target})^{1/2}.\boldsymbol{I}.(\boldsymbol{R}^{target})^{1/2} \\
&= \boldsymbol{R}^{target}
\end{aligned}
\tag{A.5}
$$

Hence by the proposed transformation, the collected data vectors can be forced to have the covariance matrix $\mathbf{R}^{target}$.

A.2 Proof of the linear least squares algorithm proposed in section 3.4

Problem statement: If $(\boldsymbol{x}, \boldsymbol{y})$ represents paired feature vectors from the clean and noisy feature space respectively over all the categories obtained from stereo training data, then it is desired to find a transformation of the feature space, $\boldsymbol{x} = \mathbf{T}\boldsymbol{y}\ \forall(\boldsymbol{x}, \boldsymbol{y})$.

Proposition: The required transformation is given by

$$\mathbf{T} = \mathbf{X}.\mathbf{Y}^{T}.(\mathbf{Y}.\mathbf{Y}^{T})^{-1} \tag{A.6}$$

where $\mathbf{X}$ and $\mathbf{Y}$ represent matrices containing the vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ respectively over the entire training data (dimension of $\mathbf{X}$ and $\mathbf{Y}$ equals number of features by number of training data points).

Proof:

The required matrix $\mathbf{T}$ mapping $\boldsymbol{y}$ to $\boldsymbol{x}$, $\forall(\boldsymbol{x}, \boldsymbol{y})$ can be found in a least squares sense such that $\sum_{i=1}^{n} \|\boldsymbol{x}_i - \mathbf{T}\boldsymbol{y}_i\|^2$ is minimized. Here 'n' is the number of vector pairs $(\boldsymbol{x}, \boldsymbol{y})$ available for learning the transformation matrix. To find such a $\mathbf{T}$, the orthogonality principle of least squares estimation is used, i.e., choose $\mathbf{T}$ so that $\boldsymbol{x}_i - \mathbf{T}\boldsymbol{y}_i$ to be orthogonal to $\boldsymbol{y}_j\ \forall \mathrm{i}, \mathrm{j}$. Thus,

$$\begin{array}{llll} (x_1 - \mathbf{T}y_1).y_1 = 0 & (x_1 - \mathbf{T}y_1).y_2 = 0 & & (x_1 - \mathbf{T}y_1).y_n = 0 \\ (x_2 - \mathbf{T}y_2).y_1 = 0 & (x_2 - \mathbf{T}y_2).y_2 = 0 & \cdots & (x_2 - \mathbf{T}y_2).y_n = 0 \\ \vdots & \vdots & & \vdots \\ (x_n - \mathbf{T}y_n).y_1 = 0 & (x_n - \mathbf{T}y_n).y_2 = 0 & & (x_n - \mathbf{T}y_n).y_n = 0 \end{array} \quad \text{(A.7)}$$

The set of equations above can be collapsed into a single matrix equation given by

$$(X - \mathbf{T}Y)Y^T = \mathbf{0} \quad \text{(A.8)}$$

where $\mathbf{0}$ is a number of features x number of features matrix of zeros. Hence

$$XY^T - \mathbf{T}YY^T = \mathbf{0} \quad \text{(A.9)}$$

giving the required solution

$$\mathbf{T} = \mathbf{X}.\mathbf{Y}^{\mathrm{T}}.(\mathbf{Y}.\mathbf{Y}^{\mathrm{T}})^{-1} \quad \text{(A.10)}$$

A.3 TIMIT specifications

The TIMIT corpus of read speech has been designed to provide speech data for the acquisition of acoustic-phonetic knowledge and for the development and evaluation of automatic speech recognition systems. Text corpus design was a joint effort among the Massachusetts Institute of Technology (MIT), Stanford Research Institute (SRI), and Texas Instruments (TI). The speech was recorded at TI, transcribed at MIT, and has been maintained, verified, and prepared for CD-ROM production by the National Institute of Standards and Technology (NIST). TIMIT contains a total of 6300 sentences, 10 sentences spoken by each of 630 speakers from 8 major dialect regions of the United States. Table 1 shows the number of speakers for the 8 dialect regions, broken down by sex. The percentages are given in parentheses.

Dialect Region(dr)	#Male	#Female	Total
1	31 (63%)	18 (27%)	49 (8%)
2	71 (70%)	31 (30%)	102 (16%)
3	79 (67%)	23 (23%)	102 (16%)
4	69 (69%)	31 (31%)	100 (16%)
5	62 (63%)	36 (37%)	98 (16%)
6	30 (65%)	16 (35%)	46 (7%)
7	74 (74%)	26 (26%)	100 (16%)
8	22 (67%)	11 (33%)	33 (5%)
8	438 (70%)	192 (30%)	630 (100%)

The dialect regions are:
dr1: New England
dr2: Northern
dr3: North Midland
dr4: South Midland
dr5: Southern
dr6: New York City
dr7: Western
dr8: Army Brat (moved around)

A.4 NTIMIT specifications

The NTIMIT, or Network speech database is a telephone bandwidth version of the widely used TIMIT database. Some useful features of NTIMIT: (excluding the characteristics that make TIMIT useful)

- Actual telephone channels were used, not simulations or models.
- Telephone channels used for transmission were varied in a "controlled" manner, in order to sample various line conditions.
- NTIMIT utterances are time-aligned with TIMIT utterances, allowing the use of existing phonetic, orthographic, and other transcriptions

A.5 Algorithm development for QCM

MATLAB was used to implement all the algorithms proposed in this thesis. The various MATLAB routines created for the algorithm implementation of QCM are as follows:

- **QCM_gauss (target pdf = gaussian)** – uses the QCM algorithm to find out the non-linearity required to transform the data specified in a vector to a Gaussian pdf.

  Usage:

  [x, y] = qcm_gauss(a,N_bins)

  Inputs:

  a - Data to be transformed (vector)

  N_bins - Number of bins to be used for quantizing the CDF

  Outputs:

  [x, y] - The non-linearity whose samples are given by 'mu_x' and 'mu_y'

- **QCM_general (target pdf = arbitrary)** – uses the QCM algorithm to find out the non-linearity required to transform the data specified in a vector to have a pdf specified by data in another vector.

  Usage:

  [x, y] = qcm_general(a, b, N_bins)

  Inputs:

  a - Data to be transformed (vector)

  b - Data to learn the target CDF from (vector)

  N_bins - Number of bins used for quantizing the CDF

  Outputs:

  [x, y] - The non-linearity whose samples are given by 'mu_x' and 'mu_y'

- **QCM_transformation** – Uses qcm_gauss or qcm_general to find out the transformation of the multi-dimensional feature data to have target pdfs, and stores the corresponding polynomial coefficients in the variable 'coeff'.

  Usage:

  [coeff] = qcm_transformation(*x*,order, N_bins)

  Inputs:

  x - Data to be transformed (matrix: feature_index x data_index)

  order - Order of the polynomial fit to the non-linearity

  N_bins - Number of bins to be used for quantizing the CDF

  Outputs:

  Coeff - vector representing the coefficients of the polynomial

  Dependent files:

  - QCM_gauss or QCM_general

- **QCM_scale** – reads in features of speech files and generates the appropriate polynomial coefficients and stores them in the 'poly_coeff.dat' text file

  Usage:

  qcm_scale

  Dependent files:

  - QCM_gauss or QCM_general

  - QCM_transformation

  - QCM_in.dat: specifies list of files to collect speech features from and contains the various transformation parameters (e.g. number of bins, polynomial order etc.)

Output files:

- 'poly_coeff.dat': stores the coefficients of polynomial transformation for each feature.

- **QCM_transfor** - uses the polynomial coefficients generated by qcm_scale and transforms the speech features, writing back the transformed files.

Usage

qcm_transfor

Dependent files:

- QCM_gauss or QCM_general

- QCM_transformation

- QCM_in.dat: specifies list of files to collect speech features from and contains the various transformation parameters (e.g. number of bins, polynomial order etc.)

- QCM_out.dat: specifies list of files to write the transformed features to.

Curriculum vitae

Saurabh Prasad

c/o: Professor Joseph Picone,
Room 2130
200 Research Blvd.
Starkville
Mississippi
39759
USA
39759-5405

Phone:
Mobile: 757-553-6914
Work: 662-325-5444
E-mail:
prasad @ cavs.msstate.edu
saurabh.prasad @ ieee.org

Objective

Active research in the area of digital and statistical signal processing.

Education

[Fall-2005 – Onwards] Mississippi State University MS
Ph.D. (Electrical and Computer Engineering).

[Fall-2003 – Summer 2005] Old Dominion University VA
M.S. (Electrical and Computer Engineering), GPA – 3.78, Graduation date – August 2005.

[1999-2003] J.M.I. New Delhi, INDIA.
B.Tech (Electrical) : 75.04%, Class rank: 4'th in a class of 50

Papers

- Saurabh Prasad, Stephen A Zahorian – *'Nonlinear and Linear Transformations of speech features to compensate for Channel and Noise effects'*, Accepted – Eurospeech-2005.
- 3'd Prize for presenting a paper in the academic year 2001-2002 at an all-India technical paper presentation contest, organized by IEEE - *'Improved Lead Compensation Technique for Digital Thermometry using 3 Wire RTD's'*
- 3'd Prize for presenting another paper at the above technical festival in the academic year 2002-2003 – *'Digital hardware realization of a linearized capacitive bridge for automatic balancing'*

Work experience

- Teaching Assistant and grader for Digital Logic Design course (ECE-241) at Old Dominion University [Fall-2003 – Spring-2005]
- Research Assistant at Old Dominion University's Speech Communications Lab since Fall-2003 [Fall-2003 – Summer-2005]
- Summer Training with the WLL group in Madras, India, established by Midas communications and Analog Devices [June-2001 – July-2001]