IMPLEMENTATION OF STATISTICAL MODELING TECHNIQUES AND CHANNEL ADAPTATION TECHNIQUES

Raja Shekhar R. Seelam

Institute for Signal and Information Processing
Department of Electrical and Computer Engineering
Mississippi State University
Mississippi State, Mississippi 39762
seelam@isip.msstate.edu

ABSTRACT

Implementation of various Statistical Modeling Techniques is necessary for the building of a Speech Recognizer. Statistical Modeling is done to learn the nature of the multi-variate random process generating the signal parameters. In this direction, pre-whitening transformations were performed on the parameters to eliminate redundancy and to make the analysis easier.

The transformations were performed on the input feature vector to produce an uncorrelated Gaussian random vector, containing only “information-bearing” parameters. For some algorithmically complex computations such as the computation of the eigen values and eigen vectors, existing software was used.

Channel adaptation techniques were implemented so as to make the parameters robust to changes in the acoustical environment. For this purpose, two particularly simple, but effective algorithms, Cepstral Mean Normalization/Subtraction and RASTA were chosen.

17. INTRODUCTION

Statistical modeling:

The primary aim of statistical modeling is to learn the nature of the multi-variate random process, assumed to be generating the signal parameters. A further insight into statistical modeling can be gained by looking at some issues like variance-weighting.

Very frequently, we will be interested in knowing the distance between two feature vectors. If we use a simple distance measure, such as the Euclidean metric, to make this comparison, the result will be most likely erroneous. This is due to the fact that if use of such a simple distance measure is made, the lower amplitude terms contribute much less, if not negligible, to the outcome compared to the larger-amplitude terms, even though the true information may lie in the smaller amplitude terms. For example, feature vectors normally include such measurements as the cepstral coefficients, derivatives of the cepstral coefficients and energy measures. Since the variances of the time-derivatives of the cepstral coefficients are larger than the cepstral coefficients, a direct distance measure between two feature vectors will be dominated by the derivatives of the cepstral coefficients even though the true information may lie in the cepstral coefficients [1]. So, we need to normalize the features so that all of them contribute equally in any comparison.

Another issue to be considered, is the presence of correlation between the features which makes the analysis complex. If two features are correlated in such a way that if one increases, the other decreases, then the result of the feature vector comparison could turn out to be erroneous as the two frames may not be as different as the outcome may indicate [2]. So, correlation has to be eliminated from our features to make the analysis easier. Also, correlation implies redundancy and we might be able to achieve some level of reduction by choosing a subset of the features, thus reducing the complexity of the problem to some extent. As we will see, this can be achieved by performing prewhitening transformations on the feature vectors.

First, the various steps involved in the prewhitening transformation will be discussed and then feature selection will be introduced.

Channel Adaptation:

The need for speech recognition systems to be more robust with respect to their acoustic environment has become more widely appreciated in recent years [3]. Performance of many automatic speech recognition machines, designed to be speaker-independent, has been found to be deteriorating if a different acoustical environment is used to test them [4,5]. Channel adaptation is an algorithm designed to make the system more robust to the changes in the acoustical environment. Use of channel adaptation techniques has proven to be very effective in the upgradation of the performances of speech recognizers. For example, Acero demonstrated that a large vocabulary speech recognition system with a base-line performance of 85% word accuracy in a matched transducer condition could only achieve less than 19% word accuracy when a different microphone was used during testing [4].

Almost all the techniques that are currently being used, make use of the fact that the variations of noise are relatively slower than the variations in speech. There are many techniques currently in use to achieve channel adaptation or robust recognition. We will discuss some of them briefly and then discuss the two techniques used by us, cepstral mean normalization (CMN) and RelAtive SpecTrAl processing (RASTA) in more detail.

2. A SIMPLE OVERVIEW OF THE FRONT END

This section aims at giving a simple overview of the front-end which is essentially the signal processing section of the speech recognition system. The primary objective of the front-end is to produce a feature vector that will be utilized to develop the acoustic models.

The pre-processing of the 16Khz sampled speech data is done by:

- taking data on a frame-basis (typically of length 20 ms),
- debiasing the data,
- preemphasizing the data (with a filter $1 - 0.95z^{-1}$),
- and windowing the data (a typical window used is a hamming window of 30 ms duration).

The speech signal is preemphasized to compensate for the attenuation caused by the radiation from the lips [25]. The overlap of the window over the frame helps in smoothing the spectral estimate of the input speech signal and also to give a longer analysis window. There exist a number of methods to generate the feature vector at this stage. A simple block diagram showing two of these that have been used to generate feature vectors is shown in Fig. 1. Studies have shown that the most reliable LP-derived feature set for speaker recognition is the cepstral coefficients [6,7]. Cepstral features are found to yield excellent performance for text-independent speaker identification when training and testing speech signals are collected under relatively clean and stationary environments [8].

A feature vector of 61 components is developed which consists of twenty mel-cepstral coefficients, first and second order time derivatives and first and second order regression features. The mel-frequency cepstral coefficients (MFCC's) are being widely used in current speech recognition systems due to their ability to achieve better performances [26] by better approximating the human hearing perception, which follows a log scale rather than a linear scale. The mel scale is often approximated as a linear scale from 0 to 1000 Hz, and then a logarithmic scale beyond 1000 Hz [1]. The derivatives and regression features for the mel-frequency cepstral coefficients were computed using a fixed length of three frames.

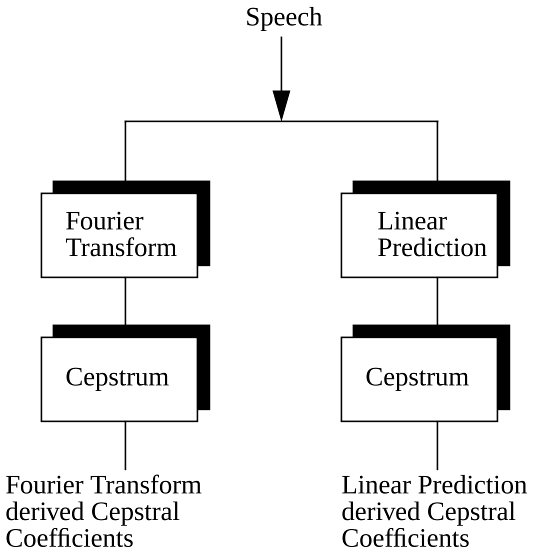
Speech
Fourier Transform
Linear Prediction
Cepstrum
Cepstrum
Fourier Transform derived Cepstral Coefficients
Linear Prediction derived Cepstral Coefficients

Fig. 1. Two methods to generate feature vectors

The prewhitening transformations and the principal component analysis are performed on these feature vectors to generate a decorrelated and normalized feature vector. We assume that the random process generating the signal parameters is a multi-variate Gaussian random process, but generally, to get better performances, a weighted sum (mixtures) of Gaussian distributions is used. A weighted sum would be able to model the input data better than a single distribution, but the associated computational complexity increases manifold.

These decorrelated and normalized feature vectors will be used to generate acoustic models of the data. Maximum likelihood classification can be used to achieve this [2]. Suppose, we have a set of classes which might be representing, say the words in a vocabulary. The probabilities for the word model given the feature vectors are calculated and then the class that has the maximum conditional probability for a given feature vector is chosen. Generally, the bigger the unit chosen, the better should be the performance of the system since a bigger unit would be able to capture the long-term context better than a smaller unit. But, when viewed in a large vocabulary recognition standpoint, this solution

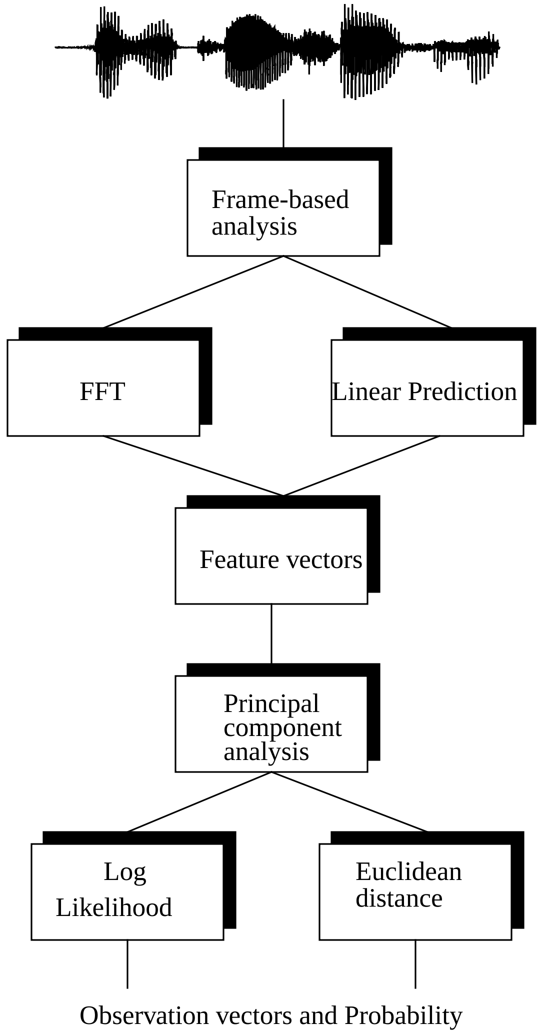

Fig. 2. A simple block diagram showing the front end and acoustic modeling

becomes impractical. Current systems are employing either phones, phonemes, bi-phones or tri-phones as the modeling unit. Context-dependent phonemes are also becoming popular with researchers. A number of distance measures exist which can be used to compute the class separations such as the Bhattacharya distance, the log-likelihood measure and the Euclidean metric.

A block diagram indicating this whole process is shown in Fig.2.

3. PRINCIPAL COMPONENT ANALYSIS

In this section, we will discuss the prewhitening transformations, which are done to decorrelate and normalize the signal parameters, and then discuss feature selection which is performed to obtain a reduced set of features. The approach will be to impose a model on the data, train this model and measure the quality of this approximation [1]. A block diagram of the various steps involved in the prewhitening transformations is given in Fig. 3.

Prewhitening Transformations [9]:

The process generating the signal parameters will be assumed to be a multi-variate Gaussian process. A Gaussian probability distribution can be defined as

$$p(\bar{v}) = \aleph[\bar{v}, \mu_v, C_v]$$

We will assume that our parameters obey this type of statistical model. A linear transformation is to be computed that will simultaneously normalize and decorrelate the signal parameters, thus allowing us to compare the feature vectors directly and also to eliminate redundancy. Let us define the transformed vector $\bar{y}$ as

$$\bar{y} = \Psi(\bar{v} - \bar{\mu}_v)$$

where $\bar{v}$ denotes the input parameter vector, and $\bar{\mu}_v$ denotes the mean value of the input parameter vector. We define Ψ as a prewhitening transformation [10, 9]. This linear operation applied to the feature vectors is called a prewhitening transformation, since it produces feature vectors whose components are uncorrelated and normalized [2].

This transformation matrix is given by

$$\Psi = \Lambda^{-1/2}\Phi$$

where Λ is a diagonal matrix of the eigen values, and Φ is the matrix of the eigen vectors of the covariance matrix of $\bar{v}$. The computation of the eigen values and eigen vectors is algorithmically highly complex [11] has an excellent discussion on this problem. So, a canned routine, which has generally provided satisfactory performance, was used to compute these [11].

The eigen values and eigen vectors can be shown to satisfy the following relation:

$$C_v = \Phi\Lambda\Phi^T$$

Eigen values, essentially, try to model the system and hence the eigen vectors can be understood to be modeling the input data. Each eigen vector attempts to model a different aspect of the speech spectrum. The first few eigen vectors of the transformation matrix attempt to model the gross spectral characteristics of the channel [1].

The covariance matrix is computed using the following equation:

$$C_v(i, j) = \frac{1}{N_f}\sum_{m=0}^{N_f-1}(v_m(i)-\mu_v(j))(v_m(j)-\mu_v(j))$$

Simplifications of the prewhitening procedure are sometimes employed to avoid the computational expense of using the full covariance matrix [2]. The most common simplification is to assume that the features are mutually uncorrelated, but inappropriately scaled relative to one another. In this case, the covariance matrix reduces to a diagonal matrix and the transformation matrix simplifies to a diagonal matrix whose off-diagonal elements are all zero and the diagonal elements will be $\frac{1}{\sigma_{v(i)}}$ where $\sigma_{v(i)}$ is the standard deviation of the *i*th component of the parameter vector $\bar{v}$. We can see from the transformation matrix that all the parameters are being normalized by their standard deviations which is essentially making each parameter count equally in the calculation. As we recall, this is known as variance-weighting and these variance-weighted cepstral coefficients are very popular in speech recognition systems. Comparisons of feature vectors can now be made directly as the variances of all the parameters have been normalized.

There exists a relation between the eigen values and the variance of the process which can be used to discard the least significant features. We can define the amount of the variance accounted for by each eigen value/eigen vector pair as [1]

$$\zeta_i = \frac{\lambda_i}{\sum_{i=0}^{N_v-1}\lambda_i}\times 100\%$$

and the total percentage of the variance accounted for by the first N_y dimensions as

$$\zeta_{N_y} = \frac{\sum_{j=0}^{N_y-1}\lambda_j}{\sum_{j=0}^{N_v-1}\lambda_i}\times 100\%$$

where N_v is the number of eigen values $\lambda_0, \lambda_1, \ldots\ldots\lambda_{N_v-1}$. The eigen values are ordered in decreasing order and the least significant features (the number of which has to be decided based on how much information they contribute) are discarded. This results in a reduced transformation matrix thereby resulting in a decrease in the computational complexity.

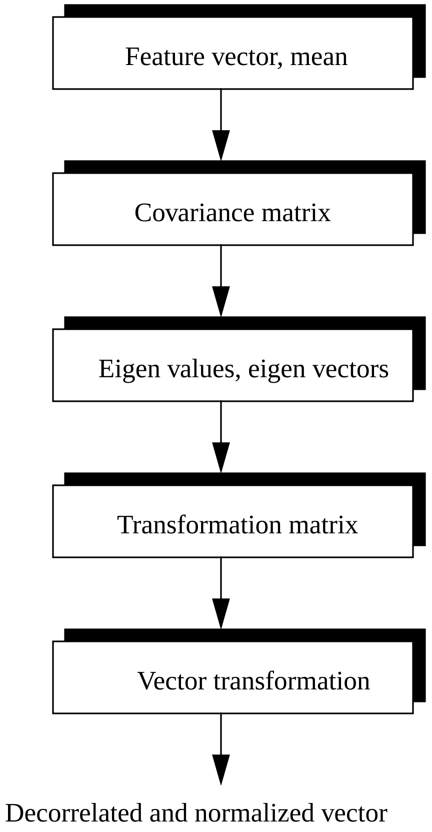

Fig. 3. Various steps in Prewhitening transformation

4. CHANNEL ADAPTATION

Channel adaptation is gaining prominence in research groups as attempts to improve the performance of speech recognition systems are increasing. Most channel distortions and many kinds of additive noise vary slowly compared to the variations naturally occurring in speech [12]. There are a number of methods which make use of this fact to improve the performance of speech recognition systems. Recent studies have shown that filters which remove slow variations in the feature vectors used in speech recognition systems can yield significantly improved recognition rates [13-17].

There are primarily three adverse conditions a speech recognition system often encounters, namely, noise, distortion and articulation effects [3]. Acoustic ambient noise is generally considered additive and it is one of the primary concerns of a speech recognizer. Distortions are introduced due to the recording environment which includes the recording room’s characteristics which affect the spectral characteristics and also the type and mounting position of the microphone which can significantly alter the speech spectrum. There are many factors that influence the sound format and rhythmic stability of the speaker [18].

One way of implementing channel adaptation is to estimate the spectral characteristics of noise by analyzing the non-speech portions of the spectrum, which could be used to introduce some sort of alleviating measure in the system. However, this approach requires that we have a noisy data-base to train the system.

There are many methods which use simple filtering operations to achieve channel adaptation like RASTA and CMN. They try to alleviate the features that are more susceptible to environmental variations by filtering operations, thereby reducing the effect of the acoustical environment on the performance of the system.

A few methods currently in use by various systems will be introduced here and then the CMN and RASTA methods will be looked into in a more detailed manner.

4.1. CDCN, SDCN & FCDCN:

Acero [19] proposed the code-word dependent cepstral normalization algorithm as a pre-processing technique that can eliminate the effects of linear filtering and additive noise. CDCN uses EM techniques to compute the Maximum likelihood estimates of the environmental parameters that characterize the contributions of the contributions of additive noise and linear filtering. These environmental parameters are chosen to best match (in the MS sense) to match the ensemble of cepstral vectors of the incoming speech to the ensemble of cepstral vectors in a universal code-book generated from the training corpus.

Acero [19,20] also proposed several pre-processing techniques that compensate for environmental mismatches by translating the noisy-testing speech to the acoustical space of the training environment in an environment-specific manner. Two of these environment-specific algorithms are the SNR-dependent cepstral normalization (SDCN) and Fixed codeword-dependent cepstral normalization (FCDCN) algorithms.

SDCN applies an additive correction in the cepstral domain, with the compensation vector depending exclusively on the instantaneous SNR of the signal. The compensation vectors equal the difference of the average cepstra between simultaneous stereo recordings of speech signal from both the training and testing environments for each SNR of speech. At high SNRs, this compensation vector primarily compensates for differences in spectral tilt between the training and the testing environments, while at low SNRs the compensation vector provides a form of noise subtraction.

The FCDCN algorithm combines some of the more attractive features of the CDCN and SDCN algorithms. Like SDCN, the compensation factor equals the difference in cepstra between the training and testing environments, but like CDCN, the compensation factor is different for different VQ codewords as well.

4.2. Adaptive Component Weighting (ACW):

The ACW [8] scheme modifies the linear predictive spectral components so as to emphasize the formant structure by attenuating the broad-bandwidth spectral components. Such components are found to introduce undesired variability in the LP spectra of speech signals due to environmental factors. The ACW cepstral coefficients represent an adaptively weighted version of the LP cepstrum. The adaptation results in deemphasizing the irrelevant variations of the LP cepstral coefficients on a frame-by-frame basis. The ACW method has been shown to offer better performance as compared to other common methods of cepstral weighting [8].

4.3. CEPSTRAL MEAN NORMALISATION:

Cepstral mean normalization (CMN) [21] provides a very simple way of implementing channel adaptation making use of the fact that noise varies slowly as compared to speech. It is a simple scheme that compensates for channel mismatches (due to microphone/speaker variability).

Cepstral mean normalization tries to remove the bias in the features introduced by the noise component in the input signal. This is done by computing the long-term mean of the input cepstral vectors and then subtracting this mean value from the cepstral vectors.

If $c_i(t)$ is the input cepstral vector, then it's mean can be computed by using the following equation:

$$\mu_i = \frac{1}{N_f} \sum_{m=0}^{N_f - 1} c_i(t)$$

where N_f is the number of frames.

Then, the mean is subtracted from the cepstral vectors as in the following equation:

$$c'_i(t) = c_i(t) - \mu_i$$

If we approximate the mean vector μ_i by M_i, the average value of μ_i over the training corpus, we have

$$c'_i(t) \approx c_i(t) - M_i$$

To get efficient performance with this technique, the above steps must be iterated over clean and noisy data. This method can be used for channel adaptation even if a noisy database is not available. All the above steps have to be iterated over data from different channels, so that the speech recognizer gets adapted to those particular channels.

One point to be noted in this context is that cepstral mean normalization is implemented in the speech recognition system in the principal component analysis stage itself. If we recall carefully, we subtract the mean from the input vector while doing the prewhitening transformation which is equivalent to doing cepstral mean normalization. Though, this technique is built into the system, a separate module has been developed to illustrate the working of this technique.

4.4. *RelAtive SpecTrAl processing (RASTA):*

RASTA [22] is another simple way of achieving channel adaptation. This method is based on the filtering of the cepstral coefficients by a RASTA filter which aims to filter out the slowly varying components of the cepstral coefficients in order to normalize environmental variations. Coupled with a bandpass liftering operation, this technique offers better performance than many RASTA derived techniques. A simple block diagram giving the details of this implementation is given in Fig. 4.

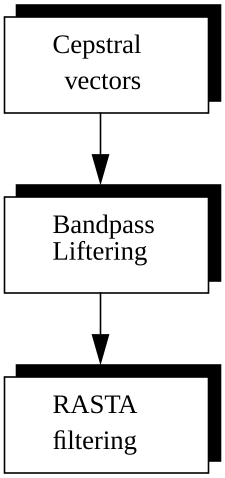

Fig. 4. A simple RASTA scheme

Given the cepstral vectors, we first bandpass lifter [23] them with the following window function:

$$w(k) = 1 + h\sin(\pi(k/L))$$

where, h = L/2, k = 1,2......L and $w(k) = 0$ for other k, with L being the number of cepstral coefficients.

Then these cepstral vectors will be processed through the following filter:

$$\frac{a_0 + a_1 z^{-1} + a_3 z^{-3} + a_4 z^{-4}}{z^{-4}(1 - b_1 z^{-1})}$$

with the coefficients chosen to approximate a bandpass frequency response. This filtering operation is known as the RASTA filtering [24].

Liftering is nothing but “low-time liftering” (analogous to low-pass filtering in the frequency domain) in the quefrency domain which is just a variation of the frequency domain.

It is known that the higher order cepstral coefficients have less discriminating power and the lower order coefficients are more susceptible to environmental variations. The bandpass liftering operation deemphasizes both the higher and lower order coefficients thereby reducing the susceptibility of the system to environmental variations.

The RASTA filter is used to smooth all the cepstral coefficients by a bandpass filtering operation thereby attempting to remove the effects of the channel and the transducer. While the bandpass liftering operation offers a static correction, the RASTA filtering operation offers a dynamic correction to the cepstral coefficients. Thus, by combining the static and dynamic techniques, we obtain the benefits of both techniques. It is interesting to note that the combination of these two techniques has been proven to give better recognition performance [22] than when using RASTA alone [24] and certain variations of RASTA.

One disadvantage of RASTA is that the performance of the recognizer degrades if the experiment is conducted under matched conditions. CMN doesn’t suffer from this disadvantage.

5. SOFTWARE DETAILS:

Software which can implement all of the previously mentioned tasks has been developed. Though, the objective was to develop software which would be later integrated into the entire speech recognizer system, there exist separate modules which can be used to implement specific portions of the afore-mentioned tasks.

The software has been developed keeping in mind that it has to be integrated into the system. The software is essentially data-driven thereby giving the control of the parameters to the user. It is easy to make any changes to the existing code so as to suit the specific needs of the user thereby allowing greater experimentation.

6. SUMMARY AND FUTURE RESEARCH

The importance of the implementation of statistical modeling techniques and channel adaptation techniques towards the building of an efficient speech recognizer has been discussed. The statistical modeling techniques discussed here are the typical ones most of the current speech recognition systems make use of. Though, some systems make use of more sophisticated statistical models, these are the basic statistical models that any system could be using. Much research is going on in the field of robust speech recognition as there has been significant improvement in the performance of speech recognition systems using channel adaptation techniques and similar other techniques. The channel adaptation techniques discussed in this paper are very basic and very simple to implement. Studies are being conducted on various modified versions of these techniques to achieve better recognition performance.

Software that can perform principal component analysis and channel adaptation techniques exists. Detailed experimentation will be performed on the code to make it more generic and to improve the performance of the code in general. Once the various modules of the speech recognizer are developed, detailed experimentation will be performed with real data so as to introduce more sophisticated models and techniques. A simple web-tool is planned to be developed to facilitate the users to experiment with, by making use of the various modules that have been developed for this project and by adding a few more.

7. ACKNOWLEDGEMENTS

I would like to thank Dr. Joseph Picone for the time and effort he put into this project all through and also for his patience in answering all my questions.

8. REFERENCES

1. J. Picone, “Signal Modeling Techniques in Speech Recognition”, in *Proc. ICASSP*, vol. 81, no. 9, pp. 1215-1247, Sep. 1993.

2. J.R. Deller, J.G. Proakis, and J.H.L. Hansen, *Discrete Time Processing of Speech signals*. New York: MacMillan, 1993.

3. B.H. Juang, “Speech Recognition in Adverse Environments”, *Computer Speech and Language*, Vol 5, No 3, pp 275-294, 1991.

4. A. Acero, R.M. Stern, “Environmental robustness in Automatic speech recognition”, *Proc ICASSP*, pp 849-852, Apr 1990.

5. A. Erell, M. Weintraub, “Estimation using Log-spectral-distance criterion for Noise-robust speech recognition”, *Proc ICASSP*, pp 853-856, Apr 1990.

6. B. Atal. Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification. *Journal of the Acoustical Society of America*, 55: 1304-1312, Jun 1974.

7. S. Furui. Cepstral analysis technique for automatic speaker verification. *IEEE Trans. Acoustics, Speech and Signal Processing*, ASSP-29(4):254-272,Apr 81.

8. K.T. Assaleh, R.J. Mammone, “Robust cepstral features for speaker identification”, *Proc ICASSP*, pp 129-132, Apr 94.

9. K. Fukunaga, *Introduction to Statistical Pattern Recognition*. New York: Academic Press, 1972.

10. E.L. Bocchieri and G.R. Doddington, “Frame specific statistical features for speaker independent speech recognition”, *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, no.4, pp 755-764, Aug. 1996.

11. W.H. Press, B.P. Flannery, S. A. Teukolsky, and W.T. Vettering, *Numerical Recipes in C: The Art of Scientific Programming.* New York: Cambridge Univ. Press, 1988.

12. B. A. Hanson, T.H. Applebaum, “Subband or ceps tral domain filtering for recognition of Lombard and Channel-distorted speech”, *Proc. ICASSP*, Vol. 2, pp 79-82, Apr 1993.

13. H. Hermansky, N. Morgan, A. Bayya, and P. Kohn, “Compensation for the effect of the communication channel in auditory-like analysis of speech (RASTA-PLP)”, *Proc EUROSPEECH*, pp 1367-1370, 1991.

14. H. G. Hirsch, P.Meyer, H.W. Ruehl, “Improved speech recognition using high-pass filtering of subband envelopes”, *Proc. EUROSPEECH*, pp 413-416, 1991.

15. D. Geller, R. Haeb-Umbach, H. Ney, “Improvements in speech recognition for voice dialing in the car environment”, *Proc. ESCA Workshop on Speech Processing in Adverse Conditions*, pp 203-206, Nov 1992.

16. K. Aikawa, H. Kawahara, Y. Tohkura, “Dynamic cepstrum parameter incorporating time-frequency masking and its application to speech recognition”, *Journal of the Acoustical Society of America*, 92:2476 (5pSP5), Fall 1992.

17. H. Murveit, J. Butzburger, M. Weintraub, “Reduced

channel dependence for speech recognition", *Proc. DARPA Speech and Natural Language Workshop*, Feb 1992.

18. I. Lecome et al, " Car noise processing for speech input", *Proc. ICASSP*, pp 512-515, 1989.

19. A. Acero, "Acoustical and Environmental Robustness in Automatic Speech Recognition", Ph.D Thesis, Dept. of ECE, Carnegie Mellon University, Sep 1990.

20. A. Acero, R. Stern, "Robust speech recognition by normalization of the acoustical space", *Proc. ICASSP*, pp 893-896, May 1991.

21. R.A. Gopinath, M. Gales, P.S. Gopalakrishnan, S. Balakrishnan-Aiyer and M.A. Picheny, "Robust speech recognition in noise-performance of the IBM continuous speech recognizer on the ARPA noise spoke task", *Proc of ARPA Spoken Language Systems*, pp 127-130, 1994.

22. Y.Kao, J.S. Baras, P.K. Rajasekaran, "Robustness study of free-text speaker identification and verification", *Proc ICASSP*, pp 379-382, Apr 1993.

23. B. Juang, L.R. Rabiner, J.G. Wilpon, "On the use of bandpass liftering in speech recognition", *IEEE Trans. Acoustics, Speech and Signal Processing*, ASSP-35(7), Jul 1987.

24. H. Hermansky, N. Morgan, A. Bayya, P. Kohn, "RASTA-PLP analysis technique", Proc ICASSP, 1993.

25. S. Young, "Large Vocabulary Continuous Speech Recognition: a Review", To appear in *IEEE Signal Processing Magazine*, 1996.

26. A.J. Robinson, J. Holdsworth, R. Patterson, F. Fallside, " A comparison of preprocessors for the Cambridge Recurrent Error Propagation Network Speech Recognition System", *Proc International Conference on Spoken Language Processing*, Nov 1990.