Neural Network Classifiers and Principal Component Analysis for Blind Signal to Noise Ratio Estimation of Speech Signals

Matthew Marbach
Lockheed Martin
matthew.marbach@lmco.com

Russell Ondusko
Navsea
dachande96@gmail.com

Ravi P. Ramachandran and Linda M. Head
Rowan University
ravi@rowan.edu, head@rowan.edu

***Abstract*— A blind approach for estimating the signal to noise ratio (SNR) of a speech signal corrupted by additive noise is proposed. The method is based on a pattern recognition paradigm using various linear predictive based features, a neural network classifier and estimation combination. Blind SNR estimation is very useful in speaker identification systems in which a confidence metric is determined along with the speaker identity. The confidence metric is partially based on the mismatch between the training and testing conditions of the speaker identification system and SNR estimation is very important in evaluating the degree of this mismatch. The aim is to correctly estimate SNR values from 0 to 30 dB, a range that is both practical and crucial for speaker identification systems. Speech corrupted by additive white Gaussian noise, pink noise and two types of bandpass channel noise are investigated. The best individual feature is the vector of line spectral frequencies. Combination of the estimates of 3 features lowers the estimation error to an average of 3.69 dB for the four types of noise.**

I. Introduction

Consider a speech signal corrupted by additive noise that is statistically independent of the signal. This noisy signal is characterized by a signal to noise ratio (SNR) calculated over the entire duration of the signal. In this paper, a pattern recognition approach using six linear predictive (LP) [1] derived features is used to blindly estimate the SNR of the noisy speech signal. Principal component analysis (PCA) [2][3] of the feature vectors is shown to improve the SNR estimate and reduce the dimension of the feature vector. In addition, a further performance improvement is achieved by combining the SNR estimates generated by the six features. A multilayer perceptron (MLP) neural network [4] classifier is used.

Blind estimation of the SNR is very useful in closed set speaker identification systems. The training of a speaker identification system involves the configuration of M models each representing a different speaker. During closed set testing, the features of an utterance are compared to the M models to render a decision of the speaker identity as being one of the M speakers [5]. Recent research has been done to develop techniques to calculate a confidence metric to accompany the decision of the speaker identity [6][7]. The confidence metric is calculated based on the mismatch between training and testing conditions, amount of training and testing data, and number of speakers (value of M). As M increases, there is usually more model overlap. The more the difference between the SNR of the training and testing speech, the more the mismatch between the two and the lower the confidence metric. An automatic and blind method of SNR estimation of the training and testing speech is an integral part of the technique of finding the confidence metric of a speaker identification system. Estimating the SNR has also been found to be very useful in noise spectrum estimation for speech enhancement and in robust speech recognition. The novel method proposed in this paper for blind SNR estimation is based on a pattern recognition paradigm.

II. Overview of Pattern Recognition System

A pattern recognition system consists of a front-end feature extractor and a classifier. The feature extractor transforms the speech signal to a collection of low dimension feature vectors such that vectors from the same class are similar and a clear distinction among vectors from different classes exists. Examples of classes depend on the application. For vowel recognition, each class is a different vowel. For speaker recognition, each class is a different speaker. In this paper, each class is a different SNR value.

For an M class problem, the classifier is trained such that a model is configured for each class. In unsupervised training, the model for class k is trained using feature vectors from class k only. An example is vector quantization in which the feature vectors for each class are grouped into clusters. In supervised training, the model for class k generally uses feature vectors for all M classes. An example is the neural network which uses a discriminator based method that divides the feature space into distinct regions by a series of hyperplanes. Each region corresponds to a particular class. After training is complete, the classifier uses test feature vectors to render a decision on the class the features belong to. In vector quantization, the cluster whose distance is closest to the test data identifies the class. In neural networks, test feature vectors falling into a specific region are deemed to have been generated by the corresponding class.

Features based on LP analysis are highly useful candidates for SNR estimation as they show differences for varying noise levels. The goal is to configure a system to estimate the SNR over an entire utterance which would be part of a larger speaker identification system. The overall SNR estimation system consists of five components, namely, (1) Linear predictive (LP) analysis, (2) Feature extraction for ensuring SNR discrimination, (3) Principal Component Analysis (PCA) to get a transformed set of feature vectors that achieve better SNR discrimination, (4) Multilayer Perceptron (MLP) classifier and decision logic for computing the SNR estimate and (5) Combination of the SNR estimates of the different features to get a final estimate. During training, an MLP network is trained for each distinct SNR value using feature vectors obtained from noisy speech corresponding to that particular SNR (class label of 1) and other SNRs (class label of 0). During testing or SNR estimation, the input to the system will be a noisy speech signal with an unknown SNR. After LP analysis, feature extraction and PCA, the set of feature vectors will be passed through each MLP to get an overall score for each MLP. Based on these scores, the output will be an estimated SNR value. A MLP classifier is trained separately for each feature and leads to an SNR estimate for each feature. A comparison of different LP based features is done with respect to the average absolute error between the actual and estimated SNR. The features considered [1][8][9] include the line spectral frequencies (LSFs), reflection coefficients (REFL), log area ratios (LAR), linear predictive cepstrum (CEP), adaptive component

978-1-4244-3828-0/09/$25.00 ©2009 IEEE

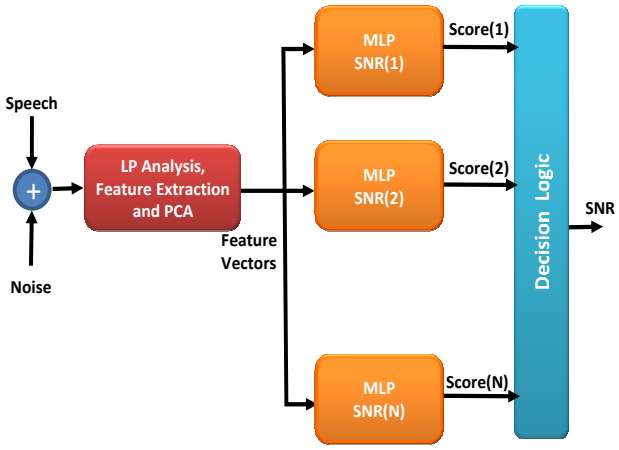

Fig. 1. Block diagram for SNR estimation using a single feature

weighted cepstrum (ACW) and the postfilter cepstrum (PFL). The SNR estimates of the individual features are combined to get an even better estimate in that the average absolute error is further reduced.

III. SNR Estimation System

Figure 1 shows the block diagram for score and SNR determination for a single feature.

A. Linear Prediction and Feature Extraction

Linear predictive (LP) analysis results in a stable all-pole model $1/A(z)$ of order p where

$$A(z) = 1 - \sum_{n=1}^{p} a(n) z^{-n} \tag{1}$$

The autocorrelation method of LP analysis gives rise to the predictor coefficients $a(n)$ and the REFL feature $refl(n)$ for $n = 1$ to p. The LAR feature is found as

$$lar(n) = \log \left[\frac{1 - refl(n)}{1 + refl(n)} \right] \tag{2}$$

for $n = 1$ to p. The LSF feature $lsf(n)$ are the angles (between 0 and π) of the alternating unit circle roots of $F(z)$ and $G(z)$ [1] where

$$\begin{aligned} F(z) &= A(z) + z^{-(p+1)} A(z^{-1}) \\ G(z) &= A(z) - z^{-(p+1)} A(z^{-1}) \end{aligned} \tag{3}$$

The predictor coefficients $a(n)$ are converted to the LP cepstrum $clp(n)$ $(n \geq 1)$ by an efficient recursive relation [1]

$$clp(n) = a(n) + \sum_{i=1}^{n-1} (\frac{i}{n}) clp(i) a(n-i) \tag{4}$$

Since $clp(n)$ is of infinite duration, the CEP feature vector of dimension p consists of the components $clp(1)$ to $clp(p)$ which are the most significant due to the decay of the sequence with increasing n.

The first step in developing the ACW cepstrum [8] is to perform a partial fraction expansion of the LP function $1/A(z)$ to get

$$\frac{1}{A(z)} = \sum_{n=1}^{p} \frac{r_n}{1 - p_n z^{-1}} \tag{5}$$

where p_n are the poles of $A(z)$ and r_n are the corresponding residues. The variations in r_n were removed by forcing $r_n = 1$ for every n. Hence, the resulting transfer function is a pole-zero type of the form

$$\begin{aligned} \frac{N(z)}{A(z)} &= \sum_{n=1}^{p} \frac{1}{1 - p_n z^{-1}} \\ &= p \left[\frac{1 - \sum_{n=1}^{p-1} b(n) z^{-n}}{1 - \sum_{n=1}^{p} a(n) z^{-n}} \right] \end{aligned} \tag{6}$$

Applying the recursion in Eq. (4) to $b(n)$ and $a(n)$ results in two cepstrum sequences $cb(n)$ and $clp(n)$ respectively. The ACW cepstrum is [8]

$$cacw(n) = clp(n) - cb(n) \tag{7}$$

The postfilter is obtained from $A(z)$ and its transfer function is given by

$$H_{pfl}(z) = \frac{A(z/\beta)}{A(z/\alpha)} \tag{8}$$

where $0 < \beta < \alpha \leq 1$. The cepstrum of $H_{pfl}(z)$ is the PFL cepstrum which is equivalent to weighting the LP cepstrum as [9]

$$cpfl(n) = clp(n)[\alpha^n - \beta^n] \tag{9}$$

The ACW feature $cacw(n)$ and PFL feature $cpfl(n)$ are taken from $n = 1$ to p.

B. Principal Component Analysis (PCA)

Principal component analysis (PCA) [2][3] has been shown to improve vowel recognition and speaker identification performance. For each of the six features, PCA is performed by first finding the covariance matrix $\mathbf{T}$ of the feature vectors. The feature vectors for all SNR levels of the training speech data are used to calculate $\mathbf{T}$. No data compression is performed. The concept of using data from all SNR levels to form a global covariance matrix is similar to incorporating data from all classes when using PCA for vowel recognition [2]. The matrix $\mathbf{T}$ is found purely from the training data and is different for each of the six features.

A linear transformation matrix $\mathbf{U}$ has the eigenvectors of $\mathbf{T}$ as its rows such that the rows are arranged in decreasing order of the eigenvalues of $\mathbf{T}$. The submatrix $\mathbf{U}_q$ contains the first q rows of $\mathbf{U}$. Given that p is the order of LP analysis and the dimension of a feature vector $\mathbf{x}$ (note that $\mathbf{x}$ is a column vector), the transformed feature vector $\mathbf{y} = \mathbf{U}_q \mathbf{x}$. Every feature vector is transformed such that a subset of $q < p$ components corresponding to the largest q eigenvalues of $\mathbf{U}$ are retained. The six features are individually compared with respect to performance improvement and dimensionality reduction as a result of PCA. The reduced dimension q that leads to the best individual performance is used for the last step of combining the SNR estimates of the features as described later.

C. MLP Classifier and Decision Logic

A MLP classifier consists of a parallel arrangement of N MLP networks that are individually trained for each candidate SNR value. Referring to Fig. 1, the ith MLP that is dedicated to a value SNR(i) is trained with feature vectors representing SNR(i) (class label of 1) and feature vectors representing SNR(j) for all $j \neq i$ (class label of 0). The training is accomplished by the back-propagation algorithm [4]. The feature vectors for a particular SNR are computed from speech corrupted by additive noise with that particular SNR.

During testing or score determination, a test noisy speech utterance of a particular SNR is converted to a set of test feature vectors. Consider a particular test feature vector. This is processed by each of the N MLP networks to get N different scores that are either 0 or 1. This process is repeated for every test feature vector. The scores are accumulated over the entire set of test feature vectors. Referring to Fig. 1, Score(i) is the accumulated score for the ith MLP.

Two methods of implementing the decision logic are investigated. A hard decision approach estimates the SNR to correspond to the MLP which renders the maximum score. In the soft decision approach, the scores from a subset of the N MLP networks are used to estimate the SNR. Consider the ith MLP trained with a class label of 1 for the value SNR(i) and rendering Score(i). Let Ind(i) denote the indicator function which equals 1 if the ith MLP is used for final SNR computation. Otherwise, Ind(i) equals 0. A probability Prob(i) is derived from Score(i) by the equations

$$\begin{aligned} \text{Total} &= \sum_{j=1}^{N} \text{Ind}(j)\ \text{Score}(j) \\ \text{Prob}(i) &= \frac{\text{Ind}(i)\ \text{Score}(i)}{\text{Total}} \end{aligned} \tag{10}$$

If an MLP is not used, the probability assumes a value of 0. The probabilities add up to 1. The experiments revealed that using the three MLPs with the largest scores led to good results. From the probabilities, the SNR is estimated as

$$\text{SNR} = \sum_{j=1}^{N} \text{Prob}(j)\ \text{SNR}(j) \tag{11}$$

For each test utterance, an absolute error between the true SNR and the estimated SNR is found. The performance measure is a mean value of this absolute error taken over the total number of test speech utterances.

D. Estimation Combination

Using hard or soft decision, six SNR estimates are found for each test speech utterance, one for each feature. A combination estimate is obtained by taking the mean, median and trimmed mean of all six or any subset of the individual feature SNR estimates. The trimmed mean is the mean of the estimates with the highest and lowest estimates not counted. It is only valid when three or more features are considered. The aim is to see if all or a subset of the features contribute to a better final SNR estimate. This approach of combining the outputs of different classifiers comes under the realm of ensemble based systems [10].

IV. Experimental Protocol

Ten sentences from each of the 38 speakers from the New England dialect of the TIMIT database are used for the experiments. The speech in this database is clean and first downsampled from 16 kHz to 8 kHz. Four types of additive noise are considered, namely, white Gaussian noise, pink noise and two types of bandpass noise. The bandpass noise has a spectrum corresponding to the frequency response of a typical telephone channel. The two types of bandpass noise used correspond to the Continental Poor Voice (CPV) channel and the Continental Mid Voice (CMV) channel [11].

For each speaker in the database, there are 10 sentences. The first five are used for training the MLP classifier system. The remaining five sentences are individually used for testing. For each type of noise, 190 sentences are used for training at each distinct SNR value. Similarly, for each type of noise, a different set of 190 sentences are used for testing at each distinct SNR value. The goal is to correctly estimate SNR values between 0 and 30 dB (inclusive). This is a significant range for practical speaker identification systems. For each utterance, the absolute error is the absolute difference between the true SNR and the estimated SNR. For each SNR value tested, there are 190 utterances over which an average absolute error (AAE) is obtained. The AAE is found for each individual feature (with and without PCA) and each type of noise using both the hard and soft decision approaches. The AAE is also found for various combination estimates.

The feature extraction step for a speech utterance for both training and testing is as follows. One of the four types of noise at a particular SNR is added to the clean speech. The noisy speech is preemphasized by using a nonrecursive filter $1 - 0.95z^{-1}$. For the LP analysis, the autocorrelation method [1] is used to get a 12th order LP polynomial $A(z)$. The LP analysis is done over frames of 30 ms duration. The overlap between frames is 20 ms. The LP coefficients are converted into 12 dimensional LSF, REFL, LAR, CEP, ACW and PFL feature vectors. For the PFL feature, $\alpha = 1$ and $\beta = 0.9$ (see Eq. (8)). The feature vectors are computed only in voiced frames that are selected based on energy thresholding.

The MLP classifier is trained with the 12 dimensional feature vectors using the back-propagation algorithm [4]. A separate classifier is used for each feature. As mentioned earlier, this leads to an ensemble of classifiers. With PCA, the feature vectors are transformed by the matrix $\mathbf{U}_q$ and a dimension $q \leq 12$ is selected. Consider the training of one classifier dedicated to a particular feature. There are $N = 33$ MLP networks trained such that the ith MLP is trained using a class label of 1 with feature vectors computed from speech with SNR$(i) = i - 2$ dB for $1 \leq i \leq 33$. Training feature vectors with a class label of 0 correspond to SNR$(i) \neq i - 2$ dB. The 33 MLP networks are trained in 1 dB increments. Although the aim is to correctly estimate SNR values from 0 to 30 dB, training from -1 to 31 dB is performed to avoid artificially high errors for speech with SNR values of 0 and 30 dB. For every noise condition used in training, there are 128 feature vectors for each class label that are used to train the ith MLP. For a class label of 1, the total number of computed feature vectors is compressed to a size of 128 using the Linde-Buzo-Gray (LBG) algorithm. For a class label of 0, 128 feature vectors are used with 4 vectors corresponding to each SNR level not represented by the class label of 1. The 4 vectors are again obtained by compression using LBG. Experiments with 10, 40, 70 and 100 hidden layer nodes are performed with the goal of maximizing performance.

During testing or score determination, a test noisy speech utterance of a particular SNR is converted to a set of test feature vectors. Transformation by the matrix $\mathbf{U}_q$ and appropriate dimensionality reduction are performed when PCA is involved. As described earlier, the score for the ith MLP is Score(i). These scores are used for the hard and soft decisions.

V. Results

For each experiment, an average absolute error (AAE) is computed for test speech having SNR values between 0 and 30 dB in 1 dB increments. There are a total of 31 AAE values and an average of these values result in an overall average absolute error (OAAE). Due to the training algorithm of the neural network and the random initialization of the interconnected weights, the neural network will never draw exactly the same decision boundaries between classes twice. To gain a generalization of the performance, 5 trials are performed for each experiment. The results shown in this section represent the mean OAAE over the 5 trials.

Feature	Type of Noise			
	AWGN	Pink	CPV	CMV
LSF	7.30	7.10	7.20	5.87
REFL	6.62	4.45	5.47	6.14
LAR	8.19	5.58	6.94	6.33
CEP	6.29	5.58	6.83	7.22
ACW	5.67	4.75	4.92	8.09
PFL	5.55	4.51	4.95	6.25

TABLE I
SOFT DECISION OAAE VALUES (IN DB) FOR TESTED NOISE CONDITIONS WITHOUT PCA

Feature	Dimension	Type of Noise			
		AWGN	Pink	CPV	CMV
LSF	4	3.86	4.21	4.10	4.40
REFL	6	4.46	3.18	4.38	6.10
LAR	6	5.09	4.92	4.96	5.20
CEP	5	4.13	4.02	3.94	4.86
ACW	6	5.23	5.52	5.61	9.09
PFL	3	4.50	5.14	4.53	5.54

TABLE II
SOFT DECISION OAAE VALUES (IN DB) FOR TESTED NOISE CONDITIONS WITH PCA

The MLPs are trained using feature vectors from speech corrupted by additive white Gaussian noise (AWGN), pink noise and bandpass CPV noise. Table I gives the OAAE for each feature and each tested noise condition when no PCA is done. Analogous results are given in Table II with PCA done for each feature. The dimension of the feature vector leading to the best performance for AWGN, pink and CPV noise is also given. Since soft decision outperforms hard decision, Tables I and II only give the soft decision results. When PCA is not done, the number of hidden layer nodes that maximized performance over AWGN, pink noise and CPV noise is 40 for LAR and 10 for all other features. When PCA is done, 10 hidden layer nodes are required for the REFL, LAR and CEP features, 40 for LSF and 70 for ACW and PFL. The bandpass CMV noise condition is not reflected in training, in determining the number of hidden layer nodes or in determining the best dimension for PCA and hence, gives an indication of which features are more robust. The LSF is the best individual feature with and without PCA. Also, PCA generally improves performance.

Combination estimates using all six features and all possible subsets of the features were attempted. In generating the individual SNR estimates, PCA and soft decision were used. The median combination estimate gave the best results. The best four subsets are presented in Table III.

The OAAE values were recalculated for different test SNR ranges as given in Table IV for the LSF/CEP combination estimate. The larger errors in SNR estimation consistently occur when the SNR of the speech is high (26-30 dB). Very noisy speech with an SNR of less than 5 dB can also lead to a relatively larger estimation error.

Features	Type of Noise			
	AWGN	Pink	CPV	CMV
LSF CEP	3.09	3.46	3.82	4.38
LSF CEP PFL	3.66	3.54	3.37	4.20
LSF CEP LAR	3.95	3.43	3.42	4.11
LSF CEP ACW	3.73	3.45	3.51	4.21

TABLE III
OAAE VALUES (IN DB) FOR THE BEST COMBINATION ESTIMATES.

SNR Test Range (dB)	Type of Noise			
	AWGN	Pink	CPV	CMV
0 to 5	1.33	3.69	3.06	5.11
0 to 10	1.73	3.55	4.08	4.59
0 to 15	1.99	3.31	4.05	4.05
0 to 20	2.10	3.00	3.69	3.61
0 to 25	2.44	2.95	3.45	3.72
5 to 25	2.72	2.76	3.62	3.36

TABLE IV
OAAE VALUES (IN DB) FOR DIFFERENT SNR TEST RANGES FOR THE LSF/CEP COMBINATION ESTIMATE.

VI. SUMMARY AND CONCLUSIONS

The MLP based pattern recognition approach to blind SNR estimation has given very good results. It is important to use PCA and combine the soft decision estimates to substantially bring down the OAAE.

VII. ACKNOWLEDGEMENT

This work was supported by the U.S. Air Force Research Laboratory, Rome NY, under contracts FA8750-05-C-0029 and F30602-03-C-0067.

VIII. REFERENCES

1) T. F. Quatieri, *Discrete Time Speech Signal Processing Principles and Practice* Prentice Hall PTR, 2002.
2) X. Wang and K. K. Paliwal, "Feature extraction and dimensionality reduction algorithms and their applications in vowel recognition", *Pattern Recognition*, vol. 36, pp. 2429–2439, 2003.
3) P. Ding and L. Zhang, "Speaker recognition using principal component analysis", *Int. Conf. on Neural Information Processing*, Shanghai, China, 2001.
4) S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice-Hall, 1999.
5) J. P. Campbell, "Speaker recognition: A tutorial", *Proc. IEEE*, vol. 85, pp. 1437–1462, September 1997.
6) M. C. Huggins and J. J. Grieco, "Confidence Metrics For Speaker Identification", *Int. Conf. on Spoken Language Processing*, Denver, Colorado, 2002.
7) M. C. Huggins and J. J. Grieco, "Speaker Identification Confidence Metrics For Heterogeneous Model Spaces", *Proc. of the 8th World Multiconference on Systemics, Cybernetics and Informatics*, Orlando, Florida, pp. 440–443, July 2004.
8) K. T. Assaleh and R. J. Mammone, "New LP-derived features for speaker identification", *IEEE Trans. on Speech and Audio Proc.*, vol. 2, pp. 630–638, October 1994.
9) M. S. Zilovic, R. P. Ramachandran and R. J. Mammone, "Speaker identification based on the use of robust cepstral features obtained from pole-zero transfer functions", *IEEE Trans. on Speech and Audio Proc.*. vol. 6, pp. 260–267, May 1998.
10) R. Polikar, "Ensemble based systems in decision making", *IEEE Circuits and Systems Magazine*, vol. 6, no. 3, pp. 21–45, 2006.
11) J. Kupin, "A wireless simulator (software)," CCR-P, April 1993.