PERGAMON

Pattern Recognition 34 (2001) 499–507

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Cochannel speaker count labelling based on the use of cepstral and pitch prediction derived features

Michael A. Lewis[a], Ravi P. Ramachandran[b,*]

[a]*Department of Electrical Engineering, City University of New York, NY, USA*
[b]*Department of Electrical and Computer Engineering, Rowan University, 201 Mullica Hill Road, Glassboro, NJ, USA*

Received 19 June 1999; received in revised form 21 June 1999; accepted 19 November 1999

Abstract

Cochannel interference of speech signals is a common practical problem particularly in tactical communications. Ideally, separation of the individual speech signals is desired. However, it is known that when two equal bandwidth signals are added, such a separation is not possible. We examine the problem of identifying temporal regions or frames as being either one-speaker or two-speaker speech. This identification is important in making automatic speaker and speech recognition systems more robust and is based on feature extraction and subsequent classification as is done in pattern recognition. The research has looked into both the closed-set problem where the identity of the tow interfering speakers are known a priori and the more difficult open-set problem where the identities are not known (speaker independent). For the feature extraction step, we propose a new pitch prediction feature (PPF) which is compared with the linear Predictive cepstral coefficients (LPCC) and the mel frequency cepstral coefficients (MFCC). The features are computed and classified on a frame-by-frame basis. We compare the performance of two classifiers, namely, the neural tree network (NTN) and vector quantizer (VQ). The results show that in both the closed-and open-set cases, (1) the VQ is the better classifier and (2) the PPF outperforms both the MFCC and LPCC features. The superiority of the PFF comes with the added benefits of using a scalar feature as opposed to the 12-dimensional vectorial LPCC and MFCC features and a lower VQ codebook size.

Keywords: Cochannel interference; Speaker count; Pitch prediction; Linear prediction; Cepstrum; Pattern recognition; Vector quantizer; Neural network

1. Introduction

Cochannel interference is a situation that develops when a speech signal is corrupted by the voices of other speakers. In applications that call for remote access by users, cochannel interference is often the cause of diminished performance. The interference may be introduced in the communication channels, at some point during the transmitting and receiving end. In tactical communication systems, where there are multiple signals transmitted over a signal channel, this problem is also common. It is also possible that interference may be introduced at the transmission site itself. This is often the case if the microphone at the transmitting end is not acoustically isolated, in which case all background noises, including voices would be transmitted along with the primary speaker. This scenario is often exemplified in speakerphones and other hands-free communication devices. No matter where the interference has occurred, the end result is a corrupted signal of multiple voices that creates major problems for automatic speaker/speech recognition systems.

In this paper, we address the problem of identifying temporal regions of a cochannel signal as being either one- or two-speaker speech. This is known as the speaker count labelling problem in that given a temporal region or frame of speech, the label corresponds to a count of either 1 or 2. Solving the speaker count labelling problem

* Corresponding author. Tel.: +1-856-256-5334; fax: +1-856-256-5241.

E-mail address: ravi@rowan.edu (R.P. Ramachandran).

0031-3203/00/$20.00
PII: S0031-3203(00)00004-2

is very important in making automatic speaker and speech recognition systems more robust. Suppose that a speaker identification system is trained on one-speaker speech only and a cochannel signal is encountered during testing. There is a mismatch between the training and testing conditions which causes serious performance degradation. If it is possible to label the regions of the cochannel signal that have count of 1, only the feature vectors from those regions can be used for speaker identification. Similarly, for speech recognition, regions having a count of more than 1 can be further processed to remove the interference. In fact, if the objective is speaker interference suppression, speaker count labelling can be used in conjunction with a knowledge of the pitch tracks of both speakers to diminish the effect of the interfering speaker.

In Ref. [1], a similar problem known as automatic talker activity labelling has been addressed. In this work, cochannel speech was used as input where frames were labelled either target (primary speaker), jammer (interfering speaker) or talker–jammer (cochannel speech). A classifier was then used to train front-end feature vectors for the 'target' speaker, the 'jammer' speaker and the combination of both speakers. During the recognition, the detector was presented with speech from the target, jammer and combination of target–jammer. The detector's task was to use the stored references to identify which of the three possible sources produced the input and report that result. The detectors were then evaluated on their ability to label the test input correctly. With the use of the mel frequency cepstral coefficient (MFCC) feature and a vector quantizer (VQ) classifier, a 80% correct detection rate was recorded.

The speaker count algorithm we propose is based on a common pattern recognition approach involving feature extraction and classification. We attempt to find features that can discriminate between one- and two-speaker speech on a frame by frame basis. Experiments are conducted with the MFCC feature and the linear predictive cepstral coefficients (LPCC). We also propose a new feature based on the concept of pitch prediction which is commonly used in speech coding [2]. For the classifier, we compare the VQ [3,5] and the neural tree network (NTN) [6,7]. Two distinctive scenarios are examined, namely, the closed-set case where the identity of the speakers is known a priori and the open-set case, where the identity of the speakers is not known. Note that in Ref. [1], the closed-set case is mostly examined. The speech is assumed to be text independent in that there is no restriction on what phonemes are uttered.

Another assumption in this work is that only one cochannel signal is available for analysis. Moreover, this signal is a linear superposition (or mixture) of the two constituent speech signals. In contrast with our assumption, much work on signal separation has dealt with the case when two signal mixtures are available. This is the area of blind signal processing for which a survey of adaptive learning algorithms are given in Ref. [8]. Other techniques which use two signal mixtures for separation include decorrelation based on the least mean-squares algorithm [9], eigendecomposition of the autocorrelation matrix [10], polyspectral analysis [11]. A further examination of the decorrelation method for accelerated adaptive filtering is given in Ref. [12] along with an application of speech recognition.

The outline of this paper is as follows. Section 2 discusses the various features we use for speaker count. In Section 3, we describe the VQ and NTN classifiers that we use along with the features to do speaker count determination as a pattern recognition task. Section 4 gives the experimental protocol and Section 5 discusses the results. In Section 6, we present the conclusions.

2. Features for speaker count determination

In this section, we discuss the various features that are considered to determine the speaker count.

2.1. Linear predictive cepstral coefficients (LPCC)

Speech consists of two major correlations, namely, the near- and distant-sample redundancies. The near-sample correlations leads to a linear prediction (LP) model for the speech as given by

$$s(n) = \sum_{k=1}^{p} a_k s(n-k) + r(n), \tag{1}$$

where $s(n)$ is the speech signal, $r(n)$ is the error or LP residual, a_k are the LP weights applied to the previous speech samples in estimating the current sample and p is the LP order. The residual signal $r(n)$ is obtained by applying a nonrecursive filter $A(z)$ to the speech as given by

$$A(z) = 1 - F(z) = 1 - \sum_{k=1}^{p} a_k z^{-k}. \tag{2}$$

The filter $A(z)$ removes the near-sample correlations in the speech and is known as the prediction error filter. From Eq. (2), we see that $F(z)$ is a formant or linear predictor that forms an estimate of $s(n)$ as a weighted linear combination of p previous samples. The coefficients a_k are computed by the autocorrelation method which minimizes the mean-square value of the residual $r(n)$ [13,14]. With the autocorrelation method, all the roots of $A(z)$ are within the unit circle [13,14]. The spectrum of $1/A(z)$ represents the spectral envelope of the speech signal which in turn specifies the vocal tract resonances or formants. The distant sample correlation is due to the inherent periodicity or pitch and is discussed later.

Based on the LP coefficients a_k, it is possible to derive a host of equivalent representations. These are the reflection coefficients, log-area ratios, linear prediction cepstral coefficients (LPCC) and the line spectral frequencies (LSF) [15]. We use the LPCC feature for speaker count which, for a minimum phase $A(z)$, can be recursively computed from the LP coefficients by the relationship

$$c_n = a_n + \sum_{k=1}^{n-1} \left(\frac{k}{n}\right) c_k a_{n-k}, \quad 1 \leqslant n \leqslant p, \tag{3}$$

where c_n are the cepstral coefficients. The LPCC feature is commonly used in speaker recognition systems [16] and it is our intention to examine it for speaker count.

2.2. Pitch prediction feature (PPF)

Once the speech signal $s(n)$ has been filtered by $A(z)$ (see Eqs. (1) and (2)), a residual signal, $r(n)$ that is free of near-sample correlations is produced. This residual signal contains only the distant sample or pitch information. The pitch prediction filter removes the pitch information. In speech coding, pitch prediction is used to parameterize the pitch information which is transmitted along with the LP parameters [2]. The simplest form of the pitch prediction filter has one tap whose transfer function is given by

$$P(z) = \beta_1 z^{-M}, \tag{4}$$

where the integral delay M represents the pitch period. Since the sampling frequency is unrelated to the pitch period, the individual samples do not show a high period-to-period distant sample correlation. Therefore, a 3 tap predictor serves like an interpolation filter and provides for interpolated estimates that show higher period to period correlation. The transfer function is

$$P(z) = \beta_1 z^{-M+1} + \beta_2 z^{-M} + \beta_3 z^{-M-1}. \tag{5}$$

In computing the predictor coefficients and M, consider the situation of a signal that is passed through the prediction error filter $1 - P(z)$ to generate the residual $e(n)$. Assuming a given value of M, the coefficients of $P(z)$ are chosen to minimize the mean-squared residual

$$E_{\text{mse}} = \sum_{n=1}^{N} e^2(n), \tag{6}$$

where

$$e(n) = r(n) - \beta_1 r(n - M + 1) - \beta_2 r(n - M) - \beta_3 r(n - M - 1) \tag{7}$$

and N is the number of samples in one frame. The minimization of E_{mse} leads to a system of equations which can be written in matrix form as $\mathbf{Ac} = \mathbf{d}$. For a 3 tap predictor, the entries of the matrix $\mathbf{A}$ are

$$A(i, j) = \phi(M + i, M + j) = \sum_{n=1}^{N} r(n - M - i)\, r(n - M - j) \tag{8}$$

for $-1 \leqslant i, j \leqslant 1$. The vector

$$\mathbf{c} = [\beta_1 \ \beta_2 \ \beta_3]^{\mathrm{T}} \tag{9}$$

and the vector

$$\mathbf{d} = [\phi(0, M-1) \quad \phi(0, M) \quad \phi(0, M+1)]^{\mathrm{T}}. \tag{10}$$

Specifically, for the one tap case, $\beta_1 = \phi(0, M)/\phi(M\ M)$. In order to determine the optimum lag value M, the mean-squared error is minimized by solving the above equations. The resulting error E_{res} is

$$E_{\text{res}} = \phi(0, 0) - \mathbf{c}^{\mathrm{T}}\mathbf{d} \tag{11}$$

in which the second term in the above equation is a function of M. The optimal value of M is that which maximized $\mathbf{c}^{\mathrm{T}}\mathbf{d}$. The procedure is to do an exhaustive search of all integral values of M within an allowable range (we used 20–147 for the 8 kHz sampling rate) to find the optimal value. Assuming that the off-diagonal terms of $\mathbf{A}$, which represent the near-sample redundancies, can be neglected, the function $\mathbf{c}^{\mathrm{T}}\mathbf{d}$ can be approximately given by [2]

$$\mathbf{c}^{\mathrm{T}}\mathbf{d} \simeq \sum_{m=M-1}^{M+1} \frac{\phi^2(0, m)}{\phi(m, m)}. \tag{12}$$

Based on these pitch prediction concepts, a new feature for speaker count has been developed. The pitch prediction feature (PPF) is defined as the standard deviation of the differences between the local peaks of the quantity $\mathbf{c}^{\mathrm{T}}\mathbf{d}$ as determined by the pitch prediction method. The local peaks are those peaks of $\mathbf{c}^{\mathrm{T}}\mathbf{d}$ that are above a given threshold. Based on our observations, peaks that are greater than 50% of the global maximum have been chosen as possible pitch peaks. If a frame of a cochannel speech signal has one speaker, strong peaks will occur at multiples of the pitch period. Therefore, the standard deviation of the differences of the peaks will be very small. In Fig. 1(a) and (b), a plot of $\mathbf{c}^{\mathrm{T}}\mathbf{d}$ is given for a frame of speech of two different speakers, one with pitch period 35 samples and the other with period 55 samples. When the speech of these two speakers are mixed as a cochannel signal, there will be a considerably larger number of strong peaks of $\mathbf{c}^{\mathrm{T}}\mathbf{d}$. This is due to the strong cross-correlation values between the pitch pulses of the two speakers. For this reason, the standard deviation of the differences of the peaks will be much higher. Fig. 1(c) shows the plot of $\mathbf{c}^{\mathrm{T}}\mathbf{d}$ for a frame of cochannel speech.

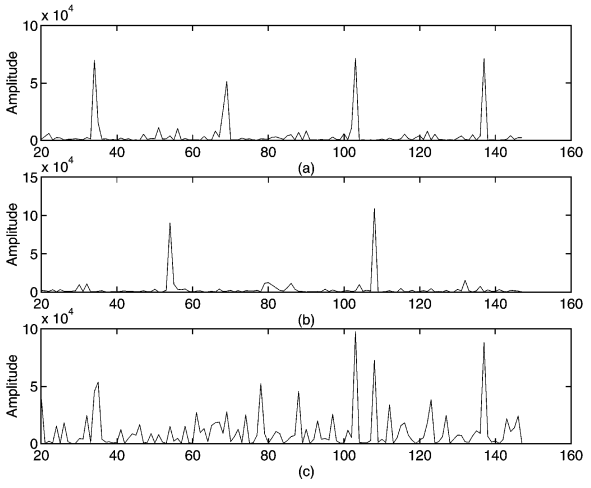

Fig. 1. Plot of $\mathbf{c}^{\mathrm{T}}\mathbf{d}$ for a frame of (a) speaker 1 with PPF value of 0.5, (b) speaker 2 with PPF value of 0.0 and (c) cochannel signal with PPF value of 14.18.

In Fig. 1(a), the peaks of $\mathbf{c}^{\mathrm{T}}\mathbf{d}$ that are above the 50% threshold correspond to 35, 70, 105 and 139 samples (multiples of the pitch period of 35 samples). The differences in the peaks are 35, 35, 35 and 34 samples. The standard deviation of these differences is 0.5 which turn in the PPF value. In Fig. 1(b), the peaks of $\mathbf{c}^{\mathrm{T}}\mathbf{d}$ are 55 and 110. The differences are 55 and 55 which in turn give rise to a PPF value of 0.0. Fig. 1(c) represents the peaks of $\mathbf{c}^{\mathrm{T}}\mathbf{d}$ for cochannel speech. The peaks are at 35, 78, 105, 110 and 139 samples. The differences are 35, 43, 27, 5 and 29. The PPF value is 14.18.

2.3. *Mel frequency cepstral coefficients* (*MFCC*)

The perception of sound by humans of either pure tones or for speech signals have been shown to follow a nonlinear scale. This has led to the definition of what is known as subjective pure tones. Thus for every pure tone defined by actual frequency measured in Hz, a subjective pitch is measured on a scale called the mel or bark scale. As a standard reference, a pitch of a 1 kHz tone, 40 dB above the hearing threshold, is defined as 1000 mels. Mathematically, it has been shown that the subjective pitch in mels increases less and less rapidly as the stimulus frequency is increased linearly [14,17].

These perceptual nonlinearities have led to modeling the peripheral auditory system by critical-band filters. The model postulates that sounds are preprocessed by a band of triangular bandpass filters, with center frequency spacings and bandwidths increasing with frequency (equivalently increasing by a constant mel frequency interval) [14]. In fact, these filters are designed similar in spacing as the auditory neurons located on the basilar membrane in the inner ear. The modified spectrum of the speech signal $S(\omega)$ thus consists of the output power of these filters when $S(\omega)$ is the input. If the power coefficients are denoted by $\bar{S}_k$, $k = 1, 2, \ldots, K$, we can calculated what is called the mel-frequency cepstral coefficients (MFCC) [18] denoted by $\bar{c}_n$, which can be expressed as

$$\bar{c}_n = \sum_{k=1}^{K} \log(\bar{S}_k) \cos\left[n\left(k - \frac{1}{2}\right)\frac{\pi}{K}\right], \quad n = 1, 2, \ldots, L, \tag{13}$$

where L is the number of cepstral coefficients. The MFCC has been used for closed set speaker count determination [1] and we compare it to the LPCC and PPF in both the closed and open set situations.

3. Classifiers

In speaker recognition systems, vector quantizers (VQ) and neural tree network (NTN) classifiers have been used successfully to render decisions about the identity of a speaker [5,7] among a group of M speakers. Each speaker is represented by a VQ codebook or NTN model that is configured during training. During testing, the feature vectors are obtained from one utterance consisting of many frames. These feature vectors are applied to each of the VQ codebook or NTN models (depending on which classifier is used) to get M distinct scores. The model with the best score identifies the speaker. The speaker count determination problem is slightly different in that a model is needed to represent a speaker count of 1 and 2. Also, in contrast to speaker recognition, a decision is taken for each individual frame rather than for an entire utterance. The speaker count is determined for each frame and hence, the decision is taken using only one feature vector. In speaker recognition, an entire utterance is processed and hence, the decision is taken using an ensemble of feature vectors. The general scheme for speaker count is shown in Fig. 2.

3.1. *Vector quantizer*

In this paper, two scenarios are investigated for the speaker count determination problem. The first looks at the closed-set case, where the speakers are known a priori. In this instance, three codebooks are developed

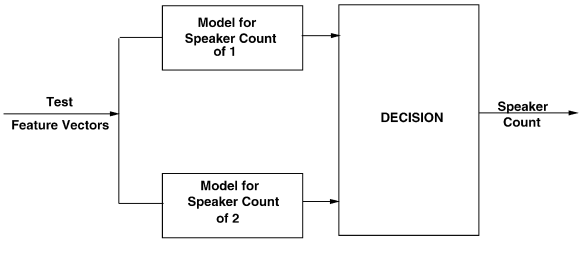

Fig. 2. Speaker count determination system.

from the training feature vectors, each dedicated to one of the three types of possible speech conditions encountered. The three conditions are (1) one-speaker speech from the first speaker, (2) one-speaker speech from the second speaker and (3) two-speaker or cochannel speech from both speakers. The codebook for each condition is designed by the Linde–Buzo–Gray (LBG) algorithm [4] from training data for that particular condition only. This is known as unsupervised learning in that training data pertaining to another condition does not influence the codebook for a particular condition. Each of the codebooks will have the same size or number of codevectors. We evaluate the performance for various codebook sizes. During testing, consider a test feature vector from a particular frame. It is quantized by each of the three codebooks. The quantized vector is that which is closest according to some distance measure to the test feature vector. We use the squared Euclidean of L_2 distance in our work. Hence, three different distances are recorded, one for each codebook. The codebook which renders the smallest distance identifies the speech condition. If condition (1) or (2) results, we have a speaker count of 1. If condition (3) results, the speaker count is 2. The performance is the number of frames identified correctly divided by the total number of frames tested. The next section gives details on how the training data, test data and correct speaker count are obtained.

In the open set case, the speakers are not known a priori. Two codebooks are designed by the LBG method, one for one-speaker speech and the other for two-speaker or cochannel speech. The testing is done as in the closed-set case but only two distances are recorded. The codebook which renders the smaller distance identifies the speaker count.

3.2. Neural tree network

The NTN classifier is a hierarchical classifier that combines the properties of decision trees and feedforward neural networks [7]. The NTN uses a tree architecture to implement a sequential linear decision strategy [19]. The architecture of the NTN is determined during training. Thus, it is self-organizing. Also, NTN training is supervised in that training data pertaining to different conditions (each having a distinct label) is used. Therefore, each training feature vector has a label indicating the condition it emanates from. Each node at every level of the NTN divides the input training vectors into a number of exclusive subsets of the training data. If a set of training data at a particular node is of the same class or condition (has the same label), then that node becomes a leaf. Otherwise, the data is split into several subsets, which became children of this node. This procedure is repeated until all the data is completely uniform at the leaf nodes. The leaf nodes of the NTN partition the feature space into homogeneous subsets, meaning

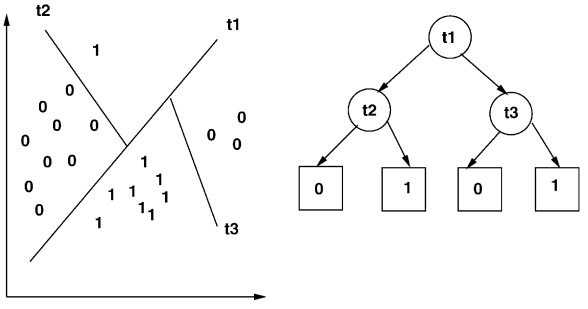

Fig. 3. Concept of neural tree network. The circles represent nodes and the squares represent leaves.

a single class at each leaf node. An illustration of this concept is given in Fig. 3. In Fig. 3, training data come from two classes labelled as 0 and 1. The circles represent nodes and the squares represent leaves. The nodes can be thought of as being hyperplanes that partition the space into exclusive subspaces. These subspaces are further partitioned until a leaf is reached.

The NTN will give a 100% performance on the training set. Since test data are always different from the training data, an optimal performance is not necessarily reached for a fully grown NTN due to overtraining [7]. Therefore, we use the strategy of forward pruning (has been used for speaker identification) [7] to avoid overtraining. When implementing forward pruning, the NTN is grown only to a specified number of levels and the nodes at the lowest level are said to be leaves. In this case, the training data for a leaf are not necessarily from the same class. A majority vote is taken and the leaf is assigned the label of the majority. We study the speaker count performance with varying number of levels.

For the speaker count determination closed set problem, an NTN is grown from training data consisting of three labels. The three labels are for the three conditions which as before are (1) one-speaker speech from the first speaker, (2) one-speaker speech from the second speaker and (3) two-speaker or cochannel speech from both speakers. Given a frame of test speech, the feature vector is found and passed through the NTN so that is reaches a particular leaf. The label assigned to the leaf classifies the speech frame. For the open-set case, an NTN is grown from training data consisting of two labels, namely, one- and two-speaker speech. The classification of a test feature vector is similarly done in that a match is made to the label of the leaf reached.

4. Experimental protocol

In the following section, the experimental protocols for the closed- and open-set schemes are discussed. In general, the experimental protocols are quite similar.

However, there are some slight variations. A key point to note, is that, in the closed-set experiment, particular attention is paid to the speakers' identity. In the open-set case, however, the identity of the speakers is not relevant. We will first describe the feature computation and then look into the training and testing phases. The New England portion of the TIMIT database is used in which the speech is downsampled to 8 kHz.

4.1. Feature computation

The computation of the features applies to both the training and testing phases. For the LPCC feature, the autocorrelation method of LP analysis is used to get the LP coefficients a_k [13]. The speech signal is first preemphasized by passing it through a nonrecursive filter $1–0.95z^{-1}$. A 30 ms long Hamming window is then applied with a 20 ms overlap thereby providing a frame size of 10 ms. A 12th-order LP analysis is done and the LP coefficients a_k are converted into a 12th-order LPCC feature vector using the recursion in Eq. (3). A 12 dimensional MFCC feature is also calculated. As for the LPCC feature, preemphasis followed by a 30 ms Hamming window with a 20 ms overlap is applied.

Consider the PPF feature. In this case, no preemphasis is applied. A 12th-order LP analysis by the autocorrelation method using a 30 ms Hamming window with a 20 ms overlap gives us the LP coefficients a_k. Using the a_k, the speech is filtered to generate the LP residual as in Eq. (1). Prior to extracting the PPF, the LP residual is passed through a Gaussian-shaped filter whose impulse response $f(x/\sigma)$ whose impulse response can be expressed as

$$f\left(\frac{x}{\sigma}\right) = \frac{1}{\sigma\sqrt{\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right).$$

The impulse response has a Gaussian shape and σ refers to the standard deviation of the Gaussian function. This type of filter has been used in image processing, particularly for edge detection [20,21]. The utilization of this filter is motivated by our observation that it acts so as to smooth the LP residual thereby enhancing the performance of the peak picking algorithm (to pick the peaks of $\mathbf{c}^T\mathbf{d}$ as described earlier) used when generating the PPF. Different values of σ were tried and the best performance was achieved with a $\sigma = 0.32$. The number of the filter taps is equal to an odd integer closest to $16\,(8\sigma + 1)$. From the Gaussian filtered LP residual, the pitch prediction algorithm was applied to get a 3 tap pitch filter $P(z)$ and the quantity $\mathbf{c}^T\mathbf{d}$ for $M = 20$ to 147. For the pitch prediction algorithm, a framesize of 10 ms was used (performance given later) and there was no overlap between frames. After finding the global maximum of $\mathbf{c}^T\mathbf{d}$, a threshold equal to 50% of this maximum was set. Again, different thresholds were tried before a decision was taken. The local peaks of $\mathbf{c}^T\mathbf{d}$ are those above the threshold from which the candidate values of M are taken. From these candidate values of M, the PPF scalar feature is found by taking the standard deviation of the differences as described earlier.

4.2. Training phase

During the training phase for all the experiments, the general aim is to derive features that represent one- and two-speaker or cochannel speech. The first five sentences for each of the 38 speakers in the New England portion of the TIMIT database represent the training speech. Six speakers (three male and three female) are selected and all possible sentence combinations are used to derive the training cochannel speech. With this exhaustive combining method, a total of 375 cochannel sentences are used for training. In generating the cochannel sentences, the individual sentences are first normalized by their maximum absolute sample value before being added. In explaining both the closed- and open-set cases, let the individual speech signals pertaining to speaker A and speaker B be $s_A(n)$ and $s_B(n)$, respectively. The cochannel signal is denoted as $s_{AB}(n) = s_A(n) + s_B(n)$.

Consider the closed-set scenario. The signal $s_A(n)$ is divided into frames and energy thresholding is used to distinguish between speech frames (not silent) and silent frames. The same procedure is repeated for $s_B(n)$ to get the speech frames of speaker B. For the cochannel signal $s_{AB}(n)$, the cochannel frames are those for which a speech frame of speaker A and a speech frame of speaker B are added. The frames of $s_{AB}(n)$ for which a speech frame of speaker A and a silent frame of speaker B are added correspond to a speech frame of speaker A only. Similarly, when a speech frame of speaker B and a silent frame of speaker A are added, we get a speech frame speaker B only. We now gather speech frames of speaker A from $s_A(n)$ and $s_{AB}(n)$, speech frames of speaker B from $s_B(n)$ and $s_{AB}(n)$ and cochannel frames of both speakers from $s_{AB}(n)$. The features are computed for these three cases and the VQ and NTN classifiers designed.

Consider the open-set scenario. As in the closed-set case, energy thresholding in performed on $s_A(n)$ and $s_B(n)$ to get the speech frames. These speech frames are one-speaker frames. For the cochannel signal $s_{AB}(n)$, the two-speaker or cochannel frames are those for which a speech frame of speaker A and a speech frame of speaker B are added. From $s_{AB}(n)$, we also extract one-speaker frames when a speech frame of one of the speakers is added with a silent frame of the other speaker. The features are computed for the one- and two-speaker cases and the VQ and NTN classifiers designed.

4.3. Testing phase

In the testing phase, energy thresholding is performed on the cochannel signal $s_{AB}(n)$. For the frames which are

declared to be speech frames, the feature is computed and classified by either the VQ or NTN classifier as described above. For the closed-set case, the decision is one-speaker speech from speaker A, one-speaker speech from speaker B or two-speaker speech. For the open-set case, the decision is either one- or two-speaker speech. To do the testing, six speakers from the TIMIT database that are different from those used for training are chosen. Three of the speakers are male and three are female. There are five testing sentences for each speaker that are different from the sentences used for training. All possible sentence combinations are used to derive the 375 cochannel speech sentences used for testing.

In measuring the performance, the decision must be compared to some notion of a correct answer which is not as obvious as in the case of assessing speaker identification systems. We formulate one simple method to get a correct answer as follows. Given a cochannel signal $s_{AB}(n)$, energy thresholding is performed on the constituent signals $s_A(n)$ and $s_B(n)$ as is done during training. The rest of the training procedure is essentially repeated to label the cochannel frames. The performance is the number of times a frame is classified correctly divided by the total number of frames tested (82, 174 in our experiments). There are two sources of error given a particular cochannel frame. The first is when a decision is taken but does not correspond to the correct frame label. The second is when a decision is taken (since the cochannel frame is declared to be a speech frame) but there is no frame label (since neither the corresponding frame of $s_A(n)$ and $s_B(n)$ are declared to be a speech frame). This second source of error is very rare and occurs less than 0.1% of the time.

5. Results and discussion

In this section, the performance of the new PPF and cepstral features are compared. In the first set of results, the performance of the VQ and NTN classifiers are compared in the closed-set case. In the second set of results, the experiments are repeated for the open-set case. The vector quantizer codebook sizes range from 16 to 256 for the LPCC and MFCC features. Lower codebook sizes of 1–16 were used in the case of the PPF. This is due to the fact that the PPF is a scalar feature as compared to the 12-dimensional LPCC and MFCC features. Therefore, the use of the higher codebook sizes for a scalar feature is not necessary and actually diminishes the performance. For the NTN classifiers tree levels of 2–10 were grown for all the features.

5.1. Closed-set case

Tables 1–3 depict all the closed-set results. The VQ classifier outperforms the NTN for all the features. We concentrate on the results obtained using VQ. The LPCC features outperforms the MFCC for all the VQ codebook sizes. The performance of the PPF is essentially equal to that of the LPCC. The PPF still maintains an advantage in that the feature dimension is substantially lower. Moreover, the smallest codebook size of 1 can be used as negligible performance gain is achieved by a larger codebook size.

Table 1
Closed-set results for the cepstral features using the VQ classifier

Codebook size	Cepstral feature	
	LPCC	MFCC
16	83.1	75.6
32	83.2	75.9
64	83.1	76.1
128	83.1	76.2
256	83.1	75.5

Table 2
Closed-set results for the PPF feature using VQ classifier

Codebook size	PPF feature
1	83.2
2	83.3
4	83.3
8	83.3
16	83.5

Table 3
Closed-set results for the cepstral and PPF features using the NTN classifier

Number of levels	Feature		
	LPCC	MFCC	PPF
2	65.9	63.9	65.2
4	64.8	65.6	66.6
6	64.6	65.2	66.6
8	64.7	64.8	66.9
10	64.5	64.6	65.6

5.2. Open-set case

Tables 4–6 depict all the open-set results. Since the open-set case is a harder problem than the closed-set case, the performance is less for the open-set case. The VQ classifier again essentially outperforms the NTN.

We concentrate on the results obtained using VQ. The LPCC an MFCC show a similar performance. The PPF shows the best performance for the smallest codebook size of 1 and outperforms the cepstral features.

Table 4
Open-set results for the cepstral coefficients using the VQ classifier

Codebook size	Cepstral feature	
	LPCC	MFCC
16	56.8	58.8
32	56.7	58.9
64	56.7	57.7
128	56.7	57.5
256	56.8	57.3

Table 5
Open-set results for the PPF feature using the VQ classifier

Codbook size	PPF feature
1	64.4
2	63.0
4	50.2
8	55.0
16	61.1

Table 6
Open-set results for the cepstral and PPF features using the NTN classifier

Number of levels	Feature		
	LPCC	MFCC	PPF
2	55.8	49.0	50.5
4	54.4	60.2	48.8
6	52.7	55.8	47.0
8	51.6	47.5	51.2
10	48.4	50.7	51.2

6. Summary and conclusions

We examine the problem of identifying temporal regions or frames of cochannel speech as being either one- or two-speaker speech. Ideally, separation of the individual speech signals that form a cochannel signal is desired. However, it is known that when two equal bandwidth signals are added, such a separation is not possible. Identifying frames as being one- or two-speaker speech is done using a pattern recognition framework based on feature extraction and subsequent classification. We develop a new feature called the pitch prediction feature (PPF) based on the concept of pitch prediction that is used in speech coding. The PPF is a scalar feature that outperforms the linear predictive cepstrum (LPCC) and the mel-wrapped cepstrum (MPCC) both of which are 12-dimensional vector features. The vector quantizer (VQ) and neural tree network (NTN) classifiers are compared and the VQ is found to be consistently better. Note that the superiority of the PPF is not only synergistic with achieving a lower feature dimension but also with being able to use a lower VQ codebook size. In fact, the lowest codebook size of 1 is used for the PPF which essentially is equivalent to a Bayesian discriminant approach [19]. Two cochannel scenarios are looked at, namely, the case when the speaker identities are known a priori (closed set) and when the identities are not known (open set). The open-set problem is more difficult and as expected, the performance for all the feature is less. For both the open-set and closed-set problems, the PPF is the best feature.

References

[1] M.A. Zissman, C.J. Weinstein, L.D. Braida, Automatic talker activity labelling for co-channel talker interference suppression, IEEE International Conference on Acoustics, Speech and Signal Processing, Albuquerque, New Mexico, April 1990, pp. 813–816.
[2] R.P. Ramachandran, P. Kabal, Pitch prediction filters in speech coding, IEEE Trans. Acoust. Speech Signal Process 37 (1989) 467–478.
[3] J. Makhoul, S. Roucos, H. Gish, Vector quantization in speech coding, IEEE Proc. 73 (1985) 1551–1588.
[4] Y. Linde, A. Buzo, R.M. Gray, An algorithm for vector quantizer design, IEEE Trans. Commun. COM-28 (1980) 84–95.
[5] A.E. Rosenberg, F.K. Soong, Evaluation of a vector quantization talker recognition system in text independent and text dependent modes, Comput. Speech Language 22 (1987) 143–157.
[6] A. Sankar, R.J. Mammone, Growing and pruning neural tree networks, IEEE Trans. Comput. C-42 (1993) 221–229.
[7] K.R. Farrell, R.J. Mammone, K.T. Assaleh, Speaker recognition using neural tree networks and conventional classifiers, IEEE Trans. Speech Audio Proc. 2 (1994) 194–205.
[8] S-C. Amari, A. Cichocki, Adaptive blind signal processing-neural network approaches, Proc. IEEE 86 (1998) 2026–2048.
[9] E. Weinstein, M. Feder, A.V. Oppenheim, Multichannel blind signal separation by decorrelation, IEEE Trans. Speech Audio Proc. 1 (1993) 405–413.
[10] Y. Cao, S. Sridharan, M. Moody, Multichannel speech separation by eigendecomposition and its application to co-talker interference removal, IEEE Trans. Speech Audio Proc. 5 (1997) 209–219.
[11] S. Shamsunder, G.B. Giannakis, Multichannel blind signal separation and reconstruction, IEEE Trans. Speech Audio Proc. 5 (1997) 515–528.
[12] K.-C. Yen, Y. Zhao, Adaptive cochannel speech separation and recognition, IEEE Trans. Speech Audio Proc. 5 (1999) 138–151.

[13] L.R. Rabiner, R.W. Schafer, Digital Processing of Speech Signals, Prentice-Hall, Englewood Cliffs, NJ, 1978.
[14] L.R. Rabiner, B.H. Juang, Fundamentals of Speech Recognition, Prentice-Hall, Englewood Cliffs, NJ, 1993.
[15] F.K. Soong, B.H. Juang, Line spectrum pair (LSP) and speech data compression, IEEE International Conference on Acoustics, Speech and Signal Processing, San Diego, CA, March 1984, pp. 1.10.1–1.10.4.
[16] B.S. Atal, Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification, J. Acoust. Soc. Am. 55 (1974) 1304–1312.
[17] S.S. Stevens, Critical bandwidth in loudness summation, J. Acoust. Soc. Am. 29 (1957) 548–557.
[18] S.B. Davis, P. Mermelstein, Comparison of parametric representations of monosyllabic word recognition in continuously spoken sentences, IEEE Trans. Acoust. Speech Signal Proc. ASSP- 28 (1980) 357–366.
[19] R.O. Duda, P.E. Hart, Pattern Classification and Scene Analysis, Wiley, New York, 1973.
[20] R.J. Schalkoff, Digital Image Processing and Computer Vision, Wiley, New York, 1989.
[21] G. Deng, L.W. Cahill, An adaptive Gaussian filter for noise reduction and edge detection, IEEE Nuclear Science Symposium and Medical Imaging Conference 1994, pp. 1615–1619.

About the Author—MICHAEL A. LEWIS received his B.E, M.E and Ph.D degrees in Electrical Engineering from the City University of New York in 1991, 1993 and 1998 respectively. His research interests include speech processing, adaptive signal processing and modeling and neural networks.

About the Author—RAVI P. RAMACHANDRAN was born in Bangalore, India on July 12th, 1963. He received the B.Eng. degree (with great distinction) from Concordia University, Montreal, P.Q., Canada in 1984 and the M.Eng. and Ph.D. degrees from McGill University, Montreal, P.Q., Canada in 1986 and 1990, respectively. From January to June 1988, he was a Visiting Postgraduate Researcher at the University of California, Santa Barbara. From October 1990 to December 1992, he worked in the Speech Research Department at AT&T Bell Laboratories, Murray Hill, NJ. From January 1993 to August 1997, he was a Research Assistant Professor at the Caip Center, Department of Electrical Engineering, Rutgers University, Piscataway, NJ. Also, from July 1996 to August 1997, he was a Senior Speech Scientist at T-NETIX Inc., Piscataway, NJ. Since september 1997, he is an Associate Professor in the Department of Electrical Engineering, Rowan University, Glassboro, NJ. His main research interests are in speech processing, data communications and digitial signal processing.