Chapter 3

Speaker Segmentation by Fusing Features

In the previous chapters, we have discussed some issues, and reviewed the existing methods of speaker segmentation. In this chapter, a method is proposed for speaker segmentation by fusing the vocal tract features with the excitation source features. The feature extraction methods are described in Section 3.2. Support vector machine (SVM) for speaker segmentation is given in Section 3.3. Section 3.4 presents the experimental results.

3.1 Introduction

Speech is a composite signal that mainly carries information about the message to be conveyed, speaker characteristics and the language. Speaker characteristics in the speech signal can be attributed to the dimensions of the vocal tract system, characteristics of excitation, and the learning habits of the speakers. The speaker specific vocal tract information is mainly represented by spectral features like mel frequency cepstral coefficients (MFCC) and linear prediction cepstral coefficients (LPCC). Efforts are being made to exploit the usefulness of other features for speaker segmentation. The goal of the search for new features is to improve the performance of the existing speaker segmentation systems, which are based mainly on the characteristics of the vocal tract system. One of the desirable properties of the new feature is that they should provide speaker-specific information complementary to the spectrum-based features like
MFCC. Then by combining the evidence from the existing features, the performance of the speaker segmentation system can be improved.

In [128], it is demonstrated that the residual phase signal contains speaker-specific information that is complementary to the MFCC features. The residual phase is defined as the cosine of the phase function of the analytic signal derived from the linear prediction (LP) residual of a speech signal. In this work, the speaker-specific information from the residual phase is captured using SVM. The speaker specific information from the MFCC features is also captured using SVM. The evidence from both the models is used to validate the speaker change points.

### 3.2 Feature Extraction for Speaker Segmentation

#### 3.2.1 Mel Frequency Cepstral Coefficients

Mel frequency cepstral coefficients have proved to be one of the most successful feature representations in speech related recognition tasks. The mel-cepstrum exploits auditory principles, as well as the decorrelating property of the cepstrum [129]. The procedure of MFCC computation is shown in Fig. 3.1 and described as follows:

![Fig. 3.1: Extraction of MFCC from speech signal.](image)
• **Preemphasis:** The digitized speech signal \( S = s(n) \), \( n = 1, \ldots, N_t \) (where \( N_t \) is the total number of samples in the speech stream) is put through a low order digital system to spectrally flatten the signal and to make it less susceptible to finite precision effects later in the signal processing. The output of the preemphasis network, \( \hat{s}(n) \) is related to the input \( s(n) \) by the difference equation

\[
\hat{s}(n) = s(n) - \alpha s(n - 1)
\]  

(3.1)

where \( \alpha \) is the scaling factor which varies from 0 to 1. The most common value for \( \alpha \) is around 0.95.

• **Frame blocking:** Speech analysis usually assumes that the signal properties change relatively slowly with time. This allows examination of a short time window of speech to extract parameters presumed to remain fixed for the duration of the window. Thus, to model dynamic parameters, we must divide the signal into successive windows or analysis frames, so that the parameters can be calculated often enough to follow the relevant changes. In this step the preemphasized speech signal, \( \hat{s}(n) \) is blocked into frames of \( N \) samples, with adjacent frames being separated by \( M \) samples. If we denote the \( l^{th} \) frame speech by \( \hat{s}_l(n) \), and there are \( L \) frames within the entire speech signal, then

\[
\hat{s}_l(n) = \hat{s}(Ml + n), n = 0, 1, \ldots, N - 1, l = 0, 1, \ldots, L - 1
\]  

(3.2)

We used a frame rate of 125 frames/sec, where each frame was 16 ms in duration with an overlap of 50 percent between adjacent frames.

• **Windowing:** The next step in the processing is to window each individual frame so as to minimize the signal discontinuities at the beginning and end of the frame. The window must be selected to taper the signal to zero at the beginning and
end of each frame. If we define the window as $w(n)$, $0 \leq n \leq N-1$, then the result of windowing the signal is

$$\tilde{s}_l(n) = s_l(n)w(n), 0 \leq n \leq N - 1$$  \hfill (3.3)

The Hamming window is used for our work, which has the form

$$w(n) = 0.54 - 0.46 \cos \left( \frac{2\pi n}{N - 1} \right), 0 \leq n \leq N - 1$$  \hfill (3.4)

- **Computing spectral coefficients:** The spectral coefficients of the windowed frames are computed using fast Fourier transform (FFT), as follows:

$$X(k) = \sum_{n=0}^{N-1} \tilde{s}_l(n) \exp^{-jk(\frac{2\pi}{N})n}, 0 \leq k \leq N - 1$$  \hfill (3.5)

- **Computing mel spectral coefficients:** The spectral coefficients of each frame is then weighted by a series of filter frequency responses whose center frequencies and bandwidths roughly match those of the auditory critical band filters. These filters follow the mel scale whereby band edges and center frequencies of the filters are linear for low frequency and logarithmically increase with increasing frequency. These filters are called as mel-scale filters and collectively a mel-scale filter bank. As shown in Fig. 3.2, the filters used are triangular and they are
equally spaced along the mel-scale which is defined by

$$\text{Mel}(f) = 2595 \log_{10}(1 + \frac{f}{700})$$  \hspace{1cm} (3.6)

where \( f \) is the linear scale frequency and \( \text{Mel}(f) \) is the corresponding mel scale frequency. Each short term Fourier transform (STFT) magnitude coefficient is multiplied by the corresponding filter gain and the results are accumulated.

- **Computing mel frequency cepstral coefficients**: The discrete cosine transform (DCT) is applied to the log of the mel spectral coefficients to obtain the mel frequency cepstral coefficients as follows:

$$x(m) = \sqrt{\frac{2}{M_f}} \sum_{i=0}^{M_f-1} E(i) \cos\left(\frac{(2i + 1)m\pi}{2M_f}\right), \ m = 1, \ldots, M_f$$  \hspace{1cm} (3.7)

where \( M_f = \) number of filters in the filter bank. Finally cepstral mean subtraction is performed to reduce the channel effects.

### 3.2.2 Extraction of Residual Phase (RP)

The basic idea behind the linear predictive analysis [130] is that a given speech sample at \( n \), \( s(n) \) can be estimated as a linear combination of the past \( p \) speech samples. The predicted samples \( \hat{s}(n) \) is given by

$$\hat{s}(n) = \sum_{k=1}^{p} a_k s(n - k)$$  \hspace{1cm} (3.8)

where \( p \) is the order of prediction and the coefficients \( \{a_k\}; \ k=1,2,\ldots,p \) is the set of linear prediction coefficients (LPCs). The prediction error \( e(n) \) is defined as the difference between the actual value \( s(n) \) and the predicted value \( \hat{s}(n) \) and is given by

$$e(n) = s(n) - \hat{s}(n) = s(n) - \sum_{k=1}^{p} a_k s(n - k)$$  \hspace{1cm} (3.9)
The LPCs are obtained by minimizing the mean squared prediction error over the analysis frame. This error \( e(n) \) is called the linear prediction residual \( r(n) \) of the speech signal. The LP residual contains much information about the excitation source. The phase of the analytic signal derived from the LP residual contains better speaker specific information \([131]\). The analytic signal \( r_a(n) \) corresponding to \( r(n) \) \([132]\) is given by

\[
r_a(n) = r(n) + jr_h(n)
\]  

where \( r_h(n) \) is the Hilbert transform of \( r(n) \) and is given by

\[
r_h(n) = \text{IFT} [R_h(\omega)]
\]

where

\[
R_h(\omega) = \begin{cases} 
  -jR(\omega), & 0 \leq \omega < \pi \\
  jR(\omega), & -\pi \leq \omega < 0 
\end{cases}
\]  

Here \( R(\omega) \) is the Fourier transform of \( r(n) \) and IFT denotes the inverse Fourier transform. The magnitude of the analytic signal \( r_a(n) \) is given by

\[
h_e(n) = |r_a(n)| = \sqrt{r^2(n) + r^2_h(n)}
\]  

and the cosine of the phase of the analytic signal \( r_a(n) \) is given by

\[
\cos(\theta(n)) = \frac{\text{Re}(r_a(n))}{|r_a(n)|} = \frac{r(n)}{h_e(n)}
\]

### 3.3 Support Vector Machine (SVM) for Speaker Segmentation

The support vector machine \([34]\) is a useful statistic machine learning technique that has been successfully applied in the pattern recognition tasks \([133]\), \([134]\), \([135]\). If the data are linearly non-separable but non-linearly separable, the non-linear support
vector classifier will be applied. The basic idea is to transform input vectors into a high-dimensional feature space using a non-linear transformation $\Phi$, and then to do a linear separation in feature space as shown in Fig. 3.3. A non-linear support vector

![Fig. 3.3: Principle of support vector machines.](image)

classifier implementing the optimal separating hyperplane in the feature space with a kernel function $K(x, x_i)$ is given by

$$f(x) = \text{sgn} \left( \sum_{i=1}^{N_{SV}} \alpha_i y_i K(x, x_i) + b \right)$$

(3.15)

where $x \in \mathbb{R}^{N_f}$ is the $N_f$ dimensional input vector, $\{x_i\}_{i=1}^{N_{SV}}$ are the support vectors that are obtained from the training set by an optimization process, $N_{SV}$ are the total number of support vectors and $y_i \in \{-1, 1\}$ are their associated class labels. $\alpha_i$ and $b$ are the model parameters. $K(., .)$ is the kernel function defining the inner product between $x$ and $x_i$ and is given by

$$K(x, x_i) = \Phi^T(x) \Phi(x_i) = \langle x, x_i \rangle$$

(3.16)

The SVM has two layers. During the learning process, the first layer selects the basis $K(x, x_i)$, from the given set of bases defined by the kernel; the second layer constructs a linear function in this space. This is completely equivalent to constructing the optimal hyperplane in the corresponding feature space.
The SVM algorithm can construct a variety of learning machines by use of different
kernel functions. Four kinds of kernel functions are usually used. They are

1. Linear kernel

\[ K(x, x_i) = \langle x, x_i \rangle \] (3.17)

2. Polynomial kernel of degree \(d\)

\[ K(x, x_i) = (\gamma \langle x, x_i \rangle + c_0)^d \] (3.18)

3. Gaussian radial basis function (RBF)

\[ K(x, x_i) = \exp \left( -\gamma \| x - x_i \|^2 \right) \] (3.19)

4. Sigmoidal kernel

\[ K(x, x_i) = \tanh \left( \gamma \langle x, x_i \rangle + C_0 \right) \] (3.20)

where kernel parameters

- \(\gamma\): width of RBF coefficient in polynomial
- \(d\): degree of polynomial
- \(C_0\): additive constant in polynomial

In [136], an SVM classification based supervised technique was proposed for speaker
change detection in which they adopted a bottom up binary tree combining three two
class SVM classifiers for content based audio segmentation. SVM based supervised
technique was proposed in [137] by labeling the speech data around the speaker change
as (+1) class and the between speaker change as (-1) class for training an SVM hyperplane
and then classified each window as (+1) or (-1). In [40], an unsupervised
algorithm using SVM was proposed for speaker change detection.
The algorithm used for speaker change detection is described below:

- The entire speech stream $S$ is divided into $L$ number of analysis frames such that $S = \{s_l : l = 1, \ldots, L\}$ and the feature vector is obtained for each frame.
- A window with $L_f$ frames has been selected and a frame $k$ within the window is selected in such a way to separate the window into two portions.
- It is assumed that the speaker change is at this $k^{th}$ frame. Then every frame in the window that is located left of the $k^{th}$ frame is set to the (-1) class and located right of the $k^{th}$ frame is set to the (+1) class.
- Now the window is divided into two different set of frames namely $w^-$ and $w^+$ where $w^- = \{s_l : l = 1, \ldots, k - 1\} \in (-1)$ class and $w^+ = \{s_l : l = k + 1, \ldots, L_f\} \in (+1)$ class.
- The SVM is trained using these two classes and the hyperplane is obtained between these two classes.
- By using this hyperplane, the frames of these two classes are classified. If the window of speech stream is from a single speaker, the hyperplane could not classify the data into two distinct classes. So the rate of misclassification will be very high. On the other hand, if the window of speech stream is for two different speakers, the hyperplane is capable of classifying the data into two distinct classes. Hence, the rate of misclassification will be very low.
- From this we know that the SVM training misclassification rate can be used to decide whether the true speaker change occurs at the $k^{th}$ frame.
- Two types of misclassification rates are computed namely $mcr^- (k)$ and $mcr^+ (k)$ where $mcr^- (k)$ is the rate of the (-1) class misclassified as (+1) and $mcr^+ (k)$ is the rate of the (+1) class misclassified as (-1).
• If the misclassification rates are smaller than the threshold $t_{mc}$, we can conclude that true speaker change occurs at $k^{th}$ frame, otherwise it is concluded that there is no speaker change at $k^{th}$ frame.

• Then the window is shifted one frame to the right of current position and the procedure is repeated. Likewise the entire speech stream must be examined.

The proposed method is unsupervised, because it can work without any knowledge of the identity of speakers and there is no need for training speaker models beforehand.

3.4 Experimental Results

3.4.1 Speech Data

The experiments are conducted using the television broadcast interviews and National Institute of Standards and Technology (NIST) 2004 speaker recognition evaluation database. A total dataset of 30 conversations is used in our studies. This includes 10 conversations for each male-male, male-female and female-female speaker conversations. The speech is sampled at 8 kHz and encoded by 16-bits. The speaker change points are manually marked. The manual segmentation results are used as the reference for evaluation of the proposed speaker segmentation method. A total of 1,021 speaker segments are marked in the 30 conversations. Excluding the silence and non speech periods, the segment duration is mostly between 1 and 4 seconds.

3.4.2 Feature Representation

The extraction of MFCC features is based on first pre-emphasizing the input speech data using a first order digital filter and then segmenting it into 16 ms frames with an overlap of 50 percent between adjacent frames using Hamming window. For each frame, the first 13 cepstral coefficients other than the zeroth value are used.
The residual phase is obtained from the speech signal using 16\textsuperscript{th} order linear prediction analysis. The extraction of LP residual is based on first pre-emphasizing the input speech data using a first-order digital filter and then segmenting it into 16 ms frames with an overlap of 50 percent between adjacent frames using Hamming window. From the LP residual, the residual phase feature is computed for the entire speech signal using the method described in Section 3.2.2. For each frame 19 samples around the highest Hilbert envelope are extracted. Fig. 3.4 illustrates the computation of residual phase for a segment of speech signal.
3.4.3 Speech Stream Scanning and Testing Procedure

The tests in this experimental investigation are conducted using the procedure mentioned in Section 3.3. The procedure is applied for MFCC features and residual phase separately in order to locate the speaker change frames. The 125 frames window size is used in our experiments. For SVM classifier, the linear kernel function with the upper bound of the Lagrange multiplier $\alpha_i=1$ has been used. The misclassification threshold $t_{mc}$ of 0.075 has been used for reliable speaker change detection.

The speaker change detection algorithm begins from the window of the first 125 frames, with the assumption that there is a speaker change at the center of the window. Then the data samples on either side of the center of this window are used to train the SVM classifier, and the hyperplane is obtained. Then the same samples are classified using this hyperplane. The misclassification rates are computed and compared with the threshold. The final decision has been taken, that is whether the true speaker change is at the center of the window or not. Then the window is shifted one frame to the right and the algorithm is repeated. The entire procedure is continued until the rightward window has reached the 125 frames before the end of the speech stream.

3.4.4 Assessment Measures

The performance of speaker segmentation is assessed in terms of two types of error related to speaker change detections namely false alarm and missed detections. A false alarm (FA) of speaker change detection occurs when a detected speaker change is not a true one. A missed detection (MD) occurs when a true speaker change cannot be detected. The false alarm rate (FAR) and missed detection rate (MDR) are defined as

$$ FAR = \frac{\text{Number of false alarmed speaker changes}}{\text{Number of detected speaker changes}} $$

(3.21)
Some authors use two other measures namely precision (PRC) and recall (RCL), which are closely related to FAR and MDR. They are defined as

\[
PRC = \frac{\text{Number of correctly found speaker changes}}{\text{Total number of changes found}}
\]  
(3.23)

\[
RCL = \frac{\text{Number of correctly found speaker changes}}{\text{Number of actual speaker changes}}
\]  
(3.24)

In order to compare the performance of different systems, the F-measure is often used and is defined as

\[
F = \frac{2 \times PRC \times RCL}{PRC + RCL}
\]  
(3.25)

The F-measure varies from 0 to 1, with a higher F-measure indicating better performance. To compute these different metrics, it is necessary to take into account that the position of the speaker turns are not exactly defined, due to the presence of inter-speaker silences or non-speech sounds. Therefore, it is considered that a change point is correctly located if it belongs to a time interval \([t_0 - \Delta t, t_0 + \Delta t]\) in which \(t_0\) is the reference mark and \(\Delta t\) is the tolerance. In our case, the tolerance is 0.25 sec.

### 3.4.5 Results

In this section, we will indicate FAR, MDR, PRC, RCL and F-measure achieved by our proposed method while segmenting the audio files and we use these parameters for comparing the performance of our method for MFCC features, residual phase features and for the combined features. For the combined system, the SVM misclassification rates are obtained using the relations
\[ M_{CS+} = \alpha M_{RP+} + (1 - \alpha) M_{MFCC+} \] (3.26)

\[ M_{CS-} = \alpha M_{RP-} + (1 - \alpha) M_{MFCC-} \] (3.27)

where

- \( M_{CS+} \) and \( M_{CS-} \) are misclassification rates of combined system for (+1) class and (-1) class, respectively.

- \( M_{RP+} \) and \( M_{RP-} \) are misclassification rates of residual phase for (+1) class and (-1), class respectively.

- \( M_{MFCC+} \) and \( M_{MFCC-} \) are misclassification rates of MFCC for (+1) class and (-1) class, respectively.

- \( \alpha \) is a scaling factor which varies from 0 to 1.

Then \( M_{CS+} \) and \( M_{CS-} \) are compared with \( t_{mc} \) to make the final decision in the combined system.

Fig. 3.5 shows the misclassification rates of residual phase, MFCC and residual phase with MFCC. From Table 3.1 and Fig. 3.6, we find that the speaker segmentation system using only the residual phase information results in low FAR but high MDR and the system using only the MFCC features results in low MDR but high FAR. However, the combined system results in moderate FAR and MDR and leads to an F-measure of 85.97% which is significantly better than the systems with either MFCC or residual phase.

Fig. 3.7 shows the snapshot of the speaker segmentation using support vector machine. The left side of the system is allocated for speaker segmentation and diarization. After segmentation, the segmented results are stored in a file as shown in the snapshot.
Fig. 3.5: (a) Speech signal. (b) and (c) Misclassification rates of residual phase. (d) and (e) Misclassification rates of MFCC. (f) and (g) Misclassification rates of residual phase with MFCC.

Table 3.1: Performance of speaker segmentation.

<table>
<thead>
<tr>
<th>Feature</th>
<th>FAR(%)</th>
<th>MDR(%)</th>
<th>PRC(%)</th>
<th>RCL(%)</th>
<th>F-measure(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Residual phase (RP)</td>
<td>20.15</td>
<td>33.61</td>
<td>63.67</td>
<td>53.00</td>
<td>57.85</td>
</tr>
<tr>
<td>MFCC</td>
<td>33.24</td>
<td>25.73</td>
<td>60.00</td>
<td>75.24</td>
<td>66.76</td>
</tr>
<tr>
<td>RP + MFCC</td>
<td>22.67</td>
<td>25.16</td>
<td>76.12</td>
<td>98.76</td>
<td>85.97</td>
</tr>
</tbody>
</table>

3.5 Summary

In this chapter we have proposed a method for improving the speaker segmentation performance by fusing the residual phase and MFCC features. This method is evaluated
Fig. 3.6: Performance of residual phase, MFCC, and residual phase with MFCC.

using television broadcast interviews and NIST 2004 database. The support vector machines are used to detect the speaker changes. The system reports a performance of 85.97%. The improvement in the performance is due to the complementary nature of speaker specific information present in the residual phase in comparison with the information present in the conventional MFCC. In the next chapter, a new approach for speaker segmentation will be proposed using autoassociative neural network.
Fig. 3.7: Snapshot of the speaker segmentation using SVM.