Wavelet-based Speaker Change Detection in Single Channel Speech Data

Michael Wiesenegger and Franz Pernkopf

Signal Processing and Speech Communication Laboratory
Graz University of Technology, Austria
mwiese@sbox.tugraz.at, pernkopf@tugraz.at

Abstract
Speaker segmentation is the task of finding speaker turns in an audio stream. We propose a metric-based algorithm based on Discrete Wavelet Transform (DWT) features. Principal component analysis (PCA) or linear discriminant analysis (LDA) [1] are further used to reduce the dimensionality of the feature space and remove redundant information. In the experiments our methods referred to as DWT-PCA and DWT-LDA are compared to the DISTBIC algorithm [2] using clean and noisy data of the TIMIT database. Especially, under conditions with strong noise, i.e. -10dB SNR, our DWT-PCA approach is very robust, the false alarm rate (FAR) increases by $\sim2\%$ and the missed detection rate (MDR) stays about the same compared to clean speech, whereas the DISTBIC method fails – the FAR and MDR is almost $\sim0\%$ and $\sim100\%$, respectively. For clean speech DWT-PCA shows an improvement of $\sim30\%$ (relative) for both the FAR and MDR in comparison to the DISTBIC algorithm. DWT-LDA is performing slightly worse than DWT-PCA.

Index Terms: Speaker change detection, discrete Wavelet transform, DISTBIC, audio diarization.

1. Introduction
Over the last years, recorded audio has increased in volume tremendously. This huge amount of data makes the search, retrieval, and indexing of information more difficult. Due to this fact it is necessary to find an efficient way to handle this data mass. Speaker segmentation and speaker clustering are tools that simplify the management of huge audio archives. Speaker segmentation is the task of finding speaker turns in an audio stream and speaker clustering performs the grouping of speech segments to find corresponding speakers, whereas every segment contains only one speaker. Speaker segmentation followed by speaker clustering is called audio diarization [3] [4].

Speaker change detection in a single channel audio stream is a very important task in various applications such as speaker tracking, automatic speaker recognition, speaker diarization, and indexing of audio recordings. In particular, speaker change detection without the use of prior knowledge (i.e. unsupervised) is of great interest [5], [6]. There are three main approaches to detect speaker turns [7]: (i) Energy-based segmentation: Speaker turns in an audio stream are detected mainly by measuring and thresholding the energy. Speaker segments are determined by cutting the audio signal at low energy regions. (ii) Metric-based segmentation: Two adjacent windows (overlapping or not overlapping) are placed and shifted over the sequence of extracted features. The similarity between the features captured by those windows is determined using a distance measure. Local maxima of the distance function indicate speaker turns. (iii) Model-based segmentation: A set of models (e.g. Gaussian mixture model) is trained prior to segmentation for a limited number of classes with different acoustic properties, e.g. speakers, music, noise, or silence. During segmentation speech is classified using the previously learned models. Segment boundaries are detected where changes of the acoustic properties occur.

In earlier research [7] [8] it turned out that metric-based and model-based speaker turn detection algorithms achieve almost the same performance. Furthermore, the metric-based approach keeps the algorithms simpler (mainly they are unsupervised) and they are less complex concerning computational requirements and tuning parameters. In [9], a metric-based approach is introduced using frequency domain features. These features are derived by reweighting the spectrogram using normalization techniques. This method is based on the assumption that the high frequency spectral components stay rather constant as long as no speaker turn occurs [10] – the fourth (sometimes the third) and higher formants are assumed to be more speaker specific. This property seems to come from the length differences of the vocal tract between speakers which influence mostly the higher order formants. Furthermore, the higher order formant frequencies provide a high interspeaker variability which supports the detection of speaker turns. Hence, these formants provide a good basis for speaker segmentation [11].

In this paper, we introduce two metric-based algorithms using Wavelet features. The potential advantage of a Discrete Wavelet Transform (DWT) [12] in speech processing is its inherent multiscale representation: namely, a DWT allows a multiscale representation of speech signals in the time-scale domain. In other words, various positions in the time-frequency plane are analyzed with different time-frequency resolutions. This allows e.g. higher frequencies to be granted the higher temporal resolution they naturally require, and lower frequencies to be granted the fine spectral resolution they require. Principal component analysis (PCA) or linear discriminant analysis (LDA) [1] are further used to reduce the dimensionality of the feature space and remove redundant information. We provide empirical results on the TIMIT data [13] showing that our DWT-PCA and DWT-LDA algorithms provide a significantly better false alarm rate (FAR) and missed detection rate (MDR) for clean speech compared to the DISTBIC algorithm [2] which is used as baseline. DWT-PCA slightly outperforms DWT-LDA. Furthermore, in the case of heavy additive white Gaussian noise, i.e. $-10$dB signal-to-noise ratio (SNR), the performance of DISTBIC rapidly degrades, whereas for DWT-PCA the FAR and MDR stay almost constant.

The paper is organized as follows: In Section 2, our DWT-PCA and DWT-LDA algorithms are introduced. Empirical results on the TIMIT data are presented in Section 3. Furthermore, this section includes the experimental setup and details on parameter optimization. Last, Section 4 concludes the paper.
2. DWT-PCA and DWT-LDA

Generally, metric-based speaker turn detection is divided into three steps, i.e. feature extraction, distance measurement, and classification, as shown in Figure 1 and introduced in the following.

![Figure 1: Metric-based speaker turn detection.](image)

2.1. Feature Extraction

A discrete-time signal \( x[k] \) can be represented as

\[
x[k] = \sum_{m=1}^{M} \sum_{n=1}^{N_m} \langle \psi_{m,n}[k], x[k] \rangle \psi_{m,n}[k],
\]

where \( \langle \cdot \rangle \) denotes the inner product, \( M \) represents the number of scales, \( N_m = \frac{N_f}{2^m} \) is the number of coefficients at the \( m \)th scale, and \( N_f \) is the number of samples in one speech frame. The set of discrete-time wavelet basis functions \( \psi_{m,n}[k] = a_0^{-m/2} \psi(a_0^{-m} k - nb_0) \) are generated by translating and dilating the mother wavelet \( \psi(k) \) using iterated filters. With \( a_0 = 2 \) and \( b_0 = 1 \) we obtain the dyadic-parameter wavelet basis functions.

The discrete-time signal \( x[k] \) can be further decomposed into the sum of one approximation plus \( M \) detail subbands at \( M \) resolution stages by a decimated non-uniform filterbank as follows:

\[
x[k] = \sum_{m=1}^{N_m} \sum_{n=1}^{X(m)[2n]} \cdot g_0^{(m)}[k-2^mn] + \sum_{m=1}^{M} \sum_{n=1}^{X(m)[2n+1]} \cdot g_1^{(m)}[k-2^mn],
\]

where \( X(m)[2n] \) and \( X(m)[2n+1] \) are the approximation coefficients (low-frequency part) and the detail coefficients (high-frequency parts), respectively. They are defined as:

\[
X^{(M)}[2n] = \langle h_0^{(M)}[2Mn-l], x[l] \rangle, \quad \text{and} \quad \nabla
\]

\[
X^{(m)}[2n+1] = \langle h_1^{(m)}[2^mn-l], x[l] \rangle,
\]

where \( g_j^{(m)}[k] \) is an equivalent filter obtained through \( m \) stages of synthesis filters \( g_j[k] \), each preceded by a factor of two up-sampler, \( h_j^{(m)}[k] \) is an equivalent analysis filter where \( h_j^{(m)}[k] = g_j^{(m)}[-k] \), \( j \in \{0, 1\} \), \( k, m, n \in \mathbb{Z} \).

The multilevel DWT decomposition known as Wavelet Packet Decomposition [12] is computed at level \( m = 5 \) using Symmlet as mother wavelet. Hence, we decompose the \( i \)th speech frame \( x[k] \) in one approximation subband and five detail subbands which form the sequence of wavelet coefficients \( W_{5,i}[n] = \{X(5)[2n], X(5)[2n+1]\} = \{X(3)[2n], X(3)[2n+1], X(3)[2n+1], X(3)[2n+1], X(3)[2n+1]\} \), where \( m = 5, 4, 3, 2, 1 \) and the number of coefficients in the 5th approximation subband \( A_5 = \frac{N_f}{32} \) and the five detail subbands are \( D_5 = \frac{N_f}{16}, D_4 = \frac{N_f}{16}, D_3 = \frac{N_f}{16}, D_2 = \frac{N_f}{16}, D_1 = \frac{N_f}{16} \}. \)

Performing the DWT on the input speech data results in a high dimensional feature space, i.e. 512. We use either PCA or LDA to reduce the dimension of the feature vectors to a space of lower dimensionality. PCA is a linear projection of the data onto an orthogonal subspace maximizing the variance of the projected data points [1]. Whereas, LDA uses the class information (i.e. speaker turn and no speaker turn) of the data to find the linear projection of the data which maximizes the ratio of the between-class variance to the within-class variance [1].

2.2. Distance Measurement

The Kullback Leibler (KL) distance [14] between two adjacent windows which are moved over the extracted and projected features is computed (see Figure 2). It is assumed that if the distance between the features within the windows is small both windows belong to one speaker. Otherwise, if this distance is large the features in the windows are associated with different speakers. The KL distance is widely used due to its fast computation and excellent results [7]. In particular, it is computed according to

\[
KL(A, B) = D(P_A(i)||P_B(i)) = \sum_i P_A(i) \log \frac{P_A(i)}{P_B(i)},
\]

where \( P_A(i) \) and \( P_B(i) \) are probability distributions of the projected feature values (either from PCA or LDA) in window \( A \) and the adjacent window \( B \), respectively. In particular, for \( P_A(i) \) (similarly for \( P_B(i) \)) we take all projected feature values in window \( A \) and rearrange them as a vector \( Y_A(i) \), whereas \( P_A(i) = \frac{Y_A(i)}{\sum_{i=1}^{N} Y_A(i)} \) and \( |Y_A(i)| \) denotes the length of the vector.

Since \( KL(A, B) \neq KL(B, A) \) we use the symmetric KL distance, i.e. \( KL2 = 1/2 [KL(A, B) + KL(B, A)] \).

![Figure 2: Distance computation.](image)

2.3. Classification

The classification of speaker turns is performed with a simple local maximum detection [2]. At first, the KL2 distance is smoothed by means of a moving average filter. Then, speaker
turns are detected if the local maxima \( d(\text{max}) \) of the smoothed KL2 distance satisfies the following condition:

\[
|d(\text{max}) - d(\text{min}_r)| > \alpha \sigma
\]

where \( d(\text{min}_l) \) and \( d(\text{min}_r) \) are the surrounding local minima, \( \alpha \) is a tuning parameter in percent to adjust the selectivity of the detector, and \( \sigma \) denotes the variance of the distance measure over the utterance. A maximum is regarded as significant if the value of the difference between the maximum \( d(\text{max}) \) and the surrounding minima \( d(\text{min}_l) \) and \( d(\text{min}_r) \) is above the threshold \( \alpha \sigma \) (see Eq. 5).

This is shown in Figure 3.

3. Experiments

3.1. Experimental Setup

For our experiments we take 200 randomly selected speakers from the TIMIT data [13], each utters one sentence. Hence, there are \( 199 \) speaker turns. We perform 100 independent runs using 200 randomly selected speakers and average the results. The average speaking length is \( 2.5 \) s and the silence between the utterances is removed. Furthermore, we test the performance of the algorithms with additive white Gaussian noise at different SNR levels, i.e. \( 80 \) dB, \( 40 \) dB, \( 20 \) dB, \( 10 \) dB, \( 5 \) dB, \( 0 \) dB, \( -5 \) dB, \( -10 \) dB, whereas the parameters of DWT-PCA, DWT-LDA, and DISTBIC are optimized for clean speech.

The FAR and MDR are used to measure the performance of the algorithms [2]. A false alarm (FA) occurs if there is a speaker turn detected although it does not exist and a missed detection (MD) occurs when the algorithm does not respond in the case of a speaker turn. Formally, the FAR and MDR are:

\[
FAR = 100 \times \frac{\text{number of FA}}{\text{number of actual speaker turns} + \text{number of FA}} \%
\]

\[
MDR = 100 \times \frac{\text{number of MD}}{\text{number of MD} + \text{number of actual speaker turns}} \%
\]

A good segmentation algorithm should recognize the real speaker turns, i.e. a segment should contain speech of only one speaker. If a segment includes more than a single speaker the MDR is large and the data is undersegmented. If there is a large FAR the data is oversegmented. For the evaluation we define a tolerance area (TA) of \( 1 \) s around the true speaker turn. Within TA the detected turn is considered to be correct. For DWT-PCA and DWT-LDA we compute the DWT for every speech frame of 512 samples (which corresponds to 32 ms at a sampling rate of 16 kHz) with a hop size of 16 ms. Then we reduce the feature vector dimension either to 20 using PCA or to a scalar value using LDA. The projection matrix for PCA is determined on the same data which are used for performance evaluation (empirical results show it performed better). For the DISTBIC algorithm the 12 Mel-frequency cepstral coefficients (MFCCs) are computed of a \( 32 \) ms frame shifted by \( 16 \) ms. DISTBIC [2] is a metric-based algorithm consisting of two steps. First, the MFCCs are extracted, the distance measurement based on the KL distance is determined assuming that the data in the adjacent windows is Gaussian distributed, and the classification of speaker turns is performed similar as in Section 2.3. In the second refinement step, the determined speaker turns are re-evaluated using the Bayesian information criterion.

3.2. Results on Clean Speech

All algorithms require additional parameters, i.e. the length \( \omega \) of the feature set windows \( A \) and \( B \) used to compute the distance measure (see Section 2.2), the step size for KL2 extraction is set to \( \delta = 96 \) ms, the smoothing coefficient \( \rho \) determines the size of the moving average filter for smoothing KL2, \( \alpha \) is the threshold of the local maximum detection (see Section 2.3), and DISTBIC uses \( \lambda \) to weight the penalty term in the refinement step [2]. Those parameters have been determined through extensive experiments for clean speech and are summarized in Table 1, Table 2, and Table 3 for DISTBIC, DWT-PCA, and DWT-LDA, respectively. The parameters are optimized so that we are near the equal error rate, which is the rate where FAR equals MDR. Once the optimal parameters have been established the best achieved FAR and MDR values over 100 independent runs (each uses 200 randomly selected speakers) are averaged and shown in the tables. DWT-PCA and DWT-LDA significantly outperform DISTBIC, whereas DWT-PCA is superior to DWT-LDA.

Table 1: FAR and MDR of the optimized DISTBIC.

<table>
<thead>
<tr>
<th>parameter</th>
<th>values</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \omega )</td>
<td>22</td>
</tr>
<tr>
<td>( \alpha )</td>
<td>25</td>
</tr>
<tr>
<td>( \lambda )</td>
<td>0.7</td>
</tr>
<tr>
<td>( \rho )</td>
<td>3</td>
</tr>
</tbody>
</table>

\[
\text{FAR} = 30.4\% \pm 2.9 \quad \text{MDR} = 32.3\% \pm 3.1
\]

Table 2: FAR and MDR of the optimized DWT-PCA.

<table>
<thead>
<tr>
<th>parameter</th>
<th>values</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \omega )</td>
<td>0.6</td>
</tr>
<tr>
<td>( \alpha )</td>
<td>10</td>
</tr>
<tr>
<td>( \rho )</td>
<td>3</td>
</tr>
</tbody>
</table>

\[
\text{FAR} = 20.0\% \pm 2.1 \quad \text{MDR} = 19.6\% \pm 2.5
\]

Table 3: FAR and MDR of the optimized DWT-LDA.

<table>
<thead>
<tr>
<th>parameter</th>
<th>values</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \omega )</td>
<td>0.4</td>
</tr>
<tr>
<td>( \alpha )</td>
<td>10</td>
</tr>
<tr>
<td>( \rho )</td>
<td>3</td>
</tr>
</tbody>
</table>

\[
\text{FAR} = 26.0\% \pm 2.3 \quad \text{MDR} = 18.7\% \pm 2.9
\]
3.3. Results on Noisy Speech

After optimizing the parameters of the algorithms we evaluate them on noise corrupted speech. The FAR and MDR values at different SNR levels (i.e. 80 dB, 40 dB, 20 dB, 10 dB, 5 dB, 0 dB, -5 dB, -10 dB) are reported in Figure 4 and 5 for DISTBIC, DWT-PCA, and DWT-LDA, respectively. Again, plots are averaged over 100 independent runs, each with 199 speaker turns. The performance show that the DISTBIC algorithm is less robust against background noise, whereas the wavelet based algorithms perform well. In particular, the DWT-PCA achieves even in the case of strong noise (i.e. -10 dB) a FAR of 21.9% and a MDR of 18.7%. Additionally, this algorithm uses one parameter less than the DISTBIC method. DWT-LDA is performing slightly worse than DWT-PCA. For all algorithms we notice that if the FAR increases the MDR decreases and vice versa. More results and different variants of wavelet-based algorithms can be found in [15].

![Figure 4: FAR performance for DISTBIC, DWT-PCA, and DWT-LDA.](image1)

![Figure 5: MDR performance for DISTBIC, DWT-PCA, and DWT-LDA.](image2)

4. Conclusion

We developed two speaker turn detection algorithms based on wavelet features using either principal component analysis or linear discriminant analysis for feature dimensionality reduction. The performance of the introduced approaches is compared to the DISTBIC algorithm which is known as a well-performing algorithm for the task of finding speaker turns. For clean speech our DWT-PCA shows a significant improvement of ∼30% (relative) for both the FAR and MDR in comparison to the DISTBIC algorithm. Furthermore, DWT-LDA significantly outperforms DISTBIC with respect to FAR and MDR. Additionally, DISTBIC performs poorly in the case of added white Gaussian noise, especially when the noise is strong. For this scenario, the DWT-PCA algorithm is very robust, the false alarm rate increases by ∼30% and the missed detection rate stays about the same compared to clean speech.

For future work it is necessary to reduce the severe influence of the parameters on the performance of all algorithms. Ideally, a learning algorithm should be used to adapt the parameters over time.

5. References


