Detection of Acoustic Change-Points in Audio Records
via
Global BIC Maximization and Dynamic Programming
Jindrich Zdansky and Jan Nouza

Department of Electronics & Signal Processing
Technical University of Liberec, Halkova 6, 461 17 Liberec 1, Czech Republic
{jindrich.zdansky, jan.nouza}@vslib.cz

Abstract
In this paper we propose a novel method for the detection of relevant changes in continuous acoustic stream. The aim is to identify the optimal number and position of the change-points that split the signal into shorter, more or less homogeneous sections. First we describe the theory we used to derive the segmentation algorithm. Then we show how this algorithm can be implemented efficiently. Evaluation is done on broadcast news data with the goal to segment it into parts belonging to different speakers. In simulated tests with artificially mixed utterances the algorithm identified 97.1% of all speaker changes with precision of 96.5%. In tests done with 30 hours of real broadcast news (in 9 languages) the average recall was 80% and precision 72.3%.

1. Introduction
Text transcription of broadcast news, political debates, talkshows or meetings is one of the most promising applications of modern voice technologies. Recently available systems for automatic speech recognition (ASR) can help in these tasks if there is a preprocessor that splits the continuous stream of spoken data into shorter parts. In the optimal case these parts are utterances spoken by a single speaker, which may help the ASR system in selecting an adequate speaker (or gender) specific model. In a more general case these segments should be at least acoustically homogeneous so that standard signal processing techniques (like cepstral mean subtraction) are applied properly.

During the last decade several speech stream segmentation methods have been proposed. Most of them are two-stage procedures. The first stage generates change-point candidates and is usually based on adaptive [1] or fixed [2] window length scenario using Bayesian Information Criterion (BIC), Kullback-Leibler distance, normalized log-likelihood ratio, etc. The second stage confirms or rejects produced candidates and it is mostly based on BIC.

In our lab we have been involved in the long-term research dealing with the automatic transcription of Czech broadcast programs. Therefore, besides the development of a language specific speech recognition system, we had to focus also on solving the problem with the broadcast stream segmentation. Our first solution consisted in the sequential application of the BIC [3]. This was done by moving a several-second-long window through the signal and searching for peaks in the BIC curve. These peaks - potential candidates for the searched speaker change-points - were rejected or confirmed in the second pass where the BIC (with another set of parameters) was employed again. The weak point in this approach was its free parameters which had to be estimated in advance and which were task dependent. A better solution consisted in a hierarchical (top-to-down) application of the BIC [4]. For the given speech record (or its sufficiently long portion) we detected the point of the largest change first and split it into two parts. For each part we applied the same splitting strategy and repeated it until the largest BIC peak got below a given threshold.

In this paper we present a more general scheme. It considers the entire acoustic stream as a random process that was generated by a finite state machine. In the proposed method - which is similar to Viterbi speech decoding - we search for the segments that belong to the machine states. Unknown is the number of the states as well as the parameters of these states. The latter makes the main difference between the classic speech modeling/decoding scheme and ours.

The paper is structured as follows: In section 2 we define the problem and describe it by a probabilistic model. In section 3 we present our solution which is based on global maximization of the BIC. Experiments performed on artificially prepared data as well as on real broadcast news recordings are described in section 4. In the final part we discuss the results.

2. Theoretical part
Let us suppose that we have a long audio record composed of diverse non-overlapping regions. The task of the segmentation is to identify all these regions, i.e. to locate the change-points in the record.

2.1. Formal definition
Let us suppose that we have data,
\[ \mathbf{x} = \{x_1, x_2, \ldots, x_T\} \]
which is a sequence of T observation vectors of an unknown stochastic process. Given the data \( \mathbf{x} \) we define growing sequence of positive integers
\[ t_i^s = \{t_0^i, t_1^i, t_2^i, \ldots, t_s^i\} \]
and denote it as i-th segmentation of S-th order, where \( t_i^s \) is position of s-th segment-end in the data. It is evident that segmentation must fulfill condition
\[ t_0^i < t_1^i < t_2^i < \cdots < t_s^i \]
where \( t_0^i = 0, t_s^i = T \).
2.2. Model of data production

In more formal terms, the segmentation task can be described as follows: Having a data sequence, we are looking for its division into regions (sub-sequences) such that every two neighboring ones can be considered as coming from independent sources. Let us further assume that these sources are modeled by states in a S-state finite state machine. For data $x$ and segmentation $t^s$ we assume probability density function

$$p(x|t^S, \phi^S_t) = \prod_{s=1}^{S} p(x_s|\phi^S_s),$$

(4)

where $\phi^S_s$ is vector of parameters for s-th state and $x_s = \{x_{ts^{-1}+1}, \ldots, x_{ts}\}$ are data belonging to the s-th segment.

2.3. Marginal likelihood approximation

The problem is that parameters $\phi^S_s$ are not known. To overcome this we employ the Bayesian Information Criterion ($BIC$) [5] which provides an approximation of the marginal logarithmic likelihood $\log p(x|m)$ given the model structure $m$. The $BIC$ is defined as

$$BIC(x|m) = \log p(x|\phi_m, m) - \frac{C}{2} \log T,$$

(5)

where $C$ is the number of free parameters of the model that should be estimated, $T$ is the data length and $\phi_m$ is the maximum likelihood ($ML$) estimate of model parameters. This criterion consists of two terms. The first one calculates how well the model predicts the data and the second one penalizes the complexity of the model.

2.4. Optimal segmentation

When we are able to express at least the approximation of the data likelihood given the segmentation, we can use a Minimum Error Rate classifier to get the best segmentation. According to the Bayes' decision theory it is based on the maximization of posterior probability $p(x|t^S_t)P(t^S_t)$. Since we have no reliable information about the prior term $P(t^S_t)$, we consider it to be uniform or non-informative. Using such an assumption makes the segmentation process fully controlled by the observed data. Then we should solve the following maximization problem

$$t_{best} = \arg \max_{\forall t^S_t} BIC(x|t^S_t),$$

(6)

i.e. compute $BIC$ for all possible numbers of segments and all possible change-points locations and find the maximum.

2.5. Solution for model with Gaussian state pdfs

It is well known that Gaussians well model speech represented by cepstral features. To make the solution easier we consider the underlying variables $X = \{X_{ts-1+1}, \ldots, X_{ts}\}$ to be independent. Using this assumption we can express log-likelihood of data in frames between position $t^{-1}_s$ and $t^s_s$ (considering the ML estimate of the Gaussian on the same data) as

$$\ell(t^s_s, t^{-1}_s) = \frac{t^s_s - t^{-1}_s}{2} \log |\Sigma| + d + d \log (2\pi),$$

(7)

where $d$ is the dimension of feature vectors and $|\Sigma|$ is the determinant of covariance matrix of data $x_s$. Now, we can express the final formula for computing $BIC$ as

$$BIC(x|t^S_s) = \sum_{s=1}^{S} \ell(t^s_s, t^{-1}_s) - \lambda \frac{Sc}{2} \log T,$$

(8)

where $\lambda$ is a weighting factor of the penalty term. Variable $c$ denotes the number of free parameters of the Gaussian model and can be computed from dimension $d$ as

$$c = d + \frac{1}{2} d (d + 1).$$

(9)

In the following section we show how to implement the maximization of Eq. 8 efficiently.

3. Proposed segmentation algorithm

The algorithm is based on the dynamic programing approach. Looking on Eq. 8 it is clear, that we can split the solution into two parts. First we compute the first term, i.e. we get the best segmentation for the given number of segments. This should be done for all possible numbers of segments and we will call it as forward part of the algorithm. Then we apply the penalty term, choose the best segmentation order and do the backtracking. This is called the backward part of algorithm.

3.1. Forward part

To estimate the covariance matrix (in Eq. 7) reliably we should put constraints on the minimal segment length $T_{min}$. This automatically determines the maximum number of segments $S_{max}$. Now, we define two time arrays for each model state. The first one $f_s(t)$ remembers $BIC$ for the best path from the beginning to the current time $t$, passing through states $1 \ldots s$. The second one $g_s(t)$ holds the position of the previous segment-end.

The outline of the forward part is shown in Figure 1. For each frame $t$ and each possible segment beginning $\tau$ we compute $\ell(t, \tau)$. Then for each state we find the position $\tau$ which maximizes current $f_s(t)$ and store it in $g_s(t)$.

```plaintext
for \ t = T_{min}, \ldots, T 
  f_1(t) = \ell(t, 0) 
  g_1(t) = 0 
for \ s = 2, \ldots, S_{max} 
  \tau = (s - 1) T_{min}, \ldots, t - T_{min} 
  f_s(t) = \max_{\forall \tau} [f_{s-1}(\tau) + \ell(t, \tau)] 
  g_s(t) = \arg \max_{\forall \tau} [f_{s-1}(\tau) + \ell(t, \tau)] 
end
```

Figure 1: Outline of the forward part of the algorithm.

3.2. Backward part

After the forward part of algorithm is solved we choose the best order of the segmentation, i.e.

$$S_{best} = \arg \max_{s=1} \ldots S_{max} \left[ f_s(T) - \lambda \frac{Sc}{2} \log T \right].$$

(10)

where $s = 1, \ldots, S_{max}$. Then we perform the backtracking (described in Figure 2), which finds the desired change-points and provides the final segmentation.
\[
t(S_{best}) = T
\]
for \( s = S_{best}, \ldots, 1 \)
\[
t(s - 1) = g_s(t(s))
\]
end

Figure 2: Getting the final segmentation - backtracking.

### 3.3. Reducing computational load

The most time-consuming part of the algorithm if the computation of Eq. 7, because we need to calculate the determinant of the covariance matrix for each time \( t \) almost \( t \)-times (see Fig. 1). So we should make the computation of the determinant more efficient. We define following arrays:

\[
\begin{align*}
  z_1(t) &= z_1(t - 1) + x_t^T \mu, \\
  z_2(t) &= z_2(t - 1) + x_t x_t^T,
\end{align*}
\]

where \( x_t^T \) denotes transposition of vector \( x_t \). Because
\[
\Sigma = E[(x - \mu)(x - \mu)^T] = E(x x^T) - \mu \mu^T,
\]

we are able to form it quickly relying on the fact, that
\[
\mu \approx \frac{z_1(t) - z_1(1)}{t - 1}, \\
E(x x^T) \approx \frac{z_2(t) - z_2(1)}{t - 1}.
\]

Since the covariance matrix is symmetric and positive definite we are able to form it quickly relying on the fact, that
\[
\mu \approx \frac{z_1(t) - z_1(1)}{t - 1}, \\
E(x x^T) \approx \frac{z_2(t) - z_2(1)}{t - 1}.
\]

### 3.4. Reduction of search-space

Another efficient way to increase the algorithm speed is to prune the number of possible paths through the model. We assume that being in time \( t \), we can get the order of the segmentation in time \( t - 1 \) as
\[
S_{act} = \arg \max_{S_{act}} \left[ f_s(t - 1) - \lambda_{min} \frac{sc}{2} \log (t - 1) \right]
\]

where parameter \( \lambda_{min} \) controls the number of actually computed state variables. Knowing that being in time \( t \), the segmentation order can not be higher than \( S_{act} + 1 \), we can substitute \( S_{max} \) in the forward part by this value.

Even a more serious problem is to constrain the possible values of \( \tau \) (see Fig. 1). We know that the higher \( \lambda \) in Eq. 16 we choose the lower segmentation order (the longer segments) we obtain. Thus we use the formula similar to Eq. 16
\[
S_{long} = \arg \max_{S_{long}} \left[ f_s(t - 1) - \lambda_{max} \frac{sc}{2} \log (t - 1) \right]
\]

to get constraint on possible values of \( \tau \). Now, it takes form of
\[
\tau = g_{S_{long}}(t - 1), \ldots, t - T_{min}
\]

and this term replaces the one in Figure 1. This kind of pruning increases the algorithm speed significantly.

Hence, the proposed segmentation algorithm requires three parameters \( \lambda_{min} < \lambda < \lambda_{max} \). Parameter \( \lambda \) has impact on the number of the produced segments and the other two ones have influence mainly on speed. It is clear that the longer record we want to segment, the more memory we need. To avoid this problem, the algorithm should be implemented using e.g. circular buffer, which puts a constraint on the maximum segment length. For the maximum segment length of \( T_{max} = 60000 \) (10 minutes), the segmentation program runs almost synchronously with real-time using P4 2.4 GHz computer with 1 GB of memory.

### 4. Experimental part

#### 4.1. Signal Processing

In all the experiments described in this section we used 16kHz sampled waveforms converted into MFCC features computed every 10 ms using 25 ms window. For the segmentation, the first 12 MFCCs were used. (The zero-th coefficient, i.e. the energy was omitted.)

#### 4.2. Segmentation performance measures

First, the change-points computed by the algorithm are assigned to the reference ones iff the former is the closest to the latter and vice versa. In addition, if the distance between the two ones is smaller than 1 second, we call these change-points linked. There are three statistical measures commonly used to evaluate the segmentation process:

\[
R = \frac{H}{N} \quad P = \frac{H}{H + T} \quad F = \frac{2RP}{R + P}
\]

where \( R, P, F \) are called recall, precision and F-rate. Symbols \( N, H, I, D \) denote reference, linked, inserted, deleted numbers of change-points respectively.

#### 4.3. Artificially mixed database

For a detailed evaluation of the algorithm we created a large set of artificially mixed data. The reason was threefold:

- in such an artificial stream we can control the number and parameters of partial segments,
- we know the exact position of change-points,
- the test set can be made large enough to get statistically credible results.

As source we used our database of broadcast news shows that contains about 5000 segments belonging to several hundreds of speakers. From each segment we removed silence and noise (non-speech) parts if they occurred at the beginning or end. (This was done to make the change-points position clearly defined.) After that the segments were randomly concatenated to form a training and testing database. Both of them contained one hundred 10-minute-long artificial records.

![Figure 3: Dependency of recall, precision and F-rate on penalty weight \( \lambda \).](image-url)

To estimate the optimal penalty weight \( \lambda \), we used F-rate measure as the criterion. The dependency of F-rate, recall and precision values on the \( \lambda \) obtained for the training data is shown.
in Figure 3. Maximum F-rate value (96.86%) was achieved for λ = 1.11.

The evaluation of the segmentation algorithm on the testing part of the database with the above penalty value yielded the following results: The algorithm found 97.12% of all existing speaker changes (recall) and 96.48% of the detected changes were correct (precision). A histogram showing how large was the difference between the true and found change-points is in Fig. 4. Let us notice that 66.7% or 95% of change-points were determined with position error smaller than 40 ms or 180 ms, respectively.

![Figure 4: Histogram of time-alignment errors.](image)

### 4.4. COST 278 BN database

The data used in the second experiment were collected as part of the pan-European Broadcast News Database by 10 institutions from 9 countries collaborating in the European COST 278 action on Spoken Language Interaction in Telecommunication [6]. Each institution provided 3 hours of its national broadcast news records.

<table>
<thead>
<tr>
<th>Language</th>
<th>Recall [%]</th>
<th>Precision [%]</th>
<th>F-rate [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dutch</td>
<td>77.1</td>
<td>82.1</td>
<td>79.5</td>
</tr>
<tr>
<td>Czech</td>
<td>76.5</td>
<td>85.8</td>
<td>80.9</td>
</tr>
<tr>
<td>Galician</td>
<td>94.4</td>
<td>64.5</td>
<td>76.6</td>
</tr>
<tr>
<td>Greek</td>
<td>76.7</td>
<td>69.5</td>
<td>72.9</td>
</tr>
<tr>
<td>Croatian</td>
<td>87.9</td>
<td>60.6</td>
<td>71.8</td>
</tr>
<tr>
<td>Hungarian</td>
<td>72.5</td>
<td>78.9</td>
<td>75.6</td>
</tr>
<tr>
<td>Portuguese</td>
<td>82.3</td>
<td>75.8</td>
<td>78.9</td>
</tr>
<tr>
<td>Slovenian</td>
<td>85.8</td>
<td>67.7</td>
<td>75.7</td>
</tr>
<tr>
<td>Slovenian 2</td>
<td>76.9</td>
<td>61.3</td>
<td>68.2</td>
</tr>
<tr>
<td>Slovak</td>
<td>69.6</td>
<td>76.5</td>
<td>72.9</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>80.0</strong></td>
<td><strong>72.3</strong></td>
<td><strong>75.3</strong></td>
</tr>
</tbody>
</table>

Table 1: Speaker segmentation results for COST 278 BN DB.

To estimate the λ for the experiment with real data we created another artificially mixed database. To make it more realistic we composed it from segments that represented not only different speakers but also different acoustic conditions. Moreover, this time we did not remove silence and noise parts, as we did it in the previous experiment. Using this data we got maximum F-rate = 85.89% value for penalty weight λ = 1.61. After that we applied the algorithm on the real multi-lingual BN data. The results obtained for all languages are in Table 1. They were achieved for the following settings of the other free parameters: $\lambda_{\min} = 1$, $\lambda_{\max} = 4$, $T_{\min} = 50$ (500ms) and $T_{\max} = 60000$ (10 min.).

### 5. Discussion

The experiments performed with the artificially created data proved that the proposed segmentation algorithm provided very good results (though the conditions were almost ideal). It found 97.12% of all existing speaker changes and 96.48% of the detected change-points were correct. Furthermore, most of the change-points were placed without affecting speech, i.e. not inside words.

When the segmentation was applied on the real BN data we obtained reasonably good results compared to our previous approaches. In this case, the system found 80% of all speaker changes and 72.3% of them were correct. After detailed analysis we learned that the most frequent type of erroneous segmentation was insertion. This is not so surprising because we focused on the evaluation of speaker changes only, although in real news streams that are many significant changes in background noise or in channel characteristics. The latter definitely make separate regions in an acoustic signal (and its MFCC representation). Another source of insertions was spontaneous speech, which is typically full of silence and hesitation sounds. The second type of errors, deletions, was mostly a side-effect of the effort to suppress the insertions by increasing the λ penalty. Sometimes, deletions occurred when a speaker change happened within the segment with high background noise or in case the segment was too short (typically less than 2 seconds).

### 6. Conclusions

In this work we present a novel approach to spoken data stream segmentation. To distinguish from the methods that were published earlier and that considered signal changes in rather local manner, we call this approach global BIC maximization. We demonstrated its performance on a very practical task, which was broadcast news segmentation made preferably with respect to speaker turns.

### 7. Acknowledgments

This work was part of activities in European COST 278 Action and it was supported by grant no. OC278.02 provided by Czech Ministry of Education.

### 8. References


