BINSEG: An Efficient Speaker-based Segmentation Technique

Jindrich Zdansky

Department of Electronics & Signal Processing
Technical University of Liberec, Halkova 6, 461 17 Liberec 1, Czech Republic
jindrich.zdansky@tul.cz

Abstract

In this paper we present a new efficient approach to speaker-based audio stream segmentation. It employs binary segmentation technique that is well-known from mathematical statistic. Because integral part of this technique is hypotheses testing, we compare two well-founded (Maximum Likelihood, Informational) and one commonly used (BIC difference) approach for deriving speaker-change test statistics. Based on results of this comparison we propose both off-line and on-line speaker change detection algorithms (including way of effective training) that have merits of high accuracy and low computational costs. In simulated tests with artificially mixed data the on-line algorithm identified 95.7% of all speaker changes with precision of 96.9%. In tests done with 30 hours of real broadcast news (in 9 languages) the average recall was 74.4% and precision 70.3%.

Index Terms: speaker change detection, acoustic segmentation.

1. Introduction

Text transcription of broadcast news, political debates, talk-shows 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.

This paper is organized as follows. Section 2 briefly describes theoretical background we have used to develop segmentation algorithms. Section 3 is devoted to the proposals of off-line and on-line versions of speaker change detection algorithms and Section 4 presents achieved results.

2. Theoretical Background

Speaker change detection can be considered as a specific task well-known from mathematical statistic - a *change point analysis*. Change point analysis is engaged in two cardinal problems: *single* and *multiple change point problem*. While several well-founded approaches to solution of single change point detection have been proposed, multiple change point problem has not been satisfactorily solved yet. That is why the problem of more than one change point detection is often solved via decomposition into sequence of single change point detection tasks. In this paper we prove that one of these decomposition procedures -*binary segmentation technique* [1]- is applicable on speaker-based segmentation task.

2.1. Speaker-based Binary Segmentation

The binary segmentation procedure has several merits: it detects both the number and positions simultaneously, it is easy to implement and it is very fast. Let us denote $X = x_i \in R^d, i = 1, 2, \ldots, T$ as the sequence of frame-based cepstral vectors extracted from an audio stream. Let us consider these vectors to be sequence of $d-$dimensional normal random vectors with underlying Gaussian process parameters $N(\mu_i, \Sigma_i)$. Assume μ_i, Σ_i are unknown and we are concerned in both mean and variance change detection. Then speaker change detection via binary segmentation can be summarized in the following steps:

Step 1: Test hypothesis that there is no change point in the data $X = x_i \in R^d, i = 1, 2, \ldots, T$:

$$H_0 : \mu_1 = \mu_2 = \cdots = \mu_T, \Sigma_1 = \Sigma_2 = \cdots = \Sigma_T \quad (1)$$

versus alternative

$$\begin{aligned} H_1 : \mu_1 &= \cdots = \mu_t \neq \mu_{t+1} = \cdots = \mu_T \quad \text{and} \\ \Sigma_1 &= \cdots = \Sigma_t \neq \Sigma_{t+1} = \cdots = \Sigma_T, \end{aligned} \quad (2)$$

where t is the location of the single change point at this stage. If H_0 is accepted, then there is no change point and the algorithm is stopped. In the case of H_0 rejection go to **Step 2**.

Step 2: Split the signal into two subsequences according to change point found in **Step 1** and test them for the possible single change point.

Step 3: Repeat this process until no further subsequences have change points.

Step 4: The locations of change points estimated by steps 1-3 are $\{\hat{t}_1, \hat{t}_2, \ldots, \hat{t}_S\}$ and their total number is S.

It was proved [1] that this segmentation procedure provides consistent estimate of true change-points.

2.2. Speaker-change Test Statistics

Hypotheses testing employed in binary segmentation procedure assumes derivation of proper test statistics. In this work we compare three reasonable test statistics matching the previously stated problem. All of them share a log-likelihood ratio term:

$$R(t) = T \log |\hat{\Sigma}| - t \log |\hat{\Sigma}_1| - (T - t) \log |\hat{\Sigma}_T|, \quad (3)$$

where $\hat{\Sigma}_1$, $\hat{\Sigma}_T$, $\hat{\Sigma}$ are covariances of data $x_1, \ldots, x_t$, $x_{t+1}, \ldots, x_T$, $x_1, \ldots, x_T$ respectively. In [1] it is shown that solution of equation

$$\hat{t} = \arg \max_{d<t<T-d} R(t) \quad (4)$$

provides strongly consistent estimate of true change point location. All the examined test statistics are known as *maximum type statistics* [2] and are based on following approaches:

Maximum Likelihood (ML) is based on the work described in [3] and the test statistics takes form:

$$\Lambda_{ML}^{\hat{t}} = \alpha \sqrt{\max_{d<t<T-d} R(t)} - \beta, \tag{5}$$

where

$$\alpha = (2 \log\log T)^{\frac{1}{2}} \text{ and} \tag{6}$$
$$\beta = 2 \log\log T + d \log\log\log T - \log\Gamma(d). \tag{7}$$

Informational 1 (I1) is described e.g. in the book [1] and the test statistics is:

$$\Lambda_{I1}^{\hat{t}} = \max_{d<t<T-d} (R(t) - D \log T) \tag{8}$$

$$D = d + \frac{1}{2} d(d+1). \tag{9}$$

Informational 2 (I2) is not a true test statistics, but it is widely used in speaker change detection algorithms, namely BIC difference based ones [4, 5].

$$\Lambda_{I2}^{\hat{t}} = \max_{d<t<T-t} \frac{R(t)}{D \log T}, \tag{10}$$

where D is defined by Equation 9.

When the condition $\Lambda^{\hat{t}} > K$ is fulfilled the null hypothesis is rejected and the estimated change point $\hat{t}$ is given by Eq. 4.

3. Proposed Segmentation Algorithms

3.1. Off-line Algorithm

The idea behind the presented approach is that change points are revealed recursively, always looking for the only single change point, examining the area between two previously detected ones. Let us define *node* (see Figure 1) that corresponds to estimated change point. Each node has assigned two basic properties: the *location* of the change point in the data and the *active* property. The latter indicates whether the appropriate node is intended for further processing or not. Further we define:

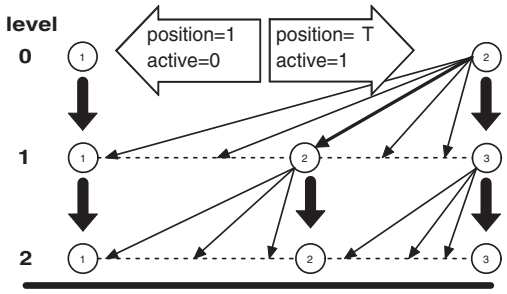
- an interval $< a; b >$ that should be checked for possible change point;
- a *gain* $G(t|a,b)$ associated with decision that t is a change point in the interval $< a; b >$.

The gain computation is represented by *arcs* (see Figure 1) and for individual approaches it takes form:
Maximum Likelihood:

$$G_{ML}(t|a,b) = \alpha \big[(b-a+1) \log|\hat{\Sigma}| - (t-a+1) \log|\hat{\Sigma}_a| - (b-t) \log|\hat{\Sigma}_b| \big]^{\frac{1}{2}} - \beta, \tag{11}$$

Informational 1:

$$G_{I1}(t|a,b) = (b-a+1) \log|\hat{\Sigma}| - (t-a+1) \log|\hat{\Sigma}_a| - (b-t) \log|\hat{\Sigma}_b| - D \log(b-a+1) \tag{12}$$




```
do
 for each node
  if is active
   assign arcs from location of node-1 to node
   get arc with maximum value of G(t|a,b)
    if G>threshold
      establish new node in the next level
    else
      active=0
    ---
   ---
  copy current node to the next level
 ---
level++;
while there is any active node
```

Figure 1: *Simplified outline of the multiple change point detection algorithm - binary segmentation technique.*

Informational 2:

$$G_{I2}(t|a,b) = \frac{(b-a+1) \log|\hat{\Sigma}| - (t-a+1) \log|\hat{\Sigma}_a| - (b-t) \log|\hat{\Sigma}_b|}{D \log(b-a+1)} \tag{13}$$

Choosing the arc that maximizes $G(t|a,b)$ we get the best possible change point location $\hat{t}$. If the condition $G(\hat{t}|a,b) > K$ is fulfilled, new node is established ($active = 1$, $position = \hat{t}$). The threshold K is the only free parameter of this algorithm, which needs to be estimated. Simplified algorithm outline and its graphical form are shown in Figure 1.

3.2. Optimal Threshold K Estimation

Important merit of binary segmentation is that it could be trained very fast. We build the tree shown in Figure 1 only once with some small value of threshold K. While building the tree we must remember true values of gain G associated with creation of each node. Then we get collection of change points (from the last level of tree) for each $K_i > K$ across whole training database and we choose optimal value K_{opt}.

3.3. On-line Algorithm

For on-line speaker-based segmentation purpose we have adopted a variable-size increasing window scheme proposed in famous paper [4]. Unlike this scheme, where only one change point was expected in actually examined data, the algorithm based on binary segmentation procedure does not put any restrictions on the number of potential change points.

After a large number of experiments we have proposed an on-line segmentation algorithm which works as follows. We initialize an analysing window of an arbitrary size. Then we perform single change point detection. If no change point is detected, the analysing window is enlarged. If a change point is found we take the left part of the data and perform detection again. This is repeated until no further change point is detected. As will be shown in Section 4, detection of the leftmost change point brings even better results than the off-line version of the algorithm. The on-line algorithm can be summarized in following steps:

Step 1: Initialize interval $< a; b >$, where $a = 1$ and $b = a + B$;

Step 2: Compute gain $G(t|a, b)$, for $a + d < t < b - d$.

Step 3: Choose $\hat{t}$ which maximizes $G(t|a, b)$.

Step 4:
- If $G(\hat{t}|a, b) < K$ (there is no change point in the interval $< a; b >$) then enlarge right border of the interval $b = b + B$ and goto **Step 2**.
- If $G(\hat{t}|a, b) \geq K$ (there is at least one change point in the interval $< a; b >$) goto **Step 5**.

Step 5: Decrease size of the interval $b = \hat{t}$ and detect further change point in $< a, b >$.

Step 6: Repeat **Step 5** until no further change point is detected. Then the interval border b is the desired change point.

Step 7: Set a new interval: $a = b+1$, $b = a+B$ and goto **Step 2**.

After the end of the audio stream is reached (i.e. $b \geq T$), b is set to $b = T$ and the off-line binary segmentation is performed on the rest of data.

3.4. Implementation Tricks

The most time-consuming part of the algorithm is the computation of Eq. 3 mainly due to calculation of the determinants of the covariance matrices. To make these computations more efficient, we define following arrays:

$$z_1(t) = z_1(t-1) + x_t \quad (14)$$

$$z_2(t) = z_2(t-1) + x_t x_t^{\mathbf{T}}, \quad (15)$$

where $x_t^{\mathbf{T}}$ denotes transposition of vector x_t. Because

$$\Sigma = E[(x-\mu)(x-\mu)^T] = E(xx^T) - \mu\mu^T, \quad (16)$$

we are able to form covariance of data $< a; t >$ quickly relying on the fact, that

$$\mu \approx \frac{z_1(t) - z_1(a-1)}{t-a+1} \quad (17)$$

$$E(xx^T) \approx \frac{z_2(t) - z_2(a-1)}{t-a+1}. \quad (18)$$

Since the covariance matrix is symmetric and positive definite we can use Choleski factorization to get the determinant of Σ very fast.

4. Evaluation of Proposed Algorithms

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:

$$R = \frac{H}{N};\ P = \frac{H}{H+I};\ F = \frac{2RP}{R+P}, \quad (19)$$

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.

To compare two speaker change detection systems operating on identical data, we employed statistic significance test (SST) described in paper [6] and the level of significance was chosen $\alpha_0 = 0.1\%$.

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 and approximately 9000 speaker changes.

4.3.1. Evaluation of the off-line algorithm

The first experiment was to draw a comparison of ML, I1 and I2 approaches to hypotheses testing in binary segmentation procedure. The optimal threshold K was estimated on the training database (see Section 3.2) with a goal to maximize *F-rate* measure. After that we ran the algorithm on testing data with the estimated threshold setting and we obtained results that are summarized in the Table 1. Improvements of ML and I1 on I2 and also ML on I1 were confirmed by SST.

Database	Training		Testing		
Approach	K	F_{max} [%]	F [%]	R [%]	P [%]
ML	72.5	96.51	96.27	95.73	96.82
I1	570	96.32	96.05	95.73	96.73
I2	2.05	94.39	94.19	92.92	95.49

Table 1: *The off-line version of the proposed binary segmentation procedure: comparison of ML, I1 and I2 approaches to hypotheses testing.*

4.3.2. *Evaluation of the on-line algorithm*

Due to the results obtained in previous experiment we have incorporated the ML approach to hypotheses testing in the on-line algorithm . The threshold $K = 72.5$ was estimated in the same way as it was described in the previous experiment and a comparison of results provided by the off-line and on-line algorithm is shown in the Table 2. Improvement of the on-line on the off-line algorithm was again proved by SST. Let us notice that 66.7% ($\Delta_{2/3}$) or 95% ($\Delta_{0.95}$) of change-points were determined with position error smaller than 30 ms or 180 ms, respectively and 45.96% (δ_{10}) of them were determined with position error smaller than 10 ms.

Algorithm	F[%]	$\Delta_{2/3}$ [ms]	$\Delta_{0.95}$ [ms]	δ_{10} [%]
On-line	96.85	30	180	45.96
Off-line	96.27	40	250	42.63

Table 2: *Comparison: off-line ($K = 72.5$) versus on-line ($B = 15$ s, $K = 72.5$) algorithm with incorporated ML approach to hypotheses testing.*

The last experiment conducted on artificially mixed data was to compare common speaker change detection algorithm (described in [4, 5]) that employs I2 (known as BIC segmentation) and variable-size increasing window scheme with the proposed on-line algorithm. Figure 2 shows, that newly proposed algorithm reaches approximately 2% better *F-rate* (it is mainly due to ML instead of I2) for short window sizes B. Further it is shown that with growing B, performance of newly proposed algorithm slightly grows, while in the case of the common algorithm it goes rapidly down.

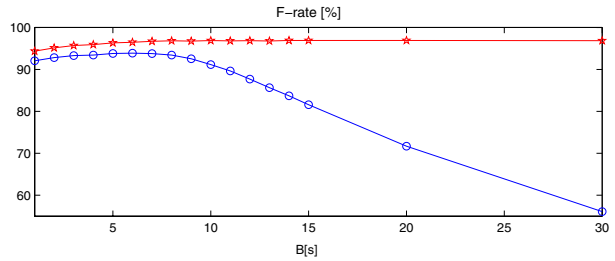


Figure 2: *Comparison of on-line algorithms incorporating variable-size increasing window scheme: proposed(⋆) ($B = 15$ s, $K = 72.5$) versus common(∘) ($B = 15$ s, $K = 2.05$).*

4.4. COST 278 BN Database

The data used in this 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 [7]. Each institution provided 3 hours of its national broadcast news records.

Because the COST BN database does not contain enough data to create training and testing databases we have adopted following scenario to increase credibility of results: training and threshold tuning is performed on one national data set and testing is done on the remaining data sets, and this procedure is repeated ten times for all national sets. Results of this experiment are summarized in the Table 3.

Database	Training		Testing		
Part	K	F_{mx} [%]	F [%]	R [%]	P [%]
Czech	80.0	81.46	71.51	82.80	62.93
Greek	95.5	71.69	72.75	71.31	74.25
Galician	104.5	76.53	69.78	63.37	77.63
Hungarian	82.5	73.55	72.61	81.75	65.31
Portuguese	95.5	77.92	72.46	71.10	73.88
Slovenian	93.0	73.46	73.02	72.56	73.48
Slovenian2	92.0	67.25	73.40	73.79	73.01
Slovak	77.0	71.71	70.16	85.24	59.61
Croatian	107.5	71.97	68.73	61.13	78.49
Dutch	82.0	76.04	72.10	81.37	64.73
ϕ	90.95	74.16	71.65	74.44	70.33

Table 3: *Evaluation of the proposed on-line segmentation algorithm ($B = 15$ s) on the COST 278 BN database.*

5. Conclusions

This paper has presented a novel approach to speaker-based segmentation - binary segmentation technique. On the basis of this technique we have proposed an on-line speaker change detection algorithm that has merits of high accuracy and low computational costs. On 3 GHz processor it takes 5% of computational power (when runs in real-time mode) and average delay of detection was 6 seconds. In simulated tests with artificially mixed data the algorithm identified 95.7% of all speaker changes with precision of 96.9%. In tests done with 30 hours of real broadcast news (in 9 languages) the average recall was 74.4% and precision 70.3%.

6. Acknowledgements

This work was partly supported by project 1QS108040569 of the Grant Agency of the Czech Academy of Science.

7. References

[1] Chen, J., Gupta, A. K., *Parametric Statistical Change Point Analysis*, Birkäuser, Boston, 2000.

[2] Lauro, C., Antoch, J., Vinzi, V. E., Saporta G., *Multivariate Total Quality Control*, Physica-Verlag, Heidelberg, 2002.

[3] Horváth, L., *The Maximum Likelihood Method for Testing Changes in the Parameters of Normal Observations*, Annals of Statistics, Vol. 21, No 2., pp. 671-680,1993.

[4] Chen, S. S., Gopalakrishnan, P. S., *Speaker, Environment and Channel Change Detection and Clustering via the Bayesian Information Criterion*, IBM T.J. Watson Research Center, Yorktown Heights, NY, Technical Report, 1998.

[5] Zhou, B., Hansen, J. H. L., *Efficient Audio Stream Segmentation via Combined T^2 Statistic and Bayesian Information Criterion*, IEEE Transaction on Speech and Audio Processing, Vol. 13, No. 4, July 2005.

[6] Zdansky, J., *Speaker Change Detection via Binary Segmentation Technique and Informational Approach*, Proc. of SPECOM 2006, St. Petersburg (Russia), 2006.

[7] Vandecatseye, A. et al., *The COST278 pan-European broadcast news database*, Proc. of LREC 2004, Lisbon (Portugal), 2004.