MultiBIC: an Improved Speaker Segmentation Technique for TV Shows

Paula Lopez-Otero, Laura Docio-Fernandez, Carmen Garcia-Mateo
Department of Signal Theory and Communications, Universidade de Vigo, Spain
{plopez, ldocio, carmen}@gts.tsc.uvigo.es

Abstract
Speaker segmentation systems usually have problems detecting short segments, which causes the number of deletions to be high and therefore harming the performance of the system. This is a complication when it comes to segmenting multimedia information such as movies and TV shows, where dialogues among characters are very common. In this paper a modification of the BIC algorithm is presented, which will reduce remarkably the number of deletions without causing an increase in the number of false alarms. This modification, referred to as MultiBIC, assumes that two change-points are present in a window of data, while conventional BIC approach supposes that there is just one. This causes the system to notice when there is more than one change-point in a window, finding shorter segments than traditional BIC.

Index Terms: speaker segmentation, TV shows, miss detection reduction

1. Introduction
A recent application where speaker segmentation systems are found to be useful is the processing of multimedia databases. Nowadays, with the widespread use of digital audio and video, the possibilities for processing and manipulating this kind of data are diverse. In many multimedia applications, it is interesting to be able to use the data to extract some information that allows moving through the contents of the audio and video in order to find something concrete. For example, laughter detection [1][2] can help to identify which parts of a movie are funny, speaker detection and tracking [3][4] finds out when a particular actor or character is talking, and combining speaker identification and speaker segmentation allows identifying dialogues [5].

To perform these tasks and similar ones it is important to have a good segmentation of the data to be processed.

There are lots of publications about speaker segmentation systems [6][7], being the Bayesian Information Criterion (BIC) algorithm widely employed [8][9][10]. The performance of these systems is good, but it has to be taken into account that not all the speaker and audio segmentation systems need to confront the same problems. In our previous research on speaker segmentation [11][12], we have focused on audio data extracted from European Parliament sessions which were recorded and annotated within the EU project “Technology and Corpora for Speech to Speech Translation” (TC-STAR) [13]. This database is characterized by long audio segments spoken by the same speaker that should not be broken. The existing conventional speaker segmentation systems have a high false alarm rate on such database. In order to solve that problem, statistical processes were used in [11][12] to reduce the number of false alarms. However, when data come from TV shows and movies, where a more natural and spontaneous talk is found, the dominant speech are dialogues, where it is common to have several speakers speaking in very short turns. The problem in this case is the opposite of the one found on the Parliament sessions: as the segments to be detected are short, the number of miss detections will be very high. Nevertheless, it is important that the reduction in the number of deletions does not imply an increase in the number of false alarms, which would solve a problem but cause a new one. In order to achieve these aims a novel BIC-based approach is proposed, which assumes that a given window of audio data can contain two change-points instead of just one, thereby detecting shorter segments and making this approach more desirable when the audio to be segmented comes from a TV show.

The rest of the paper is organized as follows. Section 2 describes the BIC Algorithm. Section 3 presents the new segmentation approach, referred to as MultiBIC. In Sections 4 and 5 the experimental framework and results are given. Finally, Section 6 summarizes the conclusions and discusses future work in this field.

2. The BIC algorithm
The BIC algorithm is broadly used in speaker segmentation systems. It is a model selection criterion that maximizes the log-likelihood penalized by the complexity of the model [6].

Given a window of data $X = (x_1, ..., x_L)$ of length $L$ frames extracted from an audio stream, it is divided into two sub-windows $X_1 = (x_1, ..., x_i)$ and $X_2 = (x_{i+1}, ..., x_L)$ (where $i$ is the $i^{th}$ frame in the window $X$) and a hypothesis test is run in order to select one of the two following possibilities:

- $H_0$: Both sub-windows $X_1$ and $X_2$ are modeled by the same model $M$.
- $H_1$: Sub-windows $X_1$ and $X_2$ are modeled by two different models $M_1$ and $M_2$.

This hypothesis test will be run splitting the window $X$ at different frames (different values of $i$). Given a resolution $r$, the value of $i$ will change according to it, and the frame $i$ that is the best candidate for a changing point has to be found by evaluating the following expression:

$$\Delta BIC(i) = L_i - \lambda P$$

where $P$:

$$P = \frac{1}{2}M \log L = \frac{1}{2}(d + \frac{1}{2}d(d+1)) \log L$$

is the penalty function as stated by Schwarz [14], where $M$ corresponds to the number of free parameters of the Gaussian model, and $\lambda$ is a weight that increases or decreases the influence of the penalty. When $\lambda$ is a small value, less changes will be discarded by the BIC algorithm; the opposite happens when...
\( \lambda \) gets bigger. This \( \lambda \) is usually selected by checking the performance of the system with different values on a development set.

If models \( M, M_1 \) and \( M_2 \) are single full covariance Gaussian models \( N(\mu, \Sigma), N(\mu_1, \Sigma_1) \) and \( N(\mu_2, \Sigma_2) \), the term \( L_i \) in Equation (1), which represents a log likelihood ratio, becomes:

\[
L_i = \frac{1}{2} \log|\Sigma| - \frac{1}{2} \log|\Sigma_1| - \frac{1}{2} \log|\Sigma_2| (3)
\]

where \( L, L_1 \) and \( L_2 \) are the number of frames of windows \( X, X_1 \) and \( X_2 \) respectively; and \( \Sigma, \Sigma_1 \) and \( \Sigma_2 \) are the covariance matrices of the models \( M, M_1 \) and \( M_2 \) respectively. Thus, there will be a potential change-point in the data when

\[
\frac{1}{2} \log|\Sigma| - \frac{1}{2} \log|\Sigma_1| - \frac{1}{2} \log|\Sigma_2| > \lambda P (4)
\]

To detect the change-point, first, the value of \( i \) that fulfills (4) and maximizes (1) will be chosen as a potential change-point. Then, to refine the position of this change-point or to discard it, a window will be centered on that potential change-point and the algorithm will be applied again but with a higher resolution, which means that the value of \( i \) will increase steadily. This change-detection approach uses a sliding window scheme for detection of change-points.

The approach presented in [10] was applied to reduce the computational cost of computing the covariance matrices when changing the value of \( i \).

3. MultiBIC: the proposed segmentation approach

The method proposed in this paper is a modification of the BIC algorithm that detects short segments. In the BIC procedure it is assumed that there is only one change-point inside the analysis window, however, the MultiBIC procedure considers that there might be one or two changes in a window. Thus, the window \( X \) will be split into two sub-windows: \( X_1 = (x_1, \ldots, x_i) \), \( X_2 = (x_{i+1}, \ldots, x_j) \) and \( X_3 = (x_{j+1}, \ldots, x_L) \). Therefore, MultiBIC will decide between the hypotheses:

- \( H_0 \): The three windows \( X_1, X_2 \) and \( X_3 \) are modeled by the same model \( M \).
- \( H_1 \): Windows \( X_1, X_2 \) and \( X_3 \) are modeled by 3 different models \( M_1, M_2 \) and \( M_3 \).

Then, a matrix of \( \Delta BIC \) values is obtained:

\[
\Delta BIC(i, j) = L_{ij} - \lambda P (5)
\]

where \( P \) is in this case

\[
P = d + \frac{1}{2} d(d+1) \log L (6)
\]

This expression comes from the following: the number of parameters of a Gaussian is

\[
M_1 = d + d((d+1)/2) (7)
\]

In (2) a model with two Gaussians is compared with a model with one Gaussian, so \( M = 2M_1 - M_1 = M_1 \) and \( P = \frac{1}{2} \lambda M \log L = \frac{1}{2} (d + \frac{1}{2} d(d+1)) \log L \). In this case, a model with three Gaussians is compared with a model with one Gaussian:

\[
M = 3M_1 - M_1 = 2M_1 = 2(d + d((d+1)/2)) (8)
\]

Then,

\[
P = \frac{1}{2} \lambda M \log L = \frac{1}{2} (d + d((d+1)/2)) \log L = (d + \frac{1}{2} d(d+1)) \log L (9)
\]

The log likelihood \( L_{ij} \) is computed like in (3), but in this case it will have another term:

\[
L_{ij} = \frac{1}{2} \log|\Sigma| - \frac{1}{2} \log|\Sigma_1| - \frac{1}{2} \log|\Sigma_2| - \frac{1}{2} \log|\Sigma_3| (10)
\]

As in traditional BIC, the values of \( i \) and \( j \) that maximize \( \Delta BIC \) are chosen, and therefore two potential change-points will appear, one in frame \( i \) and the other in frame \( j \). Then a traditional BIC step will be applied to refine or discard those potential change-points. At this point, one, two or no change-points will have been extracted from the current window.

4. Experimental framework

4.1. Database

For the evaluation of the MultiBIC segmentation procedure, as it is focused on TV shows, several episodes of two different sit-coms were used for the development and test sets. The development and Test 1 sets are episodes of How I Met Your Mother (three and two episodes, respectively), while Test 2 set is composed by two episodes of Friends. These TV shows have common characteristics, such as the abundance of canned laughter (pre-recorded laughs) and jingles to indicate changes of scene.

To find out if the results achieved in the sit-coms can be generalized to other kinds of TV shows, a third test set is included, consisting on an episode of the drama series One Tree Hill. This TV show features background music almost all the time, no canned laughter and no jingles between different scenes.

These four datasets, which were manually segmented in order to compare the correct segmentation with those obtained using BIC and MultiBIC, are described in Table 1. A forgiveness collar of 1 second will not be scored around each change point.

<table>
<thead>
<tr>
<th>Set</th>
<th>Total length</th>
<th>Number of segments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development</td>
<td>64 min</td>
<td>720</td>
</tr>
<tr>
<td>Test 1</td>
<td>41 min</td>
<td>466</td>
</tr>
<tr>
<td>Test 2</td>
<td>43 min</td>
<td>497</td>
</tr>
<tr>
<td>Test 3</td>
<td>42 min</td>
<td>402</td>
</tr>
</tbody>
</table>

To illustrate the length of the speaker turns, Figure 1 shows a histogram classifying the speech segments in the database used for the experiments in function of its length. In this figure, it can be noticed that a high number of speaker segments are short, specially with lengths about 1, 2 and 5 seconds. There are also some long speaker turns, but to a less extent. Thus, this database is suitable for this experiments, as it has many short speaker segments.

4.2. Acoustic features

The basic acoustic features used in the experiments were 12 Mel-frequency Cepstral coefficients (MFCC), which are extracted using a 25ms Hamming window at a rate of 10ms per frame. Then, these cepstral features are augmented by the normalized log-energy.
5. Experimental results

5.1. Evaluation metrics

Common metrics employed to evaluate the performance of a speaker segmentation system are Precision $P$ (% of detected points which are genuine change points), Recall $R$ (% of detected speaker change points) and F-score $F$, which is a combination of the other two measures. $P$ measures the number of insertions based on the number of changes that were found, $R$ attends to the number of deletions, and $F$ is a combination of these two parameters. The bigger those quality parameters, the better the performance of the system. The expressions used to compute these evaluation parameters are

$$P = \frac{c}{c + i} \times 100 \quad (11)$$

$$R = \frac{c}{c + d} \times 100 \quad (12)$$

$$F = \frac{(1 + \beta^2)PR}{\beta^2P + R} \quad (13)$$

where $c$ is the number of target changes, $i$ is the number of insertions (changes that are not real ones) and $d$ is the number of deletions (target changes that were not found).

The F-score has a $\beta$ value that behaves like a weight, giving more importance to precision or to recall. In this case, the value employed was 1, meaning that the F-score gives the same importance to $P$ and to $R$.

5.2. Choice of the $\lambda$ parameter

The development set described in section 4.1 was employed to choose the value of $\lambda$ that achieves the best performance for each system. Figure 2 shows the F-score achieved after applying the two algorithms to the development set using different values of $\lambda$, choosing $\lambda = 2$ for BIC and $\lambda = 1$ for MultiBIC, which means no weighting of the penalty.

Figure 3 shows the F-score obtained with MultiBIC for the test sets using different values of $\lambda$. It can be observed in the table that MultiBIC technique clearly outperforms BIC, achieving an important reduction in the miss-detection rate, causing a relative increase in the recall of 13% in Test 1, of 17% in Test 2, and of 14% in Test 3. Looking at the table it can be observed that the number of deletions (miss detections), I is the number of insertions (false alarms), $P$ is the precision, $R$ is the recall and $F$ is the F-score.

The fact that $\lambda = 1$ obtains the best performance in MultiBIC is very interesting because, as said, this means that there is no weighting of the penalty term. Thus, these experimental results seem to indicate that MultiBIC does not require a development set to tune the $\lambda$ parameter.

5.3. Results

Table 2 shows the performance of the two segmentation techniques applied to the three test datasets described in 4, where $C$ stands for the number of target change-points, $D$ is the obtained number of deletions (miss detections), $I$ is the number of insertions (false alarms), $P$ is the precision, $R$ is the recall and $F$ is the F-score.

Table 2: Performance of the Two Segmentation Techniques

<table>
<thead>
<tr>
<th>System</th>
<th>Test 1</th>
<th>Test 2</th>
<th>Test 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MultiBIC</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

It can be appreciated in Table 2 that MultiBIC technique clearly outperforms BIC, achieving an important reduction in the miss-detection rate, causing a relative increase in the recall of 13% in Test 1, of 17% in Test 2, and of 14% in Test 3.

Looking at the table it can be observed that the number of deletions has dropped under the 50% of the one obtained with BIC. In addition, there is also a reduction in the number of insertions, increasing slightly the precision. This results on a meaningful increase in the F-score when MultiBIC is used, making it useful when it comes to segmenting TV shows.
Another interesting fact is that the best performance achieved by MultiBIC is when using \( \lambda = 1 \). This means that there is no weighting of the penalty, so the algorithm do not really need a development stage in this experimental frame-work. However, such an advantage is not yet conclusive and more experiments in different databases must be done to find out if MultiBIC really avoids the weighting of the penalty, thus obtaining a system where no parameters have to be tuned.

Nevertheless, Table 2 and Figure 4 show that the number of miss-detections is still notorious, so in future work new techniques will be tested in order to improve even more the performance of speaker segmentation systems when it comes to TV shows.

It would also be interesting to perform a fusion between the audio and the video information in order to segment audiovisual documents taking advantage of the available data.

Combining MultiBIC with a clustering technique gives the possibility to perform dialog detection, scene segmentation, tracking of actors or characters. This kind of applications focused on TV shows will also be object of research in the future.

## 7. References


