BIC-BASED AUDIO SEGMENTATION BY DIVIDE-AND-CONQUER

Shih-Sian Cheng[1,2], *Hsin-Min Wang*[1], *and Hsin-Chia Fu*[2]

[1]Institute of Information Science, Academia Sinica, Taipei, Taiwan
E-mail: {sscheng, whm}@iis.sinica.edu.tw
[2]Department of Computer Science, National Chiao Tung University, Hsin-Chu, Taiwan
E-mail: hcfu@csie.nctu.edu.tw

ABSTRACT

Audio segmentation has received increasing attention in recent years for its potential applications in automatic indexing and transcription of audio data. Among existing audio segmentation approaches, the BIC-based approach proposed by Chen and Gopalakrishnan is most well-known for its high accuracy. However, this window-growing-based segmentation approach suffers from the high computation cost. In this paper, we propose using the efficient divide-and-conquer strategy in audio segmentation. Our approaches detect acoustic changes by recursively partitioning an analysis window into two sub-windows using ΔBIC. The results of experiments conducted on the broadcast news data demonstrate that our approaches not only have a lower computation cost but also achieve a higher segmentation accuracy than window-growing-based segmentation.

Index Terms— acoustic change detection, audio segmentation, Bayesian Information Criterion, divide-and-conquer

1. INTRODUCTION

The goal of audio segmentation is to detect acoustic changes in an audio stream, e.g., boundaries between two speakers or two environmental conditions. In the last decade, researchers in the speech processing community have put much effort on this problem for its potential applications to many speech and audio processing tasks, such as audio indexing [1], automatic transcription of audio recordings [2], speaker tracking [3], and speaker diarization [4]. Existing audio segmentation approaches generally fall into two categories, namely, distance-based segmentation [5, 6, 7, 8, 9, 10, 11] and model-decoding-based segmentation [12].

In distance-based segmentation, a distance measure of two audio segments is first defined, and then an acoustic change detection strategy is designed based on the distance measure. Compared to model-decoding-based segmentation, these methods have a great advantage that they do not need *a priori* knowledge about the content of the input audio stream. It is assumed that the acoustic feature vectors in each of the two audio segments are drawn from a probability distribution (e.g., multivariate Gaussian). Then, the distance between the two segments is represented as the dissimilarity between the two distributions. Many distance measures have been investigated, e.g., Kullback-Leibler distance (KL or KL2) [5], Generalized Likelihood Ratio (GLR) [10], ΔBIC [6, 8], Mahalanobis distance, and Bhattacharyya distance [11].

Fixed-size sliding window detection [5, 10, 11] and BIC-based growing-size sliding window detection [6, 7, 8, 9, 13] are two leading approaches in distance-based segmentation. In the fixed-size sliding window detection approach, a certain distance measure is used to evaluate the dissimilarity between two adjacent windows that slide along the audio stream to produce a distance curve. This distance curve is often low-pass filtered. Then, the locations of peaks are judged if they are acoustic changes by some heuristic thresholds. This method has the advantage of low computation cost. However, in order to detect the change boundary associated with a short homogeneous segment, the size of the analysis window is usually set at a small value (e.g., two seconds). This is a dilemma because a small analysis window does not contain sufficient feature vectors to obtain a reliable distance statistic.

BIC-based growing-size sliding window detection was first proposed by Chen and Gopalakrishnan [6]. For the distance measure of two audio segments, they used Bayesian Information Criterion (BIC) [14] to evaluate the following two hypotheses: 1) The union of the feature vectors of the two segments forms a Gaussian cluster in the feature space. 2) The feature vectors of each segment form a distinct Gaussian cluster. Then, the difference of the two evaluation scores, ΔBIC, was used as the distance measure. In their acoustic change detection procedure, a small analysis window is put at the beginning of the audio stream, initially. If there is no change point detected in the analysis window, it is enlarged to have a larger search range. However, with the window size growing, this approach suffers from a heavy computation cost due to numerous ΔBIC calculations, in particular when the audio stream contains many long homogenous segments. To reduce the computation cost, Tritschler and Gopinath [7] proposed some heuristics to ignore the distance computations at the locations where the acoustic changes unlikely happen. Zhou and Hansen [13] used the low computation cost Hotelling's T^2-Statistic as the distance measure in the detection process, while ΔBIC was used only to verify the acoustic change candidates. In [8] and [9], the authors proposed more efficient implementations for the ΔBIC calculation without affecting the detection accuracy. Since the growing-size sliding window detection approach detects acoustic changes using a size-growing analysis window, we denote it as *window-growing-based segmentation* (WinGrow).

In this paper, we propose two divide-and-conquer approaches that detect acoustic changes by recursively partitioning a large analysis window into two sub-windows using ΔBIC, rather than detecting acoustic changes with a size-growing analysis window. Inheriting from the efficiency property of divide-and-conquer paradigm, the proposed approaches are more efficient than WinGrow. The results of experiments conducted on the broadcast news data demonstrate that the proposed approaches not only have a lower computation cost

This work was supported by National Digital Archives Program (NDAP, Taiwan) sponsored by the National Science Council, Taiwan, under Grant: NSC95-2422-H-001-031.

but also achieve a higher segmentation accuracy than WinGrow.

2. WINDOW-GROWING-BASED SEGMENTATION

2.1. Model selection and BIC

Given a data set $\mathcal{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_n\} \subset \mathbb{R}^d$ and a set of candidate models $\mathcal{M} = \{M_1, M_2, \cdots, M_k\}$, the purpose of model selection is to choose the model that best fits the distribution of $\mathcal{Z}$ from $\mathcal{M}$. When using Bayesian Information Criterion (BIC) for model selection [14], the BIC value of M_i for $\mathcal{Z}$ is computed as

$$BIC(M_i, \mathcal{Z}) = \log p(\mathcal{Z} \mid \hat{\mathbf{\Theta}}_i) - \frac{1}{2}\lambda \#(M_i) \log n, \qquad (1)$$

where $\lambda = 1$, $\hat{\mathbf{\Theta}}_i$ is the maximum likelihood estimate of the parameter set of M_i, and $\#(M_i)$ is the number of parameters of M_i. The model that has the largest BIC value will be selected.

2.2. One-change-point detection

In the one-change-point detection algorithm proposed by Chen and Gopalakrishnan (denoted as OCD-Chen in this paper) [6], it is assumed that there is at most one change point in an audio stream $\mathcal{Z}$, and the following two hypotheses are tested sequentially on $\mathbf{z}_i$, $i = 1, \cdots, n$:

$$\begin{aligned} H_0 &: \mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_n \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma}). \\ H_1 &: \mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_i \sim \mathcal{N}(\boldsymbol{\mu}_1, \mathbf{\Sigma}_1); \\ &\quad \mathbf{z}_{i+1}, \mathbf{z}_{i+2}, \cdots, \mathbf{z}_n \sim \mathcal{N}(\boldsymbol{\mu}_2, \mathbf{\Sigma}_2). \end{aligned} \qquad (2)$$

The difference between the BIC values of H_1 and H_0 is computed as $\Delta BIC(i) = BIC(H_1, \mathcal{Z}) - BIC(H_0, \mathcal{Z}), i = 1, \cdots, n$. If max_i $\Delta BIC(i) > 0$, the time index corresponding to the maximum value is output as the change point. Otherwise, there is no change point in $\mathcal{Z}$. The penalty factor λ in Eq. (1) can be adjusted according to the tradeoff between error types in a practical audio segmentation task.

2.3. Multiple-change-points detection

For detecting multiple change points in an audio stream, OCD-Chen can be applied sequentially to a sliding, size-growing analysis window whose size is initialized at N_{ini} samples. If no change point is detected in the current analysis window, it is enlarged by N_g samples, and then OCD-Chen is applied again. The detection process continues until a change point is detected or the size of the analysis window exceeds a pre-defined upper bound N_{max}. If a change point is detected, the window size is reset to N_{ini}, and the detection process restarts at the latest change point. If no change point is detected, the analysis window of N_{max} samples is shifted by N_s samples, and OCD-Chen is applied until a change point is detected or the analysis window reaches the end of the audio stream. If a change point is detected, the detection process restarts at the latest change point with an analysis window of N_{ini} samples. In this way, the change points in the audio stream are detected sequentially.

3. DIVIDE-AND-CONQUER-BASED SEGMENTATION

3.1. The DACDec1 approach

We use the example in Fig. 1 to explain the potential advantages of detecting change points by divide-and-conquer. We assume that the audio stream consists of three homogeneous segments arising from different speakers. Initially, OCD-Chen is applied in an analysis window that includes the entire audio stream. After C_2 is detected, the audio stream is divided into two analysis windows. Then, OCD-Chen is applied in these two analysis windows to search the remaining change points, respectively, and C_1 will be detected. In this way, we can design a recursive divide-and-conquer procedure to detect the change points in an audio stream. The details of the proposed DACDec1 algorithm are illustrated in Fig. 2.

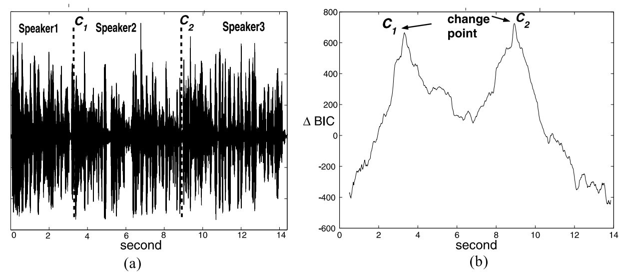

Fig. 1. (a) An audio stream that consists of three speech segments, each from a distinct speaker. (b) The ΔBIC curve obtained by OCD-Chen.

procedure $CP \leftarrow$DACDec1(W)
//input: W, the analysis window.
//output: CP, the set of change points detected in W.
Begin

1. detect whether there is a change point in W by OCD-Chen;
2. //Check termination
   if (there is no change point in W or the size of W is smaller than N_{min} samples)
   $CP \leftarrow \phi$; //empty set
   goto End;
3. //Divide
   let $\hat{t}$ be the change point detected in 1);
   divide W into two sub-windows, W_1 and W_2, at $\hat{t}$;
4. //Solve sub-instances
   $CP_{W_1} \leftarrow DACDec1(W_1); CP_{W_2} \leftarrow DACDec1(W_2);$
5. //Combine
   $CP \leftarrow \hat{t} \cup CP_{W_1} \cup CP_{W_2};$

End

Fig. 2. The DACDec1 algorithm.

In the above example, the three homogeneous segments in the initial analysis window arise from three distinct acoustic sources. However, if this condition is not met, DACDec1 may fail to detect the change points. For example, as shown in Fig. 3(a), the first and third segments arise from the same speaker (Speaker1) while the second segment arises from another speaker (Speaker2). When applying OCD-Chen to the audio stream in Fig. 3(a) with the same λ value as the example in Fig. 1, we obtain the ΔBIC curve in Fig. 3(b). From the figure, we see that the ΔBIC curve still has two peaks at the change points, C_1 and C_2. However, the ΔBIC values at C_1 and C_2 are smaller than zero; therefore no change point will be output by OCD-Chen. As shown in Figs. 3(c) and 3(d), at C_2, though H_1 models the distribution of the data samples better than it does at a non-change point R, H_1 over-fits the data samples of Speaker1 and obtains a smaller BIC value than H_0 does. We may adjust the value of λ so that, at C_2, the ΔBIC value will be larger than zero (i.e., the hypothesis testing will favor H_1). However, this may result in undesired false alarms when the recursive process

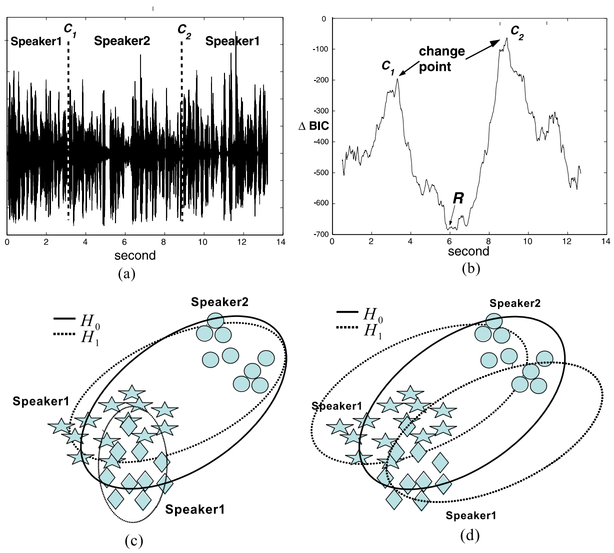

Fig. 3. (a) An audio stream that consists of three speech segmentation; the first and third segments arise from one speaker, while the second arises from another speaker. (b) The ΔBIC curve obtained by OCD-Chen. (c) The diagram of the hypothesis testing at the change point C_2. (d) The diagram of the hypothesis testing at the non-change point R.

procedure $CP \leftarrow$DACDec2(W)
//input: W, the analysis window .
//output: CP, the set of change points detected in W.
Begin

1. //Check termination
   if (the size of W is smaller than N_{min})
   $CP \leftarrow \phi$; //empty set
   goto End;
2. //Divide
   perform OCD-Chen on W, and let $\hat{t}$ be the time index with the largest ΔBIC value;
   divide W into two sub-windows, W_1 and W_2, at $\hat{t}$;
3. //Solve sub-instances
   $CP_{W_1} \leftarrow DACDec2(W_1); CP_{W_2} \leftarrow DACDec2(W_2);$
4. //Combine
   if ($\Delta BIC_{\{W_1,W_2\}}(\hat{t})$ calculated in 2) is larger than zero)
   $CP \leftarrow \hat{t} \cup CP_{W_1} \cup CP_{W_2}$;
   else
   let $\mathcal{X}$ be the segment left to $\hat{t}$ in CP_{W_1} and $\mathcal{Y}$ be the segment right to $\hat{t}$ in CP_{W_2};
   if ($\Delta BIC_{\{\mathcal{X},\mathcal{Y}\}}(\hat{t}) > 0$) //$\hat{t}$ is a change point
   $CP \leftarrow \hat{t} \cup CP_{W_1} \cup CP_{W_2}$;
   else //$\hat{t}$ is not a change point;
   merge $\mathcal{X}$ and $\mathcal{Y}$;
   $CP \leftarrow CP_{W_1} \cup CP_{W_2}$;

End

Fig. 4. The DACDec2 algorithm.

executes change point detection in a homogeneous segment. In other words, it is difficult to determine a reliable λ for an audio stream like the example in Fig. 3(a). Moreover, it is infeasible to adjust the value of λ for each specific audio stream in practical applications.

3.2. The DACDec2 approach

To overcome the shortcoming due to the unreliable BIC statistic in DACDec1, we develop an alternative implementation for the divide-and-conquer paradigm, called DACDec2. As described in Fig. 4, in the *Check termination* stage, the ΔBIC value is not used to check termination since it may be unreliable as we have explained with Fig. 3. The recursive process proceeds till the size of the analysis window is smaller than N_{min} samples. In the *Divide* stage, the analysis window is divided into two sub-windows at the time index $\hat{t}$ which has the largest ΔBIC value by OCD-Chen. Then, they are input into DACDec2 in the *Solve sub-instances* stage. In the *Combine* stage, $\hat{t}$ is labeled as a change point if the ΔBIC value at $\hat{t}$ calculated in the $Divide$ stage is larger than zero; otherwise, it needs to be checked again using its two neighbor segments $\mathcal{X}$ and $\mathcal{Y}$. In the second check, $\hat{t}$ is labeled as a change point only if $\Delta BIC_{\{\mathcal{X},\mathcal{Y}\}}(\hat{t}) > 0$.

3.3. Sequential divide-and-conquer segmentation

Given a long audio stream, such as a one-hour broadcast news show, the segmentation task becomes computationally intractable when using DACDec1 or DACDec2; besides, if their initial analysis window contains too many segments, it may be difficult for OCD-Chen to have an appropriate λ value to obtain robust ΔBIC measurements for the various hypothesis testings in the recursive process. Therefore, in practical applications we apply DACDec1 and DACDec2 in a large fixed-size analysis window, say 20 seconds, that slides from the beginning to the end of the audio stream; we call them the SeqDACDec1 and SeqDACDec2 approaches, respectively. In SeqDACDec1 (or SeqDACDec2), if there is any change point detected in the fixed-size analysis window by DACDec1 (or DACDec2), the fixed-size analysis window is shifted to the change point with the largest time index. Otherwise, the fixed-size analysis window is shifted forward by ηL samples, where η is a positive number and L denotes the size of the fixed-size analysis window. Comparing to DACDec1 and DACDec2, SeqDACDec1 and SeqDACDec2 are more suitable for on-line applications.

4. EXPERIMENTS

Our experiments were conducted on the broadcast news data. Three one-hour broadcast news shows selected from the MATBN Mandarin Chinese broadcast news corpus [15] were used as the development set (denoted as MATBN3hr); the 1998 DARPA/NIST HUB-4 broadcast news evaluation test data, which consisted of two 1.5-hour audio streams, was used as the evaluation set (denoted as HUB4-98). There are 1386 and 1184 acoustic change points in MATBN3hr and HUB4-98, respectively.

For feature extraction, each audio stream was converted into a sequence of 24-order MFCC feature vectors [6] by a 32-ms Hamming-windowed frame with 10-ms shifts.

For the performance evaluation, we adopted the Receiver Operating Characteristic (ROC) curve, which shows the tradeoff between the miss detection rate and the false alarm rate. In this study, a true change point t was considered missed if there was no hypothesized change point within $[t-1, t+1]$ (a two-second window centered on t); and a hypothesized change point $\hat{t}$ was counted as a false alarm if there was no true change point within $[\hat{t}-1, \hat{t}+1]$.

4.1. System description and parameter setting

We used fixed-size sliding window detection (FixSlid) and window-growing-based segmentation (WinGrow) as our baselines. For FixS-

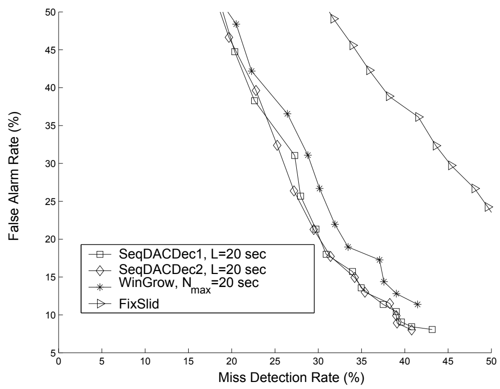

Fig. 5. ROC curves for HUB4-98 obtained by SeqDACDec1, SeqDACDec2, WinGrow, and FixSlid.

Table 1. The running time of audio segmentation approaches evaluated on HUB4-98. The last column shows the speedup over WinGrow.

Approach	CPU time	Speedup
WinGrow	8418.23 sec	1
SeqDACDec1	2003.62 sec	4.20
SeqDACDec2	3853.48 sec	2.18

lid, GLR was used as the distance measure of two adjacent analysis windows, the analysis window size was fixed at two seconds, and the decision mechanism proposed by [10] was adopted, in which all the time indices corresponding to "significant" peaks on the distance curve were considered as change points. For WinGrow, the values for N_{ini} and N_{max} were tuned with the development set; the values for N_g and N_s were set at one second and N_{max}/4 seconds, respectively. For SeqDACDec1 and SeqDACDec2, η was fixed at 0.25; L and the N_{min} in DACDec1 and DACDec2 were tuned with the development set.

4.2. Experiment results

We first conducted experiments on the development set (MATBN3hr) for tuning the parameters. We found that, for WinGrow, it was appropriate to set N_{ini} and N_{max} at three seconds and 20 seconds, respectively. For both SeqDACDec1 and SeqDACDec2, we found it was appropriate to set N_{min} at two seconds and L at 20 seconds. With the above parameter settings, the EERs by FixSlid, WinGrow, SeqDACDec1, and SeqDACDec2 were about 26%, 17%, 17%, and 16%, respectively. We then conducted experiments on HUB4-98 with the same parameter settings as on MATBN3hr. Fig. 5 shows the ROC curves obtained by the baseline systems and the proposed algorithms. We observe that FixSlid performs the worst. Both SeqDACDec1 and SeqDACDec2 achieve an EER of about 27%, while WinGrow achieves an EER of about 29%. Table 1 summarizes the running time of WinGrow, SeqDACDec1, and SeqDACDec2 in the EER case. The programs were run with a 3.2GHz Intel Pentium IV CPU. It is obvious from the table that both SeqDACDec1 and SeqDACDec2 are more efficient than WinGrow.

5. CONCLUSIONS

We have proposed two new BIC-based approaches for audio segmentation. Instead of searching the acoustic changes in an audio stream in a bottom-up manner, which has been widely adopted in previous studies, the proposed approaches adopt a divide-and-conquer procedure that searches acoustic changes in a top-down manner. The results of experiments conducted on the broadcast news data demonstrated that the proposed approaches not only have a lower computation cost but also achieve a higher segmentation accuracy than the well-known window-growing-based audio segmentation approach.

6. REFERENCES

[1] H. Meinedo and J. Neto, "Audio segmentation, classification and clustering in a broadcast news task," in *ICASSP*, 2003.

[2] P. C. Woodland, M. J. F. Gales, D. Pye, and S. J. Young, "The development of the 1996 HTK broadcast news transcription system," in *DARPA Speech Recognition Workshop*, 1997.

[3] J. F. Bonastre, P. Delacourt, C. Fredouille, T. Merlin, and C. Wellekens, "A speaker tracking system based on speaker turn detection for NIST evaluation," in *ICASSP*, 2000.

[4] S. E. Tranter and D. A. Reynolds, "An overview of automatic speaker diarization systems," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1557–1565, 2006.

[5] M. Siegler, U. Jain, B. Raj, and R. Stern, "Automatic segmentation, classification and clustering of broadcast news audio," in *DARPA Speech Recognition Workshop*, 1997.

[6] S. Chen and P. Gopalakrishnan, "Speaker, environment and channel change detection and clustering via the Bayesian information criterion," in *DARPA Broadcast News Transcription and Understanding Workshop*, 1998.

[7] A. Tritschler and R. Gopinath, "Improved speaker segmentation and segments clustering using the Bayesian information criterion," in *EUROSPEECH*, 1999.

[8] M. Cettolo, M. Vescovi, and R. Rizzi, "Evaluation of BIC-based algorithms for audio segmentation," *Computer Speech and Language*, vol. 19, pp. 147–170, 2005.

[9] P. Sivakumaran, "On the use of the Bayesian information criterion in multiple speaker detection," in *EUROSPEECH*, 2001.

[10] P. Delacourt and C. J. Welkens, "DISTBIC: A speaker-based segmentation for audio data indexing," *Speech Communication*, vol. 32, no. 1, pp. 111–126, 2000.

[11] J. W. Hung, H. M. Wang, and L. S. Lee, "Automatic metric-based speech segmentation for broadcast news via principal component analysis," in *ICSLP*, 2000.

[12] R. Bakis et. al., "Transcription of broadcast news shows with the IBM large vocabulary speech recognitoin system," in *DARPA Speech Recognition Workshop*, 1997.

[13] B. W. Zhou and John H. L. Hansen, "Efficient audio stream segmentation via the combined T^2 statistic and Bayesian information criterion," *IEEE Trans. Speech and Audio Processing*, vol. 13, no. 4, pp. 467–474, 2005.

[14] G. Schwarz, "Estimation the dimension of a model," *Annals of Statistics*, vol. 6, pp. 461–464, 1978.

[15] H. M. Wang, B. Chen, J. W. Kuo, and S. S. Cheng, "MATBN: A Mandarin Chinese broadcast news corpus," *International Journal of Computational Linguistics and Chinese Language Processing*, vol. 10, no. 2, pp. 219–236, 2005.