EFFICIENT AUDIO SEGMENTATION ALGORITHMS BASED ON THE BIC

Mauro Cettolo[1] *and Michele Vescovi*[2]

[1]ITC-irst
Centro per la Ricerca Scientifica e Tecnologica
I-38010 Povo di Trento - Italy

[2]Università degli Studi di Trento
Facoltà di Scienze MM.FF.NN.
I-38010 Povo di Trento - Italy

ABSTRACT

A widely adopted algorithm for the audio segmentation is based on the Bayesian Information Criterion (BIC), applyed within a sliding variable-size analysis window. In this work, three different implementations of that algorithm are analyzed in detail: (i) one that keeps updated a pair of sums, that of input vectors and that of square input vectors, in order to save computations in estimating covariance matrixes on partially shared data; (ii) one, recently proposed in the literature, that exploits the encoding of the input signal with cumulative statistics for the efficient estimation of covariance matrixes ; and (iii) an original one, that encodes the input stream with the cumulative pair of sums of the first approach.

The three approaches have been compared both theoretically and experimentally, and the proposed original approach will be shown to be the most efficient.

1. INTRODUCTION

In the last years, many efforts have been devoted to the problem of audio segmentation by the research community. This is due to the number of applications of this procedure, that range from the information extraction from audio data (e.g. broadcast news, meetings recording), to the automatic indexing of multimedia data, to the improvement of accuracy of recognition systems.

A widely used approach to audio segmentation is based on the Bayesian Information Criterion (BIC) [1, 2, 3, 4, 5, 6]. In particular, in [5] an efficient approach to the shift variable-size window algorithm has been proposed. The input audio stream is progressively encoded by cumulative statistics, and the encoding is used to avoid redundant operations in the computation of BIC values.

In this work, the algorithm presented in [5] is analyzed in detail, from the viewpoint of computational cost, and compared, both in theory and experimentally, with two other possible approaches. One more direct but more expensive; and an original method that merges the good ideas of the other two. It will be shown that the last method is the most efficient.

2. BIC-BASED SEGMENTATION

Segmenting an audio stream means to detect the time indexes corresponding to changes in the nature of audio, in order to isolate segments that are acoustically homogeneous.

Briefly, given a sequence $o_1 \dots o_N$ of observation vectors in the $\Re^d$ space containing at most one change, the method [1] based on the BIC [7] for audio segmentation rests on the computation of:

$$\Delta BIC_i = \frac{N}{2}\log|\Sigma| - \frac{i}{2}\log|\Sigma_1| - \frac{(N-i)}{2}\log|\Sigma_2| - \lambda P \quad (1)$$

for each time index i, where $P = \frac{1}{2}\left(d + \frac{d(d+1)}{2}\right)\log(N)$, and Σ, Σ_1 and Σ_2 are the covariance matrixes estimated on $o_1 \dots o_N$, $o_1 \dots o_i$ and $o_{i+1} \dots o_N$, respectively. The value of i that maximizes ΔBIC_i is the most likely time index for a change and if $\Delta BIC_{i_{max}} > 0$ then i_{max} is confirmed to be a change. The sensitivity of the method can be tuned by adjusting the value λ to the particular task under consideration.

2.1. Multiple Spectral Changes Detection

In order to apply the above described method to an arbitrary large number of potential changes, we implemented the algorithm depicted in Figure 1, inspired by that proposed in [6]. The main idea is to have a shifting variable-size window for the computation of BIC values. Moreover, in order to save computations, BIC values are not computed for each observation within the window, but at a lower resolution rate, that is increased when a potential change is detected, to validate it and to refine its time position.

The main steps of the algorithm are:

Search start. ΔBIC values are computed only for the first N_{min} observations. N_{min} is the minimum size of the window, that has to be small to contain no more than one change, but large enough to allow computation of reliable statistics. Values are computed with low resolution δ_l, i.e. 1 observation out of 30. In order to have enough observations for computing both Σ_1 and Σ_2, ΔBIC are not computed for the N_{margin} indexes close to the left and right boundaries of the window.

Window growth. The window is enlarged by including ΔN_{grow} input observations until a change is detected, or a maximum size N_{max} is reached.

Window shift. The N_{max}-sized window is shifted on the right by ΔN_{shift} observations.

Change confirmation. If in one of the three previous steps a change is detected, ΔBIC values are re-computed with the high resolution δ_h, i.e. $\delta_h \approx \delta_l/5$, centering the window at the hypothesized change. The current size of the window is kept, unless it is larger than N_{second} observations, in which case it is narrowed to that value. If a change is detected again, it is output by the algorithm.

Window reset. After the change confirmation step, the algorithm has to go on resizing the analysis window to the minimum value N_{min} and locating it in a position dependent on the result of the confirmation step (see Figure 2).

This work was partially financed by the European Commission under the project FAME (IST-2000-29323).

```
init_window(1, N_min)

while(not end stream)

  (ΔBIC_{i_max}, i_max) ← compute_ΔBIC(δ_l)

  while(ΔBIC_{i_max} ≤ 0 &
        current_win_size < N_max &
        not end stream)
    growth_win(ΔN_grow)
    (ΔBIC_{i_max}, i_max) ← compute_ΔBIC(δ_l)

  while(ΔBIC_{i_max} ≤ 0 & not end stream)
    shift_win(ΔN_shift)
    (ΔBIC_{i_max}, i_max) ← compute_ΔBIC(δ_l)

  if(ΔBIC_{i_max} > 0) then
    center_win(i_max, min(current_win_size, N_second))
    (ΔBIC_{i_change}, i_change) ← compute_ΔBIC(δ_h)
    if(ΔBIC_{i_change} > 0) then
      output(i_change)
      init_window(i_change + 1, N_min)
    else
      init_window(i_max − Δ_margin + 1, N_min)
```

Fig. 1. Pseudocode of the multiple change detection algorithm.

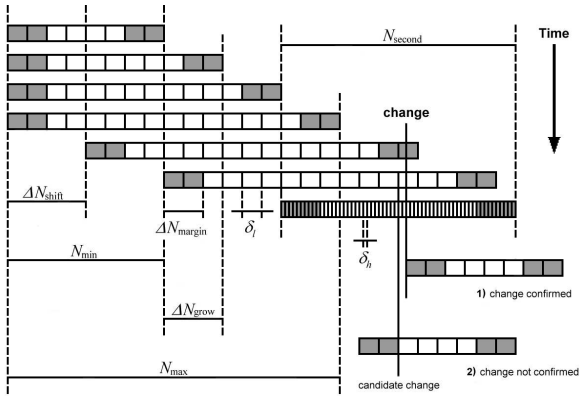

Fig. 2. The multiple change detection algorithm.

3. COMPUTATIONS

3.1. The Sum Approach (SA)

The evaluation of equation (1) determines the overall computational cost of the algorithm presented above, since a high number of ΔBIC values have to be computed for each window.

An efficient way to compute the determinant of the covariance matrix is based on the Cholesky decomposition which requires $O\left(d^3/6\right)$ operations. The estimation of the mean vector μ and of the covariance matrix Σ on N d-sized observations requires, respectively, $d(N+1)$ and $d(d+1)(N+1.5)$ operations. Typically, the window size N is significantly larger than the vector dimension d, hence the computational cost of the evaluation of the covariance matrix determinant is not relevant.

In order to reduce the computational cost of estimating likelihoods of Normal distributions, required for the computation of ΔBIC values, it is convenient to keep the sums of the input vectors (SV) and that of the square vectors (SQ):

N	current window size
n	index of the vector that preceeds the first of the window
T	set of vectors inside the window $\{o_{n+1} \dots o_{n+N}\}$
$\tilde{T}$	set of vectors inside the window after a growth or a shift
A_k	set of the first $k\delta$ vectors of the window $\{o_{n+1} \dots o_{n+k\delta}\}$
B_k	set of the last $n - k\delta$ vect. of the win. $\{o_{n+k\delta+1} \dots o_{n+N}\}$
SV_X	sum of vectors of the set X
SQ_X	sum of square vectors of the set X
Σ_X	covariance matrix on the vectors of the set X
μ_X	mean vector on the vectors of the set X

Table 1. Notation.

$$SV = \sum_{i=n+1}^{n+N} o_i \qquad SQ = \sum_{i=n+1}^{n+N} o_i \cdot o_i^t$$

In fact, besides allowing the easy computation of the needed parameters:

$$\mu = \frac{1}{N} \cdot SV \qquad \Sigma = \frac{1}{N} \cdot SQ - \mu \cdot \mu^t$$

the use of SV and SQ permits to avoid many redundant operations in the computation of ΔBIC values both within a given window and after a window growth/shift. With reference to the notation of Table 1, the following cases can happen:

- **growth** of the window by δ observations:
  - $SV_{\tilde{T}} = SV_T + \sum_{j=n+N+1}^{n+N+\delta} o_j$
  - $SQ_{\tilde{T}} = SQ_T + \sum_{j=n+N+1}^{n+N+\delta} o_j \cdot o_j^t$
- **shift** of the window by δ observations:
  - $SV_{\tilde{T}} = SV_T - \sum_{j=n+1}^{n+\delta} o_j + \sum_{j=n+N+1}^{n+N+\delta} o_j$
  - $SQ_{\tilde{T}} = SQ_T - \sum_{j=n+1}^{n+\delta} o_j \cdot o_j^t + \sum_{j=n+N+1}^{n+N+\delta} o_j \cdot o_j^t$
- **computation** of ΔBIC_i (at resolution δ):
  - $SV_{A_i} = SV_{A_{i-1}} + \sum_{j=n+(i-1)\delta}^{n+i\delta} o_j$
  - $SQ_{A_i} = SQ_{A_{i-1}} + \sum_{j=n+(i-1)\delta}^{n+i\delta} o_j \cdot o_j^t$
  - $SV_{B_i} = SV_T - SV_{A_i}$
  - $SQ_{B_i} = SQ_T - SQ_{A_i}$

With this approach, in each step of the algorithm the number of operations for computing equation (1) is:

- **growth** of the window by δ observations:

$$\underbrace{d(d+1) \cdot \delta}_{SQ_{\tilde{T}}} + \underbrace{d \cdot \delta}_{SV_{\tilde{T}}}$$

- **shift** of the window by δ observations:

$$\underbrace{d(d+1) \cdot 2 \cdot \delta}_{SQ_{\tilde{T}}} + \underbrace{d \cdot 2 \cdot \delta}_{SV_{\tilde{T}}}$$

- **computation** of the cov. matrix of the whole window:

$$\underbrace{d}_{\mu_T} + \underbrace{1.5 \cdot d(d+1)}_{\Sigma_T} \quad = \quad 1.5 \cdot d(d+1) + d$$

- **computation** of ΔBIC_i values with resolution δ ($\forall i,\ i = 1, \ldots, N/\delta - 1$):

$$\underbrace{d \cdot \delta}_{SV_{A_i}} + \underbrace{d(d+1) \cdot \delta}_{SQ_{A_i}} + \underbrace{d}_{SV_{B_i}} + \underbrace{d(d+1)/2}_{SQ_{B_i}} + \underbrace{2 \cdot d}_{\mu_{A_i}, \mu_{B_i}} + \\ + \underbrace{3 \cdot d(d+1)}_{\Sigma_{A_i}, \Sigma_{B_i}} = d(d+1) \cdot (\delta + 3.5) + d \cdot (\delta + 3)$$

3.2. The Distribution Approach (DA)

In order to further reduce the computational cost of the algorithm, it is possible to evaluate equation (1) through the approach proposed in [5].

Let Σ_N e μ_N be the sample covariance matrix and the mean of a set of N d-dimensional observations. If a (sub)set of Δ observations with covariance matrix Σ_Δ and mean vector μ_Δ has to be added or subtracted to that set, the parameters of the updated set of vectors can be computed by:

$$\Sigma_{N \pm \Delta} = \frac{N}{N \pm \Delta}\Sigma_N \pm \frac{\Delta}{N \pm \Delta}\Sigma_\Delta \pm \\ \pm \frac{N\Delta}{(N \pm \Delta)^2}(\mu_N - \mu_\Delta)(\mu_N - \mu_\Delta)^t \quad (2)$$

$$\mu_{N \pm \Delta} = \frac{N}{N \pm \Delta}\mu_N \pm \frac{\Delta}{N \pm \Delta}\mu_\Delta \quad (3)$$

This formulation requires only $3 \cdot d(d+1) + d$ and $3 \cdot d$ operations for computing $\Sigma_{N \pm \Delta}$ and $\mu_{N \pm \Delta}$, respectively, instead of $d(d+1)(N \pm \Delta + 1.5)$ and $d(N \pm \Delta + 1)$ required by the plain definitions.

The alternative approach consists in computing from the input audio stream $o_1, o_2, \ldots, o_{N_{audio}}$ the set of triples (Σ_1^n, μ_1^n, n), where $n = \delta_h, 2\delta_h, 3\delta_h, \ldots, N_{audio}$.

The keys of this processing are equations (2) and (3) which allow to obtain (Σ_1^n, μ_1^n, n) from $(\Sigma_1^{n-\delta_h}, \mu_1^{n-\delta_h}, n - \delta_h)$ and $(\Sigma_{n-\delta_h+1}^n, \mu_{n-\delta_h+1}^n, \delta_h)$, where $\Sigma_{n-\delta_h+1}^n$ and $\mu_{n-\delta_h+1}^n$ are computed directly from the vectors $o_{n-\delta_h+1}, o_{n-\delta_h+2}, \ldots, o_n$ through the definitions.

Since in this approach the estimation of a new distribution is based on already computed distributions, it will be referred with the name "distribution approach" (DA).

By constraining δ_l and N_{second} to be integers multiples of δ_h and by chosing N_{min}, N_{max}, ΔN_{grow}, ΔN_{shift}, ΔN_{margin} to be divisible by δ_l, it is possible to use the cumulative distributions (Σ_1^n, μ_1^n, n) for the evaluation of ΔBIC values and to reduce the cost of the computation. In fact, whatever the step of the algorithm is, the covariance matrixes required by equation (1) can be estimated by exploiting equation (2) (and (3)):

$$\Sigma_{n+1}^{n+N} = \frac{n+N}{N}\Sigma_1^{n+N} - \frac{n}{N}\Sigma_1^n - \frac{(n+N)n}{N^2} \times \\ \times \left(\mu_1^{n+N} - \mu_1^n\right)\left(\mu_1^{n+N} - \mu_1^n\right)^t \quad (4)$$

$$\Sigma_{n+1}^{n+i} = \frac{n+i}{i}\Sigma_1^{n+i} - \frac{n}{i}\Sigma_1^n - \frac{(n+i)n}{i^2} \times \\ \times \left(\mu_1^{n+i} - \mu_1^n\right)\left(\mu_1^{n+i} - \mu_1^n\right)^t \quad (5)$$

$$\Sigma_{n+i+1}^{n+N} = \frac{n+N}{N-i}\Sigma_1^{n+N} - \frac{n+i}{N-i}\Sigma_1^{n+i} - \frac{(n+N)(n+i)}{(N-i)^2} \times \\ \times \left(\mu_1^{n+N} - \mu_1^{n+i}\right)\left(\mu_1^{n+N} - \mu_1^{n+i}\right)^t \quad (6)$$

Of course, equation (4) is evaluated only once for a given window, while equations (5) and (6) have to be evaluated for each time index of interest (depending on the resolution). A scheme of the DA approach is given in Figure 3.

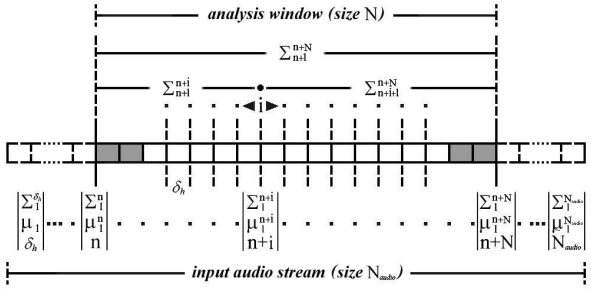

Fig. 3. ΔBIC_i computation in the DA approach.

The number of operations required by each step of the algorithm with the DA approach is:

- **growth** or **shift** of the window by δ observations:

$$\underbrace{d(\delta+1)}_{\mu_{n+N+1}^{n+N+\delta}} + \underbrace{d(d+1)(\delta+1.5)}_{\Sigma_{n+N+1}^{n+N+\delta}} + \underbrace{3 \cdot d(d+1) + 4 \cdot d}_{(\Sigma_1^{n+N+\delta}, \mu_1^{n+N+\delta}, n+N+\delta)} \\ = d(d+1)(\delta + 4.5) + d(\delta + 5)$$

  Note that this cost is that of the input stream encoding.

- **computation** of the cov. matrix of the whole window:

$$\underbrace{3 \cdot d(d+1) + d}_{\Sigma_{n+1}^{n+N}}$$

- **computation** of ΔBIC_i values with resolution δ ($\forall i,\ i = 1, \ldots, N/\delta - 1$):

$$\underbrace{6 \cdot d(d+1) + 2 \cdot d}_{\Sigma_{n+1}^{n+i}, \Sigma_{n+i+1}^{n+N}}$$

3.3. The Cumulative Sum approach (CSA)

In previous methods, the estimation of the statistics required for the computation of the BIC are based either on the use of the sum and square sum of input vectors that fall inside the analysis window, or on the use of the set of statistics computed only once, as the observations from the input stream are available. A combination of the two basic ideas gives the possibility to implement an even more efficient approach.

The idea is to encode the input stream, not through the distributions as in DA, but with the sums of the SA approach, that is with the sequence of triples (SQ_1^n, SV_1^n, n) computed at resolution δ_h. The higher efficiency is given by: (i) the redundant computations of the SA approach are avoided since each input vector is used only once, during the encoding of the input stream; (ii) the new encoding is cheaper than the DA encoding (cf. the grow/shift costs); (iii)

step	#operations
growth/shift	$d(d+1)\delta + d\delta$
Σ_{n+1}^{n+N}	$2d(d+1) + 2d$
ΔBIC_i	$4d(d+1) + 4d$

Table 2. Cost of each algorithm step with the CSA approach.

the computation of covariance matrixes from sums requires less operations than starting from other distributions.

The costs of each step of the algorithm of this new approach, that can be referred with the name "cumulative sum approach" (CSA), are reported in Table 2.

4. EXPERIMENTAL EVALUATION

For experiments, the test data was selected from the IBNC corpus [8], a collection of radio news programs. The performance was measured on recordings of 6 programs (about 75 minutes of audio) where 212 changes were manually annotated.

4.1. Costs comparison

Since the computation of ΔBIC_i values is done $N/\delta - 1$ times in each window, the total cost of the algorithm mainly depend on the cost of that operation, and this is the reason for which the DA approach is convenient with respect to the SA approach; in fact, the number of operations with the DA approach does not depend on δ, unlike the SA does, and in our case ($d = 13$) it results convenient for $\delta \geq 3$.

In order to validate the theoretical comparison done in Subsections 3.1, 3.2 and 3.3, and in particular the dependence of the overall computational cost from the resolution δ, the three approaches have been run with a simplified setup. We set $N_{min} = N_{max}$, in order to eliminate the window grow step, and the value of λ was set high enough that no candidate change was detected, constraining the computations to be done only at resolution $\delta_l = \delta$.

Given the setup in the caption, the total number of operations required by the three approaches are given in the columns "#operations"[1] of Table 3, for different values of δ. The execution times were measured on a Pentium III 600MHz on the 75-minute test set.

	#operations[1]			execution time		
	SA	DA/SA	CSA/SA	SA	DA/SA	CSA/SA
δ	($\cdot 10^6$)	%	%	(s)	%	%
1	380.6	123.9	92.1	887.9	103.8	94.1
5	127.1	80.1	61.5	290.4	68.4	62.6
10	95.4	59.0	46.3	215.5	50.2	46.2
25	76.4	37.4	31.0	168.9	32.0	30.0

Table 3. Theoretical and experimental costs comparison of SA, DA and CSA approaches. Setup: $N_{audio} = 50000,\ N_{min} = N_{max} = 1000,\ \Delta N_{shift} = 200,\ \Delta N_{margin} = 50,\ d = 13$.

Finally, the comparison of the three approaches is made on the best setup of the algorithm. Results in terms of geometric mean (*F-score*) of *precision* and *recall* of change detection is reported in Table 4, together with execution times. The slight difference in change detection score is due to some minor differences in the implementations. For what concerns the execution times, since δ_l was set to 25, the ratio between the costs of the three implementations expected from the results of Table 3 is confirmed.

[1]The values include the cost of the computation of the covariance matrixes determinant ($d^3/6$).

	F-score	execution time (s)	% vs. SA
SA	88.4	309.3	—
DA	89.4	108.0	34.9
CSA	89.4	91.2	29.5

Table 4. Performance comparison of SA, DA and CSA approaches in their best setup: $N_{min} = 500,\ N_{max} = 2000,\ N_{second} = 1500,\ \Delta N_{grow} = 100,\ \Delta N_{shift} = 300,\ \Delta N_{margin} = 100,\ \delta_l = 25,\ \delta_h = 5,\ \lambda = 2.175$.

5. CONCLUSIONS

In this work three different approaches to the implementation of a widely adopted BIC-based audio segmentation algorithm have been analyzed: (i) a simple method that uses only a sum and a square sum of the input vectors, in order to save computations in estimating covariance matrixes on partially shared data; (ii) the approach proposed in [5] that encodes the input signal with cumulative distributions; and (iii) an original approach that encodes the input signal in cumulative pair of sums. The two latter approaches exploit the typical approximation made in that algorithm, that is the use of a resolution lower than 1 for change detection.

The three approaches have been compared both theoretically and experimentally, and the proposed original approach has been shown to be the most efficient.

6. REFERENCES

[1] S. S. Chen and P. S. Gopalakrishnan, "Speaker, environment and channel change detection and clustering via the Bayesian Information Criterion," in *Proc. of the DARPA Broadcast News Transcr. & Underst. Workshop*, Lansdowne, VA, 1998.

[2] M. Harris, X. Aubert, R. Haeb-Umbach, and P. Beyerlein, "A study of broadcast news audio stream segmentation and segment clustering," in *Proc. EUROSPEECH*, Budapest, Hungary, 1999, vol. 3, pp. 1027–1030.

[3] A. Tritschler and R. Gopinath, "Improved speaker segmentation and segments clustering using the Bayesian Information Criterion," in *Proc. EUROSPEECH*, Budapest, Hungary, 1999, vol. 2, pp. 679–682.

[4] M. Cettolo, "Segmentation, classification and clustering of an Italian broadcast news corpus," in *Proc. of the 6th RIAO - Content-Based Multimedia Information Access - conference*, Paris, France, 2000.

[5] P. Sivakumaran, J. Fortuna, and A. M. Ariyaeeinia, "On the use of the Bayesian Information Criterion in multiple speaker detection," in *Proc. EUROSPEECH*, Aalborg, Denmark, 2001, vol. 2, pp. 795–798.

[6] P. Delacourt, D. Kryze, and C.J. Wellekens, "Speaker-based segmentation for audio data indexing," in *Proc. of the ESCA ETRW workshop Accessing Information in Spoken Audio*, Cambridge, UK, 1999.

[7] G. Schwarz, "Estimating the dimension of a model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[8] M. Federico, D. Giordani, and P. Coletti, "Development and evaluation of an Italian broadcast news corpus," in *Proc. of the 2nd Int. Conf. on Language Resources and Evaluation (LREC)*, Athens, Greece, 2000.