ON THE USE OF THE BAYESIAN INFORMATION CRITERION IN MULTIPLE SPEAKER DETECTION

P.Sivakumaran, J. Fortuna and A. M. Ariyaeeinia

University of Hertfordshire,
Hatfield, Hertfordshire, AL10 9AB, UK

{p.sivakumaran, j.m.r.c.fortuna, a.m.ariyaeeinia}@herts.ac.uk

Abstract

An efficient scheme, based on the Bayesian information criterion (BIC), for the detection of speaker changes in an audio stream is introduced and investigated. BIC has been the subject of considerable attention in recent years due to its effectiveness for speaker change detection (SCD) as well as the detection of other forms of acoustic changes. A main difficulty in BIC-based SCD has been reported to be that of the computational complexity. The scheme proposed here tackles this problem by reducing the computational load in the previously proposed algorithms significantly, without compromising their effectiveness. The paper describes the new scheme thoroughly and analyses its performance. Experiments are based on 3 hours of broadcast news with 416 speaker changes. With this data, the proposed scheme has been found to be capable of running in about 0.06 times real-time whilst keeping the rate of each of misdetection and false alarm close to 9%.

1. Introduction

Detecting speaker changes in a given audio stream has received a great deal of interest in recent years [1-5]. This is mainly due to its various potential applications ranging from retrieving information from audio materials to improving the accuracy of speech recognition systems. If a priori knowledge of the acoustic information on the speakers in the audio stream is available, then speaker change detection (SCD) may be formulated as a maximum likelihood classification problem. However, in many practical applications, this is not the case and thus other approaches are required. The most primitive technique for SCD is to assume that a speaker change is likely to occur at a silence and to use the energy features to detect the silences in the audio stream. If it is chosen to use silences as the basis for SCD then utilising a continuous speech recogniser as the front-end unit may provide a more effective solution in some applications.

A more formal approach to SCD is to assume that a speaker change is likely to occur at non-speech events such as a silence, music, laughter, breathing or lip-smack, and to statistically model both speech and non-speech events separately (for example, using Gaussian mixture models) [1]. In this case, the speaker changes can be determined through a maximum likelihood classification. The fundamental difficulty in such an approach is the mismatch in the acoustic conditions of the material used for training the statistical models and the audio stream being analysed.

This problem can obviously be avoided, if SCD is performed solely based on the information contained in the audio stream being analysed. The pioneering work in this area has been done by Gish *et al.* [6]. The approach is based on using a sliding window through the audio stream and measuring the similarity score between each two adjacent windows. If the similarity score falls below a threshold then a speaker change is registered. Various other similarity measures have already been proposed in the literature for this purpose [2].

The Bayesian information criterion (BIC)-based approach proposed in [3] uses a firm mathematical foundation to detect speaker changes (which are predominantly the most significant acoustic changes) using only the information contained in the audio stream. This paper is mainly concerned with this method and is organised in the following manner. The next section provides a general review of the theory behind BIC-based SCD. Section 3 discusses how BIC can be used to detect multiple speaker changes. In this section an effective scheme for reducing the computational complexity of two well known BIC-based algorithms is introduced. The experimental work and results are detailed in Section 4, and the overall conclusions are presented in Section 5.

2. SCD via BIC

In this approach, a given set of N, d-dimensional, acoustic vectors, $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, is modelled in two ways. The first model consists of representing the entire vector set using a single-Gaussian density. The second model is based on dividing the given set of N vectors into two subsets of $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_i\}$ and $\{\mathbf{x}_{i+1}, \mathbf{x}_{i+2}, ..., \mathbf{x}_N\}$, and then representing each of these using a different single-Gaussian density. The first model is expected to best fit the data when it belongs entirely to a single speaker, whilst the second model provides the best description of the data when i is the point of speaker change. Thus, the problem of detecting a point of speaker change can be addressed by quantifying the likelihood of the second model in relation to the first one. For this purpose, the following formulation which is based on BIC can be adopted [3].

$$\Delta BIC_i = R(i) - \lambda\big(d + 0.5d(d+1)\big)\log N \tag{1}$$

where $R(i) = N\log\left|\mathbf{C}_1^N\right| - i\log\left|\mathbf{C}_1^i\right| - (N-i)\log\left|\mathbf{C}_{i+1}^N\right|$ and (2)

$\mathbf{C}_a^b$ is the covariance matrix estimated using the d-dimensional acoustic vectors $\mathbf{x}_a$, $\mathbf{x}_{a+1}$,…,$\mathbf{x}_b$ and λ is a factor which is independent of i. If ΔBIC_i is known for i = 1,2,…,N then i_{max}, the value of i that maximises ΔBIC_i, is the most likely point for a speaker turn. If λ is set appropriately, then $\Delta BIC_{i_{max}}$ has a positive value which confirms the presumed speaker change. The underlying assumptions here are that firstly, there is no more than one speaker turn in the considered vector set and secondly, the most significant acoustic change in the data is produced by a speaker turn.

Of course, in order for the above scheme to be useful in practical applications, it has to be extended to detect multiple speaker changes. For this purpose, various algorithms have been proposed in the literature [2-5]. The main focal point of these algorithms has been the computational burden in estimating *ΔBIC* at all possible points. In this paper a new scheme is proposed which is capable of reducing the computational cost involved in the previously suggested efficient algorithms [4-5] significantly, without affecting the effectiveness of them. A description of this new scheme is provided in the next section following a discussion on the algorithms it builds on.

3. Multiple speaker change detection

The algorithms proposed in [4-5] for multiple speaker change detection are based on using a shifting, variable size window in the computation of *ΔBIC* values. With reference to Figure 1, the main phases of these algorithms can be described as follows.

1. Initial search*:* *ΔBIC* values are estimated in a window that covers the first $N_{\min}$ vectors in the incoming audio stream. Here, a low-resolution rate, δ_l, (for example, 1 for every 50-vector interval) is chosen for the computation.

2. Grow*:* If no speaker turn is revealed in the initial phase, then the window size is grown to include the next ΔN_g vectors in the audio stream and *ΔBIC* values are evaluated, again with the low-resolution rate δ_l. This step is repeated until the window size is reached a predefined value $N_{\max}$ or a speaker change is detected.

3. Shift: If no speaker turn is revealed in the second phase, *ΔBIC* is re-estimated after shifting the window by ΔN_s vectors while maintaining its current size. This step is continued until a speaker turn is detected.

4. Confirming Speaker Changes*:* Once a speaker change is detected, *ΔBIC* is re-computed with a higher resolution rate, δ_h ($\approx \delta_l/5$) in a window of N_{scd} ($\leq N_{\min}$) vectors centred on the point of the speaker turn. The point of speaker change determined via this refinement is then stored.

5. Re-initialisation*:* After a speaker change detection**,** the window size is re-set to its original value $N_{\min}$ and located just after the point of confirmed speaker change. This creates a condition which is similar to that in the initial point of the algorithm. Therefore, by appropriately repeating the above procedures the second speaker change can be detected. In this way, the entire audio stream is processed to determine all the points of speaker turns.

The purpose of increasing the size of window from $N_{\min}$ in steps of ΔN_g is to ensure that no more than one speaker turn is in the set, and to use as much data as possible in the evaluation of *ΔBIC*. Since the computational cost is proportional to this enlargement, the maximum size of the window is limited to $N_{\max}$. The computational burden also reduces by not evaluating *ΔBIC* at all possible points. Initially, *ΔBIC* is computed at a resolution rate of δ_l, which introduces an uncertainty of δ_l in detecting a speaker turn. In order to reduce this uncertainty, *ΔBIC* is re-estimated with a higher resolution rate of δ_h around each presumed point of speaker change. For the purpose of this paper the above procedure is referred to as ***BIC-SCD***.

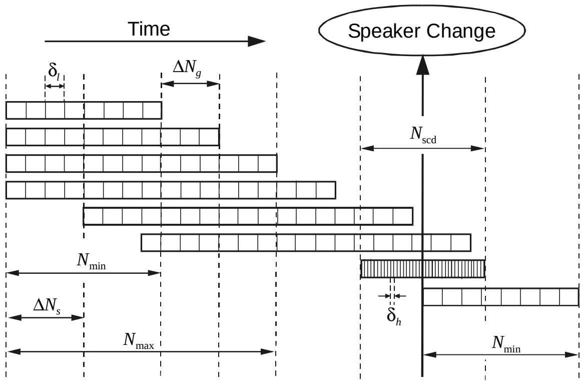

Figure 1: Procedure for detecting multiple speaker changes.

In the above procedure, for each analysis window, first the covariance matrix of the entire set of vectors has to be estimated and its determinant computed. Then, at each point of *ΔBIC* evaluation, two covariance matrices must be determined and their determinants evaluated. It is well known that the most efficient method to compute the determinant of a $d \times d$ covariance matrix (which is symmetric and positive definite) is based on Cholesky decomposition [7]. This method requires $O(d^3/6)$ operations, where O stands for *in the order of.*

Estimating the covariance matrix of a set of N, d-dimensional, vectors requires $O\{0.5d(d+1)N\}$ operations. Since the window size in the evaluation of *ΔBIC* is significantly larger than the vector dimension, the net computation involved in estimating every covariance matrix from the raw acoustic vectors surpasses that of the determinant evaluation and predominates the overall computation. In order to overcome this problem, an effective scheme is proposed here in which the total number of operations needed to compute all the required covariance matrices is kept $O\{0.5d(d+1)[8N_{\Delta BIC} + (1+4\delta^{-1})N_{aud}]\}$, where $N_{\Delta BIC}$ is the total number of *ΔBIC* evaluations and N_{aud} is the number of acoustic vectors in the audio stream to be analysed. This reduces computational cost by a factor of about $\{3N_{ave}(d + 24)^{-1}\}$, where N_{ave} is the average frame size used in evaluating *ΔBIC* values (for example, if $N_{ave} = 300$ and $d = 12$, then the computational load is reduces by approximately 25 times). The details of this approach are given in the following sub-section.

3.1. Proposed approach

Suppose that the covariance matrix and the mean vector of a set of N, d-dimensional, vectors are determined to be $\mathbf{C}_N$ and $\boldsymbol{\mu}_N$ respectively. If a subset of vectors with the covariance matrix $\mathbf{C}_\Delta$ and the mean vector $\boldsymbol{\mu}_\Delta$ is added to or removed from this set, then it can be easily shown that the covariance and the mean of the resulting vector set can be estimated in the following manner.

$$\mathbf{C}_{N\pm\Delta} = \alpha\mathbf{C}_N \pm \beta\mathbf{C}_\Delta \pm \gamma(\boldsymbol{\mu}_N - \boldsymbol{\mu}_\Delta)(\boldsymbol{\mu}_N - \boldsymbol{\mu}_\Delta)^t \,, \tag{3}$$

$$\boldsymbol{\mu}_{N\pm\Delta} = \eta N\boldsymbol{\mu}_N \pm \eta\Delta\boldsymbol{\mu}_\Delta \,, \tag{4}$$

where $\alpha = (N-1)\rho,\ \beta = (\Delta-1)\rho,\ \gamma = \eta N\Delta\rho,$ $\eta^{-1} = (N \pm \Delta)$ and $\rho^{-1} = (N \pm \Delta - 1)$. The key point in the above formulation is that the computation of the covariance matrix $\mathbf{C}_{N\pm\Delta}$ requires $O\{2d(d+1)\}$ operations, which otherwise would have taken $O\{0.5d(d+1)(N\pm\Delta)\}$ operations. It immediately follows that the *ΔBIC* formula can be modified as follows to achieve a significant saving in the computation.

$$\Delta BIC_i = \begin{cases} \Omega - i\log\left|\mathbf{C}_1^i\right| - (N-i)\log\left|\mathbf{D}_1^i\right| & i \le N/2 \\ \Omega - i\log\left|\mathbf{D}_{i+1}^N\right| - (N-i)\log\left|\mathbf{C}_{i+1}^N\right| & i > N/2 \end{cases} \qquad (5)$$

where

$$\eta\mathbf{D}_a^b = (N-1)\mathbf{C}_1^N - (b-a)\mathbf{C}_a^b - \gamma(\boldsymbol{\mu}_1^N - \boldsymbol{\mu}_a^b)(\boldsymbol{\mu}_1^N - \boldsymbol{\mu}_a^b)^t\ , \qquad (6)$$

$$\Omega = \left|\mathbf{C}_1^N\right| - \lambda(d + 0.5d(d+1)\log N\ . \qquad (7)$$

In equation (5), $\eta = N + a - b - 2$, $\gamma = N(b-a+1)/(\eta+1)$, and $\boldsymbol{\mu}_1^N$ and $\boldsymbol{\mu}_a^b$ are vector means used in the estimation of $\mathbf{C}_1^N$ and $\mathbf{C}_a^b$ respectively. It is evident that, in a given analysis window, if the covariance matrix of the entire vector set is estimated and the associated mean vector is stored then, at each point of *ΔBIC* evaluation, a single covariance matrix has to be estimated from a set of *n* (≤ *N*/2) raw vectors and the associated mean has to be stored. In this case, the reduction in computation is $\mathrm{O}\left(0.5d(d+1)\sum_i |0.5N - i|\right)$ operations. However, it is possible to use equations (3) and (4) in another, and more effective, way to reduce the computational burden in the evaluation of *ΔBIC* further.

The approach involves encoding the given audio vector stream, $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{N_{aud}}$, into $\left\{\mathbf{A}_1^n, \mathbf{m}_1^n, n\right\}$ at intervals of δ_h vectors, where $\mathbf{A}_1^n$ and $\mathbf{m}_1^n$ are the covariance matrix and the mean vector of the vectors $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$. The key to this encoding process is the recursion implied by equations (3) and (4), in which $\left\{\mathbf{A}_1^n, \mathbf{m}_1^n\right\}$ is determined by using $\left\{\mathbf{A}_1^{n-\delta_h}, \mathbf{m}_1^{n-\delta_h}\right\}$ and $\left\{\mathbf{A}_{n-\delta_l+1}^n, \mathbf{m}_{n-\delta_l+1}^n\right\}$. Since $\mathbf{A}_{n-\delta_l+1}^n$ has to be estimated from the raw vectors, in each encoding instant, the computational load is $\mathrm{O}\{0.5d(d+1)(\delta_h + 4)\}$ operations. The total computational load of the encoding process is therefore $\mathrm{O}\{0.5N_{aud}d(d+1)\left(1 + 4\delta_h^{-1}\right)\}$ operations.

Suppose that both δ_l and N_{scd} are set to be integer multiples of δ_h, and N_{min}, ΔN_g, N_{max} and ΔN_s are chosen to be divisible by δ_l. With the encoded stream, the computational cost of estimating the covariance matrices at each point of *ΔBIC* evaluation becomes negligibly small. This is because, regardless of the phase of the algorithm, the required covariance matrices can be determined in the following manner.

$$\mathbf{C}_1^N = \alpha_1\mathbf{A}_1^{n+N} - \beta_1\mathbf{A}_1^n - \gamma_1(\mathbf{m}_1^{n+N} - \mathbf{m}_1^n)(\mathbf{m}_1^{n+N} - \mathbf{m}_1^n)^t \qquad (8)$$

$$\mathbf{C}_1^i = \alpha_2\mathbf{A}_1^{n+i} - \beta_2\mathbf{A}_1^n - \gamma_2(\mathbf{m}_1^{n+i} - \mathbf{m}_1^n)(\mathbf{m}_1^{n+i} - \mathbf{m}_1^n)^t \qquad (9)$$

$$\mathbf{C}_{i+1}^N = \alpha_3\mathbf{A}_1^{n+N} - \beta_3\mathbf{A}_1^{n+i} - \gamma_3(\mathbf{m}_1^{n+N} - \mathbf{m}_1^{n+i})(\mathbf{m}_1^{n+N} - \mathbf{m}_1^{n+i})^t \quad (10)$$

where $N \in \left\{N_{\mathrm{min}}, N_{\mathrm{max}}, N_{\mathrm{scd}}\right\}$ is the size of the current window, *n* is a time instant in the audio stream where the current window's lower boundary is positioned, and $\left\{\alpha_p, \beta_p, \gamma_p\right\}$ for *p* = 1,2,3 are the terms which depend on the number of vectors used to estimate the covariance matrices in the associated equations and have similar forms to those defined for equation (2). Obviously, equation (8) has to be evaluated only once in a given window. The operations required to evaluate equations (9) and (10) at any point of the *ΔBIC* estimation is only $\mathrm{O}\{4d(d+1)\}$. It is evident that this procedure reduces the computational complexity extensively without affecting accuracy of the algorithm in any way. It should be pointed out that applying this procedure requires neither all the sets of the encoded parameters to be stored nor the end of audio stream to be known. In theory, the maximum number of encoded parameter sets to be memorised is $(N_{\mathrm{max}} + 0.5N_{\mathrm{scd}} - \delta_l)/\delta_h$ and therefore the procedure can be efficiently implemented as an *in-place* algorithm by using such structures as the circular buffer (Figure 2) for the purpose of live audio analysis.

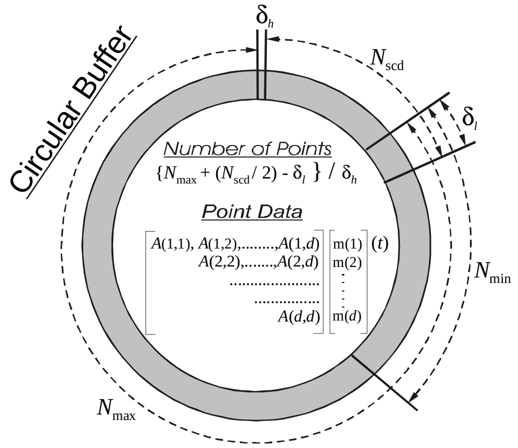

Figure 2: Possible implementation of the proposed scheme.

It should also be noted that the reduction in the computational complexity can virtually be $\mathrm{O}\{0.5d(d+1)N_{aud}\}$ operations, if the audio stream is encoded at intervals of δ_l vectors. The disadvantage, however, is that in the speaker change-confirmation phase, for most of *ΔBIC* evaluations, one covariance matrix will have to be estimated from the raw vectors. This method may be preferred in the situations where the audio stream contains relatively few speaker changes. On the other hand, encoding at a higher resolution is highly efficient when a large number of speaker changes have to be detected. Additionally, it can provide the possibility of devising more accurate methods for confirming speaker changes with less computational cost. In the remainder of this paper the proposed scheme is referred to as ***FastBIC-SCD***.

4. Experimental Investigation and Results

The data used for the experimental work consisted of 3 hours of audio material, sampled at a rate of 32 kHz. This data was collected from the BBC News 24 broadcast service and included 416 speaker changes. These speaker changes were identified and manually labelled prior to the experiments. The type of feature parameters considered for this study was linear predictive coding (LPC)-derived cepstrum (LPCC). The extraction of these feature parameters was based on segmenting the audio data into 32 ms frames at intervals of 16 ms. Ideally, the experimental evaluation of the effectiveness of a BIC-based SCD procedure should be expressed in terms of a plot of λ versus each of the misdetection and false alarm rates. However, since the study involved exceeding number of experiments, it was decided to present the individual results in terms of the equal error rate (EER), i.e. the equal rates of misdetection and false alarm obtained by setting the value of λ appropriately.

The aim of the first set of experiments was to examine the computational efficiency of the proposed scheme. The experimental work was conducted on a Pentium™ III 600 MHz -based system, and the linear predictive (LP) analysis order was arbitrarily set to 12. This investigation showed that the time taken to analyse the parameterised audio data by BIC-SCD and *Fast*BIC-SCD methods were 96 and 5 minutes respectively. These results provide a clear confirmation of the effectiveness of the proposed scheme in reducing the analysis time significantly. The resultant EER in this study was about 14% which was obtained by setting λ to 2.8.

In the course of the above investigation it was noted that the time taken to parameterise the audio material was significantly longer than that taken by *Fast*BIC-SCD for processing the parameterised data. This was thought to be due the fact that the use of a sampling frequency of 32 kHz had resulted in exceedingly large number of data samples. This led to the conclusion that the overall computational time could be considerably reduced by using a lower sampling rate such as 8 kHz. However, the question was whether such a reduction in the sampling rate would adversely affect the performance of SCD and, if so, to what extent.

In order to examine the effects of reducing the sampling rate, it was decided to repeat the experiments using down-sampled versions of the original audio data. The sampling frequencies considered for this purpose were those commonly used in the audio processing (Figure 3). Since the choice of LP analysis order depends, to some extent, on the sampling rate [8], for each considered sampling rate, the experiments were repeated with different LP analysis orders. The results of these experiments are presented in Figure 3.

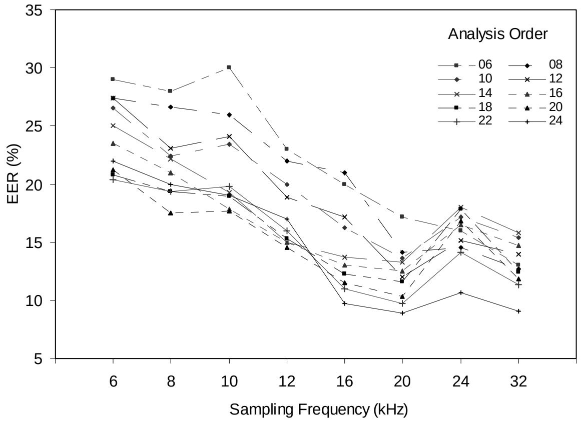

Figure 3: Dependence of SCD on the sampling rate.

A general conclusion drawn from the above results is that a sampling rate of 20 kHz is the best choice for SCD. For this sampling rate, Figure 4 gives plots of the processing time, EER and λ versus the LP analysis order. It is seen that λ reduces from 3.3 to 1.6 as the LP analysis order is increased from 6 to 24. Previous experimental results, however, showed that for a fixed LP analysis order, the variation of λ with the sampling rate was not as significant. It is also observed in Figure 4 that the processing speed tends to be inversely related to the LP analysis order. This trend is, of course, in agreement with all BIC-based SCD algorithms. Figure 4 also shows that by increasing the LP analysis order, the EER can be reduced from about 17% to around 9% at the expense of increasing the processing time from 1.7 to 11.3 minutes.

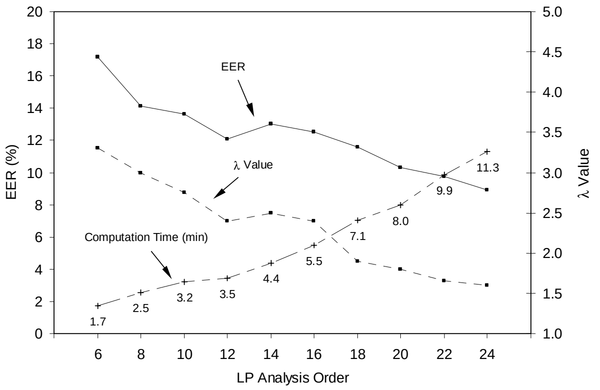

Figure 4: Effects of the LP order on the SCD performance.

5. Conclusions

The computational complexity of the previously suggested BIC-based SCD algorithms has been tackled by the introduction of a new scheme. The proposed approach achieves its efficiency by significantly reducing the number of operations involved in the estimation of the required covariance matrices, and without affecting the SCD accuracy. The experiments with the proposed scheme were conducted using a 3-hour audio stream which included 416 speaker changes. The experimental results have indicated that a sampling rate of 20 kHz is the best choice for BIC-based SCD. With this rate, it has been demonstrated that the entire parameterised audio data can be processed in about 11 minutes while keeping the resultant EER close to 9%. It has also been shown that the processing time can be reduced to as low as 1.7 minutes at the expense of the system accuracy. Furthermore, it has been observed that λ needed for obtaining the EER is considerably more dependent on the LP analysis order than on the sampling rate.

6. References

[1] Liu, D., Kubala, F., “Fast Speaker Change Detection for Broadcast News Transcription and Indexing”, *Proc. of Eurospeech’99*, 1031 - 1034.

[2] Delacourt, P., Wellekens, C., “DISTBIC: A Speaker-based Segmentation for Audio Data Indexing”, *Speech Communication*, Vol. 32, 111 – 126, 2000.

[3] Chen, S., Gopalakrishnan, P., “Speaker, Environment and Channel Change Detection and Clustering via The Bayesian Information Criterion”, *Proc. Broadcast News Trans. & Under. Workshop*, Vol. 6, 127-132, 1998.

[4] Tritschler, A., Gopinath, R., “Improved Speaker Segmentation and Segments Clustering Using the Bayesian Information Criterion”, *in Proc. Eurospeech’99*.

[5] Cettolo, M., “Segmentation, Classification and Clustering of an Italian Broadcast News Corpus”, *Proc. RIAO’2000*.

[6] Gish H., *et al*., "Segregation of Speakers for Speech Recognition and Speaker Identification", *Proc. ICASSP’91*, pp. 873-876.

[7] William H. P. *et al*. *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, 1992.

[8] Markel J. D. and Gray A. H., Linear *Prediction of Speech*, Springer-Verlag, New York, 1976.