

# Real-Time Unsupervised Speaker Change Detection

Lie Lu, Hong-Jiang Zhang  
Microsoft Research Asia  
Haidian District, Beijing, China  
{llu, hjzhang}@microsoft.com

## Abstract

The information of speaker change point is very useful for speaker tracking and other applications. In this paper, we presented an effective algorithm for automatic speaker change detection based on LSP correlation analysis. Moreover, a general case is considered, in which the speaker and speaker number are both assumed unknown. The algorithm has low complexity and can be processed in real-time with a limit delay. Our experiments have shown the algorithms produce very satisfactory results.

## 1. Introduction

There were many works on speaker recognition, which includes speaker identification and verification [1][7]. In these systems, it is supposed that the input speech belongs to one of the known speakers. However, in many cases, such as in a dialog or a meeting, speech stream is continuous and there is no information about the beginning and end of a speaker. So, it is necessary to find speaker change points in such cases if one need to classify the speeches according to speakers. This is a pre-processing step for speaker tracking and is also useful in many other applications, such as video content analysis.

Different from general speaker identification or verification problem, we assumed no pre-knowledge about the number and the identities of speakers in an audio stream. If the speaker is registered firstly, traditional speaker identification algorithm can be used, just as the work of J.N.L Brummer [2] has done. However, in many cases, the knowledge of speakers is often not available or difficult to obtain. Even in the well structured news broadcasting, we cannot assume that the anchorpersons are always the same. Therefore, it is desirable to perform an unsupervised speaker change detection algorithm in audio analysis.

There are several reported works on unsupervised speaker identification. In the work of Sugiyama [3], a simpler case was studied, in which the number of the speakers to be clustered was assumed to be known. VQ and HMM (Hidden Markov Model) are used in the implementation. The algorithm proposed by Wilcox [4] is also based on HMM segmentation, in which an agglomerative clustering method is used when the pre-knowledge of speakers is unknown. Siu [5] proposed a system to separate controller speech and pilot speech with GMM model. Speaker discrimination from the telephone speeches was studied in [6] using HMM segmentation. However, in this system, the number of speakers was limited to two. K. Mori [8] addresses the problem of detection of speaker changes and speaker

clustering with no *a priori* speaker information available. Chen [9] also presented an approach to detecting changes in speaker identity, environmental condition and channel conditions using Bayesian information criteria. An accuracy of about 80% is reported.

Previous efforts to solve the problem of unsupervised speaker clustering consist of clustering audio segments into homogeneous clusters according to speaker identity, background conditions or channel conditions. Most of the methods based on VQ, GMM or HMM model. A defect of these models is that iterative operation is unavoidable which makes these algorithms very time consuming, thus cannot be processed in real time.

In this paper, we present a simple, yet effective and robust method based on LSP correlation analysis to detect speaker change points in real time. Also, a general case is considered, in which the speaker and speaker number are both assumed to be unknown.

The rest of the paper is organized as follows. Section 2 shows the framework and details of the real time speaker change detection algorithm. In Section 3, experiments and the evaluations of the proposed algorithms are given.

## 2. Speaker Change Detection

The flow diagram of our real time unsupervised speaker change detection algorithm is illustrated in the Fig. 1. The system is mainly composed of three modules. They are front-end process module, segmentation module, and the module for clustering and updating speaker model.

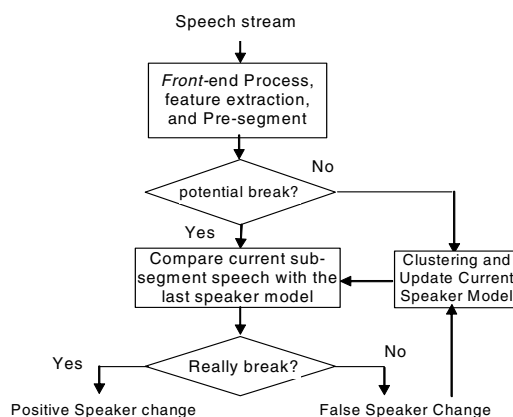


Fig. 1 Flow diagram for real-time speaker change detection algorithm

The input speech stream is first segmented into 3-second sub-segments with 2-second overlapping. Each sub-segment is pre-processed by removing silence and unvoiced frame. The *LSP* divergence distance between every consecutive two sub-segments is examined. A potential speaker change point is detected if the distance is above a threshold. Otherwise, the model of current speaker is updated incrementally by utilizing the data of current sub-segment. If a potential speaker change boundary is detected, Bayesian Information Criterion (BIC) is used to ensure if it is really a speaker change boundary.

## 2.1 Front-end Processing

The input audio stream is first down-sampled into 8KHZ, 16bits, mono channel and pre-emphasized. The speech stream is then divided into small sub-segments by 3-second window with 2-second overlapping. That is, the temporal resolution of the segmentation is one second. The sub-segment is further divided into non-overlapping 25ms-long frames. The most important feature extracted from the frame is *LSP* vector. *LSP* has discrimination for different speakers, base on the work [1]. Other features extracted include short-time energy and zero-crossing rate, they are used to discriminate silence frames and unvoiced frames, which should be excluded when estimate speaker model.

## 2.2 Detect potential speaker change point

At this step, speaker model is extracted for each sub-segment. Supposing that *LSP* satisfies Gaussian distribution, the speaker model for *i*-th sub-segment can be represented as  $N(C_i, u_i)$ . *K-L* distance is used to measure the dissimilarity between each two neighboring sub-segments at each time slot, as shown in Fig. 2 (a). *K-L* distance between *i*-th and *j*-th sub-segment is defined as,

$$D(i, j) = \frac{1}{2} \text{tr}[(C_i - C_j)(C_i^{-1} - C_j^{-1})] + \frac{1}{2} \text{tr}[(C_i^{-1} + C_j^{-1})(u_i - u_j)(u_i - u_j)^T] \quad (1)$$

The distance is composed of two parts. The first part is determined by the covariance of two segments and the second is determined by covariance and mean. Because the mean is easily biased by different environment condition, we will not consider the second part and only the first part is used to represent the distance, similar to the work [1]. It is also similar to the Cepstral Mean Subtraction (*CMS*) method used in speaker recognition to compensate the effect of environment conditions or transmission channels. It is called divergence shape distance here, which is defined by,

$$D(i, j) = \text{tr}[(C_i - C_j)(C_i^{-1} - C_j^{-1})] \quad (2)$$

Thus, a potential speaker change is found between *i*th and (*i*+1)th second, if the following conditions are satisfied:

$$D(i, i+1) > D(i+1, i+2), \quad D(i, i+1) > D(i-1, i), \quad D(i, i+1) > Th_i \quad (3)$$

where  $Th_i$  is a threshold.

The first two conditions guarantee a local peak exists, and the last condition can prevent very low peaks from being detected. Reasonable results can be achieved by using this simple criterion. However the threshold is affected by many factors and it is difficult to set. If the threshold is too small, false detection would be many; otherwise, some positive speaker change boundaries would be missed. To obtain a robust threshold, an automatic threshold setting method has been proposed as following.

The threshold is automatically set according to the previous *N* successive distances. That is,

$$Th_i = \alpha \cdot \frac{1}{N} \sum_{n=0}^N D(i-n-1, i-n) \quad (4)$$

where *N* is the number of previous distances used for predicting threshold, and  $\alpha$  is a coefficient as amplifier. We set  $\alpha = 1.2$  in our algorithm. The threshold determined in this way works well in different conditions, but false detections still exist. This is because there is no sufficient data to estimate the speaker model accurately from only one short speech sub-segment, such that the estimated speaker model would be biased.

In order to solve this problem, we should use as much data as possible to update speaker model. A more accurate refinement method has also been proposed to refine the above results.

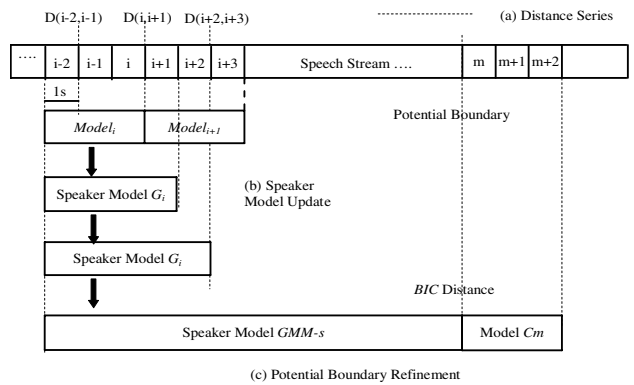


Fig. 2 An illustration of speaker change detection algorithm

## 2.3 Updating speaker model incrementally

In order to get as much data as possible to estimate speaker model more accurately, we utilize the results of potential speaker change detection. If no potential speaker change point is detected, it means the next sub-segment is as the same speaker as the previous one. Thus, we update the current speaker model using this available new data, just as Fig. 2 (b) shows.

GMM(Gaussian Mixture Model)-32 is used to model a speaker. The model is established progressively as more and more data become available. At the beginning, there is no sufficient speaker data is to accurately estimate a GMM-32 model; thus, the GMM-1 is used. With the speaker data increase, the model will grow up to GMM-32 gradually.

In general, a standard EM algorithm is used to estimate the Gaussian Mixture Models. However, this will introduce two problems. First, it requires all feature data to be saved in the memory or disk. It will cost much memory or disk storage when the speech of a speaker is long. Second, EM algorithm contains a recursive process that it might not guarantee real-time processing. Therefore, we have introduced an alternative clustering method which is less time consuming, although its accuracy is not as high as EM algorithm. However, it works well in our applications.

Supposing that the current speaker model  $G_i$  is obtained from the previous (*M*-1) sub-segments and there is no potential speaker change point between (*M*-1)th and *M*th speech segment, it means these two segments belong to the same speaker. Thus,

we update the current speaker model  $G_i$  using the feature data of the  $M$ th segment.

If the current model  $G_i$  is represented by  $N(u, C)$ , in which the number of feature vectors used is  $N$ ; and the model obtained from  $M$ th speech segment is  $N(u_m, C_m)$ , in which the number of feature vectors is  $N_m$ . It could be easily derived that the current speaker model could be updated by the following method

$$C' = \frac{N}{N+N_m}C + \frac{N_m}{N+N_m}C_m + \frac{N \cdot N_m}{(N+N_m)^2}(\mu - \mu_m)(\mu - \mu_m)^T \quad (5)$$

The third part of (5) is determined by the means, which are easily biased by environment conditions. So, in practice, we ignore the mean part of (5) to compensate the effect of different environment conditions or transmission channel. Then (5) is simplified as

$$C' = \frac{N}{N+N_m}C + \frac{N_m}{N+N_m}C_m \quad (6)$$

The above procedure is looped till the dissimilarity between the speaker models before and after updating is small enough or a potential speaker change point is met. The dissimilarity is also measured by the *LSP* divergence shape distance. When the dissimilarity is sufficiently small, it is assumed that the current Gaussian model is estimated accurately, i.e, it is not necessary to continue updating  $G_i$ . The next Gaussian model,  $G_{i+1}$ , is initiated and updated with the new data using the same method.

For one speaker, it will have several Gaussian Models estimated by the above method. Combining these Gaussian Models would form a quasi Gaussian Mixture Model. The weight of each Gaussian model is set by their corresponding number of training data.

Supposing the speaker model is GMM-s, in which each Gaussian model  $G_i$  is  $N(u_i, C_i)$  and the number of feature vectors used to estimate the  $G_i$  is  $N_i$ , ( $i = 1, \dots, s$ ). Then, the weight  $w_i$  of the  $i$ th Gaussian Model  $G_i$  is computed as  $w_i = N_i/N$ , where  $N = \sum_{i=1}^s N_i$  is the total number of feature vectors.

By using the above method, speaker model will grow from GMM-2, GMM-3, ..., up to GMM32. When the GMM32 is reached, the updating of the speaker model is stopped. This method (quasi-GMM) uses segmental clustering and it is a little bit different from the original GMM. It is less accurate than Gaussian Mixture model obtained using EM algorithms, but it is computationally simple that meet our real-time processing requirements, while it is able to achieve reasonable accuracy, which could be seen from our experiments results.

## 2.4 Speaker change boundary refinement

There are often false positives in potential speaker change points obtained with the algorithms described. To remove false positives, a refinement algorithm is applied. The algorithm is based on the dissimilarity between the current segment and the last speaker model obtained from the segments before the current potential boundary. In this step, Bayesian Information Criterion (BIC) [9][10] is used to measure the dissimilarity, as shown in Fig. 2 (c).

Bayesian Information Criterion is a likelihood criterion penalized by the model complexity: the number of parameters in the model. Supposing two Gaussian Model from two speech clips are  $N(u_1, C_1)$  and  $N(u_2, C_2)$ , the number of data used to estimate these two models are  $N_1$  and  $N_2$ , respectively; and when

one Gaussian Model is used to estimate these two speech clips, the model is  $N(u, C)$ ; then the BIC difference between the two models is:

$$BIC(C_1, C_2) = \frac{1}{2}((N_1 + N_2) \log|C| - N_1 \log|C_1| - N_2 \log|C_2|) - \frac{1}{2} \lambda (d + \frac{1}{2} d(d+1)) \log(N_1 + N_2) \quad (7)$$

where  $\lambda$  is a penalty factor to compensated for small size cases, and  $d$  is the feature dimension. In general,  $\lambda = 1$ . According to BIC theory, if  $BIC(C_1, C_2)$  is positive, the two speech clips could be considered from different sources (speakers). The advantage of using BIC is that it is threshold free

Suppose at the potential speaker boundary, the model of last speaker is GMM-s, in which each Gaussian model is  $N(u_i, C_i)$  ( $i = 1, \dots, s$ ); and the model of current segment is  $N(u, C)$ . Then the distance between them is roughly estimated as the weighted sum of the distance of  $N(u, C)$  and each  $N(u_i, C_i)$ .

$$D = w_i \cdot \sum_{i=1}^s BIC(C_i, C) \quad (8)$$

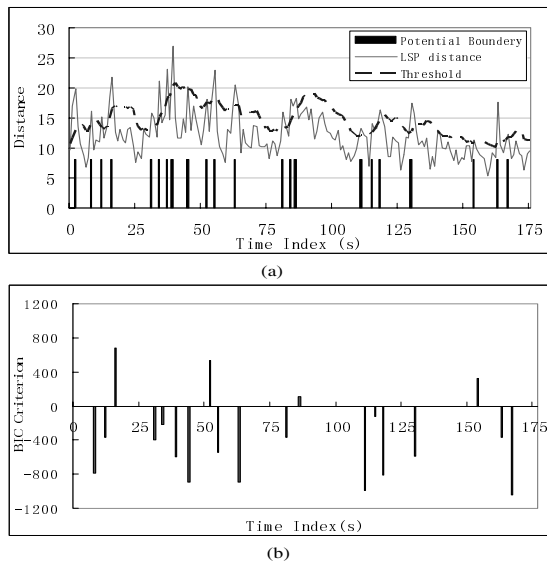
Thus, based on BIC theory mentioned above, if  $D > 0$ , it must be a real speaker change boundary. If a candidate is not a real boundary, the speaker data is used again to update the speaker model following the method described above.

We do not uniformly use *LSP* divergence distance or Bayesian Information Criterion at potential speaker boundary detection and refinement. The reason is as follows. At the step of potential speaker change detection, the data is so small that a model could not be estimated accurately, Bayesian Information Criterion is found to be easily affected by different words or different speakers, so the false alarms are too much. At the step of potential boundary refining, the model is more accurate, moreover, BIC could compensate different length of training data and threshold free, while *LSP* divergence distance depends on thresholds. It is more efficient for BIC in this step, which is also found in our experiments.

## 3. Experiments

The evaluations of the proposed speaker change detection algorithms have been performed by using our database, which includes speech in different conditions and different sample rate. The testing materials are news video programs from MPEG7 test data, CNN news, CCTV news, and they are totally about 2 hours. The audio track in the test set is sampled at 16kHz, 32kHz or 44.1kHz in one or two channels. In the experiments, each format audio is firstly converted to 8kHz and mono-channel before further processing.

Fig. 3 shows an example of 176-second long speech. The speech segment includes four speaker change boundaries, which are 17s, 52s, 86s, 154s respectively. Fig. 3 (a) shows the initial *LSP* distance between each two speech sub-segments, the adaptive threshold and the potential speaker change boundaries. It can be seen that the number of potential boundaries are much more than real boundaries. Fig. 3 (b) shows the Bayesian Information Criterion at the potential speaker change boundary with speaker model updated by using as much data as possible. If the value is positive, it is considered as a real speaker change boundary. There are four boundaries could be detected from Fig. 3 (b).



**Fig. 3 An example of speaker change detection**

The evaluations of the performance of speaker change detection are described with recall and precision. The results are listed in Table I. It can be noted that our algorithm performs very well. The overall recall is 89.89%, and the precision is 83.66%

Video Clip	Original	Detected	Miss	False	Recall	Precision
1	5	5	0	0	100%	100%
2	32	38	0	6	100%	84.21%
3	29	32	3	6	89.46%	81.25%
4	37	38	6	7	83.78%	81.58%
5	27	29	3	5	88.89%	82.76%
6	41	41	6	6	85.37%	85.37%
7	17	19	1	3	94.12%	84.21%
All	188	202	19	23	89.89%	83.66%

**Table I. Speaker Change Detection result using incremental speaker clustering algorithm**

In the experiment, we have found that if there is a laugh burst between speeches, it is easily detected as speaker change boundary. This is because we have no more coming data to be used to compare with the previous speaker model considering the real-time requirement with low delay. It is also found that the same speaker in different environment sometimes is detected as different ones. This indicates that our compensation for the effect of environment conditions or transmission channel is not sufficient. How to improve it still remains a big problem in the speaker recognition field and have a long way to go.

We have also tested the computational complexity of our algorithm in term of CPU time needed. With a Pentium III 667MHz PC with Windows 2000, the whole process can be completed in about 10% of the length of an audio stream. Therefore, our speaker change detection scheme is able to meet the requirement of real-time processing in multimedia applications.

## 4. Conclusion

Our contribution in this paper is that we developed an approach on unsupervised speaker change detection in real-time processing. This approach uses a novel scheme based on LSP analysis and suits a general case, in which the speaker identities and speaker number are both assumed to be unknown. The algorithm is low in computational complexity and suits well in the real-time processing in multimedia application. Incremental speaker modeling and adaptive threshold setting in potential speaker change detection are also described in detail in this paper. Experiments show the algorithm is very effective. The overall recall is up to 89.89%, and the precision is 83.66%.

In our future work, we will improve the performance of our speaker change detection algorithm and look for an efficient algorithm to compensate different environment or transmission channel. We would also extend the algorithm to speaker tracking.

## References

- [1] J. P. Campbell, JR. Speaker Recognition: A Tutorial. Proceedings of the IEEE, vl.85, no.9, pp.1437-1462, 1997.
- [2] J.N.L. Brummer. *Speaker Recognition over HF Radio after Automatic Speaker Segmentation*. Proceedings of the IEEE South African Symposium on Communications and Signal Processing, COMSIG-94. pp. 171-176, 1994
- [3] M. Sugiyama, J. Murakami, and H. Watanabe, *Speech Segmentation and Clustering Based on Speaker Features*. IEEE International Conference on Acoustics, Speech, and Signal Processing, 1993
- [4] L. Wilcox, F. Chen, D. Kumber, and V. Balasubramanian, *Segmentation of Speech Using Speaker Identification*. IEEE International Conference on Acoustics, Speech, and Signal Processing, 1994
- [5] M.H. Siu, G. Yu, and H. Gish. *An Unsupervised, Sequential Learning Algorithm for the Segmentation of Speech Waveform with Multiple Speakers*. IEEE International Conference on Acoustics, Speech, and Signal Processing, pp 189-192, 1992.
- [6] A. Cohen and V. Lapidus, *Unsupervised Speaker Segmentation in Telephone Conversations*, Nineteenth Convention of Electrical and Electronics Engineers in Israel, pp.102-105, 1996.
- [7] H. Gish and M. Schmidt. Text-Independent Speaker identification. IEEE Signal Processing Magazine. Pp.18-32, Oct. 1994
- [8] K. Mori and S Nakagawa. Speaker change Detection and Speaker Clustering Using VQ Distortion for Broadcast news Speech Recognition. ICASSP2000. pp.
- [9] S. Chen and P. S. Gopalakrishnan. *Speaker, Environment and Channel Change Detection and Clustering via the Bayesian Information Criterion*. Proc. of DARPA Broadcast News Transcription and Understanding Workshop, 1998.
- [10] G. Schwarz. *Estimation the Dimension of a Model*. The Annals of Statistics, Vol.6, pp. 461-464, 1978