Voting for two speaker segmentation

Balakrishnan Narayanaswamy[1,2], Rashmi Gangadharaiah[3], Richard Stern[1,3]

Department of [1]ECE, [2]ISRI, [3]LTI
Carnegie Mellon University, Pittsburgh, USA
{muralib,rgangadh,rms}@cs.cmu.edu

Abstract

The process of locating the end points of each speakers voice in an audio file and then clustering segments based in speaker identity is called speaker segmentation. In this paper we present a method for two speaker segmentation, though it can be extended to more than two speakers. Most methods for speaker segmentation and clustering start with an initial computationally inexpensive speaker segmentation method, followed by a more accurate segment clustering. In this paper we describe a simple algorithm that improves the accuracy of the segment clustering while not increasing the computational complexity. Since the clustering is done iteratively, the improvement in each segment clustering step results in a significant overall increase in segmentation accuracy and cluster purity. We borrow ideas from speaker recognition to perform segment clustering by frame based voting. We look at each frame as an independent classifier deciding which speaker generated that segment. These 'classifiers' are combined by voting to make a decision as to which segments should be clustered together. This simple change leads to 56.9% decrease in error rates on a segmentation task for the SWITCHBOARD corpus.

Index Terms: Speaker segmentation, Voting based classifier combination, Speaker change detection, Speaker clustering.

1. Introduction

In speaker identification applications, it is often assumed that the speech file contains data from a single speaker. However, in many applications such as identifying participants in a telephone conversation or in a meeting, speech from different speakers is intermixed. With the increase in the number of multimedia documents that need to be properly archived and accessed, speaker indexing has become an important area of research. One important cue for indexing can be speaker identity. Given an audio document, three different tasks are seen to be necessary. Firstly, the goal of speaker segmentation (which is the task addressed in this paper) is to segment a conversation into homogeneous parts containing the voice of only one speaker. For this we need to perform speaker change detection. Generally, no a-priori information is available on the identity of speakers involved in the conversation. The second task is speaker tracking, which consists of finding all the occurrences of a particular speaker and grouping segments based on speaker identity. The final task for audio data indexing (which is not addressed here) is usually speech recognition to identify what is being said and to allow the audio data to be searched based on content. In this paper, we look at telephone conversations where it is known a priori that there are two speakers, but the identity of the speakers is not known. We do not require any prior audio data from the speakers, as the methods in this paper are successful as long as the number of speakers is known beforehand.

Methods for speaker segmentation can be broadly classified into metric based and model based methods. Metric based splitting finds speaker change points based on maxima of some distance measure between two adjacent windows shifted along the speech signal. These segmentation algorithms suffer from a lack of stability since they rely on thresholding of distance values. Several cluster distances have been tested in [1, 2]. Model based splitting uses models for each speaker, which are trained beforehand, and is preferred when prior audio information is available about the speakers.

Among the metric based methods, the Generalized Likelihood Ratio (GLR)[3] produces the best results when audio segments are short, showing high and narrow peaks at speaker change points. The segmentation algorithm based on the Bayesian Information Criterion (BIC), cannot detect two speaker changes closer to one another in time, as it has been shown to require longer speech segments [4]. The content based indexing proposed in [6] combines the GLR distance measure to detect the speaker change points and the BIC technique to refine the results in order to fight over-segmentation. If the number of speakers in a conversation is known, the accuracy of speaker segmentation can be improved as demonstrated in [7]. This method combines the advantages of both metric based methods (no prior speaker knowledge required) and model based methods (higher accuracy) which can be done only when the number of speakers in known or can be accurately estimated.

In [7], the speaker assignment step used a method similar to Viterbi training of Hidden Markov Models (HMMs) from speech recognition, where there was one state for every speaker. Each state had gaussian mixture emission probabilities, and each segment obtained using the GLR metric was assigned to the state that had a higher probability of generating that segment. The state emission probabilities are then re-estimated based on the segment assignment. However, in [5] it was pointed out that when Gaussian Mixture Models (GMMs) are used with limited training data (as is the case in speaker segmentation), frame based voting can provide an improvement in speaker recognition accuracy. In this paper, we use frame based voting in the speaker assignment step and show that it also results in a significant improvement in speaker segmentation accuracy.

The rest of this paper is organized as follows. Section 2 describes the system used in [7] which is one of our baselines. Section 3 explains the motivation for looking for a better method for segment clustering. Section 4 describes the frame by frame voting method, used in [5] for speaker recognition, and its application here to speaker segmentation. Section 5 details the evaluation setup, and the compares the methods, showing that voting

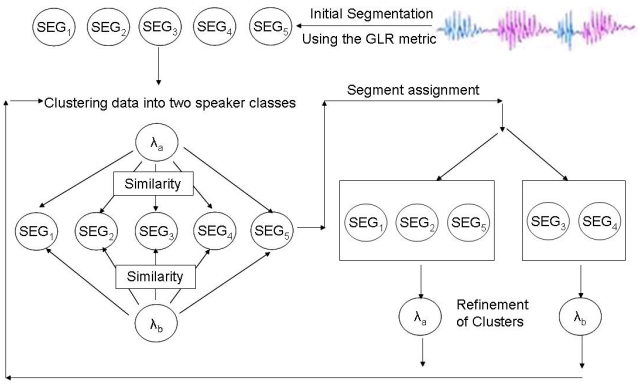

Figure 1: Graphical outline of proposed methods

improves speaker segmentation accuracy. Finally, Section 6 concludes and indicates some directions for future research.

2. Speaker Segmentation and clustering

Initial processing on the data involves removal of silence regions, which can otherwise lead to very bad segmentation results. A simple silence removal method is explained in Section 2.1. The method described in [7] consists of two major parts, the initial speaker change detection and the refining of models and change points. Fig. 1 shows a graphical outline of the proposed methods. A well known method for initial segmentation, using the GLR metric is outlined in Section 2.2. In many speech and speaker recognition tasks, models can be trained from a flat or random start, but when this is used for segmentation, the models may converge to speech classes (say consonants and vowels) rather than to different speakers. To force the models to converge to different speakers, we need to model spectral features across longer segments (at least 1-2s in length) to capture the long term speaker information but average out the short term speech information. We also need initial models that are more likely to contain data from different speakers. An algorithm to derive good initial speaker models is described in Section 2.3. The two clusters created in the initialization step are now used as the two reference models to cluster the remaining segments. This process can be repeated iteratively as described in Section 2.4, to obtain a final segmentation.

2.1. Silence Removal

A drawback of training HMMs on unlabelled data with silences is that some of the states in the model may converge to these silence regions. The method used for silence removal in this paper is similar to the second method in the NIST stnr routine [8]. To save computation, only the first 5-10s is used for detecting the speech-silence threshold. A signal power histogram is generated by computing the root mean squared (RMS) amplitude, in decibels, over 20ms windows shifted by 10ms and creating a histogram of the values. This histogram of these power coefficients is seen to be bi-modal, with a sharp low power silence mode and a flatter higher power speech mode. It should be noted that this may not be true if the noise is at the same level as the speech signal, but this is rarely the case. The point of inflection between these two modes is the boundary between speech and silence, and frames below this threshold are rejected as silence regions. After silence removal feature extraction is performed in the speech signal. We use the first 12 Mel Frequency Cepstral Coefficients (MFCCs) [10], not including the average energy or C_0 coefficient.

2.2. Initial segmentation using the GLR metric

As described in [3], the similarity between two contiguous windows of the parameterized signal is calculated using the following hypothesis test:
H_0: The segments were generated by the same speaker
H_1: The segments were generated by different speakers
where a speaker is represented by either a single gaussian model or a GMM. The negative of the log likelihood ratio of this test is called the GLR distance between the two segments. The GLR distance is computed between a pair of adjacent windows of the same size, and the windows are then shifted by a fixed step along the whole parameterized speech signal. A large distance indicates change in speaker, whereas low values signify that the two portions of the signal correspond to the same speaker. The thresholded peaks of this distance metric are labeled as initial change points. Further details can be obtained from [7]. The result of this step is a series of segments $SEG_1, SEG_2 ... SEG_n$.

2.3. Speaker model initialization

Next, we try to find data segments that have a high likelihood of being from different speakers, to initialize GMMs for the segment clustering step. We first train a GMM using the Expectation Maximization (EM) algorithm[9]) for each segment obtained from the initial segmentation, using 4 or 8 gaussians each. We can then calculate for each segment SEG_i with feature vectors $\vec{seg}_{i,1}, \vec{seg}_{i,2}, \ldots, \vec{seg}_{i,T}$ and model λ_k (a GMM with M gaussians) a measure of their similarity the log likelihood that model k generated segment i,

$$Sim(SEG_i|\lambda_k) = \Sigma_{t=1}^{T} log(p(\vec{seg}_{it}|\lambda_k)) \quad (1)$$

$$p(\vec{x}|\lambda) = \Sigma_{i=1}^{M} p_i b_i(\vec{x}) \quad (2)$$

$$b_i(\vec{x}) = \frac{1}{(2\pi)^{D/2}|\Sigma_i|^{\frac{1}{2}}} exp(-\frac{1}{2}(\vec{x} - \vec{\mu_i})' \Sigma_i^{-1} (\vec{x} - \vec{\mu_i}) \quad (3)$$

where D is the dimensionality of the data (12 in this paper). We find the two segments a and b that give the smallest pairwise similarity, and initialize our models of the two speakers to the parameters of these segments, λ_a and λ_b.

2.4. Segment clustering and refinement

The two clusters created in the initialization step are now used as the two reference models (λ_a and λ_b). The objective then is to find the model which has the maximum $aposteriori$ probability for each segment SEG_y, where, y = 1,2,3,...,n. If the reference model λ_a shows a higher $aposteriori$ probability compared to λ_b for SEG_y, then SEG_y will be labeled as the first speakers data otherwise it will be labeled as the second speakers data. Each segment is assigned to a cluster $\hat{C}_y$ based on Equation 4.

$$\hat{C}_y = \arg \max_{m=a,b} p(\lambda_m|SEG_y) \qquad y = 1, 2, 3, \ldots, n \quad (4)$$

The segments are clustered based on the labels. New models are trained for each cluster, using all the data from the segments assigned to that cluster. Using these new models, the procedure can be repeated iteratively to obtain clusters of high purity for the two speakers. The performance was found not to increase significantly

beyond the fourth iteration. Thus, final change points and clusters are obtained.

3. Problems during segment assignment

An analysis of the kinds of mistakes made by the above system was performed. Most of the errors were seen to be in the early segment clustering iterations. There seem to be two possible sources of this error. The first is that the initial segmentation is not perfect and some of the segments contain data from multiple speakers. In such situations the preferred outcome is that the segment be assigned to the cluster for the speaker for which it has more data. However, it will actually be assigned to the cluster based on which speaker had a higher probability which is not always the same thing. We would like to count the number of frames belonging to each speaker and assign clusters based on this. Thus, the models will continue to contain more data from a single speaker.

A second source of error occurs when very little data is available to train GMMs, making them unreliable. The models tend to have much higher variance than GMMs trained on a large corpus of speaker data. If these models are used to cluster segments, it was seen that in many cases the correct speaker had very low scores in only a few frames in a segment. Despite scoring better than the other speaker in all the other frames of that segment, the correct speaker was not selected. These few frames could be noisy frames, variant frames where the speaker model did not match properly, overlapping speech, a short utterance by the other speaker, or frames belonging to some sound not seen in the limited initial training data. An example of the log probabilities for an example utterance are shown in Fig. 2. Here the correct speaker has a lower probability for just a few frames, but since the dip in frames 17 to 23 is so large that taking the product of the probabilities (sum of log probabilities) gives the other speaker a higher score for the segment. Many of the segments which were clustered incorrectly using GMMs had at least five or six frames with large dips in the log probability, even upto -80 in log probability (which corresponds to multiplication of probability by a factor of 10^{-80}). When these few regions were identified and removed by hand many errors made by the system were corrected. To avoid the influence of a few bad frames causing wrong identification there is a need to make the influence of the frames more uniform. So, a voting based combination scheme is suggested, where each frame has a single vote.

4. Voting for two speaker segmentation

One solution to the problem described in the previous section, suggested in [11] is to dynamically change the number of gaussians in each iteration, increasing the number of gaussians as more data becomes available. However, the problem of how many gaussians to use is still an open one, and may differ based on the database used, recording conditions, or speaker talking characteristics (for example, if one speaker continues to speak for a long time). The solution we use is based on [5] where frame by frame voting is applied. In conventional GMM evaluation [7], the effect of the few frames may be amplified because the probabilities of each frame are multiplied to achieve the final posterior probability, possibly giving a higher weight to some frames which have a much lower score.

$$P(SEG_i|\lambda_k) = \prod_{t=1}^{T} p(s\vec{e}g_{it}|\lambda_k) \quad (5)$$

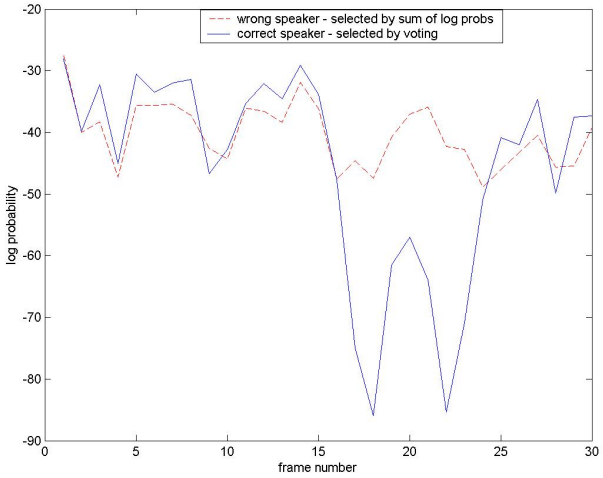

Figure 2: Output log probabilities of the Gaussian Mixtures for the two different speakers for one segment of an audio file

In the proposed voting algorithm, each frame is viewed as an independent classifier. Using the GMM parameters each classifier makes an independent decision as to who the speaker is. In the case of [7] the outputs of the frames are the probabilities $p(s\vec{e}g_{it}|\lambda_k)$ which are then combined by multiplication. In the proposed method the decisions of all the classifiers (frames) are combined by voting. Thus in the voting scheme, for each frame in a segment, we find the most likely speaker $\hat{S}$ for that frame $s\vec{e}g_{it}$ by,

$$\hat{S} = \arg\max_{k=a,b} p(s\vec{e}g_{it}|\lambda_k) \quad (6)$$

Thus, the frames together function as an ensemble classifier. In an ensemble classifier each classifier is run and casts a vote as to who the correct speaker is. The votes for each segment are then collated and the speaker with the greatest number of votes becomes the final classification. For example in the segment shown in Fig. 2, frame by frame voting choses the correct speaker where [7] would not. As before, the clustered segments are used to retrain models, and the process can be repeated iteratively.

5. Evaluation

There are two types of errors related to speaker change detection, an insertion error occurs when a speaker change is hypothesized although it does not exist, a deletion error occurs when a true speaker change is not detected. 12^{th} order MFCCs are computed for every frame of 16 ms with 6.25ms shift for parameter extraction. For speaker change detection, the length of each window is set to 1s and shifted by 0.5s. These parameters are the same for all the methods and so all the methods have a resolution of 0.5s. Errors are declared whenever the hypothesized and true change points differ by more than 0.5s. A four component GMM with diagonal covariance is used to compute the GLR distance between two consecutive windows. For segment clustering, eight component GMMs with diagonal covariance is used. We first compare our method with [6], which uses the GLR distance to detect initial

Table 1: Comparison of [6] and the Proposed Method on TIMIT and SWITCHBOARD. I - number of Insertion errors, D - number of Deletion errors F-female, M-male, CPs - Change Points

Files	GLR,BIC [6]				Prop.	
	$1^{st}Pass$		$2^{nd}Pass$		Meth.	
	I	D	I	D	I	D
TIMIT 29 CPs	26	3	9	7	2	2
TIMIT 27 CPs	23	3	9	7	3	2
sw2005(M-M) 19 CPs	41	6	17	7	0	2
sw2008(F-F)30 CPs	31	17	18	17	4	3
Total	150		91		18	

Table 2: Comparison of conventional GMM evaluation and voting on SWITCHBOARD. I - number of Insertion errors, D - number of Deletion errors F-female, M-male, CPs - Change Points

Files	Conv.. GMM[7]		Voting Method	
	I	D	I	D
sw2137(M-F) 24 CPs	19	7	9	3
sw2104(M-F) 10 CPs	18	3	14	2
sw2014(F-F) 29 CPs	12	6	2	5
sw2124(F-F) 23 CPs	23	5	5	0
Total	93		40	

speaker change points and the BIC to refine the results. We use evaluation tasks on TIMIT and SWITCHBOARD [12]. A conversation is obtained from TIMIT by concatenating sentences of average length 2s from two speakers. Two files from SWITCHBOARD, sw2005 and sw2007 are also used. The results are presented in Table 1. While the proposed method does much better than this baseline, it does exactly as well as the method using conventional GMM evaluation [7]. The reason for this is that TIMIT and these particular switchboard files are relatively clean with long speaker segments, and the method of [7] does so well that there is little room for improvement.

We proceed to compare both the methods in a larger selection of files from SWITCHBOARD. These were not selected to show the benefit of our algorithm, but were chosen at random from the rest of the database. The results are shown in Table 2. Here we see that voting gives us a substantial improvement on all the files. This also shows that the proposed method is robust to noise/silence regions and other variations in the speech signal.

6. Conclusions

In this paper, we presented a method for two speaker segmentation which uses frame by frame voting in the segment assignment step. By using information about the number of speakers present in the recording, data from different parts of the file can be collected to train a single model better, as compared to agglomerative clustering method like [6] which cannot directly use this information. When segments are short, and limited data is available from each speaker, methods like [7, 11] result in gaussians with a high variance, which leads to spiky frame posterior probabilities, since high variance gaussains give near 0 probability to feature vectors which differ even slightly from the feature vectors present in the training data. Voting ensures that even in such cases, the correct speaker for each segment is still chosen. We do not need to dynamically change the number of gaussians in the mixture model, as voting compensates for the high variance models in the early iterations. This method can be extended to N speakers (for N greater than 2), by finding the N GLR segments that are furthest apart, and use them to initialize N models in the speaker initialization step. We would like to investigate the feasibility of this method for N speaker segmentation.

7. References

[1] Laurent Couvreur and Jean-Marc Boite, “Speaker tracking in broadcast audio material in the frame work of the THISL project”, Proc. of European Speech Communication Association, ETRW Workshop on Accessing Information in Spoken Audio, Cambridge (UK),84–89, 1999.

[2] Shrikanth Narayanan and Soonil Kwon, “Speaker change detection using a new weighted distance measure”, Proc. of ICSLP, Denver, USA, volume 4, 2537–2540, 2002.

[3] M.Siu H.Gish and R.Rohlicek, “Segregation of speakers for speech recognition and speaker identification”, Proc. of IEEE ICASSP, 873–876, 1991.

[4] S.Chen and P.Gopalakrishnan, “Speaker, environment and channel change detection and clustering via the bayesian information criterion”, DARPA speech recognition workshop, Lansdowne, Virginia, 1998.

[5] B. Narayanaswamy and Rashmi Gangadharaiah, “Extracting Additional Information from Gaussian Mixture Model Probabilities for Improved Text Independent Speaker Identification”, Proc. of IEEE ICASSP, Philadelphia, PA, March 2005.

[6] David Kryze, Perrine Delacourt and Christian J. Wellekens, “Speaker-based segmentation for audio data indexing”, SCA International Speech Communication Association, Cambridge, UK, 78–83, 1999.

[7] Rashmi Gangadharaiah, B. Narayanaswamy, N. Balakrishnan, “A Novel Method for Two-Speaker Segmentation”, Proc. of ICSLP, Jeju, Korea September 2004.

[8] Casimir Wierzynski and Jon Fiscus, ’stnr.doc’ included with the nist speech quality assurance (spqa) package version 2.3 and speech file manipulation software (sphere) package version 2.5.

[9] N. M. Laird, A. P. Dempster and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm”, Journal of the Royal Statistical Society. Series B (Methodological),vol. 39, 138,1977.

[10] S.B. Davis and P. Mermelstein, “Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences”, IEEE Trans. on ASSP, vol. 28, 357366, 1980.

[11] Adami, A.G.; Kajarekar, S.S.; Hermansky, H., “A new speaker change detection method for two-speaker segmentation”, Proc. of IEEE ICASSP, Volume 4, IV-3908 - IV-3911,13-17 May 2002.

[12] Joseph P. Campbell, Jr and Douglas A. Reynolds, “Corpora for the Evaluation of Speaker Recognition Systems”, Proc. of IEEE ICASSP, paper number 2247, 1999