Report on Speaker Change Detection and Clustering Algorithms

SpeecHWareNet (I) Pvt. Ltd.
Technology Incubation Centre
IIT Guwahati, Assam, India

November 30, 2017

Chapter 1

Speaker Change Point Detection

1.1 Introduction

Speaker segmentation is the first phase of a speaker diarization system, which is followed by a speaker clustering phase. Speaker segmentation aims at finding the speaker change points in an audio stream. Therefore Speaker change detection is the most crucial part in speaker segmentation in multi speaker conversation [1] [2]. Various methods are proposed in the literature for this purpose like distance metric based, model based, silence detection based etc. along with some hybrid methods. Distance metric based methods are the most popular and widely used [1] [2], where a distance metric between every two consecutive analysis segments is used as a decision measure for determining the change points as shown in Fig. **??** [1] [2] [3]. Bayesian Information Criteria (BIC) is the dominant method among the distance metric based methods. BIC is an optimal Bayesian model selection criterion used to decide which of the model represents the data samples best. This technique searches for change points within a window using a penalized likelihood ratio test whether the data in the window is better modeled by a single Gaussian distribution or two different Gaussian if the data is modeled by Gaussian process [4] [5] [6].

Speaker diarization has three main application domain namely broadcast news, meetings and conversational telephone speech. Radio and TV programs containing commercial breaks and music recorded over a single channel comes under broadcast news category. In meeting speech data multiple speakers interact over single or multiple microphones. Based on the number of microphones used the recording condition of this category is divided into single distant microphone

(SDM) and multiple distant microphones (MDM). Last category is the conversational telephone speech where single channel recording takes place in telephone conversations between two or more people. Depending on the application domain various evaluation databases are available. For example, NIST 1996 HUB-4, Hub-4 1997 English Broadcast News Speech Database, 1998 HUB-4 broadcast news evaluation English test material, ESTER SD benchmark etc. for broadcast news. Similarly ICSI Meetings Corpus, CMU Meeting Corpus, NIST Pilot meeting corpus and CHIL meeting corpus are for meeting domain. NIST Rich Transcription evaluations, 2006 and NIST Fall Rich Transcription, 2006 are the more widely used corpora, which contain database for all the three categories [1] [2]. IITG-MV Phase III Speaker Recognition Database Part II (two speaker recognition) [12] is used here to evaluate the performance of these algorithm. It is a conversation mode database where every speaker spoke in conversational style over a conference call. It has the variabilities like multi-environment, Multi-sensor and Multi-lingual [12].

The performance of speaker change point detection algorithms are evaluated by computing the Miss Detection Rate (MDR), False Alarm Rate (FAR) and Identification Accuracy (IDA) based on the manually marked change points provided with speaker database. MDR and FAR are defined by

$$[MDR = \frac{\text{No. of miss detections}}{\text{No. of actual speaker boundaries}}] \tag{1.1}$$

$$[FAR = \frac{\text{No. of false alarms}}{\text{No. of detected speaker boundaries}}] \tag{1.2}$$

FAR and MDR both should be lowest in order to achieve best performance.
In order to measure how close are the hypothesized change point with the ground truth, Identification Error (IDE) and Identification Accuracy (IDA) are computed. Within the tolerance range of either side of a ground truth if at least one hypothesized change point comes then IDE for that change point is computed. IDE and IDA are defined as [13],

- Identification Error (IDE): The timing error between the ground truth and the hypothesized speaker change point within the tolerance range.

- Identification Accuracy (IDA): Standard deviation IDE is the identification accuracy for a given speech signal.

IDA should be low in order the ensure high accuracy of hypothesized change points.

Chapter 2

Speaker Change Detection Algorithm using Kullback Leibler Distance (KLD)

Kullback Leibler Divergence is a distance measure for finding the difference among the two distributions. The Kullback-Leibler distance (KLD) between two multivariate Gaussian distributions $N(\mu_0, \Sigma_0)$ and $N(\mu_1, \Sigma_1)$ is

$$[D_{KL(N_0 \| N_1)} = \frac{1}{2}(tr(\Sigma_1^{-1}\Sigma_0) + (\mu_1 - \mu_0)^T \Sigma_1^{-1}(\mu_1 - \mu_0) - k - ln(\frac{det\Sigma_0}{det\Sigma_1}))] \quad (2.1)$$

A high distance value indicates a possible acoustic change whereas a low value show that two portions of signal corresponds to the same acoustic environment.

2.1 Speaker change detection using KLD

Feature vectors belonging to a particular speaker form a separate cluster. Therefore at the speaker change point, KLD is high compared to other points. The steps involved in the revised KLD algorithm is explained below and algorithm level plot is shown in Fig **??**.

1. Step 1: Compute KLD contour from the MFCC feature vectors:
   Initially MFCC feature vectors are computed directly from the speech signal using the following specification:

   - Framesize = 20 ms

- Frameshift = 10 ms
- Number of cepstral coefficients excluding 0'th coefficient (default 12)= 13
- No. of filters in Filter Bank = 26

Distance measurement is performed between two windows (analysis window) shifted along the MFCC feature vectors and a distance contour is obtained. The analysis window size considered here is 100 ms with a shift of 10 ms. The distance contour is then resampled to the original speech signal length and normalized to the maximum value.

2. Step 2: Detection of peak indicating speaker change point:

   - Smoothing using hamming window : The distance contour is smoothed by convolving with a hamming window of length 500 ms.
   - Locate peaks: In the smoothed contour, mean of the distance values is calculated. Then an axis is set at the mean value. The smoothed contour and its mean value axis will give the positive and negative mean crossing points. Mark the maximum distance value between each consecutive positive and negative mean crossing points as peak points. Thus a set of peak points is obtained. In the same way two more axis are set at $mean + 0.05$ and $mean - 0.05$. Thus total three sets of peak points is derived.
   - Derive final set of speaker change points: In the next step unique points from these three sets are extracted and a threshold is set. Peaks greater than this threshold is only considered as a speaker change point. This threshold is set as which is decided after the performance evaluation for different values over the whole database.

2.2 Performance Evaluation of KLD Algorithm

2.2.1 Database

The database used for testing the performances of the algorithms is IITG-MV Phase III Speaker Recognition Database Part II (two speaker recognition). The data is conversational style speech between two speakers. There are two sessions

Table 2.1: Performance of KLD algorithm for different threshold and tolerance window

Sl. No	Th_{peak}	Tol_{win}	MDR (%)	FAR (%)
1	0.05	100	89.66	95.57
		250	67.47	86.25
		500	**30.87**	**71.10**
		1000	6.49	61.29
2	0.1	100	89.44	95.53
		250	68.82	86.41
		500	33.08	71.69
		1000	6.67	60.63
3	0.2	100	90.83	95.29
		250	72.63	85.85
		500	41.81	69.49
		1000	17.27	54.50
4	0.3	100	93.86	95.41
		250	80.22	85.05
		500	56.47	65.43
		1000	39.99	44.02

in the database. Each session has 98 conversations. Each file has a corresponding label file containing the ground truth of speaker change points. This ground truth is to be used to measure the accuracy of the algorithms developed. The testing is to be done on both the sessions and the performances are to be reported accordingly. In addition, it is to be mentioned that each file in the database is about 12 to 15 minutes long. However, we can use only the first 1 minute of the speech files to report the accuracy.

2.2.2 Result

The performance of the KLD algorithm with diffrent threshold (Th_{peak}) and and tolerence window (Tol_{win}) values for IITG-MV Phase III Speaker Recognition Database Part II (two speaker recognition) database is given in Table 3.1. The best result of MDR **30.87 %** and FAR **71.10 %** is achieved for the Th_{peak}=0.05 and Tol_{win}=500.

Table 2.2: 5 speech samples with best performance with Th_{peak} = 0.05

S.N.	Tolerance (ms)	Accuracy (%)	MDR (%)	FAR (%)
1	500	1001-2133	17.39	38.71
2		1051-1358	26.09	48.48
3		1273-2131	8.33	40.54
4		2041-2135	14.29	40.00
5		1300-1070	6.67	54.84
6		**Average**	**14.55**	**44.51**

Table 2.3: 5 speech samples with moderate performance with Th_{peak} = 0.05

S.N.	Tolerence (ms)	Accuracy (%)	MDR (%)	FAR (%)
1	500	1012-2143	40.00	62.50
2		1013-1340	23.53	56.67
3		1038-1376	38.89	65.62
4		1074-1390	33.33	61.54
5		1077-1303	33.33	56.25
6		**Average**	**33.82**	**60.52**

Table 2.4: 5 speech samples with poor performance with Th_{peak} = 0.05

S.N.	Tolerence (ms)	Accuracy (%)	MDR (%)	FAR (%)
1	500	1062-1297	62.50	72.73
2		1108-1365	69.23	87.32
3		1295-1094	80.00	96.67
4		1296-1061	75.00	96.15
5		1343-1344	66.67	83.33
6		**Average**	**70.67**	**87.24**

Chapter 3

Speaker Change Detection using Bayesian Information Criteria (BIC)

The Bayesian Information Criterion (BIC) is a model selection criterion that was first proposed by Schwarz [16] and widely used in the statistical literature. It is a likelihood criterion penalized by the model complexity: the number of parameters in the model [5]. The generalized likelihood ratio (GLR) is used to decide which of the models represents data samples best. The problem of model selection is to choose one among a set of desired candidate parametric models M_i, $i = 1, 2, 3, ..., m$, with corresponding model parameters θ_i to represent a given data set $D = D_1, D_2, D_3, ..., D_N$. The BIC of model M_i for the given data is defined as,

$$BIC(M_i) = logP(D_1, D_2, D_3, ..., D_N|M_i) - \frac{1}{2}\lambda d_i logN \tag{3.1}$$

where, $\lambda = 1$ is penalty weight, d_i is the number of *independent* parameters in the model parameter set, and $logP(D_1, D_2, D_3, ..., D_N|M_i)$ is the maximized data likelihood for the given model. In BIC, the term $\frac{1}{2}d_i logN$ is subtracted from the log-likelihood to penalized for model complexity, where BIC favors the model which maximizes the BIC values. In the case of only two competing models, the BIC difference can be seen as an approximation to the logarithm of the Bayes factor. For a given number of models including the true model, the probability that BIC will choose the correct model approaches one as the sample size $N \rightarrow \infty$. However, for finite samples, BIC often selects the simple model due to its heavy penalty against complexity. This observation suggests that, for applications dealing with varied sample sizes such as the type of audio segmentation we will consider, it is reasonable to adjust the penalty of complexity especially when sample size is small [17].

3.1 Speaker Change Point Detection Using BIC

Let us consider $X = x_i \in R^d, i = 1, ..., N$ as the d-dimensional sequence of cepstral feature vectors from an audio stream in which there is at most one segment boundary. We wish to consider if there is a boundary at frame $b \in (1, N)$. If we suppose If we suppose that each acoustic homogeneous speech block can be modeled as one multivariate Gaussian process $X \sim N(\mu, \Sigma)$, the segmentation issue can be cast as a model selection problem between the following two nested models,

$$H_0 : X = x_1, x_2, ..., x_N \sim N(\mu, \Sigma)$$

and

$$H_1 : x_1, x_2, ..., x_b \sim N(\mu_1, \Sigma_1); x_{b+1}, x_{b+2}, ..., x_N \sim N(\mu_2, \Sigma_2)$$

That is, the first model assumes that all samples are independent and identically distributed (*iid*) to an single Gaussian, and the second model assumes the first b samples are drawn from one Gaussian while the last $N - b$ samples are drawn from another Gaussian. Here, if BIC favors M_1 then the data is assumed homogeneous, otherwise break should occur within this block of data.

It can be shown that given the assumption of a normal distribution $N(\mu, \Sigma)$, the likelihood of observation data $x_1, x_2, ..., x_N$ is maximized when $\mu = \hat{\mu}$ and $\Sigma = \hat{\Sigma}$, where

$$\hat{\mu} = \bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{3.2}$$

$$\hat{\Sigma} = \frac{1}{N}\sum_{i=1}^{N} (x_i - \bar{x})(x_i - \acute{\bar{x}}) \tag{3.3}$$

Thus, according to equation 3.1, the BIC values of these two models can be computed as,

$$B\bar{I}C(H_0) = -\frac{d}{2}Nlog2\pi - \frac{N}{2}log|\hat{\Sigma}| - \frac{N}{2} - \frac{1}{2}\lambda\left(d + \frac{1}{2}d(d+1)\right)logN \tag{3.4}$$

$$B\bar{I}C(H_1) = -\frac{d}{2}Nlog2\pi - \frac{b}{2}log|\hat{\Sigma_1}| - \frac{N-b}{2}log|\hat{\Sigma_2}| - \frac{N}{2} - \lambda\left(d+\frac{1}{2}d(d+1)\right)logN \tag{3.5}$$

where, $\hat{\Sigma}|$, $\hat{\Sigma_1}|$, and $\hat{\Sigma_1}|$ are maximum likelihood covariance estimations from corresponding data, λ is the penalty factor to compensate for small sample size cases, and d is the cepstral feature dimension. Next, the BIC difference between the two models can be computed as a function of break point b

$$\triangle BIC(b) = B\bar{I}C(H_1) - B\bar{I}C(H_0) \tag{3.6}$$

$$= \frac{1}{2}(Nlog|\hat{\Sigma}| - blog\hat{\Sigma_1}| - (N-b)log|\hat{\Sigma_2}|) - \frac{1}{2}\lambda\left(d+\frac{1}{2}d(d+1)\right)logN \tag{3.7}$$

According to the BIC rule, segmenting this audio stream into two parts at frame b will be favored if $\triangle BIC(b) > 0$. The final segmentation decision can be achieved via MLE

$$\hat{b} = argmax_{1<b<N;\triangle BIC(b)>0}\triangle BIC(b) \tag{3.8}$$

In the BIC formula, the penalty weight λ is introduced in order to compensate for the differences between the theory and the practical application of the criterion. We need to take penalty weight λ value such that it provides a good tradeoff between miss rate and false-alarm. It has been found that the factor λ is task-dependent and has to be retuned for every new task.

3.2 Algorithm for speaker change detection using BIC

BIC searches for change points within a window using penalized likelihood ratio test of whether the feature vectors in the window is better modeled by a single distribution or two different distributions. Therefore at the event of speaker change, feature vectors within the window will be better modeled by two different distribution.

The steps involved in the BIC algorithm is explained below and algorithm level plot is shown in Fig **??**.

1. **Step 1: Compute BIC contour from the MFCC feature vectors:**
   Initially MFCC feature vectors are computed directly from the speech signal using the following specification:

- Framesize = 20 ms
- Frameshift = 10 ms
- Number of cepstral coefficients excluding 0'th coefficient (default 12)= 13
- No. of filters in Filter Bank = 26

Distance measurement using BIC is performed between two windows (analysis window) shifted along the MFCC feature vectors and a distance contour is obtained. The analysis window size considered here is 100 ms with a shift of 10 ms. The distance contour is then resampled to the original speech signal length and normalized to the maximum value.

2. **Step 2: Detection of peak indicating speaker change point:**

   - Smoothing using hamming window : The distance contour is smoothed by convolving with a hamming window of length 500 ms.
   - Locate peaks: In the smoothed contour an axis is set at zero and maximum distance value between a consecutive positive and negative zero crossing point is marked. In order to that positive and negative zero crossing points are marked initially and peak points are located at the maxima between every pair of zero crossing points. Thus a set of peak points is obtained. In the same way two more axis are set at $+0.1$ and -0.1. Thus two more sets of peak points are derived.
   - Derive final set of speaker change points: In the next step unique peak points from these three sets are extracted and a threshold Th_{peak} is set. Peaks which are greater than this threshold is only considered as a speaker change point. The threshold is set as 0, which is decided after the performance evaluation for different values over the whole database.

3.2.1 Database

The performance of the BIC algorithm is done using IITG-MV Phase III Speaker Recognition Database Part II (two speaker recognition). The data is conversational style speech between two speakers in telephone channel. There are two sessions in the database. Each session has 98 conversations. Each file has a corresponding label file containing the ground truth of speaker change points. This ground truth is to be used to measure the accuracy of the algorithms developed. The testing is done on both the sessions and the performances are evaluated for both the sessions. Each file in the database is about 12 to 15 minutes long, however, the performance of the BIC algorithm is evaluated only for the first 1 minute of the speech files.

3.2.2 Performance Evaluation of revised BIC and Conclusion

The performance of the revised BIC algorithm varies according to the following parameters:

- BIC penalty factor, λ
- Threshold, Th_{peak}

The performance of the BIC algorithm using IITG-MV Phase III Speaker Recognition Database Part II (two speaker recognition) is given in Table 3.1 for various λ and Th_{peak}. In our database we found that the best value of λ that gives a good tradeoff between miss rate and false-alarm arte is 1.1 with $Th_{peak} = 0$.

Table 3.1: Performance of revised BIC algorithm with Tolerance 500 ms

Sl. No	λ	Th_{peak}	MDR (%)	FAR (%)
1	0.5	0	84.39	65.43
		0.05	85.44	62.42
		0.1	86.32	60.13
		0.2	87.89	58.65
2	1	0	30.41	66.91
		0.05	34.54	64.18
		0.1	37.49	63.48
		0.2	45.60	60.97
3	1.1	0	30.14	65.31
		0.05	35.81	62.56
		0.1	39.27	61.45
		0.2	48.40	58.23
4	1.2	0	36.44	64.75
		0.05	42.25	61.39
		0.1	45.53	60.12
		0.2	54.87	57.80
5	1.3	0	43.51	63.95
		0.05	50.20	58.69
		0.1	55.16	58.63
		0.2	65.75	58.44
6	1.4	0	54.83	62.86
		0.05	61.28	57.82
		0.1	67.58	56.76
		0.2	77.16	53.70
7	1.5	0	66.58	65.51
		0.05	73.51	59.60
		0.1	77.80	58.42
		0.2	85.15	57.12

Table 3.2: Performance of revised BIC algorithm with different Tolerances

Sl. No.	Tolerance(ms)	λ	Th_{peak}	MDR (%)	FAR (%)
1	100	1	0	83.29	91.09
2	250	1	0	62.30	81.15
3	500	1.1	0	30.14	65.31
4	1000	1.1	0.05	16.43	51.29

Table 3.3: 5 speech samples with best performance with $\lambda = 1.1$ and $Th_{peak} = 0$

Sl. No.	Tolerance (ms)	Sample Number	MDR (%)	FAR (%)
1	500	2102-1362	18.52	43.59
2		2041-2135	21.43	38.89
3		1387-1388	5.56	41.38
4		1273-2131	21.43	38.89
5		1012-2143	13.04	47.37
6		**Average**	**15.99**	**42.02**

Table 3.4: 5 speech samples with moderate performance $\lambda = 1.1$ and $Th_{peak} = 0$

Sl. No.	Tolerance (ms)	Sample Number	MDR (%)	FAR (%)
1	500	1077-1303	27.27	51.52
2		1108-1365	21.43	67.65
3		1064-1298	23.08	67.74
4		2103-1379	21.05	57.14
5		1063-1372	19.05	45.16
6		**Average**	**22.38**	**57.84**

Table 3.5: 5 speech samples with poor performance $\lambda = 1.1$ and $Th_{peak} = 0$

Sl. No.	Tolerance (ms)	Sample Number	MDR (%)	FAR (%)
1	500	1026-1341	75.00	96.88
2		1268-2130	61.54	85.71
3		1370-1371	50.00	85.71
4		2034-2149	46.15	80.00
5		1038-1376	45.45	61.29
6		**Average**	**55.62**	**81.90**

Chapter 4

Speaker Change Detection Algorithm using Average Cepstral Distance (ACD)

A signal processing based algorithm is proposed here. Euclidean distance between the average of two regions of the speech signal on either side of a possible change point is computed. It is termed as average cepstral distance because two clusters of cepstral feature vectors within a window is considered and average is taken to compute the distance.
Consider $X = x_i \in R^d, i = 1, ..., N$ as the d-dimensional sequence of feature vectors. Consider

$$X_W : x_1 \cdots x_M, \text{where } M < N$$

is the region lies within an analysis window and j be the possible change point with P number of feature vectors on either side.
Then

$$DIST(j) = ED[AVR\{X_{W1} : x_1 \cdots x_j\}, AVR\{X_{W2} : x_{j+1} \cdots x_M\}]$$

is the distance value at the time instant j.
Here ED refers to the Euclidean distance and AVR refers to average. The analysis window considered here is an overlapping one and hence to compute the distance value at time instant $j+1$ window is shifted by one sample. It means $M = M+1$. A high distance value indicates a possible speaker or sound unit change whereas a low value show that two portions of signal corresponds to the same speaker or same sound unit.

4.1 Basis for speaker change detection using ACD

Cepstral feature vectors within two regions (minimum 100 ms) are considered on either side of a possible change point, hence if both side belongs to two different speakers cepstral distance is expected to be high at that point.

4.2 Algorithm

The steps involved in the algorithm is explained below.

1. **Step 1: Obtain Zero Frequency Filtered speech (ZFFS) signal and Compute MFCC feature vectors from the ZFFS signal**
   ZFFS signal is computed and MFCC feature vectors are extracted from 20 msec frame of ZFFS signal with a overlap of 10 msec. Specifications:

   - Framesize = 20 ms
   - Frameshift = 10 ms
   - No of Cepstral Coefs = 13
   - No. of filters in FB = 26

2. **Step 2: Compute average cepstral distance contour**
   Five average cepstral distance contours are computed taking 15, 20, 25, 50 and 100 feature vectors respectively on either side. The distance contours are then resampled to the original speech signal length and normalized to the maximum value.

3. **Step 3: Detection of peak indicating speaker or sound unit change point**

   - Smoothing using hamming window : The distance contours are smoothed by convolving with a hamming window of length 500 ms.
   - Locate peaks: In the smoothed contours mean of the distance values is calculated. Then an axis is set at the mean value and maximum distance value between a consecutive positive and negative mean crossing point is marked. In order to do that positive and negative mean crossing points are marked initially and peaks are located at the maxima

between every pair of mean crossing point. Thus five sets of peak points are obtained.

- Derive final set of change points: In the next step all the five sets of peak points are considered together and repeated points are removed. Thus finally one set of change point is obtained as shown in Fig **??**.

4.3 Performance Evaluation of ACD algorithm and Conclusion

The performance of the ACD algorithm using IITG-MV Phase III Speaker Recognition Database Part II (two speaker recognition). The data is conversational style speech between two speakers. There are two sessions in the database. Each session has 98 conversations. Each file has a corresponding label file containing the ground truth of speaker change points. This ground truth is to be used to measure the accuracy of the algorithms developed. The testing is to be done on both the sessions and the performances are to be reported accordingly. In addition, it is to be mentioned that each file in the database is about 12 to 15 minutes long. However, we can use only the first 1 minute of the speech files to report the performance.

The performance of ACD speaker change point detection algorithm is analyzed using three parameters, namely, false alarm rate, mis detection rate and identification accuracy. The peformance of ACD speaker change point detection algorithm is given in Table 4.3 for various tolerances.

From the Table 4.3, it can be noticed that The MDR is optimized to minimum by increasing the temporal tolerance. FAR for ACD speaker point detection algorithm is high, which does not depends much on the temporal tolerance. Along with change points due to speaker change, ACD captures the changes occurring due to speech to non speech transitions, onset and offset of glottal activities. Thus usage of speech non speech detection, glottal activity regions detection as processing approaches may improve the performance of ACD algorithm.

Table 4.1: Performance of ACD algorithm for speaker change point detection

Session No.	Temporal Tolerance (ms)	MDR (%)	FAR (%)	IDA (ms)
1	100	45.44	97.40	25.99
	250	15.08	96.49	70.17
	500	2.47	96.15	122.93
	1000	0.12	96.10	176.94
2	100	37.39	82.63	24.02
	250	13.73	81.91	59.44
	500	2.59	81.77	110.23
	1000	0.24	81.53	158.91

Table 4.2: 5 speech samples with best performance

Sl. No.	Tolerance (ms)	Sample Number	MDR (%)	FAR (%)
1	500	1310-1311	0.00	98.14
2		1322-1323	0.00	95.88
3		1324-2165	0.00	96.05
4		1326-1327	0.00	96.98
5		1329-1330	0.00	94.99
6		**Average**	**0.00**	**96.40**

Table 4.3: 5 speech samples with moderate performance

Sl. No.	Tolerance (ms)	Sample Number	MDR (%)	FAR (%)
1	500	1331-2138	20.00	99.28
2		1308-1309	4.55	96.48
3		1308-1309	9.52	96.23
4		1293-1075	10.00	98.38
5		1293-1075	6.25	97.26
6		**Average**	**10.06**	**97.52**

Table 4.4: 5 speech samples with poor performance

Sl. No.	Tolerance (ms)	Sample Number	MDR (%)	FAR (%)
1	500	1028-1353	13.33	97.33
2		1028-1353	25.00	99.48
3		1028-1353	25.00	99.39
4		2034-2199	20.00	99.28
5		1028-1353	12.50	98.74
6		**Average**	**19.16**	**98.88**

4.3.1 Performnace of the samples common across ACD, BIC, and KLD methods

Table 4.5: 5 speech samples with best performance

Sl. No.	Tolerance (ms)	Sample Number	MDR (%)	FAR (%)
1	500	2102-1362	0.00	95.13
2		2041-2135	3.57	95.33
3		1387-1388	5.56	96.99
4		1273-2131	0.00	94.73
5		1012-2143	0.00	95.70
6		**Average**	**3.04**	**95.57**

Table 4.6: 5 speech samples with moderate performance

Sl. No.	Tolerance (ms)	Sample Number	MDR (%)	FAR (%)
1	500	1077-1303	0.00	95.90
2		1108-1365	14.29	97.91
3		1064-1298	0.00	97.57
4		2103-1379	0.00	96.63
5		1063-1372	0.00	96.26
6		**Average**	**2.85**	**96.84**

Table 4.7: 5 speech samples with poor performance

Sl. No.	Tolerance (ms)	Sample Number	MDR (%)	FAR (%)
1	500	1026-1341	5.26	96.83
2		1268-2130	0.00	97.67
3		1370-1371	0.00	98.61
4		2134-2040	0.00	97.06
5		1038-1376	0.00	96.19
6		**Average**	**1.05**	**97.27**

Chapter 5

Speaker Clustering

5.1 Introduction

The second phase of the speaker diarization system is the speaker clustering. Performance of the speaker clustering algorithm is evaluated using performance matrices called Average Cluster Purity (ACP) and Average Speaker Purity (ASP). ACP reduces when a cluster includes segments from two or more speakers. On the contrary, ASP reduces when speech of a single speaker is split to more than one cluster. The best clustering scheme is the one which takes both factors into account.
The purity of cluster is defined as

$$P_i = \sum_{j=1}^{N_s} \frac{n_{ij}^2}{n_i^2}$$

and Average cluster purity

$$ACP = \frac{1}{N} \sum_{j=1}^{N_c} P_i . n_i$$

The purity of speaker is defined as

$$P_j = \sum_{i=1}^{N_c} \frac{n_{ij}^2}{n_j^2}$$

and Average speaker purity

$$ASP = \frac{1}{N} \sum_{i=1}^{N_s} P_j . n_j$$

where,
n_{ij}^2: Total number of frames in cluster i spoken by speaker j.
N_s: Total number of speakers.
N_c: Total number of clusters.
N: Total number of frames.
n_j: Total number of frames spoken by speaker j.
n_i: Total number of frames in cluster j.
In order to compare the accuracy (ACC) of a clustering method, the geometrical mean of ASP and ACP factors is computed as follows,

$$K = \sqrt{ASP * ACP}$$

The values of ASP, ACP and ACC are between zero and one. In optimal case when the clustering is done perfectly, ACP, ASP and ACC factors will be equal to 1.

5.2 Initial method of Speaker clustering

The initial method of speaker clustering adopted for this work is consisted of the following steps:

1. Initializing clusters with speech segments.
2. Computing distance or similarity of first cluster with all other clusters.
3. Merging closest clusters.
4. Recomputing the distances or similarities of remaining clusters.
5. Iterating steps 2 to 4 until stopping criteria is met.

To compute the distance of clusters BIC is used.

5.2.1 Speaker clustering using BIC

BIC is an appropriate measure for decision on the similarity of the clusters in speaker clustering. The use of the BIC is more straight forward in the clustering scheme than it is in the speaker change detection case. Let's assume we have a set of clusters $C_1, C_2, ..., C_k$. The problem to solve is to try to merge one of the clusters with another one, leading to a new set of clusters $C_1^{'}, C_2^{'}, ...C_{k-1}^{'}$ where one of the new clusters is the merge merge between two previous clusters. In order to see whether it is good to merge two clusters C_i and C_j, two models are built : the first model, say M_1, is a Gaussian model computed with the data of C_i and C_j, which leads to BIC_o. While, in case of second model, M_2, two different Gaussians are used used to model the data, one for C_i and another one for C_j, which leads to BIC_t. If the pair of clusters is best described by a single Gaussian, the $\Delta BIC = BIC_t - BIC_o$ will less than zero, i.e. negative, whereas if there are two separate distributions, implying two speakers, the ΔBIC will be positive **??**.

All segments provided by the change point detection algorithm are considered and labeled all as different cluster. Initially pairwise distance of first cluster with all the other segments are computed and closest clusters are merged. Two clusters are considered from the same speaker if their $\Delta BIC < 0$. The algorithm stops iterating when for an iteration all $\Delta BIC > 0$.
The steps involved in the algorithm is explained below.

1. **Step 1: Consider the segments obtained from speaker change detection** Here the segments obtained from speaker change detection algorithm are considered. Thus a set of change points are obtained.

2. **Step 2:Initialize clusters with all segments:** All N segments obtained from speaker change detection are marked as N different clusters.

3. **Step 3: Find ΔBIC of first cluster with all other segments :**
   Consider first cluster P and compute ΔBIC with all other segments i.e.
   $\Delta BIC(P,i)$ $i = 1$ to N, N is the total number of hypothesized speaker segments in the signal.

4. **Step 4: Find clusters having ΔBIC less than threshold and go on merging:**
   Here Threshold: $Th = 0$ and
   Stopping criteria: When all $\Delta BIC(P,i) > Th$, $i = 1$ to N and
   Initially $P =$ Cluster 1
   ITERATION P can be explained as given below.

   All cluster i which have $\Delta BIC(P,i) < Th$, are merged with cluster P.
   Cluster labeling is rearranged by removing labels of all merged cluster i.
   Recompute $\Delta BIC(P+1,i)$ $i = 1$ to M, where $M < N$.
   If any $\Delta BIC(P+1,i) < Th$, then go for next iteration else stop.

Table 5.1: Performance of BIC based clustering algorithm

Sl. No	ACP	ASP	ACC (K)
1	0.58	0.445	0.51

Update, Number of cluster= Number of iteration.

Update, Number of cluster= Number of iteration.

The performance of the BIC based clustering is given in Table 5.2.1.

5.2.2 Speaker clustering using GMM

A voice characteristics reference space is created using utterance dependent gaussian mixture model (GMM) and feature vectors are projected to this reference space to create some projection vectors. Cosine similarity between speaker segments are measured in this reference space in terms of the projection vectors.

Voice Characteristics Reference Space

Gaussian mixture modeling (GMM) is the predominant method for characterizing speaker-specific voice patterns. GMM can be applied in an unsupervised manner for the construction of a speaker-related reference space. Here, a GMM is created for each N speaker segments to be clustered, and the resulting N GMMs $\lambda_1, \lambda_2, \cdots, \lambda_N$, form a reference space with N bases $\phi_k = \lambda_k$, $1 \leq k \leq N$. For each segment X_i, the projection value on basis ϕ_k, $1 \leq k \leq N$ is then computed using

$$v(X_i, \phi_k) = logP(X_i|\lambda_k) - logP(X_i|\lambda_i)$$

Above equation is the normalized likelihood probability that segment X_i comes from the speaker characterized by GMM ϕ_k. Ideally, the value of $v(X_i, \phi_k)$ would be large if segments X_i and X_k are from the same speaker, and would be small otherwise. Thus a projection vector is created for every segments of the form,

$$v_i = [(v(X_i, \phi_1), (v(X_i, \phi_2), \cdots, (v(X_i, \phi_k)]'$$

Speaker Clustering using Voice Characteristics Reference Space

Let $X_1, X_2, \cdots, X_N$ denotes the N unlabeled speaker segments in a certain MFCC based feature representation, each of which is produced by one of the P speakers, where $N \geq P$, and P is unknown. The aim of speaker clustering is to partition the N segments into M clusters $C_1, C_2, \cdots, C_M$, such that $M = P$ and each cluster consists exclusively of segments from only one speaker.
After a reference space is constructed, each of the N segments, say X_i, is converted into a K-dimensional projection vector, $v_i = [(v(X_i, \phi_1), (v(X_i, \phi_2), \cdots, (v(X_i, \phi_k)]'$ on the space, where prime $(')$ denotes vector transpose, and $(v(X_i, \phi_k)$, $1 \leq k \leq K$ is a projection value that reflects the extent of how the segment X_i can be characterized by the basis ϕ_k. It is hoped that, if two segments, X_i and X_j are from the same speaker, say s_p, a majority of the projection values in v_i and v_j would be relatively similar in some sense, resulting in v_i being closer to v_j, instead of v_l for any segments X_l not from s_p.

Cosine similarity measurement and clustering

By associating each segments with a projection vector, the similarities between any two segments, X_i and X_j, are computed straightforwardly using the cosine measure between v_i and v_j:

$$S_u(X_i, X_j) = \frac{v_i . v_j}{||v_i|| ||v_j||}$$

Accordingly, segments deemed similar enough to each other can be grouped into a cluster.
All segments provided by the change point detection algorithm are considered and labeled all as different cluster. Initially pairwise cosine similarity of first cluster with all the other clusters are measured and closest clusters are merged. Two clusters are considered from the same speaker if their $Sc(C_m, C_l) < Th$. Here, C_m and C_l are two clusters with cluster number m and l and $Sc(C_m, C_l)$ is the cosine similarity between cluster C_m and C_l. The algorithm stops iterating when

for an iteration all $Sc(C_m, C_l) > Th$. Here the threshold Th is the mean of cosine similarity of all possible combination of segments in the speech signal. The steps involved in the algorithm are explained below.

1. **Step 1: Consider the segments obtained from speaker change detection** Here the segments obtained from hybrid algorithm for speaker change detection algorithm are considered. Speaker change detection algorithm combines average cepstral distance with BIC and KLD using the knowledge of speech non speech region obtained from glottal activity detection and silence classifier of UBM and PDS. Thus a set of change points are obtained.

2. **Step 2: Find projection vectors for all the speaker segments.** Create GMM reference space and compute projection vectors for all the speaker segments of the speech signal.

3. **Step 3: Initialize clusters with all segments.** All N segments obtained from speaker change detection are marked as N different clusters.

4. **Step 4: Find $Sc(C_m, C_l)$ of first cluster with all other segments :** Consider first cluster m and compute $Sc(C_m, C_l)$ with all other segments i.e. $Sc(C_m, C_l)$ $l = 1$ to N, N is the total number of hypothesized speaker segments in the signal.

5. **Step 5: Find clusters having $Sc(C_m, C_l)$ less than threshold and go on merging:**
   Here Threshold: $Th = mean(Sc(C_m, C_l))$ and
   Stopping criteria: When all $Sc(C_m, C_l) > Th$, $l = 1$ to N and
   Initially $m =$ Cluster 1
   ITERATION m can be explained as given below.

   All cluster l which have $Sc(C_m, C_l) < Th$, are merged with cluster m.
   Cluster labeling is rearranged by removing labels of all merged cluster l.
   Recompute $Sc(C_{m+1}, C_l)$ $l = 1$ to M, where $M < N$.
   If any $Sc(C_m, C_l) < Th$, then go for next iteration else stop.

   Update, Number of cluster= Number of iteration.
   Update, Number of speaker= Number of cluster.

The performance of the GMM reference space based clustering is given in Table 5.2.2.

Table 5.2: Performance of GMM reference space based clustering algorithm

Sl. No	ACP	ASP	ACC (K)
1	0.49	0.67	0.53
2	0.54	0.74	0.63

Bibliography

[1] M.H. Moattar and M.M. Homayounpour (2012) A review on speaker diarization systems and approaches. Speech Communication, Vol. 54, pp. 10651103.

[2] X. A. Miro, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals (2012) Speaker Diarization: A Review of Recent Research. IEEE Transactions on Audio, Speech and Language Processing, Vol. 20, No. 2., pp. 356-370.

[3] J. W. Hung, H. Wang, L. Lee (2000) Automatic metric-based speech segmentation for broadcast news via principal component analysis. In Proceedings of INTERSPEECH, pp. 121-124.

[4] M. Kotti, E. Benetos and C. Kotropoulos (2006) Automatic Speaker Change Detection with the Bayesian Information Criterion using MPEG-7 Features and a Fusion Scheme. In Proceedings of IEEE International Symposium on Circuits and Systems, Island of Kos.

[5] S. S. Chen and P.S. Gopalakrishnan (1998) Speaker, Environment and Channel Change Detection and Clustering via the Bayesian Information Criterion. Available via
http : //www.itl.nist.gov/iad/mig/publications/proceedings/darpa98/html/bn20/bn20.htm

[6] P. Delacourt and C.J. Wellekens (2000) DISTBIC: A speaker-based segmentation for audio data indexing. Speech Communication, Vol. 32, pp. 111-126

[7] L. Lu and H. J. Zhang (2005) Unsupervised speaker segmentation and tracking in real-time audio content analysis. Multimedia Systems, Vol. 10, Issue 4, pp. 332-343.

[8] C. Barras, X. Zhu, S. Meignier and J.L. Gauvain (2006) Multistage speaker diarization of broadcast news. IEEE Transactions on Audio, Speech and Language Processing, Vol.14, No.5, pp. 15051512.

[9] M.A.Siegler, U. Jain, B. Raj and R.M. Stern (1997) Automatic segmentation, classification and clustering of broadcast news audio. In Proceedings of DARPA Speech Recognition Workshop, Chantilly, pp. 9799.

[10] T.Wu , L. L. K. Chen , H. Zhang , P. R. China,Universal Background Models for Real-Time Speaker Change Detection. Available via
*http : //paginas.fe.up.pt/ ee*98235*/UNIVERSAL%*20*BACKGROUND%*
20*MODELS%*20*FOR%*20*REAL − TIME%*20*SPEAKER%*20*CHANGE%*
20*DETECTION.pdf*

[11] T. Liu, X. Liu and Y. Yan (2006) Speaker Diarization SystemBased on GMM and BIC. International Symposium on Chinese Spoken Language Processing, Singapore.

[12] Haris B C, G. Pradhan, A . Misra, S.R.M. Prasanna R.K. Das andR. Sinha (2012) Multivariability speaker recognition database in Indian scenario. International Journal of Speech Technology, Vol.15 pp. 441453.

[13] Murty K S R and Yegnanarayana B (2008) Epoch Extraction From Speech Signals. IEEE Transactions on Audio, Speech and Language Processing, Vol. 16, No. 8, pp. 1602-1613.

[14] Murty K S R, Yegnanarayana B and Joseph M A (2009) Characterization of Glottal Activity From Speech Signals. IEEE Signal Processing Letters, Vol. 16, No. 6, pp. 469-472.

[15] (2008) Temporal and Spectral Processing of Degraded Speech. In Proceedings of 16th International Conference on Advanced Computing and Communications, 2008. ADCOM 2008.

[16] G. Schwarz, ?Estimating the dimension of a model,? Ann. Statist., vol. 6, pp. 461?464, 1978.

[17] B. Zhou, J.H.L. “Hansen Efficient audio stream segmentation via the combined statistic and the Bayesian information criterion”, IEEE Trans. Speech Audio Process., Vol. 13, No. 4, (July 2005), pp. 467-474

[18] A. Tritschler and R. Gopinath, ?Improved speaker segmentation and segments clustering using the Bayesian information criterion,? in Proc. Eurospeech, 99, 1999, pp. 679-682.