Unsupervised Speaker Diarization

Akshay kumar
Electrical Engineering
akshakr@iitk.ac.in

Anurendra Kumar
Electrical Engineering
anurendk@iitk.ac.in

***Abstract*— Speaker Diarization is the first step in many early audio processing and aims to solve the problem "who spoke when". It therefore relies on efficient use of temporal information from extracted audio features. Since search for solution space is huge and often there is no ground truth available, it's a tough research problem. Most of the implementations done by different research groups fail in the case of varying number of speakers or high noise or high background music. In this project we explore the conventional techniques which involves hierarchichical agglomerative clustering and later shift to Integer Linear Programming clustering which gives state of the art results for unsupervised speaker diarization. We rely on Diarization error rate as evaluation metric for tuning the hyperparameters at different stages of pipeline.**

I. Introduction

The number of smart devices are increasing exponentially and so is the amount of data to process. In this era, storing these data in a structured way is a demanding research problem. Audio indexing which aims to organize content of multimedia using semantic information from audio data is broader class of problem for audio processing. Speaker diarization aims to label the segments of audio/video data with corresponding speaker identities. In short it solves the problem of who spoke when. Apart from audio indexing it has central application in speech research such as automatic speech recognition, rich transcription etc.
Challenges:-

- Number of speakers unknown
- Overlap of speech
- High variability in noise and background music

II. Steps

The technique of speaker diarization relies on a big pipeline with following steps :-

1) Feature Extraction
2) Noise and Music removal
3) Speaker segementation
4) Speaker Clustering

Since results of speaker segmentation often relies on speaker clustering, sometimes they use results of one another and act as semi-supervised method. An illustration of diarization pipeline is shown in fig. 1.

III. Preprocessing

A. Feature Extraction

The first step before doing any audio processing is extracting features from the audio data. Mel Frequency Cepstral Coefficents (MFCCs) are widely used features in automatic speech recognition system. Computing MFCCs from audio signal can be explained in following steps-:

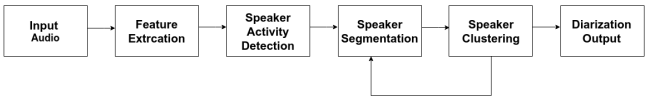

Fig. 1. Speaker Diarization pipeline

- Divide the audio signal into smaller frames. In smaller time scales audio signals are statistically unchanged. We divide them into 30ms frames with a hop of 10ms.
- Then for each smaller frames we compute the power spectrum of the signal. This periodogram estimate helps in identifying which frequencies are present in the frame.
- Above peridogram estimate still contains a lot of redundant information which is not used by speech recognition system. To remove this redundancy we take collection of periodogram bins and sum them using Mel filterbank. Number of filterbanks are around 20-40 (26 is standard). Mel filterbanks are collection of triangular overlapping windows. The first filter is very narrow tries to sum the periodograms around 0 Hz. The next filters are much wider and they care less about variation in power spectrum.
- When we have obtained filterbank energies then we take logarithm of them. This is motivated by human auditory system. In general we don't hear on a linear scale. This implies that loud variation in audio might not seem that loud to ear if we begin with initial loud volume.
- Finally we take DCT of the log filterbank energies. We only keep first 12 DCT coefficients out of 26. The reason behind this choice is that higher DCT coefficients represent fast changes in the filterbank energies and it turns out that these fast changes actually degrade performance of speech recognition system.

B. Speech activity detection

Speech activity detection is the task of detecting silent or non-speech part of the audio signal. An ideal speech activity detector should be robust to noise and performs in real time. It is useful in speaker diarization as it would be process whole audio signal if we could first filter it to remove non-speeches from the signal. In the state-of-the-art various speech activity detection algorithms have been proposed. They mostly vary based on the features used for

detection. Most commonly used feature is the short term energy (STE). But they can be easily degraded by noise present in the system. Hence much robust features are required for detection. In this we have used MFCC features, STE and first and second derivative of STE. Gaussian mixtures model for silence were trained using above features. Frames with high confidence of speech were used to train speech model. After each iteration, frames are classified into speech and non-speech. The frames with high confidence of speech and non-speech are used for training speech and non-speech model used for classification in next iteration. Using the above steps we can remove silence from audio signal. But it is unable to remove non-speech with high amplitude for example clapping, music etc. To remove these audible non-speeches or music we do further classification. We first divide the audio into smaller segments of 1s each. We extract 50 dimensional features from 20ms windows of 1s frame. These features were extracted using STE and zero crossing rate of the windowed signal. Using these features we create a histogram for each 1s frame. We model these histograms as chi-squared distribution. These histograms are compared against pre-trained universal database to label them speech or music.

IV. Speaker Segmentation

Speaker segmentation which is also known as acoustic change detection aims to detect speaker change such that each contiguous segment corresponds to single speaker only. To find if the two segments correspond to same speaker we have to define some notion of distance metric. This is followed by hypothesis testing and growing window strategy to detect speaker change.

A. Distance Metric

Let's say we have two speaker segments X_1 and X_2. We can define simple distance metric based on feature. However this does not capture variability in speech. Therefore it's assumed that each segment originate from a probability distribution. We model it using Multivariate normal distribution. Then some distance metric such as Bayesian Information Criterion, KL Divergence etc is used.

B. Hypothesis Testing

We have two hypotheses

- Null Hypothesis(H_0):- This assumes that both the segments come from different distributions and are not related.
- Alternate Hypothesis(H_1):- This assumes that both the segments come from same distribution.

C. Bayesian Information Criterion

It's the statistical measure used in statistical hypothesis testing . Let's say the model trained on segment X_1 and X_2 are M_1 and M_2 respectively. Then BIC for each segments are,

$$BIC(X_1, M_1) = logP(X_1|M_1) - \lambda k_1 logN_1$$

$$BIC(X_2, M_2) = logP(X_2|M_2) - \lambda k_2 logN_1$$

The first term is likelihood term while second term checks for complexity and therefore controls over-fitting. Similarly BIC of segments concatenating X_1 and X_2, let's say X, with respect to model M is calculated. Finally following BIC measure is calculated

$$\Delta BIC = BIC(M) - BIC(M_1) - BIC(M_2)$$

For multivariate normal distributions $M_1 = N(m_1, \Sigma_1)$, and $M_2 = N(m_2, \Sigma_2)$ with model size N_1 and N_2 having combined model $M = N(m, \Sigma)$, BIC is

$$\Delta BIC = (N_1 + N_2)log(Sigma) - N_1 log\Sigma_1 - N_2 log\Sigma_2$$

$$-\lambda(0.5 * (d + 0.5 * (d + 1)))logN$$

If the above calculated measure is negative, the two segments are from the same segment and should be merged. To avoid over-segmentation a threshold is kept. The value of threshold controls the number of change points and should be carefully tweaked.

D. Growing Window

Initially a small window is taken. If the feature vectors at the endpoints of window are better modeled by separate distributions, the midpoint is declared as speaker change point. In this case, the search is again started from the next segment. Otherwise the window is slightly increased and once again the above conditions are checked.

V. Speaker Clustering

Finally we need to cluster the different segments obtained from speaker segmentation. GMM's followed by hierarchical agglomerative clustering are often employed for it. However the current state of the art methods use i-vectors.

A. Gaussian Mixture Models with Universal Background Model

It models each speaker with a gaussian distribution,

$$p(x) = \sum_{i=1}^{N} w_i N(m_i, \Sigma_i, x)$$

$$s.t. \sum_{i=1^N} w_i = 1$$

However the number of segments to be cluster are very less. For our small audio of 5 minutes, we had only seven change points. Therefore training GMM for such small data is problematic. Therefore we use pre-trained model from Universal Background model (UBM) to train the model. The UBM is trained on large number of speakers both male and female to incorporate total variability in feature space. After training on UBM, each segment is assigned to some speaker model in UBM. The two segments are merged based on KL divergence or normalized cross likelihood ratio.

B. GMM supervector

Since there is high variability in covariance matrices, often the distance metric related to mean only also works better. The mean vector corresponding to different gaussians in mixture model is concatenated to form a GMM supervector.

C. I-vectors

The size of UBM is large and therefore GMM super vectors are high dimensional. To reduce computational issue and better accuracy, a lower dimensional representation using factor analysis was introduced, which is popularly known as i-vector in literature. Let m be the GMM supervector , M be the mean super-vector of the UBM, and x be the i-vector to be found, then

$$m = M + Tx$$

where T is a tall matrix which represents total variability space learnt on training data. We use MSR toolkit for extraction of i-vectors.

D. Hierarchical Agglomerative Clustering

It's a clustering technique based on local optimization. It merges two clusters until some optimal criterion is met. Any distance metric such as cosine or mahalanobis distance metric is utilized to see if the two segments belong to same cluster. Generally the optimal criterion is set to be inter-cluster distance to be greater than some threshold. We have used Mahalanobis distance metric,

$$D(x, y) = (x - y)^T W^{-1} (x - y)$$

Here W is the covariance matrix of the dataset.

E. Integer Linear Programming (ILP) Clustering

In the ILP clustering, the k-means problem is modified to obtain a set of clusters. Consider the set of binary decision variables: X_{ii}

$X_{ii} = 1$ indicates cluster i is leader cluster.

$X_{ij} = 1$ indicates cluster i is assigned to leader cluster j.

Now the optimization problem becomes,

$$min \sum_{i=1}^{N} X_{ii} + \frac{1}{\delta} \sum_{i=1}^{N} \sum_{j=1}^{N} d_{ij} X_{ij} \tag{1}$$

s.t.
$$\begin{aligned} \textstyle\sum X_{ij} &= 1 \quad \forall j \\ X_{ij} &\leq X_{ii} \quad \forall j \\ d_{ij} X_{ij} &\leq \delta \quad \forall i, j \\ X_{ij} &\in (0, 1) \quad \forall i, j \end{aligned}$$

The first term in objective function minimizes number of leader clusters (number of speakers), and the second is the total variation of all K clusters. The first constraint ensures that a segment is assigned to a single cluster. The second constraint ensures that a cluster centre is assigned to the same cluster. Third constraint puts a limit on the distance between cluster centre and cluster points.

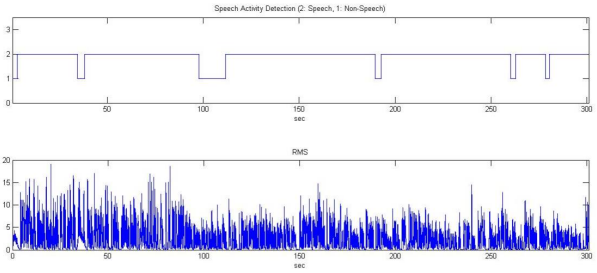

Fig. 2. Detection of speech and non-speech time intervals

VI. Evaluation

Diarization Error Rate (DER) is used for evaluation of automatic speech recognition system. DER is defined as percentage of the input signal that is wrongly labeled by the diarization output. In sth segment of duration $dur(s)$, N_{ref} and N_{hyp} are the number of speakers indicated by the annotations and hypothesized by the system respectively, and $N_{correct}$ is the number of speakers in sth segment that were a correct match between the annotation and hypothesis.

$$DER = \frac{\sum_{s=1}^{S} dur(s).(max(N_{ref}, N_{hyp}) - N_{correct})}{\sum_{s=1}^{S} dur(s).N_{ref}}$$

Contribution to DER comes from three factors namely, Missed speech rate (MSR), False alarm speech rate (FASR) and Speaker Error. When a speech is labeled as non-speech then that error comes under MSR. FASR is when a non-speech is detected as a speech segment. Speaker error is contributed due to speaker clustering and segmentation. This kind of error can be caused if a speaker change is not detected, oversegmentation, erroneously clustered. Sum of all three errors contribute to the DER,

$$DER = MSR + FASR + SpeakerError$$

VII. Experimental Setup

We use MSR toolkit for basic speech features and models.

A. Dataset

We experimented on interview dataset from youtube in which two people are talking [5]. Most of the time only one speaker is speaking with few overlaps and few instances of audience clapping. We have manually annotated to find ground truth for the dataset. We also ran google's voice id code to compare our result.

B. Results

The number of segments is largely dependent on hyper-parameters λ and threshold. Decreasing threshold leads to oversegmentation and needs to be carefully tweaked.

C. Diarization Error rate

We changed the values of λ in Integer linear programming clustering and plotted diarization error rate with respect to δ. Diarization error rate by google's voice id = 15.

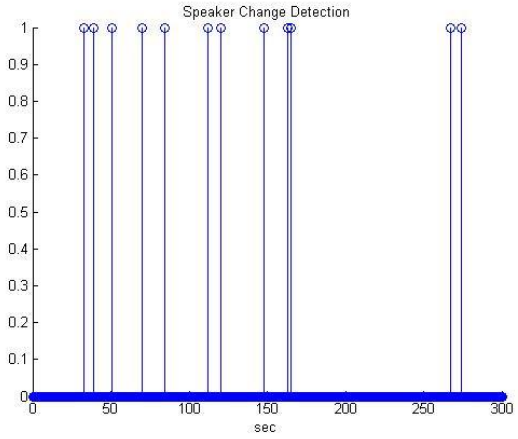

Fig. 3. Speaker change time instants

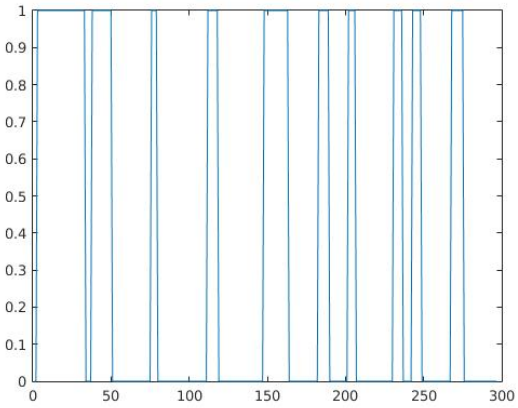

Fig. 4. Ground truth of speaker 1

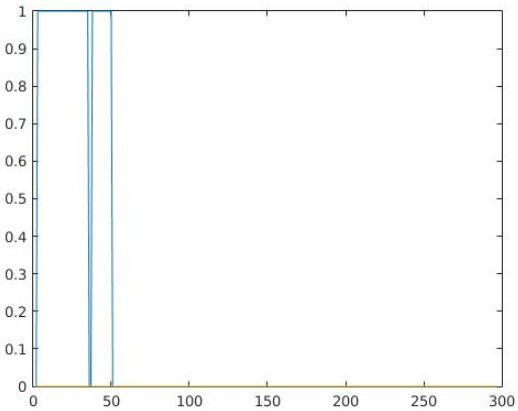

Fig. 5. Simulation results of speaker 1

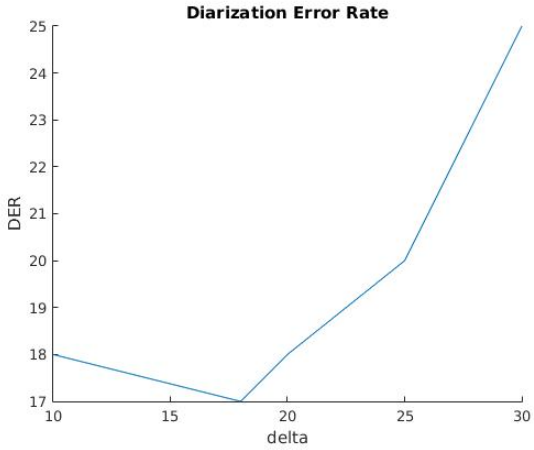

Fig. 6. Variation of DER with respect to delta

VIII. Observations and Inferences

- Our system can detects correct speaker when the segment length is greater than 15 second. This can be inferred from the plot of speaker 1.
- Each part of the pipeline has a big role and needs to be carefully tweaked.
- In speaker segmentation there is a tradeoff between oversegmentation and undersegmentation. If we don't do speaker segmentation and clustering iteratively, error due to speaker segmentation propagates to clustering
- ILP gives promising results.

IX. Conclusion and Future Work

We have got good results for simple dataset. However when number of speakers increase, it becomes a tough problem to tweak the hyperparameters. We hope to extend the framework for dataset containing multiple speaker. We also hope to extend it for multi-modal case and use video information for better speaker identification.

X. References

1) http://research.microsoft.com/apps/pubs/default.aspx?id=205119
2) Qin Jin, Kornel Laskowski, Tanja Schultz, and Alex Waibel, "Speaker Segmentation AND Clustering In meetings".
3) Miro, Xavier Anguera, et al. "Speaker diarization: A review of recent research." Audio, Speech, and Language Processing, IEEE Transactions on 20.2 (2012): 356-370.
4) Parthe Pandit, Matlab speaker diarization toolkit, GitHub repository, https://github.com/parthe/Speaker-Diarization-toolkit-MATLAB
5) Kangana Ranaut: The new face of India, https://www.youtube.com/watch?v=IvyWc0yB5Cw