Outliers Team- Speaker Diarization for Imbalance Class Problem

Aditya Jindal, Aniket Sharma, Ayush Kumar
{17817048,170111,180174}
Indian Institute of Technology Kanpur
{adityaji, anikets,ayushkmr}@iitk.ac.in
Supervisor: Vipul Arora

Abstract

To answer the question of who spoke when speaker diarization is a critical step for many speech applications in practice. And with recent advancements in deep learning technology, revolutionary changes have been made in this field.

Neural network-based audio-embeddings classification and segmentation is the most widely used method for speaker diarization problems and is currently state of the art. We have tried to address the problem of class imbalance in speaker diarization settings. We compare the performance of Spectral clustering with our proposed approach. In this approach we used deep audio embedding along with GANMM based oversampling followed by spectral clustering to achieve a difference of 9% DER on imbalance data.

1 Introduction

Speaker diarization is a task to label audio or video recordings with classes corresponding to speaker identity, or in short, a task to identify "who spoke when". It involves partitioning an audio stream with multiple people into homogeneous segments associated with each individual. It is an integral part of speech recognition systems and a well-known open problem.

Initially, it was proposed as a research topic related to automatic speech recognition, where speaker diarization serves as an upstream processing step. However, over recent years, speaker diarization has become an essential key technology for many tasks, such as navigation, retrieval, or higher-level inference on audio data.
But often the length of speech for different speakers is very skewed. This might result in an unsatisfactory performance of speaker diarization system. We shall try to overcome this problem in this project.

2 Related Work

In this paper, we are looking for Imbalanced data clustering problem where the number of samples for each class label is not balanced or where the class distribution is biased or skewed [4]. Since most of the standard clustering algorithms assume relatively balanced class distributions and equal misclassification costs, the class imbalance can be perceived as a form of data irregularity [2], and it could significantly deteriorate the performances of our method.

Coming to the baseline system [6], a text-independent speaker recognition network is used to extract embeddings from sliding windows of size 240ms and 50% overlap. A simple voice activity detector (VAD) is used to remove non-speech parts and partition the utterance into non-overlapping segments with a max length of 400ms. Then average the window-level embeddings to segment-level d-vectors and feed them into the clustering algorithm to produce final diarization results. The architecture for

computing embeddings has three LSTM layers and one linear layer. The network is trained with the state-of-the-art generalized end-to-end loss. Several clustering algorithms have been applied after extracting these embeddings, including K means, GMMs, and Spectral Clustering.
We all know that, since the seminal work by Goodfellow et al [3], there have been significant attempts to improve the performance of GANs. Following this success, the work in [7] suggested using a GAN mixture for clustering. The GANMM is hard to be trained by the classical EM procedure. Thus it is proposed to use the ϵ-EM procedure, where the ϵ introduces some controllable error so that the optimization of GANMM can continue.

3 DataSet Used

The problem of speaker diarization required us to solve the problem of 'who spoke when', so we needed the data that had audio files with transcripts that encapsulated the timestamps of when the speaker speaks. This was required in order to get the labels for training our Neural Network model. After testing many datasets, we finalized the AMI Corpus dataset [5]. This dataset contains 100 hours of recordings of meetings along with the transcripts of individuals with timestamps of their participation in the meetings. The audio files are an average of 20-30 min long .wav files with their transcripts in .xml files. Finally, we used the 'IS,' 'TS' & 'IB' meetings for training and testing our models. The selected audio files cumulatively correspond to 5 hours of audio recording with 96 different speakers.

4 Methodology

After getting the data, now we come to the core of the problem of speaker diarization. Our key procedure in brief summary is given below, the broad overview of the same is also represented by Fig. 2.

1. Get the timestamps of sliding window which assumed to have a single speaker.(As described in the above section.) / Applying Voice Activity Detection to get the timeline in audio in which someone spoke.
2. Process the audio and make Mel-Spectrograms/MFCC at the obtained timestamps.
3. Run a stacked LSTM model to get the audio embeddings (called d-vectors) of dimension 256.
4. Use clustering algorithms for e.g. K-Means, GMMs etc., by which extracted audio embeddings are clustered into the speakers.
5. For evaluation we use the Diarization error rate (DER) [1],
   where, $\text{DER} = \frac{\text{false alarm + missed detection + confusion}}{\text{total time}}$

For the processing of audio files to create the required dataset, we needed timestamps of the audio file, representing the speech part. To get this for training, we parsed through the transcripts and obtained the timestamps in which each speaker spoke. After that, we sampled the audio with a sampling rate of 22050 frames per second. For the unseen audio files, we trained a CNN Voice Activity Detector that takes mel-spectrograms of 1 second (dimension: 44x128) and does binary classification to predict whether the audio segment is speech or not. After identifying the speech sections of the audio, we partitioned them into partitions of 1 second each and extracted the MFCC features from all the partitioned intervals. The dimension of these MFCC features was 44x20, where the former and later figures represent the dimension of time and dimension of frequency, respectively. The same was done for all 24 training audio files, and finally, we got 16040 MFCC spectrum and constituting 96 speakers.

For obtaining the d-vector embeddings, we modeled neural networks and provided the 44x20 dimension MFCC features to it, with the output being 256x1 deep audio embedding. For training these embeddings, we tried various architecture which was based on Convolutional neural network(CNN) and Long short-term memory(LSTM). The goal here was to make the embeddings a speaker similar and embeddings of different speakers alike. To achieve this, we tested losses like Triplet Hard Loss, Triplet Semi-Hard Loss, and Contrastive Loss. We further experimented with various optimizers like Adam, SGD, RMSprop.

After obtaining the embeddings, to differentiate and classify speakers, clustering was performed. In the baselines, we implemented the three basic clustering algorithms that we integrated into our speaker diarization system.

We address the issue of class imbalance with each of these methods as follows:

KMeans Clustering: KMeans clustering is a simple unsupervised learning algorithm that clusters data into a K number of groups.
However, the performance of k-means algorithm tends to be affected by skewed data distributions, i.e., imbalanced data. They often produce clusters of relatively uniform sizes, even if input data have varied cluster size.

Gaussian Mixture Models: GMMs assume that the data points are formed from a mixture of Gaussian distributions with unknown parameters. It utilizes the Expectation-Maximization (EM) algorithm and groups data points by maximizing the posterior probability that it belongs in a cluster.

Spectral Clustering: In this approach[1], the data points are treated as nodes of a graph. An important point to note is that no assumption is made about the shape/form of the clusters as contrast to the Kmeans where the points assigned to a cluster are spherical about the cluster centre.
However, Spectral clustering is sensitive to how graphs are constructed from data. In particular, if the data has proximal and imbalanced clusters, spectral clustering can lead to poor performance.

Now one might use basic upsampling to correct class imbalance but that might encourage overfitting when observations are merely duplicated. In contrast with other oversampling methods, the proposed approaches avoid creating new instances in majority class regions.

GANMM: For our project, we use the GANMM model (algorithm 1) because of its ability to capture complex data distribution. Here, we employed an ϵ-E-step classifier trained from GAN generated samples and so the classifier may assign clusters with imbalanced instances. The imbalance could get reinforced through the procedure, as a result, some GAN models receive more and more data and the remaining fewer and fewer.
So to tackle this, we make sure that an equal number of samples are generated from each cluster to train the classifier. Since the number of training instances is significantly less the classifier is overfitting, and so the results obtained are not very accurate.

Algorithm 1 GANMM clustering algorithm

```
We have used WGAN along with $\epsilon$ -EM procedure.
Overview: Let N be the number of clusters.
We use N pairs of generator and discriminator along with a classifier.
Pre training: use K means/Spectral clustering to cluster data D initially
so we get $D^1_{(0)}, ..., D^N_{(0)}$ clusters
Then, we train seperate GAN model on each cluster using $\epsilon$-EM procedure as follows(see figure 3)
while convergence do
    $\epsilon$-E step:
    1.Sample data set S from N generators.
    2.Then train classifier C using these generated samples S.
    3.The classifier is used to assign clusters to each training data.
    M step:
    Train each GAN model with the clustered data for one generator and several discriminator iterations.
end while
we get $D^1_{(T)}, ..., D^N_{(T)}$ clusters
```

[1] refer: spectral-clustering, introduction-to-spectral-clustering

Here also we get similar problem of class imbalance as well and to fix this recurring issue, we use the fact that GANs can produce genuine data. So, we generate data for clusters having smaller number of instances and then add them to main data. Specifically, we first apply our GANMM algo 1, and get $D^1_{(T)}, ..., D^N_{(T)}$ as clusters. Now, we augment each of these clusters data set $D^i_{(T)}$ by adding the generated sample using trained Generator model-i. The amount of these generated samples, decided by the number of samples in each cluster. After this we cluster this augmented data with Spectral clustering. A summary of our proposed system is presented in Fig.1.

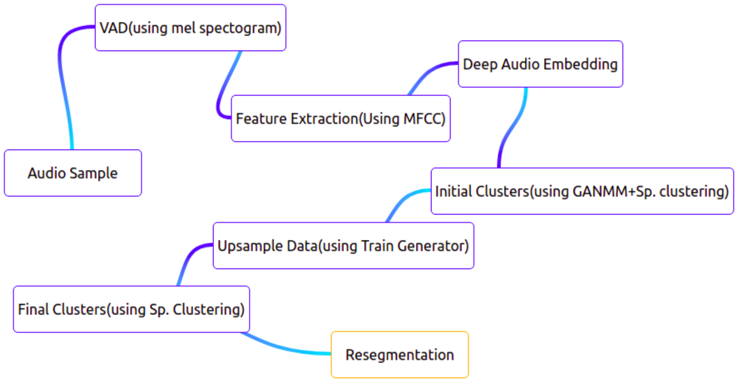
Audio Sample
VAD(using mel spectogram)
Feature Extraction(Using MFCC)
Deep Audio Embedding
Initial Clusters(using GANMM+Sp. clustering)
Upsample Data(using Train Generator)
Final Clusters(using Sp. Clustering)
Resegmentation

Figure 1: Schematic diagram of the proposed speaker diarization system

5 Results

The results obtained by the models are characterized by the Diarization error rate DER (in %). We used augmented unseen audio files from AMI corpus to test the models and averaged the DERs obtained. The DERs obtained for different clustering algo and loss functions are given in Table 1.

Clustering algorithm	Balance Data	Imbalance Data
K-means	17.50	35.21
GMM	18.67	30.55
Spectral clustering	15.10	29.80
GANMM+Spectral Clustering	**13.96**	**20.21**

Table 1: DER(%) on AMI test corpus, obtained for various models.

Finally, the best model is the GANMM+Spectral clustering, in which we augment(upsample) the data by synthetically generate the data using Generator. We see that it outperformed spectral clustering in both balanced as well as unbalanced dataset.

In appendix, Fig. 5, Fig.6 shows the clustering of embeddings in true labels as well as with the predicted labels. Fig.4 shows the timeline of the audio file with the speaker labels marked on it using both true labels and predicted labels.

6 Conclusion

Speaker diarization often has very skewed class ratios as one speaker might speak for a long time. In this regard clustering after extracting deep audio embeddings(d vectors) proves to be a difficult task but GANMM based oversampling followed by spectral clustering provides promising results. This method of clustering was able to cluster complex as well as imbalanced data. Several experimental results show that it outperforms traditional approaches like K means, GMM as well as new algos like Spectral Clustering. Although this method of oversampling to tackle cluster imbalance relies heavily on proper training of GANs which is very slow. Alternative methods of GAN training can be explored to make them fast.

References

[1] Hervé Bredin. “pyannote. metrics: A Toolkit for Reproducible Evaluation, Diagnostic, and Error Analysis of Speaker Diarization Systems.” In: *INTERSPEECH*. 2017, pp. 3587–3591.

[2] Swagatam Das, Shounak Datta, and Bidyut B Chaudhuri. “Handling data irregularities in classification: Foundations, trends, and future challenges”. In: *Pattern Recognition* 81 (2018), pp. 674–693.

[3] Ian J Goodfellow et al. “Generative adversarial networks”. In: *arXiv preprint arXiv:1406.2661* (2014).

[4] Guo Haixiang et al. “Learning from class-imbalanced data: Review of methods and applications”. In: *Expert Systems with Applications* 73 (2017), pp. 220–239.

[5] I. Mccowan et al. “The AMI Meeting Corpus”. In: *In: Proceedings Measuring Behavior 2005, 5th International Conference on Methods and Techniques in Behavioral Research. L.P.J.J. Noldus, F. Grieco, L.W.S. Loijens and P.H. Zimmerman (Eds.), Wageningen: Noldus Information Technology*. 2005.

[6] Quan Wang et al. *Speaker Diarization with LSTM*. 2018. arXiv: `1710.10468 [eess.AS]`.

[7] Yang Yu and Wen-Ji Zhou. “Mixture of GANs for Clustering.” In: *IJCAI*. 2018, pp. 3047–3053.

7 Appendix

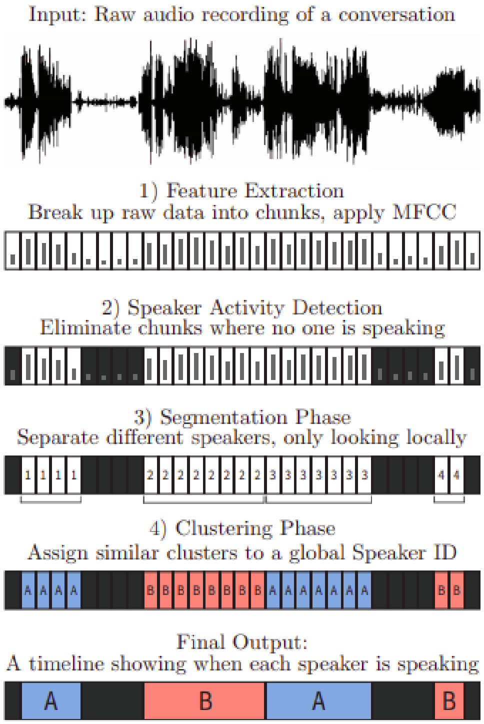
Input: Raw audio recording of a conversation
1) Feature Extraction
Break up raw data into chunks, apply MFCC
2) Speaker Activity Detection
Eliminate chunks where no one is speaking
3) Segmentation Phase
Separate different speakers, only looking locally
1 1 1 1
2 2 2 2 2 2 2 2
3 3 3 3 3 3 3
4 4
4) Clustering Phase
Assign similar clusters to a global Speaker ID
A A A A
B B B B B B B B
A A A A A A A
B B
Final Output:
A timeline showing when each speaker is speaking
A
B
A
B

Figure 2: Overview of the steps involved in bottom-up diarization algorithm.

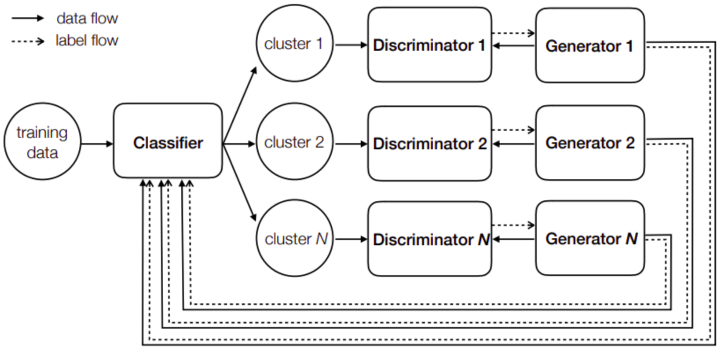

Figure 3: Schematic for GANMM

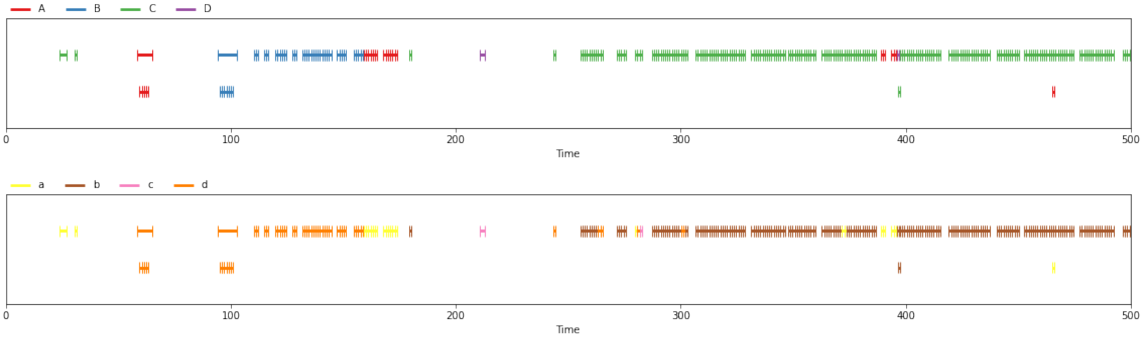

Figure 4: The timeline of a test file with true labels (upper figure) and predicted labels (lower figure)

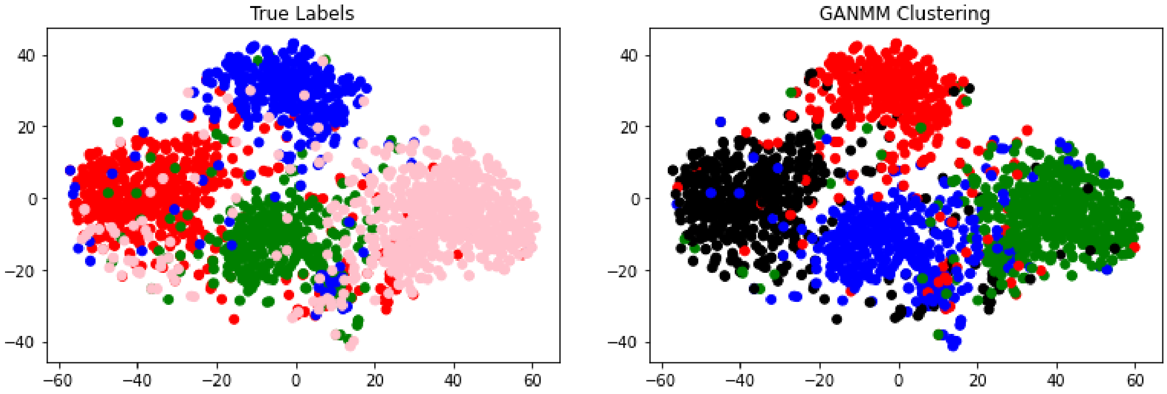

Figure 5: 2D display of the embeddings with true labels(left) and predicted labels(right) using GANMM Clustering on balanced data

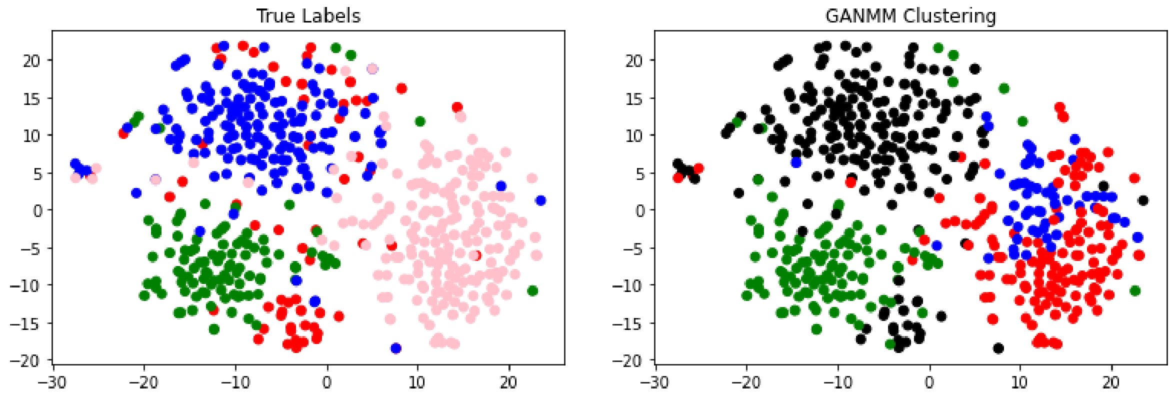

Figure 6: 2D display of the embeddings with true labels(left) and predicted labels(right) using GANMM Clustering on imbalanced data