E511-Signal Processing-Final Project

Jonathan Branam, Jamie Israel, Jinju(Hellen) Jiang

Abstract—In this project we evaluated and implemented state of the art techniques to solve the monaural speech separation (diarization) problem using a variety of approached. We investigated classification, masking, and deep learning approaches to separate a generated set of mixed speech recordings.

I. Introduction

There have been several recent developments in the use of deep learning to perform speaker diarization, or separation of multiple speech signals from a single source into separate homogeneous signals representing each individual speaker. This task is particularly challenging where the input is monophonic, there is no control over environmental conditions, the number of speakers is unknown and there are no samples of the speakers from which to train. Using the CSR-I(WSJ0) Complete dataset[?], we tested several algorithms for accomplishing monaural speech separation including classification (KMeans), various masking approaches (NMF, Ideal Ratio Mask[IRM], and complex Ideal Ratio Mask [cIRM]), and permutation-based deep learning approaches (Permutation Invariant Training [PIT] and utterance-level Permutation Invariant Training [uPIT]).

The motivating application is to separate agent and customer utterances from call center recordings, which involve short duration speech with occasional cross talk and at least one unknown speaker.

II. Literature Review

A. Jamie's Lit Review

B. Deep Learning Approaches

A seminal paper in applying deep learning to the task of speaker diarization was the Deep Clustering [?] approach proposed by Hershey, et. al. They divided the task of separating source signals into two steps. The first step is to use a deep neural network to learn a set of embeddings that produce a class-independent, low-rank approximation of the sources. These embeddings are trained to minimize the distance between embeddings in the same partition while maximizing the distances between embeddings in different partitions. The partitions used in their technique do not include class labels so the model does not assign a particular class to each embedding. The resulting embeddings can then be separated using simple clustering algorithms such as k-means. This is in contrast to previous work that relies on spectral clustering for segmentation which uses local affinity measures to optimize an objective function using spectral decomposition. The approach of Hershey, et. al. is typical in current research for deep learning: rather than using a complicated set of specially designed features the deep clustering approach uses a deep neural network to discover the best features for producing the desired partitions.

Results:

III. Dataset Preparation

A. CSR-I(WSJ0) Complete

For purposes of comparison with existing scholarship, each approach was developed based on the WSJ0 dataset, a corpus of Wall Street Journal text data organized by speakers.

B. WSJ0-mix

Developed based on...[?].

IV. Speech Separation Algorithms

A. KMeans

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

- Feature Extraction
- Speaker segmentation
- Speaker Clustering
- Evaluation

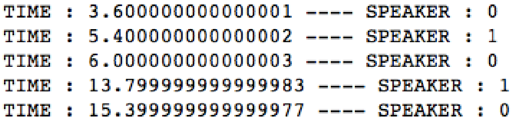
1) Feature Extraction: In this project, we used 3 approaches to extract features:

- A chroma vector (Wikipedia) (FMP, p. 123) is a typically a 12-element feature vector indicating how much energy of each pitch class, C, C#, D, D#, E, ..., B, is present in the signal. Then use mean of each element as one of input for clustering.
- The mel frequency cepstral coefficients (MFCCs) of a signal are a small set of features which concisely describe the overall shape of a spectral envelope. In this case, mfcc computed 13 MFCCs.The very first MFCC, the 0th coefficient, does not convey information relevant to the overall shape of the spectrum. It only conveys a constant offset, i.e. adding a constant value to the entire spectrum. Therefore, here I discard the first MFCC, then I scale the MFCCs such that each coefficient dimension has zero mean and unit variance.Then use mean of MFCCS as one part of input for clustering.
- We divide the audio signal into smaller frames. In smaller time scales audio signals are statistically unchanged. Then for each smaller frames we compute the power spectrum of the signal, and use it as the third part of input for clustering.

2) 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. In this case, I skipped this step and used the features extrated from above steps as input for kmeans clustering.

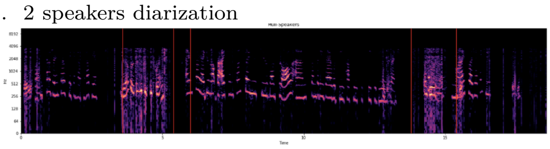
3) Kmeans Clustering: Since we knew that the audio from call center is the conversation between 2 speakers(customer service and customer), we set k=2 for kmeans clustering.
here it is the results for speaker diarization.

Fig. 1. 2 speakers diarization



```
TIME : 3.600000000000001 ---- SPEAKER : 0
TIME : 5.400000000000002 ---- SPEAKER : 1
TIME : 6.000000000000003 ---- SPEAKER : 0
TIME : 13.799999999999983 ---- SPEAKER : 1
TIME : 15.399999999999977 ---- SPEAKER : 0
```

Fig. 2. 2 speakers diarization



4) Evaluation[?]: Diarization Error Rate (DER) is used for evaluation of automatic speech recognition system.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. In this case, we don't have any labeled data to calculate the training error, we just simply used human ears to justify whether the diarization was good or not. For some training data, it works well, but some training data, it performed really bad.

B. NMF, IRM and cIRM

Nonnegative matrix factorization (NMF) is used to represent high-dminesional data as the product of two matrices (typically referenced as W and H). In the case of an audio signal, these matrices can be viewed as a representation of the signal's frequency spectrum (W) and the corresponding activations (H).

To perform NMF, we transformed isolated audio samples of two speakers (speaker 1 and speaker 2) to a frequency-time representation using short-time Fourier transform (STFT) before decomposing each signal using the following iterative update rules:

$$W = W \bigodot \frac{\frac{X}{WH} H^T}{1FxTH^T} \qquad H = H \bigodot \frac{W^T \frac{X}{WH}}{W^T 1^{FxT}}$$

Using the frequency matrices associated with the sample from each speaker, a new set of activations (W) was generated from the mixed signal composed of both individual samples. A magnitude masking matrix, reflecting the magnitudes associated with the frequency representation for each speaker at each time frame, was generated using this new set of activations and the following formula:

$$\frac{W_{S1}H_{(1:30,:)}}{W_{S1}H_{(1:30,:)} + W_{S2}H_{(31:60,:)}}$$

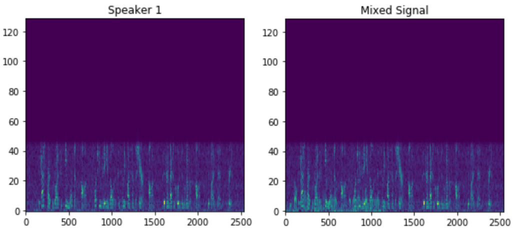
where S1 is the basis vector associated with the speaker 1 audio sample, S2 is the basis vector associated with the speaker 2 audio sample and H is the activation matrix generated from the combined speaker 1 and speaker 2 basis vectors.[?]

Using the STFT of a single speaker audio sample (speaker 1) and the STFT of the mixed audio sample that included speaker 1 [Figure 1], an IRM was generated based on the formula:

$$\frac{S^2_{(t,f)}}{S^2_{(t,f)} + N^2_{(t,f)}}$$

where S is the STFT generated from the isolated speaker 1 audio signal and N is the STFT generated fromm the mixed audio signal that included the same audio sample of speaker 1.[?]

Fig. 3. Spectrogram of isolated amd mixed audio sample from speaker 1


Speaker 1

Mixed Signal

Similarly, a cIRM was generated from the audio sample of speaker 1 and the same mixed audio sample using the formula:

$$\frac{Y_rS_r + Y_iS_i}{Y_r^2 + Y_i^2} \quad + \quad \mathrm{i} \quad \frac{Y_rS_i - Y_iS_r}{Y_r^2 + Y_i^2}$$

where S is the complex STFT representation generated from the isolated speaker 1 audio signal and Y is the complex STFT generated form the mixed audio signal that included the same audio sample of speaker 1.[?]

C. PIT and uPIT

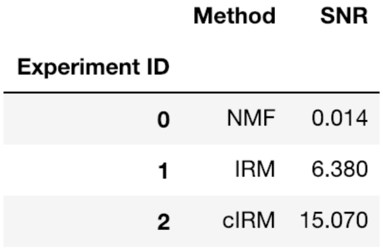
.. .... Add text here....

V. Results

A. Masking Methods

With the NMF method, we were unable to generate a substantial separation between the original isolated signal and the mixed audio. Both IMR approaches produced substantially better results with the cIMR method proving to be the best method for this type of separation on a consistent basis [Figure 2].

Fig. 4. Comparison of Average SNR for Masking Methods



	Method	SNR
Experiment ID		
0	NMF	0.014
1	IRM	6.380
2	cIRM	15.070

VI. Conclusion

...