AUTOMATIC SPEAKER DIARIZATION USING MACHINE LEARNING TECHNIQUES

Arun Chandrasekhar, Shashankar Sudarsan

[a9chandr,ssudarsa]@ucsd.edu
University of California, San Diego

ABSTRACT

In todays age of audio data proliferation, the analysis of audio data holds great relevance and significance. In this report, the authors propose a speaker diarization system for the UCSB speech corpus, using supervised and unsupervised machine learning techniques. The system includes four major modules: data preparation, feature extraction, data segmentation and the learning phase. The model was trained and tested using baseline supervised learning methods of SVM and Multilayered Neural Network and unsupervised method of k-means clustering. Additionally, the authors also present an ensemble combination of the features, which is found to have a superior performance. The results are documented and compared against the state-of-the-art diarization techniques.

Index Terms— diarization, deep neural networks, feature selection, classification

1. INTRODUCTION

The ubiquity of audio data around us in todays scenario presents a massive potential for information access. Be it recordings of important technical conferences, business meetings, news broadcasts, or even simple telephonic conversations, this data can help preserve important conversational moments. In addition to this it could offer other useful pieces of metadata that would make transcripts much more rich and useful. One such application would be audio diarization. Audio diarization is defined as the task of marking and categorizing the different audio sources within an unmarked audio sequence. On the flip side, owing to its lack of searchability, working on audio data is a tedious task. Going through hours of audio data to find such speaker specific information would require a lot of computational power. In todays burgeoning world of artificial intelligence, the application of machine learning techniques becomes the obvious solution to this issue. The model proposed in this paper utilizes supervised as well as unsupervised learning techniques to effectively perform the said task.

2. PROBLEM FORMULATION

At its core, this a clustering and recognition problem. Traditionally such problems are approached as unsupervised learning problems, although supervised approaches are also possible. The primary input to the system is a raw audio file, with recordings of conversations involving multiple speakers. The main input to the system is a raw audio file of a conversation, and the output is a timeline showing when each of the participants are speaking. In addition to the input, Other metadata, like the number of speakers can be directly fed into the system to aid the system and thereby save time. Additionally, for a supervised approach, part of the speech file is fed to the system as the training data. This portion of the data is concatenated with its corresponding label, which in this case is the hand-annotated with speaker times. The remaining portion of the data serves as the test set. The execution flow diagram of this system is presented in figure 1. As evidenced by the block diagram presented, the system is composed of four main functioning components, viz., i) Data Acquisition ii) Feature Extraction iii) Segmentation Phase iv) Training and Clustering Phase. The functioning of these modules is discussed in detail, in the forthcoming sections.

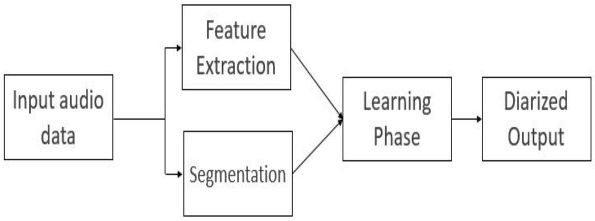


Fig. 1 Project Flow Diagram

3. DATA ENGINEERING

For complex applications such as the ones presented in this report, the raw audio file in its original form doesnt serve too many functions. This calls for some pre-processing, which is elicited in this section.

3.1. Data Acquisition

We retrieved the audio signal from the data using the wave class and associated methods in Python. The data was stored in stereo and we used only mono from the signal. The window size chosen was 1024. For speech signal 1024 is found to be the ideal size used my many methods involving speech signals. The main problem we faced was to extract the labels for our supervised learning methods, since the only data we received was the transcript to retrieve label information. From the transcript, we first retrieved the start times, and the name information and stored it in vectors. Since the number of frames we would get if we divided our entire speech signal with the chosen window is much larger than the number of name label vectors, we had to match the start times with our frames and repeat the name labels the corresponding number of times until we reached the start time of a new speakers. This resulted in getting name labels for each frame of our data. Further, to give it as input labels to our neural network, we had to convert it into one hot label vectors which would be appropriate for using Softmax layer for classification. We implemented by simply mapping the names to appropriate vectors

3.2. Segmentation

Speaker segmentation is otherwise known as speaker change detection and is very similar to detecting change in acoustic signals. We implement a variant of the KL distance method which makes a single run through the entire acoustic signal and the change-points are obtained. There are two broad categories of speaker segmentation algorithms metric based and non-metric based. The former is more popular and extensively used in speaker diarization algorithms. KL2 metric is a metric based segmentation algorithm to detect change of speakers. The KL2 is obtained by symmetrizing the KL distance.

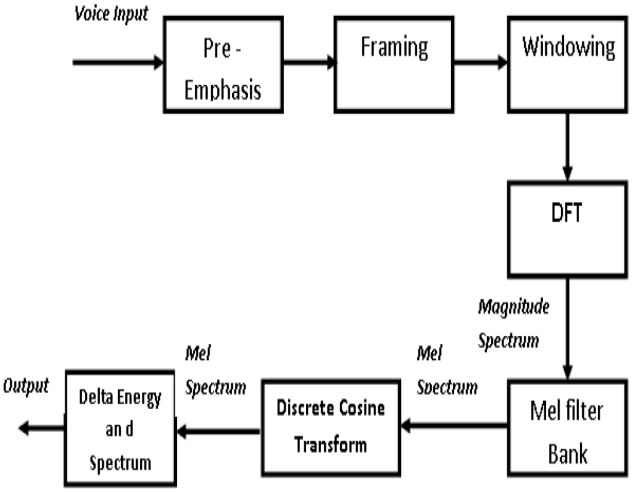
$$KL2(X;Y) = KL(X;Y) + KL(Y;X) \tag{1}$$

3.3. Feature Extraction

Extraction of the right features aids in the better performance of machine learning systems. This is especially true whilst dealing with audio data that presents many features. It was essential to select those features that best present the distinction between unique speakers in a conversation. Some of the features that best fit this description were found to be MFCC, Loudness, Spectral flatness and harmonics to noise ratio.

3.3.1. MFCC - Mel Frequency Cepstral Coefficients

Mel Frequency Cepstral coefficient or MFCC is based on human hearing perceptions which cannot perceive frequencies over 1Khz. In other words, in MFCC is based on known variation of the human ears critical bandwidth with frequency. MFCC has two types of filter which are spaced linearly at low frequency below 1000 Hz and logarithmic spacing above 1000Hz. A subjective pitch is present on Mel Frequency Scale to capture important characteristic of phonetic in speech. The overall process of the MFCC is shown.



3.3.2. Loudness

Loudness in this model is measure as the energy in each Bark Scale Critical Band, normalized by the overall sum. Bark Scale Critical Bands are 24 frequency bands, ranging from 20 Hz to 15500 Hz. This conversion is given by the following equation:

$$Bark = 13arctan(0.00076f) + 3.5arctan(f/7500)^2 \tag{2}$$

Loudness is dependent on both energy and frequency. With energy exhibited at a particular frequency by a person being speaker specific for the most part, loudness presents itself as a good feature for diarization.

3.3.3. Spectral Flatness

Spectral flatness is a feature used to characterize the amount of noise present in an audio signal. In simple terms, it gives the measure of noise plaguing the signal and consequently the likeliness of a feature to actually be a good characterizer of the signal. It is calculated as the ratio of the geometric mean to the arithmetic mean as shown in the equation below.

The model was trained using the individual features separately and later in the combined form by simply concatenating the features and flattening them into vector form.

$$Flatness = \frac{\sqrt[N]{\prod_{n=0}^{N-1} x(n)}}{\frac{\sum_{n=0}^{N-1} x(n)}{N}} = \frac{exp(\frac{1}{N}\sum_{n=0}^{N-1} ln\, x(n))}{\frac{1}{N}\sum_{n=0}^{N-1} x(n)} \quad (3)$$

4. MODELS

Upon segmenting the data and extracting the features, the training phase was implemented which is explained in this section.

4.1. K-Means

For unsupervised learning, k means clustering was used. K-means clustering, introduced by MacQueen in 1967, is a method commonly used to automatically partition a data set into K number of groups. It proceeds by selecting k initial cluster centers and then iteratively refining them as follows: (i). Each instance di is assigned to its closest cluster center. 2. Each cluster center Cj is updated to be the mean of its constituent instances. The algorithm converges when there is no further change in assignment of instances to clusters. In order to avoid the common pitfalls of k means, the seed assignments following the first one is done in such a manner that the new seed is distanced from the first by a probability weighted by the distance functions (k means++). This improves interclass variance. For the purpose of simplicity, the value of K was fed to the system as an imput. K means clustering was implemented in Python using scikit-learns KMeans module.

4.2. Supervised

As the name suggests, supervised learning algorithms work by providing target labels in the test phase, facilitating supervised classification in the testing phase. In this section, the supervised learning methods of SVM and Multilayer Perceptron for the purpose of speaker classification will be dealt with.

4.2.1. Deep Neural Networks

Deep Neural Networks are artificial neural networks with more than one hidden layer. The main purpose of increasing the number of hidden layers is to do away with the use of carefully hand-crafted feature engineering. In addition, it also introduces the necessary non-linearity required for complex logistic regression applications. Given the large number of features we gave as input and the amount of data available, we decided to use 3 hidden layers for our neural network. We used cross entropy as our loss function and used the gradient descent to train our weights. Since we coded our entire model in Python we extended to code to TensorFlow framework. Given the complexity of our model, implementing TensorFlow in GPU helped us in saving time. We used softmax at the output of each hidden layer. Since the labels were one hot vectors we seek to minimize the cross entropy. We ran the 1000 epochs to train the network. Our training-testing split was 0.8 and we performed batch learning with a batch size of about 100.

4.2.2. SVM

In classification and regression, Support Vector Machines SVM is the most common and popular method for machine learning tasks. Introduced by Vapnik in 1998, it works by mapping the input vector into comparatively higher dimensional feature space, followed by obtaining the optimal separating hyper-plane in higher dimensional feature space. In this method, a set of training examples is given, with each example marked with the category it belongs to, which in this case happens to be the speaker identity. Then, by using the Support Vector Machines algorithm, testing data is classified into the categories presented during training. The approach adopted for this method involved the use of linear kernels. Mathematically, this problem can be formulated as:

$$argmin_{w,b}(1/2) * ||w||^2, for\, i = 1, ..., no\, of\, classes \quad (4)$$

$$subject\, to\; y^{(i)}(w^T x^{(i)} - b) >= 1 \quad (5)$$

Conventional svm is a binary classifier. Since the problem demands a multi-class segregation, a OneVsRest approach was adopted wherein a given test data point was classified as belonging to a certain class or not. This was implemented in python using the scikits SVM module, by calling the OneVsRestClassifier with linear svc. The module was tuned by adding hyperparameters like L2 norm penalization and square-hinge loss function.

4.3. Ensemble

Upon testing the baseline models for proof of concept, their results were combined in an accuracy maximizing fashion. The combining operation was a simple pooling layer. This layer processes the features in a frame-by-frame basis. For each of these frames, the feature outputs were combined in a manner comparable to spatial pyramid pooling. The working of this layer is presented in the block diagram shown in figure 3.

5. RESULTS

The results of the aforesaid techniques are documented in this section. The models were trained on 81 minutes of audio data obtained from the Santa Barbara Corpus of Spoken English. The computations were performed on Intel Core i5-7200U CPU at 2.50GHz, running an NVIDIA GeForce

940MX Graphical Processing Unit. The results are furnished below.

5.1. K-means

Being an unsupervised learning algorithm, the entire data was fed to the algorithm for clustering. For the purpose of simplicity, the value of K was also pre fed into the system. The segmented clusters were then reformatted for temporal alignment. Upon performing these steps a Diarization error of 39.14% was observed for the combined feature model.

FEATURES	ACCURACY
Ensemble	**46.6113**

5.2. Neural Networks

A neural network, with three hidden layers was trained with the above mentioned data. The performance is tabulated as follows

FEATURES	ACCURACY
MFCC	67.1152
Loudness	3.4022
Spec Flat	19.1924
Ensemble	**69.2516**

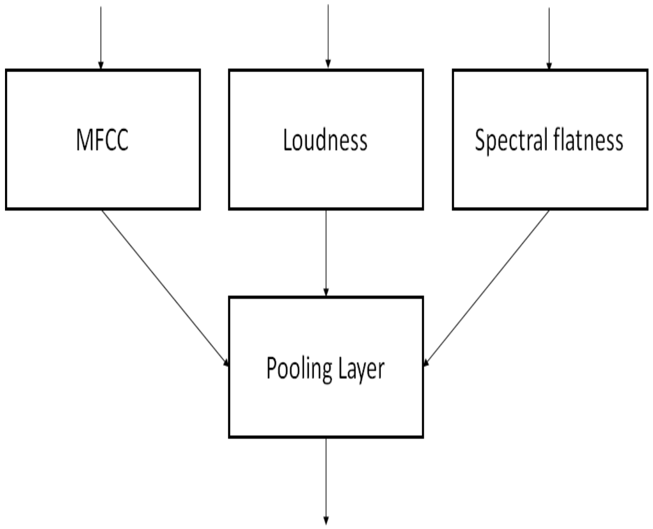
5.3. Support Vector Machines

The audio data was subsequently trained using the Support Vector Machine algorithm with linear kernel. To avoid overfitting, a penalization factor powered by L2 norm was incorporated. The results are observed as follows.

FEATURES	ACCURACY
Ensemble	**91.2**

5.4. Ensemble

The results for the combined feature model is presented below. As observed this method performs better that the baseline methods, hence justifying its inclusion



5.5. Discussion

The performance for different models vary widely. It was found that k-means had the least effectiveness among the given data. This could be partially attributed to it being an unsupervised learning algorithm. As opposed to the state of the art system, which presents an accuracy of 76%, the performance of the proposed system is rather paltry. This is ascribed to the very small size of the data, which is less than a tenth of the data size (72 hours,8GB trained on 16 GPUs for 3 days) of the state of the art system. With more data, it can be hypothesized that the system would perform better. As far as neural network is concerned, the performance of the system is comparable to the state of the art system, which produces an accuracy of 78% with three layers. The performance can be expected to improve with deeper architectures. In regard to support vector machines, which offered the best results, the system outperforms the state of the art systems. While this might vary with different datasets, for the proposed model, this performance is commendable. As observed, much of the pitfalls surrounding the proposed system stems from the inability to handle large computations owing to time and resource constraints, which is merely a physical issue that can be resolved.

6. CONCLUSION

Speech diarization and as an extension, speech recognition is an open-ended research problem. The application of machine learning techniques to augment such process has found some success as evidenced by the model proposed in this paper. Whilst the system is not quite as robust as the state of the art systems, it does present good potential. With the advent of deep learning in the past couple of years in this field, the deployment of something like a Recurrent Neural Network(LSTM) could also improve results. With additional resource like time and computational power, such techniques can be incorporated and consequently superior performance can be expected.

7. REFERENCES

[1] Friedland, Gerald, et al. ”Prosodic and other long-term features for speaker diarization.” IEEE Transactions on Audio, Speech, and Language Processing 17.5 (2009): 985-993.

[2] Anguera, Xavier, Chuck Wooters, and Javier Hernando. ”Acoustic beamforming for speaker diarization of meetings.” IEEE Transactions on Audio, Speech, and Language Processing 15.7 (2007): 2011-2022.

[3] Li, Chao, et al. ”Deep Speaker: an End-to-End Neural Speaker Embedding System.” arXiv preprint arXiv:1705.02304 (2017).

[4] Xavier Anguera Miro, ”Robust Speaker Diarization for Meetings”, PhD Thesis

[5] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, Speaker diarization: A review of recent research, IEEE Transactions On Audio, Speech, and Language Processing, vol. 20, pp. 356370, 2012.

[6] Jothilakshmi, S., Vennila Ramalingam, and S. Palanivel. ”Speaker diarization using autoassociative neural networks.” Engineering Applications of Artificial Intelligence 22.4 (2009): 667-675.

[7] Graves, Alex, Abdel-rahman Mohamed, and Geoffrey Hinton. ”Speech recognition with deep recurrent neural networks.” Acoustics, speech and signal processing (icassp), 2013 ieee international conference on. IEEE, 2013.

[8] Graves, Alex, and Navdeep Jaitly. ”Towards End-To-End Speech Recognition with Recurrent Neural Networks.” ICML. Vol. 14. 2014.

[9] Vinyals, Oriol, and Gerald Friedland. ”Towards semantic analysis of conversations: A system for the live identification of speakers in meetings.” Semantic Computing, 2008 IEEE International Conference on. IEEE, 2008.

[10] Mathieu, Benoit, et al. ”YAAFE, an Easy to Use and Efficient Audio Feature Extraction Software.” ISMIR. 2010.

[11] Shum, Stephen H., et al. ”Unsupervised methods for speaker diarization: An integrated and iterative approach.” IEEE Transactions on Audio, Speech, and Language Processing 21.10 (2013): 2015-2028.