Audio Tagging and Audio Event Detection

EE 603 : Machine Learning for Signal Processing
Advisor : Prof. Vipul Arora

Anirudha Brahma
Department of Physics, IIT Kanpur
anirbrhm@iitk.ac.in

Divyanshu Gangwar
Department of Electrical Engineering, IIT Kanpur
dvyanshu@iitk.ac.in

***Abstract*—This document is the report for the course project of MLSP 2021-22, where we discuss the various approaches we took to solve the problems of Audio Tagging and Audio Event Detection. We also discuss the other key decisions that we took and the reasons behind those decisions.**

I. Introduction to the tasks

The task provided to us is as follows: we would be given a spectrogram for a 10 second audio, we would have to first do the task of "Audio Tagging" which is to classify if the audio contains music, speech or both. And secondly we would have do the task of "Audio Event Detection" in which we have to output the specific times in the 10 second audio clip where music and speech was present.

II. Approach to the solve the problem

A. Rough classification of each window in the 10s audio

For an audio of 10 seconds, we would get the spectrogram to be of dimension (513,313) where 313 dimension refers to the time dimension. This means that the whole audio of 10 seconds is divided into 313 windows, and each window will be represented by a 513 length feature vector. We try to classify each window (313 in total) to either music, speech or silence to get an output vector of length 313 where each entry is music, speech or silence. So now our task reduces to classifying each 513 dimensional vector in one of the 3 classes.

B. Smoothening our classification

After we get our 313 dimensional output vector, we try to smooth-en out this 313 dimensional output vector using the concept of erosion and dillation inspired from the Computer Vision domain approaches.

C. Outputing Final Results

After we get our smoothed 313 dimensional output vector, we start from time = 0 and identify clusters of same label which are formed together. For example if we observe 30 labels of music together, we classify the audio clip as music for that duration of time. That is how we get the specific times in which the various events occur in the audio clip.
For Audio Tagging task, we claim that a music portion has to be of atleast 1 second in the clip for it to be called a music. Which means we identify clusters of 1 second which corresponds to roughly 32 frames. If we find one such cluster for music we determine that music is present in the clips. The same is done for speech class.

III. Creating the Dataset

We created our datasets for the 3 classes of music, speech and silence where we took 20 minutes of audio data for each of the 3 classes which were then merged to form our whole training dataset.
The training data was collected as follows :

A. Music

For the music class we collected audio clips from Youtube which consisted of mostly Indian Classical Instrumental music and Western Classical Instrumental music. A bit of other western genres of music was also added.

B. Speech

For the speech class we asked many of our friends to record short clips of them reading a passage which we later combined to form the music dataset.

1) ***An important Nuance****:* While forming the speech dataset we included a substantial part of the speeches from females, even though we knew from the recordings of the validation set that the test set will contain speeches from a male only, and including female speeches in our dataset could reduce the accuracy of our models on the test sets. Still we decided to go ahead with the substantial part of female speech keeping in mind AI ethics of designing gender-unbiased models specially when so much work is going on how to make AI gender unbiased. We chose ethics over accuracy.

C. Silence

The silence was created using zero vectors in Python.

D. Noise

Some noise was added to each of the 3 classes to make our dataset resemble the real world data more

IV. Models

Before going to the really heavy and resources intensive ML models we decided to first try Few Shot learning with some of the traditional Machine Learning Models on a very small subset of our training data. With the talks of how to implement small models on very small devices with resource constraints in the ML community we decided it would be a good idea to implement small models first.

A. Multinomial Logistic Regression Model

First we start off with the very light and simple to use model of logistic regression.
The input vector of size of (513,1) is mapped to a vector of (3,1) which represents the probabilities of that input vector being a part of that class. We trained our data on only 8,000

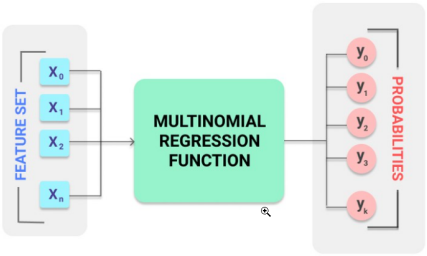

Fig. 1. Multinomial Logistic Regression Model

training samples which corresponds to roughly just 4 minutes of audio. We implemented the stochastic gradient descent method to train our model.
The model just exceed our expectations by giving an accuracy of 88.35% on the validation dataset, which just shows the true power of this simple model and how it can be used in while doing few shot learning, i.e when our training data is limited and we don't have a lot of resources.

B. K means clustering Model

We decided to take a relatively untravelled path and decided to experiment with an Unsupervised learning algorithm. We decided to experiment with K means clustering algorithm.
We divided our dataset into 3 parts, first is the training part, second was the rules based decision data and the last was the test data. First we assigned 3 clusters and trained the K means clustering with the EM Algorithm.
Now we have got 3 clusters, in which we do not know which cluster is for music, which is for speech and which one is for silence.
Here we used a rule based system with the "rules based decision data" to find out which cluster was for music, which one was for speech and which one was for silence.
This was done by carrying out the accuracy tests on the "rules based decision data" over all the possible combination of clusters and classes. For example, we assign cluster 2 to music, cluster 1 to speech and cluster 3 to silence. Now we calculate the accuracy of this model on the "rules based decision data". And choose the best combination of the clusters with the classes.

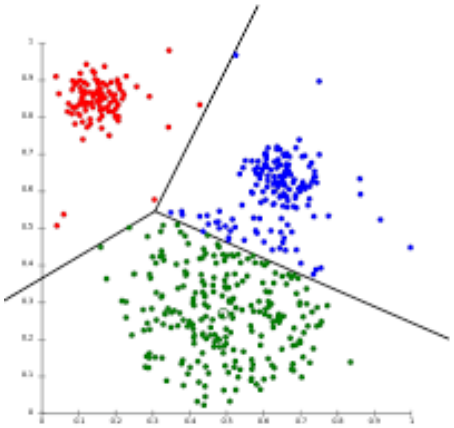

Fig. 2. K means clustering

We tested our model on our test set and it resulted in an impressive accuracy of 77.59 %. even with an unsupervised approach even though k means clustering is known to perform poorly on the high dimensional data.

C. Principal Component Analysis + Multinomial Logistic Regresion Model

While the MLR model worked very well, we felt that it might have not reached its full capacity due to the high dimension of our input vector and could still do better if we are successful in properly reducing the dimension of our input vector.
So we decided to implement the principal component analysis of our training set to extract only the most valuable components of our input vector.
We found that 99% of the variance in our data lies in the

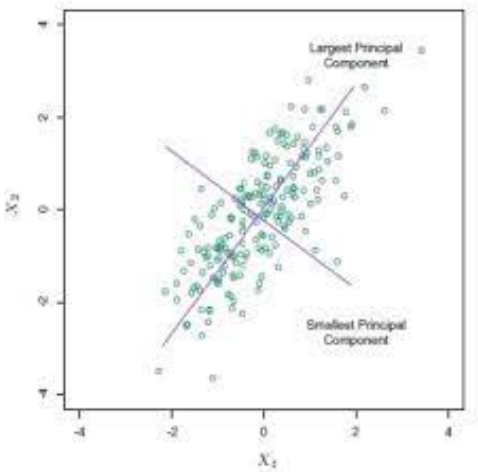

Fig. 3. Principal Component Analysis

100 most important eigenvectors of the covariance matrix. So we reduced the dimensions of our input vector from (513,1) to (100,1) and then applied a MLR classifier on top of it.
The results were superb as we achieved an accuracy of 98.05% on the validation dataset, while only training on 8000 samples. Which again just shows how effective these simple yet powerful algorithms be if we are able to analyze and modify the data properly.

D. K means clustering with Principal Component Analysis

As k means clustering is infamous for performing poorly on higher dimensional datasets, we thought that reducing the dimensions of the dataset would improve the performance of the model.
So we applied Principal Component Analysis on our dataset to reduce the input feature vectors to (100,1) dimensional vectors and then perform k means clustering on that.
This method unfortunately did not yield any significant improvement to the model with the test accuracy being 77.84%. Even though the model did not give a satisfactory result we thought it was a good try and included it in the report.

E. Feed Forward Neural Network Model

Now we turn our attention to models which can take advantage of the large dataset that we have, i.e deep learning models.

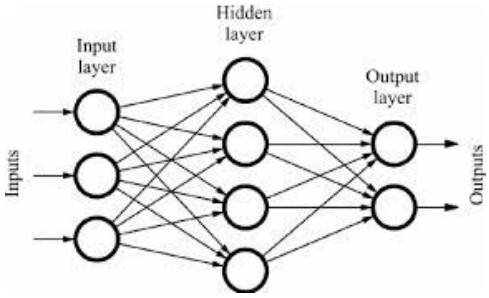

Fig. 4. Feed forward Neural Network

We experimented a bit with different combinations of architectures and finally decided to use 3 hidden layers apart from input and output layers for our model. We found this model to be very resource heavy models and takes a lot of time to generate an output from it given an input.
As expected this model performed the best among all giving us an accuracy of 99.32%.

V. Difficulties Faced

A. Dillation and Erosion

The raw output we get just from classifying all the frames of the spectrogram was very rough in classification, ie. there were many cases in which there was a single music label between many consecutive speech labels, which would disturb our final answer a lot.
So not doing anything to the raw output was simply not an option.
The problem of doing erosion and dillation with the help of softwares available online was that they were doing a very aggressive form of erosion, which was again not very useful for our final output.
We struggled a lot with figuring out how to do this smoothening of our output, and at last came up with a simple rule based system.
We would look at the immediate neighbours of a frame, and if both of the neighbours are same, we expect this frame to be the same as the neighbours as well.
The correctness of this system is debatable, but we could not come up with anything better.

B. Final Results

As we had reduced our task of Audio Tagging and Audio Event Detection to a task of classifying each frame to either music, speech or silence, we do not have a true idea of how our models perform on the above mentioned 2 tasks.
We only know that our best model classifies each frame of the spectrogram with 99% accuracy and theoretically it should do very well with the above two tasks, but in practice it does not do very well on the above 2 tasks.
A lot of thought was given on how to replicate the success in frame classification to the above 2 tasks, but we could not get a better idea other than the Erosion and Dillation Concept.

Acknowledgment

This course project would not have been possible without the support and teachings of Prof. Vipul Arora who have been very helpful in teaching the necessary concepts need to solve this problem and also very helpful in discussing the various approaches to this problem. We would also like to thank the Teaching Assistants who helped us with the logistics of the this project.
Last but not the least we would like to thank our peers from this course with whom we have had lots of fruitful conversations regarding how to approach this project.

References

[1] Multinomial-Logistic-Regression-Article - https://medium.com/@mygreatlearning/multinomial-logistic-regression-6615edda4315
[2] K-means-clustering-Article - https://medium.com/data-folks-indonesia/step-by-step-to-understanding-k-means-clustering-and-implementation-with-sklearn-b55803f519d6
[3] PCA-Article - https://medium.com/@aptrishu/understanding-principle-component-analysis-e32be0253ef0
[4] Zimek, A., Schubert, E. and Kriegel, H.-P. (2012), A survey on unsupervised outlier detection in high-dimensional numerical data. Statistical Analy Data Mining, 5: 363–387. doi: 10.1002/sam.11161
[5] Micha Wetzel, Audio Segmentation for Robust Real-Time Speech Recognition Based on Neural Networks