EE603 Course Project Report
Audio tagging and Audio Event Detection

By Dhruv Goyal (190291) and Videh Aggarwal (190960)

November 17, 2021

1 Abstract

This project report describes the processes, methods and models used to perform audio tagging and audio event detection. We started off with a simple linear classifier only for the purpose of benchmarking the results of our later models. In our case, music and speech do not overlap. Thus, even simple GMM models with discrete latent variables could perform the task at hand. Apart from basic ML models we also dwelled into Deep Learning by implementing simple Neural Networks, CNN and RNN models.

2 Data Preprocessing

Before starting to build our models, we had to create a dataset. We recorded three types of audio - music, speech and silence. Data variability was quintessential so we set out to download all forms of music - Indian and western classical, contemporary and a little bit of Rock too. For audio, we downloaded the lectures of various professors to get a wide variety of tones and timbre. Silence, on the other hand was programmatically made using python where we created a set of 2000 samples with progressively increasing values from 0.0005 to 0.1.

After downloading the above audio, we used the 'Praat' software to stitch all the individual pieces together, resampling to 16000 samples per second wherever required. Finally we ended up getting 30 min of each type of Audio. Then we computed the spectrograms of each of these (mfcc features too, in some cases) and further normalized them to get the values between 0 and 1. This final data was sent to the model.

3 Models built

In all the models discussed below we performed framewise classification. According to the specified hop and window size, the spectrogram of a 10s audio sample consists of 313 frames. All such frames were passed as samples into our model. Even while making predictions, the class of each frame was obtained by taking the maximum of the model's prediction array. This was further aggregated and certain methods described later were performed to get the outputs of the Audio tagging and Audio Event Detection tasks.

3.1 Linear

Even though this model is used for regression, we wanted to explore basic models and try to benchmark the comparison of the more advanced models later on using the linear model's accuracy metric. As expected, we did not get a very good accuracy on the test set and so decided to move on.

3.2 Neural Network

After our mediocre attempt with the linear model, we started to work on a neural network. From a simple linear one we went to deep learning models to increase accuracy. We added two hidden layers of 64 neurons each with relu activation. The final layer has 3 outputs : Music, Speech and Silence, with a softmax activation, since we are performing multilabel classification. We used stochastic gradient descent as the optimiser. We trained the model for 5000 epochs and achieved an accuracy of 91.42%.

3.3 Convolutional Neural Network (CNN)

Apart from the simple dense layers we decided to add some convolutional layers [2]. The kernels extract the characteristic features of our sample input which help in automatic feature learning. This time we used 3 convolutional layers with kernels of size 3x3 with relu activation. We also added drop out layers to prevent overfitting. We are using a flatten layer so that we can pass this data to dense layer with 64 neurons. We used the similar output as used in the previous (NN) model. Also, we used the adam optimiser, since it is faster. We trained the model for 3 epochs and achieved an accuracy of 90.18%.

3.4 Recurrent Neural Network (RNN)

In this model, we used LSTM [3] layers to take advantage of the sequential nature of the audio and to train our model using layers of previous time instances too. We used 2 LSTM layers with relu activation. This data was passed on to hidden layers with 64 neurons and finally to the output layer as above. Using adam optimiser we trained our model for 3

epochs with a batch size of 128. We achieved an accuracy of 98.12%. This is our most successful model.

3.5 Gaussian Mixture Model (GMM)

Taking a detour from the deep learning side we decided to build a latent variable based GMM model. We could have created one GMM model [4] with 3 latent variable corresponding to the 3 different types of audio, but that would only cluster the data and we would not be able to identify the clusters distinctly. Instead we created 3 different GMM models, one for each type of audio. For predicting the class we used Bayes' theorem,

$$p(c|x) = \frac{p(x|c)p(c)}{p(x)}$$

where c is the class and x is the given sample.

We assume p(x) to be constant for all classes and p(c) to be equal for all (equally likely events). Thus we only calculate p(x/c) for a class which is

$$\sum pi[k] * normal(x/mu_k, sigma_k)$$

4 Task 1: Audio event detection

The Aim of this task is to detect the onset and offset times of music and speech events in the given audio files. The given 10s spectrogram is divided into 313 time frames and for each time frame we are detecting the type of audio using our trained ML model. After this, In the cluster that we returned, we are consider 28 consecutive time frames (amounting to 0.96s) at one moment and counting the respective frequencies of each event type. Then we give this continuous segement of 28 frames the label with the maximum frequency.Bishop 2006

The next and the last step is to merge all the consecutive elements having the same event type. We will keep track of the onset and the offset times as well. The result is then written in a csv file.

5 Task 2: Audio tagging

The Aim of this task is to tag whether the given audio file is music, speech or both. The first step here is exactly similar to the first step of Task1. We form the cluster of 313 frames. Then we calculate the frequencies of music and speech type events. Using our observation and testing on multiple audio files we decided on a threshold equal to 50 for

best results. If the frequency of music or speech type events is greater than this threshold than we mark them as present. The result is then written in a csv file.

References

Bishop, Christopher M. (2006). "Pattern Recognition and Machine Learning". In: *Pattern Recognition and Machine Learning*, pp. 317–329.

([2]). `https://www.youtube.com/watch?v=WvoLTXIjBYU&t=564s`.

([3]). `https://www.youtube.com/watch?v=BSpXCRTOLJA&t=924s`.

([4]). `https://towardsdatascience.com/gaussian-mixture-models-implemented-from-scratch-1857e40ea566`.