AUDIO TAGGING AND SOUND EVENT DETECTION
TERM PROJECT FOR EE603: MACHINE LEARNING FOR SIGNAL PROCESSING

Anubhav Majumdar, Shivi Gupta

1. INTRODUCTION

In this report, we present the different methods carried out by us for two tasks: 1. Audio Event Detection and 2. Audio tagging. Given a spectrogram of a noisy audio containing music and/or speech separated by a few seconds of silence, we estimate the onset and offset timings of the music and speech components and also classify between the two. We carried out four methods which are all described in detail.

2. MODELS

2.1. Frame-wise Classification using a Neural Network

2.1.1. Frame-wise classifier

In this method, we make a frame-wise classifier, which classifies each frame of the spectrogram into 3 classes: music, silence and speech. The classifier is a 4 layer neural network (with dimensions 513, 64, 32, 3) and ReLu activation. The final layer has a softmax activation. We train this model over a large dataset which contains audio files (music/speech) separated by silence with added noise. It reaches a test accuracy of 0.69.

2.1.2. Merging

Now we merge the results of framewise classification. First we remove some isolated mislabeled datapoints. For example, taking an array called $predictions$, if

$$predictions[i-1] = predictions[i+1] \neq predictions[i]$$

Then we assign the value at $predictions[i]$ to be equal to its neighbors. Then, we calculate the onset and offset times using the indices of the $prediction$ array, at which there is a transition from or to 0 from a non-zero value. The cases for which $offset - onset < 0.5s$ are also removed.

2.2. RMSE Energy and Gaussian Mixture Models

2.2.1. Onset and offset times using RMSE Energy

Due to the presence of silence between two classes we can use this method of RMSE energy to detect time stamps for silence and then classify between music and speech. To find the onset and offset times we first calculate the frame-wise RMSE energies for our audio clip. Tuning manually, we find the best results for the $threshold$ to be

$$threshold = 0.965 * max + 0.035 * min$$

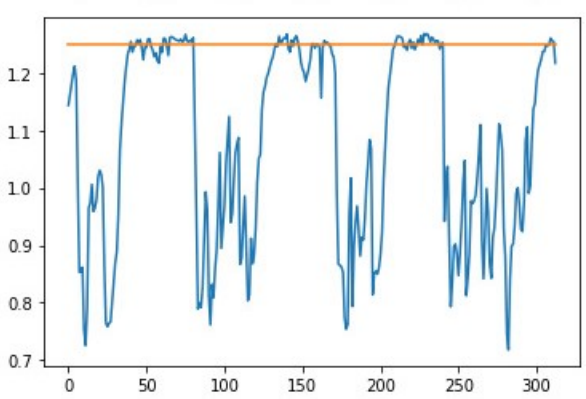

Fig. 1. Calculated threshold without filtering

Figure 1 gives the plot of RMSE energy with the calculated threshold.
Next, to improve on this, we first apply librosa's nnfilter on the spectrogram which replaces each datapoint with average of its nearest neighbors. Following this, we calculate the RMSE values, again followed by a length-k mean filter. Then we calculate the thresholds by the same formula above. While calculating onset and offset times, we remove the cases where $offset - onset < 25$. Figure 4 shows the spectrogram plotted along with onset and offset times.

2.2.2. Classification Using GMMs and PCA

We have used Gaussian mixture models with EM algorithm to classify between speech and music. The major variables that we have used in our model are $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$ and $\mathbf{k}$ where μ and σ represents the mean and covariance matrix corresponding to GMM's, π is the probability for a latent variable and k is the no. of classes in which we are divding our GMM model. The GMM model consists of three major functions **e_step**, **m_step** and **predict_proba** where in the e_step we find γ and π and the m_step we update our mean and covariance matrix. In the

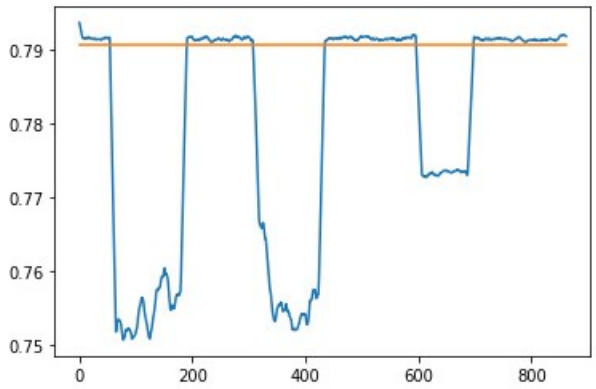

Fig. 2. Calculated threshold after filtering

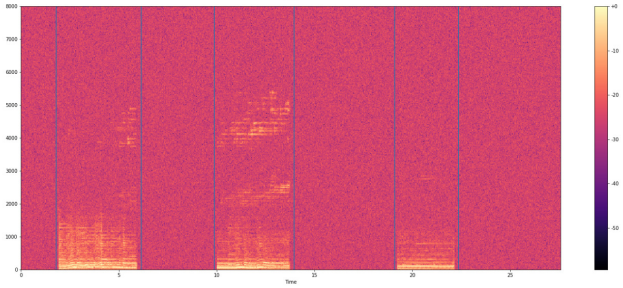

Fig. 3. Spectrogram with event detection

predict_proba function we calculate γ which is the probability of latent variable given our sample. For calculating the normal distribution we use multivariate normal distribution function defined in the scipy library. Since our spectogram contains 513 variables and the GMM model was not very fast we also implemented **PCA** reducing 513 dimensions to 16 dimensions. To find the final result we calculated the probability of each frame for music and speech using our GMM model and aggregated the results of each frame by using a simple majority function.

2.3. RMSE Energy and K-means

2.3.1. Onset and offset times using RMSE Energy

Here we use the same method as the above model for estimating the onset and offset times.

2.3.2. Classification using the K-means algorithm

We used the kmeans algorithm to classify between speech and music. Unlike GMM model in this model we didn't made seperate models for speech and music instead set the hyperparameter k with 2 and calculated the corresponding centroids. For classification the centroid which was more closer to music files was used to classify music files and the centroid which was more closer to the speech files was used to classify the speech files. In this code the is one major function fit where we are calculating the Euclidian distance of samples from the Centroids and updating accordingly. To consider initial value for the Centroid we just take a random samples which are provided.

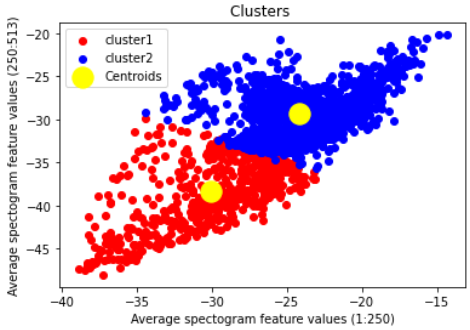

Fig. 4. Kmeans clustering of music and speech

2.4. RMSE Energy and Neural Networks

2.4.1. Onset and Offset times using RMSE Energy

Here we use the same method as the above two models for estimating the onset and offset times.

2.4.2. Classification using a deep neural network

For the classification part, we use a deep neural network trained on a large dataset to classify between speech and music. We use a neural networks with 4 layers (with dimensions 513, 64, 32, 2), the first 3 with ReLu activation and the output layer with softmax. The model classifies each frame into speech or music. Then we aggregate the framewise predictions and assign the final label corresponding to higher occurrences.
This is our best performing model

3. PREDICTIONS AND ACCURACY

Using the validation set which has 12 audio files with true labels, we make predictions using all the models. The following table shows the number of correctly tagged audio files (whether they contain music, speech or both)

Model	Number of correct tags
NN framewise classifier	3/12
RMSE energy and GMM	6/12
RMSE energy and K-means	12/12
RMSE energy and NN	12/12

Hence, both K-means and neural networks are able to work well for the classification into speech and music. The

framewise neural network classifier does not do well.
The root mean squared error calculated for the onset and offset times of the validation data for the best performing model turns out to be:

$$RMSE = 0.23s$$

Comparison of ground truth with our model's predictions for one of the validation audio files is given below:

Label		Onset Time		Offset Time	
True	Model	True	Model	True	Model
speech	speech	0.465	0.384	2.436	2.624
music	music	3.164	3.040	4.934	5.152
music	music	6.135	6.016	8.011	8.224
speech	speech	8.545	8.448	9.614	9.792

4. REPOSITORY FOR CODES AND MODEL WEIGHTS

All the codes and model weights are contained in this drive folder.