SPEECH / MUSIC DISCRIMINATOR

Steven Thomi[†]
EEE4114F
University of Cape Town
South Africa
[†]THMSTE021

***Abstract*—Digital Signal Processing and Machine Learning techniques are used in this project to realise an effective speech / music discriminator.**

I. Introduction

The focus of this project is to use digital signal processing methods, as well as the intrinsic statistics present in the signals, to classify a sound as either being speech or music. The samples used in this analysis are five seconds long and sampled at random at a frequency of 22050 Hz. These samples have been approved for use in research and can be found in [1].

This research is inspired by the Scheirer and Slaney [2] paper in which they were able to come to an accuracy of about 93 % in their speech / music discriminator system. Their research was, however, largely focused on the power spectral density components of the frequency-domain signal and the energy present in its frames. In contrast, I take a more time-domain and statistics-centric approach to form the speech / music discriminator system described in this paper. In doing so I aim to prove that a similar accuracy can be obtained using a completely different set of tools and way of thinking.

II. Problem Specification

The project description gives access to four samples of four categories of sounds:

1) music with no vocals
2) music with vocals
3) speech
4) speech with background music

These sound categories are to be classified as either being of speech form or of music form by the use of digital signal processing techniques. The aim of this project is; therefore, to uniquely identify any sound belonging to either of the four subcategories based on its characteristics, and identify the sound as either being speech or music without having prior knowledge of its identity.

III. Data

The sound samples used in this paper to develop and evaluate the system are provided in [1]. The music/speech examples were collected by Eric Scheirer and Malcolm Slaney at Interval Research Corporation. Permission for their use is granted.
The database used in [2], the "music-speech" corpus is a small collection of some 240 15-second extracts collected at random from the radio by Eric Scheirer during his internship at Interval Research Corporation in the summer of 1996 under the supervision of Malcolm Slaney.
I, unfortunately, have not been granted access to this entire database, but a small subset of it. The sounds used in this project comprise several examples of music, speech, and mixtures of the two. Each segment is 5 s long and is sampled at 22050 Hz.

IV. Evaluation Criteria

In order to measure how well the final system performs I will develop an accuracy metric to go with the system's predictions. The system will take in a set of test data points, make predictions on their class (i.e., speech, music) and calculate how many predictions match the correct output. The output of this accuracy metric is the percentage accuracy of the system.
Using the aforementioned evaluation criteria, I will be able to demonstrate and measure improvements in the system.

V. Approach

The first step to solving any problem is to understand the issue at hand. The problem description asks for the use of digital signal processing methods to identify the class of a sound. In order to satisfy this requirement, some sort of machine learning algorithm is needed to give the final word on where the data point belongs i.e., is it more like speech or more like music?

As a result of this observation, I have split this project into two categories: a digital signal processing (DSP) category and a machine learning (ML) category.

In the DSP category, I aim to identify the portions of the signal which uniquely identify it as either speech or music. In addition to this, I use the knowledge of digital systems to form a feature set to represent each point with. This feature set is aimed at differentiating speech from sound conclusively and consistently in order to create a strong decision boundary.

In the ML category, I use the identified feature set to train the algorithm, form a decision boundary, and evaluate test data along this boundary condition. In addition, I am faced with a new challenge on data partitioning - should I train with more data, test with more data, or use an equal data distribution among the training and testing stage? As a result of not having access to the entire sound database (as mentioned in the Data section), I choose to use an equal data distribution among the training and testing stage. Two sets of the four categories are allocated to the training phase of the algorithm and another two sets of the four categories are allocated to the testing phase of the algorithm.

A. DSP category

In order to understand the signal characteristics, I used **MATLAB** to plot the signal in the time domain and frequency domain. This code is provided in *Appendix A*.

1) Training Set 1: The time domain plot is illustrated below:

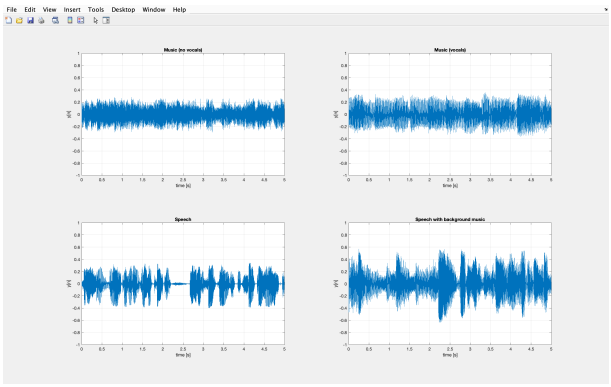

Fig. 1: Training Set 1: Time Domain Plot

The frequency domain plot is illustrated below:

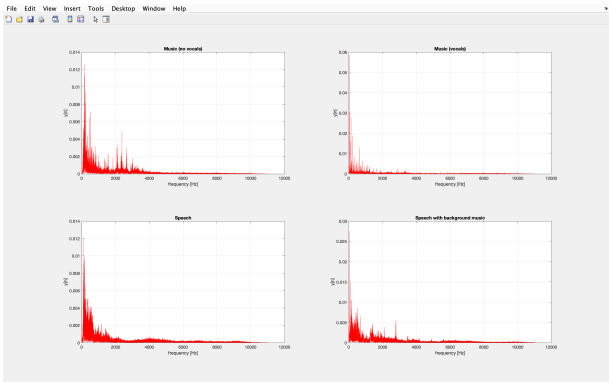

Fig. 2: Training Set 1: Frequency Domain Plot

2) Training Set 2: The time domain plot is illustrated below:

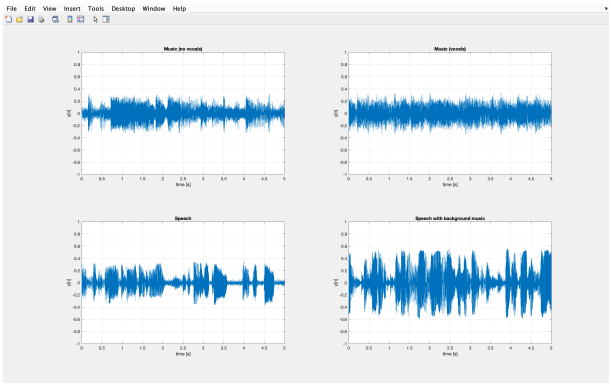

Fig. 3: Training Set 2: Time Domain Plot

The frequency domain plot is illustrated below:

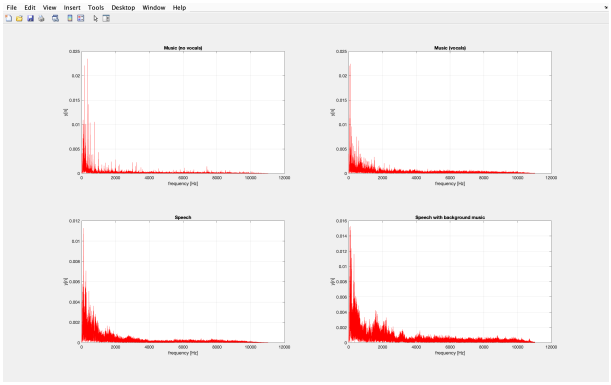

Fig. 4: Training Set 2: Frequency Domain Plot

The following observations regarding the plotted domains can be made using Figure 1, 2, 3, and 4.

- The frequency domain is more of an intra-category differentiator than an inter-category differentiator. The music with no vocals and speech with background music spectra look similar. In the same vein, the music with vocals and speech spectra also share similar spectral properties. The aim of this project is to distinguish between speech and music, we will therefore, not be working in this domain.
- The time domain; however, has a noticeable frequent lull. As noted by Joe Wolfe [3], for most of its duration, normal speech in most languages consists of quasi-periodic signals. The vowels (a,e etc), approximants (l,r etc), nasals (m,n) and some of the signal required to identify plosives, (b,g etc) are voiced: they involve quasi-periodic vibration of the vocal folds. This quasi-periodic nature of speech will form the basis of our investigation.

The time-domain data points are exported from the **MATLAB** implementation, and into text files. The ML category subsequently liaises with these files.

Using the time-domain signals, the absolute value mean of the signal is calculated and scaled by 0.5. In this case, 0.5 is a

hyper-parameter used to estimate a signal's ground level.
The feature to be used in our ML analysis is based on the quasi-periodic nature of speech. Instead of just counting the number of points below the $0.5 \times mean$ line, we count the maximum number of **consecutive** points below $0.5 \times mean$ line. The result is the single feature we will use to evaluate our signals.

B. ML category

In order to derive the identity of the sounds, I will be using a k-Nearest Neighbour (k-NN) algorithm. The k-NN algorithm is perfectly suited for our analysis due to the small data-set present - the algorithm will be able to easily keep track of both training and testing data without exhausting the computational resources available. The k in our analysis is set to 1. In addition to this, the distance of our single feature system is taken as the difference between the $0.5 \times mean$ value of the two points.

The feature derivation (in the DSP category) and the development and evaluation of the k-NN algorithm are completed using jupyter notebooks, the into-ml kernel, and the miniconda virtual environment provided in the Machine Learning practicals. The notebook is linked in *Appendix B*.

VI. Results and Analysis

The k-NN algorithm is trained and tested using the $0.5 \times mean$ feature with a k set to 1.

The trained model is illustrated below:

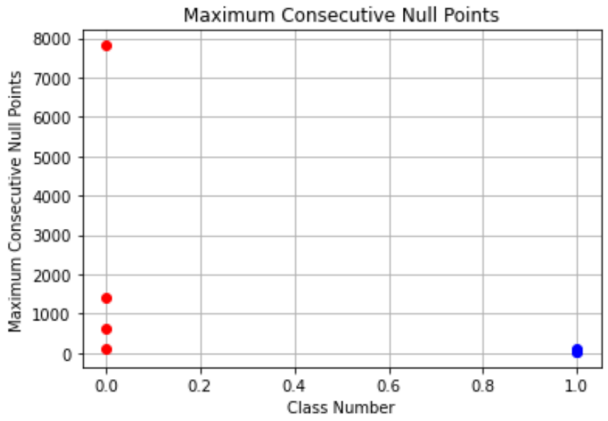

Fig. 5: Training Set 2: Frequency Domain Plot

As evident from Figure 5, the feature selected strongly differentiates the speech and music categories. The music data points are closely packed together, illustrating there is little to no silent breaks present. NB: The horizontal axis represents the class of the data point (i.e., Class 0 represents speech, Class 1 represents music).

The accuracy of the system is tabulated below:

TABLE I: Speech / Music Discriminator Accuracy

Algorithm	Accuracy
k-NN	**87.5** %

In order to attain a higher accuracy, it would be effective to increase the training of the algorithm by providing it with more test data, exploring other machine learning algorithms which strive to use the provided data more effectively (i.e., k-Fold Cross Validation), and use a higher k value in our k-NN evaluation to eliminate the effect of outlier values.

VII. Development

In order to improve the effectiveness of the machine learning algorithm, it would be recommended to use a larger amount and a more diverse range of sounds while training the algorithm.
A larger amount of data in training will make the algorithm more discerning to different types of test points (as long as precaution is taken to not overfit the algorithm). A more diverse range of musical genres should be explored in the training. How would the algorithm fair if it were tested on speech-sounding songs (i.e., Jimi Hendrix's 'Red House', or Frank Sinatra's 'That's Life')? A more diverse range of training points needs to be employed to ensure the algorithm is capable of maintaining a consistently high performance.
In addition, the rhythm of the underlying sound needs to be tested. This quantity would make for a good feature and serve to improve the algorithm's fitting.

VIII. Conclusion

The final system attains an accuracy of 87.5 %, similar to that reported in [2]. The speech / music discriminator system has; therefore, successfully been implemented using statistical properties of the time-domain signal.

IX. References

[1] E. Scheirer, M. Slaney, Music/Speech. Interval Research Corporation, 2002. Available at: https://www.ee.columbia.edu/ dpwe/sounds/musp/

[2] E. Scheirer, M. Slaney, Construction and evaluation of a robust multifeature speech/music discriminator. Proc. Int. Conf. on Acous., Speech and Signal Processing, Munich, Germany: 1997. Available at: https://engineering.purdue.edu/ malcolm/interval/1996-085/ SpeechMusicICASSP97.pdf

[3] J. Wolfe, Speech and music, acoustics and coding, and what music might be 'for'. Sydney, Australia: 2002. Available at: https://phys.unsw.edu.au/jw/ICMPC.pdf

X. Appendices

A. Appendix A: MATLAB Source Code

```
% Read in the first .wav file.
[y1, fs1] = audioread('Music (no vocals)_1.wav');
dt1 = 1/fs1;
t1 = 0:dt1:(length(y1)*dt1)-dt1;

% soundsc(y, fs);  % Here is where we actually play the music from the speakers.

% Plot the first waveform.
figure
subplot(2,2,1)
h1 = stem(t1, y1, 'filled');
set(h1, 'Marker', 'none')
xlabel('time [s]')
ylabel('y[n]')
title('Music (no vocals)')
axis([0 5 -1 1])
grid on
```

Listing 1: Time Domain Wave 1

B. Appendix B: Jupyter Notebook

The jupyter notebook can be found here: **GitHub Repository**