1.0 Performance Evaluation

Section **??**outlined the motivation for MFCCs from the audial, cognitive-psychological perspective as well as how to compute them using spectral and cepstral analysis. This section aims to discuss some of the work done by this project to determine how well MFCCs perform in classification and regression problems compared to other typical features (see section **??**). It will describe the methods used to compare these features and interpret the results of those tests.

1.1 Feature List

MFCCs have been shown to perform well in speech recognition tasks. The purpose of this performance evaluation is to compare the performance of MFCCs against a number of other statistical and musical features of audio signals. Table **??** lists and describes these features. This list is by no means exhaustive, but it aims to give a wide range comparisons. There are 4 time-based features, 9 frequency-based features, 12 musical-based features, and the 13 MFCCs for a total of 38 features.

1.2 Feature Aggregration

Table 1: Table of features used in performance evaluation tests.

Zero-Crossing Rate	Number of times signal crosses zero $\text{zcr}(x[t]) = \frac{1}{N-1}\sum_{t=1}^{N-1}\mathbb{I}\{x[t]\,x[t-1]<0\}$
Energy	Energy of discrete time signal $\text{energy}(x[t]) = \frac{1}{N}\sum_{t=1}^{N-1} x[t]\,x^*[t]$
Root-Mean Squared	Quadratic mean of signal $\text{rms}(x[t]) = \sqrt{\frac{1}{N}\sum_{t=1}^{N-1} x^2[t]}$
Energy-Entropy	Shannon Entropy of sub-divided windows ($n = 10$) $\text{H}(x[t], n) = -\sum_i^n \{e_i, \ln e_i\}$ $e_i = x[t]\,/\text{energy}(x_i[t])$
Spectral Centroid	Center of mass of spectrum (Hz) $\text{centroid}(X[n]) = \sum_{n=0}^{N-1} X[n]f(n)/\sum_{n=0}^{N-1} f(n)$
Flatness or Wiener Entropy	$\text{flatness}(X[n]) = \exp\left(\frac{1}{N}\sum_{n=0}^{N-1}\ln X[n]\right)/\frac{1}{N}\sum_{n=0}^{N-1} X[n]$
Spectral Entropy	Energy-entropy of spectrum (see above) $\text{H}(X[t], n) = -\sum_i^n \{e_i, \ln e_i\}$
Spectral Mean	Average of the spectrum (Hz) $\bar{X} = \sum_{n=1}^{N} X[n]$
Spectral Variance	Statistical variance $\text{Var}(X[n]) = \sum_{n=1}^{N}\left(X[n] - \bar{X}\right)^2$
Spectral Kurtosis	Fourth standardized moment $\text{Kurt}(X[n]) = \bar{X}_4/\text{Var}(X)^2$
Spectral Rolloff	85%-percentile of spectral energy
Spectral Skewness	Measure of left/right skewness $\text{Skew}(X[n]) = \bar{X}_3/\text{Var}(X)^{(3/2)}$
Spectral Spread	Variance about spectral centroid (above)
Chroma Coefficients	Maximum normalized histogram of frequency bins centered around each of the 12 semitones $C, C\#, \ldots, B$
MFCCs	Mel-Frequency Cepstral Coefficients (see ??)

Table 2: Second order features breakdown.

38	Mean of each feature
38	Variance of each feature
38	Predicted BPM (Beats per minute) for feature
38	BPM Confidence on each feature
1	Aggregrated expected BPM
1	Aggregrated expected BPM confidence

1.3 Beat Extraction

In order to extract some information about the long-term repetition of a value-series, there are a few things that can be done [**?**]. One possibility is constructing a recurrence plot (see figure **??**) for the feature vector. A recurrence plot is essentially an image where the pixel at the i, j coordinate is given by equation **??**.

$$R\left[i, j\right] = \text{sim}\left(x[i], x[j]\right) \tag{1}$$

Where the similarity measure is typically distance. Recurrence plots are always symmetric and their visual structure, specifically the diagonal strokes, encode the repetition in the signal x. However, as pointed out by [**?**], contruction and analysis of recurrence plots are highly non-linear; at least $O(n^2)$. Therefore, recurrence plots are not computationally feasible on large scales.

Another very common option are comb-filters [**?**]. However, like recurrence plots, comb filters are inherently computationally slow (typically $O(n^2)$).

As such, a fast, $O(n)$ algorithm was developed as part of this project for BPM prediction. The algorithm is illustrated in figure **??**. The core idea of this algorithm is that distances between adjacent peaks should be evenly spaced if there is a consistent tempo to the audio signal. For each feature:

1. Perform delta peak detection using Eli Billauer's *peakdet* algorithm originally developed for matlab [**?**].
2. Ignore all minimums, only look at maximums (red dots in figure).
3. Construct a histogram based on the distance between adjacent peaks.
4. The predicted BPM is the largest column in histogram.
5. The BPM confidence is the ratio of the largest histogram to the total number of data points.
6. The aggregate BPM and confidence is given by 4. and 5. on the combined histogram for all features (see figure **??**).

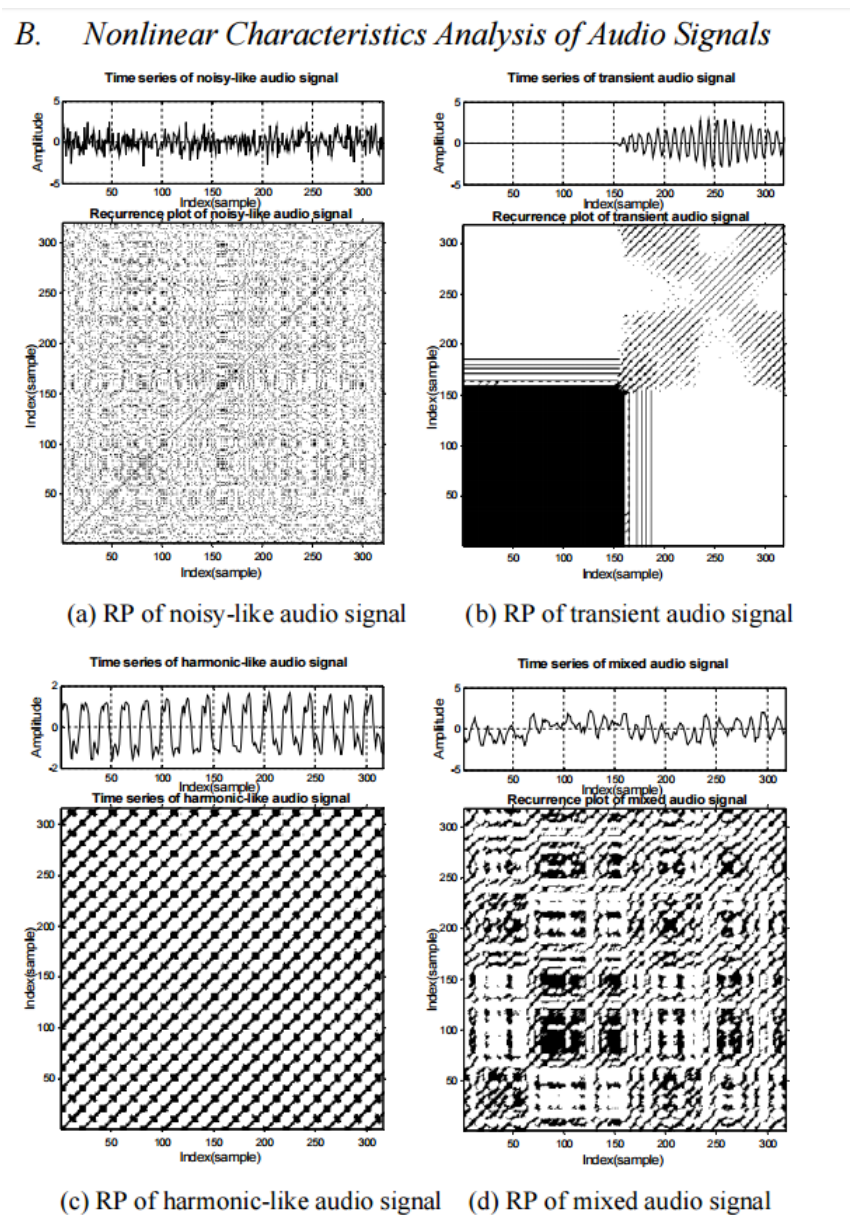

Figure 1: Recurrence plots of different audio signals. Taken from [**?**].

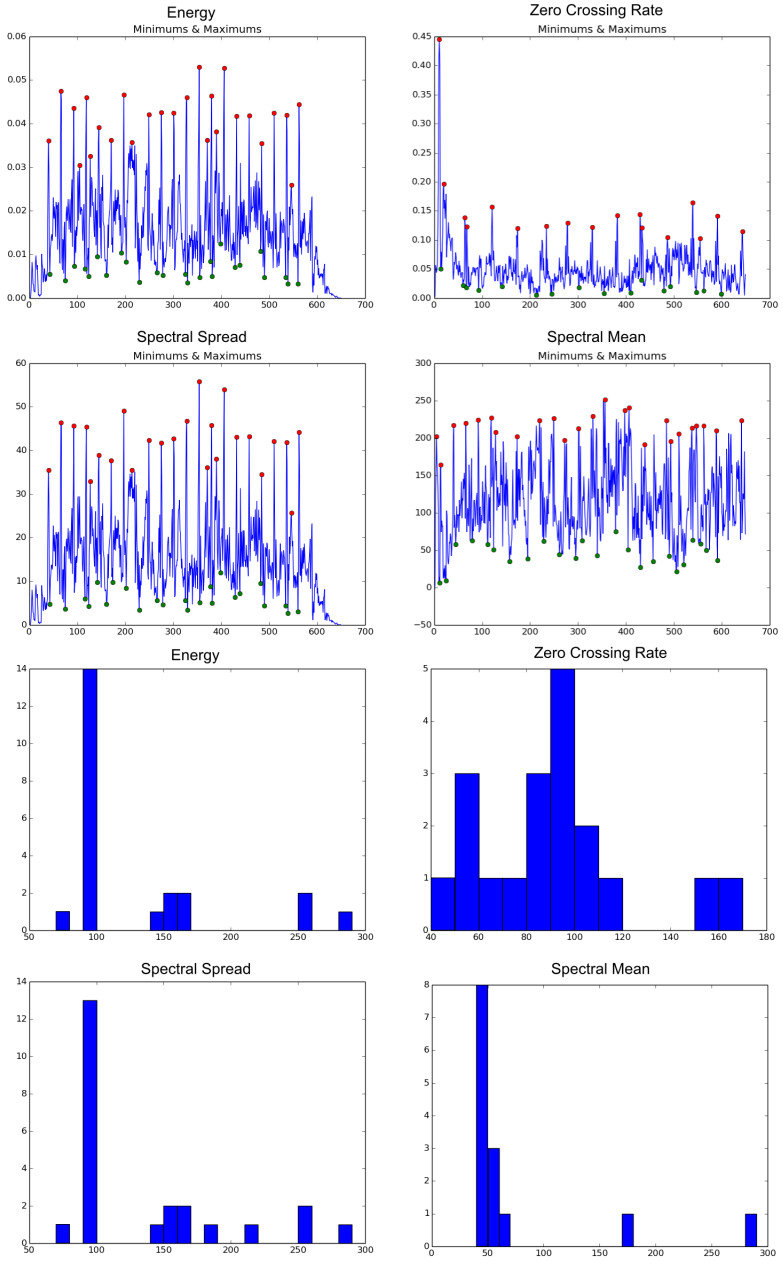

Figure 2: BPM prediction using fast $O(n)$ algorithm.

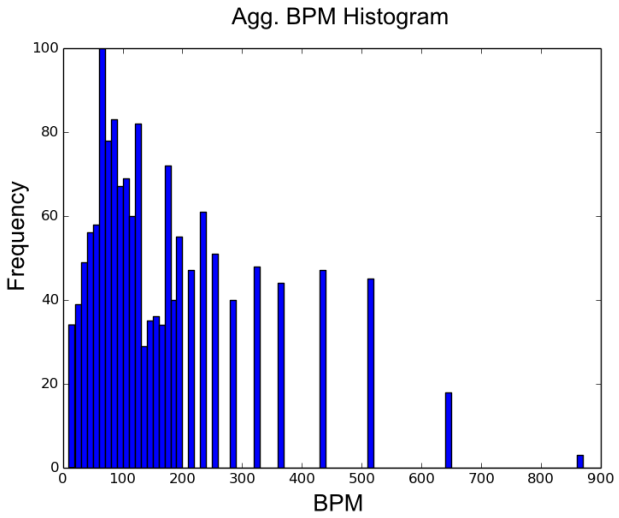

Figure 3: Aggregate BPM histogram.

1.3.1 Issues with BPM Measurements

It is crucial to notice that beat extraction is very sensitive to whole ratios of the true BPM value. Intuitively, if every other peak was missed by the algorithm, the predicted BPM would be $\frac{1}{2}$ of the actual value (i.e. 160BPM and 80BPM should both be considered "correct" because the audio is likely composed of multiple channels of repetition). Furthermore, the sampling rate of the audio signal is required to be much faster than the BPM in order for it to be detected $f >>$ BPM. Moreover, in order to perform a aggregation of each of the histograms accurately, the edges of the bins need to be aligned. For the purpose of this report, the bin width was defaulted to 10BPM and aligned with 0BPM.

1.4 Binary Classification

In order to compare the performance of MFCCs with other features of section **??** a two tier classification task was used. As outlined in section **??**, one of the important aspects of this projects pipeline is a high-level classification of between different audio environments.
The classification problem proposed and used for the analysis of this report is the seperation of audio clips into two categories: music and speech.

1.4.1 Feature Ranking via Single Feature Classification Accuracy

The first tier of the classification task was construction a support vector machine (SVM) classification model using a radial basis function (RBF). This was done for each of the second order features dicussed in section **??** and ranked by their 10-fold cross-vaidation, classification accuracy. This allowed for the forward-selection of best, most correlated features to the two classes: music and speech.

1.4.2 Multi-Feature SVM based on Feature Rankings

After performing these rankings, the top k features were selected and a multi-dimensional SVM model was trained and evaluated using 10-fold cross-vaidation. The values of k were allowed to vary to examine the accuracy convergence and signatures of overfitting. Figure **??** has two examples of a 2d SVM. The x and y axes are two features chosen at random. The figures are intended to illustrate the seperability of the data. Red dots are music clips and blue dots are speech clips. The orange radial line is the RBF decision function generated by training the SVM.

Note: k-nearest neighbors models (KNN) and logistic/linear regression models were also performed with very similar results. As such, they have been omitted from this report.

1.5 Data Sets

The binary classification problem and methodology discussed in section **??** was applied to two independent data sets. One taken from social media, and one taken from the annual Music Information Retrieval Evaluation eXchange (MIREX) contest.

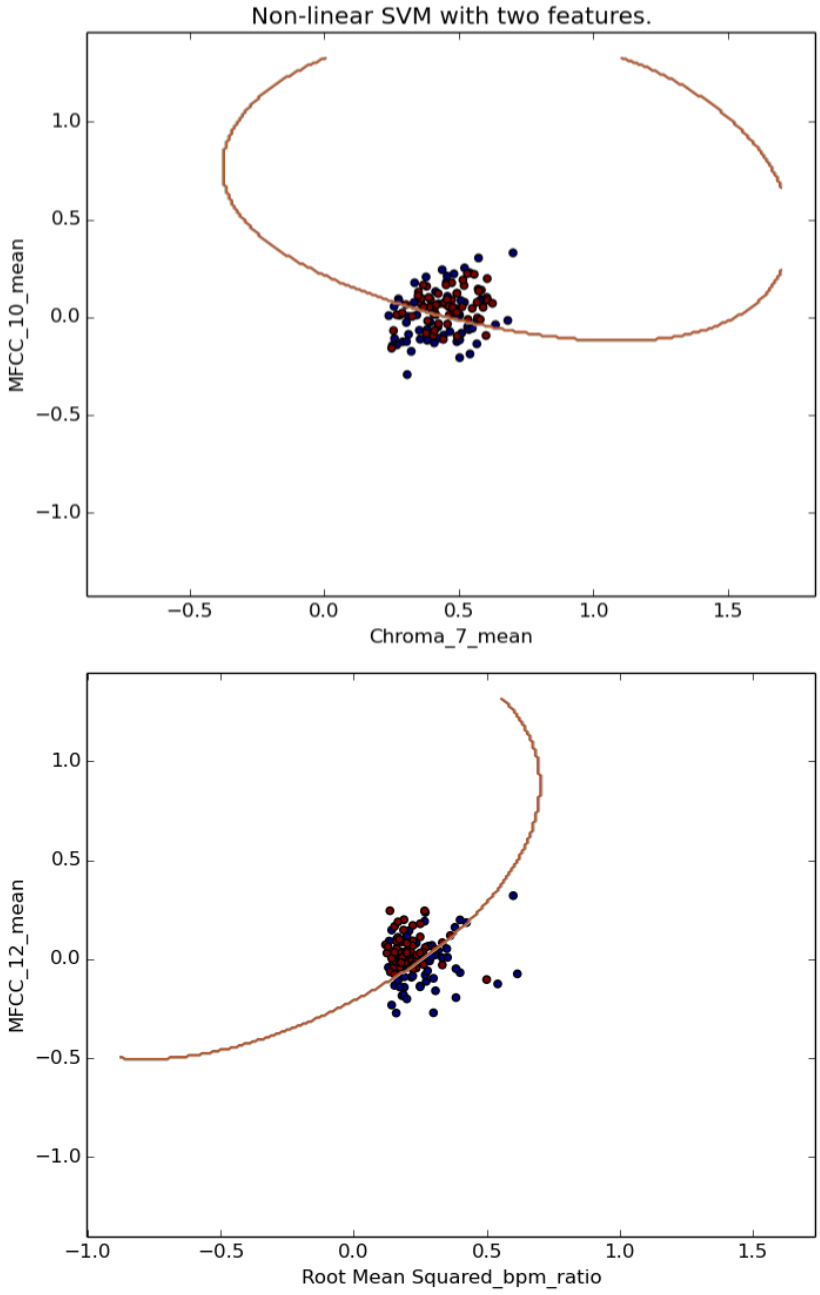

Figure 4: Two Example SVMs with RBF performed on the MIREX data set. Red = Music; Blue = Speech

1.5.1 Social Data

For the social media data set, approximately 2800 of the most popular videos from December 2015 were downloaded from Twitter, Tumblr, Vine and Instagram. These were classified manually by hand into 8 audio classes: cheering, slient, laughter, singing, music, other, talking, and broken_link. Pruning out the broken links and duplicates, 677 unique videos with average length of 5.44 seconds (totaling $\sim$ 5 GB for video and audio) were reclassified into music and speech (approx. half in each). This data set acts as a small sample of the entire population of video this project targets. It contained numerous different spoken languages, genres of music and audio environments. Each audio file was encoded at 44.1 kHz.

1.5.2 MIREX Data

The Music Information Retrieval Evaluation eXchange (MIREX) committee holds competitions each year on a variety of topics including music/speech classification and detection [**?**]. The second data set considered for this report was the dataset used by that competition. It is the "Music Speech" dataset hosted by MARSYAS (Music Analysis,

Retrival and Synthesis For Audio Signals) [**?**]. The MARSYAS data set consists of 120 audio clips each 30 seconds long with 60 belonging to each class. Each audio file was encoded at 22.05 kHz.

1.6 Results

The results of the binary classification problem dictated in section **??** are found in this section. An interpretation of the results and their implications will follow in section **??**.
Note that the feature names were encoded with suffixes to denote the way they were aggregated. For example:

- MFCC_2_var: The variance of the 2nd MFCCs out of 13 across all windows of the clip.
- Mean_mean: The average value of the spectral mean across all windows of the clip.
- Chroma_0_bpm: The beats-per-minute (BPM) predicted using the 0th Chroma Coefficient.
- Zero Crossing Rate_bpm_ratio: The BPM confidence associated with the BPM prediction while using the zero crossing rate.

Table 3: Single Feature SVM with 10-fold Cross-Validation Rankings for Social Data Set

1	MFCC_0_mean	0.752212
2	Root Mean Squared_mean	0.716814
3	Mean_mean	0.713864
4	Chroma_5_mean	0.699115
5	Variance_mean	0.693215
6	Chroma_2_mean	0.663717
7	Energy_bpm	0.651917
8	Chroma_5_bpm	0.648968
9	MFCC_1_var	0.646018
10	Chroma_9_mean	0.643068
	...	
149	MFCC_3_bpm	0.513274
150	Kurtosis_bpm	0.510324
151	MFCC_1_mean	0.507374
152	MFCC_2_bpm	0.507377
153	MFCC_6_bpm	0.498525
154	MFCC_0_var	0.483776

1.6.1 Feature Rankings

Tables **??** and **??** are the single feature SVM classification accuracies after 10-fold cross-validation on each of the 154 features.

1.6.2 Classification Models

Tables **??** and **??** are the single feature SVM classification accuracies after 10-fold cross-validation on each of the 154 features.

Table 4: Single Feature SVM with 10-fold Cross-Validation Rankings for MIREX Data Set

1	MFCC_2_var	0.9375
2	MFCC_0_mean	0.84375
3	MFCC_0_var	0.84375
4	MFCC_3_var	0.828125
5	Skewness_var	0.78125
6	MFCC_3_mean	0.734375
7	Spectral Centroid_bpm_ratio	0.71875
8	MFCC_1_mean	0.703125
9	Mean_mean	0.6875
10	MFCC_1_var	0.6875
...		
148	Chroma_0_bpm	0.6875
149	Zero Crossing Rate_mean	0.40625
150	Zero Crossing Rate_bpm_ratio	0.40625
151	Chroma_8_mean	0.390625
152	Variance_bpm_ratio	0.390625
153	MFCC_0_bpm_ratio	0.359375
154	MFCC_5_bpm	0.328125

Table 5: Precision, Recall and F1-Scores for Social Data Set across 4 SVM - RBF Models

	Precision	Recall	F1-score
$k = 1$ 1. MFCC_0_mean			
music	0.83	0.50	0.62
speech	0.72	0.93	0.81
avg / total	0.78	0.72	0.72
$k = 2$ 1. MFCC_0_mean 2. Root Mean Squared_mean			
music	0.79	0.60	0.68
speech	0.82	0.88	0.85
avg / total	0.81	0.74	0.77
$k = 4$ 1. MFCC_0_mean 2. Root Mean Squared_mean 3. Mean_mean 4. Chroma_5_mean			
music	0.74	0.63	0.68
speech	0.74	0.83	0.78
avg / total	0.74	0.73	0.73
$k =$all			
music	1.00	0.01	0.01
speech	0.54	1.00	0.70
avg / total	0.77	0.51	0.36

Table 6: Precision, Recall and F1-Scores for MIREX Data Set across 4 SVM - RBF Models

	Precision	Recall	F1-score
$k = 1$ 1. MFCC_2_var			
music	0.92	0.97	0.94
speech	0.96	0.90	0.93
avg / total	0.94	0.94	0.94
$k = 2$ 1. MFCC_2_var 2. MFCC_0_var			
music	0.93	0.97	0.95
speech	0.94	0.94	0.94
avg / total	0.94	0.96	0.95
$k = 4$ 1. MFCC_2_var 2. MFCC_0_var 3. MFCC_0_mean 4. MFCC_3_var			
music	0.94	0.85	0.89
speech	0.85	0.93	0.89
avg / total	0.90	0.89	0.89
$k =$all			
music	0.48	1.00	0.65
speech	0.00	0.00	0.00
avg / total	0.24	0.50	0.33

1.7 Interpretation

In order to digest the results of tables **??**, **??**, **??**, and **??**, dicussions will be broken into four parts. First, a comparison between the social and MIREX data sets. Second, the success of MFCCs will be revealed and causes for this justified. Moreover, the results of table **??** will be compared to the winners of the 2015 MIREX competition. Finally, the poor performance of the beat extraction will be rationalized.

1.7.1 Social vs. MIREX Data Sets

At an initial glance at tables **??** and **??** is evident that the features considered perform much differently. For the social data set (see table **??**) the best features were able to perform with classification accuracy of around $\sim 60\% - 75\%$ while for the MIREX data set (see table **??**) the best features achieved classification accuracies of around $\sim 80\% - 93\%$. This large difference was to be expected. The social data set, described in section **??**, was composed of 677 videos from all across the internet. Upon listening to a sample of them by hand,

one will recognize that these audio channels were not *uniform*. The social data set is real-world data; it is composed of all sorts of noisy signals. Furthermore the videos were manually classified into the class (speech or music) that suited it best. A large portion of audio clips had portions of speech (not singing) and portions of music, sometimes overlapping.This analysis outlines a key limitation of performing audio analysis on social media data. Specifically, there is never a clearly defined audio environment and performing high-level audio classification will never be perfect; this classification task is too idealized. Nonetheless, a maximum F1-score of 0.77 (table **??**) is very promising and users of Sysomos products are expected to be understanding of the inherent difficulties with performing this type of classification. The relatively poor classification performance of the social data set is contrasted with tables **??** and **??**. A maximum F1-score of 0.95 reveals how much more seperable the MIREX data set is.

1.7.2 MFCCs Perform Well

Secondly, table **??** illuminates the success of MFCCs. Out of the top 10 features, 7 are derivative of the first order MFCCs. This means MFCCs out-perform most other features at audio classification tasks. By admission however, the computation and motivation for MFCCs is much more complicated as outlined in section **??**. The other features

mentioned in section **??** are statistical properties of the signal and have no basis for audio processing. None of them mimic the way humans perceive and interpret sounds (maybe with exception of Chroma Coefficients). It is important to notice that the top-performing features in table **??** are all mostly variance aggregations on first order features. Intuitively, the variance of the MFCC series should indicate the dynamic range of phonemes, and by extension, words spoken. Unlike the mean of the MFCC series, which does not characterize changes in speech over time. The result, discovered through programmatic analysis validates the reoccuring use of MFCC variances and standard deviation features by top researchers that win the annual MIREX competition [**?**, **?**, **?**].

1.7.3 Comparison to MIREX Winners

The 2015 winners of the music/speech classification task on the MIREX data set was lead by a team at the Institute of Technology Kharaqpur, India [**?**]. They used the standard deviation of MFCCs taken over blocks (bandwidths) of the audio clip and used Gaussian Mixture Models (GMMs) in order to achieve results results outlined in table **??**. In comparison, the Indian team of researchers managed to get classification accuracies around 98.43%, whereas this report's methods managed to achieve accuracies of 95%. Given that the methods

No. Of filter banks	MFCC	MFCC (2 blocks)	MFCC (3 blocks)
20	95.70	94.53	92.18
40	97.65	96.09	**98.43**
60	96.48	96.87	97.65

Table 7: Classification accuracy results of 2015 MIREX winners. Taken from [**?**].

were very similar, how did the winners manage to get such high accuracy? The answer brings to light a very important part of MFCC computation: there are countless *free parameters* associated with the generation of the MFCC values. To list a few: the number of triangular filters (firgure **??**), the shape of the filters, the frequency range of the filter banks, the number of coefficients to retain after the discrete cosine transform (N_{mfcc}), the window size (N), the apodization window function, etc. The MFCC computation used in this report is very uneducated. A quick survey of parameters was taken and used without modification. The current state-of-the-art research of MFCCs involves determining which free parameters to change in order to optimize the classification accuracy [**?**, **?**, **?**]. In doing so, researchers learn a lot about the nature of how humans understand speech. Consequently, the classification accuracy can still be improved past the values found in table **??** with further fine-tuning.

1.7.4 Fast Beat Extraction is Terrible

Careful analysis of tables **??** and **??**, particularily the features ranked at the bottom of the tables demonstrates how poorly the beat extraction aggregation features (see section **??**) performed. Across both data sets, BPM features are consistently failing to achieve higher than 60% classification accuracy. These results come somewhat unsuprisingly, as the beat extraction algorithm outlined in section **??** was designed to be very fast, at the cost of being an approximation of the true BPM. One explanation of this phenomena could be that the data being considered is of a very short time scale (5.5s for social, 30s for mirex). The algorithm is expected to perform better for longer audio signals just by construction; as time progresses, a larger and larger histogram is generated. Alternatively, these results could be indicating that there is no strong correlation between speech, music and extracted beats. However, research suggests this is less likely [**?**]. Regardless, further analysis will have to be conducted in order to fully determine if beat extraction has a place in audio environment classification or not.

Interestingly, in table **??**, the BPM confidence of the spectral centroid achieved classification accuracy of 72%. Research into the automatic excitement detection of baseball game video found that the standard deviation of the spectral centroid can classify audio environments as high as 80.1% [**?**]. Under this study, BPM confidence

out-performed the standard deviation (variance) aggregation. These results can be qualitatively justified for the context of excitement detection; the tempo of audience cheering at a baseball game might be a better measure of excitement than current research suggests [**?**].

UNIVERSITY OF WATERLOO
Faculty of Physics & Astronomy

ACOUSTIC MODELLING USING MEL-FREQUENCY CEPSTRAL COEFFICIENTS

Sysomos
Toronto, Ontario

Prepared by

Thomas C. Fraser
3A Mathematical Physics
ID 20460785
January 15, 2016

154 Quarry Ave.
Renfrew, Ontario
K7V 2W4

January 15, 2016

Mr. Jeff Chen, Department Chair
Department of Physics and Astronomy
University of Waterloo
200 University Avenue West
Waterloo, Ontario
N2L 3G1

Dear Mr. Chen:
I have prepared the enclosed report "Acoustic Modelling Using Mel-Frequency Cepstral Coefficients" as my 3A Work Term Report for my work term spent at Sysomos in Toronto, Ontario. This is my fourth work term report.

The purpose of this report is to summarize the research done by me in order to determine how to be classify audio signals in two categories: music and speech. I aim to convince anyone looking to do human speech-related modelling to consider implementing algorithms to calculate Mel-Frequency Cepstral Coefficients (MFCCs). MFCCs are a form of spectral analysis on spectra themselves and thus the reader should have some experience with mathematical techinques like Fourier Transforms. This report explains the motivations behind MFCCs in detail.

Sysomos is currently a leading provider of social media data analytics for large business clients like Coca-Cola, Adidas, etc.

This report was written entirely by me and has not received any previous academic credit at this or any other institution. I give permission to Sysomos to keep a copy of this report on file and use it as necessary in the future.

Sincerely,

Thomas C. Fraser
ID 20460785

Table of Contents

List of Tables and Figures. iv

Summary vi

1.0 Introduction 1

2.0 Background 3

2.1 Social Data is Not Academic Data 3

2.2 Automatic Speech Recognition 4

2.3 Solution Exploration 5

3.0 Proposed Pipeline 5

3.1 Normalizing Signal 6

3.2 High-Level Classification 7

4.0 Mel-Frequency Cepstral Coefficients Disected 7

4.1 Windowing 7

4.2 Discrete Fourier Transform 11

4.3 Mel-Scale & Triangular Windowing 11

4.4 Discrete Cosine Transform 15

4.5 Deltas & Delta-Deltas 16

5.0 Performance Evaluation 16

5.1 Feature List 17

5.2 Feature Aggregration 17

5.3 Beat Extraction 19

5.3.1 Issues with BPM Measurements 23

5.4 Binary Classification 24

5.4.1 Feature Ranking via Single Feature Classification Accuracy . . . 24
5.4.2 Multi-Feature SVM based on Feature Rankings 24
5.5 Data Sets . . . 25
5.5.1 Social Data . . . 25
5.5.2 MIREX Data . . . 27
5.6 Results . . . 27
5.6.1 Feature Rankings . . . 28
5.6.2 Classification Models . . . 28
5.7 Interpretation . . . 33
5.7.1 Social vs. MIREX Data Sets . . . 33
5.7.2 MFCCs Perform Well . . . 34
5.7.3 Comparison to MIREX Winners . . . 35
5.7.4 Fast Beat Extraction is Terrible . . . 36
6.0 Conclusions . . . 38
7.0 Recommendations . . . 40
References . . . 42
Glossary . . . 49

List of Tables and Figures

Figure 1 Comparison between speech (woman speaking) and music (classical) waveforms and spectrograms. Taken from the MARSYAS "Music Speech" database. [9] . . 8

Figure 2 Zoomed in portion of figure 1. MFCCs characterize repeating red bands in this figure. 9

Figure 3 Mel Scale vs. Hertz Scale 12

Figure 4 Triangular Windowing on Frequency Domain . 13

Table 1 Table of features used in performance evaluation tests. 18

Table 2 Second order features breakdown. 19

Figure 5 Recurrence plots of different audio signals. Taken from [35]. 21

Figure 6 BPM prediction using fast $O(n)$ algorithm. . . . 22

Figure 7 Aggregate BPM histogram. 23

Figure 8 Two Example SVMs with RBF performed on the MIREX data set. Red = Music; Blue = Speech 26

Table 3 Single Feature SVM with 10-fold Cross-Validation Rankings for Social Data Set 29

Table 4 Single Feature SVM with 10-fold Cross-Validation Rankings for MIREX Data Set 30

Table 5 Precision, Recall and F1-Scores for Social Data Set across 4 SVM - RBF Models 31
Table 6 Precision, Recall and F1-Scores for MIREX Data Set across 4 SVM - RBF Models 32
Table 7 Classification accuracy results of 2015 MIREX winners. Taken from [53]. 36

Summary

...

2.0 Introduction

Sysomos is a Toronto based company with secondary offices all across the world. They are a leader in social media management and have over 1000 high-profile clients. Their primary business is gathering and collecting data and actionable insights from social media platforms like Twitter [17], Facebook [18], Tumblr [21], Vine [19], and Instagram [20]. Their computational resources allow for the ingestion and analysis hundreds of billions of data sources in real-time [15]. Sysomos products give clients an ability to understand and visualize their target demographic/audience for various marketing and public-relations projects.

The general ambition of the Research Labs team at Sysomos is to examine the corpus of all social media data and try and discover new ways to uncover information beneficial to the core products of the company. Typical features of social posts used to build a story include tweets, photos, comments, friendships and conversations; anything that can be found online.

One of the projects that I was fortunate enough to lay foundations for was our audio analysis project. Quite ambitiously, the project involved tackling the question: *How can we utilize the audio channel of social media videos to augment or enhance the existing data?* Augmenting data is universally useful for Sysomos products as it allows for better indexing and searchability of content. Additionally, extra data bares revenue streams that Sysomos and similar companies in the industry have yet to tap into.

The long-term goals of the project are as follows:

1. Perform automatic speech recognition (ASR) on the audio in order to extract phrases spoken by individuals.

2. Determine what music genres individuals are interested in for marketing purposes.

3. Predict which video frames are most interesting/characteristic of the entire video so that image analytics can be performed efficiently.

None of these problems are fully solved by the research conducted in this report. However significant strides are made by tackling an easier problem: *How can we predict whether or not a given video contains music, speech, laughing, cheering, silence, etc?* This categorization indirectly contributes to each of the above three goals. This report focuses on acoustic modelling through the use of audio features called Mel-Frequency Cepstral Coefficients (MFCCs). It will assume the reader has a basic knowledge of machine learning concepts such as generic classification and regression models, support vector machines, feature extraction, accuracies, precision, recall, F1-scores. A small amount of musical theory is also assumed.

3.0 Background

Currently, Sysomos is actively analyzing text and relationships between users. Only recently has Sysomos moved into the industry of image processing through the acquisition of Gaze Metrics [40]. Naturely, audio analytics on video is the next step for Sysomos. Sysomos's Audio Analytics project, led by the Research Labs Team and I aims to expose as much information from the audio signal of the video as possible. Consequently, this project is discarding the visual channel of the video entirely. Research conducted at the Standford University shows promise that deep-learning models that combine both audio and visual modalities of a video perform better at classification tasks [11]; however there techniques are currently beyond the scope of the project.

3.1 Social Data is Not Academic Data

Before going further into the analysis, it is important to recognize the context of social media video. Automatic speech recognition tasks are conducted in scenarios where humans are speaking to a machine. For example: when using an iPhone, you speak to Siri [38] or on windows 10 you speak to Cortana [49]. In either case, the end-user understands that the speech recognition algorithms perform better when speaking clearly and with little to no noise. Social media audio is not as nice to work with. Often times, many comlpications arise that diminish the integrity of the audio. For instance there can be loud music playing in the background, screaming,

numerous speakers, different languages, or speech that is inaudible. Speech recognition performs well in an academic setting where speech data sets are normalized and idealized versions of what might be found on social media. As a result, the task of Automatic Speech Recognition is currently also outside the scope of the project, but is intended to be added in the future.

3.2 Automatic Speech Recognition

Although this report will not discuss any results associated with Automatic Speech Recognition (ASR), it is important to understand how acoustic modelling fits into the scope of ASR. Acoustic modelling will prove useful for many other audio classification tasks but is also a core part of and ASR pipeline. There are three main components to a ASR pipeline [46]. Firstly, the acoustic model takes in the raw audio signal and, loosely speaking, performs dimensionality reduction in order to produce a vector of values that characterizes the phonemes (parts of speech) being spoken. Secondly, a pronounciation model determines how likely phonemes are to occur ajacent to one another while someone is speaking. It should then predict what words correspond to a group of phonemes. Pronounciation models are language dependent and are typically Hidden Markov Models (HMMs). Finally, a language model measures how probable two words are to be spoken sequentially. It will then construct phrases using words generated by the pronounciation model.

3.3 Solution Exploration

Speech recognition technologies are plentiful. One economic way of improving Sysomos's audio data stream would be to simply utilize another service that does speech recognition automatically. Early explorations into this idea were not successful. First, publicly available speech apis, namely Google [41], wit.ai [42], IBM Watson [44], and AT&T [43], were too limited for the scope of the social audio project. Also, lincensed software library built for pronounciation and language modelling like Julius [23], CMU Sphinx [24], Kaldi [25] and Nuance Dragon [26] were all found to be incompatible or too expensive for social media audio. Third and final, captioning services like VoiceBase [32], Amara [31], CaptionSync [30] and 3PlayMedia [29] actually provide transcripts of audio recordings by hand; which isn't feasible for millions of videos. As a result of a failed exploration into existing solutions, it was decided that a smaller problem, although equally useful, should be tackled first.

4.0 Proposed Pipeline

Upon consideration of the limitations discovered in section 2.3, a high-level classification on the audio is required. This categorization is sub-divided into 4 stages. First, download the videos from its respective social media website.

Next, extract the audio from the video file using the popular, cross-platform, media manipulation tool called FFmpeg [36]. Afterward, the raw audio signal needs to be normalized. Finally, the normalized audio signal is passed into a high-level classification model.

4.1 Normalizing Signal

In the context of sound as a continous pressure wave [28], the *intensity* of a sound wave is a continous pressure signature $x(t)$ (where t is time and x is the relative measure of the displacement of a speaker or microphone diaphram). In order to digitalize the signal, it is typically sampled around 44.1 kHz. This is the Nyquist frequency [47] corresponding to twice the maximum human hearing frequency of around 20 kHz [45]. Typically, an audio signal is broken into two channels; one intended for the right ear and one for the left $\vec{x}[t] = (x_L[t], x_R[t])$. Furthermore, the values encoded in $x(t)$ are always b-bit integers. In order to normalize for any bitrates, one must take the average of the two channels, and then divide by the maximum intensity. Namely,

$$x[t] \equiv \frac{x_L[t] + x_R[t]}{2 \cdot 2^{b-1}} \tag{2}$$

Henceforth, (1) will be considered the audio signal to be analyzed. The only remaining feature of $x\,[t]$ that isn't normalized at this stage is the sampling frequency f.

4.2 High-Level Classification

After normalizing the audio, it is the goal of this project to classify the clip into one or more categories. As an example, does the clip contain music, speech, laughing, cheering, silence, etc? Answering this question will require acoustic modelling techniques, such as MFCCs.

5.0 Mel-Frequency Cepstral Coefficients Disected

Mel-Frequency Cepstral Coefficients (MFCCs), developed in 1974 by Bridle, Brown and Mermelstein [1, 3, 4], are a vector of real-valued features that correspond to a short window of time within an audio signal. They are a re-spresentation of the unique *phoneme* being spoken during that short window. Phonemes, a term used in the study of linuistics, are the irreducible sound elements made when speaking. English has 44 phonemes such as /m/ as in *man, summer, palm* or /ow/ in *now, shout, bough*. What follows is a detailed exposition on how MFCCs are calculated. This is done to accomplish two things. Firstly, to outline how to implement MFCCs and identify areas that can be explored for further improvement. Secondly, this section aims to convince the reader that MFCCs are well-motivated in their construction for use in acoustic modelling. A quantitative analysis of their importance will follow in section 5.0 .

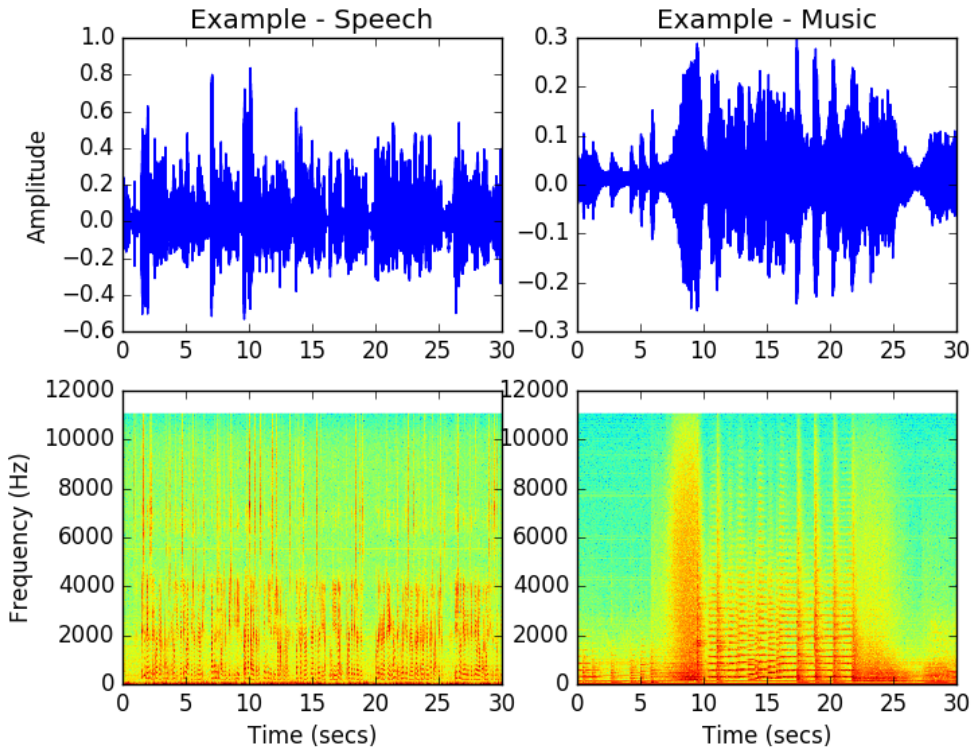

Figure 5: Comparison between speech (woman speaking) and music (classical) waveforms and spectrograms. Taken from the MARSYAS "Music Speech" database. [9]

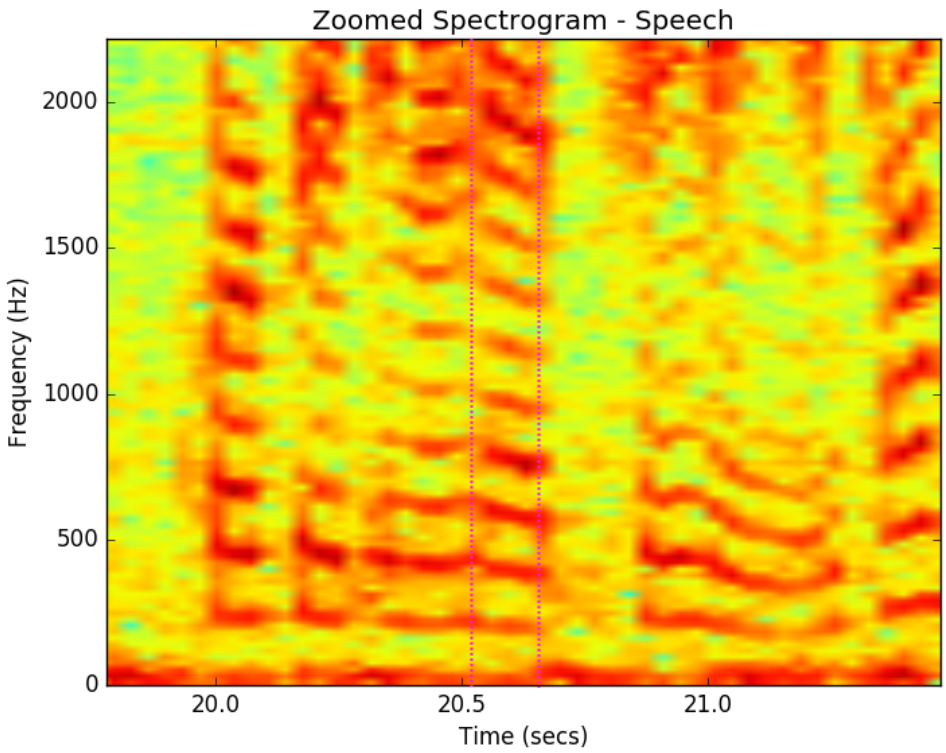

Figure 6: Zoomed in portion of figure 1. MFCCs characterize repeating red bands in this figure.

5.1 Windowing

Since phonemes spoken are considered *stationary waves* on the time length less than around $\sim 30\,\text{ms}$ [7], the total audio signal $x[t]$ given by (1) needs to be divided up into shorter intervals of time. Notice in figure 2, the spectrogram is approximately constant within the outlined interval. MFCCs characterize the structure of repeating high-intensity (red) bands. This structure is unique for each phoneme. An appropriate window size (N) should be chosen based off the sampling frequency of the raw signal. Choosing a window length of around 1024 samples is useful for two reasons:

1. Powers of 2 require no padding when taking a Discrete Fourier Transform [8].

2. At a sampling rate of $44.1\,\text{kHz}$, 1024 samples corresponds to $\sim 20\,\text{ms}$ which is completely sufficient the time scale considered.

Before performing a fast fourier transform on a finite interval of time, an *apodization window function* needs to be applied to minimize leakage artifacts induced by the periodic extension of the signal [16]. For the purposes of MFCCs, the popular Hamming Window [5] works just fine.

MFCCs are computed for each window of samples. Let $m_w^{(0)}$ represent the vector of values of contained within the window w. The '(0)' indicates the zeroth stage of the MFCC computation. Thus $m_{w,j}^{(0)}$ are the individual values of intensity $0 \leq j \leq N-1$

5.2 Discrete Fourier Transform

Next, in order to expose the frequency domain of the short window w, perform a real-valued Discrete Fourier Transform [13].

$$m_{w,k}^{(1)} = \frac{1}{N} \left| \sum_{j=0}^{N-1} m_{w,j}^{(0)} e^{-2\pi i k j / N} \right| \quad 0 \leq k \leq N-1$$

Now $m_{w,k}^{(1)}$ measures the contribution made to the signal $m_w^{(0)}$ at frequency $\frac{k}{N} \cdot f$ (where f is the sampling frequency of the original signal). Note $m^{(0)}$ is in the time-domain while $m^{(1)}$ is in the frequency domain. Note however that in principle, a Fast Fourier Transform is taken to reduce the time complexity of the computation from $O(N^2)$ to $O(N \log N)$ [8].

5.3 Mel-Scale & Triangular Windowing

Now that the frequency domain is revealed via $m_w^{(1)}$, MFCCs shift the standard frequency scale from Hertz to Mels. The Mel frequency scale, introduced in 1937 [2], is a logarithmic scale associated with the way humans perceive pitch. At larger frequencies, increasingly large frequency intervals are perceived by humans to be equal pitch increments [2]. The scale is defined with respect to 1000mels = 1000Hz. Figure 3 shows this relationship.

$$m = \frac{1000}{\log 2} \log \left(1 + \frac{f}{1000} \right)$$

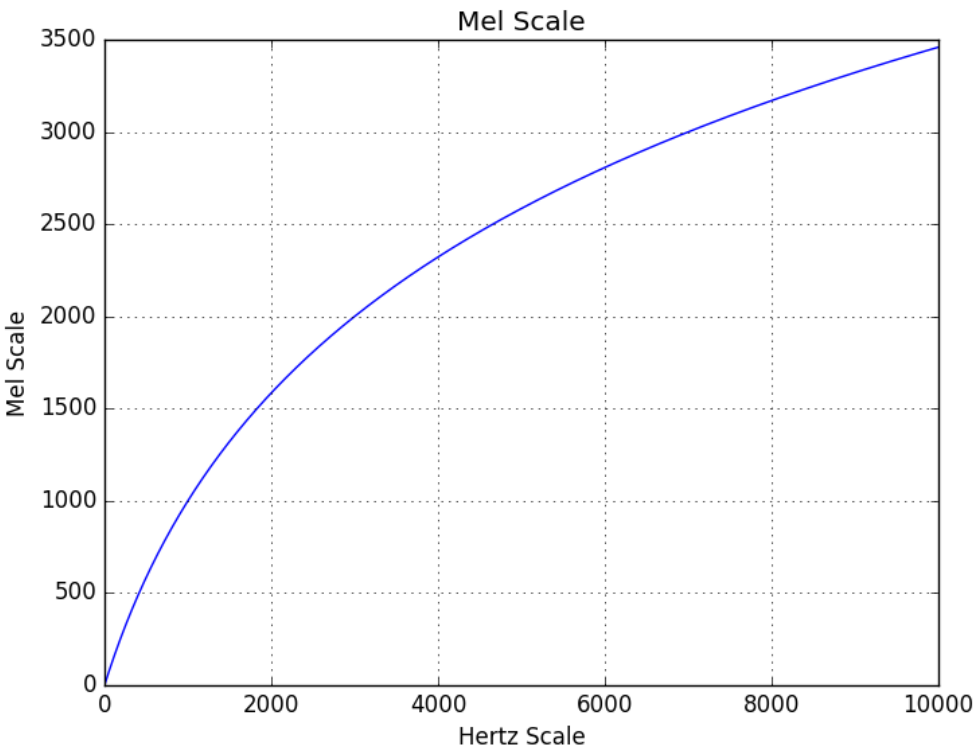

Figure 7: Mel Scale vs. Hertz Scale

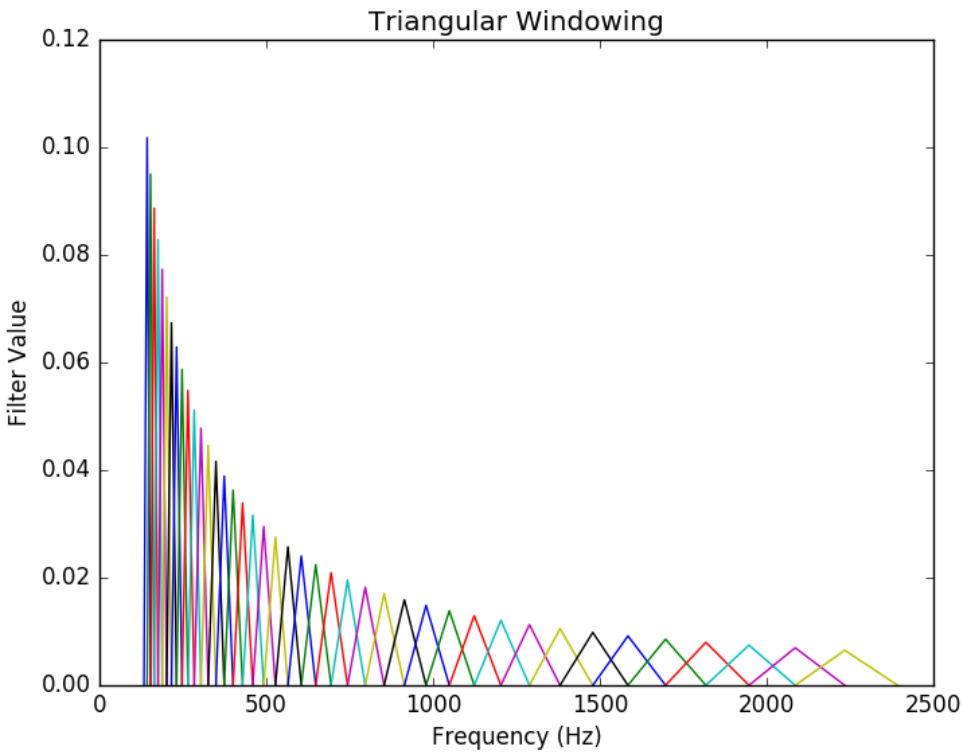

Figure 8: Triangular Windowing on Frequency Domain

In order to normalize the signal $m_w^{(1)}$ further, it is passed through a set of triangular windows. This is illustrated in figure 4. Letting each triangular window be denoted T_j, there are typically a few dozen windows (N_T) (leading research indicates 40 is optimal [53]). Thus j is an integer with $0 \leq j \leq N_T - 1 \in Z$. This step in the computation of MFCCs can be expressed in equation (2).

$$m_{w,j}^{(2)} = \sum_{k=0}^{N-1} T_{j,k} m_{w,k}^{(1)} \tag{3}$$

In equation (2), $T_{j,k}$ represents the value of triangular window j at the frequency $\frac{k}{N} \cdot f$. Thus $m_{w,j}^{(2)}$ is the response from window T_j and is still in the frequency domain. Notice in figure 4 the centers of the triangles are spaced in equally on the mel-scale, but logarithmically spaced on the hertz-scale. Furthermore, to normalize the response from each window T_j, the height of the triangular is chosen such that all windows have equal area. The width of the window is determined by the neighboring centers. Also note that the windows have freqeuncy range $\sim 100 - 2500$Hz. This range has a lot of freedom but is approximately the range of human voice production [33]. Finally, the logarithm of each $m_w^{(2)}$ is taken to normalize the difference between the results from each window [39].

$$m_w^{(3)} = \log\left(m_w^{(2)}\right)$$

5.4 Discrete Cosine Transform

Now that we have a vector ($m_w^{(3)}$) that characterizes the periodic behaviour of the signal in *time* we can explore the periodic nature of the signal in the *frequency* domain. This is the key component that differentiates MFCCs from typical signal analysis features. In order to accomplish this, we can perform yet another discrete fourier transform. However, in principle, only a discrete cosine transform is sufficient because the current signal is real-valued $m_w^{(3)} \in \mathbb{R}$ and the output is required to be real.

$$m_{w,j}^{(4)} = \sum_{j=0}^{N_T-1} m_{w,j}^{(3)} \cos\left[\frac{k\,(2j+1)\,\pi}{2N_T}\right] \quad 0 \leq k \leq N_{\text{mfcc}} - 1 < N_T \qquad (4)$$

Performing a discrete cosine transform in equation (3) moves $m_w^{(3)}$ from the frequency domain to $m_w^{(4)}$ in the *quefrency* domain, which has units of time but is not correlated with the initial time domain. Just as the discrete fourier transform exposed the *spectral* domain of the signal, (3) exposes the *cepstral* domain of the signal. It is very important to note that k takes on only N_{mfcc} values. Thus $m_w^{(4)}$ is a vector of length N_{mfcc}. MFCCs act as a low-pass filter on the quefrency domain as only the smallest N_{mfcc} quefrency values are kept. This smooths out the representation of the vector $m_w^{(3)}$ because it removes high-quefrency noise artifacts. Typically, it is customary to select the first $N_{\text{mfcc}} = 13$ cofficients [50, 53].

5.5 Deltas & Delta-Deltas

The values obtained in 4.4, namely $m_w^{(4)}$, are called the *Mel-Frequency Cepstral Coefficients*. They are a vector of N_{mfcc} real values for each window w. For the purposes of the analysis in section 5.0 , these are considered as the final MFCCs.

$$\text{MFCC}_w = m_w^{(4)}$$

Nevertheless, research suggests that the human brain determines what phonemes are spoken by context of the sounds produced nearby in time [51, 52]. Effectively, the trajectory of the MFCC vector contributes to the cognitive understanding of the spoken sounds. Often it is common to introduce the notion of *deltas* and *delta-deltas*; the discrete velocity and acceleration of the MFCCs respectively.

$$\text{MFCC}_w = \left[m_w^{(4)}, \Delta m_w^{(4)}, \Delta^2 m_w^{(4)}\right]$$

Where $\Delta m_w = m_{w+1} - m_{w-1}$ and $\Delta^2 m_w = \Delta m_{w+1} - \Delta m_{w-1}$ while appropriately handling boundary cases.

6.0 Conclusions

MFCCs model human interaction with audio.

MFCCs effectively mimic the human behaviour of listening to speech. They utilize the logarithmic perception of both pitch and loudness, the typical frequency ranges of human speech, and the harmonics generated when humans speak phonemes. This is quantitatively illustrated where variances in MFCC values are used to classify MIREX music/speech data sets to an accuracy of 95%.

MFCCs have the ability to perform well in non-speech modelling.

Typically, MFCCs are used for acoustic modelling of automatic speech recognition tasks. In this study, MFCCs are used to build SVM models of 2 features to achieve near state-of-the-art music/speech classification accuracies, thus justifying their use in other acoustic modelling tasks.

MFCCs are the best features considered for speech modelling.

When compared against 25 other statistical and musical features of an audio signal in both the time and frequency domains, MFCCs outperform all others with few exceptions. This indicates the MFCCs are the best features to use for acoustic modelling tasks.

Beat extraction is not an effective tool for audio environment classification.

For both data sets considered, using beat extraction on time-based feature

series to generate BPM and BPM confidence features will produce less than ideal results. Few beat extraction features managed to achieve a classification accuracy of more than 60% on music/speech data sets, while some managed to perform worst than random guessing (50%).

7.0 Recommendations

Fine-tune free parameters of MFCC computation.

As outlined in 5.7.3, MFCCs have a lot of free parameters to be chosen at implementation time. Fine-tuning and adjusting these parameters to maximize the performance of MFCCs can produce higher than demonstrated classification accuracies.

Build full pipline using complete model.

This project demonstrated why MFCCs are both theoretically and experimentally excellent models for speech production and understanding. By using MFCCs as an acoustic model, a full pipeline should be designed to integrate with a language model and pronounciation model in order to perform end-to-end automatic speech recognition and other speech modelling tasks.

Perform scalability analysis on pipeline.

This report did no analysis or tests of computational scalability. It is possible that although MFCCs are best in an academic setting, their computation is too expensive to be used on a large subset of all social video. Scalability tests should be performed on the proposed pipeline.

Explore deep learning techniques to improve feature optimizations.

Deep learning techniques, specifically Convolution Neural Networks (CNNs) could potentially expose audio features that could out-perform MFCCs. Ex-

ploring existing research into feature learning on audio signals is necessary if greater than 75% accuracy is desired for social audio.

References

1 P. Mermelstein (1976), Distance measures for speech recognition, psychological and instrumental, in Pattern Recognition and Artificial Intelligence, C. H. Chen, Ed., pp. 374388. Academic, New York.

2 Stevens, Stanley Smith; Volkmann; John; & Newman, Edwin B. (1937). A scale for the measurement of the psychological magnitude pitch. Journal of the Acoustical Society of America 8 (3): 185190.

3 S.B. Davis, and P. Mermelstein (1980), Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences, in IEEE Transactions on Acoustics, Speech, and Signal Processing, 28(4), pp. 357366.

4 J. S. Bridle and M. D. Brown (1974), An Experimental Automatic Word-Recognition System, JSRU Report No. 1003, Joint Speech Research Unit, Ruislip, England.

5 Weisstein, Eric W. Hamming Function. From MathWorld-A Wolfram Web Resource. http://mathworld.wolfram.com/HammingFunction.html

6 H. Boril, A. Sangwan, T. Hasan, J. H. L. Hansen. Automatic Excitement-Level Detection for Sports Highlights Generation. (2010) Center for Robust Speech Systems (CRSS), University of Texas.

7 W. Labov and M. Baranowski (8 Nov., 2004) 50 msec, submitted to Language Variation and Change. University of Pennylvannia.

8 Weisstein, Eric W. Fast Fourier Transform. From MathWorld-A Wolfram Web Resource. http://mathworld.wolfram.com/FastFourierTransform.html

9 Data Sets Music Speech. (n.d.). Marsyas Music Analysis, Retrieval and Synthesis For Audio Signals. Retrieved Jan. 12, 2016, from http://marsyasweb.appspot.com/download/data_sets/

10 2015:Music/Speech Classification and Detection Results. Retrieved Jan. 14, 2016, from http://www.music-ir.org/mirex/wiki/2015:Music/Speech_Classification_and_Detection_Results

11 J. Ngiam, A. Khosla, M. Kim. Multimodal Deep Learning. (n.d.). Department of Music, Standford University. Retrieved Dec., 11, 2015 from http://ai.stanford.edu/ ang/papers/nipsdlufl10-MultimodalDeepLearning.pdf

12 J. Schuluter. Music/Speech Classification and Detection Mirex Submission. Austrian Research Institue for Artificial Intelligence, Vienna. Retrieved Jan. 14, 2016, from http://www.music-ir.org/mirex/abstracts/2015/JS2.pdf

13 Weisstein, Eric W. Discrete Fourier Transform. From MathWorld-A Wolfram Web Resource. http://mathworld.wolfram.com/DiscreteFourierTransform.html

14 J. Trouvain. Tempo Variation in Speech Production: Implication for Speech Synthesis. (April 2003).

15 Sysomos: Social Media Monitoring Tools (10 Jan. 2016) Retrieved 10 Jan. 2016 from https://sysomos.com/

16 The Discrete Fourier Transform. (n.d.). Retrieved Jan. 13, 2016, from http://www.robots.ox.ac.uk/ sjrob/Teaching/SP/l7.pdf

17 Twitter (10 Jan. 2016) Retrieved 10 Jan. 2016 from https://twitter.com/

18 Facebook (10 Jan. 2016) Retrieved 10 Jan. 2016 from https://www.facebook.com/

19 Vine (10 Jan. 2016) Retrieved 10 Jan. 2016 from https://vine.co/

20 Instagram (10 Jan. 2016) Retrieved 10 Jan. 2016 from https://www.instagram.com/?hl=en

21 Tumblr (10 Jan. 2016) Retrieved 10 Jan. 2016 from https://www.tumblr.com/

22 E. Billauer. peakdet: Peak detection using MATLAB (2012) Retrived from http://billauer.co.il/peakdet.html

23 Open-Source Large Vocabulary CSR Engine Julius. Julius. (2014) Retrieved Dec. 13, 2015 from http://julius.osdn.jp/en_index.php?q=index-en.html#documentation

24 CMUSphinx Wiki (2015) Retrived Dec. 13, 2015 from http://cmusphinx.sourceforge.net/wiki/

25 Kaldi Documentation Retrived Dec. 13, 2015 from http://kaldi.sourceforge.net/

26 Dragon Speech Recognition Software. NUANCE. Retrived Dec. 13, 2015 from http://www.nuance.com/dragon/index.htm

27 F Statistic: Definition and How to find it. Statistics How To (n.d.). Retrieved Jan. 13, 2016 from http://www.statisticshowto.com/f-statistic/

28 Feynman, R., & Leighton, R. (1963). Sound. The wave equation. In The Feynman lectures on physics (New Millennium ed., Vol. 3). Reading, Mass.: Addison-Wesley Pub.

29 Video Captioning + Transcription + Subtitling. 3PlayMedia. Retrieved Dec. 13, 2015 from http://www.3playmedia.com/

30 About Automatic Sync Technologies. CaptionSync. Retrived Dec. 13, 2015 from http://www.automaticsync.com/captionsync/

31 Amara - Caption, translate, subtitle and transcribe video. Retrieved Dec. 13, 2015 from https://www.amara.org/en/

32 APIs for speech recognition and speech analytics. VoiceBase. Retrieved from Dec. 13, 2015 from https://www.voicebase.com/

33 Baken, R. J. (1987). Clinical Measurement of Speech and Voice. London: Taylor and Francis Ltd. (pp. 177)

34 B. A. Hutchins, Jr. and W. H. Ku. An Adapting Delay Comb Filter for the Resotration of Audio Signals Badly Corrupted with a Periodic Signal of Slowing Changing Frequency. Cornell University, School of Electrical Engineering.

35 L. Zhang, C. Bao, X. Liu. Audio Classification Algorithm Based on Nonlinear Chracteristics Analysis. Speech and Audio Signal Processing Laboratory, Beijing University of Technology, Beijing.

36 A complete, cross-platform solution to record, convert and stream audio and video. Retrieved Dec. 17, 2016 from https://www.ffmpeg.org/

37 E. D. Scheirer. Tempo and Beat Analysis of Musical Signals. (n.d.). Machine Listening Group, MIT Media Laboratory.

38 Siri. Yout wish is its command. Apple Inc. (2016) Retrieved from http://www.apple.com/ca/ios/siri/

39 S. Molau, M. Pitz, R. Schluter, and H. Ney. Computing Mel-Frequency Cepstral Coefficients On The Power Spectrum. (n.d.). Computer Science Department, University of Technology, Germany

40 Sysomos: Gaze, See Your Brand in a Whole New Way (2015). Retrieved Jan. 14, 2016 from https://sysomos.com/products/sysomos-gaze

41 Web Speech API Demonstration (n.d.). Retrieved Dec. 13, 2015 from https://www.google.com/intl/en/chrome/demos/speech.html

42 wit.ai Natural Language for Developers (2015). Retrieved Dec. 13, 2015 from https://wit.ai/

43 AT&T Speech to Text API Documentation (2015). Retrieved Dec. 13, 2015 from http://developer.att.com/apis/speech/docs

44 IBM Watson Developer Cloud Speech to Text (2015). Retrieved Dec. 13, 2015 from http://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/speech-to-text.html

45 High Frequency Range Test (8-22kHz). (n.d.). Retrieved Jan. 12, 2016 from http://www.audiocheck.net/audiotests_frequencycheckhigh.php

46 S. Furui. Automatic Speech Recognition and It's Application To Information Extraction (n.d.). Tokyo Institute of Technology

47 Weisstein, Eric W. Nyquist Frequency. From MathWorld-A Wolfram Web Resource. http://mathworld.wolfram.com/NyquistFrequency.html Retrieved 10 Jan. 2016

48 Music Information Retrieval Evaluation eXchange (MIREX) Home. Retrived Jan. 12, 2016 from http://www.music-ir.org/mirex/wiki/MIREX_HOME

49 What is Cortana? Microsoft (2016) Retrieved from http://windows.microsoft.com/en-ca/windows-10/getstarted-what-is-cortana.

50 Z. Ma, E. Fokoue. Speaker Gender Recognition via MFCCs and SVMs. (2013) Center for Quality and Applied Statistics.

51 S. Renals, M. Hocbberg, and T. Robinson. Learning Temporal Dependencies in Connectionist Speech Recognition. Cambridge University Engineering Department

52 R. S. Sutton, A. G. Barto: Reinforcement Learning: An Introduction. MIT Press, 1998.

53 V. Ghodasara, D. S. Naser, S. Waldekar, G. Saha. Speech/Music Classification Using Block Based MFCC Features. (2015) Electronics & Electrical Communication Engineering Department, Indian Institute of Technology Kharaqpur, India.

54 Champion, R., Paci, T. & Vardon, J. (2012). PD 2: Critical Reflection and Report Writing. Retrieved 1 March, 2012 from https://learn.uwaterloo.ca/d2l/le/content/80224/viewContent/605550/View **Note:** [54] was referenced to format this report.