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 14, 2016

154 Quarry Ave.
Renfrew, Ontario
K7V 2W4

January 14, 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 Report for my work term spent at Sysomos in Toronto, Ontario. This is my second work term report. The purpose of this report is to examine the problems associated with a popular technique used in application design known as a spriteSheet. I aim to convince anyone familiar with the ideas discussed to consider the solution I propose to solve these problems. This report uses the techniques I developed while tactling these problems but is targeted at a audience with a wider range of applications.

Sysomos is currently working on a multi-platform video game. My supervisor, Elyot Grant, assigned me with overcoming some limitations of graphics API used.

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

Summary

1.0 Introduction

Sysomos is a Toronto, Ontario 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 insights from social media platforms like Twitter [12], Facebook [13], Tumblr [16], Vine [14], and Instagram [15]. Their computational resources allow for the ingestion and analysis hundreds of billions of data sources in real-time [10]. 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 news ways to learn actionable insights that can be beneficial to the core products of the company. Typical features of social posts that are 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. Essentially, the project involved tackling the question: *How can we utilize the audio channel of social media videos to augment the existing data?* Augmenting data it universally useful for Sysomos products as it allows for better indexing for searchability as well as gives Sysomos an extra data stream they and similar companies in the industry have yet to tap into.

Early on, the composite problems were identified as follows:

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

2. Determine what music/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.

This report outlines and analyzes some of the work done to tackle these problems. It focuses on acoustic modelling through the use of audio features called Mel-Frequency Cepstral Coefficients (MFCCs).

2.0 Audio Survey

...

2.1 Social is not Clean

...

2.2 Audio Segmentation & Environment Detection

...

2.3 Solution Exploration

...

2.4 Use Acoustic Modelling

...

3.0 Proposed Pipeline

...

3.1 Video Downloading

...

3.2 Audio Extraction

...

3.3 Normalizing Signal

In the context of sound as a continous pressure wave [19], 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 [26] corresponding to twice the maximum human hearing frequency of around 20 kHz [25]. Also 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 value encoded for the intensity is always a b-bit integer. In order to normalize for all different bitrates, one must take the average of the two channels, and divide by the maximum intensity. Namely,

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

Henceforth, (1) will be considered the audio signal to be analyzed. All that remains to be normalized for is the sampling rate f of the signal.

3.4 High Level Classification

...

3.5 Augment Data & Frame Recommendations

...

4.0 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-values features that correspond to a short window of time within an audio signal. They are a respresentation of the components of the audio signal that correspond to the unique *phoneme* being spoken by the speaker. 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 acoustic-speech modelling. A quantitative analysis of their importance will follow in section 5.0

.

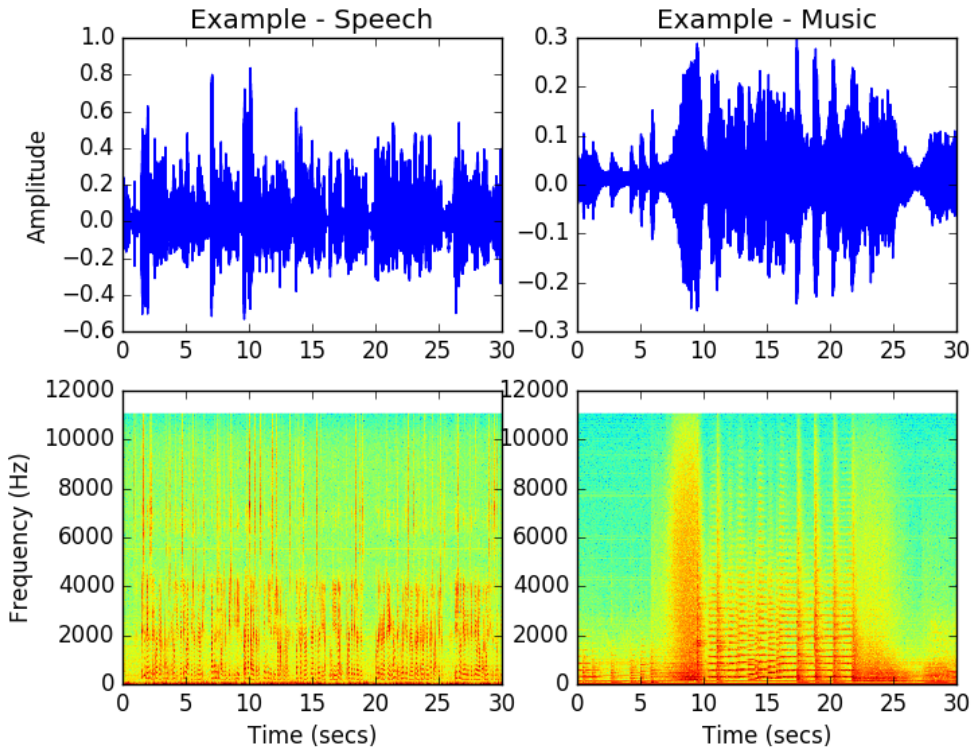

Figure 1: Comparison between speech (woman speaking) and music (classical) waveforms and spectrograms. Taken from the MARSYAS "Music Speech" database. [8]

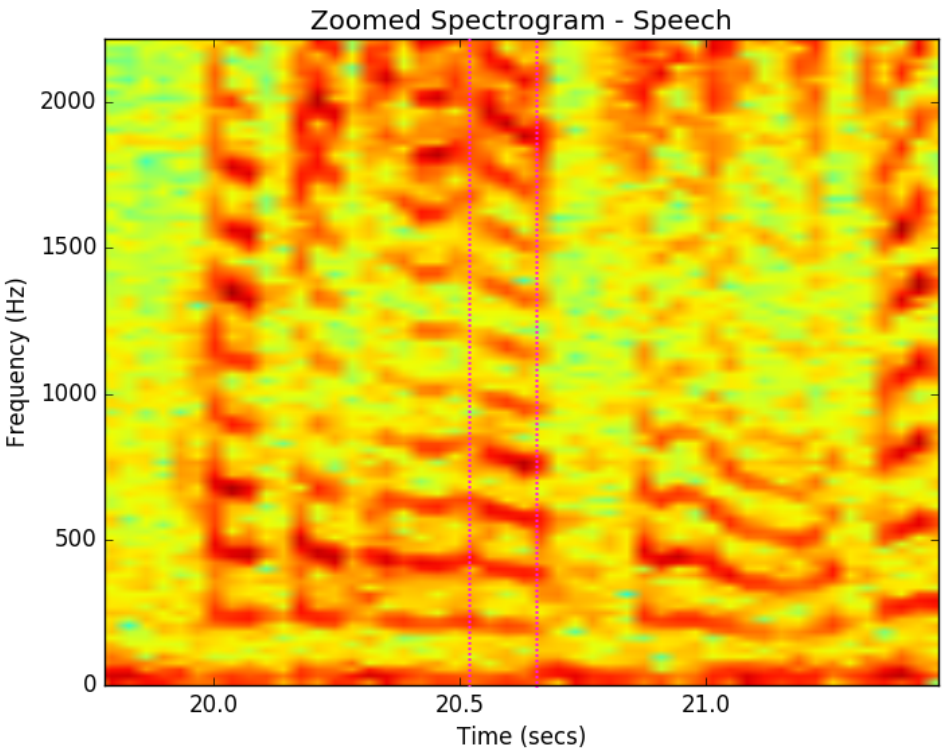

Figure 2: A Zoomed in portion of figure 1. MFCCs characterize the periodicity of the spectrum across short durations of time. The outlined interval represents a window w that is being considered.

4.1 Windowing

Since MFCCs model the phonemes spoken by individuals, the total audio signal $x[t]$ given by (1) needs to be divided up into shorter intervals of time. Typically, phonemes sounds are considered *stationary waves* on the time length less than around $\sim 30\,\text{ms}$ [6]. This is illustrated in figure 2. The spectrogram is approximately constant across this time scale (outlined interval). MFCCs measure the feature of the repeating high-intensity (red) bands; a unique signature exists for each phoneme. 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 [7].

2. At a sampling rate of 44.1 kHz, 1024 samples corresponds to $\sim 20\,\text{ms}$ which is completely sufficient for stationary waves.

An appropriate sample rate should be chosen based off the sampling frequency of the raw signal. 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 [11]. For the purposes of MFCCs, the popular Hamming Window [5] works just fine.

MFCCs are computed for each window of 1024 samples. Let $m_w^{(0)}$ represent the vector of values of sound intensity in window w. Also, (0) indicates

the zeroth stage of the MFCC computation. Thus $m^{(0)}_{w,j}$ are the individual values of intensity $0 \leq j \leq N - 1 = 1023$

4.2 Discrete Fourier Transform

Next, in order to expose the frequency domain of the short window w, perform a real-valued Discrete Fourier Transform [9].

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

Now $m^{(1)}_{w,k}$ represents the magnitude of the signal $m^{(0)}_{w}$ composed of 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 principle, a Fast Fourier Transform is taken in to reduce the time complexity of the computation from $O(N^2)$ to $O(N \log N)$ [7].

4.3 Mel-Scale & Triangular Windowing

Now that the frequency domain is revealed by the series, MFCCs shift the standard frequency scale of Hertz to Mels. The Mel frequency scale, introduced in 1937 [2], is a logirthmic scale associated with the way humans preceive pitch. At larger frequencies, increasingly large frequency intervals are preceived by humans to be equal pitch increments [2]. The

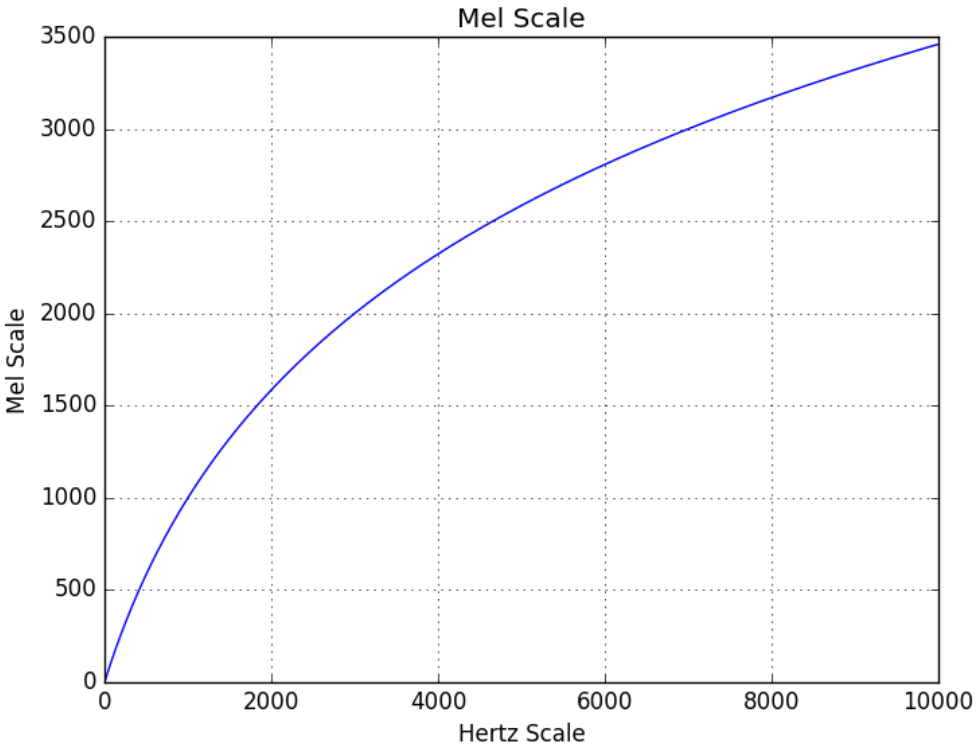

Figure 3: Mel Scale vs. Hertz Scale

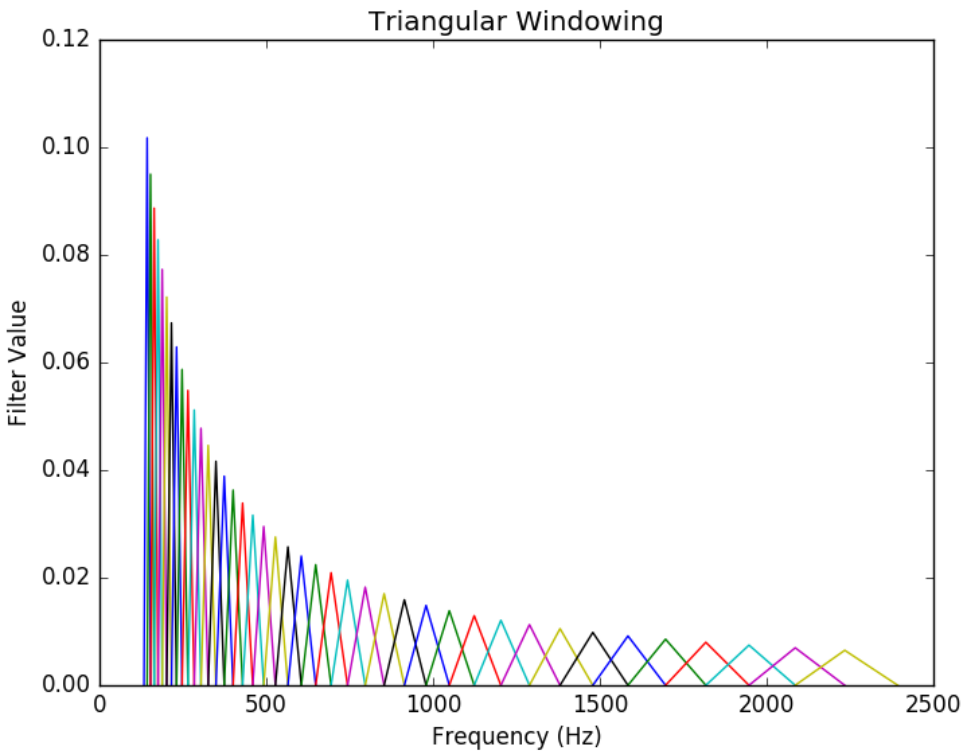

Figure 4: Triangular Windowing on Frequency Domain

scale is defined with respect to 1000mels = 1000Hz. Figure 3 shows this relationship.

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

In order to normalize the signal $m_w^{(1)}$ further, they are 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 [31]). 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).

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

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 that the windows are spaced with their centers spaced in equal intervals on the mel-scale. Thus on the hertz-scale, they appear logarithmically spaced. 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 range from around $\sim 100 - 2500$Hz. This range has a lot of freedom but is approximately the range of human voice production [20]. Finally, the logarithm of each $m_w^{(2)}$ is taken to normalize the difference between the results from each window

[24].

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

4.4 Discrete Cosine Transform

Now that we have a vector, namely $m_w^{(3)}$, 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.

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

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 [28, 31].

4.5 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_T real values for a given window w from the original signal $m^{(0)}$. For the purposes of the analysis in section 5.0 , these are considered as the final MFCCs.

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

Nonetheless research suggests that the human brain determines what phonemes are spoken by context of the sounds produced nearby in time. [29, 30]. 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.

4.6 Information Compression

...

5.0 Performance Evaluation

Section 4.0 outlined the motivation for MFCCs from the audial cognitive-psychological perspective as well as how to expose 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 5.1). It will describe the methods used to compare these features and interpret the results of those tests.

5.1 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 5.1 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.

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

Feature Name	**Desciption**
Time-Based Features	
Zero-Crossing Rate	Number of times signal crosses zero $\text{zcr}\left(x\left[t\right]\right) = \frac{1}{N-1}\sum_{t=1}^{N-1}\mathbb{I}\left\{x\left[t\right]x\left[t-1\right]<0\right\}$
Energy	Energy of discrete time signal $\text{energy}\left(x\left[t\right]\right) = \frac{1}{N}\sum_{t=1}^{N-1}x\left[t\right]x^{*}\left[t\right]$
Root-Mean Squared	Quadratic mean of signal $\text{rms}\left(x\left[t\right]\right) = \sqrt{\frac{1}{N}\sum_{t=1}^{N-1}x^{2}\left[t\right]}$
Energy-Entropy	Shannon Entropy of sub-divided windows ($n = 10$) $\text{H}\left(x\left[t\right], n\right) = -\sum_{i}^{n}\left\{e_i, \ln e_i\right\}$ $e_i = x\left[t\right]/\text{energy}\left(x_i\left[t\right]\right)$
Frequency-Based Features	
Spectral Centroid	Center of mass of spectrum (Hz) $\text{centroid}\left(X[n]\right) = \sum_{n=0}^{N-1}X[n]f(n)/\sum_{n=0}^{N-1}f(n)$
Flatness or Wiener Entropy	$\text{flatness}\left(X[n]\right) = \exp\left(\frac{1}{N}\sum_{n=0}^{N-1}\ln X[n]\right)/\frac{1}{N}\sum_{n=0}^{N-1}X[n]$
Spectral Entropy	Energy-entropy of spectrum (see above) $\text{H}\left(X\left[t\right], n\right) = -\sum_{i}^{n}\left\{e_i, \ln e_i\right\}$
Spectral Mean	Average of the spectrum (Hz) $\bar{X} = \sum_{n=1}^{N}X[n]$
Spectral Variance	Statistical variance $\text{Var}\left(X[n]\right) = \sum_{n=1}^{N}\left(X[n]-\bar{X}\right)^{2}$
Spectral Kurtosis	Fourth standardized moment $\text{Kurt}\left(X[n]\right) = \bar{X}_4/\text{Var}\left(X\right)^{2}$
Spectral Rolloff	85%-percentile of spectral energy
Spectral Skewness	Measure of left/right skewness $\text{Skew}\left(X[n]\right) = \bar{X}_3/\text{Var}\left(X\right)^{(3/2)}$
Spectral Spread	Variance about spectral centroid (above)
Musical Features	
Chroma Coefficients	Maximum normalized histogram of frequency bins centered around each of the 12 semitones $C, C\#, \ldots, B$
Quefrency-Based Features	
MFCCs	Mel-Frequency Cepstral Coefficients (see 4.0)

Table 2: Second order features breakdown.

Count	Second Order Features
38	Mean of each feature
38	Variance of each feature
38	Predicted BPM (Beats per minute) on feature
38	BPM Confidence on each feature
1	Aggregrated expected BPM
1	Aggregrated expected BPM confidence

5.2 Feature Aggregration

Initially, the audio is subdivided into windows of size $\sim$ 10ms and each of the 38 features outlined in section 5.1 is computed on those windows. However, in order to perform high-level classification tasks like the seperation of "music" and "speech" audio clips, the features need to be representative of the *entire* clip, not just the windows. With this as motivation, each of the features per window were aggregated into *second order* features. Firstly, the mean of all the values was taken as one aggregation. Secondly, the variance of the feature values were taken as the second aggregation of features. Finally, two more aggregations were made in order explore the phase space of the feature signal; beats-per-minute (BPM) and a BPM confidence (see section 5.3). In total each of the 38 features were aggregated into $4 \times 38 + 2 = 154$ second-order features. Note the +2 features are an tertiary aggregation on both BPM features.

5.3 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 [23]. One possibility is constructing recurrence plot (see figure 5). A recurrence plot is essentially an image where the pixel at the i, j coordinate is given by equation 4.

$$R\left[i, j\right] = \text{sim}\left(x[i], x[j]\right) \tag{4}$$

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 [22], 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 [21]. 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 6. The core idea of this algorithm is that distances between adjacent peaks should be evening 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 [17].

2. Ignore all minimums, only look at maximums (red dots in figure).

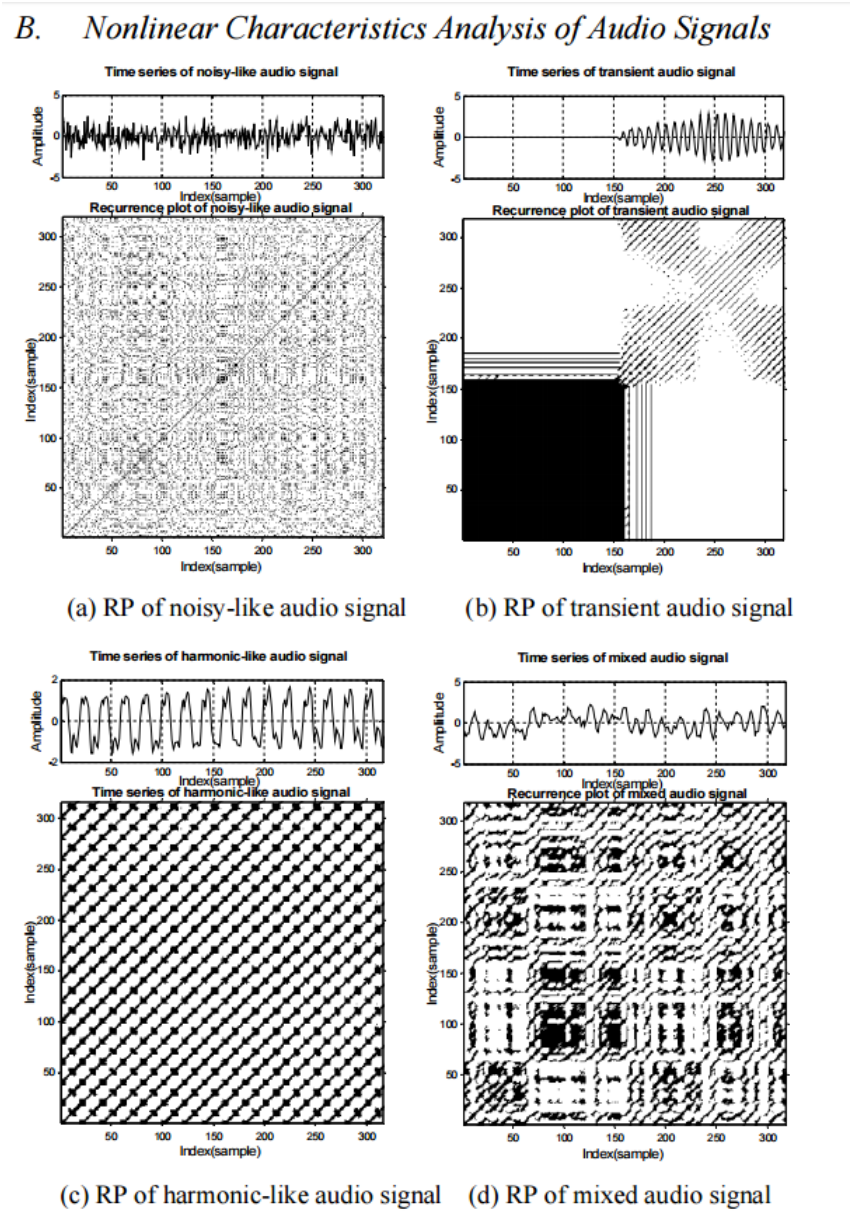

Figure 5: Recurrence plots of differnt audio signals. Taken from [22].

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 7).

5.3.1 Issues with BPM Measurements

When performing beat extraction, it is very 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 needed to be much faster than the BPM in order for it to be detected $f >>$ BPM.

5.4 Binary Classification

In order to compare the performance of MFCCs with other features of section 5.1 a two tier classification task was used. As outlined in section 3.4, one of the important aspects of this projects pipeline is a high-level

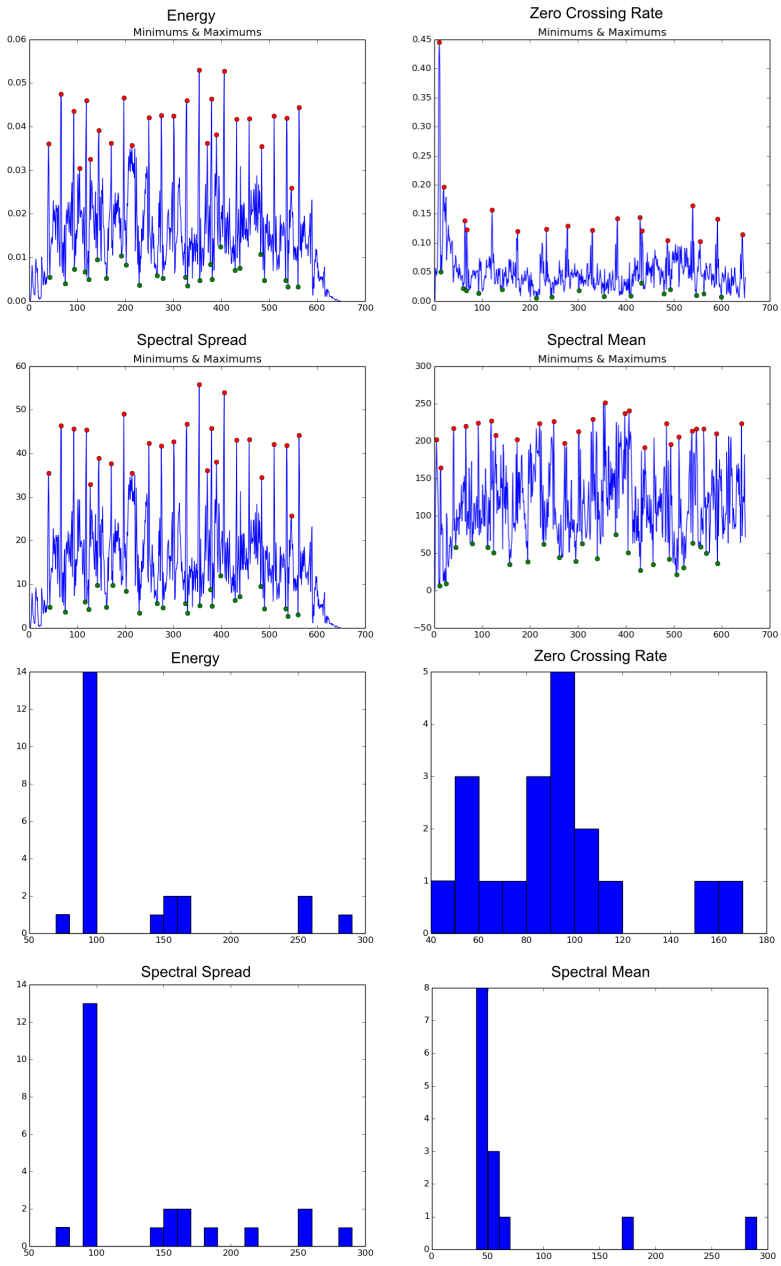

Figure 6: BPM prediction using fast $O(n)$ algorithm.

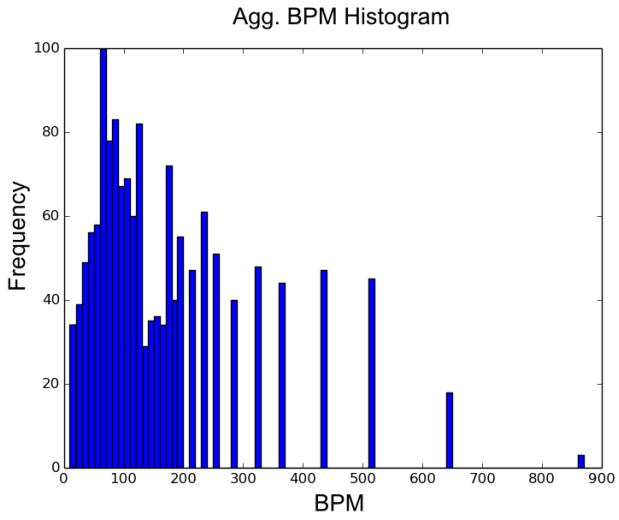

Figure 7: Aggregate BPM historgram.

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.

5.4.1 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 5.2 and ranked based off their classification accuracy used 10-fold cross validation. This allowed for the forward-selection of best, most correlated features to the two classes: music and speech.

5.4.2 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 validation. The values of k were allowed to vary to examine the covergence of accuracy and the potential for overfitting. Figure 8 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, so they have been omitted from this report.

5.5 Data Sets

The binary classification problem and methodology discussed in section 5.4 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.

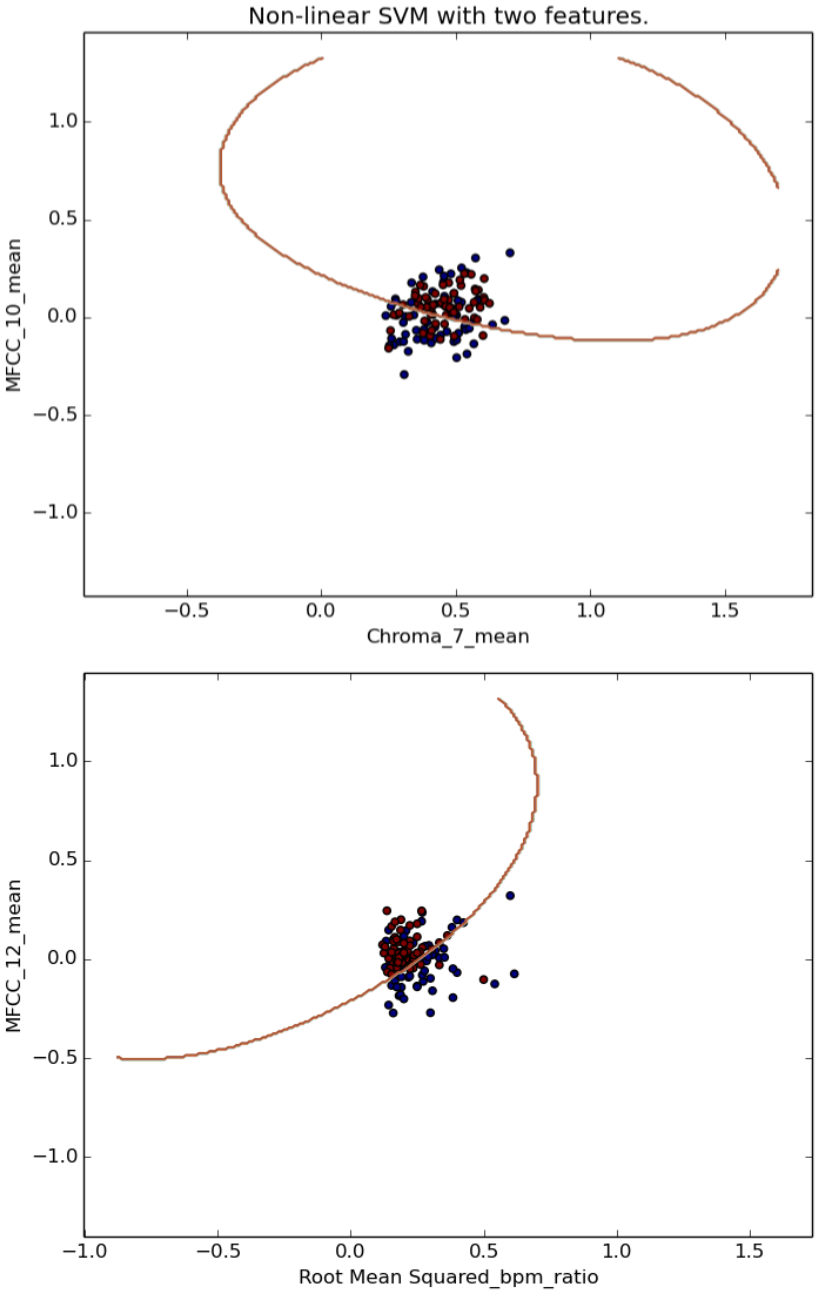

Figure 8: Two Example SVMs with RBF performed on the MIREX data set. Red = Music; Blue = Speech

5.5.1 Social Data

For the social media data set, approximately 2800 of the most popular videos of 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 of different languages, genres of music and audio environments. Every audio channel was encoded at 44.1 kHz.

5.5.2 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 [27]. 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) [8]. The MARSYAS data set consists of 120 audio clips each 30 seconds long with 60 belonging to each class. Each audio clip was encoded at 22.05 kHz.

5.6 Results

The results of the binary classification problem dictated in section 5.4 are found in this section. An interpretation of the results and their implications will follow in section 5.7.

5.6.1 Feature Rankings

Tables 5.0 and 5.0 are the single feature SVM classification accuracies after 10-fold cross-validation on each of the 154 features.

5.6.2 Classification Models

Tables 5.0 and 5.0 are the single feature SVM classification accuracies after 10-fold cross-validation on each of the 154 features.

5.7 Interpretation

... social is not pure

6.0 Conclusions

MFCCs model human interaction with audio.

...

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

Rank	**Feature Name**	**Classification Accuracy**
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

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

Rank	**Feature Name**	**Classification Accuracy**
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

SVM Model Social Data Results			
	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

SVM Model MIREX Data Results			
	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

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

...

MFCCs are the best features considered for speech modelling.

...

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

...

7.0 Recommendations

Fine-tune free parameters of MFCC computation.

...

Build full pipline using complete model.

...

Perform scalability analysis on pipeline.

...

Explore deep learning techniques to improve feature optimizations.

...

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

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

8 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/

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

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

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

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

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

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

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

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

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

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

19 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.

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

21 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.

22 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.

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

24 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

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

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

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

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

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

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

31 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.

32 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: [32] was referenced to format this report.

Glossary

Cross Validation A classification model validation technique used to predict how the given model will perform on real-world data. n-fold cross validation means take the sample data set of size N and randomly seperate it into n pieces. Then train the model on each of the n pieces except 1 to test the model with. Record the performance of the model and then repeat for each of the n pieces being the test set. This will minimize errors in the model due to overfitting.. 22

Feature In the domain of machine learning and data science, a feature is the result of performing feature extraction on a data source. Features act as the inputs to a classification or regression model. Deep-learning techniques auto-learn these features. Feature extraction is a form of dimensionality reduction. For example, the average age (the feature) of a soccer team (the data set) is 16. This reduces the data from n numbers to just 1.. 1, 2

Phoneme A term used in the study of linguistics, phonemes are the irreducible sound elements made by speaking. English has 44 phonemes such as /m/ as in *man, summer, palm* or /ow/ in *now, shout, bough*. MFCCs accoustically model these parts of speech.. 5