Entropy and the Estimation of Musical Ability

Andrew Russell

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Music and Technology

School of Music
Carnegie Mellon University
April, 2016

Thesis Committee:
Roger B. Dannenberg
Bhiksha Raj

Abstract

Musical ability determines how well a musician, or a band, can perform. From a listener’s perspective, a musician’s ability is often judged by the subjective overall impression of the music. However, others, such as teachers or talent seekers, use more specific criteria to determine a musician’s ability. Regardless of who is evaluating, judgements of musical ability are still subjective and require the judges to listen to recordings or live performances of the musicians in question. Automatic estimation techniques would greatly decrease the time required to determine a musician’s ability. Automatically estimating a musician’s ability would also be very useful for online communities of musicians to help users find other users with a similar musical talent.

In this thesis, the automatic estimation of musical ability is explored. More specifically, two features, both based on the concept of entropy, are proposed. The first feature looks at the rhythmic consistency of a recording, while the second looks at the tonal consistency. The performance and relative importance of each feature is studied by correlating the results of the feature with data that was manually labelled by 12 musicians. Using the rhythmic feature, a Pearson’s correlation coefficient of -0.55 with a p-value of 0.00052 was found, whereas the pitch feature had a coefficient of -0.32 and a p-value of 0.056.

Table of Contents

List of Figures

List of Tables

List of Equations

List of Code Charts

1. Introduction

Musical ability is defined as how well a musician can play their instrument. Is a musician able to play rhythmically on tempo with a band? Can that same musician sing a note in key? Musical ability is also commonly used to describe how well a musician can emotionally express themselves during a performance. However, we do not have a clear definition of ability. For example, if a listener greatly enjoys a musical performance for whatever reason, they might rate musical ability as high.

How musical ability is defined can also change drastically between genres. A musician can perform with a very high ability for one genre, but have a beginner's level performance for a second genre. Therefore, evaluating the musical ability of a musician across genres is a difficult task. This thesis focuses on just western rock and pop music to simplify the process. We also assume that we are analyzing polyphonic recordings.

Automatically recognizing musical ability is gaining importance for online communities of musicians as a tool for discovery. Previous works have shown that the best collaborations happen when the collaborators believe that they all have a similar skill level [1]. A study by Katira et al [2] looked at pairs of software developer students and attempted to predict their compatibility based on different attributes. They found that perceived skill level predicted a good match for all levels of developers while actual skill level predicted a good match for graduate level students.

There have recently been many communities created, such as Kompoz or Splice [3, 4], for musicians to find collaborators with whom to write or record songs. These communities rely on users manually sorting through listings of thousands of collaboration partners. Assuming that ability is a good way to find good collaborators, these online communities could help users by rating their abilities and suggesting collaborators with similar ability, interests, tastes, etc.

Additionally, automatic recognition of musical ability can be used by the music industry to discover new talent. Record labels could sort music on sites such as SoundCloud by their musical ability to help them find musicians that they would like to sign. Additionally, applications such as Spotify or iTunes could use this to recommend the better performed songs to their users.

In this thesis, the problem of ranking musical performances based on their rhythmic and tonal entropy is addressed. More specifically, this thesis discusses entropy, and defines how it is used. It will then suggest two new features, one based on the rhythmic entropy and the other based on the tonal entropy, which can be used to find the musical ability of the musicians in an audio recording.

The organization of the rest of this thesis is as follows. Section 2 presents the related work. In Section 3, entropy will be defined and discussed. Sections 4 and 5 define the rhythmic and tonal features used to find the musical ability. Section 6 will detail the evaluation of the new features. Finally, section 7 will conclude the thesis and present possible future work.

2. Related Work

The ability to measure the musical ability of a musician is not a new concept. One branch of research has attempted to draw upon the psychology of music to test musical ability. Other research has tried to measure musical ability by having the musician play along with a specific score. However, no work has attempted to automatically determine the musical ability of the musicians in an arbitrary song.

Studies of musical ability reach back to at least the early 1900's where Mursell showed that there was not yet a satisfactory test for musical ability [5]. Over the next few decades, profiles published by Seashore [6], Gordon [7], and others [8, 9] found some indication of musical ability based on various measures, such as interval and rhythm recognition. However, all studies are based on having the subject self-report or take a proctored test. Whereas the means of extracting information is different, this thesis attempts to measure a similar set of abilities.

Another area that has studied musical ability is score following. Score following attempts to map a musical performance of a song to a symbolic representation of the song. The Piano Tutor used score following to help train beginner piano students [10]. Students performed on a MIDI keyboard and the Piano Tutor would output errors such as bad timing or wrong notes. Smart Music [11] is a similar system; however it evaluates performances on wind and string instruments using pitch estimation techniques. Music Prodigy [12] and Yousician [13] are two other music learning systems which include the ability to evaluate multiple pitch instruments, such as guitar.

Over the last couple of decades, some video games, often called rhythm games, let users play along with popular songs. Some of these games use guitar-like instruments and electric drums as input, such as Rock Band and Guitar Hero. Rocksmith bridges the gap between video games and music education systems by letting musicians play along on any electric guitar, and provide both a video game experience and exercises to help train the guitarists [14]. These rhythm games award scores based on timing errors on wrong notes.

All of these systems that measure musical ability use score following or playing along with fixed media. They do not work when a score is unavailable. With western rock music, songs are often learned without scores and sometimes sections are entirely improvised. In this situation, we could attempt to transcribe the recording and determine the musical ability based on the resulting score. However, this is a very difficult problem [15, 16] as it involves figuring out what instruments are contained in the piece, then source separation of the instruments involved, both of which are difficult problems on their own. Furthermore, there may be no way to detect mistakes since we have no “ground truth” that tells what the musician was intending to play.

3. Entropy

To determine the musical ability of a recording without a score, we can instead look at methods that estimate pitch and rhythm accuracy. If we assume a recording has only notes played perfectly in tune, the resulting spectrogram would contain peaks at the frequencies in that song's scale and valleys elsewhere. This means that a well-played song, in terms of pitch, is missing the "out of tune" frequencies. Similarly, a song that has very consistent rhythms would have a consistent periodicity between the onsets. This could be determined using a beat histogram, which is an audio feature often used for determining rhythmic similarity. This would mean that a perfect song in a rhythmic sense would only have timing intervals that match the tempo and the meter of the song.

To allow us to use the frequency spectrogram and beat histogram to measure pitch and rhythm as measures of musical ability, we require a method to extract the consistency. To explore this concept further, we will describe the concept of entropy, which plays an important part in our work.

In general, the concept of entropy has been used by physicists to define the amount of energy lost in reactions. More specifically, the second law of thermodynamics states that the amount of entropy in any isolated system will increase [17]. The use of entropy has also been adopted by other fields to describe similar concepts. For example, information theory defines entropy as the loss of data in information transmission systems [18].

Entropy has also been adapted for signal processing. Ekstein and Pavelka [19] proposed a definition where a system of maximum entropy is a system of only noise. Other parts of the system, such as speech or music, would have lower entropy. This is also defined so that the more periodic a source is, the lower the entropy will be. The entropy for a source signal is defined in Equation (3.1).

$$Entropy = -\sum_{x} x * \log x \tag{3.1}$$

This version of entropy has been used for signal processing in algorithms such as automatic speech recognition (ASR) [20] and in detecting vision impairments in EEGs [21]. In this thesis, we will propose a new usage of this entropy for detecting the musical ability in a recording.

If consistency is the hallmark of musicianship, one might expect a simple measure such as standard deviation would be a good estimator. This would be true if we were looking for consistent values around a fixed mean value. However, even at a steady tempo, we expect different rhythmic values clustered around common quantities such as eighth notes and quarter notes. With pitch, we expect frequencies determined by discrete scale steps, with few frequency components in between. In both cases, we are looking for concentrated clustering around a relatively small set of means. This is a more general problem than standard deviation. We could consider forming clusters and estimating the standard deviation of each cluster, but entropy seems to be a simpler model that accomplishes much the same thing. Entropy is higher when values are clustered, and lower when values are random and independent.

4. Rhythm

One of the methods of estimating the musical ability in a song is to use rhythmic features. At a high level, the idea is to detect when the musicians are playing notes at the same time as each other and to use that information to compute their musical ability. We then correlate the extracted features with the user labelled data set to determine the usefulness of the features.

To extract the features, we first build a beat histogram, and then compute the entropy using the following steps:

1) *Detect Onsets:* We find the onsets of the entire audio recording.
2) *Calculate Onset Diff Histogram:* We subtract the timestamp of each onset with the timestamp of its neighbor and store the results in a histogram.
3) *Parzen Smoothing:* We smooth the histogram using Parzen Smoothing.
4) *Calculate Entropy:* We calculate the entropy of the smoothed histogram.

4.1. Onset Detection

To detect the onsets, we use an off-the-shelf onset detector, aubio [22], which "is a tool designed for the extraction of annotations from audio signals". We chose aubio since both the algorithms it uses and its APIs are well documented. aubio comes packaged with a command line program called "aubioonset" which we ran on each audio file with the default settings. It outputs a list of timestamps, in seconds, for which onsets were detected.

aubioonset has eight different onset detection algorithms: energy based difference, high-frequency content, complex domain, phase based, spectral difference, Kullback-Liebler, Modified Kullback-Liebler, and spectral flux [23]. We used the default onset detection function which is high-frequency content. High-frequency content is calculated for each frame by linearly weighting each frequency bin, and then summing all of the weighted bins together as shown in (4.1) [24]. aubioonset also allows other options for the buffer size, hop size, onset threshold value, and silence threshold value. We decided to use the default values of 512, 256, 0.1, and -90 dB respectively for each option.

$$HFC = \sum_{i=1}^{len(X)} i * abs(X[i]) \tag{4.1}$$

To find the actual onset time from the high-frequency content, aubio looks for the peaks using a moving mean with an adaptive threshold. The peak picker tracks the mean of the current high-frequency content value along with the previous six values. It then computes a new value with an adaptive threshold using (4.2). The peak picker then compares the result to the values from the previous two frames and detects a peak is the middle value is the largest of the three and is greater than 0. If the volume of the audio is then louder than the silence threshold value, the peak is declared an onset.

$$peak = HFC - median - (mean * onset\ threshold) \tag{4.2}$$

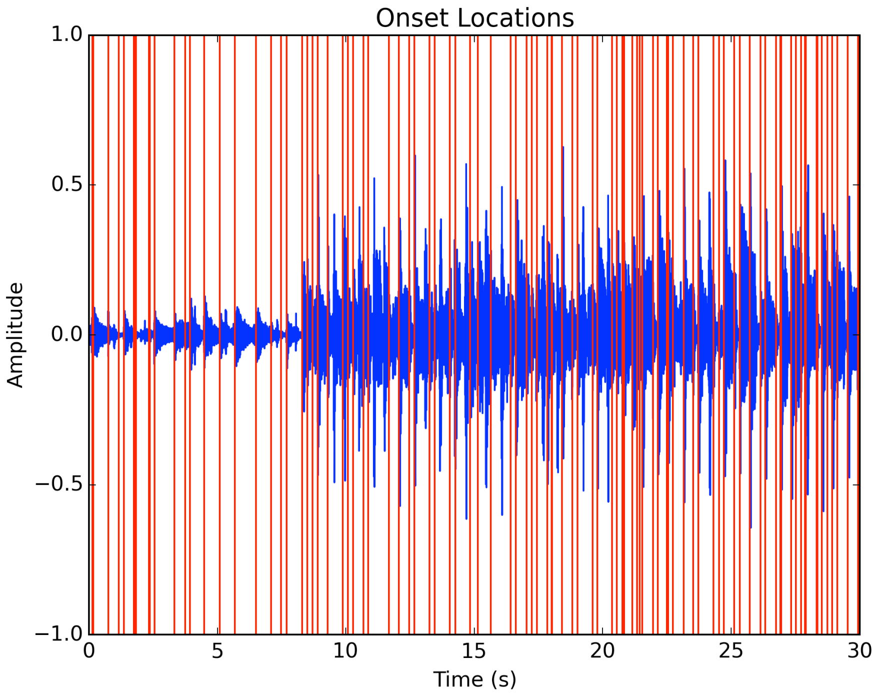


Figure 4.1: Position of onsets in a performance of amateur musicians (song 2)

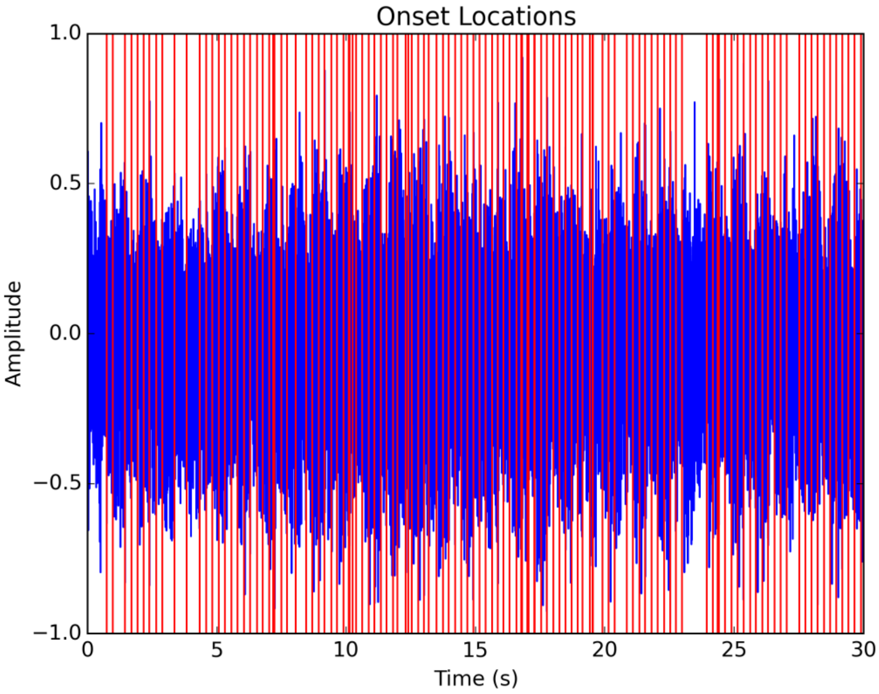


Figure 4.2: Position of onsets in a performance of professional musicians (song 38)

4.2. Onset Diff Histogram

Given the list of onset times, we sort the list in ascending order, and compute the difference in time between each adjacent element in the list. This gives us a list of timings between onsets. We then organize the list into a histogram to get a count for each onset timing difference. Code 4.1 presents Python-like pseudo code that could be used to compute the histogram.

```
def compute_onset_diffs_histogram(onsets):
  diffs = []
  for i in range(1, len(onsets)):
    diffs.append(onsets[i] – onsets[i – 1])

  histogram = {}

  for diff in diffs:
    histogram[diff] = (histogram[diff] || 0) + 1

  return histogram
```

Code 4.1: The algorithm that creates a histogram of onset timing differences.

We only compute the difference against the neighboring onsets and not between any other onsets since the difference between other onsets would just calculate information that is already present in the neighboring onset differences. Consider a perfectly played song that contains just quarter notes and the occasional half note. The onset diff histogram with just neighbors would have a peak at the quarter note value, and a smaller peak at the half note value. If we also included the difference between onsets two locations away, two peaks would appear at the values for half notes and whole notes that have the same size as the peaks current peaks for quarter notes and half notes.

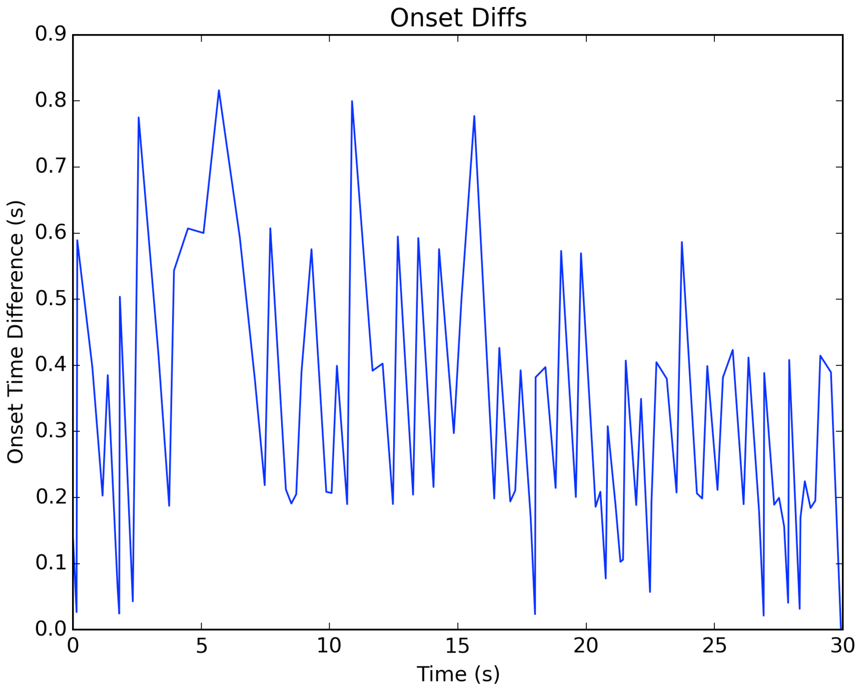


Figure 4.3: Onset differences over time for a recording of amateur musicians(song 2)

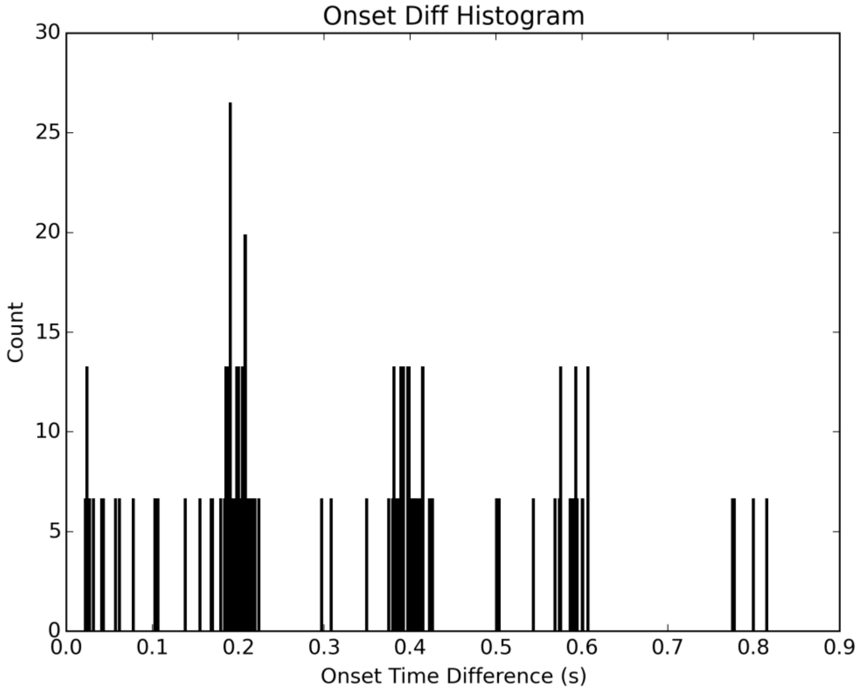


Figure 4.4: Onset histogram for a recording of amateur musicians (song 2)

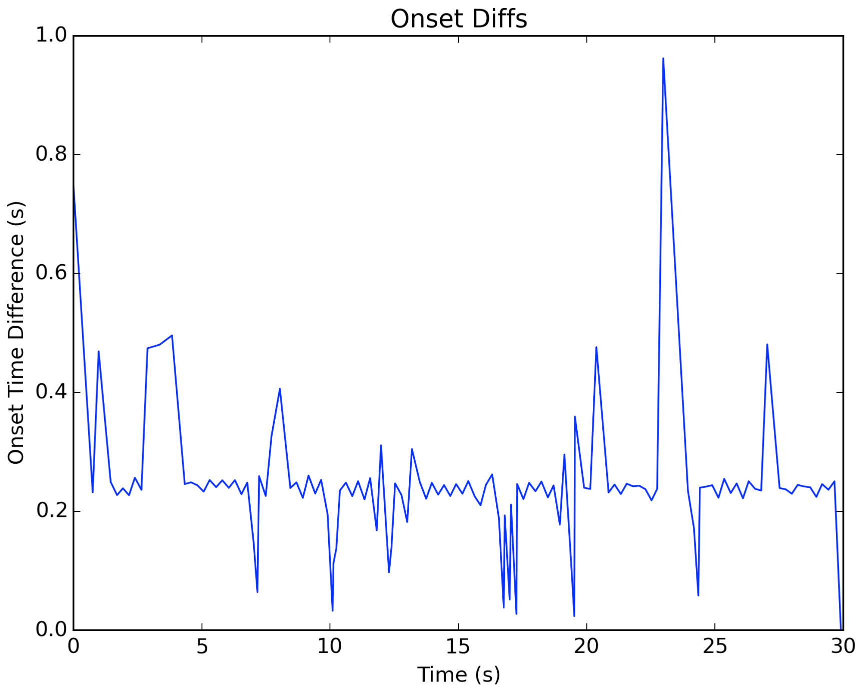


Figure 4.5: Onset differences over time for a performance of professional musicians (song 38)

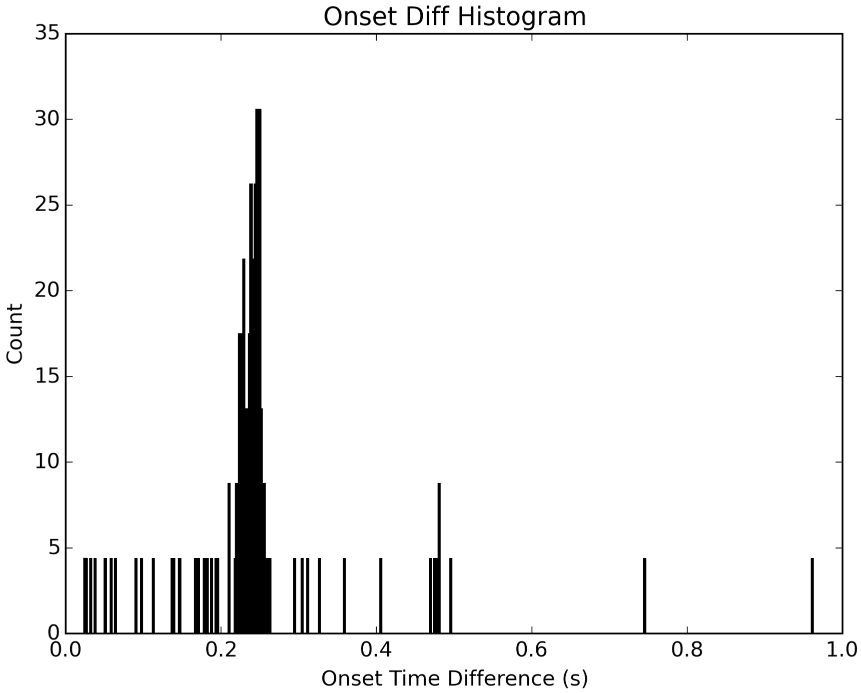


Figure 4.6: Onset histogram for a performance of professional musicians (song 38)

4.3. Parzen Smooting

We then use Parzen smoothing, which is also known as kernel density estimation, to estimate the probability density function. This is done so that we correct for the statistical errors in the observation of the onsets. Parzen smoothing works by taking a specific shape to use as a kernel in estimating the actual data's shape [25]. We use Gaussian kernels as the starting shape and use Scott's Rule [26] to detect the optimal bandwidth of the Gaussian kernel. A Gaussian shape was chosen for the kernel as we assume that the musicians' mistakes are normally distributed. The bandwidth of the kernel, or in this case the width of the Gaussian curve, is a free parameter that has a strong influence on the result of the smoothing. Scott's Rule is a rule-of-thumb bandwidth selector which attempts to optimize the bandwidth based on the length of the histogram [27]. The equation for Scott's Rule is seen in (4.3).

$$Scott's\ Rule\colon n^{-1/d+4}, \qquad n = array\ length, d = dimensions \tag{4.3}$$

4.4.Entropy

We then compute the entropy of the recording. This is done by calculating the Shannon entropy of the smoothed onset timing difference histogram. We use the implementation from SciPy which calculates (3.1) [28]. SciPy uses the natural logarithm in the calculation.

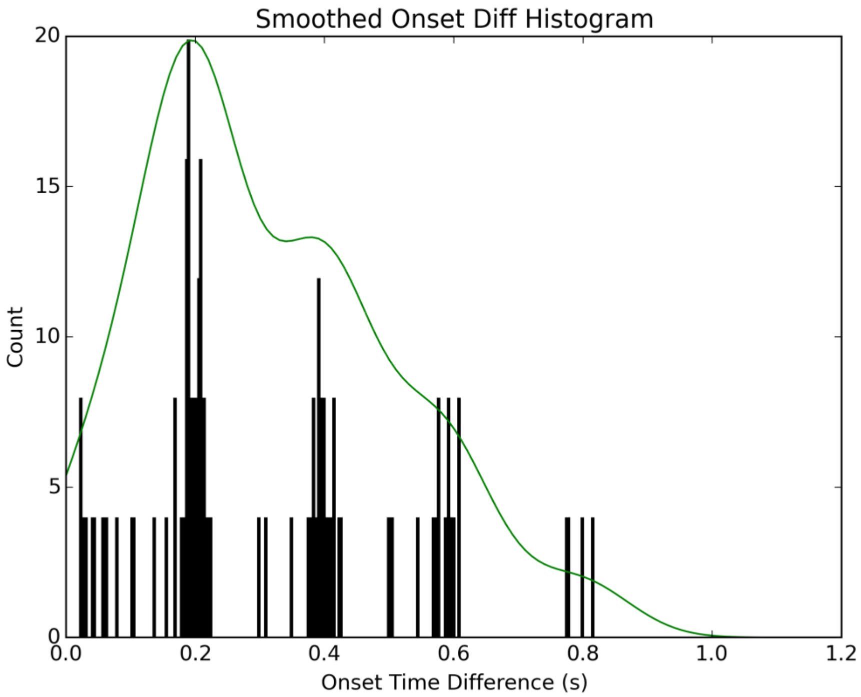


Figure 4.7: Smoothed histogram of a performance of amateur musicians (song 2). The black bars of this graph represent the onset difference histogram while the green line represents the Parzen smoothed probability density function. The density function has been vertically scaled on this graph so that its overall shape is easier to see.

4.5. Discussion

As implemented, the rhythmic feature makes some assumptions. First, it depends on the detection of note onsets. This is currently an area of active research and no algorithms are perfect. This means that there will be both false positives and false negatives in our onsets which will propagate error throughout the entire feature. Additionally, we assume that the audio recording has a consistent rhythm and a steady, unchanging tempo. A study of tempo in rock and jazz recordings showed that recordings by professional performers (even without using click tracks) have more constant tempi than recordings by amateur performers [29].

However, a rhythm or a tempo that changes will introduce more entropy, even if the musicians changed the rhythm or tempo on purpose and did so skillfully. Finally, this algorithm assumes that the musicians are all trying to play their notes exactly on the beat. However, this is not always the case as sometimes, musicians will play either ahead or behind of the beat to create a different feel in the music. Again, this could be a positive indication of musical ability rather than a negative one.

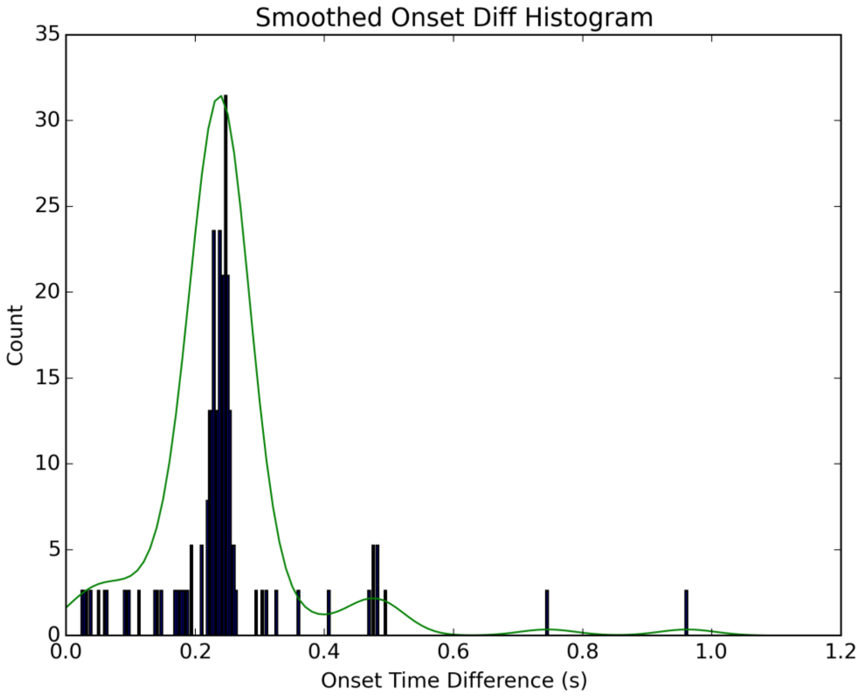


Figure 4.8: Smoothed histogram of a performance by professional musicians (song 38). The black bars of this graph represent the onset difference histogram while the green line represents the Parzen smoothed probability density function. The density function has been vertically scaled on this graph so that its overall shape is easier to see.

5. Pitch

The pitch feature to measure musical ability is based on analyzing the spectral content of the audio. At a high level, the algorithm looks for spectral smearing which indicates that the musicians are playing out of tune with each other. This is done by first calculating the loglog frequency spectrum of the recording and then computing the entropy of the spectrum.

To extract the pitch feature, we first split the audio file into chunks of 64,000 samples, using an overlap of 50%. Each chunk has a Hamming window applied to it then has the Fourier transform computed. Only the magnitude of each bin is stored. Each bin is summed across every chunk, and then averaged so that we end up with a single 64,000 bin frequency spectrum of the entire audio file.

We then transform the frequency spectrum into a loglog scale. We use decibels for the magnitude, and a log base 10 scale for the frequencies. A log scale was chosen for the frequencies so that the lower frequencies have more emphasis. This is because most melodic content is present in the lower frequencies of the audio spectrum, whereas the higher frequencies contain more harmonic content.

The entropy of the frequency spectrum is then computed. Again, SciPy's implementation of Shannon entropy is used with a natural logarithm.

The pitch feature described above makes some assumptions about the how the recording was performed. First, it assumes that there is no vibrato. Since vibrato rapidly varies the pitch of the note by small amounts, any vibrato will add more entropy. Similarly, if the overall pitch of the song drifts over time, more entropy will be detected. Also, fixed pitch instruments, such as a MIDI keyboard, will have less entropy than instruments without fixed pitch, such as violins. Finally, if the recording changes key part way through, more entropy will be present as more unique pitches will have been played.

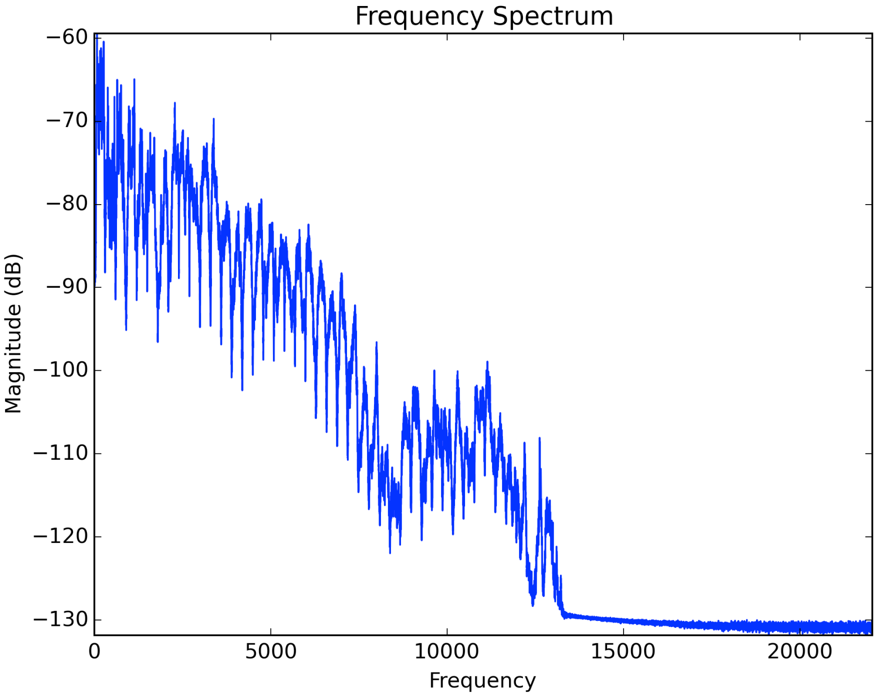


Figure 5.1: Frequency spectrum for an amateur recording (song 15)

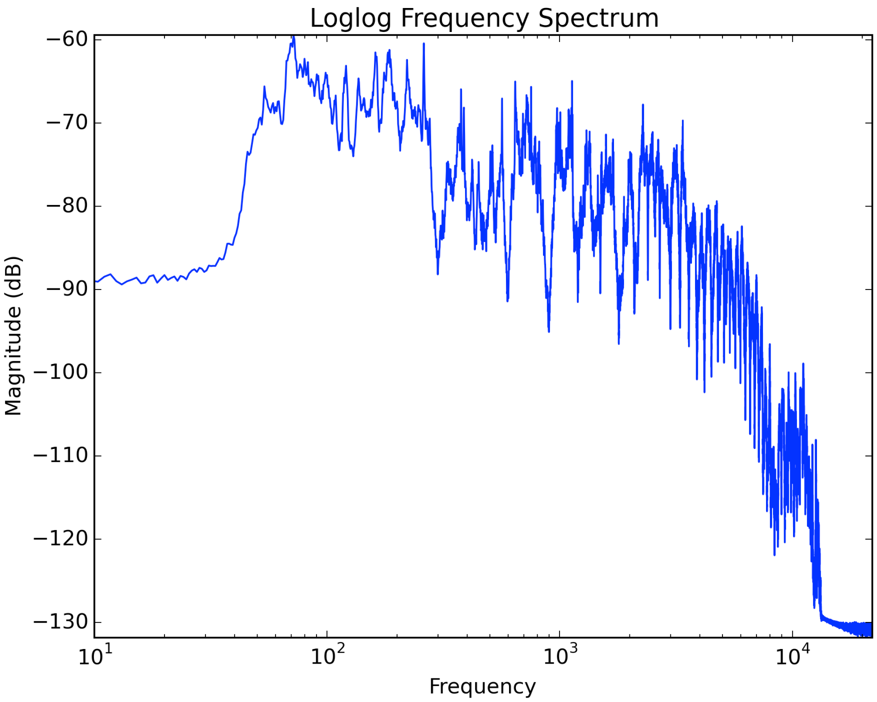


Figure 5.2: Loglog frequency spectrum for a performance of amateur musicians(song 15)

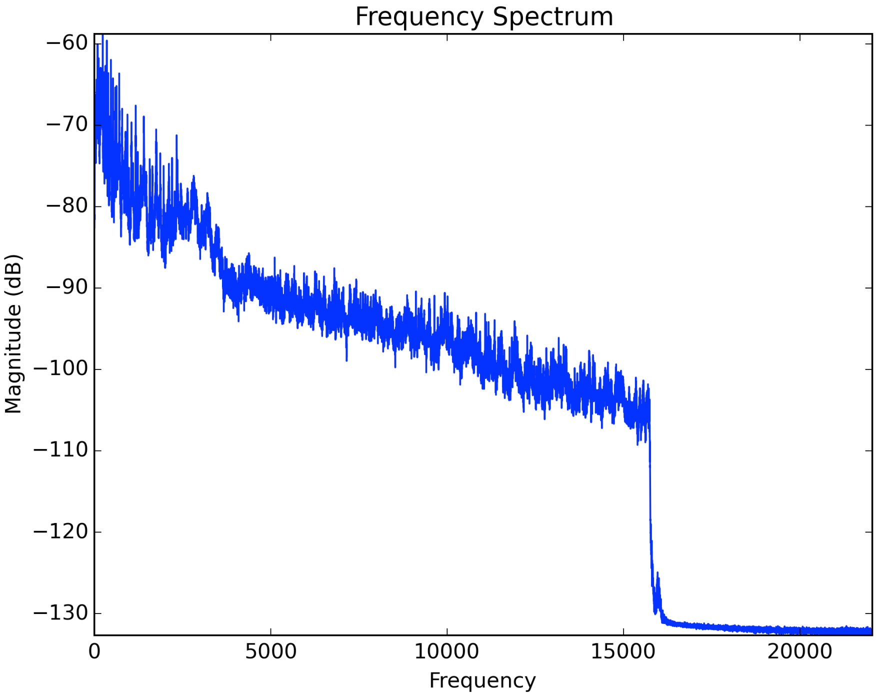


Figure 5.3: Frequency spectrum for a performance of professional musicians (song 22)

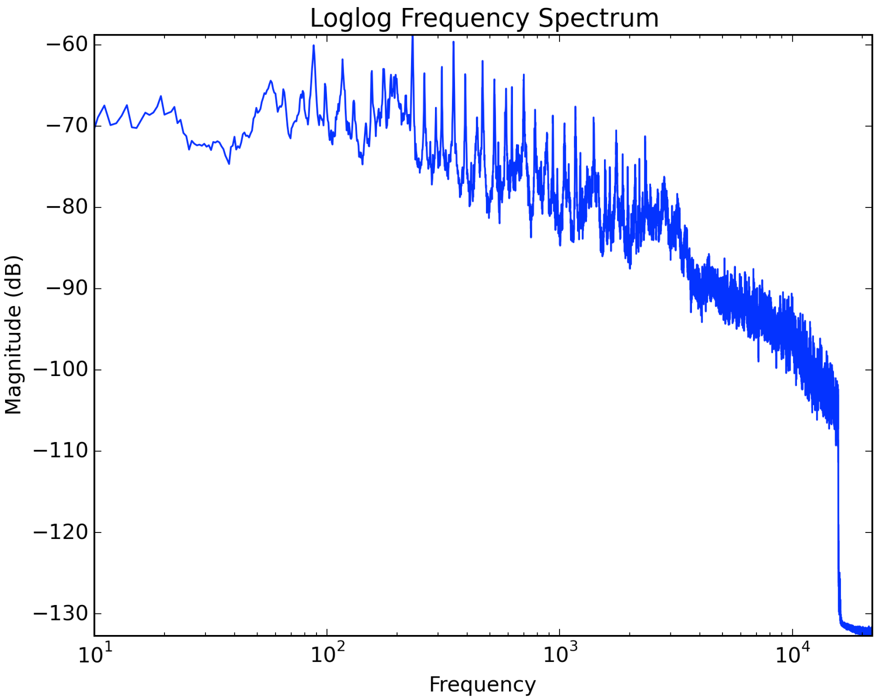


Figure 5.4: Loglog frequency spectrum for a performance of professional musicians (song 22)

6. Evaluation

To evaluate our entropy based measures, we collected music and used human subjects to rate it for musical ability.

6.1. Data Collection

The data for this project consists of two versions of twenty one different songs. One version is performed by an amateur band and the other is performed by a professional band. The songs were taken from YouTube, and a list of the songs can be found in Appendix 1.

Each song was downloaded from YouTube and was clipped to a thirty second long file. The thirty seconds chosen for each song were the same for the both the amateur and professional categories. For example, if the first chorus was chosen for the amateur version of “Song A”, the first chorus would also be chosen for the professional version of “Song A”. The songs were also all normalized in volume and converted to the mp3 format. All processing was completed using Audacity.

The songs were then rated by other members of my research group. This consisted of twelve users, and each user rated a random selection of twenty songs, ten from each of the amateur and professional categories. Each rater first listened to three sample songs which were accompanied with suggested ratings for each sample. A list of twenty songs, from the forty-two song set, was then randomly generated. Each rater was given ten random amateur songs and ten random professional songs in a random order. The rating scheme used is shown in Table 6.1.

Table 6.1: The rating scheme used by the raters to evaluate each recording.

Rating Level	**Rating Description**
5	Flawless performance
4	Minor mistakes that do not detract from the performance
3	Many mistakes that do not detract from the performance
2	Many mistakes that detract from the performance
1	More mistakes than playing; hard to listen to

The raters were also to keep in mind that the songs were to be rated on performance ability, and not on production qualities such as editing, mastering, microphone placements, etc., which can greatly affect a listeners judgement of a song. The musical ability was defined as the rhythm and intonation of the performers, both in their solo performance, and in their performance with other band members. Each song was encoded as a low bitrate mp3 to attempt to adjust for the bias introduced by differences in overall production quality.

6.2. Data Processing

Once the data was collected, it had to be preprocessed to account for a rater's bias. For example, if one rater was harsher (rated lower) than the rest of the raters, that first rater will have their ratings adjusted higher to match the other rater's ratings. This was done by calculating the z-score of each rating based on the standard deviation and average of its rater's, as shown in (6.1). The overall average z-score of each song was then computed.

$$z - score = \frac{x - \mu}{\sigma} \tag{6.1}$$

6.3.Results

To determine how useful our rhythm and pitch features are, we found both the rhythm and pitch entropy values for every song in our dataset. A correlation value for each feature was calculated against the user labelled data. The results of the correlations can be seen in Table 6.2. To calculate the correlations, we used Pearson's correlation coefficient, a measure of correlation between two normal data sets, with two-tailed p-values [30]. The use of Pearson's correlation coefficient assumes that the musical ability of all songs follows a normal distribution. More specifically, the majority of songs are assumed to have an average musical ability rating, while really well performed songs and really poorly performed songs are assumed to be much rarer.

Table 6.2: The correlation results of the rhythm and pitch features.

	r-value	p-value
Rhythm Entropy	-0.55	5.2e-4
Pitch Entropy	-0.32	0.056

Figure 6.1 and Figure 6.2 plot each song's rating versus its entropy value. As expected, the professional performances tend to have higher user ratings and lower entropy values while the amateur performances tend to have lower user ratings and high entropy values. However, it is interesting to note that not all professional performances scored well, both by the raters

and by the entropy features. Additionally, some amateur performances scored well by both the raters and the entropy features.

To be able to put our entropy correlations in perspective, we can compare the correlations of entropy to those of the human raters. Since we consider the average z-score of all raters to be the gold standard, we can compute the average z-score of all but one rater to get an almost-gold standard and then evaluate the held-out rater in the same way we evaluate entropy, using correlation. We evaluate all raters by holding out one at a time and recomputing the “almost-gold standard” each time. We then compute an overall average correlation as well as the standard deviation. This then estimates the expected correlation of any human rater, which we can then compare to the entropy correlations.

The results of this inter-user evaluation can be seen in Table 6.3. The rhythm entropy correlates well within one standard deviation of the human raters (0.05 difference in correlation) while the pitch entropy correlation is almost one standard deviation below the average human raters (-0.18 difference). This suggests that our rhythm entropy feature is as efficient as an average human rater, while the pitch entropy feature performs worse than an average human rater.

Table 6.3: The average and standard deviation of the correlation of each human rater

Average Correlation	Standard Deviation
0.50	0.21

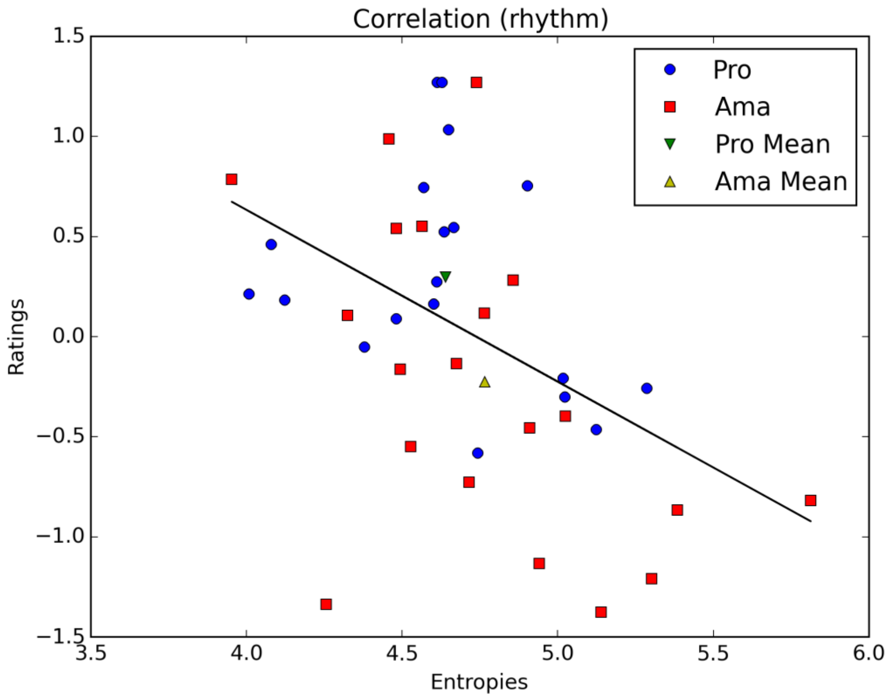


Figure 6.1: Amateur and professional rhythm entropies vs. ratings

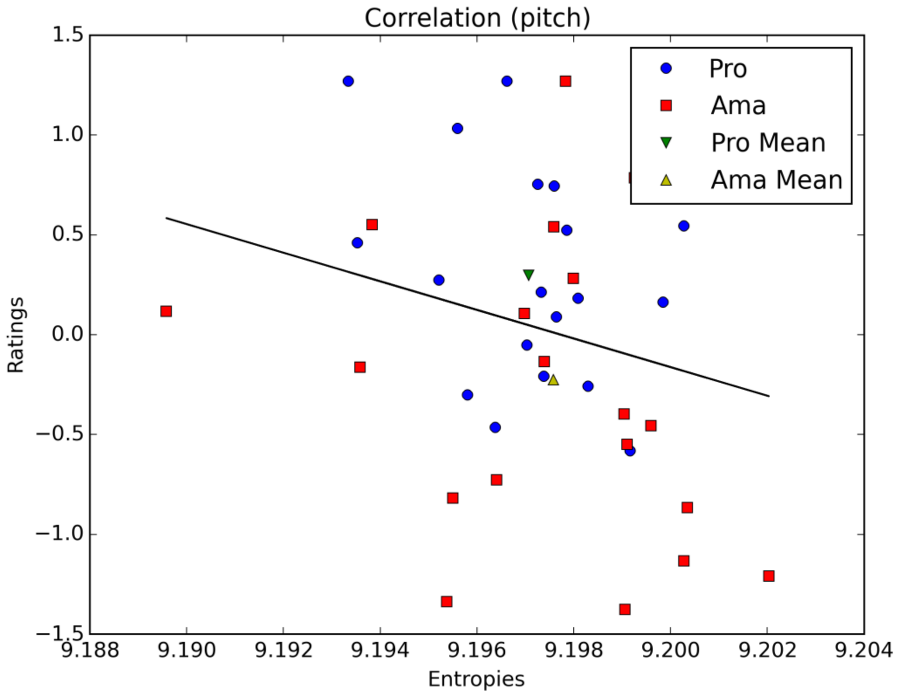


Figure 6.2: Amateur and professional pitch entropies vs. ratings

6.4.Parzen Smoothing

After conducting this study with Parzen smoothing, we also tried computing the rhythm entropy without Parzen smoothing. Intuitively, the current width of the Gaussian kernels is too wide. An example of this can be seen in Figure 4.7, where there are four distinct peaks in the underlying histogram data at 0.2s, 0.4s, 0.6s, and 0.8s. However, the Parzen smoothed curve barely distinguishes between these peaks. One might think that that either reducing the width of the kernels, or removing the Parzen smoothing altogether will improve the rhythm entropy results.

However, the results without Parzen smoothing seem to be significantly worse, with the rhythm entropy having an r-value of -0.14. This is counterintuitive and deserves further exploration. Unfortunately, there are methodological problems to this pursuit. If we were to compute the p-value as before, it would be 0.41, which would not indicate a significant correlation. Unfortunately, we cannot make this conclusion because this would be an incorrect way to compute the p-value. That is because if we reuse the relatively small set of subjective ratings with different parameters (such as Parzen smoothing kernels), the probability of finding a high correlation is increased. (This is a form of overfitting the data.) The question of significance changes from “is the probability of the null hypothesis low” to “is the probability of many related null hypotheses low.” Ignoring this distinction leads to greater significance than warranted by the data. The best approach would be to gather new data to evaluate new techniques. Lacking more data, we cannot evaluate different Parzen smoothing parameters, so we leave open the question of whether our approach is best.

7. Conclusions and Future Work

7.1.Conclusions

This thesis has introduced two new features, one based on rhythm and the other based on pitch, to determine the musical ability of the musicians in a recording. The features both take in an audio recording as an input and both output a scalar entropy value. The features were correlated against user-labelled data to determine how useful each feature was. The results of the evaluation show that the two features have potential to be used as indicators of musical ability in musicians. In particular, rhythmic entropy had a very high correlation with subjective ratings (r = -0.59, p = 0.00015). Pitch entropy appears to be correlated (r = -0.32) but the correlation was not significant in this study (p = 0.056).

7.2. Future Rhythm Entropy Work

There is much future work that could be done to enhance both of the features. One such improvement would be to not penalize recordings that contain intentional rhythmic or tempo changes. It would be interesting to look into detecting when these changes happen and then to determine the best way to correct for these changes. This could possibly be done by computing the entropy for each distinct section on its own and then combing the resulting entropy values.

Another area that could use more research is detecting when a musician is playing ahead of or behind the beat. We can speculate that a musician playing perfectly off beat will produce a smoothed beat histogram that has consistently wider peaks, due to one onset always being slightly delayed from its neighbor. However, more work is required to determine the best way to correct for this behavior. However, if the tempo changes happened for non-musical reasons, this approach might lead to false positives. Automatic evaluation of high-level musical decisions (or accidents) seems to be very challenging.

In the rhythm feature, we treated all timing errors as absolute values, meaning that missing a quarter note by 100ms would penalize the musical ability the same amount that missing an eighth note by 100ms would. However, this is not necessarily the case and both the production and the perception of timing errors may be relative. Future work should be done to consider how logs of the inter-onset times would perform rather than the absolute inter-onset times.

7.3.Future Pitch Entropy Work

There is also room for future work on the pitch feature. One such area is not penalizing recordings for changing the key part way through a recording. Similar to tempo changes, this could possibly be solved by detecting key changes and then computing an entropy value for each section individually. However, more research would have to be done to determine the best method for doing so.

Finally, more work needs to be done to allow the addition of vibrato in the recordings. Whereas vibrato adds smearing to the frequency spectrum, a good vibrato would have a consistent smearing pattern in the spectrum. More work could be done to determine how to best handle the vibrato.

7.4.Future New Features

We have also thought about other features that could be used to detect the musical ability. Tone production is a large area of musicianship that we have ignored. A note can be played perfectly in time and on pitch but with a poor sound. However, we believe that most instruments will require their own measure, as the nature of “good sound” varies greatly between instruments and playing styles.

Currently, the rhythm feature conflates two areas of musicianship, playing in tempo, and playing on beat. We believe that splitting the rhythm feature into two features, one to track how steady the tempo is and the other to track how far off beat each note is played, will improve the results of measuring the musical ability. Splitting the rhythm entropy into these two features could also make the issues of tempo changes and playing off beat easier to solve.

With the possibility of more features being available to measure the musical ability, the question of how to combine these measures to get an overall measure arises. We believe that using standard machine learning techniques, such as neural networks or support vector machines, to combine the features would help solve this problem. However, more work would be needed to verify this.

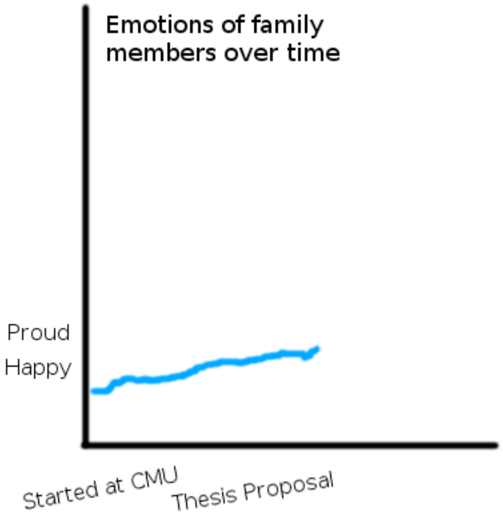
7.5.Source Code and Data

The software and data used for this paper has been published online and can be found at the following location: https://www.github.com/deadheadrussell/thesis. Documentation for how to compile and run the code to replicate the results of this thesis is provided. All provided code is licensed under the MIT license, the text of which can be found at the above location.

Acknowledgements

Some students are able start their masters, ace all of their courses, and finally, defend a brilliant thesis in just 18 months (and then go on to complete a Ph.D. thesis in just 32 months). Others struggle to find a topic, take forever to get some results, and then let a full year pass before finishing even the first full draft. That first student is the father to my fiancée, David Machina. You can guess who that second one is.

When I started this degree, my parents were of course proud of me. A graph of their emotions might have looked something like the following:



However, all good things must come to an end, and as I dragged this project along, the graph of their emotions became the following:

Emotions of family members over time
Annoyed
Proud
Happy
Started at CMU
Thesis Proposal
???

However, they've stuck with me, and I am very thankful for all of the support that they have given me. I also am grateful for the support from my brother, sister, and the rest of my family.

I also would like to thank my fiancée, Kristen Machina, for not constantly nagging me to finish my thesis when she had every right to. This also extends to her family, which will also soon be my family too, especially since they have a prodigy of completing theses among them.

This master's thesis would not have been the same without the countless people at Carnegie Mellon who helped me along. First and foremost among them is my advisor Prof. Roger Dannenberg. His feedback and advice throughout this process was monumental for this research. Without him, none of this would have come to fruition.

Also among the faculty, I would like to thank Prof. Bhiksha Raj for sticking around on the thesis committee. He always made key insights at the right times. I would also like to thank Prof. Riccardo Schulz, Prof. Rich Stern, and Prof. Tom Sullivan for their feedback along the way.

There were also many other friends, new and old, who helped me along through this process. The friends I made out in Pittsburgh, Haris Usmani, Tina Liu, Sean Brennen, Robert Kotcher, Nick Coronado, Alan V., Greg Hannenman, and too many others to list here, as well as the old friends with whom I stayed in touch. Their friendship and support was very important as I slogged my way to the finish.

References

[1] P. J. Guinan, J. G. Cooprider and S. Faraj, "Enabling software development team performance during requirements definition: A behavioral versus technical approach," *Information Systems Research,* vol. 9, no. 2, pp. 101-125, 1998.

[2] N. Katira, L. Williams, E. Wiebe, C. Miller, S. Balik and E. Gehringer, "On understanding compatibility of student pair programmers," *ACM SIGCSE Bulletin,* vol. 36, no. 1, pp. 7-11, 2004.

[3] "Kompoz Music Collaboration," Kompz, 2016. [Online]. Available: http://www.kompoz.com/. [Accessed 11 04 2016].

[4] "Splice - Music Made Better," Splice, 2016. [Online]. Available: https://splice.com/. [Accessed 11 04 2016].

[5] J. L. Mursell, The psychology of music, WW Norton & Co, 1937.

[6] H. M. Stanton, "Seashore measures of musical talent," *Psychological Monographs,* vol. 39, no. 2, p. 135, 1928.

[7] E. Gordon, Musical aptitude profile, Houghton Mifflin, 1965.

[8] T. H. Madison, "Interval discrimination as a measure of musical aptitude," *Archives of Psychology (Columbia University),* 1942.

[9] L. N. Law and M. Zentner, "Assessing musical abilities objectively: Construction and validation of the Profile of Music Perception Skills," *PloS one,* vol. 7, no. 12, p. e52508, 2012.

[10] R. B. Dannenberg, M. Sanchez, A. Joseph, P. Capell, R. Jospeh and R. Saul, "A computer-based multi-media tutor for beginning piano students," *Journal of New Music Research,* vol. 19, no. 2-3, pp. 155-173, 1990.

[11] R. B. Dannenberg and C. Raphael, "Music score alignment and computer accompaniment," *Communications of the ACM,* vol. 49, no. 8, pp. 38-43, 2006.

[12] "Music Education Software | Music Prodigy," Music Prodigy, 2016. [Online]. Available: http://www.musicprodigy.com/. [Accessed 11 04 2016].

[13] "Yousician," Yousician, 2016. [Online]. Available: https://get.yousician.com/. [Accessed 11 04 2016].

[14] B. J. Miller, "Music Learning through Video Games and Apps: Guitar Hero, Rock Band, Amplitude, Frequency, and Rocksmith, and Bandfuse, and Bit. Trip Complete, and Audiosurf, and Beat Hazard, and Biophilia (review)," *American Music,* vol. 31, no. 4, pp. 511-514, 2013.

[15] C. Dittmar, E. Cano, J. Abeßer and S. Grollmisch, "Music information retrieval meets music education," *Dagstuhl Follow-Ups,* vol. 3, 2012.

[16] A. Klapuri and M. Davy, Signal processing methods for music transcription, Springer Science & Business Media, 2007.

[17] E. A. Guggenhein, Thermodynamics: An Advanced Treatment for Chemists and Physicist, vol. 2, North-Holland, 1949.

[18] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Computing an Communications Review,* vol. 5, no. 1, pp. 3-55, 2001.

[19] K. Ekštein and T. Pavelka, "Entropy and entropy-based features in signal processing," in *Proceedings or PhD workshop systems & control*, 2004.

[20] T. Pavelka, "Hybrid Methods of Automatic Speech Recognition," *University of West Bohemia in Pilsen,* 2009.

[21] V. Vijean, M. Hariharan, S. Yaacob, M. N. B. Sulaiman and A. H. Adom, "Objective investigation of vision impairments using single trial pattern reversal visually evoked potentials," *Computers & Electrical Engineering,* vol. 39, no. 5, pp. 1549-1560, 2013.

[22] "aubio, a library for audio labelling," 2015. [Online]. Available: http://aubio.org/. [Accessed 12 04 2016].

[23] "Man page of AUBIOONSET," 18 12 2013. [Online]. Available: http://aubio.org/manpages/latest/aubioonset.1.html. [Accessed 12 04 2016].

[24] P. Masri, *Computer modelling of sound for transformation and synthesis of musical singals.,* University of Bristol, 1996.

[25] "scipy.stats.gaussian_kde - SciPy v0.17.0 Reference Guide," 20 02 2016. [Online]. Available: http://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.gaussian_kde.html.

[Accessed 12 04 2016].

[26] D. W. Scott, "Scott's Rule," *Wiley Interdisciplinary Reviews: Computational Statistics,* vol. 2, no. 4, pp. 497-502, 2010.

[27] W. K. Härdle, M. Müller, S. Sperlich and A. Werwatz, "Multivariate Kernel Density Estimation," in *Nonparametric and semiparametric models*, Springer Science & Business Media, 2012, pp. 73-74.

[28] "scipy.stats.entropy - SciPy v0.17.0 Reference Guide," 20 02 2016. [Online]. Available: http://docs.scipy.org/doc/scipy-0.17.0/reference/generated/scipy.stats.entropy.html. [Accessed 12 04 2016].

[29] Dannenberg and Mohan, "Characterizing Tempo Change in Musical Performances," in *Proceedings of the 2011 International Computer Music Conference*, San Francisco, 2011.

[30] "scipy.stats.pearsonr - SciPy v0.16.1 Reference Guide," 24 10 2015. [Online]. Available: http://docs.scipy.org/doc/scipy-0.16.0/reference/generated/scipy.stats.pearsonr.html. [Accessed 12 04 2015].

A. Appendix 1: Raw Data

Table A.1: List of songs used with their average ratings and URLs

Song	Mean Rating	Song URL
01	-1.20906004115	https://www.youtube.com/watch?v=is6m1P11GOo
02	0.540713146544	https://www.youtube.com/watch?v=KNV7SLksMRs
03	0.282561917948	https://www.youtube.com/watch?v=nQ1Z-kk9Sag
04	-1.37609415656	https://www.youtube.com/watch?v=8jlo-DVdEP0
05	0.118076965045	https://www.youtube.com/watch?v=U5n4gCeiz0k
06	-0.397310060726	https://www.youtube.com/watch?v=s5E3iGtyX18
07	0.552061535741	https://www.youtube.com/watch?v=MV-v9xqKhUE
08	-0.817280725133	https://www.youtube.com/watch?v=DEzWtijKXbc
09	0.10550775384	https://www.youtube.com/watch?v=NXmegM4528E
10	0.988164022952	https://www.youtube.com/watch?v=je0uwIhFsok
11	-0.865006934615	https://www.youtube.com/watch?v=gOpnEOnmg5Y
12	-0.727145514822	https://www.youtube.com/watch?v=8g7-Fk-ds2o
13	-0.549727146416	https://www.youtube.com/watch?v=78MlhPSxcvM
14	-1.13296091605	https://www.youtube.com/watch?v=sXLbBZmmT68
15	-1.33630620956	https://www.youtube.com/watch?v=FjeMDvCdrtc
16	(number not used)	
17	0.785223638852	https://www.youtube.com/watch?v=QZuWSrH9T9s
18	-0.13549787198	https://www.youtube.com/watch?v=H2x7elKbLl0
19	1.26949089908	https://www.youtube.com/watch?v=ALF8C1q5VrY
20	-0.4553024383	https://www.youtube.com/watch?v=Is0Dts2tT4Y
21	-0.162334161674	https://www.youtube.com/watch?v=-SJuHRMfF2w
22	0.744275186681	https://www.youtube.com/watch?v=F3wZUXpYQls
23	0.181570576065	https://www.youtube.com/watch?v=fE3mFOwUxdk
24	-0.300802391868	https://www.youtube.com/watch?v=6QUWkFeGQ0A
25	-0.464649038018	https://www.youtube.com/watch?v=8RFTB5vgV_4
26	0.753963122766	https://www.youtube.com/watch?v=XCMrXC8D05Q
27	0.46106774938	https://www.youtube.com/watch?v=-RuEDNYQQ40
28	0.522143251701	https://www.youtube.com/watch?v=r35Ius6JPS8
29	0.0898903225722	https://www.youtube.com/watch?v=wt0qhMEg-Xk
30	1.03199524634	https://www.youtube.com/watch?v=pm3WFJOzjVc
31	0.273614895995	https://www.youtube.com/watch?v=Yvz_LpQDr0k
32	-0.257767486821	https://www.youtube.com/watch?v=NJPAjiSX7Rk
33	0.54512634795	https://www.youtube.com/watch?v=FXDlwkuInBY
34	-0.0514455517722	https://www.youtube.com/watch?v=4atn3ue-nEM
35	1.26949089908	https://www.youtube.com/watch?v=4PekdeINQco

36	(number not used)	
37	(number not used)	
38	0.211627425481	https://www.youtube.com/watch?v=HYPh7UCzpHc
39	1.26949089908	https://www.youtube.com/watch?v=PvE88H8vb-4
40	-0.58227507332	https://www.youtube.com/watch?v=R9WTlP08LEg
41	0.16320814922	https://www.youtube.com/watch?v=c1huRuo6Wj0
42	-0.207865123	https://www.youtube.com/watch?v=jtN8oBjMr_E