Comparing Algorithms to Find the Period from the Beat Spectrum

Gregory Chan
Northwestern University
gregorychan2016@u.northwestern.edu

Suhong Jin
Northwestern University
suhongjin2016@u.northwestern.edu

Abstract
We describe our experiment, comparing two methods for deriving the repeating period from a beat spectrum, the simple method and the complex method. The methods for deriving the repeating period were originally developed by Zafar Rafii and implemented by Ethan Manilow and Prem Seetharaman in Bryan Pardo's lab at Northwestern University. The simple method derives the repeating period by computing the maximum of the beat spectrum and summing this with a minimum value. The complex method derives the period by computing the auto cosine similarity between the beat spectrum and itself, and deriving the result to compute the period. We evaluated the two methods by constructing our dataset and using the known period from this data to compare against the performance of both methods. Given the experiment we ran, more analysis is necessary to determine which method yields more accurate results.

1. Introduction

Rhythm is largely omnipresent in music produce by humans dating back for centuries, and its relationship to accent, metre, time, and tempo are useful measures for machine analysis. Thus, rhythm analysis is the basis for many music processing algorithms. The beat spectrum is one method for computing periodic self-similarity within an audio file as a function of time lag [1], which can be useful for estimating tempo and has applications in source separation [2]. REpeating Pattern Extraction Technique (REPET) is a source separation algorithm to separate a sound mixture that relies on the underlying assumption the signal is periodic. While there are many variants of the REPET algorithm, REPET original leverages beat spectrum as a method for deriving a repeating period. There are a number of variables that influence the accuracy of the repeating segment, but ensuring the repeating period is most representative of a repeating segment is crucial for the algorithm to separate foreground and background audio.

Two prominent methods for measuring similarity of an audio file against itself are the beat spectrum and the auto similarity matrix [3]. Given the importance of the repeating period in REPET original, we chose to evaluate the performance of two methods for computing the repeating period of a sound file. We hope by improving the accuracy of computing the repeating period from a beat spectrum may improve the results of original REPET. However, for the purposes of this experiment we limited the scope to analyze and compare the two methods.

2. Algorithms

The basis of this experiment relies on two methods for computing the repeating period from a beat spectrum. The two methods we will henceforth describe as the simple method and the complex method. Both methods rely on a precomputed beat spectrum to estimate the repeating period.

The simple method finds the highest peak within a provided minimum and maximum range, and using the time lag, we can derive the number of samples in the repeating period.

The complex method tries to find recurring patterns in the beat spectrum by trying to figure out how long it takes for the beat spectrum to come back to the same state. This is done by performing the auto cosine similarity on the beat spectrum. Using these similarities, the optimal period can be computed by deriving the result and finding the maximum. The time lag of this maximum value is then the optimal repeating period.

3. Methods

3.1 Packages used

We will use the python Northwestern University Source Separation Library (NUSSL), which includes helper functions for using REPET and provides python implementations of the simple and complex method for deriving the repeating period.

3.2 Dataset

In order to properly evaluate the performance of these period derivation algorithms we will need a dataset where we know the actual repeating period. Knowing the real period, we can compare this to compute the repeating period using both the simple and complex method. We will generate audio files where we can have control over the repeating period. To do this, we can take a segment of an original source file and repeat it a number of times. Thus, for each sound file, we construct many repeating files by varying the base segment length.

We collected a series of source files to use for our experiment. Each source file was created in such a way that there is no repetition within the file. Using these source files, we generated seed signals by dividing the source files by the set of consecutive integers ranging from one to ten. We stitched these seed signals together to create our repeating sound files, making sure to repeat the seed signal at least three times.

We generated a total of 2128 wave files from a list of 28 source files for comparison in our experiment. For each seed segment, a file of length 10, 20, 30 and 40 seconds was created.

3.3 Testing

The program we wrote to compare the simple method and the complex method, for each source file, constructed a seed signal, stored its length in samples, and computed both the simple and complex method's repeating period. The program logged these values into a database, storing the simple period, complex period, actual period, file name, sample rate, and length of the file for later analysis. We ran the tests on files with varying lengths and numbers of repetitions ranging from 10 seconds to 40 seconds long.

4. Results and Analysis

We ran the simple and complex methods on 532 repeating files, varying the lengths of the repeating files from 10 to 40 seconds in increments of 10 seconds. Running the simple method and the complex method on all of these variants resulted in a total of 2128 different files, with about 532 unique seed segments.

Analyzing the data proved to be a difficult task as both the complex and simple methods never found a period that matched the actual result. We chose to first naively compute the percentage error between the complex period and the actual period. We similarly computed the percentage error between the simple period and the actual period. In doing so, we found that the simple method had a lower standard and mean percentage error, with a mean percent error of -109.98% and a standard deviation of 317.006, while the complex method had a percent error of -252.55% and a standard deviation of 551.713.

Computing the mean and standard deviation for the percentage error was useful, but we also graphed the distribution in histograms. Figure 1 is a histogram of the percentage error for each of the 2128 repeating files using the simple method to find the period. There are quite a few outliers with percentage error as low as 3000% error. Figure 2 is similarly a histogram of the percentage error for each of the 2128 repeating files using the complex method to find the period. There are also quite a few outliers with percentage errors as high as 4000%. Visualizing the data in this way prompted us to remove outliers to get a better estimate of the standard deviation and mean for these distributions, removing outliers above 500% error. We found that removing outliers above 500% error still favored the simple method with a mean of -52.797 being the mean percentage error with a standard deviation of 113.338. The complex method had a mean of -61.283% error with a standard deviation of 146.566. The distributions for the simple method is shown Figure 5 and for the complex method in Figure 6.

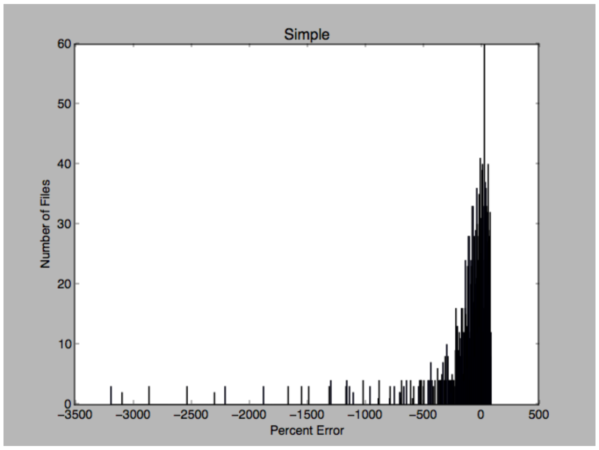

Figure 1: Percentage error distribution for the simple method across all 2128 repeating files.

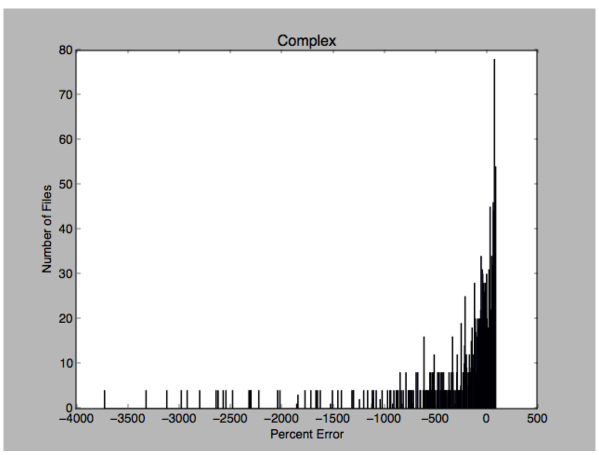

Figure 2: Percentage error distribution for the complex method over all 2128 repeating files.

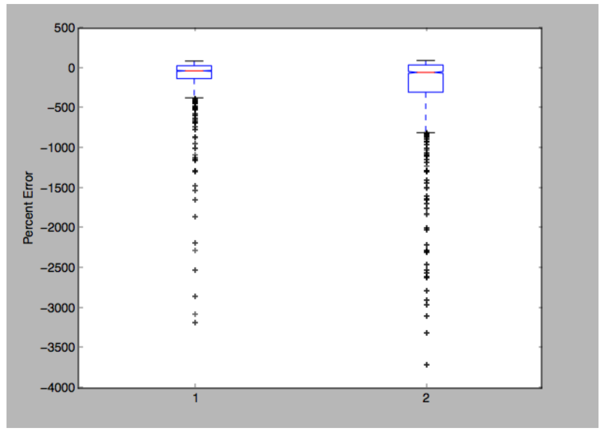

Figure 3: A side by side boxplot of the simple method distribution compared to the complex method distribution. The label 1 corresponds to the simple method and the label 2 corresponds to the complex method.

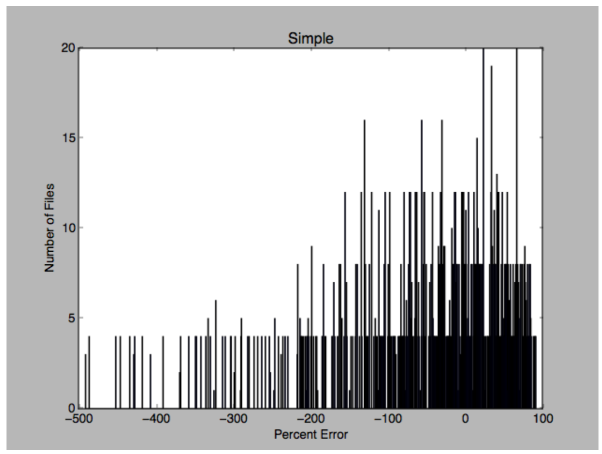

Figure 4: Histogram of the percentage error across all 2128 files using the simple method with outliers removed.

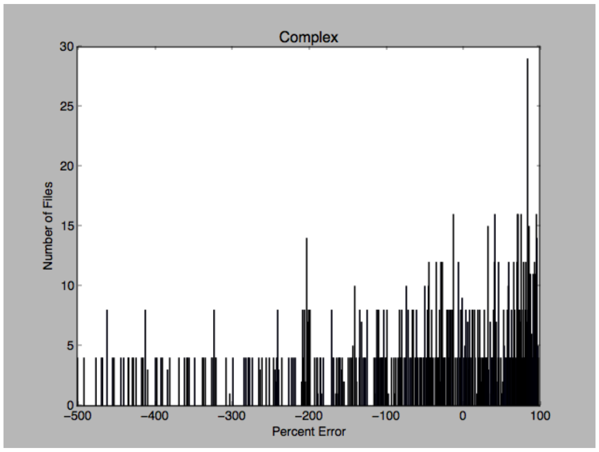

Figure 5: Histogram of the percentage error across all 2128 files using the complex method with outliers removed.

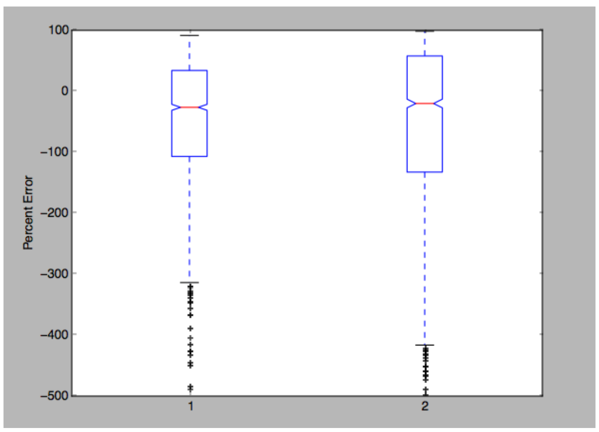

Figure 6: A side by side boxplot of the simple method distribution compared to the complex method distribution with outliers removed. The label 1 corresponds to the simple method and the label 2 corresponds to the complex method.

While the naive approach to analyzing this data shows that the simple method is better at finding the actual period, a percentage error that is an integer multiple of 100% error, plus or minus a few percent, may also be a period in the file. While a period that is an integer multiple of the actual period may not be the shortest period, it is a usable value. With this understanding of multiple integer multiples, we compared each file's percentage error to integer multiples of 100% error within a threshold. We varied the threshold from 0.5% to 10%, with the understanding that anything within 2.5% error is unlikely to be a good representation of the period. Figure 7 is a graph of the threshold on the x axis and the number of files with a percentage error within this threshold on the y axis. From this graph the complex method found more periods that were approximately integer multiples of the actual period.

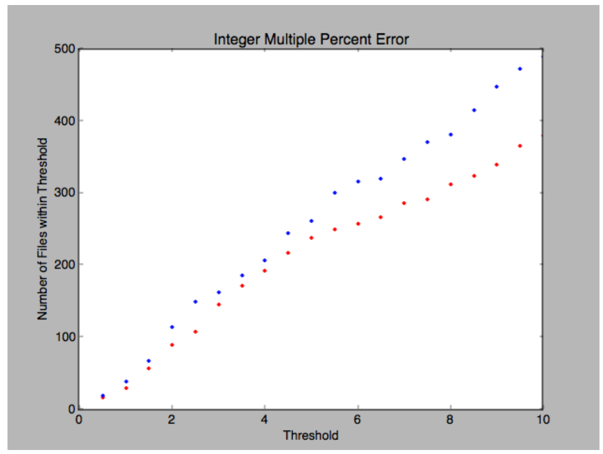

Figure 7: Classifying the found periods from the simple method and the complex method to the nearest integer multiple within a varied threshold from 0 to 10 percent.

While this less naive approach to analyzing the data shows the complex method is slightly better than the simple method for finding a period within a few percent of integer multiples of the actual period, it is difficult to conclude if it is in fact better.

5. Future Work and Conclusion

We see quite a few opportunities to continue exploring and analyzing the simple and complex methods for finding the period from a beat spectrum as well as comparison and analysis against other methods for finding the period from an audio source. More detailed analysis or data collection may be necessary for truly evaluating and comparing the simple and complex methods. Additionally, we can compare the results of these methods against other methods which use auto similarity matrices to find the period for an audio file.

While we found the simple method excelled in naively analyzing the data, we believe more analysis is necessary to determine which method is best. There is the possibility that it make come down to a case by case basis by which one method may perform better than another.

7. References

1. Foote, Jonathan, and Shingo Uchihashi. "The beat spectrum: A new approach to rhythm". IEEE, 2001.
2. Rafii, Zafar, and Bryan Pardo. "Repeating pattern extraction technique (REPET): A simple method for music/voice separation." Audio, Speech, and Language Processing, IEEE Transactions on 21.1 (2013): 73-84.
3. Jonathan Foote. Visualizing music and audio using self-similarity. In 7th ACM International Conference on Multimedia (Part 1), pages 77–80, Orlando, FL, USA, October 30-November 0 1999.