CS 559: GUITAR CHORD RECOGNITION USING MACHINE LEARNING

Esteban Schmitt
Stevens Institute of Technology
eschmitt@stevens.edu
Fall 2024

ABSTRACT

The transcription of music is a skill most musicians spend years to achieve, and some professionals never gain. In technology, transcription is also not easy. Sound waves and frequencies are noisy, and along with other instruments in a single song, they can be easily overlooked. By using Pitch Class Profiles (PCP), I will turn an audio signal into 12 distinct variables corresponding to a certain frequency that corresponds with a certain chord. This paper will focus on strategies using machine learning techniques to best find the model that can accurately predict a guitar chord from these PCPs. I will look at previous attempts to do the same thing and compare my results with theirs while also looking over what makes this task difficult.

1 Introduction

The guitar tech industry is a large growing field that has birthed new innovative technologies such as realistic virtual amplifiers and seamless digital effects. One problem for us guitarists remains and that is one of music transcription. Music transcription is the process of deriving the correct notes and chords for a given song simply by listening to it. The ability to transcribe any song given is a widely sought-after skill that professionals spend decades developing. Past efforts have been made to develop algorithms to accurately predict chords given the sound file; however, these algorithms and machine learning are either complex and attempt to do too much, or are old and don't consider new technologies that are in use today. The problem that is first encountered when attempting to interpret music in a digitized setting is that there are far too many inputs especially with the number and quality of songs being released in the modern world. To fix this, Takuya Fujishima developed the concept of chroma features (also known as PCPs) which can reduce an audio signal to the 12 fundamental notes in music which are as follows: C ,C# ,D, D#, E, F, F#, G, G#, A, A#, and B. To best understand the goals, findings, and methods of the experiment the paper will first focus on the definitions of certain musical terms, then move into the data and pre-processing, and finally the results and what we can understand from it.

1.1 Related Work

The first relevant work done in this area was by Takuya Fujishima [2]. In his article, he developed a method called Pitch Class Profiling (PCP) that uses the frequencies found in a Discrete Fourier Transform (DFT) to create a 12 vector object containing the relative frequencies of all notes in the chromatic scale (C, C#,...,B). Using these PCPs, Fujishima used a nearest neighbors algorithm to classify chords. The results were successful however in 2012 a similar experiment was done using the same PCP technique but the classifier was instead a neural network utilizing a classical gradient descent algorithm. The results from this was a 1% error rate on basic guitar chords. Moreover, in 2006 another way to calculate PCPs was created by Kyogu Lee. This method took into account that a single note doesn't just produce one frequency but it can also take other frequencies of the same note in another octave. This allowed for the creation of Enhanced PCPs (EPCPs) that greatly increased the quality of research in the area. Another method that has been used and has generated very good results is the use of a convolutional neural network on the spectrogram produced by the music and use it as an input [5]. I will not be using this method because, although it does produce better results, it is inherently more computationally heavy and my laptop is unable to run such calculations in a timely manner, thus I will pursue the method of implementing a neural network. Further research has been done to apply

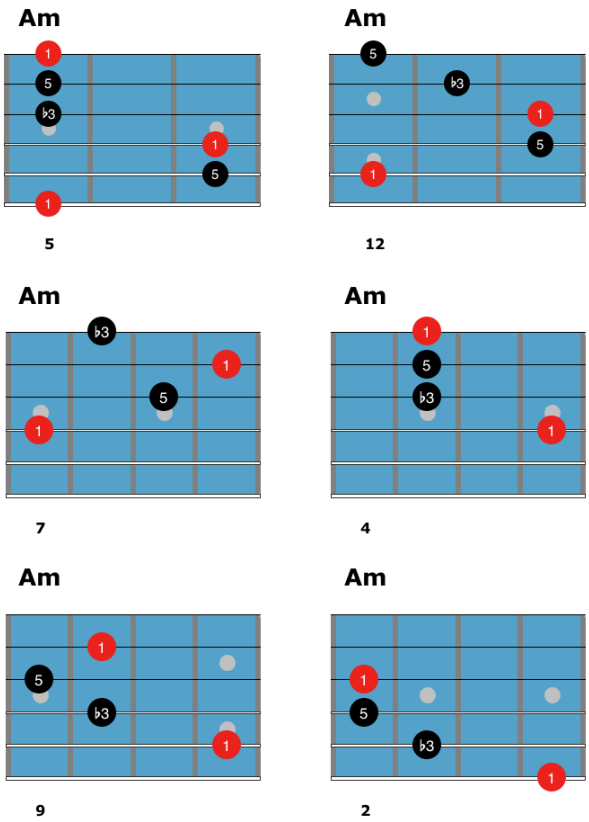

Figure 1: Different voicings of an Am chord on a guitar[8]

these methods into a recurrent neural network to analyze chords in a full song and detect chord and key changes.[4]. Problems with this approach that goes for all approaches come from a lack of number and quality of data as of course there is plenty of music out in the world already, however there must be someone who can first transcribe all of it and use it to train.

1.2 Musical Definitions

The foundation of musical analysis lies in basic music theory. Firstly, there are 12 semitones used in western music as shown previously (C,C#,...B). Each of these semitones vibrates at a distinct frequency with C1 starting at 32.7 kHz, and the distance to the next being the current multiplied by a factor of 1.059 [6]. As you can guess, the instance of a C1 implies the existence of others, and you would be correct. On a guitar with standard tuning, there are 4 C notes one could play. Actually, there are 4 of every semi tone. These notes are called octaves. An octave can be defined as being double or half the frequency of a certain note. For example, C1 is 32.7 kHz, so C2 would be 65.4 kHz. Now that the idea of notes and their frequencies is covered, we can move to the concept of chords.

A chord can be defined as a *set of simultaneous tones*[6]. So a number of different notes played at the same time is what defines a chord. For example, the A minor (Am) chord consists of the notes are A, C, and E. As previously stated, however, there are multiple A, C, and E notes on a guitar. What this does is causes there to be multiple *voicings* to play a certain guitar chord. A few of these Am guitar voicings are shown in Figure 1. The existence of these voicings poses a problem in musical analysis as it introduces the problem of having to account for not just the base frequencies of certain semi-tones, but also their octaves and harmonic features. In order to classify a chord, we must be able to input a signal and extract the amplitude of each frequency bin from C to B. To do this, we make use of pitch class profiling.

1.3 Chroma Features

The most common method that's been used to describe a guitar signal is the chroma feature, also known as a PCP. First introduced by Fujishima[2] and then refined by Lee[7], the math behind the pitch class profile is as follows.

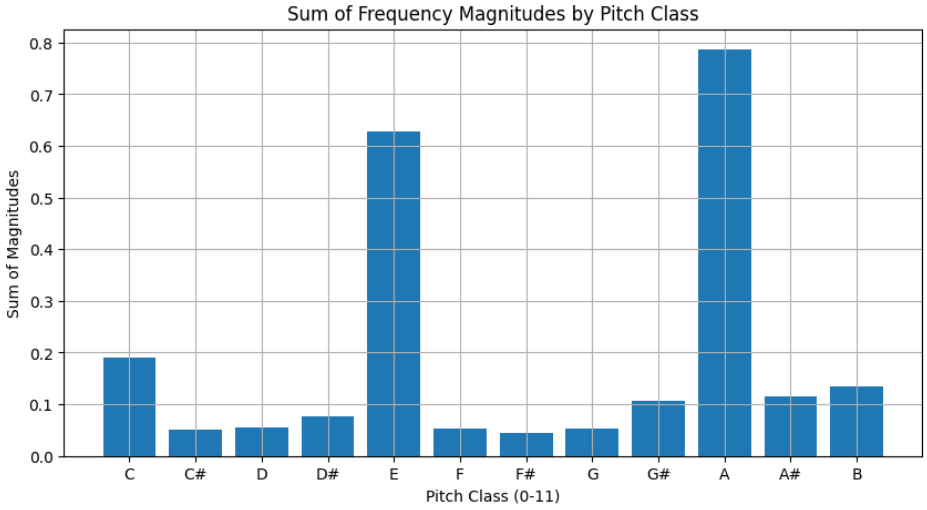

Figure 2: Pitch class profile for the Am chord (A,C,E)

$$PCP(p) = \sum_{l-where M(l)=p} ||X(l)||^2$$

where

$$M(l) = round[12log_2(\frac{f_s l}{N f_{ref}})]$$

when l is 0 M(1) is -1.

where f_s is the sample rate of a signal

p is the index from 0,11 of the bin (representing 12 semitone frequencies)

l is a frequency value coming from the DFT

N is the length of the signal

f_{ref} is the frequency corresponding to the first semitone.

With the EPCP, not every harmonic value is considered the same. For example, if a C2 note is creating C3 and C4 overtones, these are not weighed as equal as the "original" C2 note. The way the creation of a pitch class profile works is that we create the DFT from an audio signal, and once we do that we apply formula 1 with weighed harmonic factors (EPCP) and formula 2 being used as support. The results when graphed are shown in Figure 2. We must now figure out a way to generate these EPCPs from our data.

2 Methodology and Experiment

Our experiment will start will use the EPCP as defined by Lee[7]. I choose this version as it is objectively superior as it creates a more refined and noise-free output of a pitch class profile. We will apply the profiling to every piece of data and use it to create a csv dataset. With this dataset, we will train different machine learning models such as K-NN, decision trees, Support vector machine, Ada boost decision tree, and finally a neural network. Before beginning the experiment, intuition along with the help of Figure 3 tells us that K-NN may produce fine results as the distance of a certain frequency to a chords PCP, though a naive approach is what musicians naturally do when transcribing themselves. This method was also used by Fujishima[2] where he produced 94% accuracy. Moreover, looking at the chart, one can see an issue. Take for example the chords of A and A minor (Am). The only difference between the two chords is called the minor 3rd which in this case is C# to C. Although only Am is in our dataset not A, there are cases such as with Bdim and Bb where there is only 1 semitone difference and with noise and other factors, these could very

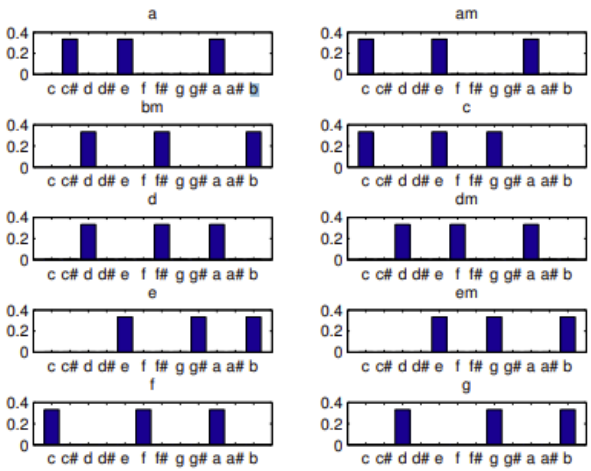

Figure 3: Chart of all the different PCPs for different chords[3]

easily be misclassified. This problem is the source of many solutions regarding chroma features as the goal in the field is to optimize the best way to highlight the KEY parts of a chord.

2.1 Data

The dataset I will be using is created by Fabiana Vinci and posted on Kaggle[1]. The dataset consists of 1440 training files, and 320 testing files. These files are split among 8 classes of chords: Am, Bb, Bdim, C, Dm, Em, F, and G. The files are a mix of digital and real guitars as well as noisy input in each of them. The mix according to the author is said to be 35 virtual guitars and 20 real guitars. This provides us with a good mix as in real world recording, electrical and virtual guitars are becoming much more common and it is important that we make sure to include them in our experiment. Unfortunately, the data is recorded at a 16 kHz sample rate, which will hurt our model as when it comes to certain electrical signals, especially noisy ones, the quality is necessary to tell certain frequencies. This setback, however, is realistic as in the real world data is rarely ever perfect.

To process the data, we will go through each sound file and first normalize the data between -1 and 1. We will then extract the EPCP and store the value of each semitone bin as a feature in data frame. This will result in 12 features for every sound file.

2.2 Experiments on basic models

The models that I used are a decision tree, KNN with k=3, SVM linear, Logistic Regression, and Adaboost decision tree with base being a depth 1 decision tree. I use a train test valid split of 0.75, 0.15, and 0.10 respectively.

The results as can be seen in Figure 4 were produced from these models. From these models SVM and K-NN appeared to both perform the best with SVM performing slightly better. This reaffirms my previous intuition with K-NN being an already effective method for predicting guitar chords as it is the inherently human way to do it. Overall, all of the models performed very well exceeding previous methods used that relied on older technologies.

2.3 Neural Network Model

In this experiment, I implement a neural network with one hidden layer and the following hyperparameters

- 15 neurons
- L2 regularization of 0.01
- ReLU activation function
- Softmax final activation function
- Adam optimizer with learning rate 0.001

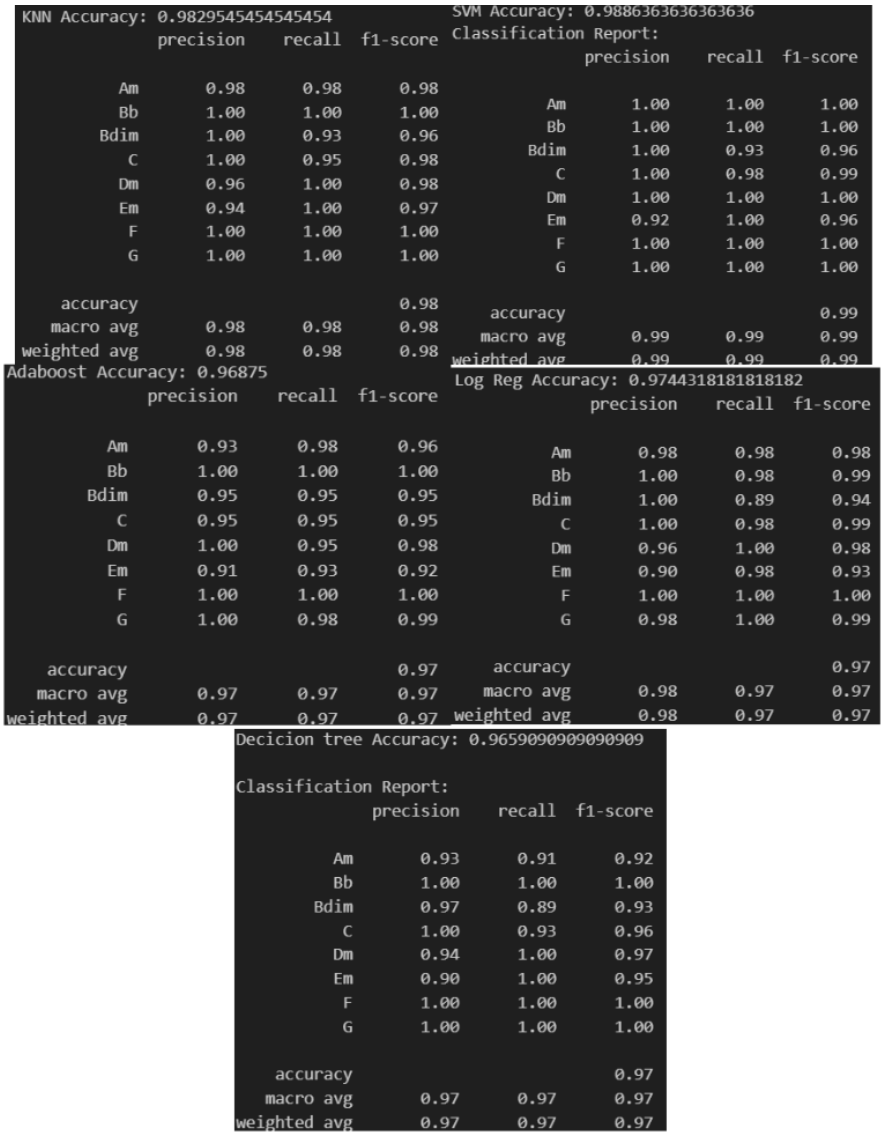

Figure 4: The results for each basic model

NN Accuracy: 0.9943181872367859

	precision	recall	f1-score
0	0.98	1.00	0.99
1	1.00	1.00	1.00
2	1.00	1.00	1.00
3	1.00	0.98	0.99
4	1.00	0.98	0.99
5	1.00	1.00	1.00
6	0.98	1.00	0.99
7	1.00	1.00	1.00
accuracy			0.99
macro avg	0.99	0.99	0.99
weighted avg	0.99	0.99	0.99

Figure 5: Neural Network Results

The model utilizes early stopping with a patience of 20 epochs. We fit the model over 500 epochs (if early stopping doesn't end it early) and a batch size of 32. The results are as follows: The neural network performed with a 99% accuracy, exceeding all of the basic algorithms. The high accuracy poses some fear of over fitting as there is a decent amount of noisy data, however upon creating a sample program that tests on chords played from other online datasets, it correctly categorizes them. Thus it is clear that the neural network, although more computationally heavy does perform significantly better than other more basic models.

3 Conclusions

This paper re-validates the results other papers have found with the neural network as well as other models being able to produce consistent results well above 95% accuracy. The neural network, however, being the superior of them all. Unfortunately, due to a lack of quality data, this experimental setup only works with 8 chords so in the future it is strongly recommended to others to either create or add to a dataset for new chords. In the future, I would seek research into recurrent neural networks in order to detect chords in a temporal space such as a song. This has already been done by Harte[4] on his Beatles dataset, however expanding it could lead to great possibilities in the music verse. This experiment also shows that low-quality data does not need to be a limiting factor when creating a learning model although it is very nice to have.

4 Acknowledgments

I would like to thank my mother and father for pushing my passion for music and helping me choose a topic for my research. I would also like to thank Matthew Feiler for teaching me the fundamentals of what I know.

References

[1] F. Vinci, "GUITAR CHORDS V3," *Kaggle.com*, 2022. https://www.kaggle.com/datasets/fabianavinci/guitar-chords-v3/data (accessed Dec. 19, 2024).

[2]Takuya Fujishima, "Realtime Chord Recognition of Musical Sound : a System Using Common Lisp Music," International Computer Music Conference, vol. 1999, pp. 464–467, Jan. 1999.

[3]J. Osmalsky, J.-J. Embrechts, V. Droogenbroeck, and S. Pierard, "Neural networks for musical chords recognition," Journées d'informatique musicale, 2024, doi: https://hdl.handle.net/2268/115963.

[4]C. Harte, "Towards automatic extraction of harmony information from music signals," Jan. 2010.

[5]E. J. Humphrey and J. P. Bello, "Rethinking Automatic Chord Recognition with Convolutional Neural Networks," *2012 11th International Conference on Machine Learning and Applications*, Dec. 2012, doi: https://doi.org/10.1109/icmla.2012.220.

[6]dsa2gamba and abbottds, "Musical intervals and temperament," *pressbooks.pub*, 1998, Available: https://pressbooks.pub/sound/chapter/pitch-perception-and-logarithms/

[7]K. Lee. Automatic chord recognition using enhanced pitch class profile. In Proceedings of the International Computer Music Conference (ICMC), pages 306–313, 2006.

[8]"Minor Chords For Jazz Guitar," *www.jazzguitar.be*, Jan. 31, 2019. https://www.jazzguitar.be/blog/minor-chords/