CHALMERS UNIVERSITY
OF TECHNOLOGY

Music Engineering
TRA21/22_MUSIC_3

Sentiment Analysis for Chord Progressions
Project report

Jesper Holsten - jesperho@student.chalmers.se
André Muricy Santos - muricy@student.chalmers.se
Jorge Mifsut Benet - mifsut@student.chalmers.se
Jules Sintes - sintes@student.chalmers.se

May 16, 2023

1 Introduction

Sentiment analysis is a common AI technique for extracting emotional content out of sequences of words. It's used extensively in many applications, from spam filters, to social media networks, to keyboard suggestions. This effect is achieved by utilizing Recurrent Neural Networks - an architecture of neural networks that is capable of interpreting temporal data through layers whose outputs that feed back into themselves - to analyze complex data that is rich in semantics, usually words.

The aim of this study, however, was to design a network that is capable of interpreting the emotional valence of chord progressions, which are objects that, in principle, can be represented in much lower dimensional spaces. Due to this simplification, and the fact that each time series will have a fixed number of chords (four), a regular densely connected deep neural network with no recurrence can be used in place of a RNN.

This project falls within the field of *Music Information Retrieval* (MIR) which can be defined as the interdisciplinary science of retrieving information from music. It is a growing field of research with different real-world applications, such as music classification for recommendation systems (Spotify for example), music generation, transcription and more. This project has background in psycho-acoustic linked to musicology and artificial intelligence as it aims to predict the emotion aroused by sequences of chords using data-science and deep learning techniques.

1.1 Preface: Cultural bias in music perception

Music is sometimes portrayed as a universal language or as a bridge between cultures, as it seems to be a ubiquitous occurrence around the world. On the other hand, the vast diversity in all aspects of music (from timbre to rhythm structure) across human societies might lead one to believe that, much like language, the meaning found in a specific piece of music is culture-specific. That is, the association of certain elements in music to particular emotions would be just a cultural convention, rather than a cross-cultural universal.

Testing the universality of emotion perception in music is not an easy task from the point of view of Western music, to which the theoretical framework used in this study belongs. This is because in today's globalized world, even most non-Western peoples have encountered Western music occasionally throughout their lives, sometimes in mass media such as movies, where explicit emotions are already linked to a specific piece, making these associations internalized.

Nevertheless, Fritz et al.[2] conducted a study with a culturally isolated ethnic group in Cameroon, the Mafa, that were initially naive to Western music, testing their reactions to some pieces alongside a group of Westerners. The researchers found that the Mafa recognized the same three fundamental emotions (happy, sad and scared) in each piece as the other group, albeit with more variability in their responses, with 2 out of 21 Mafa performing at chance level. It has been shown in some studies [3] that performance in music perception and cognition is better for participants that are culturally familiar with the music, that would explain the more consistent result for the Western group. That being noted, the Mafa still recognized the basic emotions intended. Fritz et al. [2] found correlations between the emotion of happiness and high tempo while low tempo was perceived as scared. These two emotions were also correlated with major and minor modes respectively. They also propose that prosody plays a part in the universal capability

to recognize emotions in music, as a parallel to non-verbal communication's patterns of expressiveness. Unfortunately, our choice of method with chords progressions of a (short) limited length does not allow for prosody to be a parameter to focus on.

In contrast to the aforementioned study, Balkwill et al.[1] conducted a study where it was Westerners that were exposed to an unfamiliar set of Hindustani musical pieces (*ragas*) each performed with the intention to transmit one of a list of basic moods (*rasas*). Similarly to the other study, the people native to the culture of the piece performed slightly more consistently, but the group that was not familiar with the tonal system still were sensitive to the basic emotions intended to transmit.

All things considered, even though the generation of the data in our study has been performed mostly by people with a Western cultural background or even educated in Western music tradition, according to the literature it can be argued that there will be some universality to the results.

1.2 Challenges and data labeling rationale

As is the case many in many machine learning solutions, a big challenge for this project is to have enough labeled data. While some assumptions can be made about the labels given certain music-theoretic ideas, the goal is to verify that it is possible for a network to learn how to represent these ideas just from examples. In other words, we don't want the network to overfit to a dataset that just happens to represent everything about music theory - instead, we want the network to represent and generalize patterns that are *explained* by music theory.

So how do we obtain this data? Our solution was to write software to help label several chord sequences and label it by hand; however, this approach would be hard pressed to work for a densely connected neural network without overfitting. Indeed, overfitting proved to be a problem when trying to get the network to even vaguely generalize, and a proposal for a new neural network architecture, which unfortunately could not be implemented in the scope of this project due to time constraints, is made at the end.

Another challenge was how to represent the emotional content of chords. There is not a straightforward mapping from such a human, subjective experience to cold labels on a dataset. We use Russell's circumplex model of affect [4] as a basis for labeling the data; since the goal is to translate human perception as naturally as possible, the reduced dimensionality and clear meaning of each axis seemed like a good fit. We join to this the notion of consonance to a total of 3 dimensions along which to classify a chord progression.

2 Methods & Results

2.1 Basic Music Theory Prerequisites

There is a wide range of theoretical principles in music. This section deals with a few music theory concepts that is used throughout in the project.

Notes, Scales and Chords

Western music uses 12 notes in different octave bands which are organized frequencies.

The distance between two notes is called an interval. Combinations of these notes make up scales. There are all sorts of scales used in music, but the most common scales originated from the early 17th century from the western church music. These scales are called modal scales and the most common mode scale is the ionian scale which is commonly referred to as the major scale. Each scale has a common ground note called a root note. The root note is defining what key a song is played in. The interval between two notes placed next to each other is called a "small second" or a "half step".

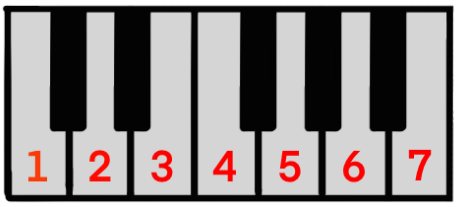

Figure 1: The 7 notes in the major scale, starting from a C (C D E F G A B).

The major scale consist of 7 steps with combinations of whole and half steps like shown in figure 1. The first three notes in the scale is separated by one note (whole steps), the third and fourth notes is next to each other and the following notes in the scales are whole steps. After the seventh note, the scale repeats itself in a new octave band. The first and seventh notes is next to each other (half step). The major scale is used in all music genres, but it reappears again and again in the popular music. Not only the scale, but also the chords associated with the major scale.

A chord is a collection of notes played simultaneously. Chords usually have three or more notes. The chords in a major scale uses the notes in the major scale as a foundation. Starting from the root note C, a C major would consist of the notes C, E and G. Each triad (chords with three notes) in the C major scale would be obtained from incrementing each note in the C major chord on the white keys on the piano.

Consonance and Dissonance

It is not necessarily true that a set of chords in the same key sounds good together. To maintain a nice balance between chords in a progression, the chords should also somehow resolve each other. The terms consonance and dissonance are opposites and describes this correspondence between a set of chords or notes. When a chord progression is consonant, people would often say that the chords fit well together. However a chord can be dissonant and resolve nicely into a consonant chord. In this project we use the terms consonance and dissonance to describe how well the chords in a chord progression (a set of chords in a sequence) fit together. Dissonance would mean that the chords don't fit well at all relation to each other.

2.2 Random Chord Generator

A random Chord generator is a system that produces random chords. Just to be precise, there is in fact no such thing as random values in computer science. The random values has to be produced by some algorithm. If the algorithm that produces the random values is known, the random values would also be known i.e. they are no longer random. Therefore pseudo-random would be the correct way to describe the chords in this system. Pseudo-random values will from now on be referred to as random values because it does not really matter for the purpose of this project. The random chord generator should be able to produce random chords in the same arbitrary key.

Chord Convention

The AI Sentiment analysis tool is based on the midi (.mid) format. Midi notes are represented by integer numbers e.g. the middle note C would be equivalent with the number `60` in the midi format. A C major would then be written as an array `[60, 64, 67]`. The problem with using such absolute values to represent chords arises when the key is changed. The chords should have a more generic representation. That's why instead of using absolute midi values, the intervals between the notes is used instead. This would hold for any arbitrary key. A major chord would then be represented as `[3,2]`. By specifying the root note of the chord, this chord can be represented in any key. In this simple example it would be very easy to just add or subtract values from the absolute value array to obtain the chord in a different key, but in general it is a good practice to write code as reusable and generic as possible.

When the chord representation is established, a dictionary of chords can be defined as shown in the code snippet below. Here the root notes in the chords are doubled one octave above.

```
"chords" : {"major" : [3,2,4],
            "minor" : [2,3,4],
            "augmented" : [3,3,3],
            "diminished" : [2,2,2],
            "suspended2" : [1,4,4],
            "suspended4" : [4,1,4],
            "major7" : [3,2,3],
            "minor7" : [2,3,2],
            "dominant7" : [3,2,2],
            "major7sus2" : [1,4,3],
            "m7b5" : [2,2,3]}}
```

To get random chords in the same key, one can't simply pick random chords from this chord bank. The legal chords with the root note at a step in the major scale has do be defined. For instance if the key was C major, the C minor chord would not be satisfactory for the 1st step in the scale because the second note would have ended up at a D sharp (or E flat). From figure 1 this would correspond to the second black key from the left on the keyboard. To solve this, a new dictionary is defined, containing the legal chords in each chord step as shown below. More chords are included in each step in the actual chord generator, but this is the basic principle.

```
legal_chords_in_arbit_key = {  1 : [chords["major"],
                               2 : [chords["minor"],
```

```
    3 : [chords["minor"],
    4 : [chords["major"],
    5 : [chords["major"],
    6 : [chords["minor"],
    7 : [chords["m7b5"]]}
```

To get a random chord sequence, the key is first selected. After the key is set, a value from 1-7 is selected randomly. This is the step in the scale of the given key. Lastly a random chord is selected from the legally defined chords within that step. This is repeated for n number of chords. The random chord generator written in a `class` environment `RandomChordSequence` which is an object with a set of variables and functions that is easy to use. The following line of code would yield 10 random chord sequences with 4 chords in each sequence.

```
chord_sequences = RandomChordSequence(nchords=4,
                                      nsequences=10,
                                      chord_type=legal_chords_in_arbit_key)
```

2.3 Labeling tool

Since this kind of problem is highly dependant on the data quantity in order to reduce bias and achieve better results, it was decided to design a user friendly labelling tool with a graphical user interface to make the task easier. The light weight of the software is really convenient when it comes to scaling up the data collection, which would be needed as the results discussions of the methods will show. In this case, sharing the labelling tool and inviting more people to label data would most likely make a dent in the issue of the our low data volume. The labelling tool was programmed using Python and the Tkinter built in library. It uses all the functions of the chord generator to generate a batch of chords progression, pre-process and playback the sequences and takes the user rating as input. When closing the application, the results of the labelling session is exported as a .csv file.

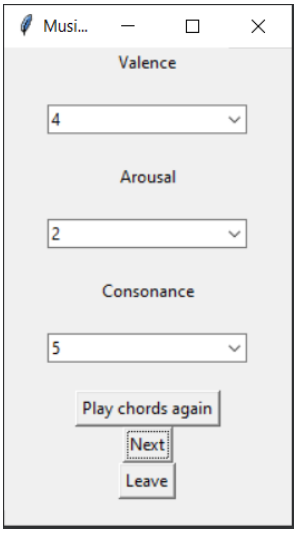

Figure 2: Screen capture of the labelling tool graphical interface

2.4 Further insights from labeled data

The data collection itself was interesting and the data analysis confirmed that there was correlations between the labelling dimensions as well as real relationship between the chord progression and the average rating of consonance, valence and arousal.

The chord itself, but also its position as well as its relation with the other chords in the sequence have a large influence on what a specific chord progression is likely to make a listener feel. Based on histograms like 4, it is already possible to create very nice chord progression that will be very likely to be rated as consonant. In spite of beeing less obvious, the same goes with the 2 other dimensions.

The heatmap 3 shows the correlation between the 2D Valence-Arousal space and the consonance of a given chord progression : A dissonant chord progression is more likely to lead to low valence which means either something *annoying* (high arousal) or *boring / sad*

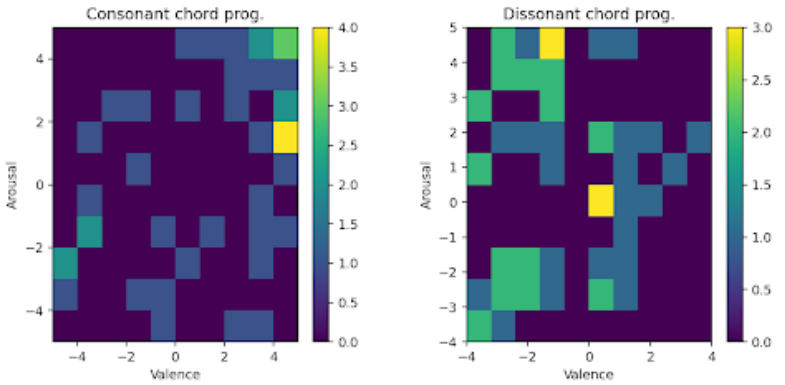

Figure 3: 2D histogram of consonant and dissonant chord progressions

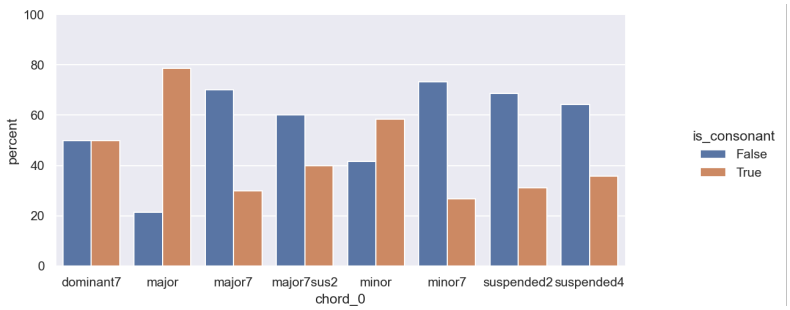

Figure 4: Histogram showing type of the first chord in a progression along with its consonance rating

(low arousal), whereas a consonant chord progression si more likely to lead to *pleasant* or *happy* feelings in the Valence-Arousal space.

Automating, and processing these prediction by using artificial intelligence is a necessity to handle the global complexity of such a problem.

2.5 AI applications

2.5.1 Deep neural network

With the data generated, a neural network is now ready to be trained. As mentioned briefly in the introduction, the idea was to go with a regular multi-layer perceptron; since all the chord progressions generated have a fixed size, this is enough to represent the input. Also, the labels are just three dimensional vectors and classifying an input maps exactly to the human experience of relaying what the emotion of a song is: if a person calls a progression "happy" and "energetic", we should expect a neural network to output high valence and high arousal values in the corresponding output nodes. The only manipulation done on the data is to convert the values in the data, which is stored as MIDI, as values from -1 to 1. This is necessary due to the fact that the numerical values in the MIDI representations

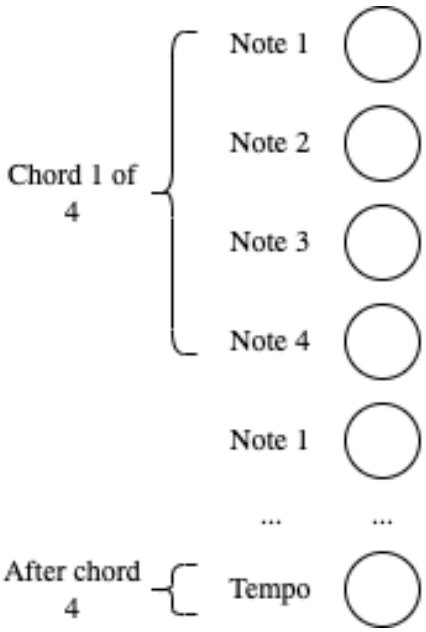

Figure 5: How the input to the network is represented

of the progressions have different ranges (tempo is not in the same scale as the notes, for instance). This is done over the entire dataset, then an 80/20 split is done for training and validation for early stopping, and training with MSE is done with default PyTorch parameters on the Adam optimizer.

Due to the low amount of data, in the order of hundreds, a deep neural network starts overfitting very rapidly. There are solutions to dealing with low amounts of labeled data, like using dropout layers, early stopping and data augmentation. Data augmentation was never an option for this project since any difference in the data could correspond to totally different labels. Early stopping was used; but for any split of the training and validation set, overfitting starts to happen too quickly for the network to be able to generalize. Dropout layers could help, but the problem was deep enough that we decided to move on to an even simpler architecture and to think about a more natural neural network architecture for this task in the future.

2.5.2 Random forest

Since we were struggling with the deep learning method with neural network, we considered different options to reduce the complexity of the problem. Firstly, among the 3 dimensions considered, consonance might be the one with the least subjective bias since it has somehow some western musical theory behind it. Secondly, the range of output can be changed to a binary problem. Then such a problem can be solved using standard machine learning algorithms. It was then decided to transform the problem into binary classification of consonance dimension and use Random Forest algorithm which is a reasonable choice for such a problem. The first results were not really good with only 0.65 test accuracy (where 0.5 means that the classification is made randomly). Thus, the dataset size was increased by using 144 standard chords progressions widely used in music production. We assumed that these chord progression were all labeled as consonant. This drastically increased the size of the dataset and the performance of the Random Forest classifier. Using this new dataset, Random Forest achieves an average f1-score of 0.81 which is really encouraging for this kind of problem.

3 Discussion

The random forest results are encouraging; they show that at least the notion of consonance is easy to isolate in a 4 chord progression. It is expected, however, that the two other dimensions are much more overloaded with subjective meaning and thus harder to predict.

The incapacity of the neural network to learn the complex function mapping 17 inputs (4 four-note chords + one tempo value) to 3 outputs (the three dimensions of valence, arousal and consonance) is probably largely explained by insufficient data. There is at least one method that could address small amount of data in a machine learning application, namely semi-supervised learning.

3.1 Alternative deep learning approach - semi-supervised learning with Graph Neural Networks

Graph Neural Networks [5] have appeared in recent years as a way of translating the idea of convolutions in images to convolutions in graphs; information travels through graphs in such a way that it becomes possible to perform many machine-assisted tasks on them, like predicting and classifying graphs and nodes. If a graph is represented as an adjacency matrix and each node has a fixed number of features, it is possible to convolve on it and learn many properties of the graph's structure.

In our case, we could represent the three-dimensional labels of a chord progression precisely as its feature vector. The thing we *want to predict* is then exactly this feature vector; a three-dimensional class. This matches well with our problem of not having enough labeled data; generating tons of chord progressions is not hard for a computer, but it is hard for humans to listen to all of them in order to do labeling. The last ingredient is how to represent connections between chord progressions; this is indeed a challenge and not obvious at all and would probably massively influence what kinds of patterns the network can pick up on.

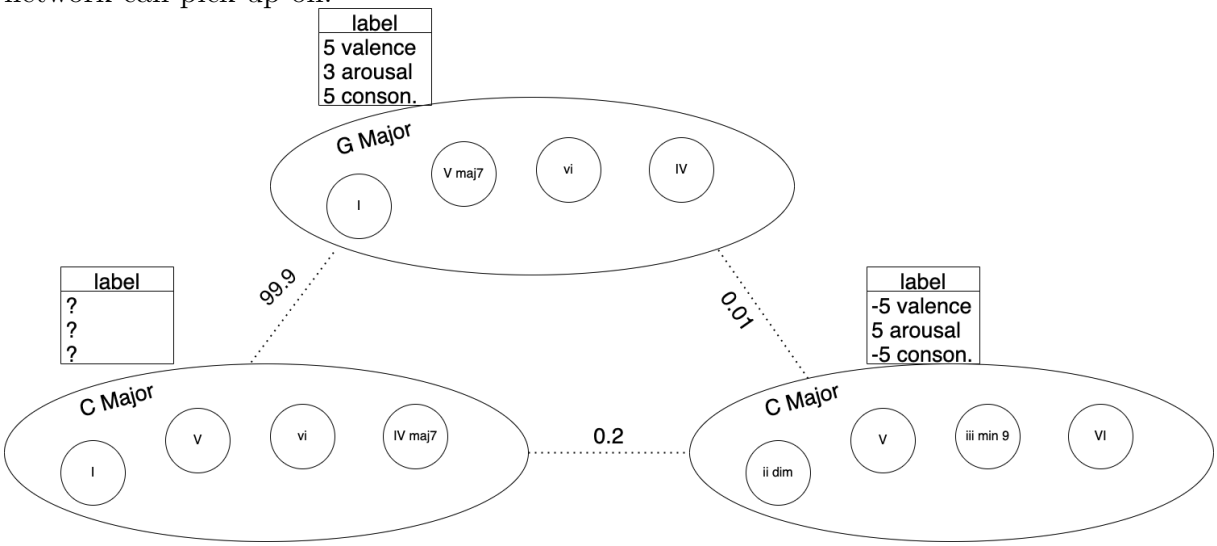

Figure 6: Imagined zoom-in on the relationship between 3 chord progressions

Figure 6 shows the proposed setup for the graph structure. The chord in the lower left is close to the top chord, but in a different key and with different maj7 chords; it is reasonable to expect however that these would sound similar to a human. The chord on the right is totally different from the top one, but has some similarity to the one on the left: both are in C major and have the same chord in the second position. The distance values in the figure are arbitrary and only meant to convey the idea here; that some notion of distance between progressions can be derived in a relatively systematic way, which means it can be programmatically inserted into a graph. It is then up to the GNN to use the graph structure to figure out what the label of unlabeled progressions should be; based only on their relative position in the network. GNNs are effective for learning on partially labeled data as long as it has strong structure.

4 Contributions

All members feel their work was of equal importance.

- André worked on the AI code and graph neural network proposed architecture.
- Jorge worked on the AI code and researched the cultural background section.
- Jules worked on the labeling part of the data generation tool, did the data analysis on the labeled data and the random forest classifier.
- Jesper worked on the chord generation part of the data generation, hosted the meetings with Affe and provided much of the music theoretic insight.
- All of us together worked on labeling the data, writing and editing the report.

5 Conclusion

This project was a good amount of fun and involved a lot of learning for all of us. Not only about music, the psychology surrounding it, its traditions and different cultural interpretations, AI and more; it was a great opportunity to organize and schedule work as a group, a fundamental skill for maximizing the impact one can have.

We want to thank again the course organizers and Chalmers for giving us the opportunity to do such exciting and creative work.

André, Jesper, Jorge, Jules

References

[1] Laura-Lee Balkwill and William Forde Thompson. A cross-cultural investigation of the perception of emotion in music: Psychophysical and cultural cues. *Music Perception*, 17(1):43–64, 1999.

[2] Thomas Fritz, Sebastian Jentschke, Nathalie Gosselin, Daniela Sammler, Isabelle Peretz, Robert Turner, Angela D. Friederici, and Stefan Koelsch. Universal recognition of three basic emotions in music. *Current Biology*, 19(7):573–576, April 2009.

[3] Steven J. Morrison and Steven M. Demorest. Cultural constraints on music perception and cognition. In *Progress in Brain Research*, pages 67–77. Elsevier, 2009.

[4] James A Russell. A circumplex model of affect. *J. Pers. Soc. Psychol.*, 39(6):1161–1178, 1980.

[5] Benjamin Sanchez-Lengeling, Emily Reif, Adam Pearce, and Alexander B. Wiltschko. A gentle introduction to graph neural networks. *Distill*, 2021. https://distill.pub/2021/gnn-intro.