Automatic Harmonic Analysis of Melodies

Finlay McAfee

MInf Project (Part 2) Report
Master of Informatics
School of Informatics
University of Edinburgh

2017

Abstract

Acknowledgements

I would like to thank Mark Steedman and Andrew McLeod for their advice and guidance on this project.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Finlay McAfee*)

Table of Contents

Chapter 1

Introduction

This is a report to detail the work carried out in the second year of the MInf Project.

1.1 Summary of contributions

1.2 Previous Year

Chapter 2

Theoretical Background

This chapter goes through the main concepts necessary to understand the design of the systems built in this project. Each section is intended to provide a brief summary of the necessary information on each topic to allow the reader to fully understand the workings of the models in the next chapter, as well as an understanding of the design choices.

2.1 Music Theory

It is safely assumed that the reader is familiar with the concept of music. If not then it is suggested that they study the following albums before returning to read this report: Blood On The Tracks by Bob Dylan [3] and Songs From a Room by Leonard Cohen [2].

Music is fundamentally concerned with two concepts: rhythm and pitch. Rhythm is, at its simplest, a regularly repeating pattern. The periods of repetition that musical rhythm is most concerned with lie around the same frequency as that of the human gait, or heart beat (the reader is left to draw their own conclusions about the significance of this). Pitch is, essentially, also concerned with regularly repeating patterns, only now the domain is hundreds of beats per second. The human ear experiences this as a musical note, rising and falling in pitch as the frequencies increase and decrease.

Music has been studied and practised by many different cultures over the course of human history and there have been many different formalisms for describing it developed. The most common, and that used in this report is *Western Tonal Music* (WTM). Under this paradigm, the continuous range of pitch is divided into repeating *octaves*, where corresponding points on each *octave* are perceived as the same note at different pitches. These ranges are then subdivided into 12 distinct notes. The method used for these divisions is based on ratios between note frequencies, the most commonly used being Equal Temperament [5]. Pieces of music are organised into one of 12 *keys*, each with a *tonic* corresponding to one of the 12 notes. Keys can be thought of as a prior over the distribution of notes in a song, where the tonic note has the highest probability of being played.

In written music, pitch is associated with vertical direction on a *stave*, shown below.

DIAGRAM.

Rhythm is concerned with the temporal positioning of these notes, both where and when they occur, as well as for how long to play them. In WTM a piece of music has an associated beat or *tempo* that defines the frequency of an underlying, unheard pulse, over which the notes of the piece are laid, typically close to 120 beats per minute. The *time signature* of a piece defines how to treat groups of these beats, with the most common being $\frac{4}{4}$. This corresponds to groupings of four beats (also called *quarter notes* or *crotchets*). These groupings are referred to as *bars* and are used to organise the temporal structure of a melody in a way that is both easy to read and represents the underlying rhythmic properties of the piece. As in western languages, time flows from left to right in written music.

DIAGRAM

The above is a very brief overview of melodic structure in a piece of music. However there is another concept that is fundamental to this problem space, and that is *harmony*. Fundamentally, *harmony* is the relationship between the pitches of two notes, as they are heard together. It is concerned with the *intervals* between the notes and how they are perceived. The most basic interval is the *semitone*, the distance between one note and its neighbour, hence there are 12 semitones in an octave. The harmonic perception of an interval of two or less semitones is dissonant and jarring. The octave itself is an interval, which is perceived as resonant and uncharacterised, another way of saying that two pitches an octave apart are perceived as the same note. The next simplest interval is when two notes are 5 semitones apart, the *perfect fifth*, referred to as a *power chord* in guitar terminology, it adds the most basic harmonic colour. The intervals of 3 and 4 semitones are where harmony takes on an emotional quality, 3 semitones is referred to as a *minor third*, and evokes a feeling of darkness and melancholy, when perceived; whereas 4 semitones, the *major third* evokes the opposite emotion, that of brightness. It is this incredible connection between emotions and intervals that makes musical harmony so fascinating.

INTERVAL DIAGRAM

A *chord* is when more than one note is played at one point in time, typically at least three. The basic three-note chord is called a *triad*, composed of notes that are based on intervals, relative to the note in the first position, the *root*. The other two are the *perfect fifth* and a *third*, either *major* or *minor*. Depending on the *third* that is chosen, the full chord with take on one of these qualities. The following diagram shows the basic C major chord.

DIAGRAM OF C CHORD

A modern musical arrangement, for example a jazz standard, typically consists of both a melody and an associated progression of chords, intended to be played as an accompaniment to the melody. The problem space of this project is concerned with the association between these two, specifically the problem of generating one if the other is not present. Generating a melody is what is typically considered musical composition, a difficult problem to solve on which much work has been done in recent years CITE ALL THE THINGS. This project concerns itself with the other direction, predicting chord sequences from melodies. This problem is conceptually half sequential classification, half generation, as there is not necessarily a unique chord sequence for every melody, but there is still a ground truth that can, theoretically, be derived from the observed notes. The following piece of music is an example from one of the corpora used for training the models.

DIAGRAM MINUET IN G

In terms of a traditional classification problem, $f(x) = y$, the x here is a sequence of notes and the y is a sequence of chord labels. Hence we have a sequence to sequence problem.

A small, simplifying assumption that must be made when addressing the problem in this way is for the case of polyphonic music, where more than one note is played at a time in the melody. In this case we can simply treat each combination of notes as a sequence of notes played in quick succession, with order being determined arbitrarily.

2.2 Natural Language Processing

An often-drawn comparison is that of the similarity between music and language. Many postulate that music is itself a form of language [1] (although perhaps it is more enlightening to say that language is a music). This is useful when analysing music as there is a wealth of literature on natural language processing.

The immediate comparison that can be drawn is to Part of Speech (POS) Tagging. In this problem we are trying to assign grammatical tags to a tokenized sequence of words. This one-to-one mapping problem is analogous to chord labelling in the case where each note is labelled with a corresponding chord:

DIAGRAM

Both are examples of a problem where the visible variables that are observed are dependant on the latent variables that are trying to be determined. This problem would be simple if said latent variables were not dependant on each other:

PLATE DIAGRAM

But this is clearly not the case. An adjective is very likely to be followed by a noun, and a G major is very likely to be followed by a C major. In reality the situation looks more like this:

NON-PLATE DIAGRAM - LATENT VARS CONNECTED

The complex relationships between the latent variables is one area where the difficulty arises in these problems. An often used approach to this is to apply the first-order Markov Assumption. This allows us to assume that each latent variable h_t is only dependant on the previous latent variable h_{t-1} in a temporal ordering $t \in \{1 : T\}$. This model is known as a Hidden Markov Model (HMM). POS Tagging is a success story for the HMM [4].

2.3 Hidden Markov Models

The HMM assumes an underlying latent state space $\mathcal{H}$, with transition probabilities between members. These variables are conditionally independent of each other, given the previous h_{t-1}, i.e. the first-order Markov Assumption:

$$P(h_t|h_{t-1}), \quad h_t, h_{t-1} \in \mathcal{H}, \quad t \in \{1:T\} \tag{2.1}$$

$$I(h_t, h_i|h_{t-1}), \quad i \in \{1:T\} \tag{2.2}$$

And a visible variable space $\mathcal{V}$, which are conditionally independent of each other given the corresponding latent variable at time t:

$$P(v_t|h_t), \quad h_t \in \mathcal{H}, \quad v_t \in \mathcal{V}, \quad t \in \{1:T\} \tag{2.3}$$

$$I(v_t, v_i|h_t), \quad \forall i \in \{1:T\} \tag{2.4}$$

Given these assumptions we can write the entire joint probability distribution as:

$$P(\{v_t\}_{t=1}^T, \{h_t\}_{t=1}^T) = \prod_{t=1}^{T} P(v_t|h_t)P(h_t|h_{t-1}) \tag{2.5}$$

Which simplifies the complexity of the problem enormously.

2.4 Recurrent Neural Networks

2.5 Review of Previous Work on Automatic Harmonic Analysis

Chapter 3

Design and Implementation

3.1 Hidden Markov Model Based System

3.1.1 Design

3.1.2 Implementation

3.2 Recurrent Neural Network Based System

3.2.1 Design

3.2.2 Implementation

Chapter 4

Experimental Results

4.1 Datasets

4.1.1 Weimar Jazz Database

4.1.2 KP Corpus

4.2 Metrics

4.3 Baselines

4.4 HMM Results

4.5 LSTM Results

4.6 Other Results

Chapter 5

Conclusion

5.1 Discussion

5.2 Future Work

Chapter 6

Interim Report Hereafter

Chapter 7

Introduction

The aim of this project is to learn a model that can map a musical melodic sequence to to harmonic sequence. More specifically this model takes a temporal, monophonic sequence of notes and predicts the corresponding sequence of chords. These chords correspond to the harmonic accompaniment intended by the composer to be played along with the piece.

The primary assumption of this project is that it is useful to treat the melody as an observed sequence generated by the underlying chord sequence. In this sense we have a noisy channel model, similar to a natural language processing problem, where the note sequence can be likened to a sequence of words in a sentence and the chord sequence corresponds to the underlying meaning represented in these notes.

To this we can add the Markov Assumption. Specifically that the notes generated by a chord a time t_i are conditionally independent of the notes generated by chords at all other times $t_j, i \neq j$, given the chord at t_i.

$$P(c_t|n_t) = P(n_t|c_t)P(c_t|c_{t-1}) \tag{7.1}$$

where c_t is the chord at time t and n_t is the sequence of notes at time t.

This divides the problem into two distinct parts.

The first is modelling the generative process from chord to sequence of notes, here called the Emission Model. This model is very dependent on the frame of notes considered to be generated by a chord. This project works with two possible variants, either the Emission Model generates all the notes observed in the time frame associated with a certain chord, or we consider each note on it's own as independently being generated by a single chord instance and ignore the concept of frames. In the second case there will be a one-to-one correspondence between the chord sequence and the note sequence, and hence the former will consist of many repeated sections of chords.

The second model we must consider is the transitions between the chords, here on the Transition Model. Each chord can be thought of as a state, and hence every transition a movement between two states. Under a first order Markov Assumption, where we assume that the current state is only dependant on the previous one, then the transition model becomes a bigram model between chords, and the combined system becomes a Hidden Markov Model. Other possible transition models will be the focus of the remainder of this project.

The data required for this project is symbolic melodic sequences annotated the accompanying chord. Hence this is not a system that is built to model raw sound data, neither is it an unsupervised model that can develop a notion of chords from unlabelled data. In the first year of this project data was used from the Weimar Jazz Database. This data consists of transcribed solos from jazz musicians over jazz standards. The data is heavily annotated and comes with chord labels drawn from the standards. However solos are a special subset of melodies with huge room for seemingly random improvisation and other forms of noise. This makes the problem significantly harder and hence one of the aims of this year was to find a dataset more suitable for the task and to further investigate the difficulties in using this data.

Chapter 8

Year 1

The focus in the first year of this project was on finding the best Emission Model to use. Three methods were developed, a 'Bag of Notes' model, a 'Chord Tone' model and a 'Concatenative HMM'. They were all combined with a simple bigram transition model and trained using the Viterbi algorithm.

8.1 Bag of Notes

The Bag of Notes model is analogous to a bag of words model, commonly applied to document analysis, where the frequencies of individual words are used to classify the document. Here the frequencies of notes in a frame are used to determine the chord. The training process creates n smoothed and normalised frequency distributions over the m possible notes, where n is the number of chords being classified.

Note that typically in this project $n = 12$ and we are training the model to classify the root note of the chord, as opposed to the particular variant of the chord i.e. whether it is major, minor or dominant. The number of possible notes, m, is typically also set to 12, where we are interested in Tonal Pitch Classes instead of absolute values. A Tonal Pitch Class (TPC) is an integer between 0 and 11, representing the relative separation between the note and the key of the song. Hence in the key of C, a D note would have a TPC of 2, and a C note would have a TPC of 0.

The failings of this model lie in the fact that it throws away sequential data and has no underlying musical intuition associated with it. The following model also discards local sequential data but is inspired by musical assumptions.

8.2 Chord Tone Model

The chords that are trying to be learned in this project are represented by root notes, but in reality are represented by a number of chord tones. In the case of majors and minors there 3 chord tones per chord, these are the root, the 3rd and the 5th. It can be safely assumed that in a melody over a chord, the chord tones of that particular chord are more likely to appear than the other notes. Hence a probabilistic model can that identify the chord tones in a melody has more information about the underlying chord than a blind frequency distribution.

Experiments were carried out to find the best method of determining which notes are more likely to be chord tones. The resulting model had two layers, each with a decision tree trained for classification. The first level took as input a feature vector representing a note. This vector had encoded metrical, structural and melodic information about the note. The dataset was modified to label which notes in the training set were chord tones, which the decision tree was then trained over. The second decision tree then took only the notes that were classified as chord tones and predicted the underlying chord. This layer was trained in a similar way, taking the chord tones in the training data as input and classifying the given chord labels. This two-layer system then formed the Emission Model and, together with the HMM Transition Model, used the Viterbi Algorithm to produce the most likely sequence of chords.

8.3 Concatenative HMM

The above model produced much more promising results than Bag of Notes, but it still disregards all local sequential data. Intuitively a lot of musical information is stored in the intervals between notes, and it was thought that a model that could capture this would perform better when determining the underlying chords. The Concatenative HMM attempts to capture this by training n different HMMs, one for each chord, over frames of notes 'generated' by that chord. An observed sequence can then be run through these models, producing probabilities. These are then the emission probabilities for that sequence of observed notes, given the chord. Combining these with the Transition HMM model can be thought of as concatenating together a long list of smaller HMMs.

There are two possible methods of training this model, supervised and unsupervised. As it is not the state sequence, but the probabilities that we are interested in for the lower level HMMs, the musical structure that the states would represent seems unclear. With an unsupervised approach, all that is necessary would be to define the number of states, then the Expectation Maximisation (EM) algorithm can be used to learn the optimum state sequences for the training data. This approach will be discussed later in the report.

The other approach is to train in a supervised way and decide the representation of states manually based on musical knowledge. Again chord tones were used as a suitable middle ground between chords and notes, and the states of the low level HMMs were trained over labelled data, where each note was annotated with chord tone information. In this model, however, multiple levels of chord tone were decided on, based on the intuition that the root and the 5th are more strongly related than the root and the 3rd, and that 7th is also used in some chords and could be considered a weaker form of chord tone. Hence multiple variants of this model were attempted, with different

Table 8.1: Year 1 Results

Model	Accuracy (%)
Bag of Notes HMM	22.5
Concat HMM	30.7
Chord Tone HMM (DT)	27.5

numbers of internal states for the HMMs.

8.4 Implementation

The language used for both years of this project in the implementation is Python. This choice was made based on a familiarity with the available libraries and a suitability for the domain of problem. Python is suitably high-level and well adept at processing scientific data.

The first part of this project involved an implementation of generalized HMMs for supervised learning, where the Emission Model and the Transition Model are modular, in the sense that the same HMM implementation could be used over the above three models, requiring only a class that could be trained and produced log probabilities given input data for the Emission Model. This was not readily available in pre-existing libraries. This involved implementation of the Viterbi algorithm.

The data was stored mainly in Python data structures and Numpy arrays. Scikit-learn was used for cross-validation methods.

8.5 Evaluation

The evaluation of these models is subtly difficult and tends to be overly critical on the jazz corpus. This is due to the presence of jazz substitutions, where many different chords can be played in the place of one another for very little change in harmonic structure. Nonetheless the clear way for evaluating this system is to test over a set of data not used in training and to compare the generated chords with the ground truth chords and create an accuracy metric. The results for the above models, classifying 12 chords, are presented in Table 8.1.

Against a baseline of random chance, 8.3%, these models show some promise, but not as much as expected. These results promoted a more in-depth investigation into the data that will be detailed in the rest of this report. Another important discussion is the use of random choice as a baseline. This is overly generous to the models, a baseline that takes into account some basic music knowledge might be more enlightening as to the quality of both the models and the data.

Chapter 9

Year 2

9.1 Work Done So Far

9.1.1 New Data

The first task faced in the second year of this project was to find a new dataset. This would allow for a more robust analysis of the previous year's models and the opportunity to develop better models this year.

A remark should be made about the difficulty in collecting data in this field. There exist many private collections of song data annotated for this purpose, some with the intention of being made public, but copyright laws cause this to be very difficult in practice. Of the data that does exist in the public domain, much of it is either melodic or chord data, rarely are the two aligned for the purposes required by this task. The Weimar Jazz Database is a superb resource, it is unfortunate that harmonic analysis of data in the specific domain of jazz solos turns out to be very difficult.

After investigation, the most promising new source of data appeared to be the KP Corpus of annotated classical melodies. It consists of midi versions of 46 classical excerpts, annotated with chord names.

The next step was to write a midi parser that would transform the data into Numpy arrays. MIDICSV was used to convert the data into csv format then Python's csv library was used to parse the data into Numpy. From here the models used last year were re-implemented. The results are presented in 9.1.

Table 9.1: KP Corpus

Model	Accuracy (%)
Bag of Notes HMM	3.4
Concat HMM	21.7
Chord Tone HMM (DT)	19.7

Although the data in this corpus is much more suited to the task, there is much less data than the Weimar Jazz Database, so the models end up performing slightly worse.

9.1.2 Unsupervised HMMs

The Concatenative HMM model used supervised HMMs as the Emission Model, as described in Section 2.3. Part of the continued work of this year was to reimplement this model with unsupervised HMMs, trained using the Expectation Maximisation (EM) algorithm.

The HMMs used previously were implemented from scratch, to allow for generality, but for these unsupervised models the hmmlearn python library was used. The notes were encoded into one-hot vectors and n Multinomial HMMs were trained, one for each chord.

9.1.3 Unframed HMMs

So far every model in this project has processed the melody into frames, with one chord 'generating' a small sequence of notes. Hence the Emission Model has been a sequential probabilistic model and the Transition Model has been a simple matrix of transition probabilities between chords.

For the next model we will alter our assumptions slightly, so that instead there is a one-to-one correspondence between notes and chords. Hence the data is modified so that the labels consist of repeated instances of each chord. This translates to many self-transitions between states and a simple chord-to-note Emission Model.

In this way the entire system can be represented as an HMM with multinomial emissions, where the notes are again encoded as one-hot vectors.

This model surprisingly out performed all others on the KP corpus, achieving an accuracy of 51.04%. Interestingly, the same model trained on the Weimar Jazz Database predicts only the tonic chord every time, scoring 32% accuracy. This can be taken as evidence that the KP corpus is better suited to the task of chord prediction.

9.2 Work Still To Do

9.2.1 Data Analysis

The next task to be completed will be a deeper analysis of the songs from both corpora, to confirm the intuitions drawn from working with both and the results of the models.

This will involve manual analysis of individual songs, where I will attempt to perform the task being carried out by these models to determine the inherent difficulties associated with each corpus. This will also help to inform an understanding of how the models will capture the associations between melodies and chords.

9.2.2 Baselines

The baseline chosen to compare the models to was one of random chance, i.e. $1/n$ where n is the number of chords. This is overly simplistic and hence two more baselines will be implemented as models for comparison.

The first will determine the tonic chord and guess this every time. Note that this is what the unframed HMM did in practice on the Weimar Jazz Database. This may be an overly harsh baseline, as it is quite hard to beat (the tonic chord is almost always the most common, and most melodies in a key will fit over the tonic).

The second baseline will take the starting note of the song as the root of the chord that it guesses every time. This seems a fairer baseline as it inherently models less information about the whole structure of the song.

9.2.3 Sequence to Sequence Model

The final step in this project will be to implement a model that differs quite significantly in its assumptions from the others. This will be a sequence to sequence model, consisting of an encoder and a decoder. The encoder will take the note sequence of the song and encode it into a fixed dimension vector. The decoder will then take this vector and decode it into a sequence of chords. These two models will be Recurrent Neural Networks (RNN), implemented in tensorflow. The RNNs used will be a basic RNN Cell, a Long Short Term Memory Cell (LSTM) and a Gated Recurrent Unit Cell (GRU).

Bibliography

[1] Cynthia Cohen. Music: A universal language. *Music and conflict transformation. Harmonies and dissonances in geopolitics*, pages 26–39, 2008.

[2] Leonard Cohen. *Songs from a Room*. Song BMG Music Entertainment, 1969.

[3] Bob Dylan. *Blood on the Tracks*. Ram's Horn music, 1975.

[4] Julian Kupiec. Robust part-of-speech tagging using a hidden markov model. *Computer Speech & Language*, 6(3):225–242, 1992.

[5] Eric Regener. *Pitch notation and equal temperament: A formal study*, volume 6. University of California Press, 1973.