N-gram Analysis of Melodic Sequences

Ivan Viro
New York University
ivan.viro1@gmail.com

Paul Odenwaldt
New York University
pao263@nyu.edu

Abstract

The aim of this paper is to demonstrate the benefits of applying Natural Language Processing techniques to musical analysis and synthesis. We apply a Hidden Markov model to melodies with respect to the time periods in which they were written. This allows for analysis of historical trends, estimation of when the melody could have been written, and generation of random melodies based on the style of a given time period.

1 Introduction

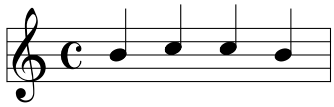
1.1 Motivation in looking at music through NLP

The analysis of human language through Natural Language Processing techniques has yielded many fruitful tools and perspectives which can examine large-scale patterns and trends, a task that would be impractical to do purely by hand.

With the techniques and insight provided by decades of NLP research, it is especially tempting that they can be implemented to process music in analogous techniques. Music has hidden structure and grammar, like text (Wołkowicz and Kulka, 2008). Recently, there has been a growing trend in using Natural Language Processing techniques to better understand and analyze music, because this approach can yield useful tools capable of examining big data.

1.2 Scope

We will focus only on the order of notes, and year of composition. The text-language analogue of note order is syntax. And since n-grams have been a popular choice to study syntax of texts, we will consider melodies as n-grams of intervals (in half-steps) between adjacent notes.

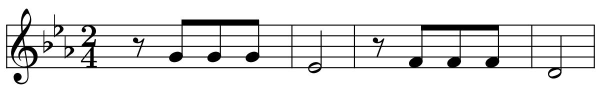


For example, this snippet would be represented as a 3-gram of `1 0 -1`, because the pitch distance between the first two notes is one half step, zero half steps between the middle two notes, and -1 between the penultimate and the last note.

1.3 Relative vs absolute pitch

By working with n-grams based on relative pitch instead of absolute pitch we avoid pigeonholing the same musical melodic idea to each specific key that we see it occur in. This allows our system to interpret music closer to how a human listener would: the sequences (themes) take precedence over key.

Notice that if we define our n-grams like this, then we will be blind to absolute pitches. Here's an example:



This snippet (the iconic beginning to Beethoven's Fifth Symphony) would be transcribed in our n-grams as the following:

```
0 0 -4 2 0 0 -3
```

Remember that each number represents the pitch interval between adjacent notes. Notice how the main idea of the music ("ba-da-da *dum*") is recorded as `(0 0 -4)` the first time and `(0 0 -3)` the second time it occurs. The difference that we see marked here is between a major third and a minor third, and nothing else. This makes it easier to see why the second "ba-da-da *dum*" is not exactly the same interval as on the first descent. It's easy to see that although the first and the second half of the snippet don't share any of the same pitches, they are still similar to both our ear and our n-grams. This is a good example of why measuring relative pitch instead of absolute pitch in melodies is better: musical periods and styles are defined by melodic sequences, and not the keys in which those melodies occur.

2 Related Work

2.1 Peachnote

Peachnote's Music Ngram Viewer is an n-gram melodic sequence search engine that came out of processing 2 million pages of scanned sheet music from the International Music Score Library Project (otherwise known as the Petrucci Library, or IMSLP) (Viro, 2011). It has the individual lookup of melodies and can display trends over years.

We aim to improve that by letting the user organize music by time periods of any length of years, rather than by only individual years. This option would help make music analysis more accessible to those who don't wish to see the detail of year-to-year changes. Instead, trends which happen between periods of 25 years or half-centuries might be more interesting to someone who doesn't study music professionally.

Our motivation was to add more layers and functionality given the available n-gram data through examining and generating melodic sequences.

2.2 Existing Research

Existing work which uses n-grams in the context of music has mostly concerned two tasks. The first one is classifying and predicting composers based on n-grams and training models (Wołkowicz and Kulka, 2008), (Wołkowicz and Kešelj, 2013).

The second one is making a search engine which determines melodic similarity by comparing n-grams (Urbano et al., 2010), (Doraisamy, 2005).

2.3 Where our work fits in

The work that we decided to take on focuses only on year of composition. This is useful because it allows for identification of trends across time. While this is something that is studied prominently by musicologists, there is a lack of research in the realm of computerized approaches to such analysis. Because of this, we are going into less charted territory. This is exciting since it is more novel, but makes it more difficult to rigorously compare our results to previous work.

3 The Given Data

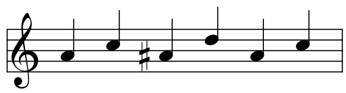
3.1 Starting from the given CSVs

The n-grams we used can be found in www.peachnote.com/datasets.html under "Melodies". It consists of separate csv files for each n-value, where each line contains the melodic sequence and number of times it appeared in a given year.

For example, the csv line of

```
3 -2 4 -5 3 1804 94
```

tells us that the sequence



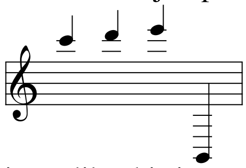
occurred 94 times in 1804.

3.2 Potential errors in the given data and considerations

Because these n-grams were generated from scanned images sheet music, this introduces a source of potential errors. Whenever interpreting the scanned images, there is a risk of misinterpretation. This can range anywhere from misreading an ink stain as an accidental sharp, thus changing the following note, to misreading the clef which would result in getting all the notes on the line incorrectly.

The good news is threefold. First, since we only care about the intervals between notes, if all notes are shifted up or down by accident, or if there is an error in reading the articulations or rhythm, then we still process them accurately in our n-grams. Secondly, the errors in relative intervals between notes that we do see are relatively rare. And finally, since the probability that an interval gets misread is independent of what the interval actually is, the result is that errors follow a pattern of noise.

This noise adds occurrences of n-grams by a small fraction of a percent fairly uniformly for a given collection of n-grams. The easiest moments to see this happen is when there is a sudden jump in a sequence:



However, a jump like this is not necessarily always an error: if the leading part was switched from one instrument to another in a lower range, for example from a flute to a tuba, then this is an accurate transcription.

And ultimately, since the noise that is from valid errors settles fairly evenly given the size of the total data, is easy to filter out and it doesn't influence the results which we will get to.

4 Data Parsing

4.1 Scraping

The first intuitive step is to count up the individual year occurrences of a given melodic sequence across a defined year range, since looking at each year individually is not very helpful when trying to see larger trends across music.

We decided to name our system of tools Ottaviano, named after Ottaviano Petrucci, the Italian renaissance music-printer after whom the collection of the scanned sheet music that we used (IMSLP) was named. In it, we decided to let the user choose the length of years to chunk identical n-grams into (for example, to look at unique n-grams for each century, or half-century, or 5 years, etc.).

4.2 Generating a Hidden Markov model

After chunking unique n-grams into buckets of time periods (or as we refer within the context of our program as *epochs*), we generate a Hidden Markov model (HMM) on those unique n-grams which are in organized by epoch.

The Hidden Markov model generates a prediction for each (n-1)-gram for what the following (final) note/pitch interval is going to be. We do this by organizing our n-grams into unique (n-1)-gram subsequences, and then writing each recorded unique option for the last interval, and mapping that pitch option to its occurrences divided by the sum of all occurrences of all final interval options for the given (n-1)-gram.

For example, if we have the following n-grams with their occurrences:

```
-2 0 2 1 3     2
-2 0 2 1 0     1
-2 0 2 1 2     1
```

The result will be that the sub-sequence `-2 0 2 1` has a probability .5 of going to `3`, and an equal probability of .25 of going to either `0` or `2`.

If we run this on every n-value file in our unique n-grams that are categorized by epoch, then immediately we can see how preferences change or remain the same across selected time periods when concerning any melodic sequence. For example, we can see if a composer from the 15th century would finish a given sequence of notes differently than a composer from the 19th century.

5 Synthesis

5.1 Generating a melody when given an epoch

Connecting all Hidden Markov models from the different n-values gives us an ability to select an epoch and then probabilistically generate a sequence by "walking down" the n-gram HMMs. The first step is to find the first interval, and this can be done by looking at the unique 1-grams for the selected epoch, and selecting one at random from the probability density function of the intervals over their occurrences. Following that, for the given epoch, we match our current sequence with the unique (n-1)-gram, and we look at our discrete probability density function of potential values to select one at random, appending it to our generated sequence. We then repeat the final step for as many times as we wish, or until we run out of n-grams (which in our data set ends at 15-grams).

The result of this is a melody which although having no information about the rhythm, is guaranteed to be like something which already exists in history. The more likely that this melodic sequence occurred in the given time period, then the more likely this algorithm would have generated it.

5.2 Comparison to similar existing work

To our best knowledge, this is the first time that someone has made a general probabilistic method to generate a melody given a time period, instead of relying on machine learning and using fitness functions to imitate audio or MIDI training sets. The keyword *general* is important here, since there have been well-defined probability models based on counterpoint rules and grammar, but those are limited to a very specific scope in time and style. In our HMM system, we don't restrict to a specific grammar, which enables us to observe how it changes over time.

5.3 Evaluation and further work

The next step for this system is to see how indicative of the time period the melodic sequence actually is. This is difficult to measure since musical periods are fairly loosely defined, apart from their years of composition. One possible way would be to survey musicologists that study this professionally by asking them to date and rate in confidence each melody from a long list of those generated, and by then comparing their answers to the actual date that the melodies were generated from. In our anecdotal experience, the system works well and is fairly representative.

6 Data Analysis

6.1 Top n-grams per epoch

After we generate our unique n-grams from 4.1, we can look and see what are the top x most common melodic sequences for each epoch. This will prove most useful in trend analysis.

6.2 Popularity vs occurrence

When looking at the most common melody of any length from any time period, it will most likely be (`0 0 0 0...`), or a series of notes at the same pitch. After a moment of thought, this becomes understandable, since repeating the same pitch is often done in a piece at the beginning, end, or when holding a certain musical moment longer for emphasis or tension. However, even melody can be at the same pitch, with the rhythm carrying the attention of the piece.

One notable example of this is Gustav Holst's *Mars, the Bringer of War*, from *The Planets* suite:

Because we are blind to how popular a specific occurrence of an n-gram is, we only measure them by how frequently they occur in sheet music, rather than how much they were listened to specifically. In other words, we can't rigorously measure the popularity of specific pieces in history, so the most popular n-grams likely skew slightly towards melodies that occurred in the highest number of works by volume, as opposed to a few specific works that were listened to the most.

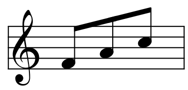
Thus, since we measure popularity by number of occurrences in unique pieces rather than popularity in performance or influence, this is another reason why a fragment of a melody that doesn't change in pitch is usually the most popular one for any given time period.

6.3 Revitalized arpeggios, or Beethoven's mark

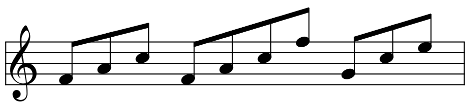
Beethoven is particularly special in Western Classical Music for not only bringing about the Romantic period, but also for seeing his own popularity and renown in his lifetime. The following generation of composers was defined by how they emulated or overcame Beethoven. In the words of Brahms, ”You have no idea how it feels, when one always hears such a giant marching behind one” (Bonds, 1996).

One characteristic mark of the Romantic music style (whether for a full orchestra or for a solo piano) is arpeggios, or notes in a chord that are played sequentially, rather than at the same time.

For this reason, the top fifteen sequences of 1775-1799 yield no arpeggios, but then in 1800-1825 (the latter part of Beethoven's career), we see



break into the top 15 most used sequences. Then following Beethoven's death in 1827, the next period of 25 years has the following arpeggios become mainstream, before falling out of popularity following 1850 until present:



And although similar remarks have already been made by musicologists studying this by hand:

> ”[one of the] three characteristics are shared by most arpeggios in Romantic piano music: They are played from the bottom up (i.e. in order of increasing pitch)” (Repp, 1997)

We can actually see this being the case for arpeggios in the 16th and 17th century, which were relatively frequent until they largely fell out of favour in the 18th century before being revitalised again in the 19th. However, an important note to make here is that the preferred ones then were not all the same intervals that became popular in the first half of the 19th century. Still, this is useful insight since we usually see arpeggios as being characteristic of the early 19th century music, and while it certainly was novel in the context of a listener from the 18th century, there is a strong case to be made for arpeggios to have been brought back from earlier style preferences.

This is an example of how this tool can be used to not only validate existing ideas in musicology, but can also make novel discoveries which furthers our understanding of the history of music.

7 Epoch Estimation

This probabilistic analysis of n-gram melodies over time is a very useful lens for viewing the changes in composition styles throughout music history, but so far it is only expanding the context in which a musicologist or researcher would normally view such information. In other words, this display of trends over the years is certainly interesting, but it still requires significant knowledge of music history in order to really explain it. What may be more useful to the average listener or musician would be the ability to go in reverse: to start with a melody and immediately determine what era it is from.

7.1 Raw Trend Recognition

Given the previously described scraped and organized data, the process of querying with a specific melody is fairly straightforward. Due to the very large amount of data, any given melody will likely occur many times in many different eras (depending on how complex that melody is). Given the frequency with which a melody occurs in some era and the total number of melodies from that era, we can calculate how popular that melody was in the given era. The advantages of using relative-pitch n-grams (as described in section 1.3) make this approach both reliable and accurate, since it highlights trending themes rather than specific melodies. In fact, a query with some melody is more precisely described as a query with a theme or idea - irreverent of pitch or key.

So, when queried with a melody/theme, the system checks how popular that theme was in each era by dividing the number of occurrences of the queried melody by the total melody count (mc) from that era:

$$probability_{\text{era}} = \frac{occurrences_{\text{query}}}{mc_{\text{era}}}$$

Given the wide range of time that the system covers, it is possible that not all eras will contain the queried melody. In fact, this is almost guaranteed for any melody that is moderately complex, since music experienced exponential growth in complexity starting in the last classical/early romantic periods. When this kind of miss occurs, the likelihood is zero.

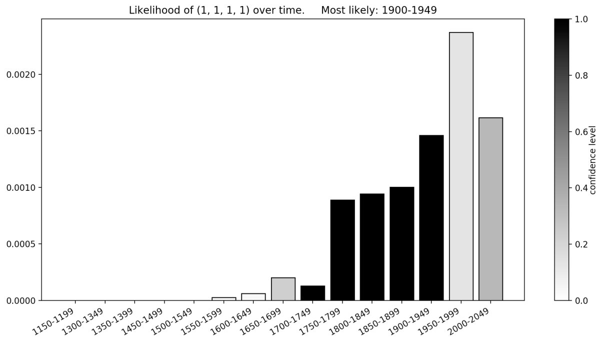


Figure 1

7.2 Confidence and Display

As stated, a melody query results in the calculation of a raw probability figure for each era. This takes care of local normalization as the probability is based on popularity within that era, but it does not account for the fact that each era contains varying numbers of total melodies, which makes overall comparisons more difficult. This problem is dealt with in two ways: the implementation of a confidence metric and an overall visualization of popularity over time.

To overcome the disparity between different amounts of data in each era, the raw probability is weighted with a confidence level. This confidence level is based solely on the melody count within a given era relative to the average melody count across all eras:

$$confidence_{\text{era}} = \min\left(\frac{1}{avgMC/mc_{\text{era}}}, 1.0\right)$$

To better understand why this is useful, consider an example: if one era has a probability rating of 20% but only contains ten total melodies, does that really indicate a peak in melodic trend in the context of the last one thousand years? With so few total examples it is hard to be sure. Compare that with an era with a raw probability of 15% and one-million total melodies. The overall popularity is less, but it can be stated with confidence that 15% of the music of the time contained that theme.

This is the purpose behind including confidence level: it is not necessarily meant to increase accuracy but rather to increase precision. By choosing eras with higher confidence, the system can be sure to make less incorrect predictions. Therefore, the total probability for each era is calculated by multiplying the raw probability with the confidence level.

In order to retain information about other likely eras with lower (or even comparable) total probabilities, a graph is produced showing the likelihood of the given melody across all eras and the confidence level associated with each era. Figure 1 is an example output showcasing the impact of the confidence metric as well as the usefulness of the graph for analysis. While the era 1950-1999 has the highest probability score, the confidence level of the era 1900-1949 is much higher, so the ultimate estimation is 1900-1949. As a music history aside, the queried melody `(1 1 1 1)` indicates 5 chromatic notes, a trend largely popular in the period of Serialism. Serialism began in the early 1900's and gained popularity through the middle of the century. So the estimation is historically correct, and the confidence level here happens to be stronger during the rise of Serialism. Not only does the graphical representation of the prediction give insight into how the actual prediction was made, but it also shows how the queried them changed in popularity over time, as is the case with Serialism in Figure 1.

7.3 Evaluation of epoch estimation

A good way to evaluate this tool is to divide the data set into separate development and training sets. Because the full n-gram collection that we used consists of approximately $200,000$ unique pieces, and since the total n-gram count is 960 million, using the system on a piece which wasn't in the data set usually yields very similar results.

8 Conclusion

After processing melodic sequences like words through n-grams, we have made programs that can do the following:

- when given a melody, show the likelihoods of what the next interval will be in a specific time period
- when given a melody, show the probability distribution and confidence of when it was written
- generate and play back a melody based on a year or time period of composition

These tools can be used by composers, musicologists, or anyone who wishes to interactively examine Western classical music. The use of techniques from NLP allowed for efficient and meaningful analysis of big data. We hope to see more research and development in this exciting intersection between NLP and music.

Further Work

The most potential that we see would be in doing what our existing melodic prediction and analysis tools can do with rhythm as well, and to merge those two. Putting that together would create a more complete score generation, in addition to being another dimension to see clustering and shifting in style and time period.

All of the tools that we have created are available to use, modify, and improve at `https://github.com/ivan-v/ottaviano`.

Acknowledgments

We would like to thank Peachnote's CSVs for n-grams, without which this project would have been impossible.

Additionally, this paper wouldn't exist without Adam Meyer's NLP class at NYU, for which this was originally written.

References

Mark Evan Bonds. 1996. *After Beethoven*. Harvard University Press, Cambridge (MA).

Shyamala Doraisamy. 2005. Polyphonic music retrieval: The n-gram approach. *SIGIR Forum*, 39:58.

Bruno H. Repp. 1997. Some observations on pianists' timing of arpeggiated chords. *Psychology of Music*, 25(2):133–148.

Julián Urbano, Juan Llorens, Jorge Morato, and Sonia Sanchez-Cuadrado. 2010. Melodic similarity through shape similarity. *Exploring Music Contents*, pages 338–355.

Vladimir Viro. 2011. Peachnote: Music score search and analysis platform. pages 359–362.

Jacek Wołkowicz and Vlado Kešelj. 2013. *Evaluation of n-Gram-Based Classification Approaches on Classical Music Corpora*. Springer Berlin Heidelberg, Berlin, Heidelberg.

Jacek Wołkowicz and Zbigniew Kulka. 2008. N-gram based approach to composer recognition.