Striking a final chord: Finishing Bach's Final Fugue with Hidden Markov Models

Anjali Nair (s4234790)
Department of Artificial Intelligence
University of Groningen
Groningen, The Netherlands
a.nair.4@student.rug.nl

Clemens Kaiser (s4460065)
Department of Artificial Intelligence
University of Groningen
Groningen, The Netherlands
c.m.kaiser@student.rug.nl

Jort Hessel (s3447626)
Department of Artificial Intelligence
University of Groningen
Groningen, The Netherlands
j.hessel.5@student.rug.nl

Mart Berends (s3006069)
Department of Artificial Intelligence
University of Groningen
Groningen, The Netherlands
m.a.w.berends@student.rug.nl

***Abstract*—Though in machine learning many advances have been made with regard to time-series prediction, any project involving automatically predicting musical continuations is laden with difficulties. In this paper, we use the (in)famous unfinished fugue Contrapunctus XIV from Bach's *the Art of the Fugue* to showcase two approaches with Hidden Markov Models (HMM) to predict the continuation of the unfinished fugue. We discuss the benefits and drawbacks that are specifically associated with these models and we end by discussing the inherent difficulties that are associated with this type of learning task.**

I. Introduction

In machine learning, there exist many approaches to tackle the problem of synthesising music. One of the more famous problems in synthesising music is finishing Contrapunctus XIV from Bach's *the Art of the Fugue*. According to the legend, either Bach failed to finish his work before he died, or he intentionally left it blank to inspire a new generation of musicians to creatively finish his work. Whether he wanted to stimulate creativity or not, it did eventuate in many musicians trying to finish Bach's work. Almost definitely unforeseen by Bach, however, was that his unfinished fugue would inspire an entirely different group of people - computer nerds. Over the past three decades a multitude of approaches has been proposed for automatically continuing Bach's fugue with many producing enjoyable continuations, but all, unsurprisingly, falling short of replicating Bach's musical virtuosity.

The problem has highlighted once again the fact that the human mind in its creative efforts is more than a number-crunching, digital machine. The issue at hand, therefore, is a highly complex one, one that we cannot plausibly expect to "solve" soon.

In this paper we will, nevertheless, give a description of our attempt at finding a method of finishing Bach's final fugue that is, if not Bach-like, at least mildly enjoyable at times.

A. Representation

The ability to compose music is largely learned. While we may imagine someone having never experienced music to be able to initiate some sort of singing, it would be difficult to imagine such a person writing entire works of classical music. However, the ability to process the different perceptual aspects of auditory stimuli that are classified as music is not learned but to a large degree hard coded in our bodies: the ear functions as a specific series of transformations on raw auditory input, and the neural circuitry directly connected to the cochlea functionally extracts specific features of that input. In addition to that, music generally adheres to certain cultural conventions that tend to be strictly followed by composers within that culture. For example, in the harmonic style of 18th century European musicians, the agreed-upon set of frequencies at which instruments were allowed to vibrate was the twelve-tone temperament system, and this convention was strictly followed by musicians in this period. Thus, it can be argued that humans are profoundly biased musical learners. This is why, for instance, using raw audio of a microphone recording of someone playing the unfinished fugue would be a representation that contains so much redundant information that it would be nearly impossible for any modern statistical techniques to extract anything useful out of this data; it would have to learn from scratch what features of the sound are relevant for the musical composition task.

In designing a system that is capable of automatic music continuation, music theory still plays an important role. This is because using music theory allows encodement of features that make output music-like. That is to say, in many conventional data types for representing auditory information, the space of values that would be considered to be even slightly music-like is only a very small subset of the total value space. The representation of notes, chords and themes should be found in a way that enables an algorithm to work with it.

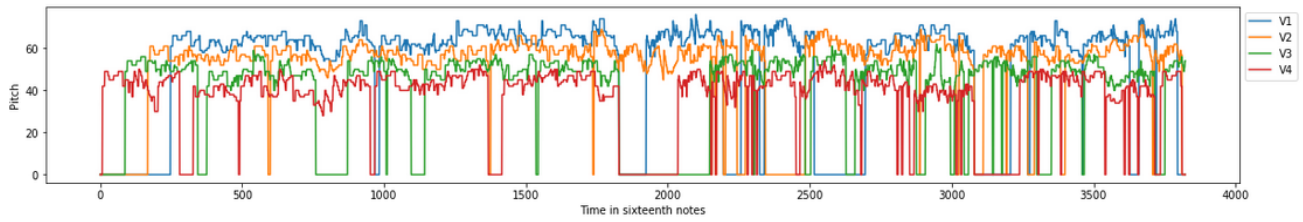

Fig. 1. Bach's Contrapunctus XIV visualized

B. Dataset

Since our objective is to make a system that can continue Contrapunctus XIV, the used dataset was the beginning of Bach's unfinished composition; the dataset consists of 3824 time steps. In the raw input file, the fugue is represented in four voices for every time step: $[V1, V2, V3, V4]$ these four voices correspond to soprano, alto, tenor and bass. These voices are based upon the MIDI note naming system, i.e. a scale from 0-11 (where octaves are represented by an addition of an integer multiple of 12 to the scale), as opposed to the Latin representation: A, A♯ /Bb, B, C, C♯/Db, D, D♯/Eb,E, F, F♯/Gb, G, G♯/A. For example, a line [0 0 0 42] represents an individual time step where the bass is playing a note whose pitch is represented by the number 42, while the remaining voices are at rest.

The dataset is visualized in Figure 1 and clearly shows the difference in pitch levels between the voices. Note too that the data set contains sixteenth notes. Throughout this paper, when we refer to "notes", we mean such sixteenth notes.

C. Objectives

The objective of this paper is to create a system that can automatically generate a good continuation of Bach's Contrapunctus XIV. A good continuation is one that sounds pleasant. However, pleasantness of sound is neither easily quantifiable, nor is it an objective measure. Unfortunately, there is no unequivocal *true* measure of the performance of the algorithm. The best way to measure the effectiveness is, therefore, still to listen to the music. As a consequence, we largely rely on the subjective measure of how pleasant we think a continuation sounds.

In this paper two different approaches will be tested and compared, both pertaining to the pleasantness of their produced continuations and to their theoretical underpinnings. Both approaches make use of Hidden Markov Models (HMMs).

II. Methods

In this section we will describe the two different approaches separately. In both cases, we conceptualize the music to be the product of the musical composition being in one of a set of hidden states. This assumption allows us to apply an HMM. In one case, we assume that the music is the result of the musical composition being in a given thematic state, and in the other we assume that it is the product of the musical composition being in a state that roughly corresponds to the concept of a musical chord.[1]

A. Model I

Every fugue has a higher-order structure, which can be described on different levels of abstraction. At a rather abstract level, it can be represented by themes. Every fugue has a primary theme – typically referred to as a fugue's subject, and some have secondary themes. At a more granular level, moreover, one might be interested in themes per a certain number of bars. Naturally, the chosen level of granularity depends on what one wishes to achieve by conducting such analyses.

For our first HMM, we were interested in a more granular thematic representation of the fugue. One approach of analysing such structures – the traditional one – requires deep knowledge of music theory. A more readily accessible approach is using a data-driven solution. We chose a data-driven approach and followed the four-step process for extracting themes outlined in [1]. The process is schematically depicted in Figure 2.

First, the data was transformed into the difference representation, which is given by $d_t = x_t - x_{t-1}$, where x_t is the note recorded at time t. Subsequently, the process illustrated in Figure 2a was carried out: a sliding window w of size 32 – corresponding to two bars – was moved over one voice after another in steps of eight notes – i. e., half a bar. Thus, the fugue was represented by 1909 overlapping data points in R^{32}.

Second, these data points were clustered using K Means. This step is indicated by Figure 2b, where each circle corresponds to a cluster of a different size. The number of clusters was determined by calculating and comparing the silhouette averages (as described in [2]) for clusters generated with K set to all values in the range from two to 50. The optimal K turned out to be 42.

Third, each cluster was treated as a symbol. The number of unique symbols lies close to the number of unique notes in the data set (46). However, as [1] point out, this is not an issue, as the extracted clusters capture the structure of the fugue at a time scale an order of magnitude slower than the notes representation.

Finally, the transitions between symbols were analysed as seen in Figure 2c. Some symbols had multiple successors,

[1]To view our code and sample outputs, visit https://github.com/anjalinair012/Bach-composition/tree/master.

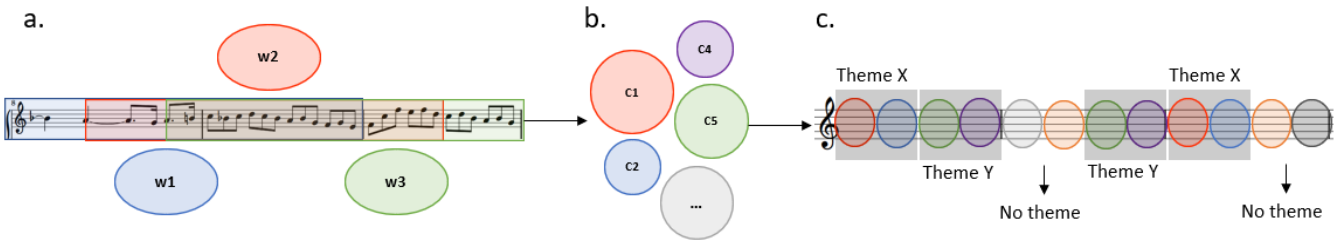

Fig. 2. Themes extraction process

while others were followed consistently by a certain other symbol. The former case indicates that there either is no theme or a theme has ended. In Figure 2c, this is, for example, shown by the orange circle, which is once followed by green, and later by dark grey circle. The latter case, on the other hand, indicates that two subsequent symbols are part of the same theme. Such is the case for the combinations of the red and blue as well as the green and purple circles. Consequently, these symbols were combined, yielding a smaller set of unique symbols. This process is then repeated until there are no symbols left that are consistently followed by some certain other symbol. In Figure 2c we see that consistent combinations of circles are grouped into themes. In our case, two iterations were sufficient to arrive at the final thematic structure, consisting of 29 unique themes.

These themes represented our HMM's hidden states. The transition matrix, hence, captures the probability of moving from one theme to the next, i.e.,

$$p(T(t)|T(t-1)), \tag{1}$$

where T stands for a given theme.

The corresponding observable states were the unique sequences of 32 notes associated with each theme. The observable is, accordingly, predicted by

$$p(S(t)|T(t)), \tag{2}$$

where S represents a unique sequence of notes.

B. Model II

The second approach is based on predicting the fourth voice from the first three voices. In this case, we are not looking at the themes of the composition, but rather at predicting the voices separately, and thereafter, using this prediction to find the pitches for the last voice. Thus, we hope to first purely model the melody of the voices and then add harmony to it through the latter part.

The HMM is built of three hidden states, each modelling a unigram prediction in voices $V1, V2$ and $V3$. The three voices together, vaguely capture the idea of a chord. The fourth voice is then predicted as an observable that harmonises with this chord.

For the three voices or hidden states, we build three individual transition matrices. Each matrix predicts a pitch for a voice with respect to only the pitch that appeared in the previous time step, i.e,

$$p(v_n(t)|v_n(t-1))\, where\, n\, \epsilon\, 1, 2, 3 \tag{3}$$

The 3 voices predicted as independent hidden states models a chord and with this assumption, we predict the fourth as an observable from the predicted hidden state as,

$$p(v_4(t)|v_1(t), v_2(t), v_3(t)) \tag{4}$$

The emission matrix defines the probabilities of transition from every known combination of $V1, V2$ and $V3$ to a pitch in $V4$. The problem of "known combination" requires some extra attention. As the three hidden states are independently predicted, there is a high probability that the combination achieved at some time step is not known (has not appeared in the dataset). This would mean that there's no possibility for this combination in our existing emission matrix. While this is a fallback, we see it as an opportunity to model some randomness. In such a situation, we make a unigram prediction for $V4$ too. So, we do not entirely overfit and see some new harmonisation, whether bad or good. Starting from the end of Bach's composition, we begin prediction with our HMM. A total of n notes were predicted which made an attempt to complete 25 seconds of the fugue.

C. Results

As discussed in the introduction, the results are hard to quantify. We converted the 4 voices to a music file, by playing this we looked at how the predictions sound. As expected, a lower proportion of encounters in the original dataset for the combinations of the predicted pitches leads to a prediction that is more in line with the original composition.

The first approach predicts sequences of 32 notes at a time. These sequences were composed by Bach himself, and therefore, are pleasant to the ear. However, in this model each voice is predicted without consideration of the other three voices. Consequently, the produced music lacks harmonisation. Naturally, this is highly detrimental to the listening experience. Furthermore, this approach, in a sense, merely reshuffles existing compositions. While this ensures that the individual sequences produced per voice are enjoyable, it strongly restricts the room for creative maneuver.

The second approach is inherently stochastic. Every run of the model gives a different result and the extent of randomness in the prediction of V4 also varies. In the sense, with certain

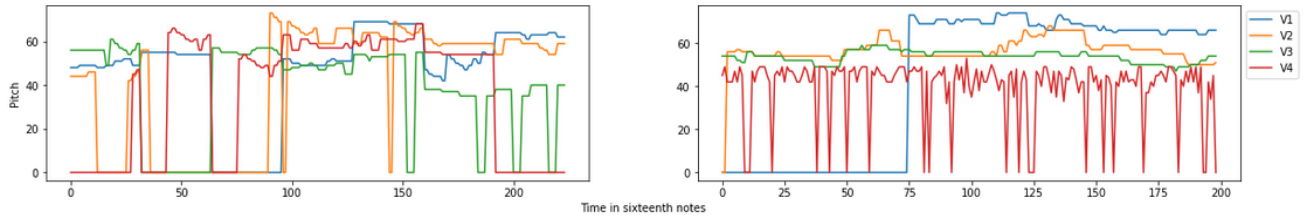

Fig. 3. Sample predictions by Model I (left) and Model II (right)

runs, every hidden state predicted is seen in the dataset and thus the predicted V4 harmonises perfectly. In this case, our model's prediction resembles that of an over fitted model and produces no new combination of pitches. On the other hand, certain runs make transitions to new hidden states which have never been seen before. This may sound good or terrible. We hypothesize, the best runs would be those which make mostly known transitions with intermittent unknown ones. This would add some creativity to the creation.

Figure 3 visualizes sample predictions made by model I (left) and model II (right). Comparing them to Figure 1, we can immediately tell that neither of them is a continuation in accordance with the original parts of the Contrapunctus XIV. Somehow even just by observing these simple figures, we find that the predictions do not possess the same level of creativity and beauty as the original, and appear less organic. A fairer comparison, perhaps, can be made between the two model's predictions. At least two differences can be observed from the plots.

First, model I predicts one or multiple voices to be zero, i.e, to be pausing, frequently and for fairly long stretches. Model II predicts the first voice to be silent for almost the entire first half, but thereafter neither of the three voices operating as hidden states goes to zero. Only the fourth voice is repeatedly made silent. In model I there is a theme that is commonly associated with stretches of pause sequences, and most commonly this theme is followed by itself. As a result, when a voice goes to zero, it is likely to stay there for some time. For an individual note, the most common transition is not to zero. Consequently, the chances of a voice turning silent in model II is small compared to model I. On the other hand, once they are silent, chances are, they will stay there for a bit because a pause is most frequently followed by another pause.

Second, while the relative difference in pitches is retained in model II, model I ends up "confusing" the four voices. This is likely a consequence of using the difference representation when clustering the sequences of notes. This representation led to the relative differences between the voices being lost. The extracted themes were commonly associated with sequences of more than one voice. As a result, a theme might predict a soprano sequence for the bass voice.

III. Discussion

In this section we will discuss the drawbacks and benefits of model I and model II. Then, we will illustrate the general difficulties associated with automated music generation. To close off, we discuss how automated music continuation might constitute a peculiar exception to the danger that is (justifiably!) associated with neglecting to implement measures that counter overfitting.

The first approach has the benefit that it can capture higher-order structure in the music. With this set-up, pieces of melody can be preserved, reproduced, and theoretically, even slightly modified. This set-up is particularly well suited for composing fugues, since themes are important building blocks for such compositions. A downside to this approach is that, when producing a window-sized piece of music, voices are not well coordinated - the harmony between them is (at least not intentionally) considered. One harmonic danger with this set-up in particular is when the exact same melody is introduced across voices but with a slight time shift such that connecting/ornamental notes in one voice temporally overlap with the notes that are being connected to or ornamented in the other voice, because this would introduce a substantial amount of dissonance.

The second approach has the benefit of mimicking another common approach for composing music, which is considering the piece as a melody that is accompanied by chords. We deviated slightly from the traditional conception of chords, by redefining a chord just to be the juxtaposition of the pitches of the bass, tenor, and alto ($V1$, $V2$ and $V3$ in our previous terminology). Since fugue composition proceeds in large part through repetition and transformations of specific themes, it is debatable to what extent a fugue can be conceptualized as consisting of a melody accompanied by chords. Still, breaking down the composition in this way may allow for a more tractable composition problem, because we can ignore thematic or horizontal (i.e. melodic) structure and focus on good-sounding harmony. Besides, horizontal structure might still be implicitly encoded in the specifics of the transition and emission distribution estimations. A drawback of this approach, besides the loss of higher-order structure and the choice for a random pitch when a given "chord" has not been encountered yet, is that these three voices are predicted as three independent processes, whereas in actuality they are dependent: in reality, the choice of a pitch for $V3$ is in fact informed by the choice of $V1$ and $V2$. However, modeling these complex dependencies is extremely difficult, due to the minuscule amount of training data points we had at our disposal for this problem. This brings us to one of our personal discoveries from this project.

As hinted at earlier, automated music continuation is inherently fraught with difficulties. As mentioned in the section on representation, music perception is a process that in humans is achieved through very complex structural bias - the human brain and body are "prewired" to perceive and understand music. Any attempt at creating a purely statistical learner (without any pre-implemented bias) that continues musical compositions is hopeless, for it has to learn to understand the basic structure of music before it can understand the relationships between elements within different structures. Therefore, the kind of creativity we expect human composers to have is nearly impossible to achieve with a purely statistical learner. There is, however, still one way for a contemporary automated music continuation system to create something musical: and that is by essentially parroting that which has already been written.

Finally, a note on overfitting: we think automated music continuation is a unique class of machine learning problem in virtue of its relationship to the classical underfitting-overfitting trade-off. We argue that, in the practice of automated music continuation, the conceivable ways in which models can underfit vastly outnumber the conceivable ways in which models can overfit. One ofthe reasons for this is that repetition is so embedded in the nature of music that it is hard to introduce *too much* variance across retraining instances of an automatic music continuation system across different compositions: after all, when considering the possibilities with a purely statistical learner for this task, a system that reproduces time series that are almost identical to the training data is still relatively desirable. Yes, ideally we want an automatic music continuation system to be able to introduce new ideas into the composition, but for the reasons discussed above this is a very nontrivial task to accomplish. Thus, we have deliberately abandoned conventional methods of preventing overfitting in hopes of 'cheating' our way into making a system that can produce acceptable output, consciously limiting the creative potential of our system in light of the larger cause of producing something that is at least remotely palatable to listen to.

A. Conclusion

Considering only individual voices, i.e. melody, the music created by model I is reasonably pleasant. After all, the predicted 32-note sequences were composed by Bach. However, the real beauty of a fugue lies to no small extent in the richness created by the interplay of the different voices, i.e., the harmony. A good, but nontrivial extension to model I would hence be a mechanism that attempts to harmonise the predictions for the four voices, and crucially, ensure that the pitches fit the voices, i.e., a soprano voice should not be forced to produce bass pitches.

Model II, by contrast, attempts to capture harmony in addition to melody. This model too, however, suffers from ailments, though of a different kind than model I. Its stochasticity, for example, is both a boon and bane in terms of the quality of music produced. A statistical model as mentioned cannot predict music that keeps in lines with the rules of music. To achieve the best sounding composition in this case, would be to have an overfitting model. If every prediction is just pulled from the already seen observations, we can be sure not to break any of these rules and thus produce a completion which sounds very good. Considering this, an improvement would be to predict each in relation to the already predicted states at time t. Thus the pitch for $V1$ $(p1)$ would be predicted as done in our model but the predicted $V2$ pitch $(p2)$ would be such that the the combination of $p1$, $p2$ exists in the dataset. Similarly prediction for $V3$ $(p3)$ is chosen such that the combination $p1, p2, p3$ exists. This way we completely do away with the randomness in the model.

REFERENCES

[1] M. Dirstt and A. S. Weigend, "On completing js bach's last fugue," in *Time series prediction: Forecasting the future and understanding the past*, pp. 151–172, 1993.

[2] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.