

Tom Richmond  
Dr. Mike Heroux  
CSCI 373  
09/29/17

### State of the Field: Classification in Algorithmic Music Composition

Music is unique, in that it is an art form that has very heavy ties with mathematics. The patterns that we see within a piece of music, the repetition, intervals, and timing, are all very deliberate and based upon formal music rules, making the process of composition almost scientific. In fact, there is a whole field of study that focuses on the science behind music, aptly called music science. For this reason, it lends itself rather well to the world of computing, and there is a lot of promise in the field of algorithmic composition.

The topic saw its early explorations back in the mid 1700's with a game called *Musikalische Würfelspiel*<sup>[11]</sup>, which translates from German to 'musical dice game'. The game's most popular iteration, allegedly devised by Mozart himself, saw the user roll a pair of dice, and their composition would proceed based on the outcome being mapped to a ruleset Mozart outlined. These early experiments laid the ground work for algorithmic music to come.

It wasn't until much later in the 1950's that the concepts were brought into the world of computing. The most famous example from this time is Hiller and Isaacson's *Illiad Suite*<sup>[16]</sup>, which used rule systems and Markov chains, a stochastic predictive system with no memory, that predicts the next note based solely on the current note. As the work was expanded upon by colleagues and interested parties<sup>[11]</sup>, the chains were designed to implement an  $n$ th-order technique, which allows the process to consider the last  $n$  notes, rather than only the most recent. This initial work with Markov chains became the genesis of computerized algorithmic compositions.

For hundreds of years, western music has been represented by means of musical score, a visual structure made up of many components, most notably measures and notes with duration values. This has been relatively unchanged because it is the perfect notation for a musician to read and perform. With the advent of computer music however, the necessity for a new representation of written music was realized. This is due to the complex nature of a

written musical score, and the difficulty of teaching a computer to parse through the sheet music and retrieve the data necessary for processing. As a result, the computer science community was met with the challenge of creating a new representation of music that could be more easily processed for the studies to come. Though many were proposed, two have risen above the others in the world of research, MIDI and \*\*kern musical files. Both have their own unique advantages and disadvantages.

MIDI is a very widely used file type<sup>[1][2][3][4][5][6]</sup>. Its popularity can be attributed to its wide variety of uses, including applications outside of the research field. The format itself offers the option of audio playback, so that the user can perceive the piece aurally, which is beneficial for composition analysis. The amount of data available is vast due to its popularity, however not all MIDIs are created equal, as the files can be both hand-encoded and automatically encoded by playing a piece using a MIDI keyboard. This reduces precision due to the higher opportunity for human error, specifically in note duration. Many toolkits are available to extract information for the files, including packages such as jSymbolic, however the extractions are limited to note values, durations, and sequences.

An alternative to MIDI, \*\*kern files have also been used in a number of studies<sup>[5][7][8]</sup>, offering a wide range of advantages. First, \*\*kern files are a textual representation of a musical score, offering the user the ability to parse its information using pattern matching in Linux command lines. This makes the feature extraction process very powerful, and given the high number of features that \*\*kern files are able to represent (everything down to the direction of a note stem), there are a vast number of possibilities. For more common feature extractions, such as notes, intervals and durations, a toolkit known as Humdrum can be used, offering a wide variety of predefined feature extractions designed specifically for \*\*kern files, as well as other file types under the Humdrum syntax. The file type is also very precise, as it can only be encoded by hand. The largest drawback of \*\*kern file use is the sparse availability of scores, as a result of both its hand encoded nature, as well as its lack of utility outside of the field of computer music research.

When it comes to data mining in music, classification has overwhelmingly been the most popular method. Defined as “the task of assigning objects to one of several predefined

categories”<sup>[15]</sup>, the objective within music classification has been to classify a certain piece of music into one of several predefined categories. Studies have been done in an attempt to classify many different types of music into a plethora of categories. Classifying popular music into genre<sup>[4][9]</sup> or decade have both been attempted, aiming to automate the music business further. Other research has focused its scope on classical music, choosing to classify pieces by classes such as composer<sup>[1][2][8]</sup> or era<sup>[3]</sup>. Interestingly, there have even been studies attempting to classify music into classes based on the emotion they make you feel.

In order for the classifier to know which category or class to group a specific item, it requires a list of features or attributes on which to base its decision. With the complex nature of music, there are a wide variety of directions that can and have been explored. This is where the concept of file type becomes very important, because a researcher looking to classify a musical score needs to be able to extract from the file enough musical properties to use as attributes. Toolkits such as jSymbolic and Humdrum become pivotal, as the range of their extraction abilities directly affects which attributes you can use in classification. Features such as pitch, duration, intervals, tempo and key divergence have all been used in studies attempting to classify music<sup>[1][5][7]</sup>.

Classification is a very broad word, used to describe a concept rather than an actual implementation of the idea. There are many models that can be used in an attempt to classify a set of data, each with different merit. Over the course of music classification history, researchers have attempted to classify music using many such methods. Two of the most successful implementations are called Support Vector Machines<sup>[1][2][4][5][7]</sup> and Logistic Regression<sup>[5]</sup>. These classifiers have returned statistically strong results across the board, showing results as strong as 95% classification precision. These results are considered excellent, and are on their own very significant. One drawback to using these classifiers is the complex output results produced by the classifiers. Because of the complex nature of the algorithms, the classifiers output is essentially unusable for further computation.

There are several other classifiers that have been used in an attempt to remedy this problem. Some of these classifiers include decision tree classifiers<sup>[10]</sup>, rule based classifiers<sup>[3][5]</sup> and Naïve Bayes<sup>[1][2][3][4][7][8]</sup>. While the performance of these classifiers have not been as

statistically strong as the above mentioned classifiers, the drop off is not severe, with results hovering between 70-85%, and the output is much more friendly. This allows the researchers to proceed with the information gained, and utilize it for future work. This is especially significant when looking at these classification techniques as a gateway to algorithmic composition, as the information can be utilized much more easily with these latter methods of classification.

On the music composition side of things, many techniques have been explored in an attempt to generate music as well. The Markov chains model mentioned previously has been examined over the years and expanded upon<sup>[11][12][13]</sup>, but many other methods have emerged since. The use of Cellular Automata<sup>[11][14]</sup>, a concept proposed by John Von Neumann in the 1960s based upon the cellular replication process, has been explored in an attempt to replicate music's pattern persistence. Formal grammars<sup>[6][11][17]</sup> have also been a popular method of replicating music composition, based upon the observation that the formal rules of music follow similar syntax tendencies as a language might. Artificial Neural Networks (ANNs)<sup>[11][14]</sup> have become increasingly popular in the field of artificial intelligence, and the principle has been applied to the field of music composition as well. The concept starts with x number of input nodes and y number of output nodes. The system then passes the inputs through a hidden node layer, which attempts to predict the output results. Through a high level of repetition, and comparing the predicted result to the true output, the computer gradually learns to more accurately predict the output. All three of these generation techniques have been explored to relative success, however each also has its own short comings.

To demonstrate the potential of an algorithmic composition process based upon classification, a system is being devised with the following premise: A large number of \*\*.kern files are parsed using Linux command lines and the Humdrum toolkit, extracting the frequency with which each interval class occurs (from unison intervals to octave intervals). These frequencies are then used as the attributes that a chosen classifier uses to derive a class. The classifier splits our \*\*.kern files into 6 distinct classes, one for each era of classical music (Medieval, Renaissance, Baroque, Classical, Romantic, Modern). After analyzing results and outputs of many classification methods, the classifier model chosen to proceed with the

composition process was a Naïve Bayes approach, which had upper tier accuracy (82% ROC), as well as a relatively easy to understand output, which is friendly to the generation process.

These results are then merged with a cellular-automata-inspired system, using 0 and 1 states to represent a four-bit binary number, which is then mapped to a certain note value (0001 is C, 0010 is C#/D-). With rules derived from the classifiers results, a new sequence is generated based on the previous note value and probability of the next note's sequence.

The future is bright within the field of algorithmic composition, and the secret may lie in hybridization. Many of the methods discussed here excel in replicating certain aspects of music, yet fall short in others. That is because there are simply so many features within music that need to be focused on that one generative method cannot account for all of them. However, combining generative techniques which make up for the short comings of each other has the potential to compose much more well-rounded pieces, and will move the computer music field closer to the overall goal of replicating music composition algorithmically.

## Works Cited

1. J. Lan and A. Saied, "Music Classification by Composer," *Book Music Classification by Composer*, Series Music Classification by Composer, ed., Editor ed.^eds., 2012, pp.
2. D. Chiu, et al., "Bartok: Music Time Period Classification," *Book Bartok: Music Time Period Classification*, Series Bartok: Music Time Period Classification, ed., Editor ed.^eds., pp.
3. R. Basili, et al., "Classification of Musical Genre: A Machine Learning Approach," *Book Classification of Musical Genre: A Machine Learning Approach*, Series Classification of Musical Genre: A Machine Learning Approach, ed., Editor ed.^eds., pp.
4. W. Herlands, et al., "A Machine Learning Approach to Musically Meaningful Homogeneous Style Classification," *Book A Machine Learning Approach to Musically Meaningful Homogeneous Style Classification*, Series A Machine Learning Approach to Musically Meaningful Homogeneous Style Classification, ed., Editor ed.^eds., AAAI Conference on Artificial Intelligence, 2014, pp.
5. D. Herremans, et al., "Classification and Generation of Composer-Specific Music Using Global Feature Models and Variable Neighborhood Search," *Computer Music Journal*, vol. 39, no. 3, 2015, pp. 71-91; DOI 10.1162/COMJ\_a\_00316.
6. M. Alfonseca, et al., "A Simple Genetic Algorithm for Music Generation by means of Algorithmic Information Theory," *Book A Simple Genetic Algorithm for Music Generation by means of Algorithmic Information Theory*, Series A Simple Genetic Algorithm for Music Generation by means of Algorithmic Information Theory, ed., Editor ed.^eds., 2007 IEEE Congress on Evolutionary Computation, 2007, pp.
7. J. Lebar, et al., "Classifying Musical Scores by Composer: A machine learning approach," *Book Classifying Musical Scores by Composer: A machine learning approach*, Series Classifying Musical Scores by Composer: A machine learning approach, ed., Editor ed.^eds., 2008, pp.
8. L. Mearns, et al., "Characterisation of composer style using high-level musical features," *Book Characterisation of composer style using high-level musical features*, Series Characterisation of composer style using high-level musical features, ed., Editor ed.^eds., ACM, 2010, pp. 37-40.
9. C. McKay and I. Fujinaga, "Automatic Genre Classification Using Large High-Level Musical Feature Sets," *Book Automatic Genre Classification Using Large High-Level Musical Feature Sets*,

Series Automatic Genre Classification Using Large High-Level Musical Feature Sets, ed., Editor ed.^eds., Universitat Pompeu Fabra, 2004, pp.

10. R. Basili, et al., "Classification of Musical Genre: A Machine Learning Approach," 2004.

11. J.D. Fernandez and F. Vico, "AI Methods in Algorithmic Composition: A Comprehensive Survey," *Journal of Artificial Intelligence Research*, vol. 48, 2013, pp. 513-582.

12. M. Edwards, "Algorithmic Composition: Computational Thinking in Music," *Communications of the Acm*, vol. 54, no. 7, 2011, pp. 58-67; DOI 10.1145/1965724.1965742.

13. T. Collins, et al., "Developing and evaluating computational models of musical style," *Ai Edam-Artificial Intelligence for Engineering Design Analysis and Manufacturing*, vol. 30, no. 1, 2016, pp. 16-43; DOI 10.1017/s0890060414000687.

14. I.D. Matic, et al., "Automatic Melody Generation using Neural Networks and Cellular Automata," *Eleventh Symposium on Neural Network Applications in Electrical Engineering (Neurel 2012)*, 2012.

15. P. Tan, et al. *An Introduction to Data Mining*. Pearson Nueva Delhi (India). 2016.

16. L. Hiller, et al. *Illiac Suite*. Score, Theodore Presser Co, New York, USA. 1957.

17. D. Quick, et al. *Grammar Based Automated Music Composition in Haskell*. Boston, Massachusetts, USA. 2013.