**Survey of Music Classification and Algorithmic Composition through Data Mining**

**Tom Donald Richmond** • [tdrichmond@csbsju.edu](mailto:tdrichmond@csbsju.edu)

1. **Introduction**

Traditionally, creative arts and computer science have had a hard time finding points of intersection. So much of what makes creative art unique revolves around the human, emotional connection to the artistic work, and it has been argued that computers simply rely too much on algorithm to successfully duplicate this in many cases. However, the exception to this rule seems to be in music. The world of music, and especially classical music, is so well grounded in the world of mathematics that the use of technology in this realm has actually yielded some impressive, significant and encouraging results. So much so that computer music is a field of great interest at the moment. One of the many technological fields that has been applied to music in recent years is the field of data mining and machine learning.

* 1. **Data Mining**

The utilities of data mining seem to be almost infinite. In fact, it seems as though this highly versatile set of techniques are applicable to almost every field of study, and the world of music is certainly no exception. Data mining is an intentionally broad term, of course, because it can refer to any of a large number of techniques to extract useful patterns from seemingly incomprehensibly large amounts of data. Over the years, many of these techniques have been used to analyze the patterns within music for various reasons. Some of the techniques that have been applied to the study of music, which will be covered later in this report, include various forms of classification -- a supervised machine learning technique. Methods such as SVMs (Support Vector Machines), Logistic Regression Models, decision trees, as well as countless others all fit under the umbrella term of ‘classification’. The information gathered from the mining process is then used for a wide variety of reasons, including the algorithmic composition of musical pieces.

* 1. **Data**

Naturally, when studying the patterns in large sets of data, the first and most important thing to look at is the data itself. Music has a wide variety of representations, and this makes the amount of potential data to be studied vast. The two most import representations of music are the written, structured score, and the performed piece. While any musician will tell you that the performance of a piece adds another, emotionally driven level to the music, much of what the performance adds to the piece is hard to quantify. While some studies have chosen to analyze actual sound waves to find patterns within their music, for simplicity’s sake, this paper will focus on a more structured data set concerning score representation. Furthermore, when dealing with classical music, there is much more importance placed on the written representation of the piece than there is when looking at a piece of popular music.

* + 1. **Data Format**

Researchers working on this topic have addressed the issue of data representation in many ways, but there have been two formats that seem to be the most popular in terms of score representation. Those two file types include MIDI files, and \*\*kern text files. MIDI files seem to have the largest spread of usage throughout the research that has been done up to this point, likely because it offers a larger sample of data files to be mined. Files of the \*\*kern type however, have been used to a high degree of success as well. Both file types have open source toolkits to help process and analyze the data presented in the file, and both file types offer unique features that are useful to the processing of its information.

The most widely used file type in the studies surveyed was the MIDI file. It has been used in studies by a large number of authors ([1], [2], [3], [4], [5], [6]). MIDI files contain representations of the musical score that is often recorded from humans playing the score, though you can also find hand compiled MIDI representations. When hand encoded, these files serve as a kind of hybrid between performed music and written music, as it is a strict representation of how a musical piece would be performed without any human error or embellishment. For this reason, it provides much more quantifiable information than a traditionally performed audio recording could, and it provides the exact representation that the composer suggested in their score.

The file type \*\*kern is a representation scheme that conforms to a broader syntax known as “Humdrum”. This database was specifically designed with the idea of computational analysis in mind. Researchers Lebar, Chang & Yu [7] used this format when attempting to classify musical scores by composer, while researchers Mearns, Tidar and Dixon [8] used this file type to characterize composers into different classical styles. Researchers Herremans, Soerensen & Martens [5] acknowledged the datatype, and its benefits, but ultimately chose instead to hand encoded MIDI files, which they based on \*\*kern files, because of their compatibility with a feature extraction software titled jSymbolic. Because \*\*kern files are coded by hand, they provide reliable data that isn’t always guaranteed in MIDI type files.

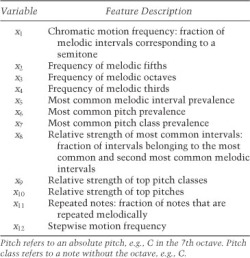
* 1. **Features**

Naturally, when classifying a set of data, you must identify a number of properties or features that the classifier will use to determine which class a specific piece of data belongs to. With music, there are so many properties and variations that it is hard to decide exactly what aspects of the music make it distinct from the others.

With so many potential variables, it is very easy to fall into the trap of overfitting, a phenomenon in datamining that occurs when a model is excessively complex, often times as a result of having too many parameters relative to the number of observations. As a consequence, overfitting can cause a model to have poor predictive performance, because it overreacts to minor fluctuations in the provided training set.

Over the course of the studies that were examined in this paper, there were many proposed properties to analyze. Lan [1] tried extracting many different quantifiable features from their MIDI files using a Matlab package called miditoolbox, as well as a Python package called music21. The features they decided on by the end of their extraction process was the mean and standard deviation of pitches, note durations, and inter-onset interval. They later included pitch intervals, song tempo, pitch range, and the divergence from the original key.

Lebar [7] chose to focus their feature extraction solely on the pitches and their durations in their attempt to classify musical scores by composer. They acknowledged that their models were context free, and further work could include giving consideration to the surrounding measures.



Figure

Herremans and company [5] focused on 12 different features within their musical scores. They cited one of their reasons, among others, for choosing these specific features as to avoid overfitting. They also mentioned that one-dimensional features were favored because of their normalized nature and their ease of handling. A list of their features can be found in *Figure 1*.

With so many potential features within a musical score to be accounted for, there are no limits on this aspect of data mining. There are an infinite number of feature combinations that have yet to be tried, and it should be an interesting challenge to optimize the feature extraction process.

1. **Classification**

Classification has overwhelmingly been the most popular method of data mining used to classify music categorically, no matter what the categories may be. There have been many different forms of classification used in this attempt, some proving more popular than others. Before the more commonly used methods are discussed, it is important to touch on the methods that haven’t been as widely explored.

A fair few studies have taken the k-Nearest Neighbor (kNN) approach to classification in their efforts of classifying music ([1], [2], [7], [9]), and while it yielded admirable results for all of the researchers mentioned, almost all of these studies tried multiple approaches and opted to continue pursuing their objectives with an alternative classification technique.

Another approach that has commonly been used was that of Naïve Bayes ([1], [2], [3], [4], [7], [8]) or some variation of it. These methodologies saw drastically different results from different researchers, with Lan and company initially getting only 50% accuracy [1], and Chiu and company receiving accuracy of just over 50% [2]. They made the conclusion that this was because the model makes the assumption that features are independent, and it is therefore a poor model for music. Alternately, Basili [3], Herlands [4], and Lebar [7] all found admirable results with the Naïve Bayes approach to varying degrees.

Another method that was implemented to a large degree of success was Logistic Regression (LR), which was the primary method of classification used by Herremans and company [5]. Interestingly enough, it was the only classifier that was used with the motive of generating music based on data gathered, and thus applies most accurately to the overall subject of this survey.

Other methodologies worth mentioning include Adaptive Boosting (AdaBoost), which was explored by Chiu [2] and Herlands [4], decision trees such as J48 [10], and RIPPER, a rule-based classifiers that implements a propositional rule learner ([3], [5]).

* 1. **Support Vector Machines**

The use of Support Vector Machines (SVMs) have been implemented in the attempt to classify music categorically perhaps more than any other method. The sheer amount of research and success with this topic merits its own subsection. There have been many instances of this throughout recent years, and it has been met with a high degree of success. Lan and Saied [1] implemented a SVM model (among other models) in their attempts to classify classical music by composer based on MIDI files, while Lebar, Chang, and Yu [7] did the same, with the same objective of classification by composer, but used the \*\*kern format. They found the SVM to be the most consistently accurate classifier they attempted. Chiu, Liu, and Wang [2] used SVM to classify classical music based on musical time period using MIDI files, and like the previously discussed attempts, they found SVM to be the most effective method to classify classical music accurately. Herlands, Der, Greenberg, and Levin [4] found that SVM classifiers were overwhelmingly the most successful approach to classifying homogenous music, and credited this finding to the fact that there was a high dimensionality in the feature space.

It is clear to see that most researchers found encouraging results when using a SVM classifier, however it was not without its problems. Herremans and his team [5] chose to continue their composer-specific generative endeavors using a logistic regression model instead of the SVM model, despite the SVM model yielding more consistently accurate results. They explained that they chose to do so because the SVM model is not very easy to comprehend, and the LR model provided a clearly defined probability for each composer. For this reason, it is important to also note the context under which researchers are attempting to classify their music, and whether each model is applicable to their end goal, as the success rate of a classifier does not guarantee its usefulness in future endeavors.

1. **Algorithmic Composition**

Of course, using data mining to classify music is only one piece of this puzzle, with the other being the actual automatic generation of similar music based on the findings from our data mining. There have been many approaches toward the algorithmic composition of music in the past, to varying degrees of success. While some of these are not strictly based around data mining and machine learning, they have potential intersection points with the world of data mining that makes it worthwhile to study them.

A survey done by Jose David Fernandez and Francisco Vico [11] in 2013 attempted to provide a comprehensive summary of the work done in algorithmic compositions, and this paper will draw heavily on that survey.

* 1. **Grammars**

One very popular method of creating compositions using a computer revolves around the idea of developing a formal grammar which drives the generative process. Past studies have used both manually derived grammatical rules (which serve no interest to the datamining aspect of this research), as well as automatically generated grammar rules [11]. Interestingly, Alfonseca, Cebrian and Ortega [6] not only derived their own grammar, but they chose also to create their own language to represent the music data (rather than using a MIDI or \*\*kern formatted file). This is a field that could be explored to a fuller extent, or combined with other methods to enhance the comprehensiveness of a piece, and help it fit formal music theory rules more accurately.

* 1. **Markov Chains**

Another widely used method to generate music is with a Markovian Chain. Simply put, this method is a stochastic process which transits between a set of different states as time goes on, without a memory [11]. This allows the music to evolve over the course of a piece.

Fernandez and Vico [11] discuss in their survey the use of Markov chains in music composition. Often the Markovian chains were extended to consider n-th order, meaning the next state depended on the last n states, not just the most recent one. This addition was important to include, as patterns and repetition are very important in the world of classical music, and a Markov Chain with no memory would not take into account previous parts of the piece. They note that the application of Markov chains in music composition was quite popular early in the history of algorithmic composition, however its limitations became apparent quickly. They cited these limitations to be a lack of coherent musicality, and stated that these chains are more useful as a source of raw material, rather than a fully composed work. Additionally, Edwards [12] notes that Markov models tend to need a significant amount of data to be trained adequately.

Collins, Laney, Willis and Garthwaite [13] studied the use of Markov models in stylistic composition, with the goal of creating music in the style of composer Frederic Chopin. They tested two versions of their model against a benchmark model called the Experiments in Musical Intelligence (EMI), as well as stylistically comprised human compositions, and found what they described as promising results. Through a series of tests in which they had judges, both concertgoers and experts, categorize the composed music, they found that their system often produced results that could be considered stylistically on par with human-composed music. The conductors of the study went on to say that the results suggest “some aspects of musical style are being modeled well”, and that the open ended responses by the judges suggest that it is phrasing and rhythm that are accomplished particularly well with these models.

* 1. **Cellular Automata**

Another way of generating a melody is through the use of Cellular Automata (CA). A cellular automaton (singular) is a discrete model implemented in many fields of study, including computability theory, mathematics, physics, complexity science, theoretical biology, and microstructure modeling [14]. It is comprised of a grid of cells, each of which contain one of a finite set of states. The state of each cell is determined by the state of neighboring cells at any singular moment. While CA does not strictly speaking relate to the idea of machine learning as a result, a lot of studies have attempted to use cellular automata in the efforts of automatic music composition.

Matic, Oliveira, and Cardoso [14] used CA in their attempts to automatically generate music of a specific emotional content. They used three different states: quiescent, depolarized, and burned. The researchers determined that this approach to automatic composition offers a wide range of relations between its cells, resulting in more possible solutions of melodies’ qualities. Fernandez and Vico [11] highlighted many such attempts in their survey, and they conclude that CA should be regarded as a source of inspiration rather than as a proper way to automate the composition of music. However, they feel that the combination of this generative method with other methods holds potential.

* 1. **Neural Networks**

The use of Artificial Neural Networks (ANNs) in an attempt to generate musical compositions has also been widely used. According to Fernandez and Vico [11], ANNs are “computational models inspired in biological neural networks, consisting of interconnected sets of artificial neurons”.

ANNs have been used by many researchers in an attempt to generate musical compositions. Matic, Oliveira, and Cardoso [14] employed this method while creating a program which attempted to generate music with a specific emotional content. Their idea was to train the ANN using a set of melodies, and have the ANN generate new melodies which preserved the stylistic characteristics of the original set. They chose to implement a Recurrent Neural Network (RNN), which stores a number of previous events, to maintain a cyclical characteristic that is important to music. The researchers found that, when given an adequate corpus, the use of ANNs in music generation is promising. Fernandez and Vico also documented many cases of ANNs being used to algorithmically generate music in their survey. They noted that hybrid systems that combined ANNs with evolutionary algorithms quickly became the most popular approach. In their conclusion, they dictated that this technique, as well as the aforementioned Markov chains method, are used primarily for imitation.

1. **Conclusion**

Overall, there has been a wide variety of studies over the past several years having to do with music classification by means of datamining, as well as algorithmic composition. Despite this, there isn’t a distinct, best available option as of now. Each method has been found to have many flaws, and is far from refined, especially in regards to automatic composition. Nonetheless, the information gained from these previous studies in invaluable going forward, as it serves as a template of what has potential, and what falls short in terms of effectiveness.

* 1. **Classification**

Classification technology has a distinct head start on composition technology, and many avenues have been explored as a result. While there certainly isn’t a method that can be universally praised, or deemed perfect, there have been an exceptional number of studies that find acceptably accurate results.

Out of all the methods analyzed, Support Vector Machines seem to have the largest amount of individual success, however it has been found in other studies [5] that using the data gathered from it in future academic endeavors is not as easy as one would hope, because the algorithm does not allow for easy information processing.

With all of this in mind, it seems that the field of musical classification by means of datamining is a well-covered topic, and while there is certainly much more that can be done to improve the field, it is the hybridization of this classification with other fields, such as algorithmic composition, that yields the greatest potential.

* 1. **Composition**

This field is still pretty vastly unexplored, and the potential of further research is enormous. The overwhelming conclusion that has been gathered from research on algorithmic composition is that it is not enough to just rely solely on the generative methods that have been tested thus far. As Fernandez and Vico [11] pointed out, there is a lot of potential that hybridization of multiple methods can help improve the overall world of algorithmic composition.

Moreover, algorithmic composition up to this point has not exactly been completely computer generated. All of the methods explored in this study required some sort of human interaction to derive the rules that the generative process followed. This is where its potential intersection with data mining is exciting.

Not a lot of intersection has taken place to this point, and the possibilities are endless. Many previous studies have attempted to teach the computer how to compose music by means other than data mining, such as random generation following a set of rules derived by hand, much like the grammars, chains and cellular automates discussed. If these rules were adapted by a machine learning process that preceded it, perhaps it could gradually learn to become more apt.

* 1. **Future Work**

After this survey, it can be seen that plenty of work has yet to be done in this field, and the hybridization of these two fields of computational musical studies could potentially bring about a vast improvement upon the systems we currently have.

Our future direction seeks to use a training set of data in the \*\*kern format from various eras of classical composition (Medieval, Renaissance, Baroque, Classical, Romantic and Modern) to derive epoch-specific rule sets by means of data mining. As SVMs have had the largest success rate, this method would be employed to ensure accurate results, as well as a more logic based method such as LR or decision tree.

Moving forward, these rule sets would then be used in tandem with a hybridization of different algorithmic composition methods to help produce the most period-accurate results. Previously stated in this survey, many of the generative methods have been deemed useful for ‘inspiration’ of classical music, but not for fully composed classical pieces. This is where the information gathered from the data mining process would be put to work. The intent is to use these newly discovered rules to shape to generative process, combining the inspiration of generation and the logical rules of classification to create a more comprehensible piece of music.

1. **References**

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.