ISEA2015
Proceedings of the 21st International Symposium on Electronic Art
ISSN: 2451-8611
ISBN: 978-1-910172-00-1

Musical Structure Imitation using Segmentation, and k-Nearest Neighbors (kNN)

Evan X. Merz

San Jose State University
San Jose, CA. USA
evan.merz@sjsu.edu

Abstract

Segmenting music is important in academic and commercial settings. Imitating musical structure requires interpretation and generalization of discovered structure. The program shown here is a work in progress that demonstrates an approach to structure imitation using a segmentation algorithm with a look back algorithm based on a probabilistic variant of kNN. A monophonic piece of music is segmented, then kNN is used to generate the structure of a new piece. This work shows that although the problem of structure generation is complex, it is not clear that a solution must be similarly complex.

Keywords

algorithmic music, machine learning, style imitation, segmentation, k nearest neighbors, midi, self similarity, structure, form

Introduction

Discovering the structure of a piece of music is important for academics and commercial music outlets. There are a variety of approaches to the task of melodic segmentation. [1] Still, the structure of music is difficult to quantify because it is a combination of artistic and cultural preferences. It is even more difficult to interpret the structure of a piece and use that information in a generative music algorithm. This requires not only segmenting a piece, but understanding how to create similar segments that can be reconfigured creatively while conforming to the style of the original.

The task of style imitation has been explored by many authors and composers. [2, 3] The work presented here deals with the specific task of structure imitation. This paper presents the prototype of a system for structure imitation using simple machine learning techniques. The data extracted from this preliminary work reveals some interesting insights into the task of structure imitation.

The work presented here is in progress. It does not deal with musical form, which I differentiate from musical structure. Form requires a beginning and an ending while structure only indicates non-random change over time. This work also employs an ad-hoc segmentation technique using a self-similarity matrix. Other segmentation techniques will be inserted into this algorithm as the project continues.

There is a lot of work on melody segmentation. [4] On the specific task of imitating musical structure algorithmically, many composers have published strategies they have used in their own algorithmic pieces. [5, 6]

This work was motivated by a search for a simple algorithm that generates satisfying musical structure in my own algorithmic pieces. In previous experiments, I found that structure could be generated from simple systems. [7] One of the simplest structural algorithms is the look back algorithm. The look back algorithm generates musical structure by repeated previously generated material and optionally transforming it. In this work, I wanted to see if I could implement a probabilistic look back algorithm on a segmented melody to imitate the structure in that melody.

Procedure

This program works in two stages, analysis and generation. In the first stage, the software analyzes a monophonic piece of music in order to build a model of the structure of the piece. In the second stage, that model is used to generate a new piece of music with similar structure.

The analysis stage builds a three tiered model. Each tier contains data at a different structural level. In the lowest tier, each note is an instance containing features such as pitch, start tick in the source midi file, and duration in beats. The next level is the section model. In the section tier, each discovered section is an instance containing features such as note count, mean pitch, and duration in beats. By definition the section tier does not contain a particular type of section. The sections aren't tied to phrases, or sections that a human analyst might discover. The sections are simply designed to hold whatever sections are discovered by the segmenter. The top tier contains pieces, which contain sections. In the current version of the software, only one piece instance is used, but future versions may employ multiple pieces. The piece instance contains features such as pitch mean, pitch mode, duration mean, and duration mode, although these features are not used in the current program.

A significant challenge in the analysis phase is the segmentation of the source midi file. There are many ways to segment a piece of music. [8] The approach used here employs a self-similarity matrix. This technique is loosely based on other self-similarity approaches, but it has been experimentally modified for the current project. [9]

The segmentation algorithm begins by generating a self-similarity matrix based on the distance between notes. The distance between every note is calculated using four features: midi pitch number, pitch name, duration, and start position in measure. In this distance function, one is added to the distance for each non-matching feature. The distance matrix is then normalized and inverted so that zero represents totally dissimilar notes, while one represents maximum similarity. An edge detection algorithm and threshold then reveal the notes where similar sections most commonly begin and end. The threshold is automatically tuned to generate sections that average at least two notes in length.

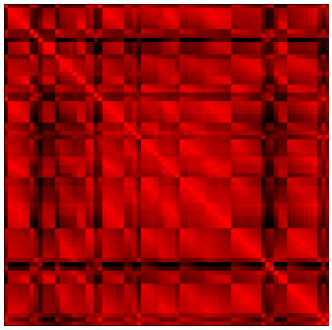

Figure 1. A self-similarity matrix of a short piece of music

The analysis stage ends by calculating the mean, median, mode, and standard deviation on several properties of each section, including the pitch, duration, number of notes, duration, and start position in measure. These are used in the generation section to accept or reject generated sections.

The generation stage relies on a probabilistic version of the k-nearest neighbors (kNN) algorithm that is used for prediction. The kNN algorithm is typically used for classification or regression. In this program, kNN is used to generate new music data. The k nearest neighbors are discovered, then one is chosen randomly. In the trials shown here, k was set to three for both notes and sections. The next instance that is added to the output is whatever comes after the selected neighbor. Random selection is necessary in a procedural content program. If the program always selected the nearest neighbor then it would always generate very similar, repetitive music. The randomness allows it to pick between several options that occurred in the training data.

The probabilistic kNN algorithm is used in two ways. First it is used to pick a model section that determines the features of the section that should come next in the generated piece. Then the section randomly becomes either a look back section, which repeats earlier material, or it is filled with notes picked using the probabilistic kNN algorithm on notes.

Results

The program is not yet optimized to generate the quantity of data necessary to judge its performance in general. In this paper I will simply list the results of small scale test runs. These results may not be predictive of future performance, but the data does shed light on the problem of structure imitation.

In each of these tests I generated three pieces. The piece in column A was generated by randomly selecting notes from the input file. The piece in column B was generated by using the probabilistic kNN algorithm on notes alone, with no attempt to model structure. The piece in column C was generated using the method detailed above.

Each piece was then segmented using the self-similarity algorithm detailed previously. This algorithm only sees the output of the other programs, so it won't necessarily discover the same sections as those that were generated. Then each section was analyzed based on note count, duration, and section start position. This data is intended to show the similarity between the segments in the input music and the segments in the generated music. It's difficult to quantify the structural similarity between two pieces of music. A better method might include human ratings of structural similarity, but that is beyond the means of this research.

In the tables shown here, I chose to look at the results generated from three pieces. The first table shows the results from a run on the first movement of Bach's *Partita for Flute 1*. The second table shows the results from a run on the melody of a folk song called The *Wonderful Crocodile*. The third table shows the results from a run on a short piece of my own composition called *Short Piece 01*. The Bach Partita has very regular phrases that respect measure boundaries. The folk song has slighly less regular phrases. My own piece only rarely bounds phrases based on measures.

First Movement from Partita for Flute 1 by J.S. Bach				
	Source	**A**	**B**	**C**
notes in piece	1024	1024	1024	1047
sections discovered	66	66	64	65
notes per section				
mean	15.5	15.42	15.92	15.98
mode	16	16	16	16
std deviation	2.22	2.12	1.02	0.12
duration (12ths of a beat)				
mean	46.87	47.27	48.0	47.98
mode	48	48	48	48
std deviation	4.87	4.30	4.24	0.12
start position in measure (12ths of a beat)				
mode	0	0	0	0

Table 1. Sections discovered in music based on Partita 1 for Flute by J. S. Bach

The Wonderful Crocodile				
	Source	**A**	**B**	**C**
notes in piece	92	92	92	80
sections discovered	10	10	10	10
notes per section				
mean	9.1	6.6	8.9	6.9
mode	6	3	7	6
std deviation	5.24	3.38	3.75	2.21
duration (12ths of a beat)				
mean	120.0	96.0	120.0	96.0
mode	96	54	96	96
std deviation	53.66	38.23	45.61	30.35
start position in measure (12ths of a beat)				
mode	0	36	0	0

Table 2. Sections discovered in music based on the folk song The Wonderful Crocodile

Short Piece 01 by Evan X. Merz				
	Source	**A**	**B**	**C**
notes in piece	76	76	76	90
sections discovered	13	13	14	17
notes per section				
mean	5.69	5.61	5.28	5.11
mode	2	4	2	2
std deviation	3.53	2.78	3.21	3.21
duration (12ths of a beat)				
mean	48.0	45.0	41.64	42.58
mode	48	32	48	48
std deviation	21.04	24.70	20.55	19.24
start position in measure (12ths of a beat)				
mode	0	1	0	0

Table 3. Sections discovered in music based on Short Piece 01 by Evan X. Merz

Discussion

These data are inconclusive, but they elucidate the problem of generating musical structure in several interesting ways.

In each test, the random music program generated the most chaotic structure, the approach outlined here generated the most regular structure, and the kNN on notes only was somewhere in the middle. This is shown by the standard deviation of the notes per section and duration in twelfths of a beat. The regularity generated by the program outlined here does not necessarily represent the structure that occurs in the source piece, as the discovered sections in the Bach piece are not as regular as the generated ones. This is partially a result of using a note's position in the measure as a feature in the segmentation algorithm.

In the other pieces, this approach clearly generated structure closer to that of the source file than was generated by randomly selecting pitches from the source file. It is not clear that this approach imitated the structure of the source piece any better than the system that generated music using probabilistic kNN on notes alone. This is interesting because it implies that the structure of a piece of music emerges from the notes. This may be obvious from the perspective of a musicologist, but it means that generating musical structure does not absolutely require an approach that models musical structure.

The data tables do not reveal anything about the qualitative experience of listening to the music that was generated with no attempt to model structure versus listening to the music generated by a system that does model structure. The following two images show the structure discovered in Short Piece 01 versus the structure discovered in a piece that was generated by imitating the structure in that piece. The top of each image is the start of the piece. Each line represents the start of a discovered section. These images show the relationship between the internal structure of each piece, with clusters of short sections interspersed with groups of longer sections.

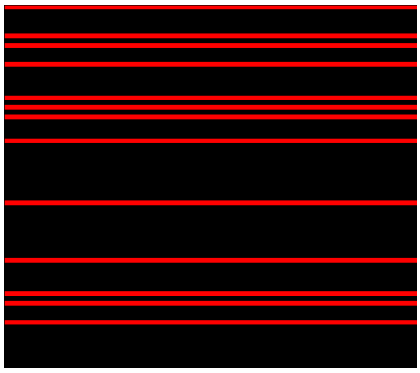

Figure 2. Sections discovered in Short Piece 01

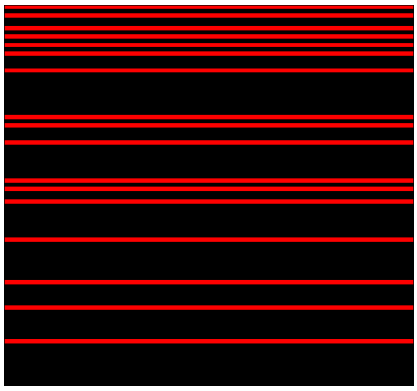

Figure 3. Sections generated using segmentation and kNN

This program is still very limited, and as the data show, it will need significant changes to actually emulate the structure or form of a piece of music. This program does not model beginnings or endings, nor does it have any representation of the large scale flow of sections throughout a piece. It relies on a single piece to infer musical structure, which may be made easier by the use of many pieces. Future versions of this program will compare the results of imitating musical structure using various segmentation algorithms.

Conclusion

The program shown here is a work in progress that uses a machine learning approach to model musical structure. This work shows that although the problem of structure generation is complex, it is not clear that a solution must be similarly complex. Basic structure can be imitated using the look back algorithm and kNN. This work also shows that to some still uncertain degree musical structure can emerge from a model of the note level alone. More work is necessary to determine the general effectiveness of these algorithms, and to disambiguate the idea of musical structure as it relates to musical metacreation.

References

1. Marcus Pearce, Daniel Müllensiefen, and Geraint A. Wiggins, "A Comparison of Statistical and Rule-Based Models of Melodic Segmentation" (paper presented at The Ninth International Conference on Music Information Retrieval, Philadelphia, PA, 2008).
2. David Cope, *Virtual Music: Computer Synthesis of Musical Style (*Cambridge: MIT Press, 2004).
3. Olivier Lartillot, Schlomo Dubnov, Gerard Assayag, and Gill Bejerano, "Automatic Modeling of Musical Style" (paper presented at The International Computer Music Conference, Havana, Cuba, 2001).
4. Pearce, "A Comparison."
5. Cope, *Virtual Music*.
6. Arne Eigenfeldt, "Generating Structure: Towards Large-Scale Format Generation" (paper presented at the 3rd International Workshop on Musical Metacreation, Raleigh, North Carolina, October 4, 2014).
7. Evan Merz, "Composing with All Sound Using the FreeSound and Wordnik APIs" (paper presented at the 1st International Workshop on Musical Metacreation, Boston, Massachusetts, 2013).
8. Pearce, "A Comparison."
9. Jonathan Foote, and Matthew Cooper, "Visualizing Musical Structure and Rhythm via Self-Similarity" (paper presented at The International Computer Music Conference, Havana, Cuba, 2001).

Bibliography

Cambouropoulos, Emilios. "The local boundary detection model (LBDM) and its application in the study of expressive timing." In *Proceedings of the international computer music conference*, pp. 17-22. 2001.

Cope, David. *Virtual Music: Computer Synthesis of Musical Style*. Cambridge: MIT Press, 2004.

Conklin, Darrell, and Christina Anagnostopoulou. "Representation and Discovery of Multiple Viewpoint Patterns." Paper presented at The International Computer Music Conference, Havana, Cuba, 2001.

Ehmann, Andreas F., Mert Bay, J. Stephen Downie, Ichiro Fujinaga, and David De Roure. "Music Structure Segmentation Algorithm Evaluation: Expanding on MIREX 2010 Analyses and Datasets. Paper presented at the 12th International Society for Music Information Retrieval Conference, Miami, Florida, October 24-28, 2011.

Eigenfeldt, Arne. "Generating Structure: Towards Large-Scale Format Generation." Paper presented at the 3rd International Workshop on Musical Metacreation, Raleigh, North Carolina, October 4, 2014.

Ferrand, Miguel, Peter Nelson, and Geraint Wiggins. "A Probabilistic Model for Melody Segmentation." Paper presented at the 2nd International Conference on Music and Artificial Intelligence, 2002.

Foote, Jonathan, and Matthew Cooper. "Visualizing Musical Structure and Rhythm via Self-Similarity." Paper presented at The International Computer Music Conference, Havana, Cuba, 2001.

Lartillot, Olivier, Shlomo Dubnov, Gerard Assayag, and Gill Bejerano. "Automatic Modeling of Musical Style." Paper presented at The International Computer Music Conference, Havana, Cuba, 2001.

Merz, Evan. "Composing with All Sound Using the FreeSound and Wordnik APIs." Paper presented at the 1st International Workshop on Musical Metacreation, Boston, Massachusetts, 2013.

Meudic, Benoit, and Emmanuel St. James. "Automatic Extraction of Approximate Repetitions in Polyphonic Midi Files Based on Perceptive Criteria." In *Computer Music Modeling and Retrieval*, 124-142. Berlin: Springer.

Pearce, Marcus, and Geraint A. Wiggins. "Improved Methods for Statistical Modelling of Monophonic Music." *Journal of New Music Research* 33, no. 4 (2004): 367-385.

Pearce, Marcus, Daniel Müllensiefen, and Geraint A. Wiggins. "A Comparison of Statistical and Rule-Based Models of Melodic Segmentation." Paper presented at The Ninth International Conference on Music Information Retrieval, Philadelphia, PA, 2008.

Paulus, Jouni, Meinard Muller, and Anssi Klapuri. "State of the Art Report: Audio-Based Music Structure Analysis." Paper presented at The International Society for Music Information Retrieval Conference, Utrecht, Netherlands, 2010.

Spevak, Christian, Belinda Thom, and Karin Höthker. "Evaluating melodic segmentation." In *Music and Artificial Intelligence*, pp. 168-182. Springer Berlin Heidelberg, 2002.

Author Biography

Evan X. Merz earned a Master's Degree in electronic music from Northern Illinois University in 2010. He earned a doctorate in algorithmic composition from The University of California at Santa Cruz in 2013. Currently he teaches computer science at San Jose State University.