2014-06-04

Musical Motif Discovery in Non-Musical Media

Daniel S. Johnson
Brigham Young University - Provo

Musical Motif Discovery in Non-Musical Media

Daniel S. Johnson

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Dan Ventura, Chair
Neil Thornock
Michael Jones

Department of Computer Science

Brigham Young University

June 2014

ABSTRACT

Musical Motif Discovery in Non-Musical Media

Daniel S. Johnson
Department of Computer Science, BYU
Master of Science

Many music composition algorithms attempt to compose music in a particular style. The resulting music is often impressive and indistinguishable from the style of the training data, but it tends to lack significant innovation. In an effort to increase innovation in the selection of pitches and rhythms, we present a system that discovers musical motifs by coupling machine learning techniques with an inspirational component. The inspirational component allows for the discovery of musical motifs that are unlikely to be produced by a generative model, while the machine learning component harnesses innovation. Candidate motifs are extracted from non-musical media such as images and audio. Machine learning algorithms select the motifs that best comply with patterns learned from training data. This process is validated by extracting motifs from real music scores, identifying themes in the piece according to a theme database, and measuring the probability of discovering thematic motifs verses non-thematic motifs. We examine the information content of the discovered motifs by comparing the entropy of the discovered motifs, candidate motifs, and training data. We measure innovation by comparing the probability of the training data and the probability of the discovered motifs given the model. We also compare the probabilities of media-inspired motifs with random motifs and find that media inspiration is more efficient than random generation.

Keywords: music composition, machine learning

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Computational music composition is still in its infancy, and while numerous achievements have already been made, many humans still compose better than computers. Current computational approaches tend to favor one of two compositional goals. The first goal is to produce music that mimics the style of the training data. Approaches with this goal tend to 1) learn a model from a set of training examples and 2) probabilistically generate new music based on the learned model. These approaches effectively produce artefacts that mimic classical music literature, but little thought is directed toward expansion and transformation of the music domain. For example, David Cope [7] and Dubnov et al. [8] seek to mimic the style of other composers in their systems. The second goal is to produce music that is radically innovative. These approaches utilize devices such as genetic algorithms [2, 5] and swarms [3]. While these approaches can theoretically expand the music domain, they often have little grounding in a training data set, and their output often receives little acclaim from either music scholars or average listeners. A great deal of work serves one of these two goals, but not both.

While many computational compositions lack either innovation or grounding, great composers from the period of common practice and the early 20th century composed with both goals in mind. For instance, influential classical composers such as Haydn and Mozart developed Sonata form. Beethoven's music pushed classical boundaries into the beginnings of Romanticism. The operas of Wagner bridged the gap between tonality and atonality. Schoenberg's twelve-tone music pushed atonality to a theoretical maximum. Great composers of this period produced highly creative work by extending the boundaries of the musical domain without completely abandoning the common ground of music literature. We must note that some contemporary composers strive to completely reject musico-historical precedent. While this is an admirable cause, we do not share this endeavor. Instead, we seek to compose music that innovates and extends the music of the period of common practice and the early 20th century. While we are aware of the significance of modern and pre-Baroque music, we keep our work manageable and measurable by limiting its scope to a small period of time. After this work is thoroughly examined, we plan to extend this work to include modern and pre-Baroque music.

Where do great composers seek inspiration in order to expand these boundaries in a musical way? They find inspiration from many non-musical realms such as nature, religion, relationships, art, and literature. George Frideric Handel gives inspirational credit to God for his *Messiah.* Olivier Messiaen's compositions mimic birdsong and have roots in theology [4]. Claude Debussy is inspired by nature, which becomes apparent by scanning the titles of his pieces, such as *La mer* [The Ocean], *Jardins sous la pluie* [Gardens in the Rain], and *Les parfums de la nuit* [The Scents of the Night]. Debussy's *Prélude á l'aprés-midi d'un faune* [Prelude to the Afternoon of a Faun] is a direct response to Stéphane Mallarmé's poem, *L'aprés-midi d'un faune* [The Afternoon of a Faun]. Franz Liszt's programme music attempts to tell a story that usually has little to do with music. While it is essential for a composer to be familiar with music literature, it is apparent that inspiration extends to non-musical sources.

We present a computational composition method that serves both of the aforementioned goals rather than only one of them. This method couples machine learning (ML) techniques with an inspirational component, modifying and extending an algorithm introduced by Smith et al. [16]. The ML component maintains grounding in music literature and harnesses innovation by employing the strengths of generative models. It embraces the compositional approach found in the period of common practice and the early 20th century. The inspirational component introduces non-musical ideas and enables innovation beyond the musical training data. The combination of the ML component and the inspirational component allows us to serve both compositional goals. Admittedly, our system in its current state does not profess to compose pieces of music that will enter mainstream repertoire. However, our system contains an essential subset of creative elements that could lead to future systems that significantly contribute to musical literature.

1.1 Musical Motifs

We focus on the composition of motifs, the atomic level of musical structure. We use White's definition of motif, which is "the smallest structural unit possessing thematic identity" [19]. There are two reasons for focusing on the motif. First, it is the simplest element for modeling musical structure, and we agree with Cardoso et al. [6] that success is more likely to be achieved when we start small. Second, it is a natural starting place to achieve global structure based on variations and manipulations of the same motif throughout a composition.

Since it is beyond the scope of this research to build a full composition system, we present a motif composer that performs the first compositional step. The motif composer trains an ML model with music files, it discovers candidate motifs from non-musical media, and it returns the motifs that are the most probable according to the ML model built from the training music files. It will be left to future work to combine these motifs into a full composition.

Chapter 2

Related Work

A variety of machine learning models have been applied to music composition. Many of these models successfully reproduce credible music in a genre, while others produce music that is radically innovative. Since the innovative component of our algorithm is different than the innovative components of many other algorithms, we only review the composition algorithms that effectively mimic musical style.

Cope extracts musical signatures, or common patterns, from the works of a composer. These signatures are recombined into a new composition in the same style [7]. This process effectively replicates the styles of composers, but its novelty is limited to the recombination of already existing signatures. Aside from Cope's work, the remaining relevant literature is divisible into two categories: Markov models and neural networks.

2.1 Markov Models

Markov models are perhaps the most obvious choice for representing and generating sequential data such as melodies. The Markov assumption allows for inference and learning to be performed simply and quickly on large data sets. However, low-order Markov processes do not store enough information to represent longer musical contexts, while higher-order Markov processes can require intractable space and time.

This issue necessitates a variable order Markov model (VMM) in which variable length contexts are stored. Dubnov et al. implement a VMM for modeling music using a prediction suffix tree (PST) [8]. A longer context is only stored in the PST when 1) it appears frequently in the data and 2) it differs by a significant factor from similar shorter contexts. This allows the model to remain tractable without losing significant longer contextual dependencies. Begleiter et al. compare results for several variable order Markov models (VMMs), including the PST [1]. Their experiments show that Context Tree Weighting (CTW) minimizes log-loss on music prediction tasks better than the PST (and all other VMMs in this experiment). Spiliopoulou and Storkey propose the Variable-gram Topic model for modeling melodies, which employs a Dirichlet-VMM and is also shown to improve upon other VMMs [17].

Variable order Markov models are not the only extensions explored. Lavrenko and Pickens apply Markov random fields to polyphonic music [13]. In these models, next-note prediction accuracies improve when compared to a traditional high-order Markov chain. Weiland et al. apply hierarchical hidden Markov models (HHMMs) separately to pitches and rhythms in order to capture long-term dependencies in music [18].

Markov models generate impressive results, but the emissions rely entirely on the training data and a stochastic component. This results in a probabilistic walk through the training space without introducing any actual novelty or inspiration beyond perturbation of the training data.

2.2 Neural Networks

Recurrent neural networks (RNNs) are also effective for learning musical structure. However, similar to Markov models, RNNs still struggle to represent long-term dependencies and global structure due to the vanishing gradient problem [12]. Eck and Schmidhuber address the vanishing gradient problem for music composition by applying long short-term memory (LSTM). Chords and melodies are learned using this approach, and realistic jazz music is produced [9, 10]. Smith and Garnett explore different approaches for modeling long-term structure using hierarchical adaptive resonance theory neural networks. Using three hierarchical levels, they demonstrate success in capturing medium-level musical structures [15].

Like Markov models, neural networks can effectively capture both long-term and short-term statistical regularities in music. This allows for music composition in any genre given sufficient training data. However, few (if any) researchers have incorporated inspiration in neural network composition prior to Smith et al. [16]. Thus, we propose a novel technique to address this deficiency. Traditional ML methods can be coupled with sources of inspiration in order to discover novel motifs that originate outside of the training space. ML models can judge the quality of potential motifs according to learned rules.

Chapter 3

Methodology

An ML algorithm is employed to learn a model from a set of music themes. Pitch detection is performed on a non-musical audio file, and a list of candidate motifs is saved. (If the audio file contains semantic content such as spoken words, we defer speech recognition and semantic analysis to future work.) The candidate motifs that are most probable according to the ML model are selected. This process is tested using six ML models over various audio input files. A high-level system pipeline is shown graphically in Figure 3.1.

In order to generalize the concept of motif discovery from non-musical media, we also extend our algorithm to accept images as inputs. With images, we replace pitch detection with edge detection, and we iterate using a spiral pattern through the image in order to collect notes. This process is further explained in its own subsection. All audio and image inputs are listed in Appendix D.

The training data for this experiment are 9824 monophonic MIDI themes retrieved from The Electronic Dictionary of Musical Themes.[1] The training data consists of themes rather than motifs. We make this decision due to the absence of a good motif data set. An assumption is made that a motif follows the same general rules as a theme, except it is shorter. In order to better learn statistical regularities from the data set, themes are discarded if they contain at least one pitch interval greater than a major ninth. This results in a final training data set with 9383 musical themes.

[1] http://www.multimedialibrary.com/barlow/all_barlow.asp

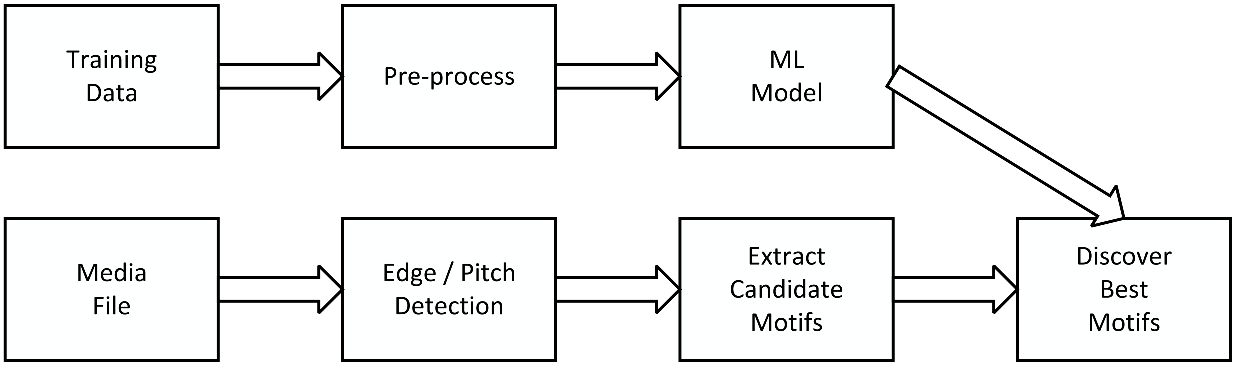


Figure 3.1: A high-level system pipeline for motif discovery. An ML model is trained on pre-processed music themes. Pitch detection is performed on an audio file or edge detection is performed on an image file in order to extract a sequence of notes. The sequence of notes is segmented into a set of candidate motifs, and only the most probable motifs according to the ML model are selected.

3.1 Machine Learning Models

A total of six ML models are tested. These include four VMMs, an LSTM RNN, and an HMM. These models are chosen because they are general, they represent a variety of approaches, and their performance on music data has already been shown to be successful. The four VMMs include Prediction by Partial Match, Context Tree Weighting, Probabilistic Suffix Trees, and an improved Lempel-Ziv algorithm named LZ-MS. Begleiter et al. provide an implementation for each of these VMMs,[2] an LSTM found on Github is used,[3] and the HMM implementation is found in the Jahmm library.[4]

Each of the ML models learns pitches and rhythms separately. Each pitch model contains 128 possible pitches, where 1-127 represent the corresponding MIDI pitches and 0 represents the absence of pitch (a rest). Each rhythm model contains 32 possible rhythms which represent each multiple of a 32nd note up to a whole note.

In the RNN pitch model, there are 128 inputs and 128 outputs. To train the model, we repeatedly choose a random theme from the training data and iterate through each note.

[2] http://www.cs.technion.ac.il/~ronbeg/vmm/code_index.html

[3] https://github.com/evolvingstuff/SimpleLSTM

[4] http://www.run.montefiore.ulg.ac.be/~francois/software/jahmm/

VMM Model	D	M	S	P_{min}	α	γ	r
CTW-pitches	2						
CTW-rhythms	5						
LZMS-pitches		2	4				
LZMS-rhythms		2	2				
PPM-pitches	3						
PPM-rhythms	3						
PST-pitches	4			.01	0	.005	1.05
PST-rhythms	5			.001	0	.001	1.05

Table 3.1: Parameters chosen for each variable-order Markov model. These were manually chosen after performing preliminary tests on a validation set.

For each note, the input for the RNN is a set of zeros except for a 1 where the pitch value for that note is found. The output is the same as the input, except it represents the next note in the sequence. The RNN rhythm model is the same as the RNN pitch model, except there are only 32 inputs and 32 outputs. After training, each RNN becomes a next-note predictor. When an RNN is given an input vector of notes at a given time step, the highest activation values in the RNN's output are used to choose an output vector of notes for the following time step.

The HMM pitch and rhythm models are standard HMMs with 128 and 32 discrete emissions, respectively. Each is initialized with a standard Dirichlet distribution and trained using the Baum-Welch algorithm. The HMM pitch model employs 8 hidden states, and the HMM rhythm model employs 5 hidden states. These values were manually chosen after analyzing results on a validation set. Similarly, each of the VMM pitch and rhythm models have 128 and 32 discrete alphabet members, respectively. The VMMs are trained according to the algorithms presented by Begleiter et al. [1], and the parameters for each model are shown in Table 3.1. Please refer to Begleiter et al. [1] for a description of each parameter.

3.2 Audio Pitch Detection

Our system accepts an audio file as input. Pitch detection is performed on the audio file using an open source command line utility called Aubio.[5] Aubio combines note onset detection and pitch detection in order to output a string of notes, in which each note is comprised of a pitch and duration. The string of detected notes is processed in order to make the sequence more manageable: the string of notes is rhythmically quantized to a 32nd note grid; pitches are restricted between midi note numbers 55 through 85 by adding or subtracting octaves until each pitch is in range.

3.3 Image Edge Detection

Images are also used as inspirational inputs for the motif discovery system. We perform edge detection on an image using a Canny edge detector implementation,[6] which returns a new image comprised of black and white pixels. The white pixels (0 value) represent detected edges, and the black pixels (255 value) represent non-edges. We also convert the original image to a greyscale image and divide each pixel value by two, which changes the range from [0, 255] to [0, 127]. We simultaneously iterate through the edge-detected image and the greyscale image one pixel at a time using a spiral pattern starting from the outside and working inward. For each sequence of b contiguous black pixels (delimited by white pixels) in the edge-detected image, we create one note. The pitch of the note is the average intensity of the corresponding b pixels in the greyscale image, and the rhythm of the note is proportional to b.

3.4 Motif Discovery

After the string of notes is detected and processed, we extract candidate motifs of various sizes (see Algorithm 1). We define the minimum motif length as l_min and the maximum

[5] http://www.aubio.org

[6] http://www.tomgibara.com/computer-vision/canny-edge-detector

motif length as l_max. All contiguous motifs of length greater than or equal to l_min and less than or equal to l_max are stored. For our experiments, the variables l_min and l_max are set to 4 and 7 respectively.

After the candidate motifs are gathered, the motifs with the highest probability according to the model of the training data are selected (see Algorithm 2). The probabilities are computed in different ways according to which ML model is used. For the HMM, the probability is computed using the forward algorithm. For the VMMs, the probability is computed by multiplying all the transitional probabilities of the notes in the motif. For the RNN, the activation value of the correct output note is used to derive a pseudo-probability for each motif.

Pitches and rhythms are learned separately, weighted, and combined to form a single probability. The weightings are necessary in order to give equal consideration to both pitches and rhythms. In our system, a particular pitch is generally less likely than a particular rhythm because there are more pitches to choose from. Thus, the combined probability is defined as

$$P_{p+r}(m) = Pr(m_p){N_p}^{|m|} + Pr(m_r){N_r}^{|m|} \tag{3.1}$$

where m is a motif, $|m|$ is the length of m, m_p is the motif pitch sequence, m_r is the motif rhythm sequence, $Pr(m_p)$ and $Pr(m_r)$ are given by the model, N_p and N_r are constants, and $N_p > N_r$. In this paper we set $N_p = 60$ and $N_r = 4$ (N_p is much larger than N_r because the effective pitch range is much larger than the effective rhythm range). The resulting value is not a true probability because it can be greater than 1.0, but this is not significant because we are only interested in the relative probability of motifs.

Algorithm 1 *extract_candidate_motifs*

```
**Input:** $notes$, $l\_min$, $l\_max$
$candidate\_motifs \leftarrow \{\}$
**for** $l\_min \leq l \leq l\_max$ **do**
    **for** $0 \leq i \leq |notes| - l$ **do**
        $motif \leftarrow (notes_i, notes_{i+1}, ..., notes_{i+l-1})$
        $candidate\_motifs \leftarrow candidate\_motifs \cup motif$
**return** $candidate\_motifs$
```

Algorithm 2 *discover_best_motifs*

```
1: **Input:** $notes$, $model$, $num\_motifs$, $l\_min$, $l\_max$
2: $C \leftarrow extract\_candidate\_motifs(notes, l\_min, l\_max)$
3: $best\_motifs \leftarrow \{\}$
4: **while** $|best\_motifs| < num\_motifs$ **do**
5:     $m^* \leftarrow \underset{m \in C}{\mathrm{argmax}}[norm(|m|)Pr(m|model)]$
6:     $best\_motifs \leftarrow best\_motifs \cup m^*$
7:     $C \leftarrow C - \{m^*\}$
8: **return** $best\_motifs$
```

Since shorter motifs are naturally more probable than longer motifs, an additional normalization step is taken in Algorithm 2. We would like each motif length to have equal probability:

$$P_{equal} = \frac{1}{(l\_max - l\_min + 1)} \tag{3.2}$$

Since the probability of a generative model emitting a candidate motif of length l is

$$P(l) = \sum_{m \in C, |m| = l} Pr(m|model) \tag{3.3}$$

we introduce a length-dependent normalization term that equalizes the probability of selecting motifs of various lengths.

$$norm(l) = \frac{P_{equal}}{P(l)} \tag{3.4}$$

This normalization term is used in step 5 of Algorithm 2.

Chapter 4

Validation and Results

We perform four stages of validation for this system. First, we compare the entropy of pitch-detected and edge-detected music sequences to comparable random sequences as a baseline sanity check to see if images and audio are better sources of inspiration than are random processes. Second, we run our motif discovery system on real music scores instead of media, and we validate the motif discovery process by comparing the discovered motifs to hand annotated themes for the piece of music. Third, we evaluate the structural value of the motifs. This is done by comparing the entropy of the discovered motifs, candidate motifs, and themes in the training set. We also measure the amount of innovation in the motifs by measuring the probability of the selected motifs against the probability of the training themes according to the ML model. In the second and third stages of evaluation, we also compare results when smaller subsets of the training data are used to train the ML models. Fourth, we compare the normalized probabilities of motifs discovered by our system against the normalized probabilities of motifs discovered by random number generators. We argue that motif discovery is more efficient when media inspirations are used and less efficient when random number generators are used.

4.1 Preliminary Evaluation of Inspirational Sources

Although pitch detection is intended primarily for monophonic music signals, interesting results are still obtained on non-musical audio signals. Additionally, interesting musical inspiration can be obtained from image files. We performed some preliminary work on fifteen audio files and fifteen image files and found that these pitch-detected and edge-detected sequences were better inspirational sources than random processes. We compared the *entropy* (see Equation 4.1) of these sequences against comparable random sequences and found that there was more rhythm and pitch regularity in the pitch-detected and edge-detected sequences. In our data, the sample space of the random variable X is either a set of pitches or a set of rhythms, so $Pr(x_i)$ is the probability of observing a particular pitch or rhythm.

$$H(X) = -\sum_{i=1}^{n} Pr(x_i) \log_b Pr(x_i) \tag{4.1}$$

More precisely, for one of these sequences we found the sequence length, the minimum pitch, maximum pitch, minimum note duration, and maximum note duration. Then we created a sequence of notes from two uniform random distributions (one for pitch and one for rhythm) with the same length, minimum pitch, maximum pitch, minimum note duration, and maximum note duration. In Tables 4.1 and 4.2, the average pitch and rhythm entropy measures were lower for pitch-detected and edge-detected sequences. A heteroscedastic, two-tailed Student's t-test on the data shows statistical significance with p-values of 2.51×10^{-5} for pitches from images, 1.36×10^{-18} for rhythms from images, and 0.0004 for rhythms from audio files. Although the p-value for pitches from audio files is not statistically significant (0.175), it is lowered to 0.003 when we remove the three shortest audio files: *DarthVaderBreathing.wav*, *R2D2.wav*, and *ChewbaccaRoar.wav*. This suggests that there is potential for interesting musical content [20] in the pitch-detected and edge-detected sequences even though the sequences originate from non-musical sources.

4.2 Evaluation of Motif Discovery Process

A test set consists of 15 full music scores with one or more hand annotated themes for each score. The full scores are fetched from KernScores,[1] and the corresponding themes are removed from the training data set (taken from the aforementioned Electronic Dictionary of Musical

[1] http://kern.ccarh.org/

Inspirational Audio File Name	**Pitch Entropy**	**Random Pitch Entropy**	**Rhythm Entropy**	**Random Rhythm Entropy**
Reunion2005.wav	4.521	5.122	1.478	2.58
Neverland.wav	4.376	5.153	1.641	2.804
Birdsong.wav	4.835	5.156	3.317	5.154
ThunderAndRain.wav	4.465	5.152	3.196	6.283
SparklingWater.wav	5.002	5.151	0.54	2.321
TropicalRain.wav	4.994	5.164	2.485	4.083
PleasantBeach.wav	4.698	5.136	3.856	6.761
ChallengerDisasterAddress.wav	4.071	4.87	2.034	3.57
InauguralAddress.wav	4.865	5.162	1.914	5.037
MLKDream.wav	5.013	5.16	1.913	5.796
DarthVaderBreathing.wav	2.86	2.795	1.429	2.104
R2D2.wav	4.868	4.746	1.364	3.203
Lightsabers.wav	3.671	5.042	1.867	3.567
ChewbaccaRoar.wav	2.722	2.922	1.357	2.171
Blasters.wav	4.17	4.272	2.251	3.726
Average	4.342	4.734	2.043	3.944

Table 4.1: Pitch and rhythm entropy from audio inspirations. The entropy from pitch-detected sequences is lower than comparable random sequences. This suggests that pitch-detected audio sequences are better inspirational sources for music than random processes.

Inspirational Image File Name	**Pitch Entropy**	**Random Pitch Entropy**	**Rhythm Entropy**	**Random Rhythm Entropy**
Motif.jpg	6.269	6.953	4.19	14.399
Fociz.jpg	6.451	6.999	4.095	15.437
Bioplazm2.jpg	6.743	6.988	4.201	15.369
LightPaintMix.jpg	5.989	6.869	4.922	14.487
Variation-Investigation.jpg	6.52	6.965	3.903	15.813
Pollock-Number5.jpg	6.099	6.79	3.75	12.737
Dali-ThePersistenceofMemory.jpg	6.115	6.684	4.634	13.662
Monet-ImpressionSunrise.jpg	5.073	6.583	4.486	13.813
DaVinci-MonaLisa.jpg	6.305	6.657	4.985	11.8
Vermeer-GirlWithaPearlEarring.jpg	6.465	6.869	4.844	14.156
Landscape.jpg	6.304	6.999	4.373	15.076
Stonehenge.jpg	5.739	6.374	4.851	14.787
River.jpg	6.252	6.869	5.057	14.994
Fish.jpg	5.59	6.882	4.547	15.104
Bird.jpg	5.837	6.227	5.655	14.012
Average	6.117	6.78	4.566	14.376

Table 4.2: Pitch and rhythm entropy from image inspirations. The entropy from edge-detected sequences is lower than comparable random sequences. This suggests that edge-detected sequences are better inspirational sources for music than random processes.

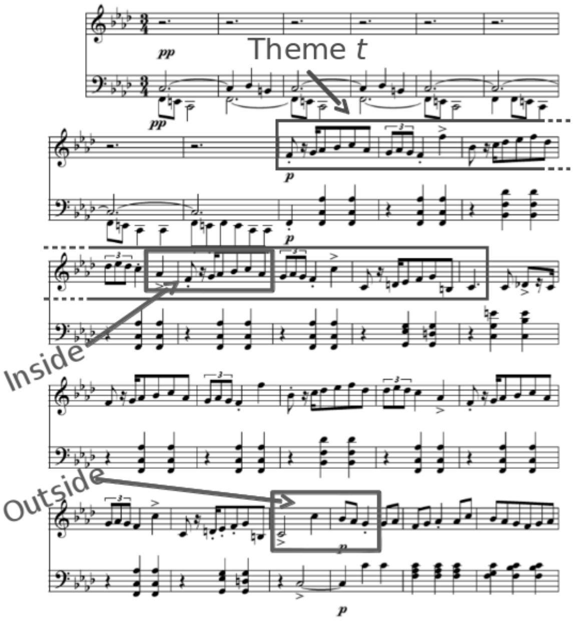


Figure 4.1: An example of a motif inside the theme and a motif outside the theme for a piece of music. Given a model, the average normalized probability of the motifs inside the theme are compared to the average normalized probability of the motifs outside the theme.

Themes). Each theme effectively serves as a hand annotated characteristic theme from a full score of music. This process is done manually due to the incongruence of KernScores and The Electronic Dictionary of Musical Themes. In order to ensure an accurate mapping, full scores and themes are matched up according to careful inspection of their titles and contents. We attempt to choose a variety of different styles and time periods in order to adequately represent the training data.

Due to the manual gathering of test data, we perform tests on a static test set and refrain from cross-validation. For each score in the test set, candidate motifs are gathered into a set C by iterating through the full score, one part at a time, using a sliding window from size l_min to l_max. This is the same process used to gather candidate motifs from audio and image files. C is then split into two disjoint sets, where C_t contains all the motifs that are subsequences of the matching theme for the score, and C_{-t} contains the remaining motifs. See Figure 4.1 for a visual example of motifs that are found inside and outside of the theme.

A statistic Q is computed which represents the mean normalized probability of the motifs in a set S:

$$Q(S|model) = \frac{\sum\limits_{m \in S} norm(|m|)Pr(m|model)}{|S|} \tag{4.2}$$

$Q(C_t|model)$ informs us about the probability of theme-like motifs being extracted by the motif discovery system. $Q(C_{-t}|model)$ informs us about the probability of non-theme-like motifs being discovered. A metric U is computed in order to measure the ability of the motif discovery system to discover desirable motifs.

$$U = \frac{Q(C_t|model) - Q(C_{-t}|model)}{\min\{Q(C_t|model), Q(C_{-t}|model)\}} \tag{4.3}$$

U is larger than zero if the discovery process successfully identifies motifs that have motivic or theme-like qualities according to the hand-labeled themes.

We use a validation set of music scores and their identified themes in order to fine tune the ML model parameters to maximize the U values. After these parameters are tuned, we calculate U over a separate test set of scores and themes for each learning model. The results are shown in Table 4.3.

Given the data in Table 4.3, a case can be made that certain ML models can effectively discover theme-like motifs with a higher probability than other motif candidates. Four of the six ML models have an average U value above zero. This means that an average theme is more likely to be discovered than an average non-theme for these four models. PPM and CTW have the highest average U values over the test set. LSTM has the worst average, but this is largely due to one outlier of -91.960. Additionally, PST performs poorly mostly due to two outliers of -24.363 and -31.614. Outliers are common in Table 4.3 because the themes in the music scores are sometimes too short to represent a broad sample of data. Except for LSTM and PST, all of the models are fairly robust by keeping negative U values to a minimum.

Score File Name	**CTW**	**HMM**	**LSTM**	**LZMS**	**PPM**	**PST**	**Average**
BachBook1Fugue15	4.405	4.015	3.047	2.896	11.657	4.951	5.162
BachInvention12	-2.585	-5.609	26.699	1.078	0.534	13.191	5.551
BeethovenSonata13-2	1.065	-0.145	7.769	8.876	4.973	9.182	5.287
BeethovenSonata6-3	-0.715	-5.320	2.874	0.832	1.283	4.801	0.626
ChopinMazurka41-1	6.902	0.808	-7.690	3.057	18.965	-24.363	-0.387
Corelli5-8-2	-6.398	-1.270	-0.692	-2.395	-1.166	1.690	-1.705
Grieg43-2	2.366	1.991	-2.622	0.857	8.800	-7.740	0.609
Haydn33-3-4	14.370	2.370	1.189	6.155	8.475	0.841	5.567
Haydn64-6-2	1.266	2.560	-1.092	0.855	1.809	-0.133	0.878
LisztBallade2	-0.763	-0.610	-1.754	-0.046	1.226	0.895	-0.175
MozartK331-3	0.838	0.912	3.829	0.756	3.222	5.413	2.495
MozartK387-4	-4.227	-0.082	-91.960	-2.127	-3.453	-31.614	-22.244
SchubertImprGFlat	49.132	3.169	0.790	8.985	59.336	1.122	20.422
SchumannSymph3-4	0.666	2.825	-2.154	0.289	1.560	-6.830	-0.607
Vivaldi3-6-1	7.034	2.905	0.555	7.055	9.633	-0.367	4.469
Average	4.890	0.568	-4.081	2.475	8.457	-1.931	

Table 4.3: U values for various score inputs and ML models. Positive U values show that the average normalized probability of motifs inside themes is higher than the same probability for motifs outside themes. Positive U values suggest that the motif discovery system is able to detect differences between theme-like motifs and non-theme-like motifs.

In order to understand the effects of training on different sets of data, we collect the same U values by training on various subsets of the data. For instance, U values are computed after training on only the themes in the data set composed by Bach, Beethoven, or some other composer. The U values for several subsets of the training data are shown in Appendix B, and the median is also included in these tables in order to minimize the effects of outliers. Outliers are especially common in this data for the same reason they are common in Table 4.3. We show Table 4.4 here, which contains the U values for each score and ML model after training on only the themes by Bach in the training set. Table 4.4 and all the tables in Appendix B generally give lower U values and more negative outliers than when the entire training set is used.

As expected, the mean and median U values on the upper right side of Table 4.4 for the two Bach scores are fairly high when only Bach themes are used in training. Strong mean and median pairs are also found for the two works by Haydn. This could be due to the fact

Score File Name	**CTW**	**HMM**	**LSTM**	**LZMS**	**PPM**	**PST**	**Mean**	**Median**
BachBook1Fugue15	5.421	11.259	3.742	7.463	9.198	7.335	7.403	7.399
BachInvention12	0.375	-2.840	9.200	-1.859	0.626	25.351	5.142	0.500
BeethovenSonata13-2	13.057	0.278	2.190	4.217	2.511	2.490	4.124	2.500
BeethovenSonata6-3	-0.555	-1.706	5.588	-0.933	0.090	6.614	1.516	-0.233
ChopinMazurka41-1	28.394	5.482	-19.081	1.020	9.915	-377.693	-58.661	3.251
Corelli5-8-2	-40.103	-9.672	-0.018	-11.364	-17.721	5.819	-12.176	-10.518
Grieg43-2	3.399	7.232	-1.365	-0.385	8.187	-9.831	1.206	1.507
Haydn33-3-4	21.489	12.861	1.044	28.487	23.981	4.451	15.385	17.175
Haydn64-6-2	7.344	4.420	-1.303	1.864	7.226	-2.316	2.872	3.142
LisztBallade2	0.426	-0.414	-1.268	0.097	-0.234	-0.170	-0.261	-0.202
MozartK331-3	0.352	-0.445	7.057	-0.414	1.325	12.214	3.348	0.839
MozartK387-4	-3.223	-2.825	-48.039	-495.799	-20.821	-69.631	-106.723	-34.430
SchubertImprGFlat	-0.764	-0.146	7.800	5.716	3.255	9.671	4.255	4.486
SchumannSymph3-4	-14.501	-1.129	-7.549	-191.425	-23.026	-19.069	-42.783	-16.785
Vivaldi3-6-1	-0.013	-2.725	-5.421	-0.394	-0.072	-1.076	-1.617	-0.735
Mean	1.406	1.309	-3.161	-43.581	0.296	-27.056		
Median	0.375	-0.414	-0.018	-0.385	1.325	2.490		

Table 4.4: U values when ML model is trained on only works by Bach.

that Haydn's era was shortly after Bach's era. In contrast, the mean and median U values for Corelli and Vivaldi (both living about the same time as Bach) are all negative. This suggests that some composers are influenced more by composers in past eras than in their current era.

In order to quickly visualize the effects of training on various subsets, we include Figure 4.2. In this figure, the x-axis contains the name of the composer for each subset of the training data along with their birth year. The y-axis contains the name of the score along with the birth year of the composer. Using only CTW, HMM, and PPM(the highest performing models from Figure 4.3), we calculate the median U value for each musical score trained on each subset. In order to simplify and smooth the data, we rank each row from 1 to 11, where 1 is the highest median and 11 is the lowest median. We color each rank with a different shade of grey, where higher ranks are darker and lower ranks are lighter.

We originally expected the data in Figure 4.2 to show dark grey starting at the bottom left corner and moving to the upper right corner. If this were the case, it would mean that training on subsets of earlier music would help our system better discover theme-like motifs from earlier scores, and training on subsets of later music would help our system better

Composer Birth Year	Score											
1843	Grieg43-2.krn	3	1	5	4	11	9	10	8	6	2	7
1811	LisztBallade2.krn	3	10	11	5	7	6	8	4	1	9	2
1810	SchumannSymphony3-4.krn	2	3	5	4	10	11	1	6	8	9	7
1810	ChopinMazurka41-1.krn	11	10	8	9	4	1	5	3	6	2	7
1797	SchubertImpromptuGFlat.krn	11	10	8	9	3	2	7	4	5	1	6
1770	BeethovenSonata13-2.krn	5	7	8	2	6	9	1	11	4	10	3
1770	BeethovenSonata6-3.krn	3	10	11	1	2	9	4	7	8	6	5
1756	MozartK387-4.krn	9	7	8	11	1	6	4	3	10	2	5
1756	MozartK331-3.krn	5	2	8	3	9	6	11	4	1	10	7
1732	Haydn64-6-2.krn	1	6	7	4	9	3	10	2	8	11	5
1732	Haydn33-3-4.krn	1	4	5	3	9	2	11	6	7	8	10
1685	BachBook1Fugue15.krn	1	10	11	7	5	4	8	9	3	6	2
1685	BachInvention12.krn	2	5	8	7	11	9	4	10	6	1	3
1678	Vivaldi3-6-1.krn	11	7	5	6	4	1	2	3	10	8	9
1653	Corelli5-8-2.krn	11	10	6	9	1	8	4	3	2	7	5
	Composer	Bach	Haydn	Mozart	Beethoven	Chopin	Schumann	Wagner	Brahms	Dvorak	Debussy	Prokofiev
	Birth Year	1685	1732	1756	1770	1810	1810	1813	1833	1841	1862	1891

Figure 4.2: Rankings of median U values from CTW, HMM, and PPM for various training subsets. For each combination of a training subset and score, we calculate the median U value from the three most reliable ML models: CTW, HMM, and PPM. We order the x-axis according to the birth year of each training subset composer, and we order the y-axis according to the birth year of the composer of each piece. We rank each row from 1 to 11 and color each cell in various shades of grey according to their rank. The results are inconclusive, suggesting that motifs are too short to encapsulate time-specific styles.

discover theme-like motifs from later scores. However, we do not see any conclusive pattern in Figure 4.2 that would suggest what we expected. Perhaps motifs are too short to encapsulate time-specific styles.

One could argue that musical style is influenced more by locale rather than time period. This appears to be the case with Corelli and Vivaldi (both Italian) showing little correlation with Bach (German) in Figure 4.2, even though these three composers were from the same era. In future work, it would be interesting to compare the stylistic influences of locale and time period among various composers.

We also compare the mean and median U values for the various ML models in Figure 4.3. In this figure, we tally up the number of times that the mean and median values are both positive for each learning model on the various training subsets. It is clear that CTW, HMM, and PPM are robust and perform well for many different training subsets; it is also clear that LSTM, LZMS, and PST perform poorly over the various training subsets.

An interesting difference in the subset training results is the change in performance for LZMS. LZMS has an average U value of 2.475 when the entire training data set is used

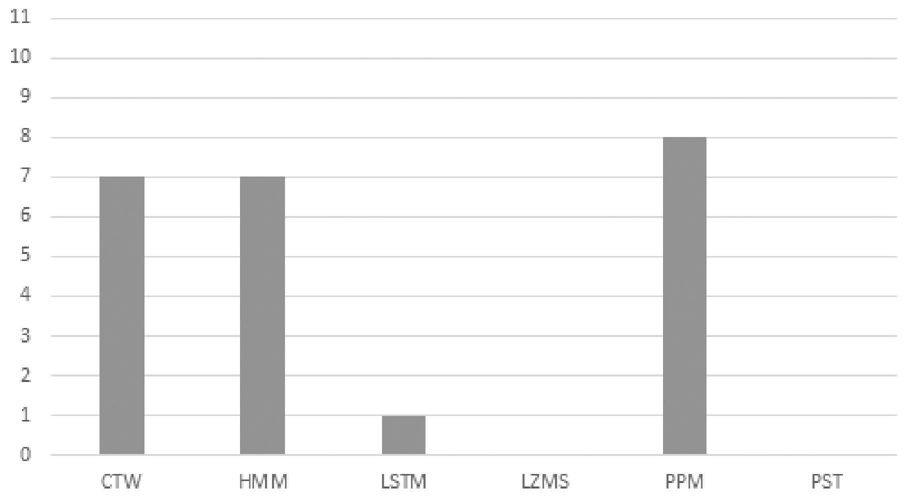


Figure 4.3: Number of positive mean and median U values for various ML models. We tally up the number of times that the mean and median values are both positive for each learning model on the 11 training subsets. It is clear that CTW, HMM, and PPM perform well for most of the 11 training subsets.

(see Table 4.3), but it never has both a mean and median U value above zero for any of the training subsets (see Figure 4.3). This suggests that LZMS performs better with more training data while CTW, HMM, and PPM perform well on small and large training data sets.

4.3 Evaluation of Structural Quality of Motifs

We also evaluate both the information content and the level of innovation of the discovered motifs. First, we measure the information content by computing entropy as we did before. We compare the entropy of the discovered motifs to the entropy of the candidate motifs. We also segment the actual music themes from the training set into a set of motifs using Algorithm 1, and we add the entropy of these motifs to the comparison. In order to ensure a fair comparison, we perform a sampling procedure which requires each set of samples to contain the same proportions of motif lengths, so that our entropy calculation is not biased by the length of the motifs sampled. The results for two image input files and two audio input files are displayed in Table 4.5. The images and audio files are chosen for their textural and aural variety, and their statistics are representative of other files we tested. *Bioplazm2.jpg* is a computer-generated fractal while *Landscape.jpg* is a photograph, and *Lightsabers.wav* is a sound effect from the movie *Star Wars* while *Neverland.wav* is a recording of a person reading poetry.

The results are generally as one would expect. The average pitch entropy is always lowest on the training theme motifs, it is higher for the discovered motifs, and higher again for the candidate motifs. With the exception of *Landscape.jpg*, the average rhythm entropy follows the same pattern as pitch entropy for each input. One surprising observation is that the rhythm entropy for some of the ML models is sometimes higher for the discovered motifs than it is for the candidate motifs. This suggests that theme-like rhythms are often no more predictable than non-theme rhythms. However, the pitch entropy almost always tends to be lower for the discovered motifs than the candidate motifs. This suggests that theme-like pitches tend to be more predictable. It also suggests that pitches could be more significant than rhythms in defining the characteristic qualities in themes and motifs.

Next, we measure the level of innovation of the best motifs discovered. We do this by taking a metric R (similar to U) using two Q statistics (see equation 4.2), where A is the set of actual themes and E is the set of discovered motifs.

$$R = \frac{Q(A|model) - Q(E|model)}{\min\{Q(A|model), Q(E|model)\}} \tag{4.4}$$

When R is greater than zero, A is more likely than E given the ML model. In this case, we assume that there is a different model that would better represent E. If there is a better model for E, then E must be novel to some degree when compared to A. Thus, If R is greater than zero, we infer that E innovates from A. The R results for the same four input files are shown along with the entropy statistics in Table 4.5. Except for PPM, all of the ML models produce R values greater than zero for each of the four inputs.

While statistical metrics provide some useful evaluation in computationally creative systems, listening to the motif outputs and viewing their musical notation will also provide

Bioplazm2.jpg	**CTW**	**HMM**	**LSTM**	**LZMS**	**PPM**	**PST**	**Average**
training motif pitches	1.894	1.979	1.818	1.816	1.711	1.536	1.793
discovered motif pitches	2.393	2.426	1.944	1.731	2.057	1.759	2.052
candidate motif pitches	2.217	2.328	2.097	2.104	1.958	1.784	2.081
training motif rhythms	1.009	1.051	0.976	0.970	0.927	0.822	0.959
discovered motif rhythms	2.110	2.295	1.789	2.212	0.684	1.515	1.767
candidate motif rhythms	2.387	2.466	2.310	2.309	2.132	1.934	2.256
R	7.567	13.296	20.667	4.603	-0.276	7.643	8.917

Landscape.jpg	**CTW**	**HMM**	**LSTM**	**LZMS**	**PPM**	**PST**	**Average**
training motif pitches	1.894	1.979	1.818	1.816	1.711	1.536	1.793
discovered motif pitches	1.974	2.074	2.143	1.833	2.027	1.675	1.954
candidate motif pitches	2.429	2.531	2.598	2.341	2.271	2.028	2.367
training motif rhythms	1.009	1.051	0.976	0.970	0.927	0.822	0.959
discovered motif rhythms	1.984	1.863	2.175	1.983	0.727	1.455	1.698
candidate motif rhythms	1.549	1.712	1.810	1.509	1.396	1.329	1.551
R	0.805	0.236	1.601	0.429	4.624	1.283	1.496

Lightsabers.wav	**CTW**	**HMM**	**LSTM**	**LZMS**	**PPM**	**PST**	**Average**
training motif pitches	1.894	1.979	1.818	1.816	1.711	1.536	1.793
discovered motif pitches	2.076	1.884	1.881	1.652	2.024	1.586	1.850
candidate motif pitches	2.225	2.097	2.217	1.876	2.115	1.755	2.048
training motif rhythms	1.009	1.051	0.976	0.970	0.927	0.822	0.959
discovered motif rhythms	1.534	1.309	2.024	1.623	0.860	1.225	1.429
candidate motif rhythms	1.540	1.524	1.541	1.502	1.548	1.276	1.489
R	5.637	0.793	27.227	4.812	6.768	7.540	8.796

Neverland.wav	**CTW**	**HMM**	**LSTM**	**LZMS**	**PPM**	**PST**	**Average**
training motif pitches	1.894	1.979	1.818	1.816	1.711	1.536	1.793
discovered motif pitches	1.823	2.480	2.132	1.773	1.997	1.701	1.984
candidate motif pitches	2.153	2.248	2.250	2.141	2.242	1.839	2.146
training motif rhythms	1.009	1.051	0.976	0.970	0.927	0.822	0.959
discovered motif rhythms	1.550	1.587	1.560	1.779	0.289	1.128	1.315
candidate motif rhythms	1.472	1.469	1.471	1.477	1.469	1.226	1.431
R	1.520	10.163	24.968	4.283	0.257	6.865	8.010

Table 4.5: Entropy and R values for various inputs. We measure the pitch and rhythm entropy of motifs extracted from the training set, the best motifs discovered, and all of the candidate motifs extracted. On average, the entropy increases from the training motifs to the discovered motifs, and it increases again from the discovered motifs to the candidate motifs. The R values are positive when the training motifs are more probable according to the model than the discovered motifs. R values represent the amount of novelty with respect to the training data.

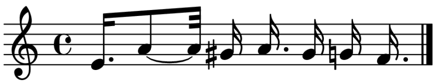
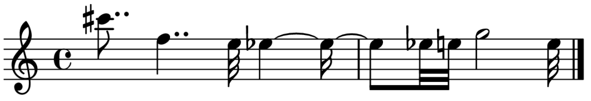
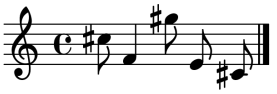
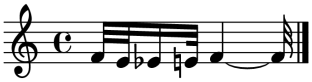
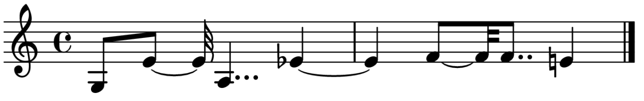
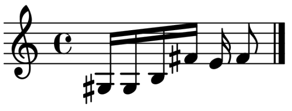
ML Model	Input File	Motif Discovered
CTW	*MLKDream.wav*	
HMM	*Birdsong.wav*	
LSTM	*Pollock-Number5.jpg*	
LZMS	*Lightsabers.wav*	
PPM	*Bioplazm2.jpg*	
PST	*Neverland.wav*	

Table 4.6: Six motifs discovered by our system.

valuable insights for this system. We include six musical notations of motifs discovered by this system in Table 4.6. These six motifs represent typical motifs discovered by our system, and they are not chosen according to specific preferences. We invite the reader to view more motifs discovered by our system in Appendix A and listen to sample outputs at `http://axon.cs.byu.edu/motif-discovery`.

4.4 Comparison of Media Inspiration and Random Inspiration

We have shown the efficacy of the motif extraction process and the structural quality of motifs, but one could still argue that a simple random number generator could be used to inspire the composition of motifs with equal value. While we agree that random processes could inspire motifs of similar quality (if given enough time), we argue that our system discovers high quality motifs more efficiently.

In order to show this, we compare the differences in efficiency between media-inspired motifs and random-inspired motifs. We extract candidate motifs from a media file and, given a model, we select a portion of motifs with the highest normalized probabilities. This is the same process described in our methodology section, except we report the results for various percentages of motifs selected among all the candidate motifs. We also generate a set of random motifs that are comparable to the candidate motifs. We do this by recording the minimum and maximum pitches and rhythms from the set of candidate motifs and restricting a random generator to only compose pitches and rhythms within those ranges. For each of the media-inspired candidate motifs, we generate a new random motif that has the same length as the media-inspired motif. This ensures that the set of random motifs is comparable to the set of media-inspired candidate motifs in every way except for pitch and rhythm selection. After the random motifs are gathered, we select the random motifs with the highest normalized probabilities given a model.

We gather the average normalized probability of the motifs selected from each set as a function of the percentage selected. These values are calculated on 12 audio files, averaged, and plotted in Figure 4.4. We use all of the audio files found in Appendix D except for *DarthVaderBreathing.wav*, *R2D2.wav*, and *ChewbaccaRoar.wav*. We remove these files because they are extremely brief and likely to misrepresent the data due to an insufficient number of candidate motifs. This process is also performed on all 15 image files found in Appendix D, and the plots are shown in Figure 4.5.

With the exception of LZMS using audio-inspired motifs, every media-inspired model selects motifs with higher normalized probabilities than random-inspired models on average. HMM does not separate the two distributions as well as the other models, but it still clearly places the media-inspired models above random-inspired models on average. The only time when HMM fails to do so is in Figure 4.4, where the audio-inspired motifs are equal to the random-inspired motifs at the first percentage line. This is probably due to the non-deterministic nature of HMMs, and this issue is resolved when higher percentages of motifs are selected. This is strong evidence that our system discovers higher quality motifs than a random generation system with the same number of candidate motifs. A random motif generator would need to generate a larger number of candidate motifs before the quality of the selected motifs matched those in our system. Thus, our system more efficiently discovers high quality motifs than a random motif generator.

We remind the reader that we are not measuring the quality of the ML models in this section, but instead we are using the ML models to judge the quality of motifs extracted from media-inspired and random-inspired sources. Due to this fact, some of the models deceptively perform well or poorly. For instance, LSTM and PST show a large difference between the normalized probabilities for the two modes of inspiration. At first glance, this seems surprising because LSTM and PST performed poorly in the validation of the motif discovery process (see Table 4.3, Table 4.4, and Figure 4.3). These unexpected positive results suggest that these models learn significant statistical information about motifs without learning enough to be useful in practice. Contrastingly, Figure 4.4 shows that LZMS measures roughly the same normalized probabilities for both modes of inspiration. However, a majority of the ML models clearly measure a significant advantage for media-inspired data over random-inspired data.

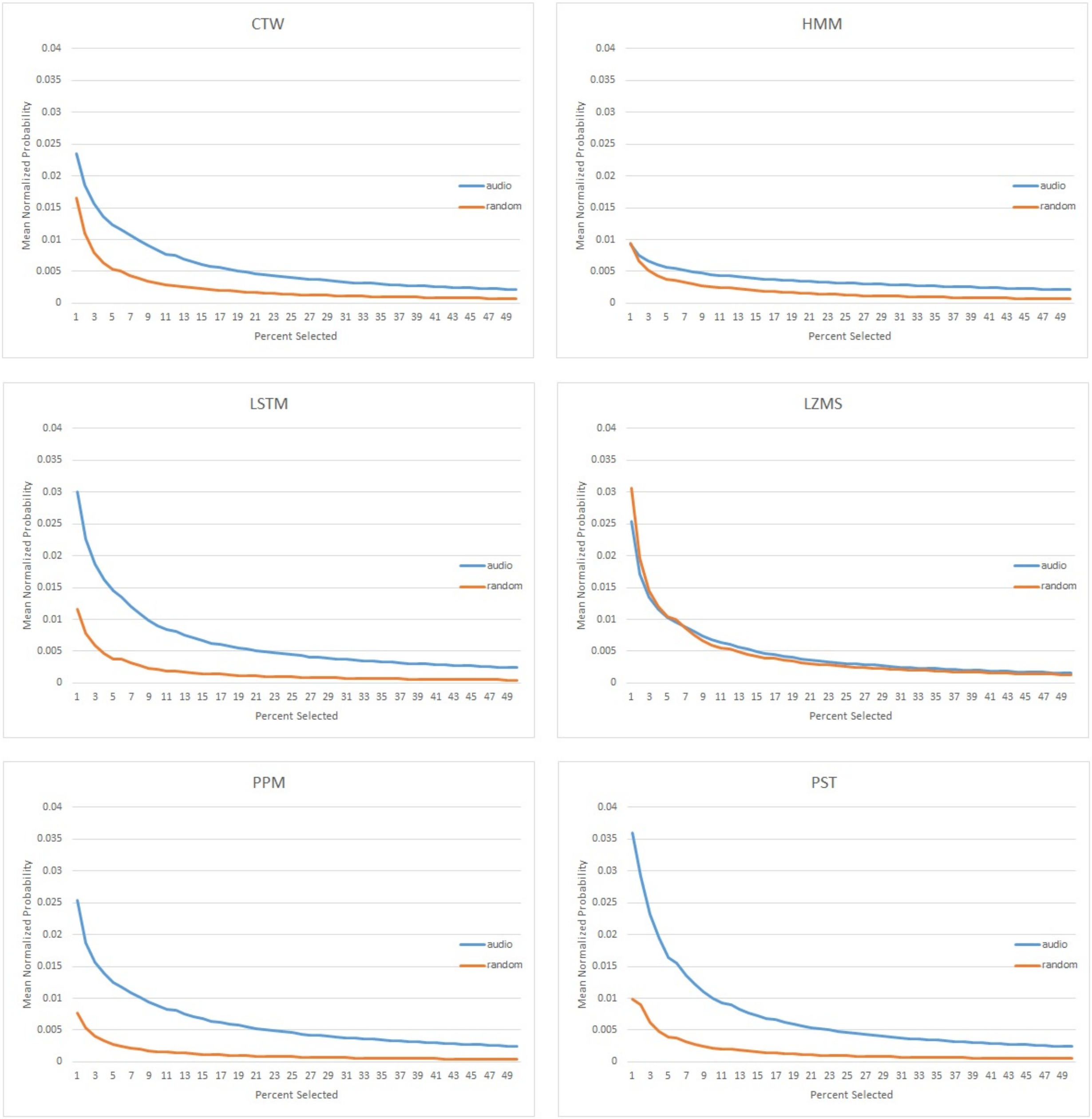


Figure 4.4: Mean normalized probability of motifs selected from audio files vs. random motifs. We extract candidate motifs from an audio file, select motifs according to normalized probabilities, and then we report the mean normalized probabilities for the selected motifs. We also generate a set of comparable random motifs with minimum and maximum pitch and rhythm values determined by the minimum and maximum pitch and rhythm values from the set of candidate motifs. We average the results over 12 audio files. The results suggest that audio files are more efficient sources of inspiration than random number generators.

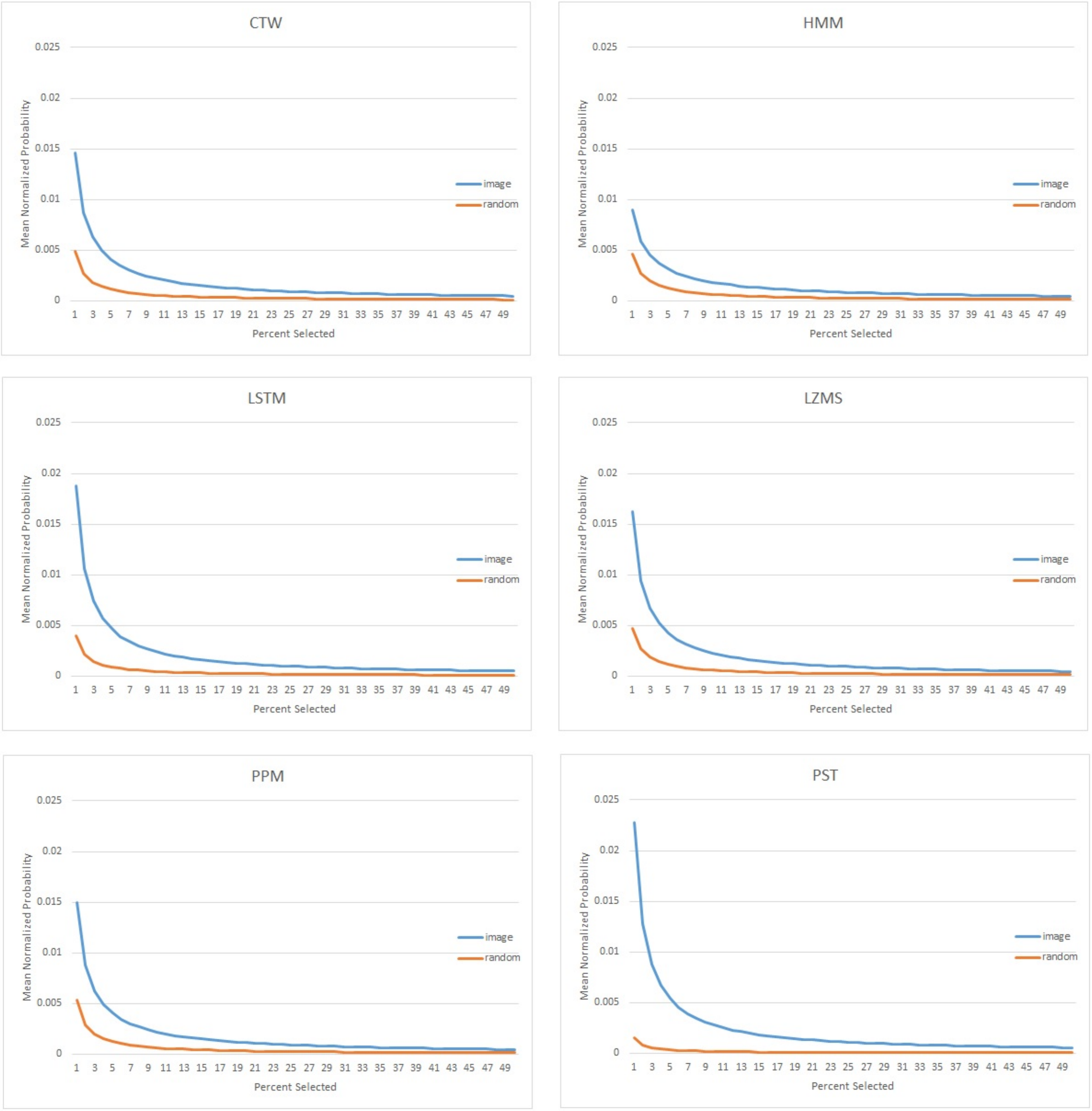


Figure 4.5: Mean normalized probability of motifs selected from images vs. random motifs. We extract candidate motifs from an image file, select motifs according to normalized probabilities, and then we report the mean normalized probabilities for the selected motifs. We also generate a set of comparable random motifs with minimum and maximum pitch and rhythm values determined by the minimum and maximum pitch and rhythm values from the set of candidate motifs. We average the results over 15 image files. The results suggest that images are more efficient sources of inspiration than random number generators.

Chapter 5

Conclusion

The motif discovery system in this paper composes musical motifs that demonstrate both innovation and value. We show that our system innovates from the training data by extracting candidate motifs from an inspirational source without generating data from a probabilistic model. The innovation is validated by observing high R values. The inspirational media sources in this system allow compositional seeds to begin outside of what is learned from the training data. This method is in line with many human composers such as Debussy, Messiaen, and Liszt, who received inspiration from sources outside of music literature.

Additionally, our motif discovery system maintains compositional value by learning from a training data set. The motif discovery process is tested by running it on actual music scores instead of audio and image files. The results show that motifs found inside of themes are, on average, more likely to be discovered than motifs found outside of themes. Generally, a larger variety and number of training data makes the system more likely to discover theme-like motifs rather than non-theme-like motifs.

Our evaluation of the motif discovery process shows that CTW, HMM, LZMS, and PPM are more likely to discover theme-like motifs than the other two ML models on the entire training data set. When only subsets of the training data set are used, LZMS no longer performs as well as CTW, HMM, and PPM. Thus, CTW and PPM stand out in both scenarios as models that perform well according to our metrics.

We find that media inspiration enables more efficient motif discovery than random inspiration. According to almost every ML model, media-inspired motifs are more probable than random-inspired motifs. A larger number of random motifs would need to be generated for the probabilities of these two sets of selected motifs to match.

Chapter 6

Future Work

The discovered motifs are the contribution of this system, and it will be left to future work to combine these motifs, add harmonization, and create full compositions. This work is simply the first step in a novel composition system.

A challenge in computational music composition is the notion of global structure. The motifs composed by this system offer a starting point for a globally structured piece. While there are a number of directions to take with this system as a starting point, we are inclined to compose from the bottom up in order to achieve global structure. Longer themes can be constructed by combining the motifs from this system using evolutionary or other approaches. Once a set of themes is created, then phrases, sections, movements, and full pieces can be composed in a similar manner. This process can create a cohesive piece of music that is based on the same small set of interrelated motifs that come from the same inspirational source.

A different system can compose from the top down, composing the higher level features first and using the motifs from this system as the lower level building blocks. This can be done using grammars [14], hierarchical neural networks [15], hierarchical hidden Markov models [18], or deep learning [11]. Inspirational sources can also be used at any level of abstraction: candidate themes, phrase structures, and musical forms can be extracted in addition to candidate motifs.

Since our system seeks to discover the atomic units of musical structure (motifs), we are now inclined to discover musical form, which is the global unit of musical structure. In one paradigm, global structure can be viewed as the most important element in a piece of music, and everything else (e.g., harmony, melody, motifs, and texture) is supplementary to it. Musical structure could be discovered from media using a process similar to motif discovery. The combination of a structure discovery system with a motif discovery system could produce pieces of music with interesting characteristics at multiple levels of abstraction.

This system can also be extended by including additional modes of inspirational input such as text or video. Motif composition can become affective by discovering semantic meaning and emotional content in text inputs. Motifs can be extracted from video with the same process described for images, except time can inspire additional features. With a myriad of inspirational sources available on the internet, our system could be improved by allowing it to favor certain inspirational sources over others. For instance, a motif discovery system that favors images of sunsets might be more interesting than a system that is equally inspired by everything it views. Additionally, inspirational sources could be combined over time rather than composing a single set of motifs for a single inspirational source. Humans are usually inspired by an agglomeration of sources, and many times they are not even sure what inspires them. Our motif discovery system would become more like a human composer if it were to incorporate some of these ideas in future work. Our goal in future work is for this system to be the starting point for an innovative, high quality, well-structured system that composes pieces which a human observer could call musical and creative.

References

[1] Ron Begleiter, Ran El-Yaniv, and Golan Yona. On prediction using variable order Markov models. *Journal of Artificial Intelligence Research*, 22:385–421, 2004.

[2] John Biles. GenJam: A genetic algorithm for generating jazz solos. In *Proceedings of the International Computer Music Conference*, pages 131–137, 1994.

[3] TM Blackwell. Swarm music: improvised music with multi-swarms. In *Proceedings of the AISB Symposium on Artificial Intelligence and Creativity in Arts and Science*, pages 41–49, 2003.

[4] Siglind Bruhn. *Images and Ideas in Modern French Piano Music: the Extra-musical Subtext in Piano Works by Ravel, Debussy, and Messiaen*, volume 6. Pendragon Press, 1997.

[5] Anthony R. Burton and Tanya Vladimirova. Generation of musical sequences with genetic techniques. *Computer Music Journal*, 23(4):59–73, 1999.

[6] Amílcar Cardoso, Tony Veale, and Geraint A Wiggins. Converging on the divergent: The history (and future) of the international joint workshops in computational creativity. *AI Magazine*, 30(3):15–22, 2009.

[7] David Cope. *Experiments in Musical Intelligence*, volume 12. AR Editions Madison, WI, 1996.

[8] Shlomo Dubnov, Gerard Assayag, Olivier Lartillot, and Gill Bejerano. Using machine-learning methods for musical style modeling. *Computer*, 36(10):73–80, 2003.

[9] Douglas Eck and Jasmin Lapalme. Learning musical structure directly from sequences of music. Technical report, University of Montreal, Department of Computer Science, 2008.

[10] Douglas Eck and Jürgen Schmidhuber. Learning the long-term structure of the blues. In *Proceedings of the International Conference on Artificial Neural Networks*, pages 284–289. 2002.

[11] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.

[12] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, and Jürgen Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In *A Field Guide to Dynamical Recurrent Neural Networks*, pages 237–244. IEEE Press, 2001.

[13] Victor Lavrenko and Jeremy Pickens. Music modeling with random fields. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 389–390, 2003.

[14] Jon McCormack. Grammar based music composition. *Complex Systems*, 96:321–336, 1996.

[15] Benjamin D Smith and Guy E Garnett. Improvising musical structure with hierarchical neural nets. In *Proceedings of the Eighth Artificial Intelligence and Interactive Digital Entertainment Conference*, pages 63–67, 2012.

[16] Robert Smith, Aaron Dennis, and Dan Ventura. Automatic composition from non-musical inspiration sources. In *Proceedings of the International Conference on Computational Creativity*, pages 160–164, 2012.

[17] Athina Spiliopoulou and Amos Storkey. A topic model for melodic sequences. *ArXiv E-prints*, 2012.

[18] Michele Weiland, Alan Smaill, and Peter Nelson. Learning musical pitch structures with hierarchical hidden Markov models. Technical report, University of Edinburgh, 2005.

[19] John David White. *The Analysis of Music*. Prentice-Hall, 1976.

[20] Gerraint A Wiggins, Marcus T Pearce, and Daniel Müllensiefen. Computational modelling of music cognition and musical creativity. *Oxford Handbook of Computer Music*, pages 383–420, 2009.

Appendix A

Motif Outputs

We limit our system to discovering only two motifs from an input file, and we present these two motifs from each combination of seven different inputs (4 audio files and 3 image files) with six ML models. The audio files are chosen in order to represent a variety of sounds (nature, sound effects, poetry, and speeches). The image files are chosen in order to represent a variety of images (fractals, nature, and art). Beyond this, there are no particular reasons why we choose any of the audio or image files over other media. There are no inherent time signatures associated with the motifs, so we display them all in a common time signature here.

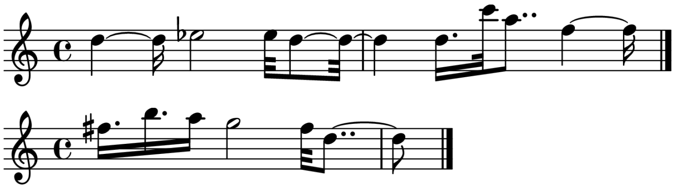
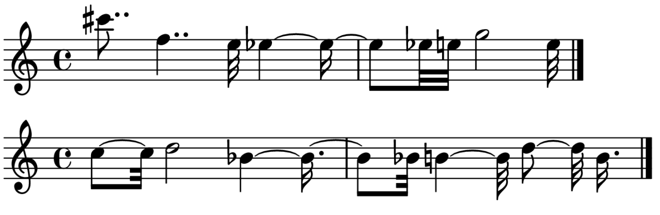
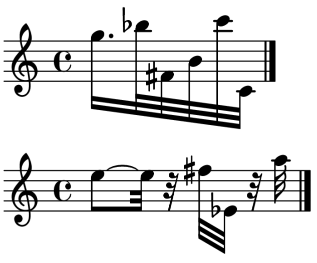
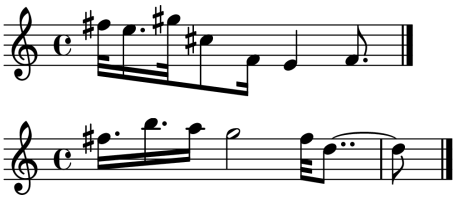
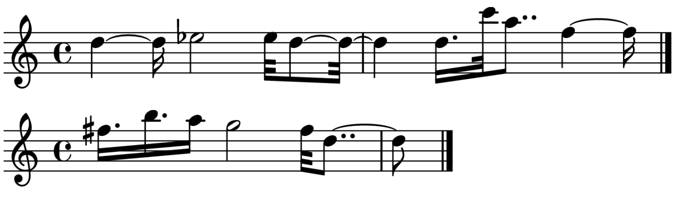
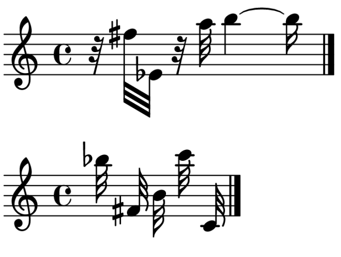
ML Model	2 Motifs Discovered from ***Birdsong.wav***
CTW	
HMM	
LSTM	
LZMS	
PPM	
PST	

Table A.1: Motifs discovered from *Birdsong.wav* for 6 ML models.

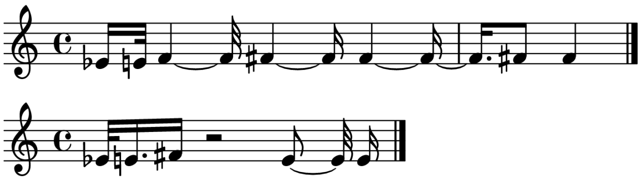
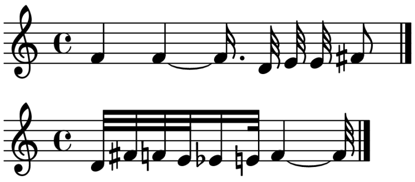
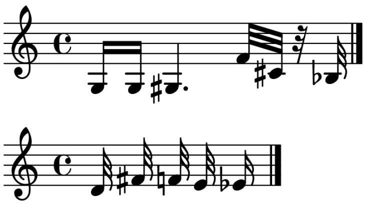
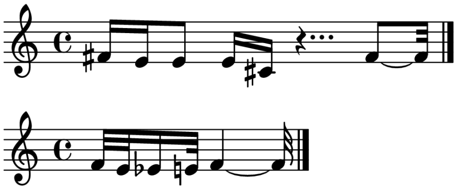
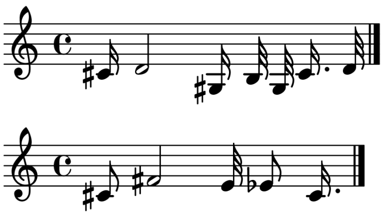
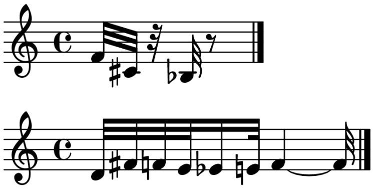
ML Model	2 Motifs Discovered from ***Lightsabers.wav***
CTW	
HMM	
LSTM	
LZMS	
PPM	
PST	

Table A.2: Motifs discovered from *Lightsabers.wav* for 6 ML models.

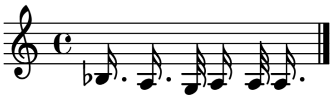
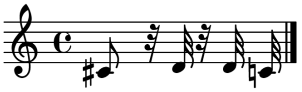
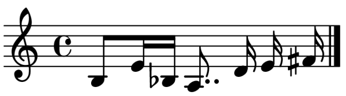
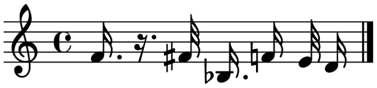
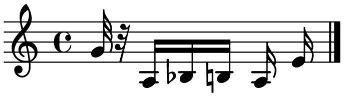
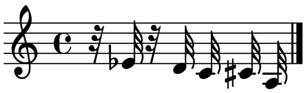
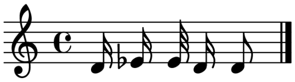
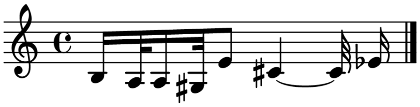
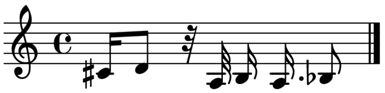
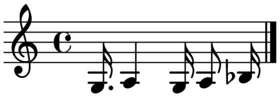
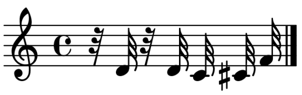
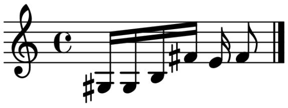
ML Model	2 Motifs Discovered from ***Neverland.wav***
CTW	 
HMM	 
LSTM	 
LZMS	 
PPM	 
PST	 

Table A.3: Motifs discovered from *Neverland.wav* for 6 ML models.

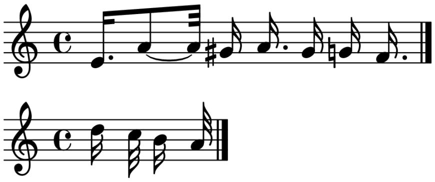
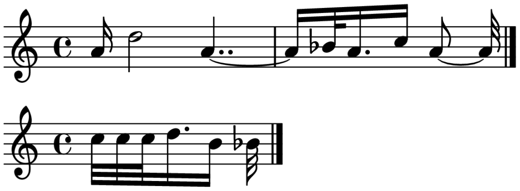
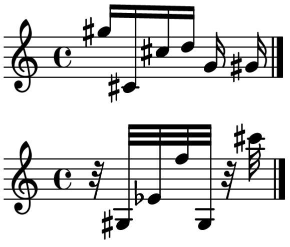
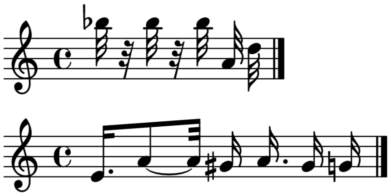
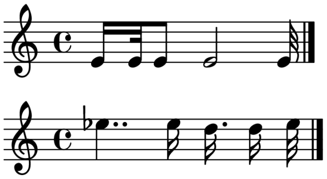
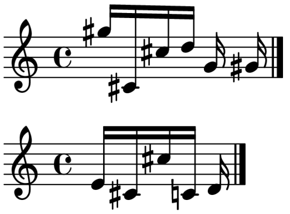
ML Model	2 Motifs Discovered from ***MLKDream.wav***
CTW	
HMM	
LSTM	
LZMS	
PPM	
PST	

Table A.4: Motifs discovered from *MLKDream.wav* for 6 ML models.

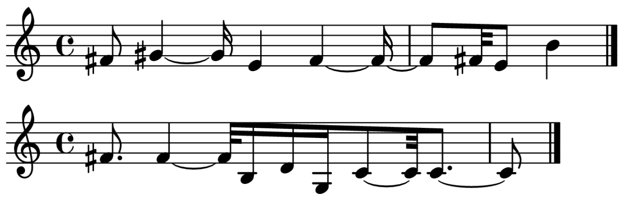
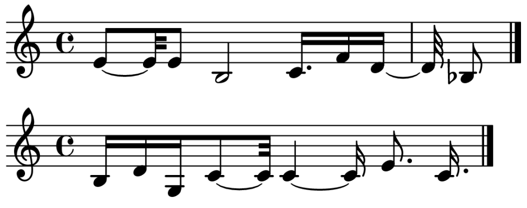
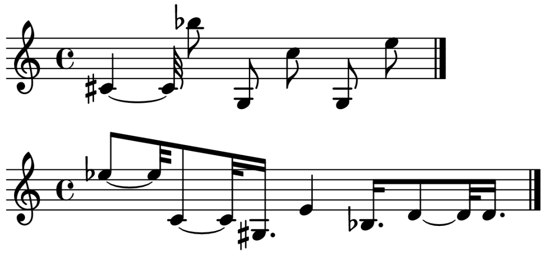
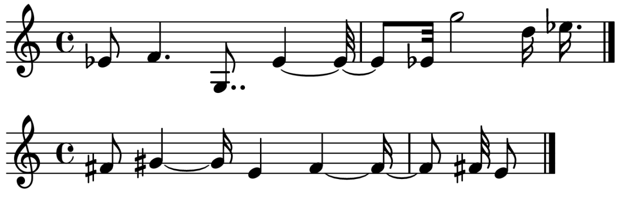
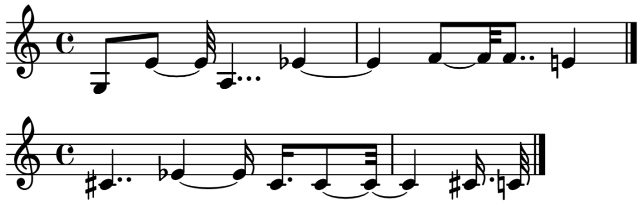
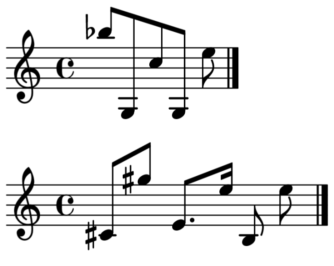
ML Model	2 Motifs Discovered from ***Bioplazm2.jpg***
CTW	
HMM	
LSTM	
LZMS	
PPM	
PST	

Table A.5: Motifs discovered from *Bioplazm2.jpg* for 6 ML models.

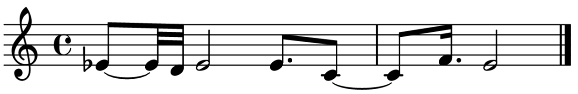
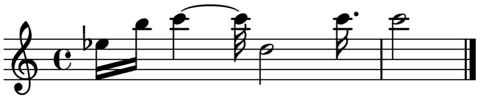
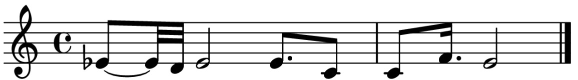
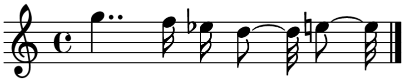
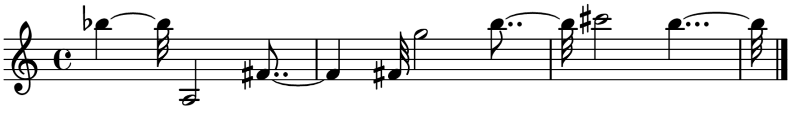
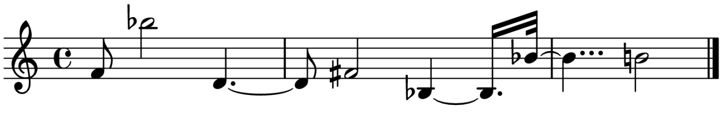
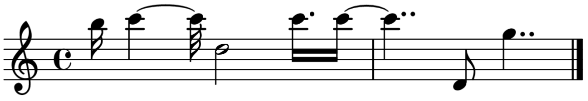
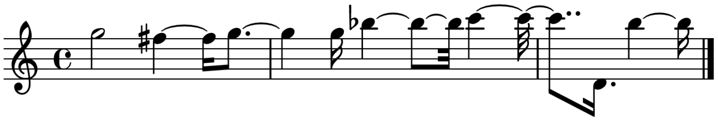
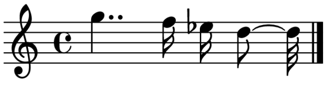
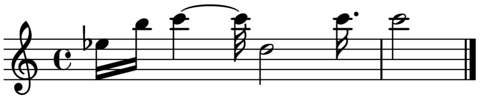
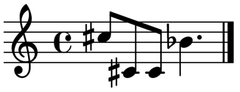
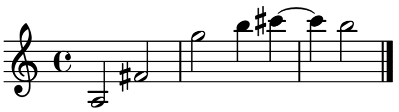
ML Model	2 Motifs Discovered from ***Landscape.jpg***
CTW	 
HMM	 
LSTM	 
LZMS	 
PPM	 
PST	 

Table A.6: Motifs discovered from *Landscape.jpg* for 6 ML models.

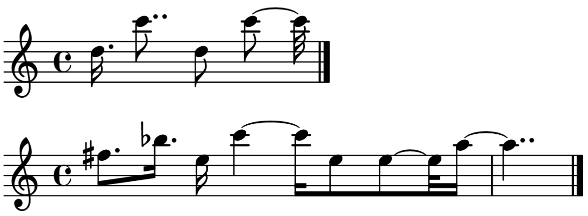
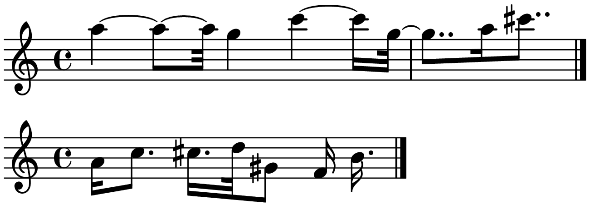
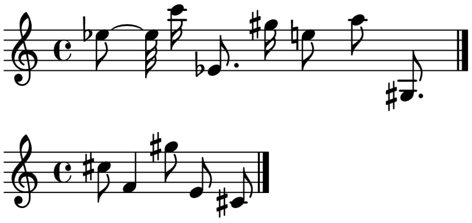
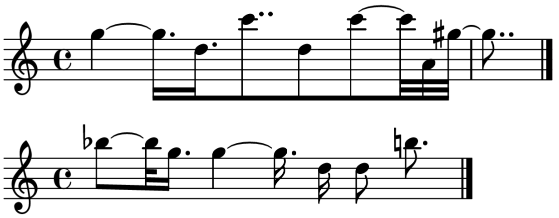
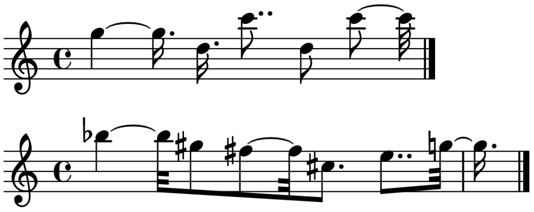
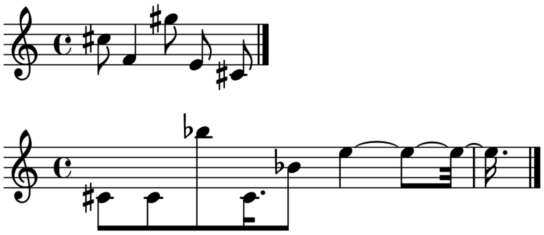
ML Model	2 Motifs Discovered from ***Pollock-Number5.jpg***
CTW	
HMM	
LSTM	
LZMS	
PPM	
PST	

Table A.7: Motifs discovered from *Pollock-Number5.jpg* for 6 ML models.

Appendix B

Evaluation of Motif Extraction Process for Subset Training

U values are reported for a set of scores for each ML model. A subset of the training data is used in each example which consists of the themes of only a single composer. In addition to reporting the mean values for rows and columns, we also include the median values in order to minimize the effects of outliers. Outliers are common in these tables because 1) themes are sometimes too short to represent a broad sample of data, and 2) the nature of training on subsets of the full data eliminates some of the smoothing that occurs over a large training data set.

Score File Name	**CTW**	**HMM**	**LSTM**	**LZMS**	**PPM**	**PST**	**Mean**	**Median**
BachBook1Fugue15	5.421	11.259	3.742	7.463	9.198	7.335	7.403	7.399
BachInvention12	0.375	-2.840	9.200	-1.859	0.626	25.351	5.142	0.500
BeethovenSonata13-2	13.057	0.278	2.190	4.217	2.511	2.490	4.124	2.500
BeethovenSonata6-3	-0.555	-1.706	5.588	-0.933	0.090	6.614	1.516	-0.233
ChopinMazurka41-1	28.394	5.482	-19.081	1.020	9.915	-377.693	-58.661	3.251
Corelli5-8-2	-40.103	-9.672	-0.018	-11.364	-17.721	5.819	-12.176	-10.518
Grieg43-2	3.399	7.232	-1.365	-0.385	8.187	-9.831	1.206	1.507
Haydn33-3-4	21.489	12.861	1.044	28.487	23.981	4.451	15.385	17.175
Haydn64-6-2	7.344	4.420	-1.303	1.864	7.226	-2.316	2.872	3.142
LisztBallade2	0.426	-0.414	-1.268	0.097	-0.234	-0.170	-0.261	-0.202
MozartK331-3	0.352	-0.445	7.057	-0.414	1.325	12.214	3.348	0.839
MozartK387-4	-3.223	-2.825	-48.039	-495.799	-20.821	-69.631	-106.723	-34.430
SchubertImprGFlat	-0.764	-0.146	7.800	5.716	3.255	9.671	4.255	4.486
SchumannSymph3-4	-14.501	-1.129	-7.549	-191.425	-23.026	-19.069	-42.783	-16.785
Vivaldi3-6-1	-0.013	-2.725	-5.421	-0.394	-0.072	-1.076	-1.617	-0.735
Mean	1.406	1.309	-3.161	-43.581	0.296	-27.056		
Median	0.375	-0.414	-0.018	-0.385	1.325	2.490		

Table B.1: U values when ML model is trained on only works by Bach.

Score File Name	**CTW**	**HMM**	**LSTM**	**LZMS**	**PPM**	**PST**	**Mean**	**Median**
BachBook1Fugue15	4.215	1.759	3.634	8.660	11.534	4.794	5.766	4.504
BachInvention12	-27.147	-26.746	12.467	-14.903	0.009	11.122	-7.533	-7.447
BeethovenSonata13-2	2.856	0.187	5.111	4.917	2.895	19.864	5.972	3.906
BeethovenSonata6-3	0.096	-0.133	3.448	-0.323	4.251	3.869	1.868	1.772
ChopinMazurka41-1	12.702	0.113	-11.743	-1.056	7.535	-92.563	-14.169	-0.471
Corelli5-8-2	-28.108	-0.682	-0.675	-3.417	-11.113	0.585	-7.235	-2.049
Grieg43-2	7.202	0.412	-2.786	20.266	10.400	-8.919	4.429	3.807
Haydn33-3-4	14.215	3.087	3.648	23.277	6.250	0.251	8.455	4.949
Haydn64-6-2	7.884	1.657	-0.434	4.846	0.922	-0.044	2.472	1.290
LisztBallade2	-2.332	-0.420	-0.618	-100.993	-0.021	0.767	-17.269	-0.519
MozartK331-3	-0.128	-0.330	3.444	-0.006	2.355	3.013	1.391	1.175
MozartK387-4	-1.801	-1.254	-15.822	-0.918	-1.971	-43.008	-10.796	-1.886
SchubertImprGFlat	1.262	-0.510	-17.023	4.186	77.731	2.010	11.276	1.636
SchumannSymph3-4	-0.214	0.049	-1.984	-1.890	-0.549	-8.535	-2.187	-1.219
Vivaldi3-6-1	2.334	-0.586	-1.057	1.871	4.446	-1.090	0.986	0.643
Mean	-0.464	-1.560	-1.359	-3.699	7.645	-7.192		
Median	1.262	-0.133	-0.618	-0.006	2.895	0.585		

Table B.2: U values when ML model is trained on only works by Beethoven.

Score File Name	**CTW**	**HMM**	**LSTM**	**LZMS**	**PPM**	**PST**	**Mean**	**Median**
BachBook1Fugue15	1.066	1.665	2.489	-1.290	0.652	5.355	1.656	1.366
BachInvention12	-49.770	-39.155	6.373	-24.257	-28.573	14.239	-20.190	-26.415
BeethovenSonata13-2	0.991	0.512	4.424	0.118	5.693	18.699	5.073	2.708
BeethovenSonata6-3	-5.591	-11.936	1.117	-11.074	-2.977	5.537	-4.154	-4.284
ChopinMazurka41-1	5.296	1.609	-29.043	-0.417	14.468	-128.227	-22.719	0.596
Corelli5-8-2	-3.161	-0.334	2.245	-3.068	-1.193	0.814	-0.783	-0.763
Grieg43-2	0.896	0.752	-6.278	-416.823	1.704	-14.240	-72.332	-2.763
Haydn33-3-4	27.327	4.190	13.609	9.344	12.796	0.228	11.249	11.070
Haydn64-6-2	1.324	2.344	-71.767	0.421	-0.052	0.182	-11.258	0.302
LisztBallade2	-0.283	-0.312	-0.741	-13.180	-0.485	1.107	-2.316	-0.398
MozartK331-3	1.529	-0.008	2.167	2.160	7.311	4.523	2.947	2.163
MozartK387-4	-1.770	-2.885	-20.425	-453.180	-5.151	-28.098	-85.251	-12.788
SchubertImprGFlat	84.785	224.506	2.593	52.659	140.264	1.113	84.320	68.722
SchumannSymph3-4	127.567	21.330	-1.995	7.559	19.387	-17.489	26.060	13.473
Vivaldi3-6-1	8.354	3.144	-1.669	5.053	18.746	-0.520	5.518	4.099
Mean	13.237	13.695	-6.460	-56.398	12.173	-9.118		
Median	1.066	0.752	1.117	-0.417	1.704	0.814		

Table B.3: U values when ML model is trained on only works by Brahms.

Score File Name	**CTW**	**HMM**	**LSTM**	**LZMS**	**PPM**	**PST**	**Mean**	**Median**
BachBook1Fugue15	0.567	0.322	4.420	0.089	-0.618	5.898	1.780	0.444
BachInvention12	-13.225	-5.879	11.099	-11.964	-4.169	41.342	2.867	-5.024
BeethovenSonata13-2	0.005	2.802	5.940	-0.317	2.623	6.560	2.935	2.712
BeethovenSonata6-3	2.490	-3.413	5.049	-64.036	-0.621	3.734	-9.466	0.935
ChopinMazurka41-1	1.014	0.683	-1.740	5.633	5.394	-4.277	1.118	0.848
Corelli5-8-2	0.017	0.104	2.023	1.364	-2.138	1.546	0.486	0.734
Grieg43-2	-6.289	-3.936	-1.050	-22.793	-58.105	-21.765	-18.990	-14.027
Haydn33-3-4	3.481	1.908	7.626	0.232	0.477	2.849	2.762	2.379
Haydn64-6-2	-0.182	0.651	0.151	1.479	0.751	-2.074	0.129	0.401
LisztBallade2	-10.019	-5.105	-0.134	-1.108	-1.899	-0.556	-3.137	-1.504
MozartK331-3	3.046	3.834	37.194	11.353	4.991	10.879	11.883	7.935
MozartK387-4	-23.279	-6.106	-350.715	-359.657	-32.038	-58.043	-138.306	-45.041
SchubertImprGFlat	56.197	165.012	-3.429	21.489	936.532	-2.961	195.473	38.843
SchumannSymph3-4	15.820	143.432	-1.438	-9.996	8.935	-4.211	25.424	3.749
Vivaldi3-6-1	3.269	7.154	-5.680	2.594	6.807	-1.677	2.078	2.932
Mean	2.194	20.097	-19.379	-28.376	57.795	-1.517		
Median	0.567	0.651	0.151	0.089	0.477	-0.556		

Table B.4: U values when ML model is trained on only works by Chopin.

Score File Name	**CTW**	**HMM**	**LSTM**	**LZMS**	**PPM**	**PST**	**Mean**	**Median**
BachBook1Fugue15	9.363	0.756	2.158	14.997	13.545	5.138	7.660	7.251
BachInvention12	-30.933	-1.533	5.305	-5.893	-15.057	26.294	-3.636	-3.713
BeethovenSonata13-2	1.584	-1.993	3.774	0.531	4.945	11.697	3.423	2.679
BeethovenSonata6-3	-9.730	0.058	2.989	-23.992	-3.849	5.174	-4.892	-1.895
ChopinMazurka41-1	2.582	4.155	-11.075	4.579	15.196	-342.221	-54.464	3.368
Corelli5-8-2	-7.158	-0.508	1.222	-77.267	-11.196	2.397	-15.418	-3.833
Grieg43-2	100.825	-0.653	-0.610	56.644	14.014	-22.235	24.664	6.702
Haydn33-3-4	0.257	0.747	-0.303	-0.465	0.318	3.564	0.686	0.288
Haydn64-6-2	-0.237	1.257	-4.695	-13.128	0.713	-2.331	-3.070	-1.284
LisztBallade2	-9.822	-8.908	-1.802	-48.866	-1.247	-0.425	-11.845	-5.355
MozartK331-3	3.235	3.222	1.759	1.424	1.972	9.481	3.516	2.597
MozartK387-4	-360.273	-0.076	-36.535	-1563.260	-188.723	-319.737	-411.434	-254.230
SchubertImprGFlat	1909.811	252.566	-2.182	113.835	7067.309	-1.407	1556.655	183.200
SchumannSymph3-4	93.621	39.735	-4.705	-5.161	18.111	-9.495	22.018	6.703
Vivaldi3-6-1	1.198	0.971	-6.087	1.501	6.712	-1.157	0.523	1.085
Mean	113.621	19.320	-3.386	-102.968	461.518	-42.351		
Median	1.198	0.747	-0.610	-0.465	1.972	-0.425		

Table B.5: U values when ML model is trained on only works by Debussy.

Score File Name	**CTW**	**HMM**	**LSTM**	**LZMS**	**PPM**	**PST**	**Mean**	**Median**
BachBook1Fugue15	5.064	1.833	3.123	3.186	8.173	4.054	4.239	3.620
BachInvention12	0.045	-2.700	5.143	1.397	1.195	2.314	1.232	1.296
BeethovenSonata13-2	0.684	0.583	1.721	-0.529	0.256	2.062	0.796	0.634
BeethovenSonata6-3	0.033	0.001	9.721	-2.020	-0.698	13.048	3.347	0.017
ChopinMazurka41-1	4.280	0.925	-13.085	7.605	18.297	-17.406	0.103	2.603
Corelli5-8-2	-2.077	-0.040	1.573	-0.802	-0.102	1.467	0.003	-0.071
Grieg43-2	2.775	1.017	-1.650	1.380	20.062	-13.618	1.661	1.198
Haydn33-3-4	2.443	3.662	-0.648	8.154	4.707	-0.726	2.932	3.052
Haydn64-6-2	1.149	2.841	0.666	0.128	1.197	3.141	1.520	1.173
LisztBallade2	-0.903	1.898	3.036	-0.084	4.525	3.796	2.045	2.467
MozartK331-3	-0.011	0.078	4.880	-3.224	0.931	2.001	0.776	0.504
MozartK387-4	-1.935	-0.175	-8.449	-407.364	-0.530	-590.247	-168.117	-5.192
SchubertImprGFlat	76.287	7.024	-1.737	279.232	36.621	3.501	66.821	21.823
SchumannSymph3-4	2.264	3.943	-2.010	-2.452	1.409	-2.127	0.171	-0.301
Vivaldi3-6-1	-0.178	0.029	-1.649	1.045	-0.027	0.354	-0.071	0.001
Mean	5.995	1.395	0.042	-7.623	6.401	-39.226		
Median	0.684	0.925	0.666	0.128	1.197	2.001		

Table B.6: U values when ML model is trained on only works by Dvorak.

Score File Name	**CTW**	**HMM**	**LSTM**	**LZMS**	**PPM**	**PST**	**Mean**	**Median**
BachBook1Fugue15	5.065	3.092	2.922	6.195	10.069	6.533	5.646	5.630
BachInvention12	-75.376	-11.368	8.680	-48.948	-56.603	8.462	-29.192	-30.158
BeethovenSonata13-2	0.004	0.014	2.884	-15.234	0.336	15.547	0.592	0.175
BeethovenSonata6-3	-0.909	-1.775	3.344	-169.279	-0.272	8.124	-26.794	-0.591
ChopinMazurka41-1	14.227	0.376	-54.751	-0.629	9.163	-50.371	-13.664	-0.126
Corelli5-8-2	-11.588	-0.707	0.254	-19.718	-14.462	2.289	-7.322	-6.148
Grieg43-2	23.949	3.936	-2.778	12.277	28.704	-5.654	10.072	8.107
Haydn33-3-4	5.924	2.564	0.474	10.406	9.767	0.483	4.936	4.244
Haydn64-6-2	4.177	1.295	-3.160	4.076	1.636	0.902	1.488	1.466
LisztBallade2	-9.440	0.615	0.292	-7.920	-11.589	3.154	-4.148	-3.814
MozartK331-3	0.587	0.214	6.269	0.252	1.334	4.582	2.207	0.961
MozartK387-4	-1.475	-1.285	-69.958	-1.631	-12.345	-107.813	-32.418	-6.988
SchubertImprGFlat	4.200	-0.764	15.038	5.781	1.260	33.584	9.850	4.991
SchumannSymph3-4	-4.350	0.020	-1.875	-7.061	-4.262	-20.770	-6.383	-4.306
Vivaldi3-6-1	3.616	-0.221	0.527	0.855	2.152	-0.022	1.151	0.691
Mean	-2.759	-0.266	-6.122	-15.372	-2.341	-6.731		
Median	0.587	0.020	0.474	-0.629	1.260	2.289		

Table B.7: U values when ML model is trained on only works by Haydn.

Score File Name	**CTW**	**HMM**	**LSTM**	**LZMS**	**PPM**	**PST**	**Mean**	**Median**
BachBook1Fugue15	1.277	1.834	1.198	3.630	5.426	4.986	3.059	2.732
BachInvention12	-53.254	-86.410	7.687	-68.378	-65.228	9.531	-42.675	-59.241
BeethovenSonata13-2	-0.530	0.755	5.752	0.148	-0.410	16.369	3.681	0.452
BeethovenSonata6-3	-1.554	-6.325	3.659	-0.940	1.575	4.066	0.080	0.317
ChopinMazurka41-1	32.701	3.295	-6.053	12.330	6.686	-45.259	0.617	4.991
Corelli5-8-2	-4.767	-0.798	1.276	-88.400	-2.753	1.192	-15.708	-1.776
Grieg43-2	1.667	3.298	-2.862	-0.468	9.791	-10.204	0.204	0.600
Haydn33-3-4	4.390	1.432	0.663	16.325	7.419	-0.020	5.035	2.911
Haydn64-6-2	1.511	9.253	-0.145	1.572	0.578	0.736	2.251	1.123
LisztBallade2	-73.006	-2.841	0.640	-13.883	-36.644	2.064	-20.612	-8.362
MozartK331-3	0.428	0.020	10.974	1.053	0.997	2.926	2.733	1.025
MozartK387-4	-6.625	0.761	-39.962	-1079.720	-8.401	-603.174	-289.520	-24.181
SchubertImprGFlat	0.258	2.304	0.417	7.915	3.112	17.531	5.256	2.708
SchumannSymph3-4	-0.993	0.607	-2.424	-7.406	0.576	-18.527	-4.694	-1.708
Vivaldi3-6-1	5.265	0.533	-1.217	1.569	8.542	-0.529	2.361	1.051
Mean	-6.215	-4.819	-1.360	-80.977	-4.582	-41.221		
Median	0.258	0.755	0.640	0.148	0.997	1.192		

Table B.8: U values when ML model is trained on only works by Mozart.

Score File Name	**CTW**	**HMM**	**LSTM**	**LZMS**	**PPM**	**PST**	**Mean**	**Median**
BachBook1Fugue15	6.285	2.088	3.252	11.191	13.317	3.385	6.586	4.835
BachInvention12	-0.831	0.579	8.125	-2.025	0.226	4.542	1.769	0.402
BeethovenSonata13-2	2.260	0.085	7.078	3.726	3.348	4.339	3.473	3.537
BeethovenSonata6-3	0.557	0.061	3.539	-0.702	-0.265	6.777	1.661	0.309
ChopinMazurka41-1	9.463	3.525	-16.985	1.583	4.687	-37.548	-5.879	2.554
Corelli5-8-2	-2.654	-1.379	2.013	-18.718	-9.501	1.632	-4.768	-2.017
Grieg43-2	1.040	1.311	-2.903	-0.395	2.143	-32.795	-5.267	0.322
Haydn33-3-4	6.246	14.266	12.216	0.941	1.738	-0.530	5.813	3.992
Haydn64-6-2	0.548	2.404	-0.353	-7.318	-0.544	3.139	-0.354	0.098
LisztBallade2	-0.491	-0.161	1.639	2.516	-0.068	5.543	1.496	0.786
MozartK331-3	1.807	0.312	10.663	-3.409	1.000	2.726	2.183	1.404
MozartK387-4	-4.114	0.015	-38.431	-140.548	-15.049	-127.499	-54.271	-26.740
SchubertImprGFlat	9.197	-0.606	-0.590	1.284	7.040	19.629	5.992	4.162
SchumannSymph3-4	4.762	1.756	-2.417	0.152	2.050	-4.565	0.290	0.954
Vivaldi3-6-1	0.878	0.432	-0.482	1.391	2.479	0.020	0.786	0.655
Mean	2.330	1.646	-0.909	-10.022	0.840	-10.080		
Median	1.040	0.432	1.639	0.152	1.738	2.726		

Table B.9: U values when ML model is trained on only works by Prokofiev.

Score File Name	**CTW**	**HMM**	**LSTM**	**LZMS**	**PPM**	**PST**	**Mean**	**Median**
BachBook1Fugue15	0.646	1.183	0.918	0.390	2.537	5.906	1.930	1.050
BachInvention12	-4.602	-20.163	8.734	-18.001	-3.353	21.817	-2.595	-3.978
BeethovenSonata13-2	-0.021	0.105	3.506	-0.492	4.398	17.176	4.112	1.805
BeethovenSonata6-3	-1.588	-3.677	3.111	-150.893	-0.017	5.053	-24.668	-0.803
ChopinMazurka41-1	0.627	-0.111	-7.476	-0.493	-1.853	-27.750	-6.176	-1.173
Corelli5-8-2	-8.082	-3.497	1.026	-21.377	-11.333	1.014	-7.042	-5.790
Grieg43-2	-2.031	0.355	-6.733	-22.718	0.975	-15.088	-7.540	-4.382
Haydn33-3-4	10.829	16.972	1.074	4.112	2.998	0.965	6.158	3.555
Haydn64-6-2	2.130	1.863	-1.666	1.837	0.969	-0.232	0.817	1.403
LisztBallade2	-4.724	-0.411	-3.323	-78.071	-9.815	0.870	-15.912	-4.024
MozartK331-3	1.556	0.143	6.602	1.101	0.695	10.320	3.403	1.328
MozartK387-4	-2.856	-4.023	-6.388	-631.040	-13.278	-26.414	-114.000	-9.833
SchubertImprGFlat	520.541	110.120	-2.298	554.273	4098.296	1.410	880.390	315.330
SchumannSymph3-4	77.105	57.636	-2.882	12.772	13.165	-4.942	25.476	12.969
Vivaldi3-6-1	22.539	15.379	-0.925	23.237	48.954	-0.936	18.041	18.959
Mean	40.805	11.458	-0.448	-21.691	275.556	-0.722		
Median	0.627	0.143	-0.925	-0.492	0.969	0.965		

Table B.10: U values when ML model is trained on only works by Schumann.

Score File Name	**CTW**	**HMM**	**LSTM**	**LZMS**	**PPM**	**PST**	**Mean**	**Median**
BachBook1Fugue15	5.760	5.835	4.413	6.507	18.815	4.160	7.581	5.797
BachInvention12	-51.084	-13.878	20.452	-16.695	-32.324	29.240	-10.715	-15.287
BeethovenSonata13-2	3.634	1.914	2.676	4.914	2.326	-0.507	2.493	2.501
BeethovenSonata6-3	0.661	-16.560	3.601	1.027	2.189	1.204	-1.313	1.116
ChopinMazurka41-1	38.422	2.315	-4.191	6.470	19.052	-4.601	9.578	4.393
Corelli5-8-2	-4.198	2.193	0.821	-1.858	-1.456	2.678	-0.303	-0.317
Grieg43-2	-0.724	4.503	-1.009	0.267	-0.877	-8.215	-1.009	-0.800
Haydn33-3-4	1.792	1.061	0.969	9.671	11.377	2.320	4.532	2.056
Haydn64-6-2	-1.720	0.289	0.494	-57.123	-0.168	-3.347	-10.263	-0.944
LisztBallade2	-7.099	-2.837	-5.571	-93.199	-21.473	-0.636	-21.802	-6.335
MozartK331-3	1.301	0.146	12.527	0.353	2.356	7.032	3.952	1.828
MozartK387-4	-3044.949	-4.638	-34.527	-183.514	-206.721	-2.467	-579.469	-109.021
SchubertImprGFlat	2.218	46.437	-13.660	-50.416	2.390	5.484	-1.258	2.304
SchumannSymph3-4	7.770	3.100	-1.151	-8.795	1.370	-5.512	-0.536	0.109
Vivaldi3-6-1	14.648	11.548	-2.402	-4.732	15.748	-2.895	5.319	4.573
Mean	-202.238	2.762	-1.104	-25.808	-12.493	1.596		
Median	1.301	1.914	0.494	-1.858	2.189	-0.507		

Table B.11: U values when ML model is trained on only works by Wagner.

Appendix C

Evaluation of Structural Quality of Motifs for Subset Training

We present the entropy and R values for *Bioplazm.jpg* after training with various subsets of data. In order to conserve space, we only display the results for one input file which was chosen for no particular reason. We measure the pitch and rhythm entropy of motifs extracted from the training set, the best motifs discovered, and all of the candidate motifs extracted.

Bach Subset	**CTW**	**HMM**	**LSTM**	**LZMS**	**PPM**	**PST**	**Average**
training motif pitches	1.904	1.945	1.991	1.778	1.695	1.859	1.862
discovered motif pitches	1.75	2.087	2.532	2.293	1.758	2.089	2.085
candidate motif pitches	2.155	2.29	2.308	1.945	1.792	2.139	2.105
training motif rhythms	0.77	0.773	0.798	0.693	0.692	0.746	0.745
discovered motif rhythms	2.225	2.252	1.251	2.293	1.96	0.634	1.769
candidate motif rhythms	2.35	2.228	2.348	2.081	2.087	2.231	2.221
R	23.95	2.474	2.289	1.394	2.944	0.686	5.623

Table C.1: Entropy and R values for *Bioplazm.jpg* after training with only works by Bach.

Beethoven Subset	**CTW**	**HMM**	**LSTM**	**LZMS**	**PPM**	**PST**	**Average**
training motif pitches	1.788	1.956	1.957	1.885	1.787	1.829	1.867
discovered motif pitches	2.011	2.025	2.387	2.223	2.259	2.086	2.165
candidate motif pitches	1.964	2.308	2.308	2.158	1.959	2.137	2.139
training motif rhythms	0.931	1.000	1.004	0.968	0.937	0.938	0.963
discovered motif rhythms	2.118	2.244	1.158	2.279	2.117	0.622	1.756
candidate motif rhythms	2.198	2.345	2.346	2.346	2.196	2.233	2.277
R	1.459	1.540	0.317	27.548	2.457	0.257	5.596

Table C.2: Entropy and R values for *Bioplazm.jpg* after training with only works by Beethoven.

Brahms Subset	**CTW**	**HMM**	**LSTM**	**LZMS**	**PPM**	**PST**	**Average**
training motif pitches	1.872	1.833	1.998	1.873	1.938	1.711	1.871
discovered motif pitches	2.119	2.260	2.386	2.247	2.585	1.961	2.260
candidate motif pitches	2.137	1.963	2.307	2.134	2.289	1.792	2.104
training motif rhythms	0.919	0.906	0.969	0.915	0.937	0.860	0.918
discovered motif rhythms	2.056	1.921	1.154	2.287	2.252	0.410	1.680
candidate motif rhythms	2.231	2.197	2.348	2.230	2.225	2.080	2.219
R	100.227	1.625	0.123	194.79	6305.42	-2.773	1099.90

Table C.3: Entropy and R values for *Bioplazm.jpg* after training with only works by Brahms.

Chopin Subset	**CTW**	**HMM**	**LSTM**	**LZMS**	**PPM**	**PST**	**Average**
training motif pitches	1.849	1.810	1.998	1.774	1.993	1.912	1.889
discovered motif pitches	1.810	2.118	2.528	1.922	2.301	2.223	2.150
candidate motif pitches	2.138	1.965	2.307	1.987	2.308	2.156	2.143
training motif rhythms	0.928	0.925	1.000	0.904	0.985	0.962	0.951
discovered motif rhythms	2.287	2.261	1.243	2.322	2.696	0.837	1.941
candidate motif rhythms	2.232	2.203	2.347	2.233	2.345	2.351	2.285
R	2.291	5.305	2.319	49.880	605.359	0.590	110.957

Table C.4: Entropy and R values for *Bioplazm.jpg* after training with only works by Chopin.

Debussy Subset	**CTW**	**HMM**	**LSTM**	**LZMS**	**PPM**	**PST**	**Average**
training motif pitches	1.665	1.841	1.794	1.798	1.714	1.794	1.767
discovered motif pitches	2.126	2.387	2.363	2.225	2.013	2.222	2.223
candidate motif pitches	1.946	2.311	2.155	2.154	1.965	2.155	2.114
training motif rhythms	0.965	1.028	1.001	1.004	0.973	1.015	0.998
discovered motif rhythms	2.126	2.411	0.917	2.078	1.869	0.832	1.706
candidate motif rhythms	2.082	2.345	2.351	2.351	2.198	2.349	2.279
R	9.610	44.090	1.553	3011.436	160.849	0.228	537.961

Table C.5: Entropy and R values for *Bioplazm.jpg* after training with only works by Debussy.

Dvorak Subset	**CTW**	**HMM**	**LSTM**	**LZMS**	**PPM**	**PST**	**Average**
training motif pitches	1.837	1.898	1.837	1.841	1.692	1.783	1.815
discovered motif pitches	2.089	2.369	2.223	2.024	1.530	2.085	2.053
candidate motif pitches	2.156	2.308	2.157	2.156	1.941	2.136	2.142
training motif rhythms	1.103	1.144	1.116	1.100	1.023	1.083	1.095
discovered motif rhythms	2.564	2.534	0.837	2.421	2.292	0.618	1.878
candidate motif rhythms	2.348	2.346	2.347	2.346	2.080	2.226	2.282
R	6.098	2.745	1.041	8.078	0.183	0.189	3.056

Table C.6: Entropy and R values for *Bioplazm.jpg* after training with only works by Dvorak.

Haydn Subset	**CTW**	**HMM**	**LSTM**	**LZMS**	**PPM**	**PST**	**Average**
training motif pitches	1.612	1.869	1.946	1.862	1.959	1.796	1.841
discovered motif pitches	1.762	2.421	2.531	1.936	2.244	2.085	2.163
candidate motif pitches	1.792	2.160	2.307	2.154	2.307	2.138	2.143
training motif rhythms	0.884	1.008	1.032	0.987	1.032	0.969	0.985
discovered motif rhythms	1.710	2.564	1.249	2.279	2.163	0.619	1.764
candidate motif rhythms	2.082	2.349	2.348	2.351	2.347	2.226	2.284
R	1.132	4.257	1.779	16.533	964.971	-0.044	164.771

Table C.7: Entropy and R values for *Bioplazm.jpg* after training with only works by Haydn.

Mozart Subset	**CTW**	**HMM**	**LSTM**	**LZMS**	**PPM**	**PST**	**Average**
training motif pitches	1.815	1.827	1.877	1.745	1.747	1.749	1.793
discovered motif pitches	2.024	2.252	2.531	2.087	2.048	2.088	2.172
candidate motif pitches	2.155	2.291	2.308	2.134	2.136	2.139	2.194
training motif rhythms	1.006	1.021	1.044	0.983	0.985	0.984	1.004
discovered motif rhythms	1.882	2.585	1.249	2.087	2.122	0.629	1.759
candidate motif rhythms	2.353	2.226	2.348	2.229	2.230	2.230	2.269
R	92.121	20.220	0.561	96.088	69.524	0.194	46.451

Table C.8: Entropy and R values for *Bioplazm.jpg* after training with only works by Mozart.

Prokofiev Subset	**CTW**	**HMM**	**LSTM**	**LZMS**	**PPM**	**PST**	**Average**
training motif pitches	2.036	2.096	2.046	1.927	1.865	1.973	1.990
discovered motif pitches	2.077	2.324	2.532	2.086	2.260	2.219	2.250
candidate motif pitches	2.307	2.327	2.308	2.139	1.962	2.154	2.200
training motif rhythms	0.861	0.880	0.869	0.855	0.838	0.867	0.861
discovered motif rhythms	2.190	2.236	1.251	1.691	2.118	0.824	1.718
candidate motif rhythms	2.347	2.466	2.347	2.227	2.199	2.345	2.322
R	34.599	0.193	1.309	33.495	0.173	0.371	11.690

Table C.9: Entropy and R values for *Bioplazm.jpg* after training with only works by Prokofiev.

Schumann Subset	**CTW**	**HMM**	**LSTM**	**LZMS**	**PPM**	**PST**	**Average**
training motif pitches	1.855	1.854	2.041	1.899	1.962	1.894	1.918
discovered motif pitches	2.064	2.258	2.386	2.287	2.225	2.082	2.217
candidate motif pitches	1.964	1.954	2.307	2.136	2.155	2.132	2.108
training motif rhythms	0.834	0.844	0.897	0.855	0.857	0.857	0.857
discovered motif rhythms	2.119	2.258	1.154	1.644	2.078	0.609	1.644
candidate motif rhythms	2.201	2.200	2.344	2.230	2.352	2.235	2.260
R	10.409	1.962	1.005	190.796	667.539	-0.298	145.235

Table C.10: Entropy and R values for *Bioplazm.jpg* after training with only works by Schumann.

Wagner Subset	**CTW**	**HMM**	**LSTM**	**LZMS**	**PPM**	**PST**	**Average**
training motif pitches	1.916	1.914	1.902	1.834	1.731	1.790	1.848
discovered motif pitches	2.388	2.244	2.388	2.220	2.406	2.088	2.289
candidate motif pitches	2.307	2.308	2.306	2.155	1.962	2.140	2.196
training motif rhythms	1.219	1.229	1.218	1.182	1.153	1.172	1.195
discovered motif rhythms	2.388	2.244	1.161	2.563	2.262	0.736	1.892
candidate motif rhythms	2.346	2.349	2.348	2.350	2.200	2.231	2.304
R	1964.763	6835.629	0.115	0.551	5.652	0.317	1467.838

Table C.11: Entropy and R values for *Bioplazm.jpg* after training with only works by Wagner.

Appendix D

Inspirational Input Sources

We list the audio and image files used as inspirational inputs for our motif discovery system.

Image File Name	**Source**	**Description**
Motif.jpg	`enchgallery.com`	Fractal
Fociz.jpg	`enchgallery.com`	Fractal
Bioplazm2.jpg	`enchgallery.com`	Fractal
LightPaintMix.jpg	`enchgallery.com`	Fractal
Variation-Investigation.jpg	`enchgallery.com`	Fractal
Pollock-Number5.jpg	`conservapedia.com`	Painting
Dali-ThePersistenceofMemory.jpg	`conservapedia.com`	Painting
Monet-ImpressionSunrise.jpg	`conservapedia.com`	Painting
DaVinci-MonaLisa.jpg	`conservapedia.com`	Painting
Vermeer-GirlWithaPearlEarring.jpg	`conservapedia.com`	Painting
Landscape.jpg	`desktopwallpaperhd.net`	Photograph
Stonehenge.jpg	`desktopwallpaperhd.net`	Photograph
River.jpg	`desktopwallpaperhd.net`	Photograph
Fish.jpg	`desktopwallpaperhd.net`	Photograph
Bird.jpg	`desktopwallpaperhd.net`	Photograph

Table D.1: Image files used as inspirational inputs for our motif discovery system. A variety of images are chosen in order to extract varying musical information.

Audio File Name	**Source**	**Description**
Reunion2005.wav	`poets.org`	Poetry read aloud
Neverland.wav	`poets.org`	Poetry read aloud
Birdsong.wav	`archive.org`	Bird chirping
ThunderAndRain.wav	`archive.org`	Thunder and rain
SparklingWater.wav	`archive.org`	Sparkling water
TropicalRain.wav	`archive.org`	Tropical rain
PleasantBeach.wav	`archive.org`	Pleasant beach
ChallengerDisasterAddress.wav	`americanrhetoric.com`	Ronald Reagan's Challenger disaster address
InauguralAddress.wav	`americanrhetoric.com`	John F. Kennedy's inaugural address
MLKDream.wav	`americanrhetoric.com`	Martin Luther King's "I Have a Dream" speech
DarthVaderBreathing.wav	`soundboard.com`	Sound effect from *Star Wars* of Darth Vader breathing
R2D2.wav	`soundboard.com`	Sound effect from *Star Wars* of R2D2
Lightsabers.wav	`soundboard.com`	Sound effect from *Star Wars* of lightsabers
ChewbaccaRoar.wav	`soundboard.com`	Sound effect from *Star Wars* of Chewbacca roaring
Blasters.wav	`soundboard.com`	Sound effect from *Star Wars* of blasters

Table D.2: Audio files used as inspirational inputs for our motif discovery system. A variety of audio files are chosen in order to extract varying musical information.