Musical Motif Discovery in Non-musical Media

Daniel Johnson and **Dan Ventura**
Computer Science Department
Brigham Young University
Provo, UT 84602 USA
daniel.johnson@byu.edu
ventura@cs.byu.edu

Abstract

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. Unlike many generative models, the inspirational component allows the composition process to originate outside of what is learned from the training data. Candidate motifs are extracted from non-musical media such as images and audio. Machine learning algorithms select and return the motifs that most resemble the training data. This process is validated by running it on actual music scores and testing how closely the discovered motifs match the expected 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.

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 (1996) and Dubnov et al. (2003) 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 (Burton and Vladimirova 1999; Biles 1994) and swarms (Blackwell 2003). 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 large portion of work serves one of these two goals, but not both.

While many computational compositions lack either innovation or grounding, great human composers from the period of common practice and the early 20th century composed with both goals in mind. For instance, Beethoven's music pushes classical boundaries into the beginnings of romanticism. The operas of Wagner bridge the gap between tonality and atonality. Schoenberg's twelve-tone music pushes atonality to a theoretical maximum. Great composers of this period produce 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.

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. Olivier Messiaen's compositions mimic birdsong and have roots in theology (Bruhn 1997). 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. Many pop musicians are clearly inspired by relationships and social interactions. 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. (2012). 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.

Media Inspiration

Just as humans often rely on inspiration for their creative work, our motif discovery system relies on non-musical audio files for inspiration. Non-musical audio is a natural starting place for musical inspiration because audio and music both exist in the sound medium. We also generalize one step further by allowing our system to be inspired by other forms of media, specifically images. A human might look at a painting, understand its meaning, and compose a piece of music based on the way he feels about it. He might also feel inspired to compose a piece of music shortly after attending a speech, listening to a bird chirp, watching a movie, or reading poetry. Since computer technology has not yet matched the full capacity of humans in understanding events in the world, we begin with unsophisticated means for extracting musical inspiration from media (our precise methods are described in a later section).

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" (1976). 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. (2009) 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.

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 vastly different than the innovative components of 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 (1996). 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.

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, first-order Markov processes do not store enough information to represent longer musical contexts, while high-order Markov processes require intractable space and time.

This issue necessitates a variable order Markov model (VMM) in which variable length contexts are stored. Dubnov et al. (2003) implement a VMM for modeling music using a prediction suffix tree (PST). 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. (2004) compare results for several variable order Markov models (VMMs), including the PST. 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 (2012) propose the Variable-gram Topic model for modeling melodies, which employs a Dirichlet-VMM and is also shown to improve upon other VMMs.

Variable order Markov models are not the only extensions explored. Lavrenko and Pickens (2003) apply Markov random fields to polyphonic music. In these models, next-note prediction accuracies improve when compared to a traditional high-order Markov chain. Weiland et al. (2005) apply hierarchical hidden Markov models (HHMMs) in order to capture long-term dependencies in music. HHMMs are used to model both pitch and rhythm separately.

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.

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 (Hochreiter et al. 2001). Eck and Schmidhuber (2008; 2002) 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. Smith and Garnett (2012) 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.

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. (2012). 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.

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. For our purposes, semantic content in the audio files is ignored. The candidate motifs that are most probable according to the ML model are returned. This process is tested using different ML model classes over various audio input files. A high-level system pipeline is shown graphically in Figure 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.

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 of 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. Themes and motifs are represented using the Phrase class from the jMusic library. We also utilize core functionality from jMusic for reading, writing, and manipulating musical structures.[2]

Machine Learning Models

A total of six ML model classes are tested. These include four VMMs, an LSTM RNN, and an HMM. These model classes 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,[3] an LSTM found on Github is used,[4] and the HMM implementation is found in the Jahmm library.[5]

Each of the learned ML models is used on both pitches and rhythms separately. Each model contains 128 possible pitches (0-127) and 32 possible note durations (32nd note multiples up to a whole note). The set of inputs in the RNNs represents which note is played, and the set of outputs represents the next note in the sequence to be played. The RNNs train for a fixed number of iterations before halting. The HMMs are trained using the Baum-Welch algorithm for a fixed number of iterations. The VMMs are trained according to the algorithms presented by Begleiter et al. (2004).

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.[6] More precisely, we use the *aubionotes* Windows binary from version 0.4.0 of Aubio, *schmitt* pitch detection, *kl* onset detection, and a threshold of 0.5. 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: given a tempo of 120 beats per minute, note durations are quantized to a 32nd note value; and note pitches are restricted to MIDI note values in the range [55, 85] by adding or subtracting octaves until each pitch is in range.

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,[7] 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 its way 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 duration of the note is b 32nd notes. The pitches are restricted to MIDI note values in the range [55, 85] as they were for pitch-detected sequences. Quantization is not performed for edge-detected sequences, since all of the note durations are already multiples of 32nd notes.

Motif Discovery

After the string of notes are 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 motif length as l_max. All contiguous motifs of length

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

[2] http://explodingart.com/jmusic

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

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

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

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

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

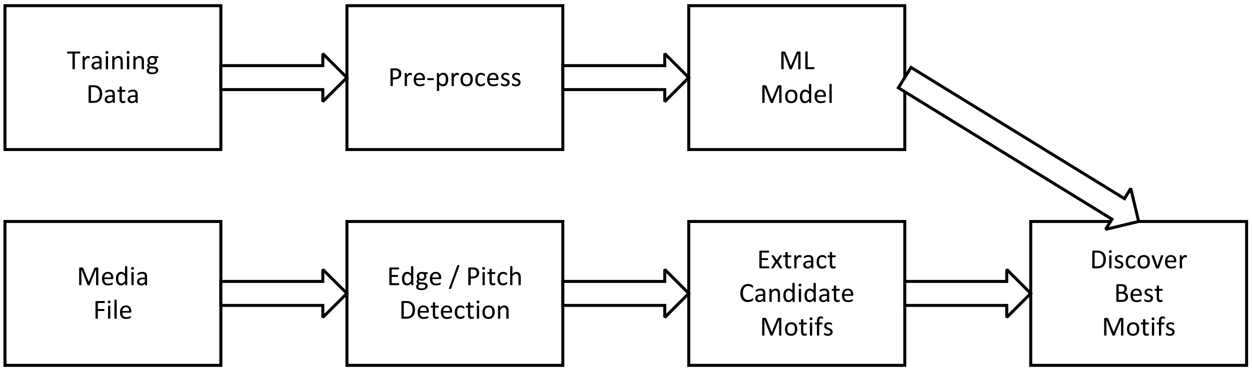


Figure 1: A high-level system pipeline for motif discovery. An ML model is trained on pre-processed music themes. Pitch detection is performed on an audio file or edge detection is performed on an image file in order to extract a sequence of notes. The sequence of notes is segmented into a set of candidate motifs, and only the most probable motifs according to the ML model are selected.

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.

Algorithm 1 *extract_candidate_motifs*

```
1: Input: $notes$, $l\_min$, $l\_max$
2: $candidate\_motifs \leftarrow \{\}$
3: for $l\_min \leq l \leq l\_max$ do
4:     for $0 \leq i \leq |notes| - l$ do
5:         $motif \leftarrow (notes_i, notes_{i+1}, ..., notes_{i+l-1})$
6:         $candidate\_motifs \leftarrow candidate\_motifs \cup motif$
7: 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:     $m^* \leftarrow \underset{m \in C}{\operatorname{argmax}}[norm(|m|)Pr(m|model)]$
6:     $best\_motifs \leftarrow best\_motifs \cup m^*$
7: return $best\_motifs$
```

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

$$P_{p+r}(m) = Pr(m_p){N_p}^{|m|} + Pr(m_r){N_r}^{|m|} \qquad (1)$$

where m is a motif, m_p is the motif pitch sequence, m_r is the motif rhythm sequence, N_p and N_r are constants, and $N_p > N_r$. In this paper we set $N_p = 60$ and $N_r = 4$. 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. For convenience, in what follows, we will use the simpler notation $Pr(m)$ as a short hand for $P_{p+r}(m)$ as well as the conditional notation $Pr(m|M)$ as a shorthand for $P_{p+r}(m|M)$, where $P_{p+r}(m|M)$ is computed as in Eq. 1, replacing the independent probabilities with their respective conditional counterparts.

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:

$$P_{equal} = \frac{1}{(l\_max - l\_min + 1)} \qquad (2)$$

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

$$P(l) = \sum_{m \in C, |m| = l} Pr(m|model) \qquad (3)$$

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

$$norm(l) = \frac{P_{equal}}{P(l)} \qquad (4)$$

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

Validation and Results

We perform three 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 learned ML model.

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. This evaluation was performed as a sanity check, and we did not select motifs or use machine learning at this stage. Instead, we compared the *entropy* (see Equation 5) of pitch-detected and edge-detected 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 a rhythm.

$$H(X) = -\sum_{i=1}^{n} Pr(x_i) \log_b Pr(x_i) \tag{5}$$

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. The average pitch and rhythm entropy measures were lower for pitch-detected and edge-detected sequences. A homoscedastic, two-tailed Student's t-test on the data shows statistical significance with p-values of 1×10^{-5} for pitches from images, 1×10^{-23} for rhythms from images, and 0.0003 for rhythms from audio files. In addition, although the p-value for pitches from audio files is not statistically significant (0.175), it is still fairly low. This suggests that there is potential for interesting musical content (Wiggins, Pearce, and Müllensiefen 2009) in the pitch-detected and edge-detected sequences even though the sequences originate from non-musical sources.

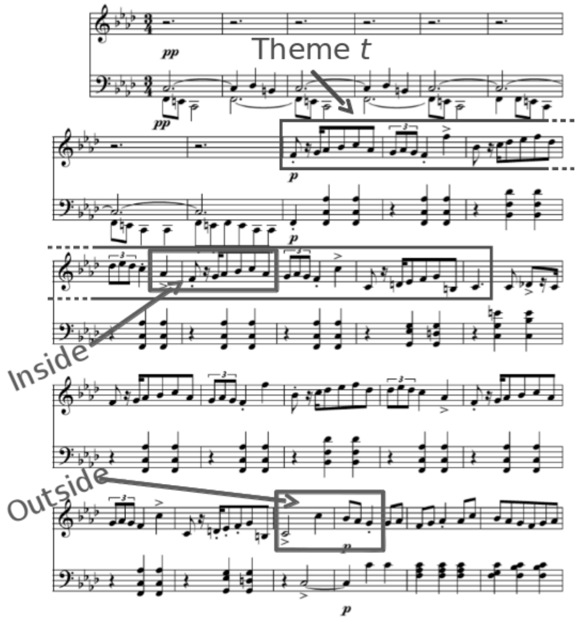


Figure 2: An example of a motif inside the theme and a motif outside the theme for a piece of music. The average normalized probability of the motifs inside the theme are compared to the average normalized probability of the motifs outside the theme.

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,[8] and the corresponding themes are removed from the training data set (taken from the aforementioned Electronic Dictionary of Musical 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.

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(s) for the score, and C_{-t} contains the remaining motifs. See Figure 2 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 given a model M:

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

Algorithm 3 *evaluate_discovery_process*

T is the set of all 9383 themes, V and S are sets of scores. Each $r \in V$ contains a set of themes $\{t_1 ... t_n\}$, $t_i \in T$ and each $s \in S$ contains a set of themes $\{u_1 ... u_k\}$, $u_i \in T$. $V \cap S = \emptyset$ and $\forall s \in S$ and $\forall r \in V$, $s \cap r = \emptyset$

.

```
 1: Input: $T$, $V$, $S$
 2: for each ML model class $\mathcal{M}$ do
 3:    $best = -\infty$
 4:    for each setting $p$ of $\mathcal{M}$'s hyperparameters do
 5:       $ave = 0$
 6:       for each score $s \in V$ do
 7:          learn $M_p$ using $T - s$ as training data
 8:          $ave = ave + U(s|M_p)$
 9:       $ave = ave/|V|$
10:       if $ave > best$ then
11:          $best = ave$
12:          $p_{best} = p$
13:    $p^*_{\mathcal{M}} = p_{best}$
14: for each ML model class $\mathcal{M}$ do
15:    for each score $r \in R$ do
16:       learn $M_{p^*_{\mathcal{M}}}$ using $T - r$ as training data
17:       results $\leftarrow U(r|M_{p^*_{\mathcal{M}}})$
18: return results
```

$$Q(S|M) = \frac{\sum\limits_{m \in S} norm(|m|) Pr(m|M)}{|S|} \qquad (6)$$

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

$$U(C|M) = \frac{Q(C_t|M) - Q(C_{-t}|M)}{\min\{Q(C_t|M), Q(C_{-t}|M)\}} \qquad (7)$$

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

Given our collected set T of 9383 themes, we use leave-one-out cross validation on a set V of music scores and their hand-labeled themes in order to fine-tune the ML model class hyperparameters to maximize U, as shown in Algorithm 3. For each score $s \in V$, we learn an ML model M from the model class $\mathcal{M}$ using $T - s$ as training data (line 7), and using the learned model we calculate the average U value for the set V (lines 8-9). We perform this validation under various hyperparameter configurations for all $s \in V$ for each ML model class (lines 2-6). After this is done, we select the hyperparameter configuration that results in the highest average value for U (lines 10-13). Finally, after these hyperparameters are tuned, we calculate U over a separate test set S of scores and themes (disjoint from V) for each model class (lines 14-17). The results are shown in Table 1.

Algorithm 4 *evaluate_motif_quality*

T is the set of all 9383 themes, F is a non-musical (inspirational) media file, $M_{p^*_{\mathcal{M}}}$ is a learned model

.

```
1: Input: $T$, $F$, $M_{p^*_{\mathcal{M}}}$
2: allmotifs $\leftarrow$ extract_candidate_motifs from $T$
3: $H_m$ = average_entropy(allmotifs)
4: candidates $\leftarrow$extract_candidate_motifs from $F$
5: $H_c$ = average_entropy(candidates)
6: best $\leftarrow$discover_best_motifs from candidates using
   model $M_{p^*_{\mathcal{M}}}$
7: $H_b$ = average_entropy(best)
8: results $\leftarrow R(T, best|M_{p^*_{\mathcal{M}}})$
9: return $H_m$, $H_c$, $H_b$, results
```

Given the data in the table, a case can be made that certain ML model classes can effectively discover thematic motifs with a higher probability than other motif candidates. Four of the six ML model classes 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 classes. 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. Except for LSTM and PST, all of the models are fairly robust by keeping negative U values to a minimum.

Evaluation of Structural Quality of Motifs

We also evaluate both the information content and the level of innovation of the discovered motifs, as shown in Algorithm 4. First, we measure the information content by computing *entropy* as we did before. We compare the entropy of the discovered motifs (lines 6-7) to the entropy of the candidate motifs (lines 4-5). 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 (lines 2-3). 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 2, with each column for each input file the result of running Algorithm 4 twice, once for pitch and once for rhythm. 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 GalwayKinnell-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 ob-

Score File Name	**CTW**	**HMM**	**LSTM**	**LZMS**	**PPM**	**PST**
BachBook1Fugue15.krn	4.405	4.015	3.047	2.896	11.657	4.951
BachInvention12.krn	-2.585	-5.609	26.699	1.078	0.534	13.191
BeethovenSonata13-2.krn	1.065	-0.145	7.769	8.876	4.973	9.182
BeethovenSonata6-3.krn	-0.715	-5.320	2.874	0.832	1.283	4.801
ChopinMazurka41-1.krn	6.902	0.808	-7.690	3.057	18.965	-24.363
Corelli5-8-2.krn	-6.398	-1.270	-0.692	-2.395	-1.166	1.690
Grieg43-2.krn	2.366	1.991	-2.622	0.857	8.800	-7.740
Haydn33-3-4.krn	14.370	2.370	1.189	6.155	8.475	0.841
Haydn64-6-2.krn	1.266	2.560	-1.092	0.855	1.809	-0.133
LisztBallade2.krn	-0.763	-0.610	-1.754	-0.046	1.226	0.895
MozartK331-3.krn	0.838	0.912	3.829	0.756	3.222	5.413
MozartK387-4.krn	-4.227	-0.082	-91.960	-2.127	-3.453	-31.614
SchubertImpromptuGFlat.krn	49.132	3.169	0.790	8.985	59.336	1.122
SchumannSymphony3-4.krn	0.666	2.825	-2.154	0.289	1.560	-6.830
Vivaldi3-6-1.krn	7.034	2.905	0.555	7.055	9.633	-0.367
Average	4.890	0.568	-4.081	2.475	8.457	-1.931

Table 1: U values for various score inputs and ML model classes. 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 thematic motifs and non-thematic motifs.

servation is that the rhythm entropy for some of the ML model classes is sometimes higher for the discovered motifs than it is for the candidate motifs. This suggests that thematic rhythms are often less predictable than non-thematic rhythms. However, the pitch entropy almost always tends to be lower for the discovered motifs than the candidate motifs. This suggests that thematic pitches tend to be more predictable.

Next, we measure the level of innovation of the best motifs discovered (line 8). We do this by taking a metric R (similar to U) using two Q statistics (see equation 6), where A is the set of 9383 themes from the training database and E is the set of discovered motifs.

$$R(A, E|M) = \frac{Q(A|M) - Q(E|M)}{\min\{Q(A|M), Q(E|M)\}} \qquad (8)$$

When R is greater than zero, A is more likely than E given the ML model M. 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 2. Except for PPM, all of the ML model classes 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 valuable insights for this system. We include six musical notations of motifs discovered by this system in Figure 3, and we invite the reader to listen to sample outputs at `http://axon.cs.byu.edu/motif-discovery`.

Conclusion and Future Work

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. This assumption is validated by observing high R values.

Additionally, the motif discovery system maintains compositional value by grounding it in 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.

Improvements and modifications can be made in the analysis and methodology of our system. We are currently preparing another manuscript which evaluates the difference between motifs discovered by our system and comparable random motifs. The results show that using (non-musical) media as inspiration for the motif discovery process is more efficient at producing “musical” motifs than is randomly generating “reasonable” motifs.

The discovered motifs are the contribution of this system. While work presented here is a proof-of-concept for the use of non-musical media sources as inspiration in creating musical motifs, more sophisticated techniques should be explored. In the future, we plan to utilize machine vision to extract meaning from images; we plan to study saccades from human subjects on various images in order to train the computer to see them in a more human, natural way; and we plan to incorporate digital signal analysis on audio files in order to hear audio more like a human would hear it. (While it is certainly not necessary for a computer to be inspired in the same way as a human might be, if the goal is to compose music that people can appreciate, it seems worthwhile to explore human-centric models of musical inspiration.)

In addition to improving the motif creation process, future work will investigate combining these motifs, adding harmonization, and creating full compositions. This work is simply the first step in a novel composition system. While there are a number of directions to take with this system as

Bioplazm2.jpg	**CTW**	**HMM**	**LSTM**	**LZMS**	**PPM**	**PST**	**Average**
pitch entropy training motifs	1.894	1.979	1.818	1.816	1.711	1.536	1.793
pitch entropy discovered motifs	2.393	2.426	1.944	1.731	2.057	1.759	2.052
pitch entropy candidate motifs	2.217	2.328	2.097	2.104	1.958	1.784	2.081
rhythm entropy training motifs	1.009	1.051	0.976	0.970	0.927	0.822	0.959
rhythm entropy discovered motifs	2.110	2.295	1.789	2.212	0.684	1.515	1.767
rhythm entropy candidate motifs	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**
pitch entropy training motifs	1.894	1.979	1.818	1.816	1.711	1.536	1.793
pitch entropy discovered motifs	1.974	2.074	2.143	1.833	2.027	1.675	1.954
pitch entropy candidate motifs	2.429	2.531	2.598	2.341	2.271	2.028	2.367
rhythm entropy training motifs	1.009	1.051	0.976	0.970	0.927	0.822	0.959
rhythm entropy discovered motifs	1.984	1.863	2.175	1.983	0.727	1.455	1.698
rhythm entropy candidate motifs	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**
pitch entropy training motifs	1.894	1.979	1.818	1.816	1.711	1.536	1.793
pitch entropy discovered motifs	2.076	1.884	1.881	1.652	2.024	1.586	1.850
pitch entropy candidate motifs	2.225	2.097	2.217	1.876	2.115	1.755	2.048
rhythm entropy training motifs	1.009	1.051	0.976	0.970	0.927	0.822	0.959
rhythm entropy discovered motifs	1.534	1.309	2.024	1.623	0.860	1.225	1.429
rhythm entropy candidate motifs	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

GalwayKinnell-Neverland.wav	**CTW**	**HMM**	**LSTM**	**LZMS**	**PPM**	**PST**	**Average**
pitch entropy training motifs	1.894	1.979	1.818	1.816	1.711	1.536	1.793
pitch entropy discovered motifs	1.823	2.480	2.132	1.773	1.997	1.701	1.984
pitch entropy candidate motifs	2.153	2.248	2.250	2.141	2.242	1.839	2.146
rhythm entropy training motifs	1.009	1.051	0.976	0.970	0.927	0.822	0.959
rhythm entropy discovered motifs	1.550	1.587	1.560	1.779	0.289	1.128	1.315
rhythm entropy candidate motifs	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 2: 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. Higher R values represent higher amounts of innovation from the training data.

a starting point, we are inclined to compose from the bottom up. 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, and multiple voices can be composed in a similar manner. Contrastingly, another system could 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 system could also be extended by including additional modes of inspirational input such as text or video. Our intent 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 creative.

References

Begleiter, R.; El-Yaniv, R.; and Yona, G. 2004. On prediction using variable order Markov models. *Journal of Artificial Intelligence Research* 22:385–421.

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

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

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

Burton, A. R., and Vladimirova, T. 1999. Generation of

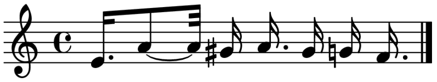
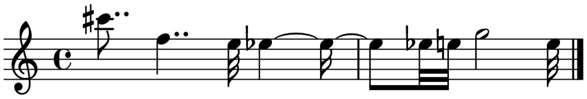
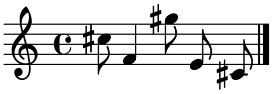
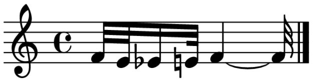
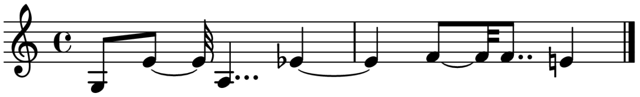
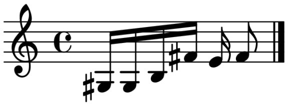
ML Model	Input File	Motif Discovered
CTW	*MLKDream.wav*	
HMM	*Birdsong.wav*	
LSTM	*Pollock-Number5.jpg*	
LZMS	*Lightsabers.wav*	
PPM	*Bioplazm2.jpg*	
PST	*GalwayKinnell-Neverland.wav*	

Table 3: Six motifs discovered by our system.

musical sequences with genetic techniques. *Computer Music Journal* 23(4):59–73.

Cardoso, A.; Veale, T.; and Wiggins, G. A. 2009. Converging on the divergent: The history (and future) of the international joint workshops in computational creativity. *AI Magazine* 30(3):15–22.

Cope, D. 1996. *Experiments in Musical Intelligence*, volume 12. AR Editions Madison, WI.

Dubnov, S.; Assayag, G.; Lartillot, O.; and Bejerano, G. 2003. Using machine-learning methods for musical style modeling. *Computer* 36(10):73–80.

Eck, D., and Lapalme, J. 2008. Learning musical structure directly from sequences of music. Technical report, University of Montreal, Department of Computer Science.

Eck, D., and Schmidhuber, J. 2002. Learning the long-term structure of the blues. In *Proceedings of the International Conference on Artificial Neural Networks*. 284–289.

Hochreiter, S.; Bengio, Y.; Frasconi, P.; and Schmidhuber, J. 2001. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In *A Field Guide to Dynamical Recurrent Neural Networks*. IEEE Press. 237–244.

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

Smith, B. D., and Garnett, G. E. 2012. Improvising musical structure with hierarchical neural nets. In *Proceedings of Eighth Artificial Intelligence and Interactive Digital Entertainment Conference*, 63–67.

Smith, R.; Dennis, A.; and Ventura, D. 2012. Automatic composition from non-musical inspiration sources. In *Proceedings of International Conference on Computational Creativity*, 160–164.

Spiliopoulou, A., and Storkey, A. 2012. A topic model for melodic sequences. *ArXiv E-prints*.

Weiland, M.; Smaill, A.; and Nelson, P. 2005. Learning musical pitch structures with hierarchical hidden Markov models. Technical report, University of Edinburgh.

White, J. D. 1976. *The Analysis of Music*. Prentice-Hall.

Wiggins, G. A.; Pearce, M. T.; and Müllensiefen, D. 2009. Computational modelling of music cognition and musical creativity. *Oxford handbook of computer music* 383–420.