Classical Music Composition Using Hidden Markov Models

by

Anna Yanchenko

Department of Statistical Science
Duke University

Date: ____________________________

Approved:

____
Sayan Mukherjee, Supervisor

____
Surya T Tokdar

____
Mike West

Thesis submitted in partial fulfillment of the requirements for the degree of
Master of Science in the Department of Statistical Science
in the Graduate School of Duke University
2017

Abstract

Classical Music Composition Using Hidden Markov Models

by

Anna Yanchenko

Department of Statistical Science
Duke University

Date: ____________________________
Approved:

____
Sayan Mukherjee, Supervisor

____
Surya T Tokdar

____
Mike West

An abstract of a thesis submitted in partial fulfillment of the requirements for
the degree of Master of Science in the Department of Statistical Science
in the Graduate School of Duke University
2017

Abstract

Hidden Markov Models are a widely used class of probabilistic models for sequential data that have found particular success in areas such as speech recognition. Algorithmic composition of music has a long history and with the development of powerful deep learning methods, there has recently been increased interest in exploring algorithms and models to create art. To this end, we explore the utility of Hidden Markov Models in composing classical music. Specifically, we train various Hidden Markov Models on piano pieces from the Romantic era and consider the models' ability to generate new pieces that sound like they were composed by a human. We evaluate the compositions based on several quantitative metrics that measure the originality, harmonic qualities and temporal structure of the generated piece. We additionally conduct listening evaluations with listeners of varying levels of musical background to assess the generated musical pieces. We find that Hidden Markov Models are fairly successful at generating new pieces that have largely consonant harmonies, especially when trained on original pieces with simple harmonic structure. However, we conclude that the major limitation in using Hidden Markov Models to generate music that sounds like it was composed by a human is the lack of global structure and melodic progression in the composed pieces.

Contents

List of Tables

List of Figures

List of Abbreviations and Symbols

Symbols

X	Observed states in a Hidden Markov Model.
Z	Hidden states in a Hidden Markov Model.
π	Initial state distribution for a Hidden Markov Model.
A	Transition matrix for a Hidden Markov Model.
B	Emission distribution for a Hidden Markov Model.
λ	Parameter vector for a Hidden Markov Model, $\lambda = (\pi, A, B)$.

Abbreviations

ACF	Autocorrelation Function
ARHMM	Autoregressive Hidden Markov Model
HMM	Hidden Markov Model
HSMM	Hidden Semi-Markov Model
LSTM	Long Short Term Memory
MIDI	Musical Instrument Digital Interface
NSHMM	Non-stationary Hidden Markov Model
PACF	Partial Autocorrelation Function
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
SCFG	Stochastic Context-Free Grammar

Acknowledgements

I would like to thank my advisor Professor Sayan Mukherjee for his advice on the modeling aspects of this thesis and my committee members, Professor Surya Tokdar and Professor Mike West. I wish to additionally thank Professor Jeff Miller for his advice at the very early stages of this project and his help with the development of the two hidden state Hidden Markov Model. I am thankful to Mike Kris for his extensive help with the musical aspects of this thesis. I wish to thank all of the people that listened to the generated musical pieces and provided insightful and thorough evaluations. Finally, I wish to thank my family for their endless support and encouragement.

1

Introduction

Hidden Markov Models (HMMs) are a class of probabilistic models for time series data that are widely used in many domain areas, in large part due to their flexibility and relative efficiency of parameter estimation (Ghahramani and Jordan (1997)). While speech recognition is one of the original domain areas where HMMs have been and continue to be widely used (Rabiner (1989)), HMMs are used in a variety of applications, ranging from computational biology (Krogh et al. (1994)) to the classification of music (Pikrakis et al. (2006)).

The increase in computational power in recent years has had a large impact on both the study of mathematical and statistical properties of music, as well as on algorithmic composition. Statistical techniques such as time series analysis of periodograms for different instruments, hierarchical models for the decompositions of melody and harmony, Markov chains to estimate the probability of transitions between pitch intervals and discriminant analysis for identifying pitch and historic periods of composition, have all been employed to study classical music (Beran (2004)). Algorithmic composition, on the other hand, has long been a problem of interest from a mathematical and musical standpoint. The first composition completely generated by a computer was the “Illiac Suite” produced from 1955 - 1956 (Nierhaus (2009)), and since then, work aimed at composing music algorithmically has only increased. Nierhaus (2009) summarizes efforts in algorithmic composition using a wide variety of techniques, such as HMMs to compose jazz improvisations and counterpoint to a cantus firmus and to classify folk music pieces, generative grammars to generate folk song rounds and children’s songs and artificial neural networks to both harmonize existing melodies and to generate melodies over chord progressions.

Additional methods of algorithmic composition include the work of Cope (1991), who used transition networks to perform Experiments in Musical Intelligence and to generate compositions that conform to a given musical style. Whorley and Conklin (2016) used a multiple viewpoint system with sequential and vertical components and a few general rules of harmony with a iterative random walk technique applied to four-part harmonies in hymn tunes by Vaughan Williams. Dirst and Weigend (1993) explored data driven approaches to analyzing and completing Bach’s last, unfinished fugue. Meehan (1980) discussed efforts to develop models for tonal music theory using artificial intelligence.

HMMs have also been utilized for a variety of musical analysis and algorithmic composition tasks. A survey of previous work in using Markov models for music composition appears in Ames (1989). More recently, Suzuki and Kitahara (2014) used Bayesian networks to harmonize soprano melodies, Weiland et al. (2005) modeled the pitch structure of Bach chorales with Hierarchical Hidden Markov Models and Pikrakis et al. (2006) used variable duration HMMs to classify musical patterns. Allan and Williams (2004) used HMMs to harmonize Bach chorales, where the visible states were the notes of the melody and the hidden states corresponded to chords. Models were evaluated based on the probability assigned by the model to Bach’s actual harmonizations.

Artificial neural networks and deep learning methods have recently exploded in popularity and have been applied to, and found success in, the generation of visual art, such as the neural style work of Gatys et al. (2015), the modeling of raw audio (van den Oord et al. (2016)) and the composition of music. As applied to music composition, Recurrent Neural Networks (RNNs), capable of modeling sequential data, are of particular interest. Mozer (1994) and Eck and Schmidhuber (2002) explored variants of RNNs for music composition and Boulanger-Lewandowski et al. (2012) used variants of RNNs for modeling polyphonic (multiple-voice) music. Hadjeres and Pachet (2016) developed the interactive DeepBach system, that uses Long-Short Term Memory (LSTM) units to harmonize Bach chorales with no musical knowledge built into the system. Johnson (2015) explored the use of RNNs for composing classical piano pieces. Vincent (2016) reported on a recent concert featuring work composed by artificial intelligence, where the composed pieces sought to emulate the styles of known composers and were generated using RNNs. Google's Magenta project (Magenta (2016), Developers (2017)) is interested in generating art and media through deep learning and machine learning and maintains an open source code repository and blog detailing their efforts. In general, there is a great amount of interest in exploring the ability of deep learning and other modeling techniques to compose music and create art.

While HMMs have been used for music classification and the harmonization of melodic lines, their use for the composition of melodies has been more limited. Due to the ease of implementation of HMMs, especially compared to RNNs, their flexibility and their ability to model complex time series and signals such as speech, we explored the utility of HMMs for composing classical piano pieces from the Romantic era. We were primarily interested in the ability of HMMs to generate piano pieces that sounded like they were composed by a human. We were particularly interested in three related sub-questions:

1. How original are the generated pieces?

   While we wanted the generated pieces to sound like piano pieces composed in the Romantic era, we did not want the models to overfit the training pieces to the extent that the new piece was exactly the same as the original, with no originality.

2. How dissonant or consonant are the harmonies produced by the generated piece as compared to the original piece?

3. How much global structure and melodic progression over the extent of the piece is there in the generated pieces?

In this thesis, we find that HMMs are fairly successful at generating new pieces that have mostly consonant harmonies and sound like they could have been composed by a human, especially when trained on original pieces with simple harmonic structure. However, we find that the HMMs considered are unable to capture sufficient global structure to create melodic progression over the entirety of a generated piece.

2

Methods

2.1 HMM Overview

A Hidden Markov Model (HMM) is a doubly stochastic process, where the underlying process is non-observable, or hidden, and follows a Markov process. The hidden process is only observed through the second stochastic process that produces the observable states (Rabiner (1989), Yu (2010)). An HMM can be factorized as

$$p(x_{1:n}, z_{1:n}) = p(z_1)p(x_1|z_1)\prod_{t=2}^{n} p(z_t|z_{t-1})p(x_t|z_t) \tag{2.1}$$

where $x_{1:n}$ is the sequence of observed states and $z_{1:n}$ is the sequence of hidden states. $p(z_1)$ is called the initial state distribution, $p(z_t|z_{t-1})$ denotes the transition probabilities of the hidden states and $p(x_t|z_t)$ is the emission distribution. An HMM respects the directed graph in Figure 2.1 (Miller (2016a)).

2.1.1 Notation

We assume that the hidden states can take one of m possible discrete values, that is $Z_t \in \{1, \ldots, m\}$. Let n be the total number of observations in the sequence. Following the notation of Rabiner (1989), we denote the transition probabilities as

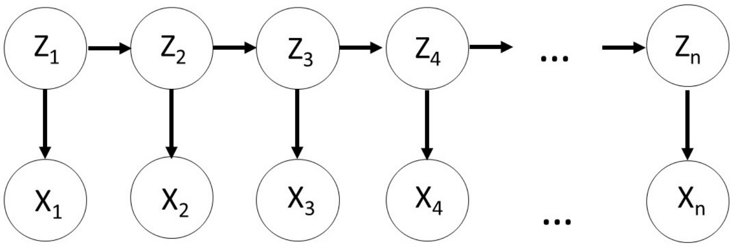

FIGURE 2.1: Directed graph for a first order HMM.

a matrix, $A = \{a_{ij}\}$, where $a_{ij} = P(Z_{t+1} = j|Z_t = i)$, $i, j \in \{1, \ldots, m\}$. The emission distribution can be written as $B = \{b_i(x_t)\}$, where $b_i(x_t) = P(x_t|Z_t = i)$, $i \in \{1, \ldots, m\}$. Finally, we denote the initial state distribution as $\pi_i = P(Z_1 = i)$ for $i = 1, \ldots, m$. Then, we can represent an HMM with these three parameters as

$$\lambda = (\pi, A, B) \tag{2.2}$$

2.1.2 Three Problems for HMMs

Rabiner (1989) provides an excellent tutorial on HMMs and describes three problems for HMMs:

1. (Likelihood Problem) Given an HMM $\lambda = (\pi, A, B)$ and an observation sequence $x_{1:n}$, find the likelihood $p(x_{1:n}|\lambda)$.

2. (Decoding Problem) Given an HMM $\lambda = (\pi, A, B)$ and an observation sequence $x_{1:n}$, find the hidden state sequence $z_{1:n}$ that "best" explains the observations.

3. (Learning Problem) Given an observation sequence $x_{1:n}$, learn the model parameters $\lambda = (\pi, A, B)$ that maximize $p(x_{1:n}|\lambda)$. (Jurafsky and Martin (2009))

All three problems can be solved by dynamic programming algorithms; the likelihood problem by the Forward-Backward Algorithm, the decoding problem by the Viterbi Algorithm and the learning problem by the Baum-Welch Algorithm (Baum

(1972)), a special case of the Expectation-Maximization algorithm (Dempster et al. (1977)). The Forward-Backward Algorithm, Viterbi Algorithm and Baum-Welch Algorithm are detailed in Appendix A.

After the parameters for the HMM are learned, a new sequence can be generated as follows:

1. Sample z_1 from the initial distribution π. Assume $z_1 = i$.
2. Sample x_1 from the emission distribution based on $z_1 = i$, that is, set $x_1 = l$ with probability $b_i(l)$.
3. For $t = 2, \ldots, n$, transition from hidden state $z_{t-1} = i$ to $z_t = j$ with probability a_{ij} and emit observed state $x_t = l$ with probability $b_j(l)$. (Rabiner (1989))

As is often the case in speech recognition applications, further constraints can be placed on the transition matrix A to reflect domain knowledge about the system. Unlike speech applications where the hidden states often have clear interpretations (such as a noise-free, “pure” signal), there is not a clear interpretation of the hidden states in this application of using HMMs for music generation. The hidden states represent some aspects of the musical theory that the composer used to compose the original piece, but the observed notes are not some noisy version of this underlying process. As a result, apart from the left-right models discussed below, we did not place any constraints on the transition matrix A that conformed to some aspect of music theory.

2.2 Models

2.2.1 First Order and First Order Left-Right HMM

The most basic model considered was the first order HMM described in section 2.1 above, where the order of the model indicates the order of the Markov process governing the hidden states. The left-right model is a variant of the basic first order HMM with the additional constraint on the transition matrix A that $a_{ij} = 0, \quad \forall j < i$, and is often used in applications like speech recognition where the properties of the signal change over time (Rabiner (1989)).

2.2.2 Higher Order Models

A limitation of the first order HMM is that the hidden states only depend on the previous hidden state, so longer-term dependencies cannot be modeled well. As we expect longer term dependencies to occur in music (i.e. the next note is highly likely to depend on more notes than just the immediately preceding one), HMMs with second and third order Markov processes governing the hidden state transitions were considered, as well as their left-right extensions. The Baum-Welch algorithm and generative process outlined for the first order HMM can be easily extended to the higher order models, however, now there is an expanded parameter space for the models (Mari and Schott (2001)).

2.2.3 Two Hidden State HMM

Due to the desire to include more hierarchical structure in the model and to allow for multiple hidden processes to change at potentially different rates over time, we considered an HMM that had two hidden states for each observed state (Miller (2016b)). The addition of this second hidden process allowed for the potential for the model to capture additional aspects of the hidden state dynamics, corresponding to the original compositional process. The graphical model for the HMM with two hidden states is below (Figure 2.2) with a derivation of the Baum-Welch Algorithm for this model in Appendix B.

2.2.4 Hidden Semi-Markov Model

The Hidden Semi-Markov Model (HSMM), also called the variable length or variable duration HMM, extends the HMM framework to allow for variable duration for each

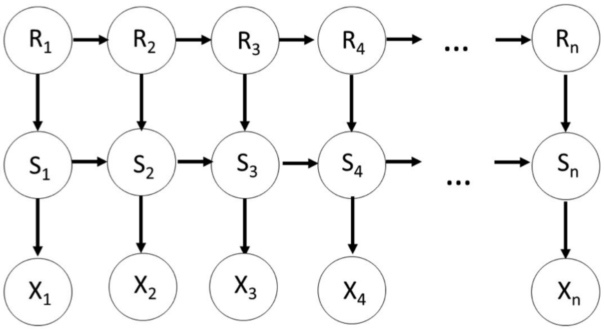

FIGURE 2.2: Directed graph of the HMM with two hidden states. Both the $R_{1:n}$ and the $S_{1:n}$ are hidden states.

hidden state, that is, the underlying process for the hidden states is now a semi-Markov chain. Each hidden state remains in the same state for a duration of time d (and emits d observed states), where d is a random variable (Yu (2010), Murphy (2002)). In music, we might expect for a piece to remain in the same state over the course of a few bars (perhaps over the course of a motif), so the HSMM seeks to model this variable state duration explicitly.

2.2.5 Factorial HMM

Ghahramani and Jordan (1997) extend the basic HMM to a factorial HMM that allows for distributed state representations, which decompose the underlying state into the different dynamic processes that it is composed of. While several models with varying degrees of dependence between the different dynamic processes are considered in Ghahramani and Jordan (1997), as the simplest case of a factorial HMM, we considered three completely independent underlying hidden processes of different numbers of possible hidden states and then averaged their generated observation sequences. The use of a factorial HMM, even in this very simple, model averaging form, was again an attempt to model the different dynamic processes that combine to form a generated piece of music.

2.2.6 Layered HMM

Layered HMMs are related to the idea of Stacked Generalization, where layers of HMMs are learned one on top of the other (Oliver et al. (2004)). The lowest level of the HMM is a first order HMM that is trained as usual. After the parameters are learned using the Baum-Welch Algorithm, the Viterbi Algorithm is used to find the most likely sequence of hidden states. These hidden states are then used as the observed states for another HMM, which is again trained with the Baum-Welch Algorithm and this layering process can be continued. We considered three layers of HMMs, all with the same number of possible hidden states. This representation is again introducing hierarchical structure to model different processes occurring at different rates.

2.2.7 Autoregressive HMM

Autoregressive models are extremely common in time series applications. Since each note is expected to depend not just on the hidden state at time t, but also the previous note at time $t-1$, we considered an Autoregressive HMM (ARHMM) to model this autoregressive structure in the observed states as well as the hidden states (Rabiner (1989), Guan et al. (2016)).

2.2.8 Non-stationary HMM

A shortcoming of traditional HMMs is that they cannot model state durations (a shortcoming which HSMMs also seek to address). Non-stationary HMMs (NSHMMs) explicitly model state transition probabilities as functions of time, and are equivalent to HSMMs but can be slightly more tractable (Djuric and Chun (1999), Sin and Kim (1995)). Djuric and Chun (2002) present a Markov Chain Monte Carlo (MCMC) algorithm for the estimation of parameters in NSHMMs.

2.3 Music Theory

Laitz (2003) and Gauldin (2004) provide in-depth treatments of many aspects of musical theory. Romantic era music is considered tonal music, meaning that the music is oriented around and pulled towards the tonic pitch. For this thesis, we primarily consider a high-level view of the intervals and chords present in the generated pieces. Briefly, there are two types of musical intervals, the melodic interval, where two notes are sounded sequentially in time, and the harmonic interval, where two or more notes are sounded simultaneously. The simple intervals (intervals that are less than an octave) in increasing number of half steps between notes are perfect unison (0 half steps between notes, the same pitch twice), minor second, major second, minor third, major third, perfect fourth, tritone (6 half steps between notes), perfect fifth, minor sixth, major sixth, minor seventh, major seventh and the octave (12 half steps between notes). The octave, perfect unison and the third, fifth and sixth intervals are consonant intervals, while the second and seventh intervals are considered dissonant. The perfect fourth is considered dissonant in some contexts and consonant in others. The perfect unison, octave, perfect fourth and perfect fifth are perfect consonances, in that they are stable intervals that do not need to resolve to a more stable pitch. Thirds and sixths are imperfect consonances and are moderately stable; they again do not need a resolution. Dissonant intervals on the other hand need to be resolved to consonant intervals.

A chord is a combination of three or more different pitches, and the predominant chords occurring in the Romantic era were the triad and the seventh chord. The triad is composed of a root pitch and the third and fifth above the root, while a seventh chord is the root, third, fifth and seventh. Chords can be closed or open depending on the spacing between pitches in the higher and lower registers.

2.4 Musical Training Pieces

All models were trained on ten piano pieces from the Romantic period. All original training pieces considered were either originally composed for piano or were arranged for piano. The Romantic period of music lasted from the end of the eighteenth century to the beginning of the twentieth century. This period was marked by a large amount of change in politics, economics and society. In general, music from the Romantic era tended to value emotion, novelty, technical skill and the sharing of ideas between the arts and other disciplines. Some key themes of Romanticism were the emphasis of emotion over reason, national pride, the extolling of common people and an interest in the exotic. Romantics emphasized the individual, national identity, freedom and nature and the Romantic era saw an increase in virtuosity in musical performance (Warrack (1983)).

The piano as an instrument also changed considerably during the Romantic era, resulting in new trends in composition for the instrument. Technical and structural improvements to the piano lead to an instrument that was capable of a larger range of pitches and dynamics as compared to the pianoforte of the seventeenth and early eighteenth centuries. Improvements in the sustaining pedal in particular enabled the piano to create a more dramatic and sustained sound. Furthermore, the piano saw an increased presence in musical, commercial and social circles in the Romantic era (Ratner (1992)).

The original training pieces were selected to have a range of keys, forms, meters, tempos and composers. Piano pieces were chosen for their simplicity, as only one instrument needed to be modeled, as opposed to orchestral pieces, where multiple, dependent instruments would need to be modeled. In addition to training on pieces originally composed for piano, several hymn tunes from the Romantic era arranged for piano were also considered. The success of previous work in harmonizing and generating Bach chorales (for example, Hadjeres and Pachet (2016)) and other hymn tunes (Allan and Williams (2004)) as well as the simple form and melody of hymn tunes suggested their inclusion as training pieces. For longer pieces, an excerpt from the beginning of the piece that ended in a cadence was considered. The ten original training pieces, as well as their keys and time signatures are listed in Table 2.1 below.

Table 2.1: Summary of the Romantic era piano training pieces modeled, including the composer, key and time signature of each piece.

Composer	Piece	Key	Time Signature
Beethoven	Piano Sonata No. 14 (Moonlight Sonata), 1st Movement	C# minor	3/4
Chopin	Piano Sonata No. 2, 3rd Movement (Marche funebre)	Bb minor	4/4
Mussorgsky	Pictures at an Exhibition, Promenade - Gnomus	Bb major	6/4
Mendelssohn	Song without Words Book 1, No. 6	G minor	6/8
Mendelssohn	Song without Words Book 5, No. 3	A minor	4/4
Liszt	Hungarian Rhapsody, No. 2	C# minor	4/4
Tchaikovsky	The Seasons, November - Troika Ride	E major	4/4
Beethoven	Ode to Joy (Hymn Tune)	D major	4/4
Hopkins	We Three Kings (Hymn Tune)	G major	3/4
Mendelssohn	Hark! The Herald Angels Sing (Hymn Tune)	F major	4/4

2.5 Metrics

Based on the desired characteristics of the generated pieces and following the evaluation measures used in Coffman (1992), Whorley and Conklin (2016), Suzuki and Kitahara (2014) and Hadjeres and Pachet (2016), several quantitative measures were considered to numerically evaluate the generated musical pieces. These quantitative measures were used to obtain a baseline measure of how well the models were performing at generating original, musically and temporally coherent pieces and were used as rankings to select the pieces that were evaluated by human listeners and to compare models. For this thesis, we were primarily interested in how original, musically coherent and temporally structured the generated pieces were, and we considered metrics that aimed to measure some aspect of each of these three areas of focus.

2.5.1 Originality Metrics

Entropy

Entropy was used as a proxy for the originality of both the original piece and the generated piece, following Coffman (1992), where entropy is considered as a measure of how predictable a musical piece is. Entropy is a measure of information and can be calculated as (Jurafsky and Martin (2009)):

$$H(X) = \sum_{x \in X} p(x) \log_2 p(x) \tag{2.3}$$

X is the random variable with probability distribution $p(X)$ and the entropy of X is $H(X)$ and is measured in bits. The entropy for the original musical pieces can be found in Table 2.2. As expected, the simple hymn tunes are among the pieces with the lowest entropy of those considered, while the complex and more “original” pieces such as Liszt’s Hungarian Rhapsody No. 2 and Mussorgsky’s Promenade - Gnomus from Pictures at an Exhibition had higher entropy.

Mutual Information

For two random variables X and Y, the mutual information between them can be calculated as

$$\mathbb{I}(X;Y) = \sum_{x} \sum_{y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \tag{2.4}$$

where the mutual information can be interpreted as measuring how much the uncertainty in the random variable X is reduced by knowing about the random variable Y, or vice versa (Murphy (2012)). The mutual information is 0 if and only if the

Table 2.2: Entropy for the ten original training pieces considered, ordered from lowest to highest entropy.

Composer	Piece	Entropy
Beethoven	Ode to Joy (Hymn Tune)	2.328
Hopkins	We Three Kings (Hymn Tune)	2.521
Chopin	Piano Sonata No. 2, 3rd Movement (Marche funebre)	2.759
Mendelssohn	Hark! The Herald Angels Sing (Hymn Tune)	2.780
Mendelssohn	Song without Words Book 5, No. 3	2.897
Beethoven	Piano Sonata No. 14 (Moonlight Sonata), 1st Movement	3.000
Tchaikovsky	The Seasons, November - Troika Ride	3.063
Mendelssohn	Song without Words Book 1, No. 6	3.227
Liszt	Hungarian Rhapsody, No. 2	3.436
Mussorgsky	Pictures at an Exhibition, Promenade - Gnomus	3.504

random variables X and Y are independent. While we expected the generated piece to carry a large amount of information about the original piece, we did not want this amount of information to be too large, which would indicate that the models were potentially overfitting the training piece and were generating pieces that were not original. We thus wanted to see generated pieces that had low mutual information with the original piece, indicating a generated piece that deviated more from the original piece and was thus more creative in some sense.

Minimum Edit Distance

The minimum edit distance is the minimum number of insertions, deletions and substitutions that are necessary to transform one string into another (Jurafsky and Martin (2009)). A generated piece that was very similar to the original piece would have a lower minimum edit distance than a generated piece that was quite different from the original piece.

2.5.2 Musicality Measures

Normalized Count of Dissonant Intervals

The minor second, major second, minor seventh and major seventh intervals are considered dissonant (Gauldin (2004)). The number of dissonant melodic and harmonic intervals in the generated piece were counted and normalized by the length of the piece. We expected the amount of dissonance in the generated pieces to be similar to the amount of dissonance in the original training piece. In particular, if the original piece contained no dissonance or very little dissonance (as was the case for all of the pieces considered, particularly the hymn tunes), we did not want a lot of dissonance in the generated piece, as unresolved dissonance was not common in Romantic era pieces.

Normalized Count of Large Intervals

Following the metrics considered in Whorley and Conklin (2016) and Suzuki and Kitahara (2014) and general composition practice of the Romantic era, we did not expect to see many notes in either the melodic line or the bass line that were more than an octave apart from the previous note in their line. We counted the number of such large interval jumps that occurred in the generated piece and normalized the count by the length of the piece. The original pieces tended to have few, if any, melodic or harmonic intervals larger than an octave.

Distribution of Note Pitches

The distribution of note pitches was compared between the original and generated pieces by considering the normalized count of each pitch. We expected that pitches that were used less in the original piece to also be less prevalent in the generated pieces.

2.5.3 Temporal Structure

Autocorrelation Function

For a time series $\{y_t\}$, the autocorrelation function (ACF) is defined as

$$\rho(s,t) = \frac{\gamma(s,t)}{\sqrt{\gamma(s,s)\gamma(t,t)}} \tag{2.5}$$

where $\gamma(s,t) = Cov(y_t, y_s)$ is the autocovariance function of the time series $\{y_t\}$. The ACF measures the linear dependence between y_t and past or future values of the time series (Prado and West (2010)). For all original pieces considered, the ACF plots showed structure out to a lag of 40. The melodic progression and global structure of the original piece can be seen in the ACF plot, as the plot demonstrates how subsequent pitch values in the original piece depend on each other. A generated piece that had a high degree of global structure and a melodic progression over the course of the piece was expected to have an ACF plot with some structure out to high lag as well. The ACF was calculated for each generated piece out to lag 40.

Partial Autocorrelation Function

The partial autocorrelation function (PACF) measures the correlation between observations y_t and y_{t+h} in the time series $\{y_t\}$, where the linear dependence on the intervening values $y_{t+1} : y_{t+h-1}$ is removed (Shumway and Stoffer (2011), Prado and West (2010)). Again, the PACF plots for the original pieces showed structure out at least a few lags, and we hoped to see some degree of similar structure in the generated pieces, indicating some degree of structure over time in the piece. The PACF was again calculated for each generated piece out to lag 40.

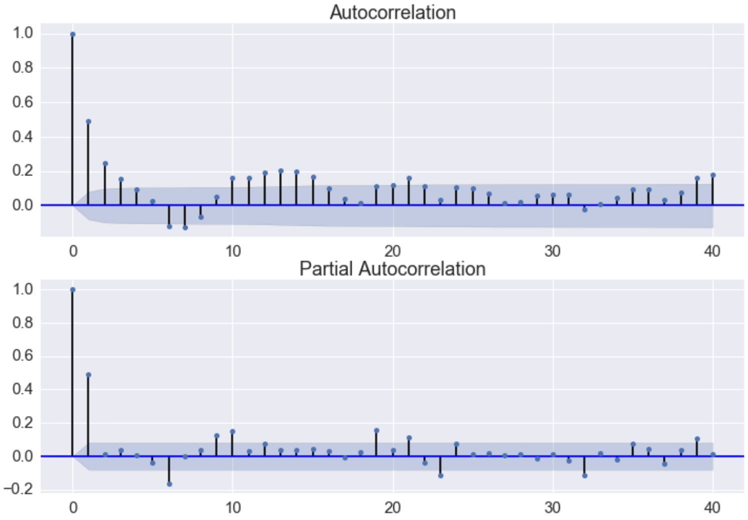

FIGURE 2.3: ACF and PACF plots for the original training piece, Chopin's Piano Sonata No. 2, 3rd Movement (Marche funebre). In the ACF plot there is clear structure out to high lags and there is structure out to perhaps lag 10 in the PACF plot.

2.6 Procedure

2.6.1 Data

The original piano pieces were downloaded from mfiles.co.uk (2017), Krueger (2016) and MIDIworld (2009) in Musical Instrument Digital Interface (MIDI) format. MIDI is a data communications protocol that allows for the exchange of information between software and musical equipment and symbolically represents the data needed to generate the musical sounds encoded in the data (Rothstein (1992)). We then used open source code from Walker (2008) to convert the original files from MIDI format to CSV. The converted CSV files contained several pieces of header information, including the key signature, time signature, tempo and the number of MIDI clock ticks per metronome click (i.e. 24 MIDI clicks for 1 quarter note) (Walker (2008)). Each row of the data itself contained the note pitch, note velocity and MIDI time click for each note in the piece. The note pitch was coded as an integer between 0 and 127, where 60 represented middle C and each integer increase indicated an increase of a half step in the pitch and each integer decrease indicated a lowering of a half-step in pitch. The note velocity was also encoded as an integer between 0 and 127, where 0 meant that the note was off and 127 was the loudest possible volume. The first occurrence of a note pitch with non-zero velocity indicated the start of that pitch, which was sustained until the note pitch occurred again with zero velocity, indicating that that note pitch had ended.

The time clicks at which note pitch observations occurred were largely regular in time, with the amount of time elapsing between observations almost always the same. However, we assumed that all observations were equally spaced in time for modeling purposes. Additionally, some time clicks had multiple note pitches on at the same time, indicating musical chords. However, we assumed that the note pitches were a univariate time series and treated each observation as sequential in time, whether or not subsequent observations occurred at the same time click.

We did not alter the time signature, key signature, tempo or the number of MIDI clock ticks per quarter note from the original training piece to the generated piece. We assumed that only the note pitches that were in the original piece could be generated by the model. We did not alter the times at which observations occurred and all generated pieces had the same number of note observations as the original training piece.

2.6.2 Model Training and Evaluation Procedure

For each piece and model considered, only the note pitches were used as the observed states. Each model was run on all ten of the original training pieces considered and the appropriate Baum-Welch algorithm was run until convergence, with all models implemented in Python and Cython. The first and second order HMMs, the first and second order left-right HMMS, the ARHMM, the NSHMM and the layered HMM were run with 25 hidden states ($m = 25$). The factorial HMM was the average of three first order HMMs with 15, 10 and 5 hidden states for each HMM. The third order and third order left-right models were run with 10 hidden states. The two hidden state HMM was run with 10 hidden states for the first layer and 5 hidden states for the second layer, as well as the opposite configuration, 5 hidden states for the first layer and 10 hidden states for the second hidden layer. The number of hidden states is a parameter that needs to be chosen before model training. Rabiner (1989) discusses the selection of the number of hidden states and suggests choosing the number of hidden states to be similar to the number of observed states, which is what we considered here. For the more computationally intensive models, such as the third order and two hidden state models, we considered fewer hidden states for computational efficiency. We also considered a model where the parameters for a first order HMM were randomly generated as a comparison point for the pieces generated by the models trained on original pieces.

After the considered model converged, 1000 new pieces were sampled from the learned model using the appropriate generative description of each model. All metrics were calculated for the original training piece and the 1000 generated pieces. Five generated pieces were selected at random to save and to further evaluate through listening. Since only the note pitches were modeled for each considered model, the generated note pitch controlled the harmonic, melodic and rhythmic structure of the generated piece. After generating the new notes, for each pitch, every other note pitch was set to have zero velocity (which controlled the duration of the note pitch as in the original training piece) and splines fit to the velocity of the original piece were used to fit the non-zero velocities of the generated piece. Since we were most interested in modeling the note pitches as opposed to the note velocities, we assumed the note velocities would be unchanged between the original and generated piece. The saved generated pieces were saved as a CSV in the same format as the original training piece (with only the generated note pitches and new velocities changed from the original piece to the generated piece). The open source code from Walker (2008) was again used to convert the CSV files back to MIDI format so that the generated pieces could be listened to.

To quantitatively evaluate the generated pieces, the root mean squared error (RMSE) for each metric (except mutual information and edit distance) was calculated for the 1000 generated pieces compared to the original piece. The RMSE was considered as a summary of the metrics, as we wanted each generated piece to be similar to the original training piece in terms of each metric considered. That is, if the original training piece contained a lot of dissonance, we expected the generated piece to contain a fair amount of dissonance as well, but if the original training piece had no dissonance, we did not want to see any, or at least very little, dissonance in the generated piece. The RMSE was primarily used to rank the pieces and to select the top generated pieces for evaluation by human listeners and to gain insight into some general trends observed in the generated pieces. The RMSE can be calculated as

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y_0)^2} \tag{2.6}$$

where y_i is the considered metric, y_0 is the value of the metric for the original piece and n is the number of generated pieces, in this case $n = 1000$.

3

Results

3.1 Model Comparison

From a comparison of the RMSE values for the different metrics by model and by listening to each of the pieces generated by each model, some general model trends emerged. Overall, the layered HMM had one of the lowest RMSE values for all of the metrics considered. The pieces generated by the layered HMM tended to have minimal dissonance and while they did not have a significant amount of long term structure or melodic progression, these generated pieces did have the most structure of the pieces generated by the different models. The layered HMM also produced generated pieces that had the lowest edit distance and highest mutual information as compared to the original piece. Pieces generated by the layered HMMs were the most similar to the original piece, which led to more pleasing musical results, but also to the generation of pieces that might be considered too similar to the original piece.

The most simple model, the first order HMM, also tended to perform well in terms of the musicality measures. The pieces generated by the first order HMM did not have too much dissonance, too many large melodic or harmonic intervals and had note pitch distributions similar to that of the original piece. The first order HMM did not have as much global structure as the layered HMM, however. The two hidden state HMM with $m_1 = 5$ and $m_2 = 10$ or with $m_1 = 10$ and $m_2 = 5$ tended to repeat the same note for a long time, and thus sounded repetitive and had note pitch distributions that were quite different from that of the original piece. The pieces generated by the "random HMM" (that is, all model parameters were randomly generated without regard to the training piece) tended to produce very dissonant pieces that obviously lacked any kind of long-term structure and especially suffered from several large harmonic and melodic intervals. These models thus tended to perform poorly on all metrics, especially the large interval metric. The NSHMM also suffered in particular from a lack of global structure and had relatively high RMSE values for the temporal metrics as compared to the other models considered. The factorial HMM tended to produce quite dissonant pieces, though the pieces did have some progression, relative to the other models considered.

The relative performance of the other models varied according to the training piece. The RMSE was used to gauge relative performance between models and pieces and to select the "top" pieces to be evaluated by human listeners. The RMSE values were thus not necessarily meaningful, but were instead a way to quantitatively compare relative performance for different metrics, models and pieces.

As an illustration of some of these general model trends, we considered the example of Mussorgsky's Pictures at an Exhibition, Promenade - Gnomus. Since the top pieces selected in terms of RMSE for the listening evaluation described below were simple tunes with straightforward melodies, we selected a more complex piece (with the highest entropy of the original pieces) to use for the model comparison evaluation, Mussorgsky's Pictures at an Exhibition, Promenade - Gnomus. The RMSE for each metric was calculated for each model trained only on this piece, and the model results were ranked from lowest to highest RMSE to give a sense of how well each model performed on each metric. The models are listed from lowest to highest RMSE in Table 3.4, Table 3.5 and Table 3.6. Additionally, the average entropy of the generated pieces (Table 3.1), edit distance (Table 3.2) and mutual information (Table 3.3) are provided. The layered HMM had the lowest average edit distance (normalized by the length of the training piece) and the highest average mutual information with the original piece of the models considered, which explains why the layered HMM had low relative RMSE values for the considered metrics below.

Table 3.1: Average entropy for pieces generated by different models trained on Pictures at an Exhibition, Promenade - Gnomus.

Model	Average Entropy
Random	2.872
NSHMM	2.760
Second Order	2.619
Third Order	2.600
Third Order Left-Right	2.510
Layered	2.510
Second Order Left-Right	2.506
First Order Left-Right	2.506
ARHMM	2.505
HSMM	2.505
First Order	2.504
Factorial	2.454
Two Hidden States $m_1 = 10, m_2 = 5$	1.962
Two Hidden States $m_1 = 5, m_2 = 10$	1.614

Figure 3.1 shows the first few lines of the original training piece by Mussorgsky. There is no dissonance in the opening to this piece, there is a clear melodic progression with the repetition of the motif in the first two bars and there are also no harmonic or melodic intervals of more than an octave. The repetition of a melodic motif, in particular, was a clear trait of the original training piece that was not seen in any of the generated pieces.

Table 3.2: Average edit distance for pieces generated by different models trained on Pictures at an Exhibition, Promenade - Gnomus as compared to the original piece.

Model	Average Edit Distance
Random	0.860
Two Hidden States $m_1 = 10, m_2 = 5$	0.835
Two Hidden States $m_1 = 5, m_2 = 10$	0.814
NSHMM	0.805
Third Order	0.796
Second Order	0.794
ARHMM	0.766
HSMM	0.764
Third Order Left-Right	0.763
Second Order Left-Right	0.763
First Order Left-Right	0.753
Factorial	0.742
First Order	0.700
Layered	0.535

Table 3.3: Average mutual information for pieces generated by different models trained on Pictures at an Exhibition, Promenade - Gnomus as compared to the original piece.

Model	Average Mutual Information
Two Hidden States $m_1 = 5, m_2 = 10$	0.077
Two Hidden States $m_1 = 10, m_2 = 5$	0.116
ARHMM	0.219
Second Order Left-Right	0.219
HSMM	0.220
Third Order Left-Right	0.221
First Order Left-Right	0.221
First Order	0.223
Second Order	0.225
Third Order	0.225
NSHMM	0.241
Random	0.248
Factorial	0.717
Layered	1.085

Figure 3.2 shows the opening bars of the piece generated using the layered HMM and trained on Pictures at an Exhibition, Promenade - Gnomus. There is some

Table 3.4: RMSE values for the dissonance metric for pieces generated by different models trained on Pictures at an Exhibition, Promenade - Gnomus.

Model	Dissonance RMSE
Layered	0.003
First Order	0.004
Two Hidden States $m_1 = 10, m_2 = 5$	0.004
Two Hidden States $m_1 = 5, m_2 = 10$	0.004
Random	0.006
NSHMM	0.006
ARHMM	0.006
Third Order Left-Right	0.006
Third Order	0.006
First Order Left-Right	0.006
HSMM	0.006
Second Order Left-Right	0.006
Second Order	0.007
Factorial	0.009

Table 3.5: RMSE values for the large interval metric and the note count metric for pieces generated by different models trained on Pictures at an Exhibition, Promenade - Gnomus.

Large Interval RMSE		Note Count RMSE	
Layered	0.017	Layered	0.006
First Order	0.019	First Order Left-Right	0.007
Second Order	0.021	HSMM	0.008
HSMM	0.023	ARHMM	0.008
ARHMM	0.024	Second Order Left-Right	0.008
First Order Left-Right	0.027	Third Order Left-Right	0.008
Second Order Left-Right	0.028	First Order	0.008
Third Order Left-Right	0.029	Second Order	0.012
Third Order	0.039	Third Order	0.012
Two Hidden States $m_1 = 10, m_2 = 5$	0.062	NSHMM	0.012
NSHMM	0.095	Random	0.019
Two Hidden States $m_1 = 5, m_2 = 10$	0.113	Factorial	0.022
Factorial	0.157	Two Hidden States $m_1 = 10, m_2 = 5$	0.023
Random	0.167	Two Hidden States $m_1 = 5, m_2 = 10$	0.026

Table 3.6: RMSE values for the ACF and PACF for pieces generated by different models trained on Pictures at an Exhibition, Promenade - Gnomus.

ACF RMSE		PACF RMSE	
Two Hidden States $m_1 = 10, m_2 = 5$	0.068	Two Hidden States $m_1 = 10, m_2 = 5$	0.0589
Factorial	0.069	Two Hidden States $m_1 = 5, m_2 = 10$	0.060
Layered	0.069	Factorial	0.060
First Order	0.070	Layered	0.061
Two Hidden States $m_1 = 5, m_2 = 10$	0.070	First Order Left-Right	0.061
First Order Left-Right	0.071	First Order	0.061
Random	0.072	Random	0.062
Third Order	0.072	Third Order	0.062
Second Order Left-Right	0.072	Second Order	0.062
Second Order	0.072	ARHMM	0.062
HSMM	0.072	HSMM	0.062
Third Order Left-Right	0.072	Third Order Left-Right	0.062
NSHMM	0.072	NSHMM	0.062
ARHMM	0.072	Second Order Left-Right	0.062

dissonance in this excerpt from the generated piece, especially beginning in the last two beats of the second bar with the C followed by the Db (a dissonant minor second interval), for example. However, there is no dissonance in the first bar, and while this generated piece is certainly more dissonant than the original training piece, it is the least dissonant of the generated pieces. There is not any clear melodic progression or

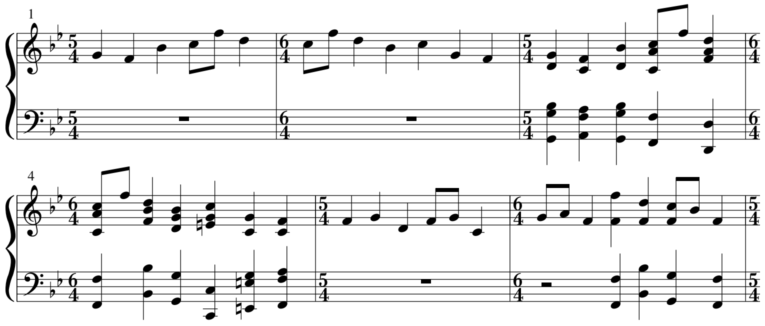

FIGURE 3.1: The first few bars from the original training piece, Pictures at an Exhibition, Promenade - Gnomus by Mussorgsky. There is no dissonance in the opening to the original piece and there is a clear melodic progression.

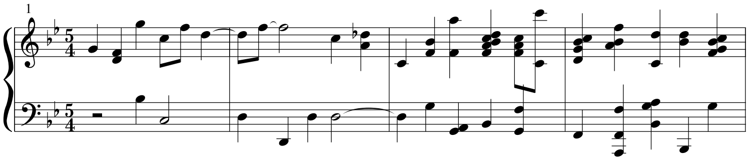

FIGURE 3.2: The first few bars from a piece generated by a layered HMM trained on Pictures at an Exhibition, Promenade - Gnomus by Mussorgsky.

repetition of motifs as there is in the original piece, however, the rhythmic structure of the first two bars is similar to that of the second two bars. There is a melodic interval larger than an octave in the first bar, from the F to the G, and there are more large intervals present than in the original piece. The fourth and fifth beats in the first measure are identical in terms of pitch and rhythm between the original piece and the treble line of the generated piece, and the generated piece does sound similar to the original piece in this opening. This may be indicative of overfitting, however, the piece generated by the layered HMM is mostly consonant and the structure of the piece progresses somewhat over these first few bars.

The first four measures of the piece generated by the first order HMM are shown in Figure 3.3. This piece generated by the first order HMM was similar to the layered HMM, though slightly more dissonant, especially starting in the last two beats of the third measure, where there are multiple dissonant intervals and chords occurring, as well as several accidentals, which we do not observe in the first bars of the original training piece. There are also some large intervals of more than an octave in the first measures of this generated piece, more than occurred for the piece generated by the layered HMM or in the original piece. There is some slight melodic progression, for example, the descending treble line in the first bar, but any structure is minimal and there are no clear motifs in this generated piece.

In the case of the piece generated by the factorial HMM, there is dissonance occurring as soon as the second beat of the first measure and there is a high degree

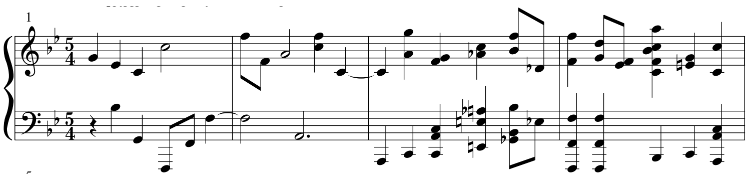

FIGURE 3.3: The first few bars from a piece generated by a first order HMM trained on Pictures at an Exhibition, Promenade - Gnomus by Mussorgsky.

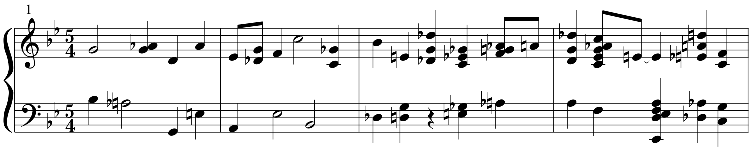

FIGURE 3.4: The first few bars from a piece generated by a factorial HMM trained on Pictures at an Exhibition, Promenade - Gnomus by Mussorgsky.

of dissonance throughout the rest of the first few measures of the piece (Figure 3.4). Again, there is little in the way of melodic progression, partially due to the amount of dissonance present in these first measures. The generated piece does not bear very much resemblance to the original piece by Mussorgsky, apart from starting on the same note (G) in the treble line as in the original piece and some rhythmic similarities, for example, the ascending eight notes in the first beat of the second bar.

The first measures from the piece generated by the NSHMM bear little resemblance in terms of rhythm to the original training piece (Figure 3.5). In general, there are longer notes in this piece generated by the NSHMM than are seen in the training piece (for example, the dotted quarter note in the first measure) and this piece is dissonant from the first note. There is no evidence of melodic structure or even of a shorter motif, and this was reflected in the RMSE for the temporal structure metrics, as the NSHMM ranked towards the bottom.

There are only three distinct note pitches in the first two measures of the piece generated by the two hidden state HMM with $m_1 = 5$ and $m_2 = 10$ (Figure 3.6),

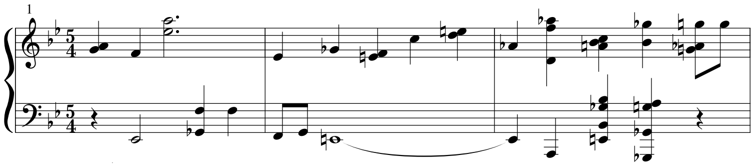

FIGURE 3.5: The first few bars from a piece generated by a NSHMM trained on Pictures at an Exhibition, Promenade - Gnomus by Mussorgsky.

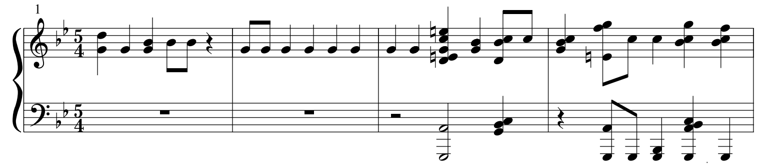

FIGURE 3.6: The first few bars from a piece generated by a two hidden state HMM (with $m_1 = 5$ and $m_2 = 10$) trained on Pictures at an Exhibition, Promenade - Gnomus by Mussorgsky.

which was reflected in the metrics as a relatively high RMSE for the note counts for this model. The second measure in the generated piece only consists of the note G, while none of the notes in the second measure of the original training piece are the same pitch. There are additionally some large intervals in the bass line in the fourth measure. There is again very little, if any melodic progression, in large part due to the repetition of the same few notes throughout the first few bars.

Finally, the piece generated with random first order HMM parameters is highly dissonant and has a large number of large intervals. Nearly every melodic and harmonic interval is dissonant (Figure 3.7). The large harmonic and melodic intervals are measured by the metrics as a relatively high RMSE for the large interval metric. Since this model has no relation to the training piece, as the parameters are randomly generated, it makes sense that there is a high degree of dissonance and large intervals for the generated piece. The rhythm of the generated piece matches closely to the rhythm of the original piece, though the large amount of dissonance practically hinders any sense of melodic progression.

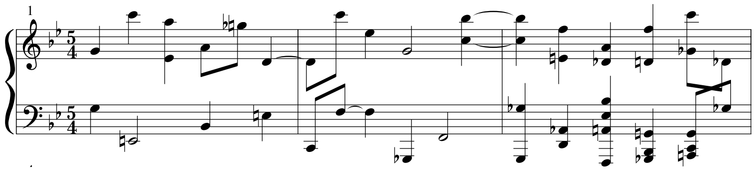

FIGURE 3.7: The first few bars from a piece generated by a first order HMM with random parameters trained on Pictures at an Exhibition, Promenade - Gnomus by Mussorgsky.

3.2 Evaluated Pieces

The measured metrics were used to select the top piece and model in terms of the RMSE for entropy, an average of the musicality measures and an average of the temporal structure measures (without repeating pieces). The generated piece with the lowest RMSE for the entropy was Mendelssohn's Hark! The Herald Angels Sing modeled by a layered HMM. The generated piece with the lowest RMSE for the average of the musicality metrics was Beethoven's Ode to Joy modeled by a first order HMM, and the generated piece with the lowest RMSE for the temporal structure measures was Chopin's Piano Sonata No. 2, 3rd Movement (Marche funebre) modeled by a layered HMM. These three pieces were sent to sixteen individuals to listen to, evaluate and rank. The individuals were not randomly selected to evaluate the generated pieces. Eight individuals were considered "more" musically knowledgable (based on the fact that they were currently in a musical ensemble), while the other eight individuals were not currently in a musical ensemble. The RMSE values for the entropy, musicality metric average and temporal structure (an average of all ACF and PACF RMSE values) can be found in Table 3.7. The piece generated by training a layered HMM on Chopin's Marche funebre had a lower RMSE for entropy than the piece generated by a layered HMM trained on Hark! The Herald Angels Sing, but had already been selected as the piece with the lowest temporal structure RMSE. The piece generated by a layered HMM trained on Hark! The Herald Angels

Sing had the second lowest entropy RMSE and was thus selected for the listening evaluation.

Table 3.7: Summary of the entropy, musicality average and temporal structure average RMSE values for the three pieces selected for the listening evaluation.

Training Piece	Model	Entropy RMSE	Average Musicality RMSE	Average Temporal Structure RMSE
Ode to Joy	First Order HMM	0.042	0.018	0.142
Marche funebre	Layered HMM	0.021	0.019	0.049
Hark! The Herald Angels Sing	Layered HMM	0.022	0.027	0.075

For the evaluation and ranking of these three generated pieces, each individual was told that all of the pieces had been generated by a computer using statistical models and that each model was trained on an original piano piece from the Romantic era, then the learned parameters were used to generate a new piece of music. The pieces were labeled A, B and C, so the listeners did not know a priori the original training piece for each generated piece. In addition to ranking the three pieces in order of their favorite to least favorite, each individual was asked: "What did you like and not like about each piece? Did any of the pieces sound like they were composed by a human? If so, why and if not, why not?" and for any other general comments about the pieces that they had.

3.2.1 Rankings of the Evaluated Pieces

The rankings for each of the three generated pieces for each individual are shown in Figure 3.8 and Figure 3.9, where listeners were asked to rank the pieces from favorite (1) to least favorite (3). All but one of the listeners that was in a musical ensemble ranked the three pieces exactly the same way, with the piece generated from a layered HMM trained on Chopin's Marche funebre considered the favorite and the piece generated by a layered HMM trained on Mendelssohn's Hark! The Herald Angels

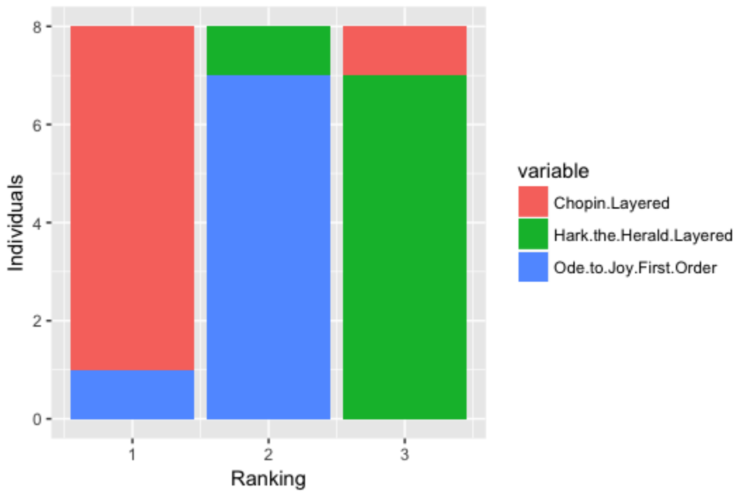

Figure 3.8: The rankings of generated pieces from a layered HMM trained on Chopin's Marche funebre and Mendelssohn's Hark! The Herald Angels Sing and the piece generated from a first order HMM trained on Beethoven's Ode to Joy as evaluated by eight listeners who were currently in a musical ensemble.

Sing considered the least favorite. There was much more parity among the three evaluated pieces for the rankings by listeners not currently in a musical ensemble as each of the three pieces was chosen as the favorite by at least one listener. Again, however, five of the eight listeners liked the piece generated by training a layered HMM on Chopin's Marche funebre the best. The piece generated by training a layered HMM on Hark! The Herald Angels Sing was ranked more favorably by this group of listeners that were not in a musical ensemble; four listeners ranked this piece as their second favorite and it had the same number of least favorite rankings as the piece generated by training a first order HMM on Beethoven's Ode to Joy.

In terms of specific comments by the listeners for each piece, several listeners that were currently in musical ensembles commented on the "dark" tone and minor key of the piece generated by training a layered HMM on Chopin's Marche funebre. There were also comments from this group about liking the melody on top of the bass line for this piece, which several listeners mentioned added complexity to the

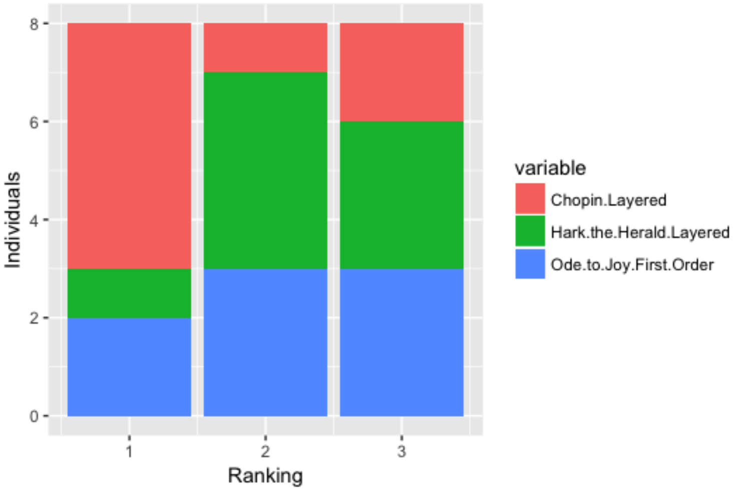

FIGURE 3.9: The rankings of generated pieces from a layered HMM trained on Chopin's Marche funebre and Mendelssohn's Hark! The Herald Angels Sing and the piece generated from a first order HMM trained on Beethoven's Ode to Joy as evaluated by eight listeners who were not currently in a musical ensemble.

piece, although some listeners thought that the piece was simplistic. One listener liked the large range of octaves that occurred in this generated piece and thought that the movement between octaves made the piece sound more sophisticated and like it could have been composed by a human. A few listeners mentioned how this generated piece could be used as the basis for a larger work, either through further development or as a bass line in another piece. One listener enjoyed the ending of the piece and felt that there was "somewhat of a melody" that built towards the end of the piece. Another listener commented on how the dissonance occurring in this piece was resolved more so than in the other two pieces because of the slower tempo and simple rhythm of this piece. One listener complained of missed notes in the recording that sounded like "clumsy hands", while another listener thought that the piece could have been performed by an experienced pianist. One listener mentioned specifically that this sounded like a "funeral" piece and another mentioned that it sounded like a Chopin nocturne.

The listeners that were not in musical ensembles seemed to be split about their opinion on the complexity of this piece generated by training a layered HMM on Chopin's Marche funebre. About half of the listeners commented that the piece seemed to be complex, to have "dimension" and to be melodic with a few different themes, making it sound more likely to be composed by a human, while the other half thought that the piece was repetitive and "rote", and therefore less likely to have been composed by a human. One listener in this group again mentioned that it sounded like there were missed or wrong notes played by the "pianist". Several listeners again commented on the "dark tone" of the piece. Two listeners did not like the ending of the piece, which they felt was abrupt. Two listeners also commented on the lack of dissonance in this generated piece. One listener again commented on enjoying the range of notes in the piece. In this group of listeners, one person commented on how this piece sounded modern, while another thought that it sounded like Chopin.

For the piece generated by a layered HMM trained on Mendelssohn's Hark! The Herald Angels Sing, five of the listeners that were currently in musical ensembles commented on how they thought that this piece sounded like Hark! The Herald Angels Sing, with more missed notes and dissonance. One listener especially did not like this piece because of their expectation for consonance from the original Hark! The Herald Angels Sing, making the dissonance in this piece more jarring. One listener thought that this piece was more complicated than the Chopin trained piece because they liked that this piece had a melody and harmony. Several listeners noted that they thought there were "out of place" chords and dissonances that were not resolved, or too many half steps, in the piece for it to have been composed by a human. One listener thought that the piece was too simple and predictable, since the notes were mostly moving up and down the major scale and another listener thought that the piece was "choppy". A few listener's commented on the "happy" tone of this piece and the fact that there again sounded like there were missed or clumsy notes played by the performer.

Only one individual in the group of listeners that were not currently in a musical ensemble thought that the piece generated by a layered HMM trained on Mendelssohn's Hark! The Herald Angels Sing sounded like Hark! The Herald Angels Sing. The group was approximately split on whether they thought this piece could have been composed by a human or not. One individual thought that it could have been composed by a human because they thought that the ending sounded like that of a "typical" piano piece and another thought the note progressions sounded like a human composed them. However, three different listeners thought that the piece did not sound like it was composed by a human because it was too dissonant in an "unintended" way, did not "flow well together" and the dissonance seemed to occur randomly (which they commented could have been the result of a more modern composer). Several listeners, even those that enjoyed the piece, commented on the dissonance, which the listeners overall described as "unintentional" or the result of playing "mistakes". Finally, two listeners commented that they liked this piece because it sounded "triumphant" or "upbeat".

The group of listeners that was currently in a musical ensemble were divided on their opinion of the melodic characteristics of the final evaluated piece, the piece generated by a first order HMM trained on Beethoven's Ode to Joy. Three listeners thought that the melody "didn't go anywhere" and that the piece had melodies that were not "interesting" or were too repetitive. One listener complained that the lower voice in the piece was "boring" because it only played the same two pitches and another that the repetition of the piece meant it could be used as interlude music and had possibility. Another listener also commented that this piece could be developed into a larger work. Another set of listeners thought that this piece had better phrase and melodic progression than the other pieces and that there was a "good structure" to the piece. One listener particularly enjoyed the syncopated rhythm, the lower chords that they thought added direction to the melody and the "clever" ending, which they believed could have been composed by a human. Several listeners commented on the "atonal" nature or out-of-place dissonance in the piece, with one listener commenting that the extended chords and dissonance made the piece sound like it was a modern composition.

The group of listeners not currently in a musical ensemble echoed some of the sentiments expressed by the group in a musical ensemble for the piece generated by a first order HMM trained on Beethoven's Ode to Joy. A few listeners commented on the sometimes "jarring" dissonance, but most seemed to think that there were fewer, "milder errors" in dissonance as compared to some of the previous pieces. Two listeners commented on enjoying the faster tempo of the piece and the note changes. Another listener enjoyed the depth and texture of the piece and thought that the piece seemed like it had been composed by a human and another thought it sounded like "new age piano music" that they had previously heard and enjoyed. Two listeners commented on the repetitive nature of the piece, with one adding that it seemed like the piece "forgot" what had previously occurred in the piece and repeated the previous part of the piece over and over again. One listener thought that the "jarring" dissonance precluded the piece from sounding like it had been composed by a human, but thought that it sounded like Christmas music.

For overall comments on all of the pieces, several members in the group of listeners that were currently in a musical ensemble thought that the pieces sounded like they could have been composed by a human, but that the composition style was typical of a "twentieth century", "modern", "contemporary" or "post-classical" composer and commented that the pieces did not sound like they had been composed in the Romantic era. In particular, the "atonal chords and sustained base chords" and the "harmonic variation and non-diatonic chords" were provided as justifications for the more modern sound of the composed pieces, though one listener noted that this could have been due to the fact that modern composers “are less obligated to obey the rules of tonal harmony”. Several of the listeners in this group commented on how they wished there was more phrasing or structure in the generated pieces, as there was not much “complexity” in the pieces. Two listeners also thought that a human performer could improve the interpretation of the generated pieces by adding some phrasing to the music itself.

One of the listeners in the group not currently in a musical ensemble commented that overall each piece sounded “distinct” and suggested that this “experimental” method of composition might be applied to “free jazz”. Another listener thought that each piece sounded “related” to the piece it was trained on, though they noted that they noticed this especially for the piece generated by a layered HMM trained on Mendelssohn’s Hark! The Herald Angels Sing, a piece that they were familiar with.

3.2.2 Discussion

All three of these pieces were in the top four for lowest entropy of the ten original training pieces considered, and each piece was among the simpler of the pieces considered, in terms of melody and harmony. The original training pieces with the highest entropy, such as Liszt’s Hungarian Rhapsody, No. 2, were much more difficult for the HMMs to model and ranked towards the bottom for the RMSE of the majority of considered metrics. Of the evaluated pieces, the piece generated by training a layered HMM on Chopin’s Marche funebre was in general the most well received among the listeners and had the fewest comments complaining of un-resolved or out of place dissonance. Chopin’s Marche funebre is built on chords that are fifths, which is an open, perfect interval, widely used in the music of non-Western cultures. Thus, even when there was a passing dissonance in the piece generated by training an HMM on Chopin’s Marche funebre, the dissonance could be resolved by relaxation to a pure interval, making the piece sound less dissonant to the listener, as the majority of the dissonance in the generated piece could be and was resolved. This simplicity of the harmony in the original piece by Chopin contributed to the relative success in the generation of new pieces from HMMs trained on Chopin's Marche funebre.

Mendelssohn's Hark! The Herald Angels Sing and Beethoven's Ode to Joy, however, are built on major chords comprised of thirds and thus had more "potential" for dissonance in the generated piece, as any interval that was not in perfect unison, or a third, fifth or octave in the chord or as an interval from the previous note sounded dissonant. For these pieces built on thirds, the dissonance could not be resolved as easily as in the case of Chopin's Marche funebre built on open fifths, thus the generated pieces sounded much more dissonant. In the examination of the pieces generated by HMMs trained on Mussorgsky's Pictures at an Exhibition, Promenade - Gnomus, the generated pieces were also quite dissonant, especially compared to the pieces generated by HMMs trained on Chopin's Marche funebre, and again, Mussorgsky's original piece is built on major chords where the third of the chord is present.

To further explore the idea that training HMMs on original pieces with simple harmonic structure built on open intervals tend to produce generated pieces that have less dissonance and thus are slightly more successful at sounding as if they had been composed by a human, two additional training pieces from different eras were considered; Johann Pachelbel's Canon in D, a piece from the Baroque period that is built on fourths, and the theme from Jupiter from Gustav Holst's The Planets (arranged for piano), a piece from the early twentieth century with chords that are primarily built on the fifth. Both of these original training pieces have relatively simple melodies and primarily consist of open chords (either perfect fourths or fifths). The generated pieces, while not evaluated by a group of listeners, had very little dissonance and most of the dissonance that did occur in the generated pieces was resolved. These two generated pieces were among our personal favorites of the generated pieces considered in this thesis. We conclude that original training pieces with a simple harmonic structure, one that is built on perfect, open intervals, lead to generated pieces from an HMM with less dissonance than training pieces that are primarily composed of major and minor chords that have a third.

However, all of the generated pieces, even those that had little dissonance, lacked an overall melodic progression or global structure. Several listeners commented on this fact repeatedly, and it was also evident in the metrics. For example, plots of the ACF and PACF for the piece generated by training a first order HMM on Beethoven's Ode to Joy (Figure 3.11) show significantly less structure, especially over long lags, than the ACF and PACF plots for the original piece (Figure 3.10). This indicates that there is little dependence between notes over time at various lags in the generated pieces, especially as compared to the dependence between notes over various lags in the original pieces. Additionally, as seen by the example of the models trained on Mussorgsky's Pictures at an Exhibition, Promenade - Gnomus, even relatively short, simple motifs in an original training piece that are highly repetitive were not modeled well by any of the HMMs considered. This lack of global structure is to be expected, as the models considered made quite restrictive assumptions about long-term dependence between hidden states.

In terms of style of the generated pieces, none of the evaluated pieces sounded like they were composed during the Romantic era. Most listeners seemed to think that the pieces could have been composed by a human, albeit one in the modern or contemporary period (especially due to the prevalence of large intervals more than an octave that are much more common in modern music than in Romantic era music). The HMMs considered did not model the traits of Romantic music well and this was reflected in the generated pieces. Additionally, the layered HMM in particular, may have suffered from "overfitting", as several listeners were able to identify that one of the evaluated pieces was indeed trained on Hark! The Herald Angels Sing.

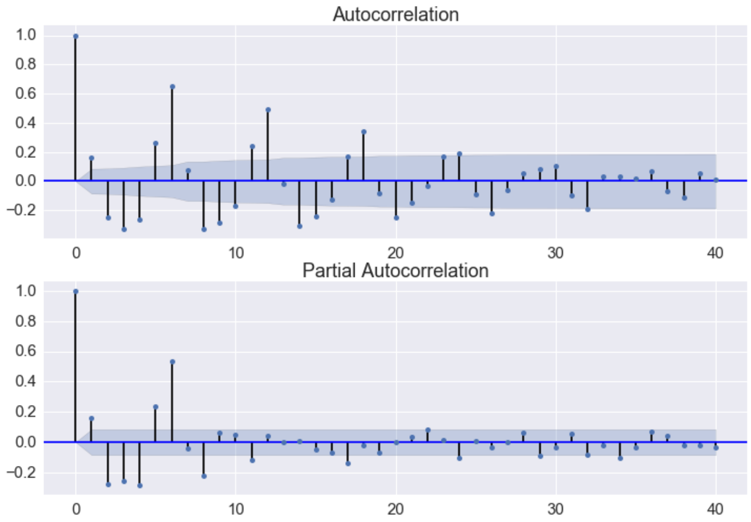

Figure 3.10: Plots of the ACF and PACF for Beethoven's Ode to Joy.

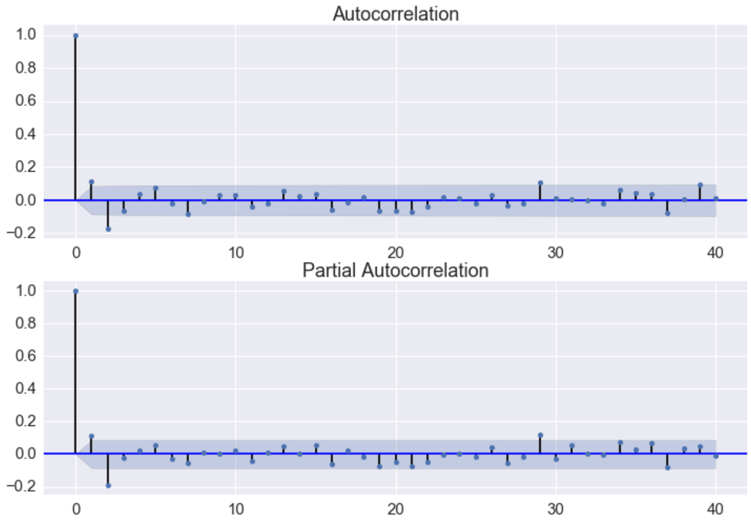

Figure 3.11: Plots of the ACF and PACF for the piece generated by a first order HMM trained on Beethoven's Ode to Joy.

However, we do expect the generated pieces to bear similarity to the training piece and this potential "overfitting" is not necessarily an issue. Many listeners could not identify the training piece, only overall themes, and none of the models reproduced the training piece exactly.

All sheet music for the five generated pieces discussed above can be found in Appendix C. Additionally, MP3s of the generated pieces and code for the implementation of the models above can be found at `https://github.com/aky4wn/Classical-Music-Composition-Using-Hidden-Markov-Models`.

4

Conclusions and Future Work

4.1 Conclusions

Overall, we find that HMMs trained on pieces from the Romantic era do have some success in generating piano music that sounds like it was composed by a human. The models considered tended to be more successful at modeling harmony than melodic progression, though unexpected and unresolved dissonances did occur in the generated pieces and these unresolved dissonances were very unlikely to occur in a Romantic era piano piece composed by a human. The HMMs were more successful at generating consonant harmonies when trained on pieces with simple harmonies, particularly original pieces that were built on perfect intervals. Models that incorporated some type of hierarchical structure, particularly the layered HMM, tended to be more successful at generating pieces with less dissonance and slightly more melodic progression than other models considered. However, layered HMMs also had the potential for overfitting, leading to pieces that could sound too much to the listener like the original training piece, just with more dissonance. Furthermore, the generated pieces sounded more like pieces composed in the Modern era than like pieces composed in the Romantic era. The generated pieces suffered most notably from a lack of global structure or long-term melodic progression. The overall progression of a musical piece and its ability to evoke different emotions in the listener as a result is a key component of Romantic era music. Generating musical pieces that have a global structure will be a key challenge to overcome in order for statistical models such as HMMs to be able to compose music at the same level as humans.

4.2 Future Work

Based on the shortcomings of the considered HMMs, several directions for future work are suggested that attempt to resolve some of the problems with the generated pieces of the considered models. These future directions include considering hierarchical models, natural language models and RNNs.

4.2.1 Hierarchical Models

In original piano pieces from the Romantic era, there are several layers to the music that evolve over different time periods, such as short reoccurring motifs, longer melodies and the global form of the piece, for example. Of the considered models, those that had a hierarchical component tended to perform better in terms of the RMSE and listening evaluations and came closer to modeling the actual hidden structure of an original piece. Thus, models with a more hierarchical structure than the majority of HMMs considered here, such as the hierarchical HMM (Fine et al. (1998)), dynamic Bayesian networks with a hierarchical structure (Ghahramani (1997)), factorial HMMs where there is dependence between the underlying processes (Ghahramani and Jordan (1997)) or hierarchical RNNs (Hihi and Bengio (1996)), would likely improve the ability of modeling both short-term and global structure in musical pieces. In particular, explicitly modeling the hierarchy of rhythm, melody and harmony in the original training piece could improve the ability of hierarchical models to generate music that sounds like it was composed by a human.

4.2.2 Natural Language Models and Musical Grammar

Classical music composition follows basic rules and guidelines just like a spoken language. The composer and conductor Leonard Bernstein expressed great interest in developing a concrete musical grammar in his talks at Harvard (Bernstein (1976)). Inspired by these talks, Lerdahl and Jackendoff (1983) developed a generative theory of tonal music. Baroni et al. (1983) proposed a grammar of melody based on Bach chorales and Pearce and Wiggins (2006) explored the concept of expectation in melodic music using statistical models based on n-grams. Nierhaus (2009) surveyed additional work using natural language models to generate or model music.

The clearer development of a grammar for music with production rules and syntax would likely improve the harmonies and melodic aspects of the generated pieces, but would also have utility beyond music generation, such as allowing for the use of existing topic models for insight into common aspects in works by a particular composer, for example. The concept of a musical grammar would also contribute to a more concrete understanding and interpretation of the hidden states in an HMM and could allow for the ability to explicitly build musical theory into the HMMs. HMMs are in part so successful in speech recognition applications because knowledge about physical speech production and speech signals can be implemented in the models as constraints in the transition matrix. In general, the more prior knowledge about the series of interest that can be built into the HMM, the better the model is expected to perform and the concept of a musical grammar could thus introduce music theoretic constraints, leading to generated music that obeyed these grammatical rules.

Stochastic context-free grammars are an extension of HMMs and have learning algorithms similar to the Baum-Welch algorithm for HMMs (Lari and Young (1990)). Stochastic context-free grammars have probabilistic production rules and are hierarchical in structure, although they are computationally intensive. However, stochastic context-free grammars are likely a promising next step in exploring and understanding the underlying processes in original musical pieces.

4.2.3 Recurrent Neural Networks

One of the main criticisms of the music composed by the HMMs considered above was the lack of global structure in the generated pieces. By model design, HMMs have very limited memory and are thus incapable of modeling the longer term structure that occurs in original musical pieces. Recurrent neural networks (RNNs) have been used as an alternative to HMMs in music composition (Mozer (1994), Eck and Schmidhuber (2002), Boulanger-Lewandowski et al. (2012), Johnson (2015), Developers (2017), Magenta (2016), Hadjeres and Pachet (2016)). RNNs are neural networks that are specialized for processing sequential data (Goodfellow et al. (2016)).

However, Mozer (1994), Eck and Schmidhuber (2002) and Johnson (2015) note that RNNs suffer from a similar problem as HMMs in music composition, that is, a lack of global structure. RNNs by themselves are unable to capture long-term dependencies that occur in classical pieces of music, thus the music composed by RNNs can produce repetitive generated pieces (Johnson (2015)) or generated pieces that lack global structure (Mozer (1994), Eck and Schmidhuber (2002)). Eck and Schmidhuber (2002) note that this is likely due to the problem of vanishing gradients in RNNs and use Long Short Term Memory (LSTM) units to successfully capture longer-term structure in a corpus of blues music. Boulanger-Lewandowski et al. (2012) use recurrent temporal restricted Boltzmann machines and a generalization that they call the recurrent neural network restricted Boltzmann machine to model polyphonic music. They find that this model outperforms other models of polyphonic music, including HMMs, at learning harmonic and rhythmic probabilistic rules, where they are primarily interested in musical transcription of polyphonic music. Hadjeres and Pachet

(2016) likewise utilize LSTM units to generate harmonized Bach chorales. Exploring models, like the LSTM, that are capable of modeling longer-term dependencies in the original musical pieces would likely improve the global structure of the generated musical pieces and attempt to solve the problem of the lack of melodic progression in the pieces generated by HMMs.

Appendix A

HMM Algorithms

A.1 Forward-Backward Algorithm

The likelihood HMM problem can be solved using the Forward-Backward Algorithm. Let $\alpha_t(i) = P(X_{1:t}, Z_t = j|\lambda)$ for $j = 1, \ldots, m$ and $t = 1, \ldots, n$, where $\lambda = (\pi, A, B)$ is the considered HMM, m is the number of hidden states and n is the length of the observed sequence. Then, the Forward Algorithm is as follows:

1. Initialization: $\alpha_1(j) = \pi_j b_j(x_1), \quad 1 \leqslant j \leqslant m$
2. Recursion:
$$\alpha_t(j) = \sum_{i=1}^{m} \alpha_{t-1}(i) a_{ij} b_j(x_t), \quad 1 \leqslant j \leqslant m, \quad 1 < t \leqslant n$$
3. Termination:
$$p(x_{1:n}|\lambda) = \sum_{i=1}^{m} \alpha_n(i)$$

The Backward Algorithm can be analogously defined. Let $\beta_t(j) = P(X_{t+1:n}|Z_t = j, \lambda)$. Then, the Backward Algorithm is as follows:

1. Initialization: $\beta_n(i) = 1, \quad 1 \leqslant i \leqslant m$

2. Recursion:

$$\beta_t(i) = \sum_{j=1}^{m} a_{ij} b_j(x_{t+1}) \beta_{t+1}(j), \quad 1 \leqslant j \leqslant m, \quad 1 \leqslant t < n$$

(Jurafsky and Martin (2009), Rabiner (1989), Miller (2016a))

A.2 Viterbi Algorithm

The Viterbi algorithm can be used to solve the decoding problem. Define $\mu_1(z_1) = p(z_1)p(x_1|z_1)$. Then, similar to the Forward Algorithm, the Viterbi Algorithm is:

1. Initialization: $\mu_1(Z_1 = j) = p(Z_1 = j)p(x_1|Z_1 = j), \quad \alpha_1(j) = 0 \quad 1 \leqslant j \leqslant m$

2. Recursion:

$$\mu_t(j) = \max_{1 \leqslant i \leqslant m} \mu_{t-1}(i) a_{ij} b_j(x_t) \quad 1 \leqslant j \leqslant m, \quad 1 < t \leqslant n$$

$$\alpha_t(j) \in \operatorname*{argmax}_{1 \leqslant i \leqslant m} \mu_{t-1}(i) a_{ij} b_j(x_t) \quad 1 \leqslant j \leqslant m, \quad 1 < t \leqslant n$$

3. Termination:

$$M = \max_{1 \leqslant i \leqslant m} \mu_n(i)$$

$$z_n^* \in \operatorname*{argmax}_{1 \leqslant i \leqslant m} \mu_n(i)$$

Then, we backtrace to get the optimal sequence of hidden states, $z_{1:n}^*$ using the recursion:

$$z_{t-1}^* = \alpha_t(z_t^*), \quad t = n, n-1, \ldots, 2$$

(Jurafsky and Martin (2009), Miller (2016a))

A.3 Baum-Welch Algorithm

Finally, the Baum-Welch Algorithm (Baum (1972)) can be used to learn the parameters λ and to solve the third HMM problem. The Baum-Welch Algorithm is a special case of the Expectation Maximization (EM) algorithm (Dempster et al. (1977)). Define $\eta_t(i,j) = P(Z_t = i, Z_{t-1} = j | x_{1:n}, \lambda)$ and $\gamma_t(j) = P(Z_t = j | x_{1:n}, \lambda)$. Using the Forward and Backward variables,

$$\eta_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(x_{t+1}) \beta_{t+1}(j)}{p(x_{1:n}|\lambda)}, \quad 1 \leqslant t \leqslant n-1, \quad 1 \leqslant i,j \leqslant m$$

$$\gamma_t(j) = \frac{\alpha_t(j)\beta_t(j)}{p(x_{1:n}|\lambda)} \quad 1 \leqslant t \leqslant n, \quad 1 \leqslant j \leqslant m$$

Then, the Baum-Welch Algorithm can be written as:

1. Initialization: Randomly initialize $\lambda = (\pi, A, B)$
2. Iterate until convergence of $p(x_{1:n}|\lambda)$:

   (a) E-Step:

$$\eta_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(x_{t+1}) \beta_{t+1}(j)}{p(x_{1:n}|\lambda)}, \quad 1 \leqslant t \leqslant n-1, \quad 1 \leqslant i,j \leqslant m$$

$$\gamma_t(j) = \frac{\alpha_t(j)\beta_t(j)}{p(x_{1:n}|\lambda)} \quad 1 \leqslant t \leqslant n, \quad 1 \leqslant j \leqslant m$$

   (b) M-Step:

$$\pi(i) = \gamma_1(i) \quad 1 \leqslant i \leqslant m$$

$$a_{ij} = \frac{\sum_{t=1}^{n-1} \eta_t(i,j)}{\sum_{t=1}^{n-1} \gamma_t(i)} \quad 1 \leqslant i,j, \leqslant m$$

$$b_j(k) = \frac{\sum_{t:x_t=k} \gamma_t(j)}{\sum_{t=1}^{n} \gamma_t(j)} \quad 1 \leqslant j \leqslant m$$

(Jurafsky and Martin (2009)Rabiner (1989))

Appendix B

Baum-Welch Algorithm for the Two Hidden State HMM

For the HMM with two hidden states, $R_{1:n}$ and $S_{1:n}$ are the hidden states. Each state in the hidden process $S_{1:n}$ can take on one of m_1 possible values, while each state in the hidden process $R_{1:n}$ can take on one of m_2 possible values. The length of both series is still n. We define the following parameters:

$$\begin{aligned}
C_{ij} &= P(R_t = j | R_{t-1} = i) \\
D_{j,k,l} &= P(S_t = l | R_t = j, S_{t-1} = k) \\
A_{ik,jl} &= C_{ij} D_{jkl} = P(R_t = j, S_t = l | R_{t-1} = i, S_{t-1} = k) \\
Z_t &= (R_t, S_t)
\end{aligned}$$

The constraints are $\sum_j C_{ij} = 1, \sum_l D_{jkl} = 1$.

Let $\theta = (\pi, A, B, C, D)$ be the model parameters, where π and B are the initial state distribution and emission distribution, respectively, as defined for the first order HMM. Let $\theta^{(t)}$ be the current values of these parameters at time t in the Baum-Welch Algorithm. Define c to be a constant. Then, the auxiliary function for the E step of the update Baum-Welch Algorithm for the HMM with two hidden states can be written as:

$$\begin{aligned} Q(\theta,\theta^{(t)}) &= \mathbb{E}_{\theta^{(t)}}(\log p_\theta(X_{1:n}, Z_{1:n}|X_{1:n} = x_{1:n})) \\ &= c + \sum_{t=2}^{n}\sum_{i,k}\sum_{j,l} P_{\theta_k}(R_{t-1} = i, S_{t-1} = k, R_t = j, S_t = l|X_{1:n}) \log C_{ik,jl} \end{aligned}$$

Let

$$D_{t,ik,jl} = P_{\theta^{(t)}}(R_{t-1} = i, S_{t-1} = k, R_t = j, S_t = l|X_{1:n}).$$

We have that

$$\log A_{ik,jl} = \log C_{ij} + \log D_{j,k,l}.$$

Then, we can find the value of θ to maximize $Q(\theta, \theta^{(t)})$ (where ν is a Lagrange multiplier to handle the constraints placed on C and D):

$$\begin{aligned} 0 &= \frac{\partial}{\partial C_{ij}}\left(Q(\theta,\theta^{(t)}) - \nu\sum_j C_{ij}\right) \\ 0 &= \left(\sum_{t=2}^{n}\sum_{k}\sum_{l} D_{t,ik,jl}\frac{1}{C_{ij}}\right) - \nu \\ \nu C_{ij} &= \sum_{t=2}^{n}\sum_{k}\sum_{l} D_{t,ik,jl} \\ \nu &= \sum_{j}\sum_{t=2}^{n}\sum_{k}\sum_{l} D_{t,ik,jl} \\ C_{ij} &\propto \sum_{t=2}^{n}\sum_{k,l} D_{t,ik,jl} \quad \forall 1 \leqslant i,j \leqslant m_2 \end{aligned}$$

Likewise,

$$\begin{aligned}
0 &= \frac{\partial}{\partial D_{j,k,l}}\left(Q(\theta,\theta^{(t)}) - \nu \sum_{l} D_{j,k,l}\right) \\
&= \sum_{t=2}^{n}\sum_{i} D_{t,ik,jl}\frac{1}{D_{j,k,l}} - \nu \\
\nu D_{j,k,l} &= \sum_{t=2}^{n}\sum_{i} D_{t,ik,jl} \\
D_{j,k,l} &\propto \sum_{t=2}^{n}\sum_{i} D_{t,ik,jl} \quad \forall 1 \leqslant l,k \leqslant m_1, 1 \leqslant j \leqslant m_2
\end{aligned}$$

The Forward-Backward Algorithm is exactly the same as in the first order HMM case, where A as defined above is the transition matrix used. π and B are updated exactly the same way as in the Baum-Welch Algorithm for the first order HMM (Miller (2016b)).

Appendix C

Sheet Music for Generated Pieces

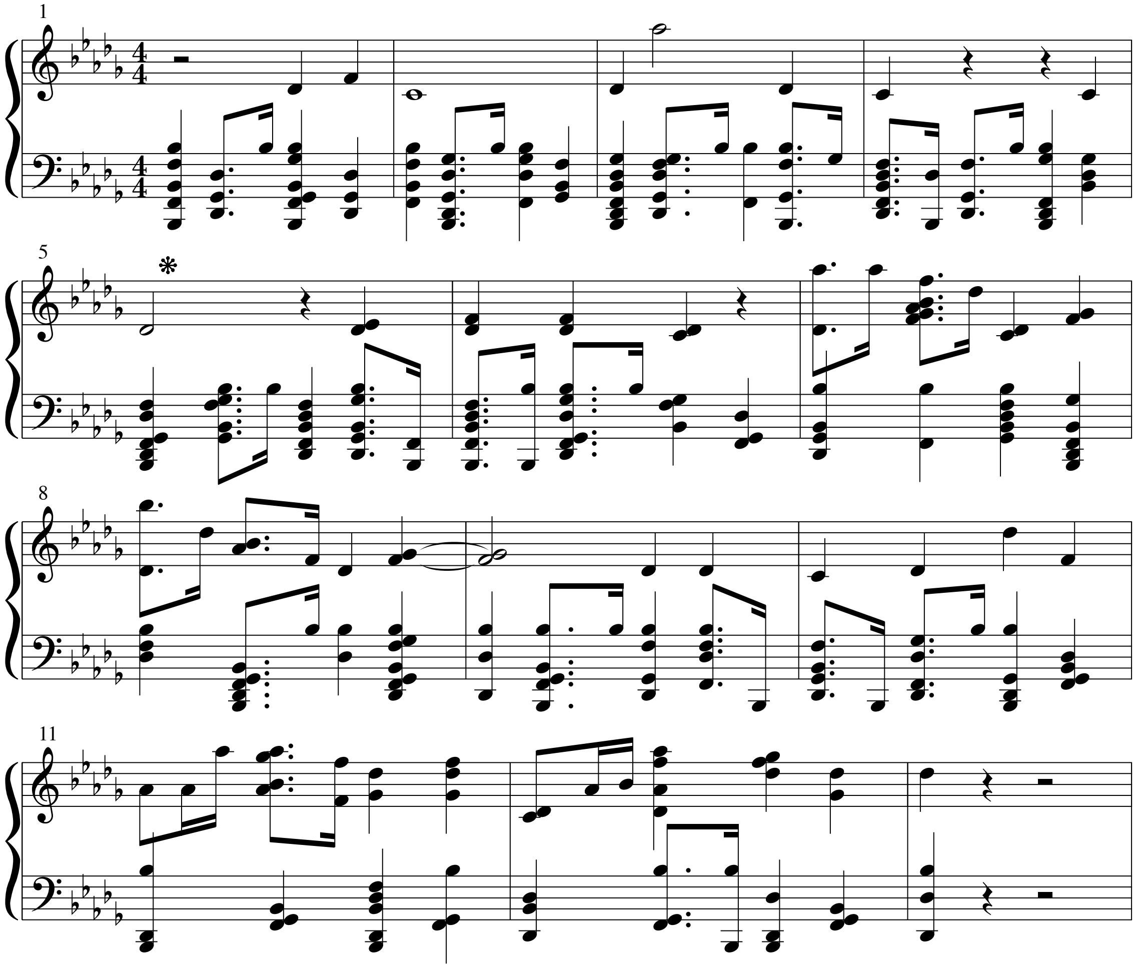

FIGURE C.1: Sheet music for a piece generated by a layered HMM trained on Chopin's Marche funebre.

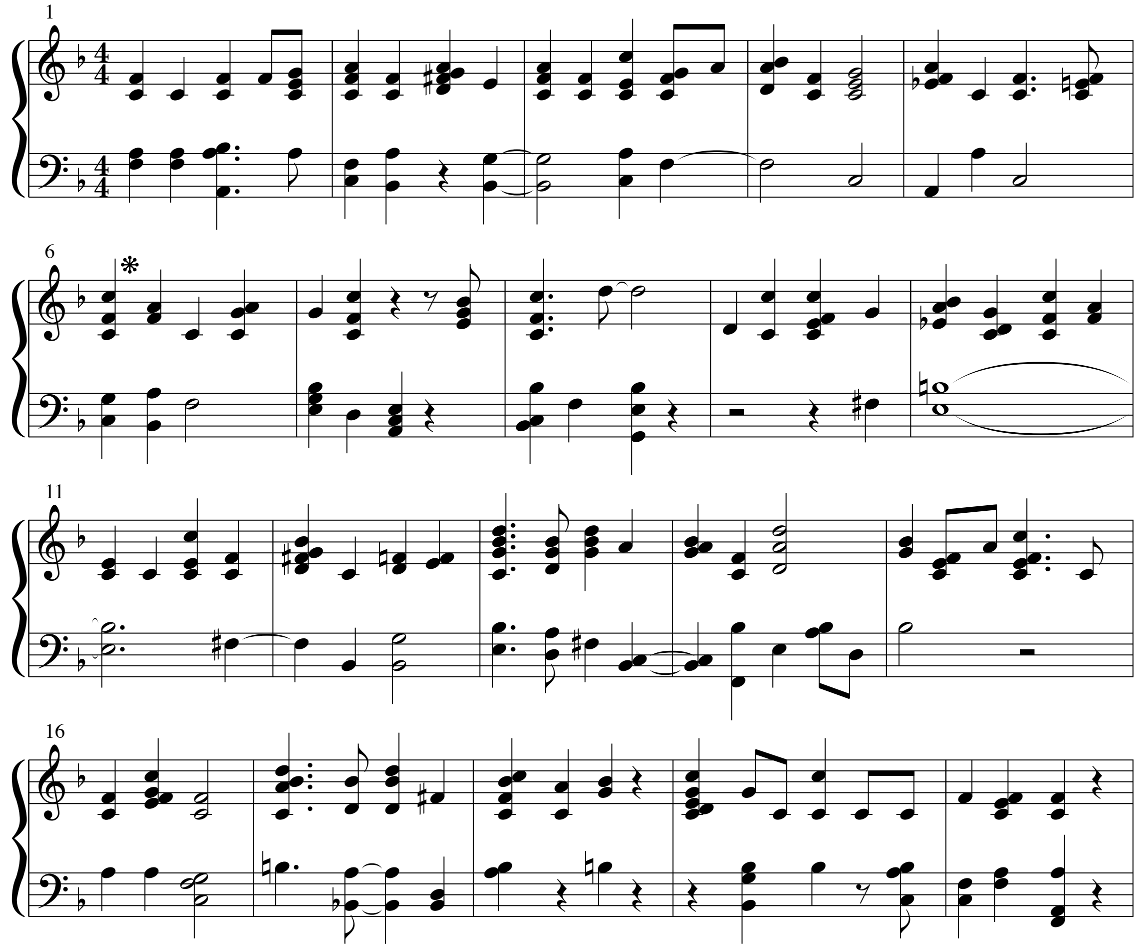

FIGURE C.2: Sheet music for a piece generated by a layered HMM trained on Mendelssohn's Hark! The Herald Angels Sing.

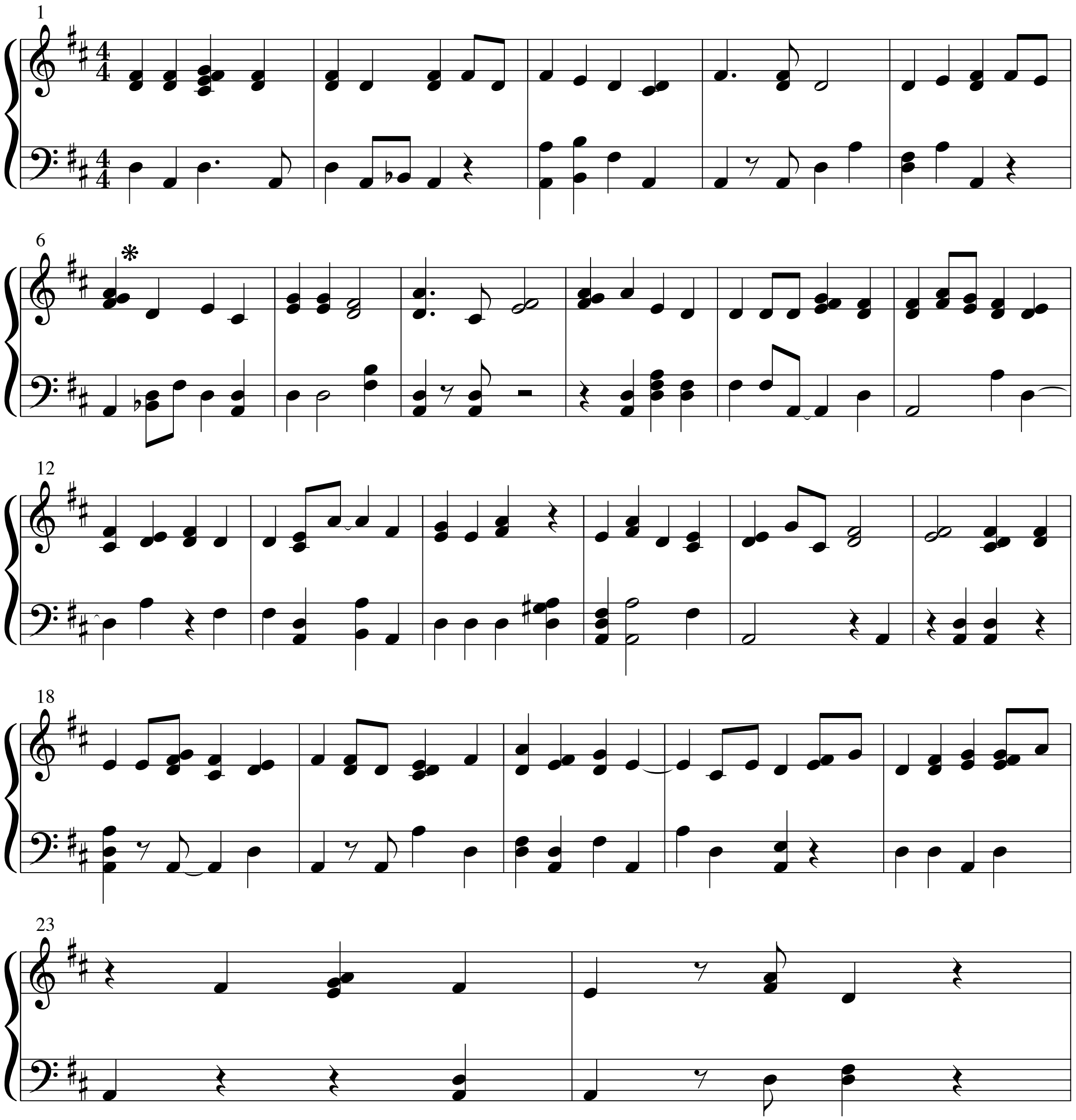

Figure C.3: Sheet music for a piece generated by a layered HMM trained on Beethoven's Ode to Joy.

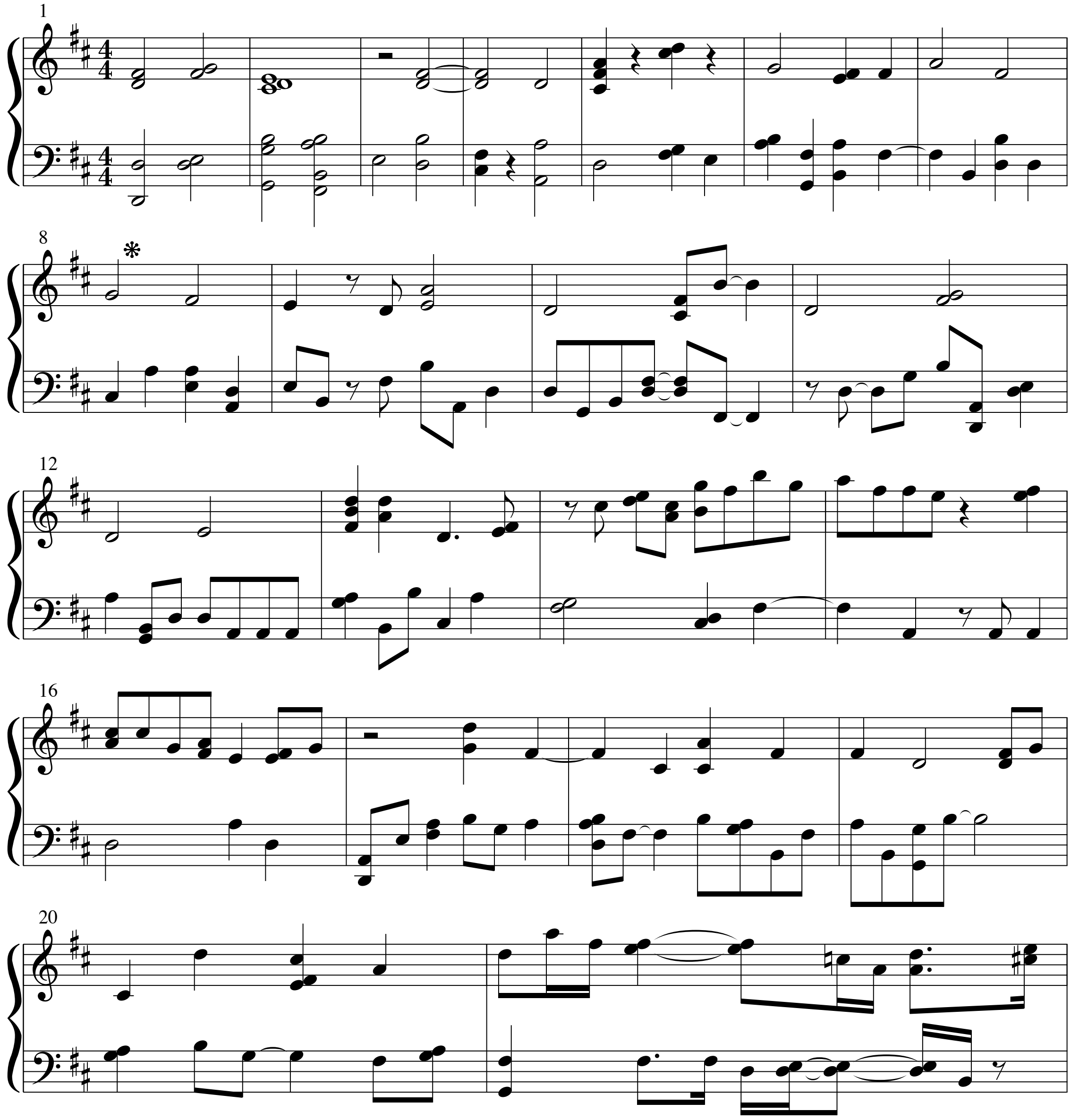

FIGURE C.4: Sheet music for the first 21 bars of a piece generated by a layered HMM trained on Pachelbel's Canon in D.

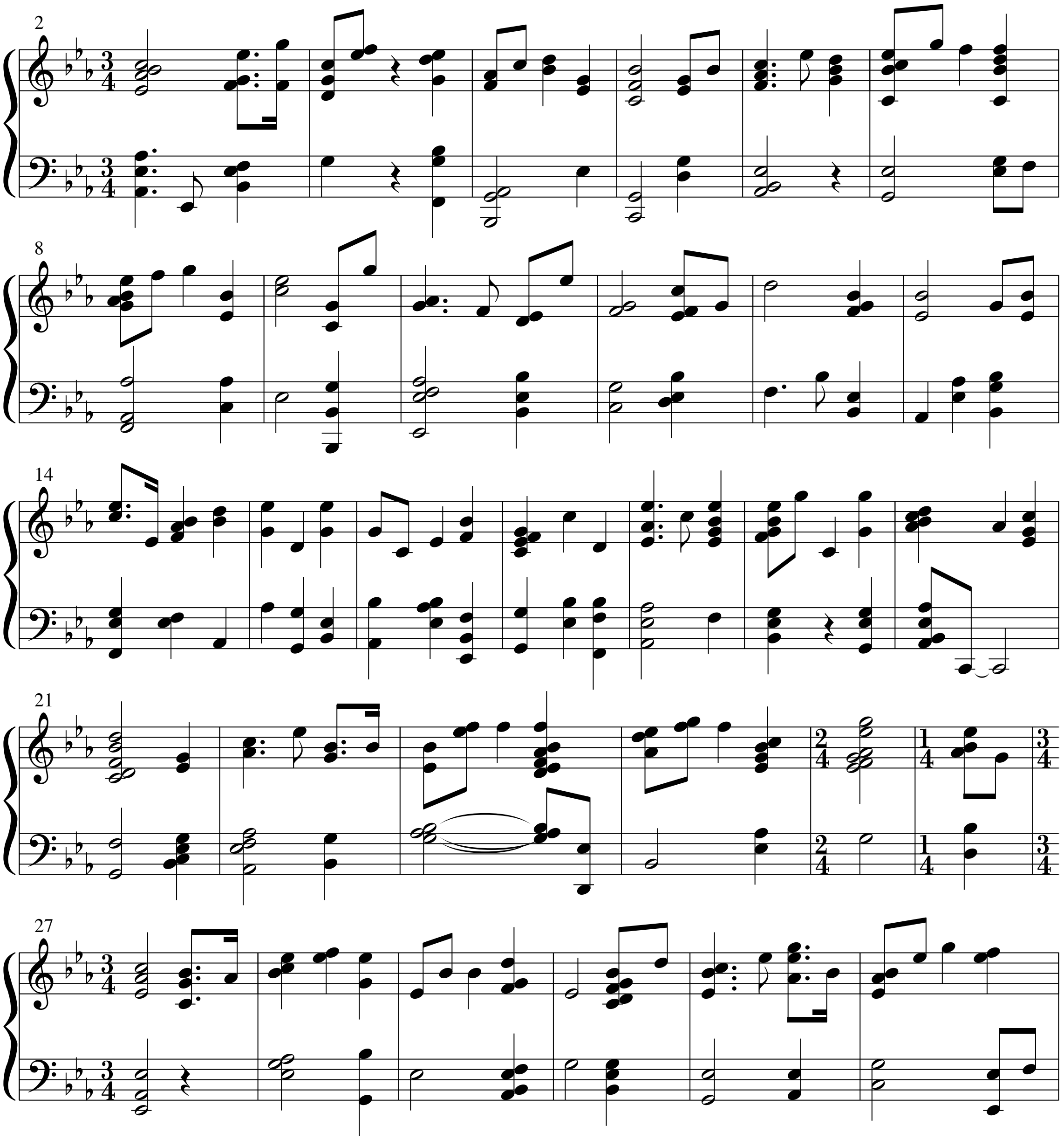

FIGURE C.5: Sheet music for the first 32 bars of a piece generated by a layered HMM trained on the theme from Jupiter from Gustav Holst's The Planets.

Bibliography

Allan, M. and Williams, C. (2004), "Harmonising Chorales by Probabilistic Inference," in *Advances in Neural Information Processing Systems 17*, pp. 25–32.

Ames, C. (1989), "The Markov Process as a Compositional Model: A Survey and Tutorial," *Leonardo*, 22, 175–187.

Baroni, M., Maguire, S., and Drabkin, W. (1983), "The Concept of Musical Grammar," *Music Analysis*, 2, 175–208.

Baum, L. E. (1972), "An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes," *Inequalities*, 3, 1–8.

Beran, J. (2004), *Statsitics in Musicology*, Chapman & Hall/CRC.

Bernstein, L. (1976), *The Unanswered Question: Six Talks at Harvard*, Harvard University Press.

Boulanger-Lewandowski, N., Bengio, Y., and Vincent, P. (2012), "Modeling Temporal Dependencies in High-Dimensional Sequences: Application to Polyphonic Music Generation and Transcription," in *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pp. 1159–1166.

Coffman, D. D. (1992), "Measuring Musical Originality Using Information Theory," *Psychology of Music*, 20, 154–161.

Cope, D. (1991), *Computers and Musical Style*, The Computer Music and Digital Audio Series, A-R Editions, Inc.

Dempster, A., Laird, N. M., and Rubin, D. B. (1977), "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, 39, 1–21.

Developers, G. (2017), "Magenta: Music and Art Generation (TensorFlow Dev Summit 2017)," `https://www.youtube.com/watch?v=vM5NaGoynjE`.

Dirst, M. and Weigend, A. S. (1993), *Time Series Prediction: Forecasting the Future and Understanding the Past*, chap. Baroque Forecasting: On Completing J. S. Bach's Last Fugue, Addison-Wesley.

Djuric, P. M. and Chun, J.-H. (1999), "Estimation of Nonstationary Hidden Markov Models by MCMC Sampling," in *ICASSP '99 Proceedings of the Acoustics, Speech, and Signal Processing, 1999. on 1999 IEEE International Conference - Volume 03*, pp. 1737–1740.

Djuric, P. M. and Chun, J.-H. (2002), "An MCMC Sampling Approach to Estimation of Nonstationary Hidden Markov Models," *IEEE Transactions on Signal Processing*, 50, 1113–1123.

Eck, D. and Schmidhuber, J. (2002), "A First Look at Music Composition using LSTM Recurrent Neural Networks," Tech. rep., Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale.

Fine, S., Singer, Y., and Tishby, N. (1998), "The Hierarchical Hidden Markov Model: Analysis and Applications," *Machine Learning*, 32, 41–62.

Gatys, L. A., Ecker, A. S., and Bethge, M. (2015), "A Neural Algorithm of Artistic Style," *CoRR*, abs/1508.06576.

Gauldin, R. (2004), *Harmonic Practice in Tonal Music*, W. W. Norton & Company, second edn.

Ghahramani, Z. (1997), "Learning Dynamic Bayesian Networks," Tech. rep., University of Toronto.

Ghahramani, Z. and Jordan, M. I. (1997), "Factorial Hidden Markov Models," *Machine Learning*, pp. 1–31.

Goodfellow, I., Bengio, Y., and Courville, A. (2016), *Deep Learning*, MIT Press.

Guan, X., Raich, R., and Wong, W.-K. (2016), "Efficient Multi-Instance Learning for Activity Recognition from Time Series Data Using an Auto-Regressive Hidden Markov Model," in *Proceedings of The 33rd International Conference on Machine Learning*, vol. 48.

Hadjeres, G. and Pachet, F. (2016), "DeepBach: a Steerable Model for Bach chorales generation," *CoRR*, abs/1612.01010.

Hihi, S. E. and Bengio, Y. (1996), "Hierarchical Recurrent Neural Networks for Long-Term Dependencies," in *Advances in Neural Information Processing Systems 8 (NIPS'95)*, MIT Press.

Johnson, D. (2015), "Composing Music with Recurrent Neural Networks," `http://www.hexahedria.com/2015/08/03/composing-music-with-recurrent-neural-networks/`.

Jurafsky, D. and Martin, J. H. (2009), *Speech and Language Processing*, Prentice Hall, second edn.

Krogh, A., Brown, M., Mian, I., Sjolander, K., and Haussler, D. (1994), "Hidden Markov models in computational biology: Applications to protein modelling," *Journal of Molecular Biology*, 235, 381–383.

Krueger, B. (2016), "Classical Piano MIDI Page," `http://www.piano-midi.de/midi_files.htm`.

Laitz, S. G. (2003), *The Complete Musician*, Oxford University Press.

Lari, K. and Young, S. J. (1990), "The estimation of stochastic context-free grammars using the Inside-Outside algorithm," *Computer Speech and Language*, 4, 35–56.

Lerdahl, F. and Jackendoff, R. (1983), *A Generative Theory of Tonal Music*, The MIT Press.

Magenta (2016), "Magenta," `https://magenta.tensorflow.org/welcome-to-magenta`.

Mari, J.-F. and Schott, R. (2001), *Probabilistic and Statistical Methods in Computer Science*, Springer.

Meehan, J. R. (1980), "An Artificial Intelligence Approach to Tonal Music Theory," *Computer Music Journal*, 4, 60–65.

mfiles.co.uk (2017), "mfiles," `http://www.mfiles.co.uk/classical-midi.htm`.

MIDIworld (2009), "midiworld.com," `http://www.midiworld.com/classic.htm/beethoven.htm`.

Miller, J. W. (2016a), *Lecture Notes on Advanced Stochastic Modeling.*

Miller, J. W. (2016b), "Two Hidden State Markov Model," Discussions.

Mozer, M. C. (1994), "Neural network music composition by prediction: Exploring the benefits of psychophysical constraints and multiscale processing," *Cognitive Science*, 6, 247–280.

Murphy, K. P. (2002), "Hidden semi-Markov models (HSMMs)," Tech. rep., MIT.

Murphy, K. P. (2012), *Machine Learning: A Probabilistic Perspective*, MIT Press.

Nierhaus, G. (2009), *Algorithmic Composition: Paradigms of Automated Music Generation*, SpringerWienNewYork.

Oliver, N., Garg, A., and Horvitz, E. (2004), “Layered representations for learning and inferring office activity from multiple sensory channels,” *Computer Vision and Image Understanding*, 96, 163–180.

Pearce, M. T. and Wiggins, G. A. (2006), “Expectation in Melody: The Influence of Context and Learning,” *Music Perception: An Interdisciplinary Journal*, 23, 377–405.

Pikrakis, A., Theodoridis, S., and Kamarotos, D. (2006), “Classification of musical patterns using variable duration hidden Markov models,” *IEEE Transactions on Audio, Speech, and Language Processing*, 14, 1795–1807.

Prado, R. and West, M. (2010), *Time Series: Modeling, Computation and Inference*, Chapman & Hall/CRC.

Rabiner, L. R. (1989), “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition,” in *Proceedings of the IEEE*, vol. 77, IEEE.

Ratner, L. G. (1992), *Romantic Music: Sound and Syntax*, Schirmer Books.

Rothstein, J. (1992), *MIDI: A Comprehensive Introduction*, The Computer Music and Digital Audio Series, A-R Editions, Inc.

Shumway, R. H. and Stoffer, D. S. (2011), *Time Series Analysis and Its Applications*, Springer.

Sin, B. and Kim, J. H. (1995), “Nonstationary hidden Markov model,” *Signal Processing*, 46, 31–46.

Suzuki, S. and Kitahara, T. (2014), “Four-part Harmonization Using Bayesian Networks: Pros and Cons of Introducing Chord Notes,” *Journal of New Music Research*, 43, 331–353.

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. W., and Kavukcuoglu, K. (2016), “WaveNet: A Generative Model for Raw Audio,” *CoRR*, abs/1609.03499.

Vincent, J. (2016), “A night at the AI jazz club,” `http://www.theverge.com/2016/10/12/13247686/ai-music-composition-jazz-club-london-deep-learning`.

Walker, J. (2008), “MIDI-CSV,” `http://www.fourmilab.ch/webtools/midicsv/#midicsv.5`.

Warrack, J. (1983), *The New Oxford Compantion to Music*, Oxford University Press.

Weiland, M., A.Smaill, and Nelson, P. (2005), "Learning musical pitch structures with hierarchical hidden Markov models," *Journees d'Informatique Musical.*

Whorley, R. P. and Conklin, D. (2016), "Music Generation from Statistical Models of Harmony," *Journal of New Music Research*, 45, 160–183.

Yu, S.-Z. (2010), "Hidden semi-Markov models," *Artificial Intelligence*, 174, 215–243.