Factorial Hidden Markov Models for Algorithmic Composition

Anna K. Yanchenko
Duke University
anna.yanchenko@duke.edu

Abstract

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. We explore the ability of state space models with distributed representations, specifically factorial Hidden Markov Models, to generate new pieces of music with improved harmonic and melodic characteristics relative to simpler state space models. Additionally, we compare various approximate inference methods for training factorial Hidden Markov Models for this task. We find that factorial Hidden Markov Models generate pieces with improved melodic structure characteristics and increased originality relative to simpler models.

1 Introduction

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. Success in algorithmic composition is interesting both from a generative modeling perspective, as well as a musicology and music information retrieval perspective. Music is a complex, highly structured time series, which makes the successful modeling and generation of music challenging and applicable to other domain areas. While the majority of the current work in this area utilizes deep learning approaches, simpler state space models such as Hidden Markov Models (HMMs) have previously been found to work fairly well [2].

Deep learning methods utilize distributed representations of latent states, allowing for the representation of much larger state spaces than is possible without this distributed representation. Distributed representations have proven to be very successful in natural language processing applications [1] and correspond to aspects of music theory. For example, a piece of music can be broken down into the melodic, harmonic and rhythmic aspects, each of which can evolve at a different rate. To this end, distributed state representations suggest a promising approach in state space models for algorithmic composition.

The goal of this project is to explore the utility of factorial Hidden Markov Models (FHMMs) [6] for algorithmic composition, as compared to state space models without distributed representations. The success of the compositions generated by FHMMs is evaluated using previously developed musical metrics [2] and compared to pieces generated by simpler HMMs. A secondary goal of this project is to compare and explore various inference methods within the context of training FHMMs. State space models quickly become computationally intractable for only a few hidden states, requiring the use of approximate inference methods. Various approximate inference techniques for FHMMs [6], [5] are compared in terms of efficiency and quality of generated pieces. We find that FHMMs are less prone to overfitting than simpler models and are able to generate pieces with more melodic structure. However, the FHMMs are not able to model harmony well and produce pieces that tend to be dissonant and not in the style of the original training pieces.

This report is organized as follows: previous work in algorithmic composition and FHMMs is discussed in 2. The specific musical metrics, data processing methods and inference approximations explored are discussed in 3. Finally, results are presented and discussed in 4 and a summary and suggestions for future work are in 5.

2 Previous Work

HMMs have proven to be extremely popular models, both for their relative simplicity of implementation and ability to model complex time series. [6] proposed FHMMs as an extension to HMMs, utilizing a distributed state space, allowing for a much larger state space to be efficiently represented. The graphical model for a FHMM is shown in Figure 1. In the original paper [6], the authors propose Gibbs sampling and two methods of variational inference for models with continuous observation vectors. These methods have been extended to models with discrete observations [5, 1].

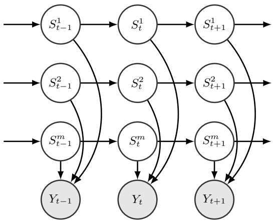


Figure 1: FHMM [1], Y_t are the observed states and S_t are the hidden states.

Algorithmic composition has long been a problem of interest. The first musical composition completely generated by a computer was the “Illiac Suite", produced from 1955 - 1956 [4]. Since then, many methods and models for algorithmic composition have been used, including generative grammars, artificial neural networks and HMMs [4]. More recently, deep learning methods have dominated the area, in particular, the work of Magenta, Google’s team utilizing machine learning and deep learning for creativity and design [3].

We have previously explored various types of state space models for algorithmic composition in [2]. In this work, we explored FHMMs that were completely independent and not coupled in the emission distribution at all (essentially the average of independent first order HMMs). These models tended to generate pieces that had among the highest temporal coherence of the methods considered, suggesting FHMMs with coupling between hidden states in the emission distribution as a promising next step for algorithmic composition with state space models. State space models are explored for their relative simplicity of implementation and interpretability, as compared to deep learning methods, considered by [3], for example.

3 Methods

3.1 Model

We start with the FHMM as defined in [6]; the graphical model is Figure 1. The joint probability for the FHMM is

$$P(\{S_t, Y_t\}) = P(S_1)P(Y_1|S_1)\prod_{t=2}^{T} P(S_t|S_{t-1})P(Y_t|S_t), \tag{1}$$

where $\{S_t\}$ is the sequence of hidden states and $\{Y_t\}$ is a sequence of continuous, real-valued observations. The state space variables are assumed to evolve according to their own, independent dynamics, and we have

$$P(S_t|S_{t-1}) = \prod_{m=1}^{M} P\left(S_t^{(m)}|S_{t-1}^{(m)}\right), \tag{2}$$

where M is the total number of hidden state chains. We further let $K^{(m)}$ be the number of hidden states possible for chain m.

In [6], the parameters of the FHMM are estimated using the Expectation-Maximization Algorithm. The E-Step is computationally intractable, as it involves summing over all possible hidden state configurations at each time step, t. Thus, [6] considers three approximate inference methods for the E-Step: (1) Gibbs sampling (FHMM-Gibbs), (2) completely factorized variational inference (FHMM-FVI) and (3) structured variational inference (FHMM-SVI). The completely factorized variational inference approach assumes that, conditional on the observed data, all of the hidden state variables are independent. The structured variational approach treats each chain independently, though with the Markov structure of the state variables retained [6]. With these approximations to the E-Step, the M-Step has closed-form update equations.

[5] take a different approach to the problem of inference in FHMMs. They draw parallels between FHMMs and generalized additive models (GAMs) to develop a generalized backfitting algorithm to perform inference in the FHMM (FHMM-GAM). The advantage of this approach is that it is naturally extended to various types of data, beyond continuous, real-valued observations, such as binary and multinomial observations sequences. The Forward-Backward Algorithm is utilized as a subroutine in the generalized backfitting algorithm [5].

3.2 Data

The observed data consisted of a sequence of MIDI note pitches for a particular piece of music. These MIDI note pitches were integers between 0 and 127, where each integer corresponded to a particular note pitch. Using MIDI audio files from http://www.piano-midi.de/midi_files.htm, http://www.mfiles.co.uk/classical-midi.htm and http://www.midiworld.com/classic.htm/beethoven.htm, three training pieces were considered: “Twinkle, Twinkle, Little Star", “Pachelbel’s Canon" and Bach’s Fugue No. 2 in C Minor, BMV 871 from the Second Book of the Well-Tempered Clavier, all arranged for piano. These pieces represented a range of rhythmic, melodic and harmonic complexity, from very simple (Twinkle, Twinkle, Little Star) to complex (Bach’s Fugue No. 2).

The input MIDI files were converted to CSV using open source software from http://www.fourmilab.ch/webtools/midicsv/#midicsv.5 so that the sequence of note pitches could be easily extracted. The observed notes were assumed to be equally spaced in time, although this was not true for chords in the piece, for example. Furthermore, although there was timing and dynamic information in the CSV file for each piece, these aspects were not explicitly modeled. The sequence of note pitches was the only component of the observation sequence that was modeled by the FHMM.

Additionally, as the models considered assumed continuous, real-valued data, for the FHMM models, the input sequence of note pitches was first normalized for inference, then binned to the nearest pitch occurring in the original piece after the generation of a new piece. That is, we assumed that only note pitches that were observed in the original piece could occur in the generated piece and this constraint was enforced as a post-processing step.

3.3 Music Theory

The metrics developed below to quantitatively evaluate the quality of the generated pieces require high-level knowledge of some music theory concepts. The two main concepts of music that the metrics seek to capture are harmony and melody. Melody can be considered to be a combination of pitches and rhythms that combine to give a piece distinctive themes and phrases. However, the metrics developed below for melody are based on time series correlations and do not depend on knowledge of music theory.

The musical metrics, however, do try to measure basic aspects of musical harmony, which will be briefly reviewed here. All of the original training pieces considered are tonal, meaning that there is a central, “tonic" pitch that the music is oriented around and that is frequently repeated throughout a piece, particularly at the end of musical phrases. The metrics consider a high-level view of harmony that focuses on musical intervals. There are two types of musical intervals: harmonic intervals, where two or more pitches are sounded simultaneously, and melodic intervals, where pitches are sounded sequentially in time. Intervals can either be consonant or dissonant. Consonant intervals are the most stable and do not require a resolution; dissonant intervals, on the other hand, sound unstable and do require a resolution to a more stable harmony. Dissonant intervals sound incomplete or transient, while consonant intervals tend to sound complete on their own. In the original training pieces considered, all dissonant intervals are resolved to more stable consonant intervals, a characteristic that we want to be replicated in the generated pieces.

3.4 Metrics

Several musical metrics explored in [2] were used to quantitatively compare the generated pieces. These metrics sought to evaluate three main components of the generated pieces: originality, musicality and temporal structure and were all in reference to the original training piece.

Originality Metrics: The three originality metrics were based on information theory and distances between strings. These metrics were the empirical entropy of a piece (intended to capture the "creativity" of a piece, with higher entropy indicating more originallity), and the mutual information and edit distance between a generated and original training piece, to capture the originality of the generated pieces.

Musicality Metrics: We considered several related metrics to capture harmonic aspects of the generated pieces. The first class of musical metrics were counts of melodic and harmonic intervals, normalized by the length of the piece, with the amount of dissonance in the generated piece expected to be similar to the amount in the original training piece. The other metric was the distribution of note pitches, where, again, pitches that were used less in the original piece were expected be less prevalent in the generated pieces.

Temporal Structure Metrics: We used measures of decay correlations in time series to capture the amount of temporal and melodic structure in a musical piece. The autocorrelation function (ACF) of a sequence and the partial autocorrelation function (PACF) were calculated out to lag 40 to give a sense of the melodic progression of a piece. Original pieces (all of which had clear melodies) exhibited structure in the ACF and PACF plots out to high lags (Figure 3), with similar structure in the generated pieces desired and indicative of melodic structure over time.

Inference Efficiency Metrics: In order to compare the relative efficiency of the various inference approximation methods, the average time per iteration was considered. A single pass of the E and M steps in the FHMM-Gibbs, FHMM-FVI and FHMM-SVI models was considered one iteration, while a single pass of the algorithm in Figure 2 in [5] was counted as one iteration. Ideal approximation methods would yield generated pieces with high musical metrics and fast convergence.

3.5 Procedure

The overall procedure for the conducted experiments was as follows. First, a FHMM model (with a specific inference approximation) was trained to a specified level of convergence on a single, given training piece. Then, using the learned parameters, 1000 new pieces were generated and post-processed to only include the observed note pitches in the original piece. The metrics described in subsection 3.4 were used to quantitatively evaluate the generated pieces as compared to the original training piece. Additionally, one generated piece was converted back to MIDI format using the software from http://www.fourmilab.ch/webtools/midicsv/#midicsv.5 and evaluated subjectively through listening and analysis of the generated score in GarageBand (https://www.apple.com/mac/garageband/).

After the considered model converged, 1000 new pieces were sampled from the learned model using the appropriate generative description of each model and the root mean squared error (RMSE) for each metric (except mutual information and edit distance) was calculated. The RMSE was primarily used to rank the generated pieces to give insight into some general trends observed in the generated pieces. The RMSE can be calculated as

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y_0)^2} \tag{3}$$

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 $n = 1000$. The RMSE was calculated for the musicality and temporal structure metrics, as well as for the empirical entropy, but not for the mutual information or minimum edit distance. For the mutual information and minimum edit distance, both metrics were calculated for each of the 1000 generated pieces with respect to the original training piece, then the average of these 1000 values was taken to use for comparison between models and training pieces, in lieu of the RMSE.

The various FHMM models were compared to a first order HMM, an HMM with completely random parameters (no learning from the original training piece), an “independent factorial" HMM with no coupling between chains for the observed states and a layered HMM. The “random" HMM served as a baseline, with all musical metrics expected to be the worst for this model (except for the originality metrics). The first order HMM, independent FHMM and layered HMM were among the best performing state space models considered in our previous work [2]. All FHMM models had three hidden state chains, with five hidden states each, while all of the simpler HMMs also had five hidden states.

4 Results

The success of the FHMMs in terms of the musical metrics, efficiency and overall composition quality is discussed below. The four FHMM inference methods were compared to each other and to the simpler HMMs. Overall, the FHMMs offered improvements over the simpler models in several areas of algorithmic composition and were less successful than the simpler models in other areas.

4.1 Musical Metrics Results

Overall, the four FHMMs considered improved in some, but not all, of the quantitive metrics considered, relative to simpler HMMs (Table 1). In particular, the FHMMs showed major improvement in the “originality" metric category. The FHMM-Gibbs, FHMM-FVI and FHMM-SVI generated pieces trained on Bach’s Fugue No. 2 in C Minor, for example, had the highest empirical entropy values, second only to the random HMM. Additionally, all four FHMM methods had among the lowest mutual information and the highest edit distance values as compared to the training piece of the models considered. While layered and independent factorial models generated pieces that were very similar to the training piece and seemed to “overfit" [2], the four FHMM models considered did not suffer from this problem and generated pieces that sounded distinct from the training piece.

However, the four FHMM methods considered did not offer much improvement in way of musicality. The generated pieces tended to be much more dissonant than the original training pieces and more dissonant than the best performing simpler HMM models (in particular, the layered HMM, Table 1). The FHMM-GAM did tend to produce more consonant pieces, though the generated pieces were quite repetitive and did not utilize all of the notes in the original piece, so this was only a marginal success. The distributed state space representation did not seem to offer improvements in modeling the harmony of the training pieces, as compared to simpler HMMs. While the distributed state space allowed for a more efficient state space representation with less hidden states relative to a first order HMM, this expansion of the state space did not appear to offer increased ability to model harmony.

Finally, the FHMM-SVI in particular demonstrated major improvements in the temporal metrics category. While the actual ACF and PACF RMSE values did not show improvement in Table 1, when the full ACF and PACF plots were considered, the FHMM-SVI was able to generate pieces with correlation structure out to a few lags, which was not true of any of the other models considered (Figure 4). While there is still significant room for improvement in terms of generating pieces with long term structure on the order of that in the original training pieces, FHMMs, specifically those trained via structured variational inference, showed a step in the right direction.

The trends discussed above were seen in all of the training pieces, from simple (Twinkle, Twinkle, Little Star) to complex (Bach’s Fugue No. 2 in C Minor). The FHMM-GAMs tended to perform the best on some metrics and the worse on others consistently, and thus had a wide spread of RMSE metric rankings, while the FHMM-SVI tended to perform in the upper half of the models considered and consistently the best among the four FHMMs (Figure 2).

4.2 Inference Efficiency Results

In terms of inference efficiency, the FHMM-GAM performed the best across shorter and longer sequences. The sequence length of Twinkle, Twinkle was 180 notes, of Pachelbel’s Canon was 1050 notes and of Bach’s Book 2 Fugue 7 was 1408 notes. The FHMM-SVI was the next most efficient in

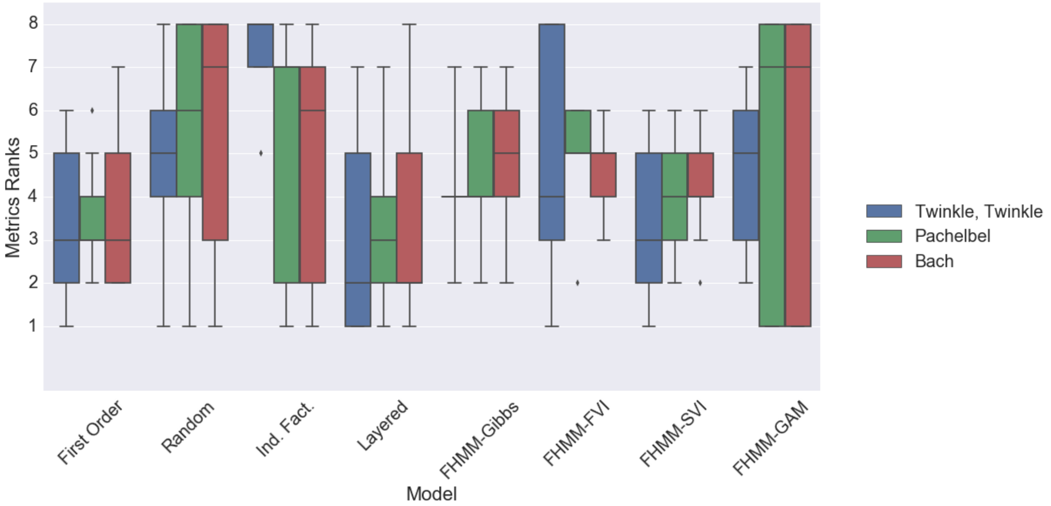
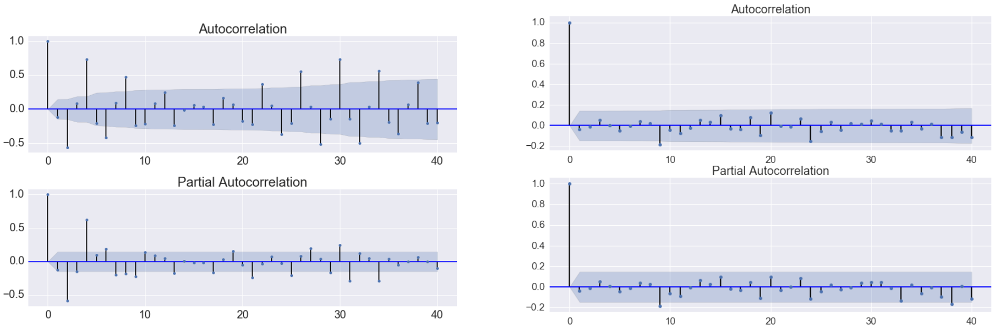


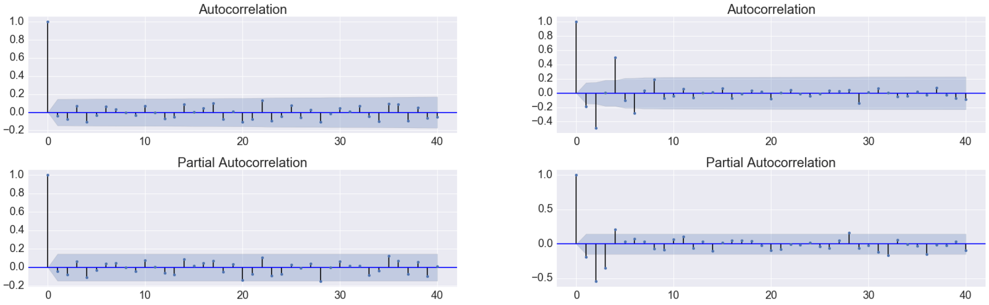
Figure 2: Boxplots of rankings of models across all 9 metrics for the 3 evaluated pieces. The FHMM-SVI tended to perform the best on the considered metrics among the four FHMM inference methods, while the FHMM-GAM performed the best on some metrics and the worst on others. Relative performance of the different models was fairly consistent across the marginally complex Pachelbel and the quite musically complex Bach piece.



(a) Original Piece

(b) FHMM-GAM Generated Piece

Figure 3: ACF and PACF plots for the original training piece Twinkle, Twinkle and a FHMM-GAM generated piece. The FHMM-GAM generated piece exhibits none of the temporal correlation structure evident in the original piece.



(a) FHMM-FVI Generated Piece

(b) FHMM-SVI Generated Piece

Figure 4: ACF and PACF plots for the FHMM-FVI and FHMM-SVI generated pieces trained on Twinkle, Twinkle. The FHMM-SVI generated piece has the most temporal structure of the three generated pieces considered here.

Table 1: Evaluation metrics for each of the models considered for Bach's Fugue No. 2 in C Minor 1000 generated pieces. All models had 5 hidden states and the FHMMs had 3 chains of hidden states, with 5 hidden states each. Ind. Fact. is the independent factorial HMM, FHMM-FVI is the Factorized VI approximation and FHMM-SVI is the Structured VI approximation. Ent. corresponds to the average entropy of the generated pieces, while MI and ED correspond to the mutual information and edit distance, respectively, calculated with respect to the original piece and averaged over all the generated pieces. The remaining metrics are the RMSE between the original piece and the generated pieces, averaged over all metrics in that category. For example, ACF is calculated out to lag-40 and the value reported is average RMSE value over all 40 lags. Per. refers to the percentage of intervals (both harmonic and melodic) that are perfect consonances, imperfect consonances and dissonances. H. Ints refers to harmonic intervals and M. Ints refers to melodic intervals. NC refers to the note counts in the piece. The best performing model for each metric is highlighted in blue and the worst in red.

Model	Ent.	MI	ED	H. Ints	M. Ints	Per.	NC	ACF	PACF
First Order	3.170	0.353	0.870	85.942	170.635	0.119	0.005	0.100	0.059
Random	3.587	0.428	0.926	90.394	181.423	0.125	0.022	0.122	0.067
Ind. Fact.	3.079	0.524	0.871	62.857	162.630	0.154	0.018	0.046	0.047
Layered	3.182	0.404	0.845	78.619	155.151	0.113	0.004	0.081	0.056
FHMM-Gibbs	3.420	0.401	0.903	89.663	180.746	0.128	0.016	0.119	0.066
FHMM-FVI	3.365	0.384	0.900	87.145	180.686	0.130	0.015	0.118	0.064
FHMM-SVI	3.412	0.392	0.904	85.625	180.335	0.131	0.016	0.112	0.060
FHMM-GAM	2.088	0.132	0.896	48.143	117.022	0.175	0.035	0.122	0.067

terms of time per iteration, while the FHMM-Gibbs took significantly more time per iteration than any other method. The simpler HMMs were, unsurprisingly, more efficient in time per iteration than the FHMMs.

Table 2: Average time per iteration in seconds for each model and inference algorithm considered. The random HMM does not learn from the data, and thus has 0 iterations.

	Twinkle, Twinkle	Pachelbel's Canon	Bach Book 2 Fugue 7
First Order	0.120	0.320	0.466
Random	0.000	0.000	0.000
Layered	0.049	0.296	0.425
Independent Factorial	0.107	0.838	0.967
FHMM-Gibbs	3.467	19.623	26.244
FHMM-FVI	1.270	7.314	9.736
FHMM-SVI	1.057	3.412	3.777
FHMM-GAM	0.393	2.202	2.934

4.3 Discussion

The results discussed above were additionally evident when listening to the generated pieces. All of the FHMM generated pieces were much more dissonant than the original training pieces and had no clear sense of melody. There was little musical coherence or structure over even the span of a few notes. Furthermore, any musical style evident in the original training pieces was lost in the generated pieces. The generated pieces sounded more like they were from the Contemporary era rather than the Baroque or Classical eras.

The sheet music excerpt in Figure 5 offers a representative example. While there is a short phrase of ascending and descending eighth notes in measure 9 (the second measure shown), there is no clear structure over a period of longer than a few subsequent beats. There is a strong dissonance in measure 10 that is not heard in the original training piece and there are larger interval jumps between subsequent notes than expected. In contrast, in the original piece (Figure 6), there are clear chord progressions and melodic lines that last for several measures. The melody is in the ascending and descending eighth notes patterns that repeat and evolve over this section of the piece and there is no dissonance and very few large interval jumps. While the generated (Figure 5) and original (Figure 6) pieces share the same notes and similar rhythms, they are also clearly distinct and the FHMM did generate pieces that are more “original" than some of the simpler HMM models, while still performing similarly in terms of the musical, quantitative metrics. Furthermore, the FHMM models did not suffer from overfitting, a result also found in [6].

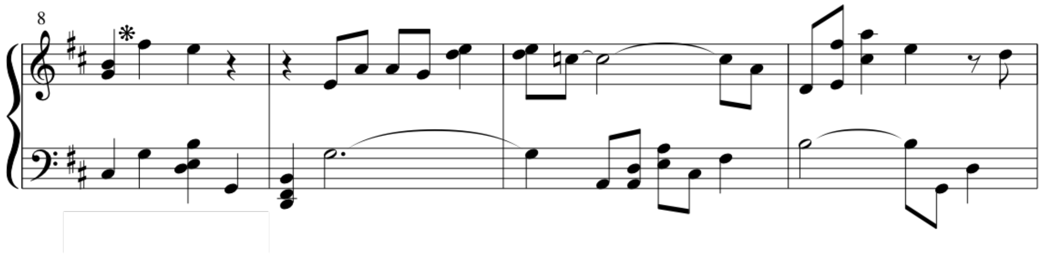


Figure 5: Sheet music excerpt of a piece generated by an FHMM-SVI trained on Pachelbel’s Canon.

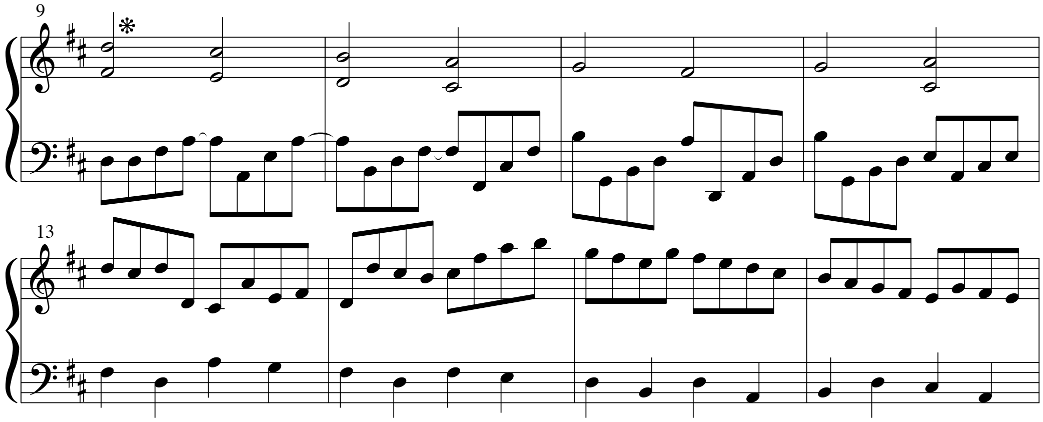


Figure 6: Sheet music excerpt of original Pachelbel’s Canon.

The FHMM-SVI generally tended to produce the most musically pleasing pieces to listen to, of the four FHMM inference methods considered. The pieces generated by the FHMM-Gibbs had even less temporal structure and more large interval jumps, while the FHMM-FVI generated pieces tended to perform somewhere in between the FHMM-SVI and FHMM-Gibbs. The FHMM-GAM tended to produce pieces that were more repetitive and did not exhibit the full range of possible notes. However, the generated pieces, even though they contained less distinct notes, were slightly more consonant than the FHMM-SVI generated pieces.

Finally, there did not appear to be a clear interpretation of the hidden states in the FHMMs considered. Looking at the learned parameters for each inference method, there were no clear trends in the transition matrices or emission parameters that corresponded to musical ideas. For the first order HMM, there were clearer music theory interpretations; for example, one hidden state emitted the tonic pitch with high probability and another tended to emit notes in the primary chord with high probability [2]. There was no clear analogy in the case of the FHMMs, a result also found in the original paper, when looking at Bach chorales [6].

However, in the FHMM-SVI in particular, the transition matrices for the three hidden state sequences did look distinct from each other. That is, each hidden state chain appeared to be modeling a different aspect of the piece and evolving over different time scales, as at least one chain’s transition matrix tended to have fairly uniform transition probabilities, while at least one chain had a fairly sparse transition matrix. If the separate hidden state chains could be encoded with more music theory information, this ability of the chains to evolve over different time scales offers exciting possibilities for algorithmic composition.

5 Conclusions and Future Work

Overall, the FHMM models considered offered improvements in certain aspects of algorithmic composition. The FHMM-SVI was able to generate pieces with longer melodic structure than the simpler models considered and all four of the FHMMs generated pieces that were distinct from the original training piece and did not suffer from overfitting. However, while the distributed state space offered improvements in terms of originality and temporal structure, it was not able to model harmony well. In terms of inference efficiency and musical metrics results, the FHMM-SVI performed the best of the four FHMMs considered on the task of algorithmic composition.

These results suggest several directions for future work. In terms of further improving the temporal structure in the generated pieces, distributed state spaces appear very promising and inference methods that more explicitly take into account the sequential nature of the model (i.e. the FHMM-SVI) seem the most promising. Additionally, if music theoretic constraints could be put into the model, such that each hidden state chain modeled a specific aspect of harmony, for example, likely the consonance and style of the generated pieces would be much improved.

Although the training sequences of notes were in fact discrete observations, using a continuous approximation and binning as a post-processing step did not appear to significantly limit the quality of the generated pieces. Inference tends to be easier and more efficient in models with continuous observations and it appears that such models are able to adequately model the discrete note pitches. Related to this, exploring different emission distributions in the same model framework is a logical next step. It may be that the various hidden state chains are able to capture harmonic aspects of the training pieces, but that information is lost or distorted when being combined in the emission parameters. Other emission distributions, for example, neural networks, may help avoid this problem and could be more powerful in their representation, while still allowing for a specific specification of the state space. Finally, considering more complex and more efficient MCMC techniques that more explicitly take into account the sequential structure of the model and perhaps could be encoded with music theoretic constraints, offer another promising direction.

References

[1] Anjan Nepal and Alexander Yates. Factorial Hidden Markov Models for Learning Representations of Natural Language. *CoRR*, abs/1312.6168, 2013.

[2] Anna K. Yanchenko and Sayan Mukherjee. Classical Music Composition Using State Space Models. *CoRR*, abs/1708.03822, 2017.

[3] Magenta. Magenta. https://magenta.tensorflow.org/, 2016.

[4] G. Nierhaus. *Algorithmic Composition: Paradigms of Automated Music Generation*. Springer-WienNewYork, 2009.

[5] Robert A. Jacobs and Wenxin Jiang and Martin A. Tanner. Factorial hidden markov models and the generalized Factorial Hidden Markov Models and the Generalized Backfitting Algorithm. *Neural Computation*, 14:2415–2437, 2002.

[6] Zoubin Ghahramani and Michael I. Jordan. Factorial Hidden Markov Model. *Machine Learning*, pages 1–31, 1997.