Modeling music using hidden Markov models

Daniel Wysocki and Craig Graci

May 13, 2015

Abstract

We model songs as a sequence of notes emitted by a hidden Markov model. A model is trained on a given song, and can be used to randomly generate new, similar songs, as well as study the structure of the composition. We produce a "signature" for each song, defined by two feature vectors, based on the emission ($\mathbf{e}$) and transition ($\mathbf{t}$) probabilities.

1 Introduction

Algorithmic music composition is a subject of much interest in the artificial intelligence research community. Many approaches have been taken to produce acceptable music, as the task is not straightforward. The classic work, Experiments in Music Intelligence by Cope (1996), explores a range of approaches, from knowledge-based systems, to machine learning approaches. A knowledge-based system works by following a set of rules defined by the programmer, and is very difficult to get right, due to its rigidity. Machine learning, on the other hand, creates a model from existing compositions, which are used as generators for new compositions. The latter approach is also employed by Marchini and Purwins (2011). In order to produce satisfying music, it is often necessary to combine the two approaches in some manner.

The current work employs a pure machine learning approach, training a hidden Markov model (Rabiner 1989) on a song (Section 3.1). This model is used as a generator (Section 3.3), as well as a means of observing the structure of the training songs (Section 3.4).

	Title
1	Auld Lang Syne
2	Barbara Allen
3	Frere Jacques
4	Happy Birthday
5	I'm a Little Teapot
6	Mary Had a Little Lamb
7	Scarborough Fair
8	This Old Man
9	Three Blind Mice
10	Twinkle Twinkle Little Star

Table 1: Table of melodies used, see A for transcriptions.

2 Data

2.1 Format

Training data were provided to the model using the JFugue (Koelle 2002–2014) MusicString format. The format is ideal, as it allows a song to be represented as a simple linear sequence of notes. JFugue also has the ability to convert MusicStrings into MIDI, for easy playback.

2.2 Selected Compositions

The training songs used were limited to simple melodies, listed in Table 1. The selected songs were taken from Durey and Clements (2001), who used hidden Markov models to identify melodies. While more complex songs may be used in the future, such simple melodies may be more accurately modeled by low-dimensional hidden Markov models.

3 Modeling

3.1 Hidden Markov Model

A hidden Markov model is used to describe each song. A hidden Markov model can be described by the following (Rabiner 1989):

1) The set of hidden states, denoted $S = \{s_1, s_2, \ldots, s_N\}$. At time t, the current state is denoted q_t. These states represent states of the song, which are determined automatically, and are described in more detail in Section 3.4.

2) The set of possible observation symbols, denoted $V = \{v_1, v_2, \ldots, v_M\}$, which are emitted by each state with a certain probability distribution. These observations are the individual notes of the song. There is also a special "end of song" note appended to the song, which denotes the termination of the song.

3) The state transition probability distribution $A = \{a_{ij}\}$, where

$$a_{ij} = P[q_{t+1} = s_j | q_t = s_i], \quad 1 \leq i, j \leq N. \tag{1}$$

This defines the probability that the song will make a transition between each pair of states.

4) The observation probability distribution in state j, $B = \{b_j(k)\}$, where

$$b_j(k) = P[v_k \text{ at } t | q_t = s_j] \quad \begin{array}{l} 1 \leq j \leq N \\ 1 \leq k \leq M. \end{array} \tag{2}$$

This defines the probability of a particular note being played while the song is in a particular state.

5) The initial state distribution $\pi = \{\pi_i\}$, where

$$\pi_i = P[q_1 = s_i], \quad 1 \leq i \leq N. \tag{3}$$

This defines the probability of the song beginning in any given state.

A model is denoted concisely as $\lambda = (A, B, \pi)$, where N and M are contained within A and B. Once a model is created – either with random or uniform probabilities – it is improved to better describe a sequence of observations, $O = (o_1, o_2, o_3, \ldots, o_T)$. This is done using the Baum–Welch algorithm (Baum and Petrie 1966), which iteratively improves the model in order to find a local maxima for the likelihood $P[O|\lambda]$. Likelihood is determined using the forward algorithm. These algorithms were implemented according to Mann (2006) and Ibe (2013).

3.2 Model Selection

When the system being modeled has well defined states, N can be chosen such that each state is described uniquely by the model. However, in the case of a song, it is not entirely clear what a state corresponds to. Therefore, it was necessary to implement a simple algorithm for selecting N.

A minimum and maximum number of states, $N_{\min}$ and $N_{\max}$ are provided manually, in order to restrict the search space. Then the search space is divided into a number of bins, b. A model is trained at $N_{\max}$, and b (or fewer) equally spaced integer values of N_i, such that $N_{\min} \leq N < N_{\max}$. The likelihood, $P[O|\lambda]$, is computed for each model, and the model which maximizes that likelihood is taken. If multiple models maximize the likelihood, the one with the lowest N is taken. The procedure then repeats with $N_{\min} = N_{i-1}$, and $N_{\max} = N_i$, until the search space cannot be further subdivided.

3.3 Algorithmic Composition

Once a model has been obtained for a song, it can be used as a generator for new songs. The procedure for song generation is as follows:

1. With probability π_i, let the initial state $q_1 = s_i$.
2. When at time t, the current state is $q_t = s_j$, emit note v_k with probability $b_j(k)$. If that note is the "end of song" note, terminate, otherwise let $q_{t+1} = s_i$ with probability a_{ji} and repeat this step.

Once the procedure terminates, the notes are converted to a MIDI file using JFugue. Ten compositions have been made for each of the songs of interest, and can be listened to at
https://dwysocki.github.io/csc466/simple_compositions/

3.4 Signatures

A "signature" is computed for each song, based on its model. It is a 2-tuple $\mathcal{S} = (\mathbf{e}, \mathbf{t})$, where $\mathbf{e}$ is the emission feature vector, and $\mathbf{t}$ is the transition feature vector.

$\mathbf{e}$ is defined as

$$\mathbf{e} = \langle e_1, e_2, e_3, \ldots \rangle, \tag{4}$$

where e_i is the number of states in the model with i notes likely to be emitted. Here, "likely" means that the probability of a note being emitted is more than one standard deviation above the mean for that state. If e_1 is the dominant term, that means that the majority of states correspond to precisely one

Title	N	t_1	t_2	t_3	t_4	t_5	t_6	e_1	e_2
Auld Lang Syne	55	32	18	5	0	0	0	55	0
Barbara Allen	55	31	23	1	0	0	0	54	1
Frere Jacques	60	33	17	8	1	1	0	60	0
Happy Birthday	54	34	17	3	0	0	0	54	0
I'm a Little Teapot	43	32	9	2	0	0	0	43	0
Mary Had a Little Lamb	47	25	11	11	0	0	0	47	0
Scarborough Fair	53	44	7	2	0	0	0	53	0
This Old Man	49	24	17	7	1	0	0	49	0
Three Blind Mice	60	25	12	15	5	2	1	60	0
Twinkle Twinkle Little Star	33	15	11	6	2	0	0	34	0

Table 2: Signatures for the selected songs.

note. Interestingly, $e_2 = e_3 = \ldots = 0$ for every song except Barbara Allen, which has $e_2 = 1$.

$\mathbf{t}$ is defined as

$$\mathbf{t} = \langle t_1, t_2, t_3, \ldots \rangle, \tag{5}$$

where t_i is the number of states in the model with i states likely to be transitioned to. When a state only transitions to one state, that state is part of a "linear progression", and if it transitions to $n > 1$ states, it is an "n-way branch". Linear progressions comprised the majority of states for all songs tested, with varying distributions of n-way branches. Table 2 lists the signatures of the songs of interest.

4 Conclusion

This work has demonstrated that hidden Markov models are capable of capturing some of the essence of music, and serve as effective generators. However, to create truly satisfactory music, one will have to extend the method beyond a simple hidden Markov model. Extensions of the hidden Markov model, such as the left–right HMM (Rabiner 1989) and the generalized HMM (Kulp et al. 1996) may be able to perform better.

Further investigation into the song models should be performed. The feature vectors may be used for k-means clustering. Linear progressions may also be extracted, identifying the defining parts of songs.

References

Baum, Leonard E. and Ted Petrie (1966). "Statistical Inference for Probabilistic Functions of Finite State Markov Chains". In: *The Annals of Mathematical Statistics* 37.6, pp. 1554–1563.

Cope, David (1996). *Experiments in Musical Intelligence*. Vol. 12. Computer Music and Digital Audio Series. A-R Editions.

Durey, A. S. and M. A. Clements (2001). "Melody spotting using hidden Markov models". In: URL: http://ismir2001.ismir.net/pdf/durey.pdf (visited on 05/12/2015).

Ibe, Oliver (2013). *Markov Processes for Stochastic Modeling*. 2nd ed. Elsevier.

Koelle, David (2002–2014). *JFugue*. http://www.jfugue.org/4/.

Kulp, D. et al. (1996). "A generalized hidden Markov model for the recognition of human genes in DNA". In: *Fourth International Conference on Intelligent Systems in Molecular Biology*. Menlo Park: AAAI Press, pp. 134–141.

Mann, Tobias P. (2006). "Numerically stable hidden Markov model implementation". In: URL: http://bozeman.genome.washington.edu/compbio/mbt599_2006/hmm_scaling_revised.pdf.

Marchini, Marco and Hendrik Purwins (2011). "Unsupervised Analysis and Generation of Audio Percussion Sequences". In: *Exploring Music Contents*. Ed. by Slvi Ystad et al. Vol. 6684. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 205–218.

Rabiner, Lawrence (1989). "A tutorial on hidden Markov models and selected applications in speech recognition". In: *Proceedings of the IEEE* 77.2, pp. 257–286.

A Song Transcriptions

The JFugue MusicStrings used as training data are provided here verbatim.

Auld Lang Syne

C4 F4 E4 F4 A4 G4 F4 G4 A4 G4 F4 F4 A4 C5 D5
D5 C5 A4 A4 F4 G4 F4 G4 A4 G4 F4 D4 D4 C4 F4

Barbara Allen

C4 E4 F4 G4 F4 E4 D4 C4 D4 E4 G4 C5 C5 B4 G4

C5 C5 A4 G4 F4 A4 G4 E4 D4 C4 D4 E4 F4 G4 F4 E4 D4

Frere Jacques

G4 A4 B4 G4 G4 A4 B4 G4 B4 C5 D5 B4 C5 D5
D5 E5 D5 C5 B4 G4 D5 E5 D5 C5 B4 G4
G4 D4 G4 G4 D4 G4

Happy Birthday

G4 G4 A4 G4 C5 B4
G4 G4 A4 G4 D5 C5
G4 G4 G5 E5 C5 B4 A4
F5 F5 E5 C5 D5 C5

I'm a Little Teapot

C4 D4 E4 F4 G4 C5 A4 C5 G4
F4 F4 G4 E4 E4 D4 D4 E4 C4
C4 D4 E4 F4 G4 C5 A4 C5 G4
C5 C4 D4 E4 F4 E4 D4 C4

Mary Had a Little Lamb

B4 A4 G4 A4 B4 B4 B4 A4 A4 A4 B4 D5 D5
B4 A4 G4 A4 B4 B4 B4 B4 A4 A4 B4 A4 G4

Scarborough Fair

D4 D4 A4 A4 A4 E4 F4 E4 D4
A4 C5 D5 C5 A4 B4 G4 A4
D5 D5 D5 C5 A4 A4 G4 F4 E4 C4
D4 A4 G4 F4 E4 D4 C4 D4

This Old Man

D5 B4 D5 D5 B4 D5 E5 D5 C5 B4 A4 B4 C5
B4 C5 D5 G4 G4 G4 G4 G4 A4 B4 C5 D5
D5 A4 A4 C5 B4 A4 G4

Three Blind Mice

E4 D4 C4 E4 D4 C4 G4 F4 F4 E4 G4 F4 F4 E4
G4 C5 C5 B4 A4 B4 C5 G4 G4
G4 C5 C5 C5 B4 A4 B4 C5 G4 G4
G4 G4 G5 G5 B4 A4 B4 C5 G4 G4 G4 F4 E4 D4 C4

Twinkle Twinkle Little Star

G4 G4 D5 D5 E5 E5 D5 C5 C5 B4 B4 A4 A4 G4
D5 D5 C5 C5 B4 B4 A4 D5 D5 C5 C5 B4 B4 A4
G4 G4 D5 D5 E5 E5 D5 C5 C5 B4 B4 A4 A4 G4