Making Music with Hidden Markov Models

Anna Yanchenko
Duke University
anna.yanchenko@duke.edu

Megan Robertson
Duke University
megan.robertson@duke.edu

Abstract

Hidden Markov Models (HMMs) are widely used models for sequential data with many varying applications. In this project, we use HMMs to capture the latent elements involved in music creation. The observed states are the pitch and velocity (volume) of each note in an existing song, while the latent states include elements such as chord progression, harmonics, dynamics and melody. In this paper, we implement three different Hidden Markov Models and compare the resulting compositions for each model. These models are trained on various classical piano pieces, and the resulting compositions are compared.

1 Introduction

Hidden Markov Models are known for their use in language applications such as speech recognition and handwriting recognition programs. Inspired by a recent homework assignment to use Hidden Markov Models to generate text, we wanted to explore how successful these same models would be in generating music. The Thus, this project served as a platform to combine our shared interests of music and statistics.

The project used music data in the form of MIDI (Musical Instrument Digital Interface) files. This interface creates a form of communication between computers and musical instruments that enables them to send instructions back and forth. MIDI files can contain multiple channels that store information about the notes and velocities of different instruments. Each note's pitch is coded as an integer between 0 and 127, where 60 refers to middle C. The velocities are also coded as integers from 0 to 127. The files additionally contain information about when a particular note starts and stops. [1].

Much thought and musical theory goes into composing any piece. The notes much progress in a logical, harmonic way that is pleasing to the listener and thus the composer must pay close attention to the circle of fifths to determine which notes sound musically pleasing together. Dissonance can be used to elicit specific moods in the piece but does not occur by chance. In addition, the composer must use melody, harmonies and dynamics to create a piece of music which tells a story or conjures a specific emotion in the listener. [2]

However, the MIDI data of a note's pitch and velocity in sequential order does not capture all of the theory and ideas that create a composition. Therefore, we use Hidden Markov Models (HMMs) in order to model the latent variables, specifically the musical theory that goes into composing a piece. We utilize three different HMMs and train them on classical piano pieces. The estimated parameters are then used to generate new music based on the modeling of the original piece. Our goal was to compare how successful each of the three

[1] MIDI Manufacturers Association, "An Introduction to MIDI"
[2] Music Composition for Dummies, http://www.dummies.com/how-to/content/music-composition-for-dummies-cheat-sheet.html

HMMs was in generating a musically pleasing piece and how similar our generated music was to the original composition.

2 Methods

Classical piano pieces were used to train each of the models. In this paper we present the result for Gustav Holt's Jupiter from "The Planets" (arranged for piano) and Johann Pachelbel's Canon in D. The MIDI files for these songs were converted to Comma Separated Value (CSV) files.[3]

The classical pieces were used to estimate the respective parameters of the models described below. In this training, the tempo, key signature and length of the original song were not changed. The only possible observed states were the velocities and notes present in the original pieces. We treated the pitch of each note and its velocity as independent variables. Once the parameters had been estimated, they were used to generate a new piece of music. This process generates a CSV file of the new composition and this file was then converted back to a MIDI file using the same program as above. The files could then be played using a synthesizer or a program such as GarageBand.

The three models that we considered were a first order HMM, a second order HMM and a first order HMM with two hidden states. We used the Baum-Welch algorithm, in tandem with the forward and backward algorithms, to estimate the parameters in our model. In all three models used the following parameters: the initial distribution, $\pi_i = \mathbb{P}(Z_1 = i)$, the transition matrix, $T_{ij} = \mathbb{P}(Z_{t+1} = j|Z_t = i)$ and the emission distribution, $\phi_i(X_t) = p(X_t|Z_t = i)$. For each of the three models, we assumed that Z could take m possible discrete states and that X could take k possible discrete states, where $1, \cdots, k$ corresponded to the note pitches or velocities occurring in the original piece. For example, if the original piece had 10 different note pitches and 5 different velocities, $k = 10$ for the HMM run on the observed notes (with each state corresponding to a specific one of those 10 unique pitches) and $k = 5$ for the HMM run on the observed velocities. The Baum-Welch algorithm was adjusted from the first order form to take into the account the structure of the parameters in the second order and two hidden state models. A new song was generated for each of the three models using both Jupiter and Pachelbel's Canon as the original song (the observed states). In order to evaluate the performance of each model, twenty-two survey respondents determined which of the compositions generated using the three different HMMs they liked best, for either Pachelbel's Cannon or Holst's Jupiter. They also were provided with the original versions for both Jupiter and Pachelbel's Cannon,and were asked which of the original versions they thought inspired the remixes.

2.1 Model 1: First Order Hidden Markov Model

The simplest of the three models was the first order Hidden Markov Model. In this model, the observed values, X_i, were the notes' pitch or their velocity (modeled independently). The model had only one hidden state, Z_i, for each pitch or velocity and each hidden state only depended on the prior hidden state (Figure 1).

[3]This was done using a program found at http://www.fourmilab.ch/webtools/midicsv/

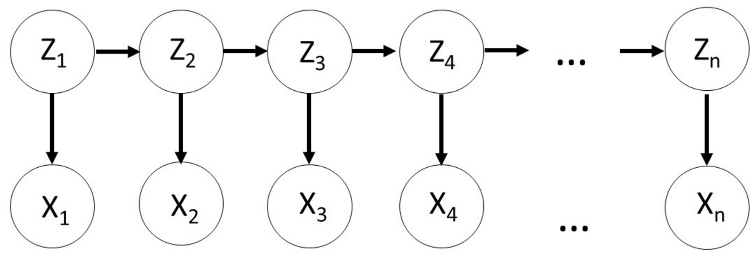


Figure 1: Graphical Model - First Order HMM

The Baum-Welch Algorithm as derived in class was used to estimate the parameters of this model. For details of the derivations and algorithm of this first order HMM see class notes.[4]

2.2 Model 2: Second Order Hidden Markov Model

The second order HMM assumed that each hidden state depended on the two previous states (Figure 2). This enabled the model to capture more of the musical structure inherent in the piece, since this structure was evolving over time and depended on more than just the previous note. The Baum-Welch Algorithm was again used to estimate the parameters, this time modified to accommodate the addition of a transition matrix, $T_{ijk} = \mathbb{P}[Z_t = k | Z_{t-1} = j, Z_{t-2} = i]$ which modeled the dependence of the current hidden state on the two previous hidden states. The details of the algorithm for this model can be found in Mari and Schott's "Probabilistic and Statsitical Methods in Computer Science"[5], with a summary of the main results in the Appendix (subsection 5.1).

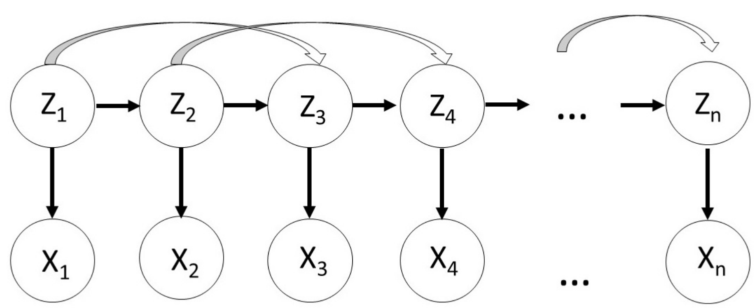


Figure 2: Graphical Model - Second Order HMM

2.3 Model 3: HMM with Two Hidden States

The final model expanded on the initial model by adding a second hidden state (Figure 3). This meta-state allowed the model to capture additional aspects of the hidden states and the music generation process. This processes might evolve at a rate different than the processes that are present in the observed states and one latent state. This HMM involved a specific form of the transition matrix that allowed for a more efficient implementation of the derived Baum-Welch algorithm. In particular, we assumed that the transition matrix factored as

$$T_{ik,jl} = A_{ij} B_{jkl} = \mathbb{P}(R_t = j, S_t = l | R_{t-1} = i, S_{t-1} = k),$$

[4]Statistics 531, Duke University Spring 2016. Instructor: Jeff Miller

[5]Jean-Francoix Mari and Rene Schott, "Probabilistic and Statistical Methods in Computer Science", Springer Science, pg. 161-167 and Brett Watson and Ah Chung Tsoi, "Second Order Hidden Markov Models for Speech Recognition", Univsersity of Queensland

where $A_{ij} = \mathbb{P}(R_t = j | R_{t-1} = i)$ and $B_{jkl} = \mathbb{P}(S_t = l | R_t = j, S_{t-1} = k)$. A full derivation for the Baum-Welch algorithm for this model can also be found in the Appendix (subsection 5.2).

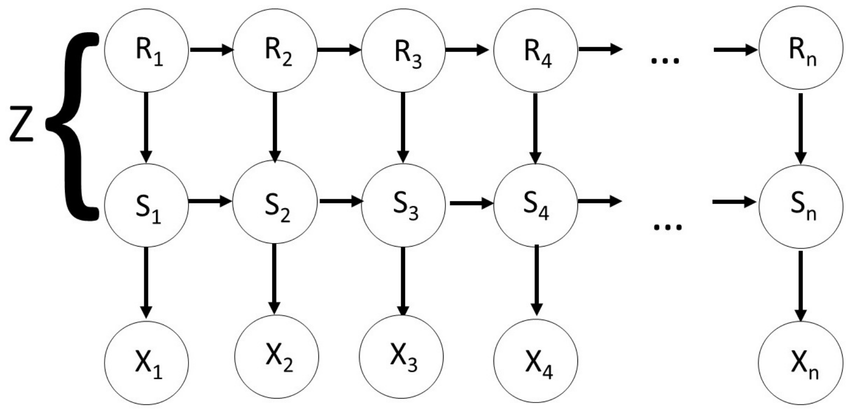


Figure 3: Graphical Model - First Order HMM with Two Hidden States

3 Results

3.1 Original Pieces

The three models were originally trained on four different classical piano pieces (or orchestral pieces arranged for piano). These four pieces included Claude Debussy's Clair de Lune and Antonin Dvorak's Largo from Symphony No. 9 ("The New World"), in addition to Jupiter and Pachelbel's Canon. Of the four pieces, Jupiter and Pachelbel's Cannon provided the best results, for all of the HMMs considered. It is possible that these pieces performed the best because of their prominent melody from the beginning. For example, in Clair de Lune, the piece builds for a while before the melody emerges, while the version of Jupiter that we used to train the piece featured just the main melody and so did not have to build over a long period of time. Our models may not have worked well for this because they did not have the capability to capture the building structure and melody of the song over a long stretch of notes.

Both Jupiter and Pachelbel's Cannon are well-known classical pieces whose themes occur frequently in advertisements and other media forms. The two pieces have a logical progression. It is obvious that these pieces are structurally sound and musically cohesive given their popularity.

A sample of the sheet music for the first 13 measures of Jupiter in it's original form is included below (Figure 4). The top row of each line is played by the right hand of the piano player and is written in Treble Clef (Figure 5), while the bottom row of each line is played by the left hand and is written in Bass Clef (lower than Treble Clef)[6]. The two lines are played at the same time, thus there are 13 measures included in this original sheet music. A sample of the sheet music for Pachelbel's Canon is included in the Appendix (Figure 4).

[6] http://www.guitarsite.de/p1042.htm

Figure 4: Jupiter - Original Song

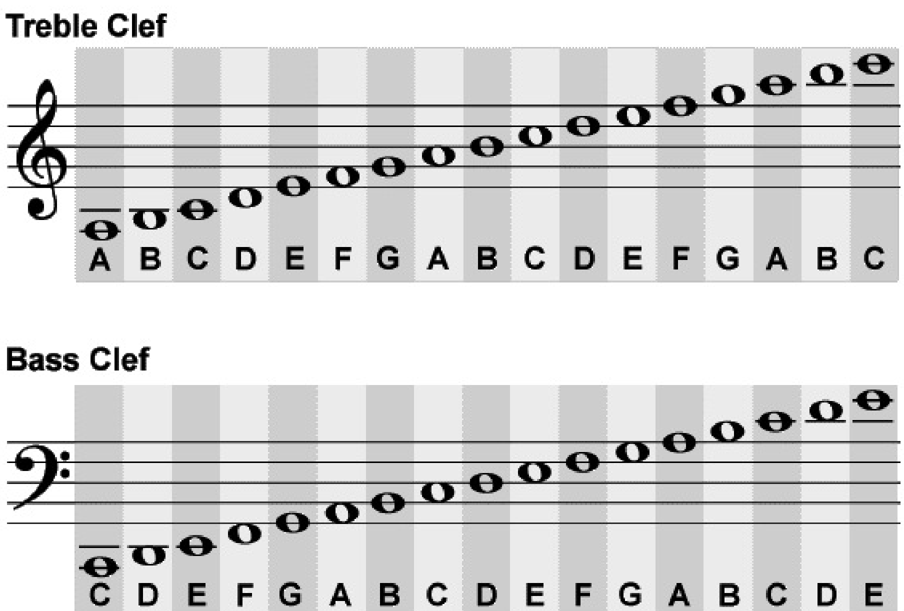


Figure 5: Musical Clefs: Treble Clef and Bass Clef

3.2 First Order Hidden Markov Model

The Jupiter remix created using the first order Hidden Markov Model had lots of octave and large interval jumps that were not present in the original song. For example, in measures one, three and four in the sheet music for the piece generated by this model (Figure 6), pitches jump over an octave or more from one note to the next. In addition, there are more rests in the first two lines of the remix and there are also more quarter and half notes than occurred in the original piece. The melody of the song seems to be distributed about equally between the left and the right hands, while in the original composition the right hand had the entire melody, while the left hand played the accompaniment. There is also no musical resolution at the end of the remix.

Figure 6: Jupiter Remix - Model 1

The remixed version of Pachelbel's Cannon had less sixteenth notes but more dotted notes than the original piece. Pachelbel's Cannon consists of mostly eighth or sixteenth notes for the melody. There are also more rests in this remixed version than there are in the original. Finally, there are large octave jumps and jumps of large intervals in this version (Figure 15).

The first order HMM compositions sounded very different from the original compositions. This probably occurs because the model does not capture enough of the overall structure or the original piece. Thus there are some notes that occur together that musically should not, as there is no dissonance in the originals. For example, the melody for Pachelbel's cannon builds over several measures, yet our first order HMM was not able to capture enough of this building structure to generate an analogous progressing melody.

3.3 Second Order Hidden Markov Model

The Jupiter composition created using the second order Hidden Markov Model is an improvement on the first remix. It still has some large interval jumps in terms of pitch, but fewer jumps than that from Model 1 (Figure 7). The chords also make more sense in the piece and there is less dissonance. The piece also has fewer rests than the first order composition, but more rests than in the original composition.

Figure 7: Jupiter Remix - Model 2

The Cannon composition created using the second order Hidden Markov Model has more sixteenth notes than the first remix, but fewer sixteenth notes than the original version (Figure 16). There are still awkward interval jumps present in the piece. Overall, the chords made more sense and there was less dissonant in this piece than the first remix. The progression of the song seemed to be more logical than the first model. This probably occurred because the second order HMM captures more structure than the first order HMM.

3.4 First Order Hidden Markov Model with Two Hidden States

Of the three remixes, this version had the fewest rests of the remixed compositions, but more than the original version (Figure 8). It also contained large interval jumps. These jumps between notes were larger than those that occurred in the original song, but were smaller than the jumps in the other versions.

Figure 8: Jupiter Remix - Model 3

Likewise, for Pachelbel's Canon, the two latent state version had more eighth and sixteenth notes than the other two remix versions (Figure 17). However, it still did not have as many as the original version and there were still large jumps in between notes.

The following figures (Figure 9,Figure 10, Figure 11, Figure 12) display the velocities for successive notes of the original compositions and the remixes. The same figures presented in the Results section and the below can be found in the Appendix for the Pachelbel's Cannon application (Figure 18).

For the original Jupiter piece, the velocity appears to be approximately randomly scattered about 80 with a few outliers but no apparent structure. For Model 1 and Model 2, however, there appears to be significant structure in the velocity over time as there appear to be straight lines of constant velocity that are not present in the original piece. Finally, Model 3 (2 hidden states) appears to best replicate the velocity found in the original piece, as the velocities again appear randomly scattered, though perhaps with slightly more structure than originally.

For Pachelbel's Canon, the velocities increase to about the middle of the piece and then decrease again in a parabola-like shape. None of the three models are able to recreate this pattern in the velocity and they all have significant, straight structure and much less variation in velocity than the original piece.

3.5 Summary

Overall, the remix created using the second order Hidden Markov Model appeared to have the best musical structure of all of the remixes. On the other hand, the remix created using the first order Hidden Markov Model with two latent states had better musical chords. In all of the new compositions, the melody seemed to be split between the two hands. The models did not catch the general structure that lower notes, those played by the left hand, were longer in the original versions of the pieces and tended to be accompaniment, while most of the melody was in the notes played by the right hand. In all of the models, the dynamics were modeled independently and as a result the dynamics did always occur in a logical fashion. There were many instances where certain notes are uncomfortably loud compared to the rest of the piece and did not logically fit in with the rest of the generated piece.

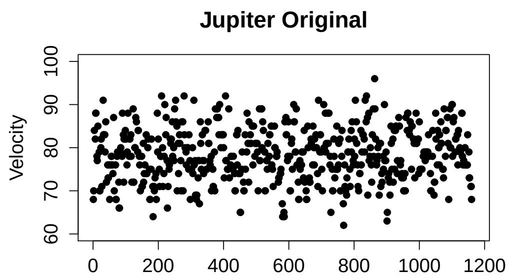


Figure 9: Initial condition

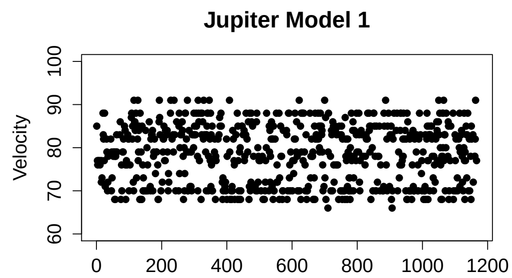


Figure 10: First Order HMM

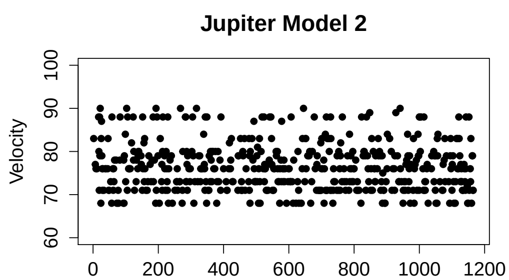


Figure 11: Second Order HMM

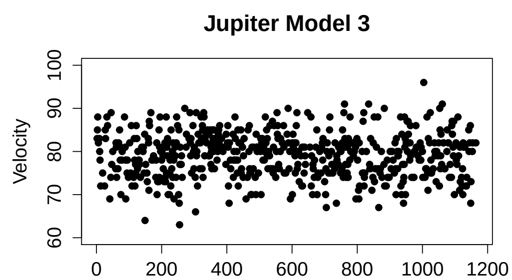


Figure 12: Two Hidden States

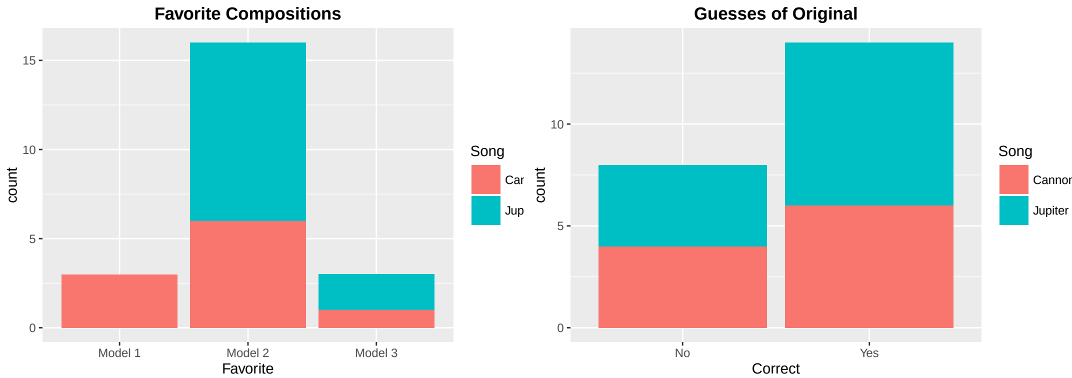
Twenty-two individuals were sent the three versions of one of the two original songs and were asked which version they preferred (Figure 13a) and which original song, Jupiter or Pachelbel's Canon, they thought generated the new versions (Figure 13b). Of the listeners asked about the compositions inspired by Pachabel's Cannon, three preferred Model 1, six preferred Model 2 and only one preferred Model 3. Six out of ten correctly identified the original composition. On the other hand, eight of twelve respondents correctly chose Jupiter as the original composition for the other remixes. Of those that listened to Jupiter, none of the respondents preferred Model 1, ten preferred Model 2 and two preferred Model 3. For both groups, Model 2, the second order HMM, was by far the overall preferred composition.

Full MP3 versions of all generated songs, as well as the original versions of Jupiter and Pachelbel's Canon are included with the paper. As above, in the naming of the MP3s, Model 1 refers to the first order HMM, Model 2 to the second order HMM and Model 3 to the two hidden state HMM.

4 Conclusion and Further Research

Of the three models used to create the remixed compositions, none of the models produced pieces that sounded like coherent music that was trying to tell a story or evoke an emotion. This is not surprising given the simple structures of our models. There are certain points in some of the pieces where individual measures sound good, but the overall piece is still not as cohesive as we would have liked. The second order HMM seemed to perform the best in creating music.

One extension of this project we would like to explore are higher order Hidden Markov Models. Models of higher orders would allow for the current note pitches and velocities to depend on more previous notes. This would introduce more structure and dependence into



(a) Preferred Compositions

(b) Inspiration for the Remixes

Figure 13: Survey Results

the models that could create the cohesion in the pieces that was missing before. The second order HMM seemed to perform the best, so it would be exciting to see how a third, fourth or fifth order HMM could improve the generated compositions.

In addition, we need to include the dependencies between the dynamics and the notes in the model. This could improve the disconcerting loud notes in the pieces. Modeling such dependency could also extend to a piece with multiple instruments. This would require the dependencies between instruments to be modeled as well.

The resulting compositions from these models showed promising results, but there significant room for improvement. The improvements and extensions described above could elevate our results to more cohesive and complicated compositions that have more of a resemblance to actual music composition done by humans.

5 Appendix

5.1 Baum-Welch Algorithm for Second Order HMM

1. Run first order Baum-Welch to estimate π, T, ϕ.
2. Use second order Baum-Welch to estimate T_{ijk}:

Forward Algorithm:

$$\alpha_1(i) = \pi_i \phi_i(x_1)$$
$$\alpha_2(i,j) = \alpha_1(i) T_{ij} \phi_j(x_2)$$
$$\vdots$$
$$\alpha_{t+1}(j,l) = \sum_{i=1}^{M} \alpha_t(i,j) T_{ijl} \phi_l(x_{t+1})$$

where 2 $\leq$ t $\leq$ T-1 and 1 $\leq$j,k $\leq$ M.
Thus, $P(X|\lambda) = \sum_{i=1}^{M} \alpha_T(i, M)$.

Backward Algorithm:

$$\beta_t(i,j) = \sum_{i=1}^{M} \beta_{t+1}(j,k) T_{ijk} \phi_k(x_{t+1})$$

where 2 $\leq$ t $\leq$ T-1 and 1 $\leq$ j,k $\leq$ M.

Baum-Welch Estimate for T_{ijk}:

$$\beta_t(i,j,k) = \frac{\alpha_t(i,j) T_{ijk} \phi_k(x_{t+1}) \beta_{t+1}(j,k)}{P(x|\lambda)}$$

where 2 $\leq$ t $\leq$ T-1.

$$\bar{T_{ijk}} = \frac{\sum_t \beta_t(i,j,k)}{\sum_{k,t} \beta_t(i,j,k)}$$

Iterate until convergence.

5.2 Baum-Welch for First Order HMM with Two Hidden States

Define the following:

$$A_{ij} = \mathbf{P}(R_t = j | R_{t-1} = i)$$
$$B_{jkl} = \mathbf{P}(S_t = l | R_t = j, S_{t-1} = k)$$
$$T_{ik,jl} = A_{ij} B_{jkl} = \mathbf{P}(R_j = j, S_t = l | R_{t-1} = i, S_{t-1} = k)$$
$$Z_t = (R_t, S_t)$$

The constraints are $\sum_j A_{ij} = 1, \sum_l B_{jkl} = 1$.

Define c to be a constant. Thus,

$$Q(\theta, \theta_k) = c + \sum_{t=2}^{n} \sum_{i,k} \sum_{j,l} \mathbf{P}_{\theta_k}(R_{t-1} = i, S_{t-1} = k, R_t = j, S_t = l | x) \log T_{ik,jl}$$

$$B_{tikjl} = \mathbf{P}_{\theta_k}(R_{t-1} = i, S_{t-1} = k, R_t = j, S_t = l|x)$$

$$\log T_{ik,jl} = \log A_{ij} + \log B_{jkl}$$

Thus,

$$0 = \frac{\partial}{\partial A_{ij}}(Q(\theta, \theta_k) - \lambda \sum_j A_{ij}) = (\sum_{t=2}^{n} \sum_k \sum_l B_{tiklj} \frac{1}{A_{ij}}) - \lambda$$

$$\lambda A_{ij} = \sum_{t=2}^{n} \sum_k \sum_l B_{tikjl}$$

$$\lambda = \sum_j \sum_{t=2}^{n} \sum_k \sum_l B_{tiklj}$$

For each i, $A_{ij} \propto \sum_{t=2}^{n} \sum_{k,l} B_{tikjl}$

$$0 = \frac{\partial}{\partial B_{jkl}}(Q(\theta, \theta_k) - \lambda \sum_l B_{jkl}) = \sum_{t=2}^{n} \sum_i B_{tiklj} \frac{1}{B_{jkl}} - \lambda$$

$$\lambda B_{jkl} = \sum_{t=2}^{n} \sum_i B_{tikjl}$$

For each j, k,

$$B_{jkl} \propto \sum_{t=2}^{n} \sum_i B_{tikjl}$$

5.3 Pachabel's Cannon Resuts

Figure 14: Pachelbel Original

Figure 15: Pachelbel Remix - Model 1

Figure 16: Pachelbel Remix - Model 2

Figure 17: Pachelbel Remix - Model 3

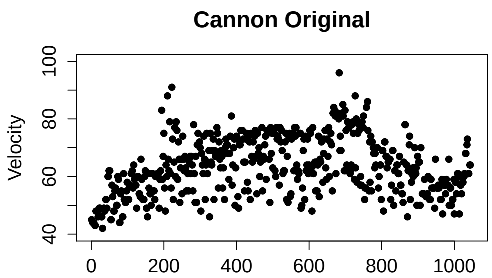


Figure 18: Initial condition

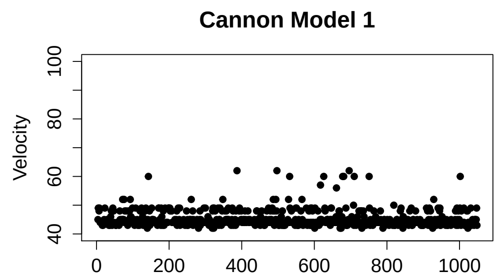


Figure 19: First Order HMM

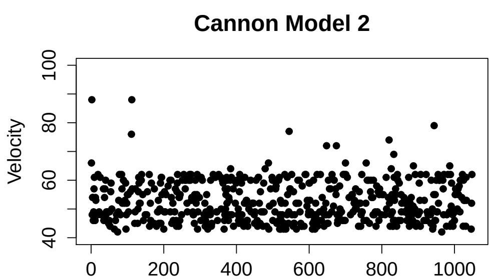


Figure 20: Second Order HMM

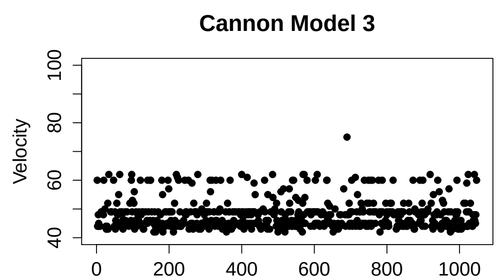


Figure 21: Two Hidden States

6 References

1. Wattson, Brett and Ah Chung Tsoi. "Second Order Hidden Markov Models for Speech Recognition", University of Queensland, 146-151.
2. MIDI Manufacturers Association, "An Introduction to Midi".
3. Music Composition for Dummies, http://www.dummies.com/how-to/content/music-composition-for-dummies-cheat-sheet.html