# MELODY INFORMED MUSICAL CHORD GENERATION USING HMM

Sheng Xu, Albert Peyton, Ryan Bhular
*Department of Electrical and Computer Engineering, University of Rochester, Rochester, New York, 14627*

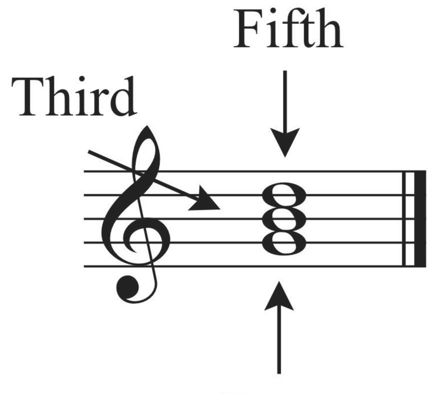
## Abstract

Our project introduces a melody informed chord generation model using hidden markov models (HMM). We claim that by using pitch information from a melody the HMM should be able to output an acceptable chord progression based both on the note played at the particular instance of the melody and the preceding chord. We assume that one of the three notes in a triad must be the current note in the melody . We trained 2 transition matrices of chordal states, based on a dataset of about 1,300 pop songs and another of 32 Beethoven sonatas (classical). With this information we were able to use the viterbi algorithm in order to calculate the most likely chord progression for our input melody. This chord progression is realized in discrete time according to each observation, and is written to MIDI so to be used in a DAW.

## Music Theory

Music can be broken down into 2 main parts: melody and harmony. Melody can be thought of as the subject of a song, and harmony can be thought of as giving context to this subject. Melody typically is known to be a single sequence of pitches, while harmony is usually a group of pitches played simultaneously called chords.

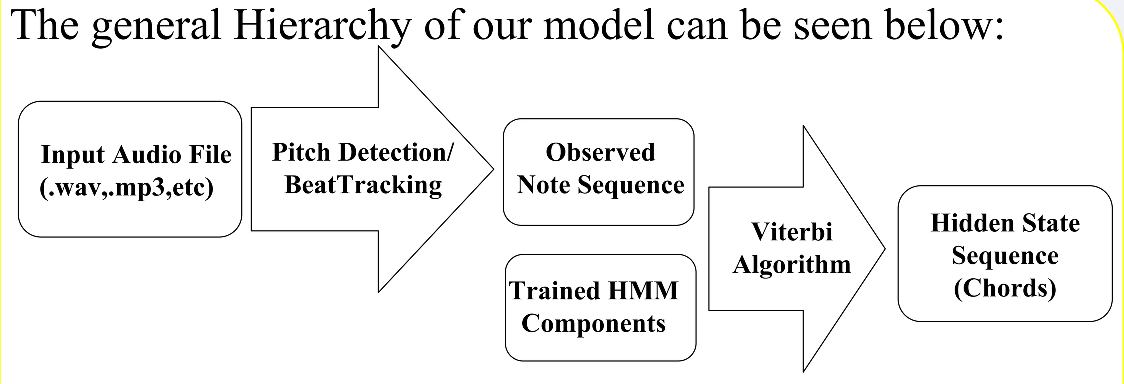
In basic music theory, chords can be thought of as a collection of 3 pitches called triads. With more advanced music, this is still typically true but with the potential of additional different pitches to give chords different colors or functions. A triad can be visualized below.




In our project we look only to recognize chords as triads, with the exception of the V7 dominant chord. With that in mind, if we treat each observed pitch in the melody as a chordal tone, there are only 3 chords that include this chordal tone: either as the root, the third or the fifth.

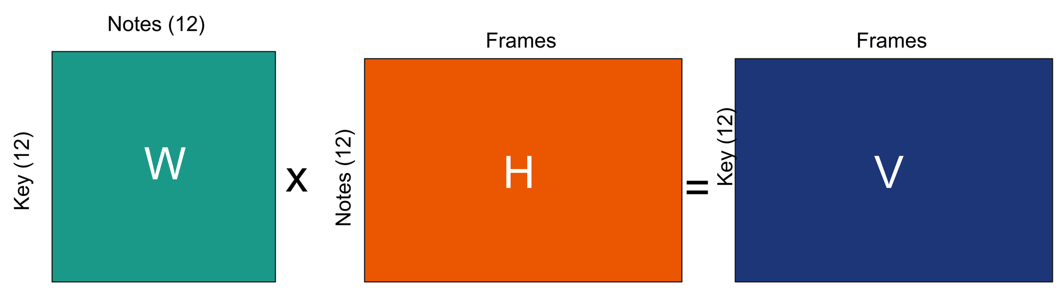
## System Model

The general Hierarchy of our model can be seen below:




## Pitch/Key Estimation

To determine the pitches (or notes) present in the audio we have used the YIN Algorithm [3].
Each musical key only has 7 available notes. With this prior knowledge a dictionary matrix can be derived with dimension keys x notes. Each row is then intuitively weighted based on what notes are available and how often the corresponding scale degree occurs in that particular key.
Multiplying both matrices defined, we can find a matrix containing the probability of each frame belonging to a key. By taking the sum of each row we can find the key with the highest probability.
We can determine if the key is major or it's relative minor based on the distribution of pitches.




## Beat detection

We use Dynamic Programming to detect the beats so that logical and evenly spaced chords can be generated. This algorithm uses a recursive approach to estimate the locations of onsets in the musical piece. The score D of each onset can be determined by:

$$D(n) = \begin{cases} \Delta(n) , & L = 1 \\ \Delta(n) + \lambda P_{\hat{\delta}}(n - b_{L-1}) + D(b_{L-1}) , & L > 1 \end{cases}$$

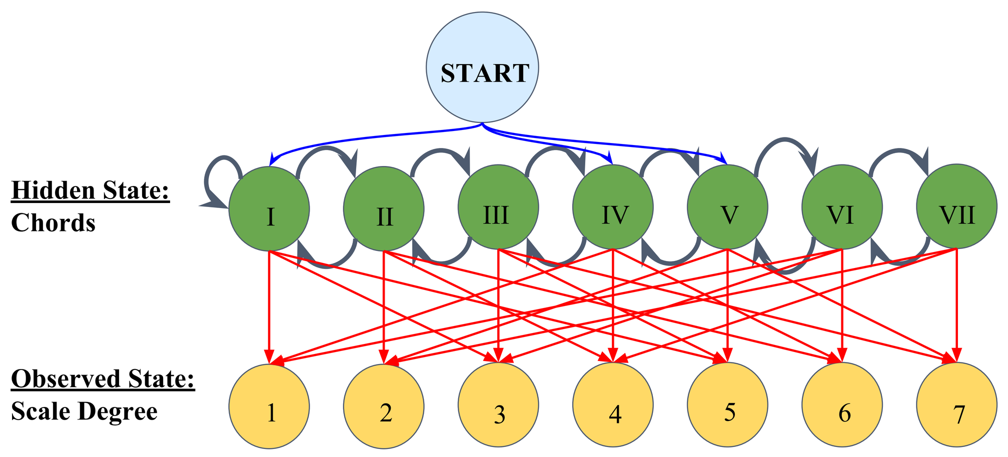
Where **$\lambda$** defines the tradeoff between onset strength and beat, **b** is the beat instance, **$\Delta$** is the observed onset strength, and **P** is a penalty function defined by:

$$P_{\hat{\delta}}(\delta) = -\log(\delta/\hat{\delta})$$

With $\hat{\delta}$ being the rough estimate of beat spacing. With the found beat times, we use the pitches present at these beats to be our observed melody sequence.

## Hidden Markov Model

**Data Representation:**
Once the key has been determined we can normalize all pitches to numeric scale degree values (1-7) based on the major tonic or the tonic of the relative minor key. By doing this we can also treat chords in a numeric fashion by only considering their relative chord number for the scale.
**Transition Matrix (A):**
We used the McGill Billboard Dataset and the Beethoven sonata Dataset in training our initial transition matrix [1][2]. By counting the transitions of all the songs and normalizing the matrix we can generate our transition space. This transition matrix shows the probability of one chord number to another. This creates a transition space from defined by the probability of one chord number transitioning to another.
**Emission Matrix (B):**
With a dimensionality of Observed States x 7 (for each scale degree), the emission matrix aims to map the possible chord choices to accompany the melody based on the note at that time instance. This has been done by using the broad assumption that the note in the melody must be present in the accompanying chord, with probabilities ½, ⅓, and ⅙ of the melody note being in the root, third, or fifth (respectively).
**Initial Probability ($\pi$):**
Like the emission matrix, we pick an initial probability where this first observation is a chordal tone. However, we weight chords I, IV, and V more than their counterparts as typically a musical phrase will begin with on one of these harmonies.




## Viterbi Algorithm

The Viterbi Algorithm can be used to determine the best possible path for the chord progression.
We Define a hidden Markov chain by $\theta = (A,B,\pi)$, where:

1. **A** is the initial transition matrix of chords
2. **B** is the initial emission matrix
3. **$\pi$** is the initial probabilities for all the states (chords).

The initial parameters A, B and **$\pi$** are estimated by using the general parameters estimated from the datasets. In addition, we define **Y** as our observation sequence (chord progression).

## Results

Our project was successful in generating chords in the form of midi notes. Below are a few transcriptions of our results.

classical major


pop major


classical minor


pop minor


As one can see, our program delivers in creating a simple harmonic structure around the melody that makes musical sense.

## Conclusion

In summary, our model worked adequately in determining a chord progression for a sequence of melody notes. Both the classical and pop transition matrices yielded decent results, but the results created with the classical matrix were undoubtedly better. We were successful in achieving our goal, but still recognize that our program could be expanded and improved.

## References

[1] John Ashley Burgoyne, Jonathan Wild, and Ichiro Fujinaga, 'An Expert Ground Truth Set for Audio Chord Recognition and Music Analysis', in Proceedings of the 12th International Society for Music Information Retrieval Conference, ed. Anssi Klapuri and Colby Leider (Miami, FL, 2011), pp. 633–38

[2] Chen, T. and Su, L. (2018). Functional Harmony Recognition of Symbolic Music Data With Multi-Task Recurrent Neural Networks. Proceedings of the 19th International Society for Music Information Retrieval Conference, Paris, France.

[3] de Cheveigné, A. and Kawahara, H. (2002). YIN, a fundamental frequency estimator for speech and music. The Journal of the Acoustical Society of America, 111(4), pp.1917-1930.