Algorithmically Harmonizing a Melody with Western Harmony

Overview

Songwriters, after creating a melody, face the next challenge of harmonizing it. Though this is a facile task for some, for a computer, it is challenging. A computer has no sense of what sounds “good” or “correct,” what different genres might use as the harmonic progression under a melody, or how different choices affect the mood of the song.

On the other hand, some songwriters fall into a rut, using the same harmonies that have been set out for at least the past 50 years of popular music. This is so much the case, the comedy band Axis of Awesome aggregated a sampling of them into the so called “4 chords song”, switching seamlessly from song to song simply by using the I V vi IV progression. A computer could potentially produce harmony that a human would not consider initially, bringing new life to old songs.

I intend to create a music generation system with minimal compositional constraints. Specifically, the system input will be a melody, and it will need to infer or construct the other components: chords, harmony type, and key. The proposed system will be based on abstract harmonic principles that will generate a range of harmonic progressions for a given melody.

Details

Harmonizing requires understanding what sounds “good” to the human ear, what western harmony dictates would be done, what rules can be broken and how frequently. I intend to explore a variety of paradigms for automated composition such as stochastic models and rule-based systems. The final system may comprise a combination of techniques.

Mathematical models of composition are often non-deterministic. The sequence of notes or chords are the result of a chance event, such as the role of dice or a random number generator. Composers in the 18th century, including Mozart and C.P.E. Bach created musical dice games in which the sequence of measures depended on rolls of the dice. For example, a waltz dice game attributed to Mozart had 16 separate sets of measures, each with 11 different choices, corresponding the 11 possible rolls of a pair of dice. Thus, there are $11^{16}$ possible different waltzes. In the 20th century, composers such as John Cage created aleatoric music using rolls of the dice or other chance events.
Compositional choices can be determined by statistical distributions. Stochastic algorithms such as Markov chains can be used to generate music. These methods can be applied in either a bottom-up fashion, selecting notes or chords left to right, or in a top down manner, guided by a generative grammar that decomposes the structure of a type of music, such as 12-bar blues.

Another approach is knowledge-based systems that use a set of rules to create or complete a composition. The rules correspond to composition choices consistent with a given style. For example, traditional counterpoint can be viewed as rules that constrain a composition. An example of such a rule is to avoid parallel fifths.

I envision a hybrid approach for my project which combines rules for a specific genre with some chance elements when multiple rules may apply. The system might also try to generate all possible harmonic progressions consistent with the rules, and then rank them according to aesthetic principles.

Using Haskell, the system will have a graphical user interface, and produce audio output that can be converted to sheet music using music software.

**Timeline and Deliverables**

**Week 1**

Research methods of harmonizing melodies.

**Week 2**

Research what makes harmony “good” historically and what can work or not work.

**Week 3**

Implement basic scale degree triad based harmony, taking in a key and a melody and outputting the melody harmonized by scale degree triads.

**Week 4**

Implement melodic analysis, deducing time signature, and key from inputted melody.

**Week 5**

Research how harmonic rhythm decision making works. Look at sample songs and analyze their harmonic rhythm and if and how it correlates with melodic rhythm and time signature.
Week 6
Plan and implement harmonic tempo decision making based on analysis.

Week 7
Implement harmony changes based on melodic interval changes.

Week 8
Implement harmonization with multiple possible outcomes.

Week 9
Create graphical user interface. Add 7th chords, and non-scale chords to create more complex harmonic possibilities.

Week 10
Test the system with additional melodies. Refine and expand the algorithm as needed.

Week 11
Miscellaneous final coding and write up.

Week 12
Develop website for cleaner presentation of project.