HarMLny Proposal

Kalen Chang

June 11, 2020

Abstract: HarMLny is a machine-learning based harmonizer. The proposed harmonizer takes a melody and returns the melody harmonized with chords below, making decisions based on information learned from musical data.

Literature review: haven't done it yet. But I will soon, hopefully.

Foundations of harmony: Harmony in the Common Practice Period (CPP, eg. Baroque, Classical, etc.) and beyond (some kinds of pop music, Asian pop music, some kinds of soundtracks, ragtime, etc.) follows certain "rules"—which are more like tendencies. Summarized roughly, some of these include:

- harmonic progression and cadential movement
  - phrases tend to move from Tonic to PreDominant to Dominant (T > PD > D)
  - this results in patterns (in major) like: ii/IV > V/vii^{o} > I/vi
  - circle of fifths motions: iii > vi > ii > V > I, also manifested as secondary dominants (II=V/V > V)
  - cadences tend to be marked by melodic, harmonic, and rhythmic "finality" coming together. Thus, if we can detect notions of cadentiality/finality in the melody, we can perhaps determine more easily the harmonic possibilities.
- melody notes tend to be included in the chord (note that this is the reverse of what we want, though we could *probably* assume the reverse is true)
  - melody notes not in the chord are called Non-Chord Tones (NCT), and in CPP, NCT are restricted to specific types (passing, neighbor, escape, etc.) which tend to be scalarly close to chord tones, and (possibly) more diatonic than chromatic
  - melodically strong notes should match the chord more. this, however, can be hard to determine; syncopation may create "strong" off-beats. perhaps duration is a good determinant?
- beyond diatonic triads in root position:
  - chord inversions often have unique/different functions. for example, a cadential 6/4 (I chord) precedes a dominant V
  - chords extensions (7ths, 9ths, etc.) often function similarly to the triad it extends, but sometimes has different functions (ex. dominant 7ths "need" to resolve to I). When you get to some 7ths, most 9ths, and beyond, this is usually in the "jazz" territory. If a

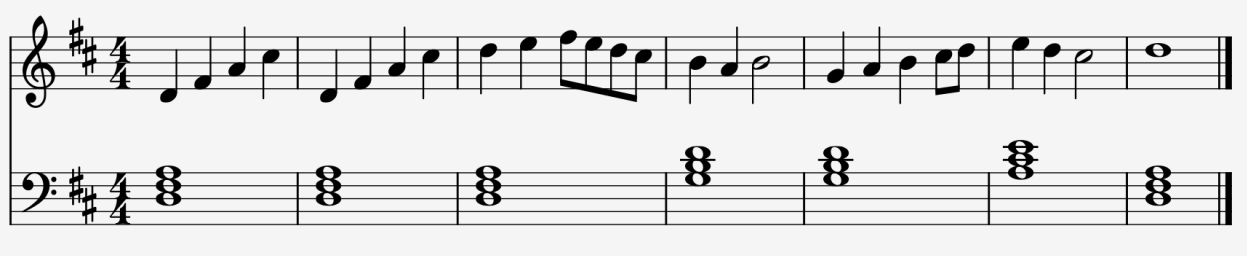
harmonizer is built well, ideally it could harmonize jazz as well, as long as you train the machine right.

- other additions (add/sus2, add6, sus4) are possible, but perhaps these chords should be treated as functionally the same as the basic triad
- modal mixture (ex. iv in a major key) may correspond to chromatic notes in the melody. in addition, other qualities (ex. augmented chords, diminished chords besides vii^{o})

- finally, some chords inherently are more popular/common than others

Building a harmonizer: A harmonizer would take a melody (as a midi) generate a set of possible chords for a given "harmonic rhythmic unit" (see below), take the above constraints, and give values for each of the chord possibilities (numeric/boolean). Then it would calculate a score for each of the chord possibilities based on certain weights for the constraints, and choose the optimal chord. It then outputs the melody with chords (as a midi, probably).

An example is given below. For the first 3 measures, DM is chosen because of the melody notes D F# A which appear in the chord. Not sure how it calculates m4. M5 clearly has a scale starting on G, with G B on beats 1 3, so it picks GM. IV leads to V leads to I, which is also supported by the notes in the melody.



Note that harmonizing is not always a linear task from left to right. If we assume the last measure ends on I, we can guess that before that comes some kind of dominant. So working backwards may also be a valid approach. But we can't do both, unless we have some way of making decisions on whether to prioritize going forward or backward (or skipping around!). Perhaps we could calculate the "scores" for chords for either 1. the first/last measure without a chord, or 2. each measure without a chord, then depending on how high the scores are, choose which measure to assign a chord to, and then assign the highest scoring chord for that measure.

Harmonic rhythm is also a big issue (though I have not studied much about it). Typically you have one chord per beat or 1-2 per measure (depending on the time signature of course), and maybe we would want to calculate the scores for different possible harmonic rhythms, i.e. compare the score of one chord per measure to two.

In the realms of possibilities: of course we should start with basic diatonic chords (I ii iii IV V vi vii^{o}) or even I IV V initially, but how much we want to expand really depends. There is a balance between getting the very basic harmonic-functional aspects (i.e. T vs. PD vs. D) and letting the user embellish and substitute chords as necessary, versus trying to have the harmonizer account for all possible variations of chords. For example, V and V+ (augmented) both can be dominant Vs, but if you have #2 and no 2 in the melody, we would definitely want V+ and not V. Harmonizing a melody definitely allows some degree of artistic flexibility, which we can grant to the user, but at the same time, it is worth testing the "artistic limits" of such a system, i.e. to what extent can a harmonizer make human-like artistic decisions. One possible danger of expanding possibilities to include very similar chords (ex. ii and ii7): the harmonizer can be easily influenced by small, imperceptible differences in the data, because many characteristics of the chords are similar (harmonic function, number of chord tones in melody, etc.).

A voice leader: A voice leader (sounds like a choir position haha) would take either: 1. melody and chords, or 2. soprano and bass, and return full SATB parts. In the case of 1, this could be the output of a harmonizer. In the case of 2, this could be the output of a harmonizer IF the harmonizer gives inversions. In either case, it could also be user input—though for 1, a midi file would not be sufficient.

Do we want a voice leader? If the desired end goal is keyboard accompaniment, then no. We could write an accompanier that chooses (randomly? or based on the melodic rhythm somehow?) some kind of accompaniment, e.g. block chords, waltz, alberti bass, etc. Or just return block chords. However, a voice leader could be cool—then combined with the harmonizer would achieve a similar effect as the Bach Google doodle.

There are somewhat strict rules determining voice leading, perhaps even more strict than harmony. Some of these include:

- limits on voice ranges
- avoiding parallel 5ths and octaves
- inner voices should be somewhat smooth (minimize jumps)

Of course, this would almost be a project on its own, of similar size to a harmonizer.

On machine learning: How does ML come into play? To be honest, my understanding of ML goes no further than cheese-vs-disease. But if we give the machine some data, it can calculate to what extent the musical data follows the constraints, then determines which constraints to prioritize and value more when making decisions. The two types of data I can see us using are:

1. previous compositions, such as midi files of Bach chorales

2. having the program generate many random harmonies, and surveying people (us?!) on what sounds good/what sounds better.

Both cases would most likely require a lot of data, which may be hard to get.

A side note: I don't particularly like ML work done in NLP. That's because it's usually clear when some piece of language is grammatical or logical, and that language structure (syntactically) is not just based on linear structures, but also hierarchical ones. And thus non-linguistically informed NLP tends to fall short. But is music similar? It's claimed that musical syntax has hierarchical structure. But whether music is "good" or "bad" is perhaps less clear cut than whether language is grammatical or not.