

Take a Break, Bach!

Let Machine Learning Harmonize That Chorale For You

Chris Lewis
Stanford University
cmslewis@stanford.edu

Abstract

In this project, I explore the effectiveness of the Naive Bayes Classifier and the Hidden Markov Model in generating Alto-Tenor-Bass harmonies for a given Soprano line. Basing these models on the corpus of Bach's chorales, I measured the predictive accuracy of each model on a test set to obtain an empirical perspective on the musical correctness of my models' harmonizations. I found that both models ultimately produced qualitatively "pleasing" harmonizations, and they were each marginally accurate at best in predicting harmonies for each note.

1. Introduction

A Bach chorale is by definition comprised of four vocal parts in the Soprano, Alto, Tenor, and Bass ranges, respectively. Virtually every chorale abides by a strict set of rules dictating the realm of possibility for the melodic, harmonic, and rhythmic motion over the course of each piece. As a result, the corpus of Bach chorales is musically quite homogenous, making the chorale form an ideal candidate for machine learning algorithms that aim to generate appropriate harmonic voicings by example.

There are additional qualities of the chorale form that lend it well to machine learning applications. While Bach occasionally composed both melodies and their resultant harmonizations himself, his most common compositional tactic was to rehash melodies from Gregorian chants and devise accompanying chord progressions from the melodies' implied harmonic motion. I take an identical approach to har-

monization here, attempting to generate Alto, Tenor, and Bass lines according to a given Soprano melody, just as Bach often did. Concretely, my goal was to automatically produce pleasing and musically correct chord progressions to underlie an arbitrary Soprano melody given as input.

2. Models

2.a. Naive Bayes Classifier

In some sense, the task of harmonizing a melody can be treated as a classification problem, wherein every Soprano pitch must be classified as belonging to a specific chord voicing. I thus began with a Naive Bayes (NB) Classifier for this classification task, as it would require only basic frequency counts from the data, yet it was still likely to produce fairly pleasing harmonizations using a Maximum Likelihood Estimator to pick the most common chord for each melody note:

$$\text{classify}(f_i) = \underset{c}{\operatorname{argmax}} p(C = c) \prod_{j=1}^n p(F = f_j | C = c)$$

2.b. Hidden Markov Model

On the other hand, the central assumption of the NB classifier is that each feature (in this context, a melody note) F_i is conditionally independent of all other features (melody notes) F_j for $j \neq i$; therefore, the NB model would completely disregard music theory's rules governing chord progressions (see Figure 1). Thus, to incorporate these rules without hard-coding them directly, I also trained a Hidden Markov Model (HMM), which added basic consideration for the "correctness" of progressions by leveraging not only *emission probabilities* of chords mapping to a melody note:

$$P(F_i = f | C_i = c)$$

but also *transition probabilities* between successive chords:

$$P(C_{t+1} = c_a | C_t = c_b)$$

as well as *start probabilities* describing how likely each chord was to appear as the first chord in a piece:

$$P(C_1 = c).$$

The Viterbi algorithm could then analyze this information and calculate the most likely sequence of *hidden states* (four-part chord voicings), given a string of *observed states* (Soprano melody notes).

These NB and HMM models would both be quantitatively tested for predictive accuracy, later on. As a separate exercise meant purely for qualitative improvements, I extended the HMM approach to produce a model for adding ornamentation notes, wherein the observed states would be the four individual, part-wise pitch differences between notes in successive four-part chord voicings, and the hidden states would be a "rhythmic spelling" (discussed in

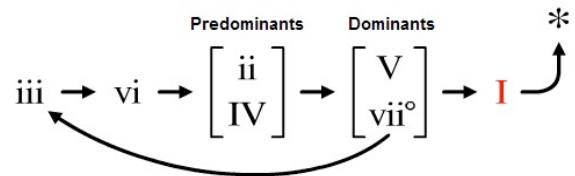


Figure 1: Permissible chord progressions in a major key. We can expect the HMM harmonization to demonstrate some attention to these rules because of its consideration of transition probabilities between chords in the training data (Source: http://www.electricchili.com/wp-content/uploads/2010/06/ScreenHunter_13-Jun.-20-08.47.jpg)

the next section) that could be interpreted to add ornamental notes in the final score.

3. Data Encoding

I used the Python-based music21 library to access a corpus of Bach chorales encoded in the MusicXML format. This format proved impractical for the data analysis methods required in this project; thus, I researched and/or devised several textual encoding schemes for the data at hand.

3.a. Encoding pitches

Soprano pitches needed to be encoded in such a way that every distinct pitch was distinguishable from all others. A simple name/octave combination proved sufficient to uniquely identify each pitch value (see Figure 2).

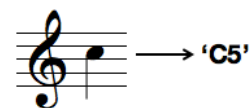


Figure 2: Sample pitch encoding for a Soprano note.

3.b. Encoding harmonies

Harmony pitches in the Alto, Tenor, and Bass parts were encoded as an ordered four-tuple (a "chord spelling") with the accompanying soprano pitch as the first element. To reduce the size of the chord spelling vocabulary and to

ameliorate the process of comparing chord voicings from chorales written in different keys, I represented each harmony pitch in a chord spelling not with its name and octave, but rather with an integer specifying that pitch's distance in semitones from the soprano pitch of that chord (see Figure 3). This encoding was inspired by a similar metric used by Allan and Williams in their research^[1].

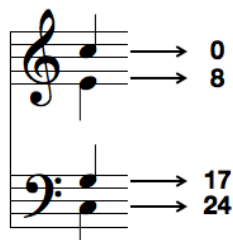


Figure 3: Sample chord spelling based on semitone distance from the Soprano pitch.

3.c. Encoding rhythms

Because chord spellings encoded information vertically across all four parts for each pitch, developing a similar approach for encoding ornamental non-chord tones would be crucial to the success and efficiency of the second, ornamentation HMM. Rhythm is therefore represented for each Soprano pitch as four, part-specific four-tuples indicating both the beat-relative locations and the semitone distances for any ornaments occurring in conjunction with a Soprano pitch (see Figure 4).

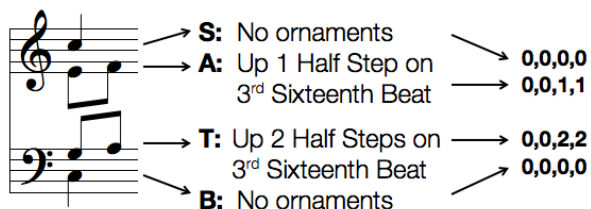


Figure 4: Sample rhythm spelling for a one-beat window.

4. Training Phase

4.a. Naive Bayes Classifier

In accordance with the basic goals of this model, the training phase for the NB model was straightforward. Given a subset of chorales in MusicXML format for training, the model first encoded all vertical pitch-harmony pairs in the corpus using the aforementioned

encodings. Then, the model computed the appropriate prior and conditional probabilities using raw occurrence frequencies.

4.b. Hidden Markov Models

For the HMM harmonization model, along with the absolute soprano pitch and pitch-distance spelling for each chord, I recorded whether the chord in question occurred at the very beginning of the chorale in which it appeared, as well as the spelling of the next chord that appeared in the chorale. With this information, I was able to complete the training phase by calculating the necessary start probabilities, transition probabilities, and emission probabilities for all chords and/or Soprano pitches in the training corpus. I used a virtually identical approach in training the HMM ornamentation model, simply substituting chord spellings for pitches and rhythm spellings for the harmonic pitch-distance spellings.

5. Testing Phase

5.a. Design

I trained both models on a random subset amounting to 70% of the corpus of Bach chorales, which left 30% for testing purposes. The testing phase first extracted the Soprano melody lines from the chorales in the test set, and then gave each of these melody lines as input to both classifiers in turn (note: the ornamentation HMM was not utilized for empirical testing purposes). The models then emitted their likeliest SATB chord spellings for each pitch based on their respective parameters and considerations. Ultimately, each of these chord spellings was compared with the actual chord spelling for the associated pitch in the original chorale, and an overall predictive accuracy was then calculated for each model. The ornamentation HMM was evaluated separately, with no quantitative measure, as it was simply developed as a proof of concept in the first place.

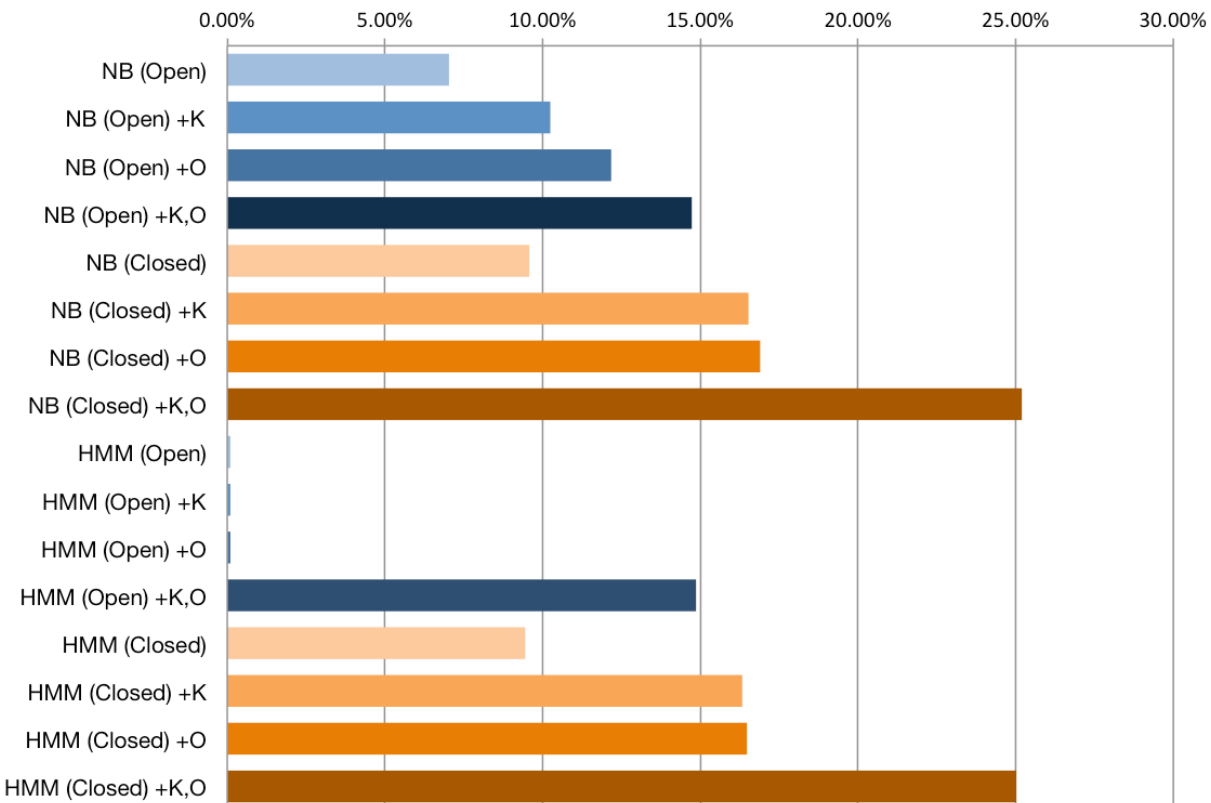


Figure 5: Predictive accuracies of the Naive Bayes (NB) model and the Hidden Markov Model (HMM) on a test set of Soprano lines extracted from Bach chorales. In “open” trials, models tried to predict the exact chord voicing for each harmonized melody note. In “closed” trials, models simply tried to predict the correct chord type (e.g. C Major, F Minor) for the harmonization of each melody note. K = keys normalized to C Major/A Minor, O = off-beat ornaments removed

5.b. Results

While superficial checks demonstrated that both models could indeed generate pleasing, generally correct harmonic progressions, initial testing showed that the NB model only predicted 6.45% of chord spellings correctly, while the HMM was a complete failure, reporting an accuracy of 0%.

As an attempt to improve accuracy, I normalized all chorales from their original keys to C Major (or A minor, if the chorales were originally in a minor mode) in both the training set and the testing set. This resulted in slight improvement to the NB model’s accuracy, which rose to 10.2%, but the HMM still reported 0%. I also tried removing all notes that occurred on off-beats, hypothesizing that most were likely

non-chord tones polluting the training and test data with pitch combinations that were not meant to be considered as part of the chorales’ true harmonic structures. This hypothesis was verified through further testing, as the NB accuracy nearly doubled when ornament normalization alone was added (the HMM still reported 0%); however, adding key and ornament normalization together caused both the HMM and NB accuracies to jump to nearly 15%. Accuracies jumped most of all when I modified the test criteria to compare only the types of chords (in “closed” position) predicted by the models, as opposed to the chords’ exact pitch spelling. In this trial, both models’ predictive accuracy rose to approximately 25%. (Finally, I also tried working with major- and minor-keyed chorales separately, but this metric showed no improvement whatsoever in



Figure 6: A harmonization of “Mary Had a Little Lamb” produced by the two HMM models, with the harmonizing HMM producing the chord progression in a first pass, and the ornamentation HMM adding rhythmic variation in a second pass. The excessively high tenor and alto parts are likely due to the appearance of similar harmonies in lower keys within the training data (all chorales were normalized upward or downward to C Major beforehand).

my trials). Figure 5 displays the results of each trial, including those in which normalizations were added to the closed-position tests.

5.c. Analysis

Overall, the accuracy of the models exceeded my expectations, especially when predicting the exact open chord voicing for each Soprano note. The respective improvements from key normalization and ornament normalization did not surprise, however, as both of these additions clearly helped to reduce the vocabulary of pitches and chord spellings and ultimately improve the signal-to-noise ratio in the training data.

The 25% maximum accuracy is also not surprising. With four voice parts that each have 1.5- to 2- octave ranges, there are on the order of 10,000 possible chord spellings with the notes in a C-Major diatonic scale, and many times more spellings with the chromatic scale. Moreover, the data set is rife with tricky edge cases (some of which actually arise all too often): modulations to and from different keys; suspensions, appoggiaturas, and other non-chord tones that occur on the down-beat, instead of the offbeat, as well as deceptive cadences, which are deliberately designed to substitute unexpected relative minor chords instead of conclusive tonics. My models could not account for any of these, which probably resulted in significant drops in accuracy.

6. Conclusion

There are likely many additional features one could consider in future attempts at a predictive task like this one. However, my models nevertheless produced reasonably complex harmonizations that were often completely permissible by music theory’s rules of chord progressions (see Figure 6 for a nostalgic example). Past efforts in the field of machine-learned harmonization have focused almost exclusively on such qualitative measures; hopefully, the predictive analysis aspect in this project will serve as a stimulus for others to pursue further research in quantitatively verifying the performance of various chorale harmonization models.

7. References

- [1] Allan, M. and C.K.I. Williams. Harmonizing Chorales by Probabilistic Inference. In *Advances in Neural Information Processing Systems*, volume 17, pages 25-32, 2005.
- [2] Allan, M. Harmonizing Chorales in the Style of J.S. Bach. MS thesis University of Edinburgh, 2002. Print.
- [3] Biyikoglu, K.M. A Markov Model for Chorale Harmonization. Middle East Technical University. In *Proceedings of the 5th Triennial ESCOM Conference*, pages 81-84, 2003.
- [4] Cuthbert, M. S. and C. Ariza. music21: A Toolkit for Computer-Aided Musicology. MIT. 2011. Web. Oct. 2011 <<http://mit.edu/music21>>.
- [5] Hidden Markov Model. *Wikipedia*, 2011. Web. Oct. 2011 <http://en.wikipedia.org/wiki/Hidden_Markov_model>.
- [6] Schulze, W. and B. v.d. Merwe. Music Generation with Markov Models. Stellenbosch University. In *Multimedia, IEEE*, volume 18 (3), pages 78-85, 2011.
- [7] Viterbi Algorithm. *Wikipedia*. 2011. Web. Oct. 2011 <http://en.wikipedia.org/wiki/Viterbi_algorithm>.