5C-03

N-gram Based Chord Progression Interpolation for the Completion of Partial Four-Part Chorales

Christoph M. Wilk, Souya Ito, and Shigeki Sagayama

Graduate School of Advanced Mathematical Sciences, Meiji University

1 Introduction

In this paper, we present the first of two steps of an approach to complete partial four-part chorales. Designed for composition assistance, the system allows users to place notes for up to four voices as well as harmonies at any beat in the piece. The system infers all missing harmonies and inserts the missing voices. Thus, the approach can be applied to harmonize a chorale where only the soprano voice is given. It can also be used to help a composer who wants to implement only a set of musical ideas and have the algorithm fill in the rest, either to avoid the difficulty or time effort to do it by hand.

In this paper, we discuss a method to interpolate chord progressions, accounting for partial voicing and harmony. The obtained chord progressions are used in [1] to compute the four-part voicing.

2 Problem Formulation

The aim of the presented algorithm is to obtain the chord progression C consisting of n chords $c_1, \ldots, c_n$ that is most probable given statistics extracted from a music corpus. This can be formulated as follows:

$$C_{\max} = \arg\max_{C} \prod_{i=1}^{n} p(c_i | c_1 \ldots c_{i-1}) \qquad (1)$$

However, during the optimization process, the chord progressions are subject to two types of constraints that can be input as conditions at any beat in the chord progression to be generated (see Figure 1).

1. Partial voicing: The user can set up to four notes at each beat. The program infers all possible chord candidates from these notes. It is possible that no chord fits the note pattern (e.g. four successive semi-tones), in which case the program reports the unsatisfiability of the constraints.
2. Partial harmony: For each beat, the user can input an arbitrary amount of chord candidates to be allowed at that point in the progression.

Chord Definition

A chord is defined as a tuple $c = (d, q, b, v)$, where d is its scale degree, which is needed to account for the harmonic functionality of chords. q is the quality of the chord, such as major, minor or dominant seventh. b and v denote the beat position and note value (length) of the chord, respectively, which are needed to reproduce correct harmonic rhythm.

Harmonic Functionality

Humans generally perceive a musical chord in relation to the surrounding harmonic context, called key. For example, the G major chord is a dominant chord in the key of C major, expected to be followed by the tonic C major chord. In the key of D major, however, the G major chord would be subdominant, and one would rather expect the next chord to be a dominant A major chord. Therefore, the scale degree, denoted with roman numerals from I to VII, encodes information that is more relevant to harmonic progressions than the absolute pitch class.

Harmonic Rhythm

The importance of harmonic rhythm can be understood considering the concept of cadences, which are harmonic progressions creating a sense of resolution. For example, a musical phrase concluded by the perfect authentic cadence “I - IV - V - I” evokes a feeling of completion. However, if the phrase already ends on the V chord, one obtains the half cadence “I - IV - V”, which makes the listener expect a continuation, even if it is followed by a I chord. Where a phrase ends is perceived based on the rhythmic structure of the music. Therefore, rhythmic information of chords is important.

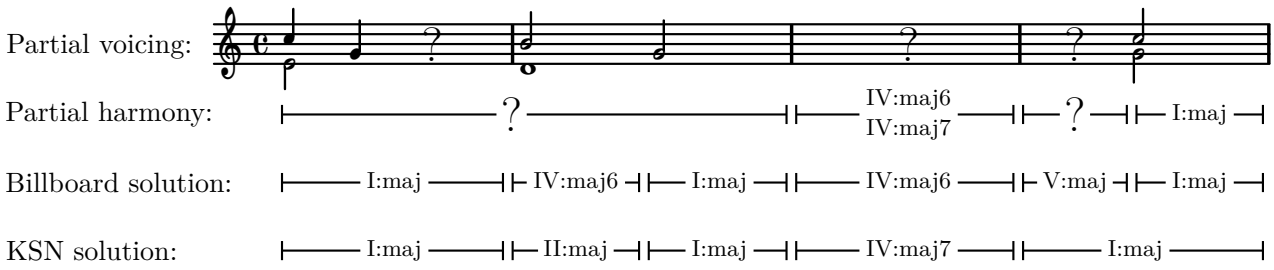

Figure 1: Exemplary results of chord progression interpolation in C major. The algorithm is constrained by both partial voicing and harmony shown in the upper half. In the lower half, the most probable chord progressions based on n-gram statistics of the Billboard data set as well as the KSN data set are shown.

3 Interpolation Algorithm

We compute the probabilities of chord progressions using n-grams, which have been used successfully in previous research on harmony analysis [2].

A chord progression can be viewed as a path through a forest graph, where each node represents a chord, each root node a progression's first chord, and each leaf node a final chord. An edge's weight corresponds to the n-gram probability of the next node. For solving the optimization problem (1), the conditional probability of the edge from c_{i-1} to c_i given the previous chords, is approximated as follows:

$$p(c_i|c_1 \dots c_{i-1}) \approx \sum_{n=1}^{N} a_n p_n(c_i|c_{i-n+1} \dots c_{i-1}) \quad (2)$$

where a_n are weight parameters and p_n the n-gram probabilities. For this paper, we used $N = 3$ with weights $a_1 = 0.01$, $a_2 = 0.09$ and $a_3 = 0.9$. This means that trigrams are prioritized, but since they often do not exists for complicated constraints, the inclusion of lower order n-grams increases the number of satisfiable constraints.

Considering that all edge weights (2) of the chord progression graph are positive, on can use Dijkstra's algorithm to find the chord progression with the highest probability. The harmony constraints are accounted for by removing all nodes from the graph that violate the constraints. This implies that there might be no valid path, in which case the program reports the unsatisfiability of the constraints.

4 Exemplary Results

Data sets that can be used for the presented algorithm have to provide information about both harmonic functionality and rhythm of chords. Two data sets that satisfy both conditions are the McGill-Billboard data set [3], which contains chord progressions of 890 popular songs, and the KS notation data set [4], which contains functional harmony annotations for 81 classical pieces of the RWC music database. Figure 1 displays exemplary chord progressions obtained from the algorithm.

5 Conclusion

In this paper, we discussed the musically important properties of a chord progression and presented an algorithm that generates chord progressions based on n-gram statistics while satisfying both voicing and harmony constraints. The obtained harmonies are used in [1] to compute four-part voicing.

To improve the algorithm, one could use more sophisticated n-gram smoothing, or introduce skip grams to handle n-gram sparsity in small data sets. Eventually, we plan to combine chord progression interpolation and automatic voicing in order to perform joint optimization.

References

[1] S. Ito, C. M. Wilk, and S. Sagayama. An HMM based automatic voicing algorithm for the completion of partial four-part chorales. *IPSJ*, 2018.

[2] R. Scholz, E. Vincent, and F. Bimbot. Robust modeling of musical chord sequences using probabilistic n-grams. *ICASSP*, 53–56, 2009.

[3] J. A. Burgoyne, J. Wild, and I. Fujinaga. An expert ground truth set for audio chord recognition and music analysis. *ISMIR*, 633–638, 2011.

[4] H. Kaneko, D. Kawakami, and S. Sagayama. Functional harmony annotation data-base for statistical music analysis. *ISMIR*, 2010.