Harmonic Analysis of Jazz MIDI Files Using Statistical Parsing 5th International Workshop on Machine Learning and Music

Mark Granroth-Wilding MARK.GRANROTH-WILDING@ED.AC.UK
Mark Steedman STEEDMAN@INF.ED.AC.UK
School of Informatics, University of Edinburgh, 10 Crichton Street, Edinburgh, UK

Abstract

Harmonic analysis involves identifying hierarchical structure, similar to that found in the syntax and semantics of language, in the harmonic progressions underlying tonal melodies. In previous work, we have used grammar-based parsing, with related machine-learning techniques, for automatic harmonic analysis of jazz chord sequences.

We now turn to the harder task of harmonic analysis of jazz MIDI performances using similar techniques. First, we evaluate a strict pipeline approach: we use an HMM to perform chord recognition and our previous system to analyze the output. Then, we use the chord recognizer to propose a chord lattice and analyze this using an adaptation of the previous system.

1. Introduction

Hierarchical structure, similar to that found in the syntax of language, can be identified in the harmonic progressions that underly melodies in tonal music. Harmonic analysis is the process of identifying this structure given an ambiguous musical surface form. We have previously described the use of grammar-based parsing, using supertagging and statistical parsing models adapted from natural language processing (NLP), to perform automatic harmonic analysis of textual jazz chord sequences (Granroth-Wilding & Steedman, 2012 (to appear)). We represent a harmonic analysis of a passage of musical as a set of tonal relations between chords, expressed in the infinite three-dimensional space of Longuet-Higgins (1962a;b). In some cases, these exist between distant chords, just as linguistic syntactic dependencies between words may strech over many intervening words.

In *5th International Workshop on Machine Learning and Music*, Edinburgh, Scotland, UK, 2012.
Funded by ERC Advanced Fellowship 249520 GRAMPLUS and ESRC Postgraduate Studentship ES/H012648/1.

The present paper is an extension of our previous work on chord-based analysis to MIDI input. We demonstrate a naive technique, then an improvement, evaluated against hand-labeled gold-standard analyses.

2. Corpus

We have constructed a small annotated corpus of chord sequences for jazz standards (76 songs, 3000 chords). Each sequence is fully annotated with the syntactic information required to produce a harmonic analysis. We used the corpus to train and evaluate our chord parsing models (Granroth-Wilding & Steedman, 2012 (to appear)) and here use the same corpus to evaluate a MIDI parsing system. It also contains MIDI performance data for many songs, collected from the web.

3. Evaluation

In NLP, *dependency graphs* are commonly used to represent the syntactic relations between the words of a sentence. Each node of the graph is associated with a word in the sentence. Each arc represents a relation between two words and may be labeled to describe the nature of the relation. The same representation can be used for a harmonic analysis of a chord sequence: nodes denote chords and labeled arcs denote the constraints on the tonal relations between chords. An analysis from the parser may be evaluated against a gold-standard labeled chord sequence using the usual metric of labeled attachment score (LAS).

A system that produces a harmonic analysis of MIDI data must segment the input into the units of harmonic structure – chords. We cannot evaluate its output against the gold standard, since the segmentation may differ, so that there is not a one-to-one match between the nodes of the dependency graphs. Instead, we

	P	R	F	Cov
Chord	83.46	81.63	82.53	96.05
Pipeline	63.80	50.33	56.27	78.05
Lattice	69.94	54.30	61.14	92.68

Table 1. Dependency recovery (OLAS) of the chord sequence parser and the two extensions to the harder MIDI analysis task. For each evaluation we report precision (P), recall (R), f-score (F), and coverage (Cov), all percentages.

use an evaluation metric, *optimized LAS* (OLAS), that rewards any structure shared between the two graphs: we find the alignment between their nodes that results in the most recovered dependencies and report the resulting LAS.

This metric allows us to evaluate the similarity of the parser's output for a MIDI file to the gold-standard analyses of the chord sequence of the same song in our corpus. The chord sequence in the corpus may differ from that used in the MIDI file, so a perfect analysis of the MIDI file may not acheive a 100% match by this metric. However, we can expect the harmonic structure to be very similar and assume that the metric therefore gives a reasonable indication of the quality of the analysis.

4. Extension to MIDI Input

4.1. A Naive Pipeline

An obvious way to extend the previous, chord-parsing system to the task of MIDI parsing is first to use a simple model to perform chord recognition, producing a sequence of chord symbols from the MIDI performance data, and then to parse the output chord sequence using the old system without modification. Although most recent work on chord recognition has focused on taking audio input, the HMM described by Ni et al. (2011) is easily adapted to our task. We use an identical transition distribution, but for the omission of the bass note from the state labels, since we do not consider it important to distinguish chord inversions. In place of their distribution over frequency-band loudness features corresponding to equally-tempered pitch classes, we use a distribution over MIDI pitch classes. We train the model using EM. We refer to this system as Pipeline below.

4.2. Lattice Parsing

Pipeline allows trivial extension of the chord parser, but has the drawback that the parser's interpretations are limited by the chord recognizer's choice of chord. We can use the chord recognizer without committing to a single chord label. We modify the parser to take input in the form of a *lattice*: multiple possible chord symbols per timestep, weighted by the state occupation probability assigned by the chord recognizer. The parser can produce interpretations using probable chord labels from the chord recognizer other than the single most probable label. We call this system Lattice.

4.3. Results

We report the results of evaluation by OLAS in table 1. A small amount of pre-processing of the MIDI inputs is required before the chord recognizer can be applied to them. We evaluate on 41 MIDI files – the full set currently available to us corresponding to songs in the corpus. For comparison, we include the parsing results for the chord parser evaluated using the same metric.

Using a lattice achieves a substantial improvement in f-score. This is partly due to the greater coverage of Lattice: constraining the parser by considering only one chord label prevents it in some cases from producing any interpretation at all.

5. Future Work

The results above demonstrate that simple machine learning techniques can be used to adapt the chord analysis system of Granroth-Wilding & Steedman (2012 (to appear)) to perform harmonic analysis of MIDI data. We are currently considering ways of incorporating a model of MIDI data directly into the front-end component of the parser (the supertagger), removing the need to produce chord labels as an intermediate step.

References

Granroth-Wilding, Mark and Steedman, Mark. Statistical parsing for harmonic analysis of jazz chord sequences. In *Proceedings of the International Computer Music Conference*. International Computer Music Association, 2012 (to appear).

Longuet-Higgins, Christopher. Letter to a musical friend. *The Music Review*, 23:244–248, 1962a.

Longuet-Higgins, Christopher. Second letter to a musical friend. *The Music Review*, 23:271–280, 1962b.

Ni, Yizhao, Mcvicar, Matt, Santos-rodriguez, Raul, and Bie, Tijl De. Harmony progression analyzer for mirex 2011. In *Proceedings of the 12th International Society for Music Information Retreival (MIREX)*, pp. 1–4, 2011.