Symbolic Representation of Tonal Progressions for Rule-Based Evaluation

Eddy K. M. Chong[*], Qin Ding, Yu Zhenyu, Jia Ruitao [†]

[*]Visual and Performing Arts Academic Group, Institute of Education, Nanyang Technogical University, Singapore

[†]Institute of System Science, National University of Singapore, Singapore

eddy.chong@nie.edu.sg

Abstract

This paper presents a data representation system to encode musical chords used in a classical tonal context. Whilst a number of approaches to representing musical chords have been proposed for various computational or data retrieval purposes, the present project differs in its being motivated and shaped by music-pedagogical intentions, which include offering automated error feedback. Hence, instead of a data-driven pattern-discovery approach, we adopt a rule-based method using declarative rules. Both the encoding system and the formulated rules are based on classical theory of functional harmonic voice leading. The ultimate aim is to develop a musically-intelligent interactive system that automates both the evaluation of tonal progressions and the provision of assessment feedback for the purpose of teaching western classical tonal theory. In a nutshell, the computational system's input are chords drawn from the major-minor tonal system. Roman numerals with figured-bass indications (e.g. I^6, vii°4/2) are the chord symbols to be encoded. The encoding reflects a number of pertinent theoretical elements: (i) the key context, (ii) the scale degree of the chord, (iii) the chord type, and (iv) the chord inversion. A rule-based system based on JBoss Drools and the Rete algorithm is then designed to evaluate chord progressions based on considerations of root motion, bass movement, and other tonal voice-leading factors. The musical intelligence of this system therefore simulates human musical thinking as encapsulated in a typical undergraduate theory of harmony.

Keywords: music chords, tonal progression, music computation, data representation, rule-based

1. Modeling tonal harmony

The grammar of music harmony has a complexity that defies simple computational representation and that has certainly engaged the imagination of music theorists for centuries. Broadly speaking, in music computational endeavours, two directions have been explored—either to model after human musical thinking, or to aim for an output capability that is comparable to human efforts, no matter if the system is musically naive or whose "musical thinking" is opaque to the system designer. Both approaches have their pros and cons, the choice sometimes depends on specific applicational interests. For the purpose of automating analysis of tonal progressions, especially if pedagogical use is central to the agenda, the preference will naturally be for the former direction.

In our project, we necessarily align with common musical understanding in order to develop an interactive system that not only evaluates the validity of chord progressions in the manner of a trained musician, but offers automated error feedback to the user similar to that given by a music theory teacher. The knowledge base of this AI system—from its data encoding to the declarative rules and constraints implemented—is grounded in music theory as opposed to being based on statistical comparison of patterns or structures. Within the confines of this paper, we shall present in more detail the symbolic representation system and only briefly touch on the rule-based system that is used for the evaluative algorithm.

2. Input data

In designing the evaluation sub-task, one of the first decisions to be taken concerns choosing the type of data to work with. Given that music is a sound art, one would expect that from a human-and-sound interaction point of view, the preferred input would be sonic in nature. Indeed, there have been quite a number of such precedents (Conklin, 2002; Dixon, 2010; Mauch & Dixon, 2010; Mauch, Noland, & Dixon, 2009). However, our pedagogical objective is to focus the students' learning on mastering the basic principles of harmony, expressed in Roman numeral and voice-leading terms. Hence, we have chosen not to have students use audio input (e.g. playing the progression on a midi keyboard or recording the progression) or even music-score notation, but to input using the Roman numeral system taught in class. Of course, for future developments of the system, we can add a conversion component to accept alternative "raw" inputs, but our present focus is to develop the evaluative function itself, which is the core of the system.

By opting for the abstract form of Roman numerals, we are also simplifying the overall computational task by eliminating the need to handle non-harmonic tones, a complication that a number of researchers have chosen to tackle based on their differing motivations but only succeeded to a limited extent (Harte, Sandler, Abdallah, & Gómez, 2005; Mauch, 2010; Maxwell, 1992; Sapp, 2007; Winograd, 1968). Nor do we need to deal with key-finding (like those in Maxwell, 1992; Sapp, 2007; Taube, 1999; Winograd, 1968) or the parsing of different types of musical texture (e.g. Maxwell, 1992). Admittedly, the use of Pop chord symbols (e.g. Anglade & Dixon, 2008; Granroth-Wilding & Steedman, 2012; Temperley & Sleator, 1999) may sidestep a number of the above computational challenges, but it still entails the additional sub-tasks of key-finding and of determining chord relationship, the two being strongly mutually dependent.

Having opted for the Roman numeral system, there is still the need to choose the particular version. In accordance with the theory that is taught in our context, our Roman numeral system differentiates between chord types, as illustrated in **Table 1**. This separation between major and minor key paves the way for handling borrowed/mixture harmonies for future extension of the system, which currently deals only with diatonic harmonies. We use figured-bass numbers to indicate the chord inversion—for example, 6/3 for first inversion, 6/4 for second, and so forth—as opposed to using suffix alphabet letters (e.g. Ib, ivc, V_7d). The notion of harmonic function is also integral. Hence, cadential six-four is clearly distinguished from its other counterparts, which retain their Roman numeral. For example, I 6/4 may be a neighbouring or passing six-four, but it is decidedly not a cadential six-four, which is symbolized as C 6/4.

Major key	Minor key
I	I
ii, ii_7	ii, $ii^{ø}_7$
iii	^{b}III
IV^6	iv^6
V, V^4_2	v, V, V^4_2
vi	^{b}VI
vii^{o6}, vii^{o}_7	^{b}VII, vii^{o6}, vii^{o}_7

Table 1: List of sample diatonic chords for the major and minor keys

3. Data representation

The next step is designing the data representation system. Whilst there are quite a number of versions currently available, none of them entirely meets our pedagogical need. This is not surprising since the suitability of any representation system depends very much on the task it is designed to serve, and it must factor in both the context of the notation itself as well as the processes that it is subject to (Wiggins, Miranda, Smaill, & Harris, 1993).

In our case, the data representation must be aligned with the tonal theory that is taught. Specifically, it must reflect diatonic and chromatic relations as well as other features pertinent to harmonic voice-leading considerations. With this in mind, we adopt a four-element vector <a,b,c,d> in which a represents the key (1 = major, 2 = minor), b represents the scale degree (1 to 7), c represents the chord type (1 = major triad, 2 = minor triad, 3 = diminished triad) and d represents the chord inversion (0 = root position, 1 = first inversion, and so forth). For scale degree indications, the major-key scale step is used as a reference. Hence, in the minor key, the mediant chord is ^{b}III and is encoded as "36", where the second digit "6" signifies the chromatic lowering ("6" morphologically resembles the flat sign). On this basis, mixture/borrowed harmonies can be easily distinguished subsequently.

For seventh chords, the encoding rationale is a little more complicated. With extended major chords, a second digit "7" is added. In the context of classical usage, whether the added seventh is a major or minor one depends on the chord in question. For example, in the major key, c = 17 will mean added major seventh in the case of I_7 and IV_7, but added minor seventh in the case of V_7. Extended minor chords are more straightforward since only minor seventh (c=27) is typically added in classical harmonies. To differentiate between full- and half-diminished seventh chords, "37" and "34" are used respectively. Additionally, we assign the cadential six-four a special category (c = 564, regardless of the key).

With this chord encoding system, a progression would be represented as a string of vectors. For instance, the commonplace progression $ii^{ø6}_{5}$-V_7-i in the minor key is encoded as <2,2,34,1>, <2,5,17,0>, <2,1,2,0>.

4. A case for declarative rules

In music, harmonic grammar is either grasped intuitively by musicians or explicitly learnt. To model this grammar, music computation researchers either induce such rules from a set of data or adopt declarative rules based on known theory. While the statistical approach of the former may yield relevant results, we have opted for a rule-based approach for the following reasons:

i. The musical logic of the "discovered" rules remains opaque, hence inductively-derived rules are ill-suited to provide the specific error feedback that we desire, especially one that is aligned with the theory that is taught.
ii. Even if we attempt to interpret the "discovered" rules in musical terms, it is likely to be a tedious process with limited benefits. For example, one fairly recent inductive logic experiment applied to jazz and pop harmonies has yielded over 12000 rules, most of which cover less than 5% of the data (Anglade & Dixon, 2008).
iii. The significance of inductively-derived rules is based on statistical count rather than musical consideration, so unless one is interested only in the input-output efficiency of the model, one would need to further ascertain the music-theoretic validity of the rules. In other words, frequency of occurrence and statistical significance do not automatically equate with musical significance, the latter needs to be humanly evaluated (Conklin & Anagnostopoulou, 2001), and for our purpose, this would be necessary.

5. Overview of rule-based system

Since this paper is focused on presenting the data representation aspect, we shall offer here a quick overview of our rule-based system. In computer science, a rule-based system embodies knowledge that can be used to interpret information in useful ways. In artificial intelligence applications, such domain-specific expert system emulates the decision-making ability of a human expert. Our rule-based system is based on JBoss Drools, which is a Business Logic integrated and unified platform for a rule engine to operate with a particular work flow. Rete algorithm, an efficient pattern matching algorithm, is the basis of our Drools rule engine.

Our musical rules are formulated based on considerations of root motion, bass movement, and other tonal voice-leading factors. These can grow to unwieldy dimensions—especially as we move towards more advanced harmonic styles subsequently—such that the traditional "if-else" algorithm will be less efficient in processing the input data. We therefore chose to develop a rule-based system.

There are broadly two categories of rules implemented. The first stipulates certain constraints in the light of the specific kind of harmonic progressions we wish the user to be focusing on. For example, "All chords in a progression must be in the same key" limits the user to diatonic progressions for now. The second category pertains more specifically to harmonic voice-leading considerations. For instance, "If the current chord inversion (d) is 2 and the current nature (c_x) is 564, then the next scale degree must be 5 and its d value must be 0 or 3"; this stipulates the two common resolutions of the cadential six-four chord. In this connection, if this second-inversion chord is a passing or neighbouring one, three other rules are implemented to ensure proper voice-leading handling of this unstable harmony.

6. Beta testing

As a preliminary test, we have used a number of Bach's harmonized chorales from the Riemenschneider collection to test the system. Bach's chorales have been a popular choice amongst music computation researchers (Conklin, 2002; Ebcioğlu, 1992; Kröger, Passos, Sampaio, & de Cidra, 2008; Sapp, 2007; Taube, 1999; Winograd, 1968). For our purpose, not all progressions from Bach are amenable for testing our system, which currently deals only with diatonic progressions. Most of Bach's progressions involve tonicization; these are excluded since we have not included applied chords or any other chromatic harmonies for that matter. On the other hand, some of Bach's harmonizations are modal rather than functionally tonal in nature; these then wonderfully serve to test the system's discriminating ability. In general, based on the above selection considerations, some of the test progressions have involved as many as ten chords. The testing was done manually by the first author and any anomalous error feedback was used to ascertain whether the set of implemented rules needed to be revised or expanded.

7. Interim evaluation and future work

Thus far, our system can by and large successfully evaluate progressions drawn from over fifty chorale harmonizations and provide appropriate error feedback. This includes correctly detecting modal progressions that violates certain tonal functional rules. A number of faulty progressions have also been created to further test the system. The next phase of testing will involve student-created ones as well as progressions drawn from other tonal repertoire.

Once the robustness of the system is sufficiently tested, we will expand the system to embrace chromatic harmonies such as applied chords and modal mixtures. In the longer term, other harmonic styles (e.g. pop, Jazz) can be evaluated by changing the list of chords and set of rules. In fact, the encoding system itself is flexible enough to be modified to tackle even non-triadic harmonies, with the encoding vector being shortened or extended accordingly.

References

Anglade, A., & Dixon, S. (2008). Characterisation of harmony with Inductive Logic Programming. *Proceedings of the 9th ISMIR Conference on Music Information Retrieval* (pp. 63-68).

Conklin, D. (2002). Representation and discovery of vertical patterns in music. In C. Anagnostopoulou, M. Ferrand & A. Smaill (Eds.), *Music and Artificial Intelligence: Proceedings the ICMAI Conference* (pp. 32-42). Berlin, Heidelberg: Springer-Verlag.

Conklin, D., & Anagnostopoulou, C. (2001). *Representation and discovery multiple viewpoint patterns.* Paper presented at the 2001 International Computer Music Conference, La Habana, Cuba.

Dixon, S. (2010). *Computational modelling of harmony*. Paper presented at the 7th International Symposium on Computer Music Modeling and Retrieval.

Ebcioğlu, K. (1992). An expert system for harmonizing chorales in the style of J.S. Bach. In M. Balaban, K. Ebcioğlu & O. Laske (Eds.), *Understanding music with AI: Perspectives on music cognition* (pp. 295-333). Cambridge, MA: AAAI Press & MIT Press.

Granroth-Wilding, M., & Steedman, M. (2012). Statistical parsing for harmonic analysis of jazz chord sequences. *Proceedings of the International Computer Music Conference*.

Harte, C., Sandler, M. B., Abdallah, S. A., & Gómez, E. (2005). Symbolic representation of musical chords: A proposed syntax for text annotations. *Proceedings of the 6th ISMIR Conference on Music Information Retrieval* (pp. 66-71).

Kröger, P., Passos, A., Sampaio, M., & de Cidra, G. (2008). Rameau: A system for automatic harmonic analysis. *Proceedings of the 2008 International Computer Music Conference* (pp. 273-281).

Mauch, M. (2010). *Automatic chord transcription from audio using computational models of musical context.* Unpublished Ph.D., Queen Mary, University of London, London.

Mauch, M., & Dixon, S. (2010). Simultaneous estimation of chords and musical context from audio. *IEEE Transactions on Audio, Speech, and Language Processing, 18*(6), 1280-1289.

Mauch, M., Noland, K., & Dixon, S. (2009). Using musical structure to enhance automatic chord transcription. *Proceedings of the 10th ISMIR Conference on Music Information Retrieval* (pp. 231-236).

Maxwell, H. J. (1992). An expert system for harmonizing analysis of tonal music. In M. Balaban, K. Ebcioğlu & O. Laske (Eds.), *Understanding music with AI: Perspectives on music cognition* (pp. 335-353). Cambridge, MA: AAAI Press & MIT Press.

Sapp, C. S. (2007). Computational chord-root identification in symbolic musical data: Rationale, methods, and applications. *Computing in Musicology, 15*, 99-119.

Taube, H. (1999). Automatic tonal analysis: Toward the implementation of a music theory workbench. *Computer Music Journal, 23*(4), 18-32.

Temperley, D., & Sleator, D. (1999). Modeling meter and harmony: A preference-rule approach. *Computer Music Journal, 23*(1), 10-27.

Wiggins, G., Miranda, E., Smaill, A., & Harris, M. (1993). A framework for the evaluation of music representation systems. *Computer Music Journal, 17*(3), 31-42.

Winograd, T. (1968). Linguistics and the computer analysis of tonal harmony. *Journal of Music Theory, 12*, 2-49.