

---

# Learning melodic analysis rules

---

**Plácido R. Illescas**      **David Rizo**      **José Manuel Iñesta**  
Department of Software and Computing Systems  
University of Alicante  
placidoroman@gmail.com, {drizo, inesta}@dlsi.ua.es

**Rafael Ramirez**  
Department of Information and Communication Technologies  
Universitat Pompeu Fabra  
rafael.ramirez@upf.edu

## 1 Introduction

Musical analysis is a mean to better understand the composer’s intentions when creating a piece and can be used as an intermediate description of a musical work for other purposes, e.g. expressive performance [4] or music comparison [5]. A musical analysis can be decomposed in melodic, harmonic, and tonal function analyses. Melodic analysis studies the stylistic characteristics of a note from a contrapuntal point of view, while tonal and harmonic analyses investigate chord roles in particular musical pieces.

Automatic musical analysis has been approached from different perspectives: grammars, expert systems, probabilistic models, and model matching have been proposed for implementing tonal analysis (a comprehensive review can be found in [1]).

In this work we focus on automatic melodic analysis. One question that arises when building a melodic analysis system using a-priori music theory is whether it is possible to automatically extract analysis rules from examples, and how similar are those learnt rules compared to music theory rules. This work investigates this question, i.e. given a dataset of analyzed melodies our objective is to automatically learn analysis rules and to compare them with music theory rules.

## 2 Melodic analysis

The data set used in this work consists of the transcriptions, in MusicXML format, of the harmonized chorals from J.S.Bach (BWV-26, 29, 253, 272, 274, 275, 280, 285, 437, and 438). Each of the 2528 notes in the data set is characterised by its interval with the previous and following note, the direction of both intervals, whether they are tied or not, a ratio [3] describing the rhythm based on the Inter-onset Interval ratio, and the position in the bar that determines the stability or metrical strength of the note. After a manual melodic analysis each of the notes has been classified as harmonic tone, passing tone, neighbor tone, appoggiatura, escape note, suspension, anticipation, pedal note. These classes are unbalanced (more than half of notes are tagged as harmonic)<sup>1</sup>.

We have applied the RIPPER algorithm [2] to learn melodic analysis rules that classifies each note as either harmonic, passing, neighbor, appoggiatura, escape, suspension, anticipation, or pedal note. The algorithm grows one rule by greedily adding antecedents to the rule until the rule is 100% accurate and trying every possible value of each attribute and selects the condition with highest information gain. Less prevalent classes are considered first.

---

<sup>1</sup>Note that we are including all four voices of harmonized chorals, and commonly the lower voices just contain harmony

Musicologist analysis	#Rules	Example rule
Strongly agree	1	$prevIntervalDir = EQUAL \wedge duration \leq 0.5 \wedge \neg tied \wedge nextIntervalDir = DESC \wedge instability \leq 3 \wedge nextInterval \leq 2 \wedge nextIntervalMode = MINOR \wedge ratio \leq 0.5 \rightarrow melodictag = appoggiatura$
Agree	20	$instability \geq 5 \wedge nextIntervalDir = ASCENDING \wedge prevIntervalDir = DESCENDING \wedge prevInterval \leq 2 \wedge nextIntervalMode = MAJOR \rightarrow melodictag = neighbor$
Neither agree nor disagree	15	$tied \wedge nextIntervalMode = MAJOR \wedge duration \geq 2 \rightarrow melodictag = suspension$
Disagree	3	$instability \geq 5 \wedge instability \geq 13 \wedge instability \leq 13 \wedge prevIntervalDir = DESCENDING \wedge nextIntervalMode = MINOR \rightarrow melodictag = neighbor$
Strongly disagree	0	

Table 1: Likert-type scale analysis of rules. The rule that exemplifies the *Agree* entry needs the condition  $nextInterval = 2$  to exactly correspond to music theory knowledge. The rule for *Neither agree nor disagree*, contains irrelevant clauses ( $nextIntervalMode = MAJOR$ ), and on the contrary,  $nextInterval = 2$  is missing and the instability factor should be small. Regarding the *Disagree* rule, it is irrelevant because it is ambiguous.

### 3 Results

The results show that it is possible to obtain accurate melodic analysis models (the obtained accuracy is 92%) and that the learning algorithm is able to extract meaningful rules that can be applied to melodically analyze baroque music. Table 1 shows the Likert-type scale analysis of the rules by a musicologist. From a musicology point of view, just one of the rules corresponds exactly with music theory knowledge, although most of the rules are extremely similar to music theory rules. The difference between the learnt rules and music theory rules is that the former disregard conditions that are common to most notes in the data set, e.g. next intervals of two semitones. Due to the fact that this is the most common interval in the dataset, it is not surprising that the algorithm has not considered this information discriminant. Besides, the algorithm has generated, as expected, groups of rules that could be merged into music theory knowledge.

To conclude, it seems that it is indeed possible to automatically extract analysis rules from examples using machine learning techniques, and that the resulting rules correspond to music theory knowledge. We plan to extend our data set specially with pieces containing a greater proportion of non-harmonic notes. We believe this would improve the generality and accuracy of the melodic analysis model.

### References

- [1] Jérôme Barthelemy. Figured bass and tonality recognition. In *ISMIR*, 2001.
- [2] William W. Cohen. Fast effective rule induction. In *Proc. of the Twelfth International Conference on Machine Learning*, 115-123, 1995.
- [3] Plácido R. Illescas, David Rizo, and José Manuel Iñesta. Harmonic, melodic, and functional automatic analysis. In *Proceedings of the 2007 International Computer Music Conference*, pages 165–168, 2007.
- [4] R. Ramírez, A. Pérez, S. Kersten, D Rizo, P.R. Illescas, and José Manuel Iñesta. Modeling celtic violin expressive performance. In *Proc. Int. Workshop on Machine Learning and Music, MML 2008*, pages 7–8, Helsinki (Finland), 2008.
- [5] David Rizo. *Symbolic music comparison with tree data structures*. PhD thesis, Universidad de Alicante, 2010.