

# Learning Sets of Musical Rules

Rafael Ramirez  
Technology Department, Pompeu Fabra University  
Ocata 1, 08003 Barcelona, Spain  
rramirez@iua.upf.es

## Abstract

*If-then rules are one of the most expressive and intuitive knowledge representations and their application to musical knowledge raises particularly interesting questions. This paper briefly introduces several approaches to learning sets of rules and provides a representative sample of the issues involved in applying such techniques in a musical context. We then proceed to describe our approach to learning rules for the harmonization of popular music melodies.*

## 1. Introduction

In many cases it is useful to learn and represent a target knowledge as a set of if-then rules. The aim of this paper is to introduce different approaches to learning sets of rules focusing on their application to learning musical concepts. In particular we present our experimental investigations on learning rules for the harmonization of popular music melodies.

The reader should be cautioned that this article is not a comprehensive review of rule-learning algorithms or its applications in the field of computer music. Rather, the goal of the paper is to provide a brief introduction to the subject and a representative sample of the issues involved in applying such techniques in a musical context.

The rest of the paper is organized as follows: Section 2 briefly introduces different types of rules used in inductive learning. In section 3, some rule learning systems applied to musical concepts are reported. Section 4 describes our approach to learning harmonization rules in popular music, and finally section 5 presents some conclusions and further work.

## 2. Learning sets of rules

There are basically four different approaches to inductive learning in machine learning applications: *classification learning* is a learning scheme which takes a set of classified *examples*, characterized by the values of its *attributes*, and it is expected to learn a way to classify unseen examples; *associative learning* aims at learning any association among the examples features; *clustering* aims at splitting the set of input examples into groups examples that belong together; and *numeric prediction* is similar to classification learning but here the outcome to be predicted is not a discrete class but a numeric value.

In many cases, it is useful to learn the target knowledge represented as a set of *if-then* rules that together define the knowledge. Classification rules are a popular approach to classification learning in which the *antecedent* of a rule is generally a conjunction of tests (e.g.  $age < 25 \wedge sex = male$ ) and the *consequent* is the predicted class (e.g.  $drive = yes$ ) or classes (possibly with a probability associated to each class). It turns out that a set of classification rules can be directly obtained from one of the most widely used and practical methods for inductive reasoning: *decision trees* [8]. It is possible to read a set of rules from a decision tree by generating a rule for each leaf and making a conjunction of all the tests encountered on the path from the root to that leaf. This produces rules that are unambiguous, i.e. it does not matter in which order they are considered. However, in general rules that are read directly off a decision tree are far more complex than necessary, and are usually pruned to simplify them. When representing knowledge by using rules, each rule seems to represent an independent unit of

knowledge. In many cases, new rules can be added to an existent rule set without disturbing the ones already in the set, whereas the addition to a tree structure may need restructuring the whole tree.

Association rules [3] are similar to classification rules except that they can predict any attribute or combinations of attributes, not just the class. Association rules are not intended to be used together as a set, as classification rules are. Each association rule express different properties about the data set. It is possible to derive a relatively large number of association rules from even a tiny data set, so the set of rules is usually restricted to those that apply to a large number of examples and have a high accuracy on the examples they apply to. It is possible to discover association rules in the same way as classification rules, by executing a divide-and-conquer rule induction procedure, such as a decision tree algorithm, for each possible expression that could occur in the right hand-side of the rule. However, this approach is extremely inefficient and infeasible in the general case: any attribute might occur in the right-hand side with any possible value, and a single association rule can often predict the value of more than one attribute. Thus, to find association rules it would be necessary to execute the rule induction procedure once for every possible combination of attributes, with every possible combination of values on the right-hand side. A more practical approach is to focus our interest on association rules with high *coverage*, i.e. with a high number of instances which they predict correctly. Combinations of attribute-value pairs that have a pre-specified minimum coverage are collected. Once all sets of such pairs are generated, each of them is turned into a rule, or set of rules, with at least the specified minimum *accuracy*, i.e. the number of instances a rule predicts correctly as a proportion of the number of instances that the rule applies to.

It is often assumed implicitly that the conditions in (classification and association) rules involve testing an attribute value against a constant. Such rules are called *propositional* because they have the same expressive power as *propositional logic*. In many cases, propositional rules are sufficiently expressive to describe a concept accurately. However, there are cases where more expressive rules would provide a more intuitive concept description. These are cases where the

knowledge to be learned is best expressed by allowing variables in attributes (e.g.  $age(x) < 25$ ) and thus in rules (e.g. *if sex(y)=female then ...*). One important special case involving learning sets of rules containing variables is called *inductive logic programming* [11, 9]. A type of rules in inductive logic programming, called Horn clauses, are of particular interest because rules in this form can be interpreted and directly executed as a program in the logic programming language Prolog [4]. As an example of sets of rules containing variables, consider the following two rules jointly describing the concept of ancestor ( $parent(x, y)$  indicates that  $y$  is the mother or the father of  $x$  and  $ancestor(x, y)$  indicates that  $y$  is the ancestor of  $x$ ):

*if parent(x,y) then ancestor(x,y)*  
*if parent(x,z)  $\wedge$  ancestor(z,y) then ancestor(x,y)*

This two rules describe a recursive function that would be very difficult to represent using a decision tree or other propositional representation.

### 3. Learning rules in music

Previous research in learning sets of rules in a musical context has included a broad spectrum of music domains. Widmer [12, 13] has focused on the task of discovering general rules of expressive classical piano performance from real performance data via inductive machine learning. The performance data used for the study are MIDI recordings of 13 piano sonatas by W.A. Mozart performed by a skilled pianist. In addition to these data, the music score was also coded. The resulting substantial data consists of some 106,000 performed notes along with information about the nominal note onsets, duration, metrical information and annotations. When trained on the data the inductive rule learning algorithm named PLCG [14] discovered a small set of 17 quite simple classification rules [12] that predict a large number of the note-level choices of the pianist.

Morales has reported research on learning counterpoint rules [7]. The goal of the reported system is to obtain standard counterpoint rules from examples of counterpoint music pieces and basic musical knowledge from traditional music. The system was provided with musical knowledge which includes the classifica-

tion of intervals into consonances and dissonances, the description of whether two notes of different voices form a perfect or imperfect consonance or a dissonance, and whether two notes from the same voice form a valid or invalid interval. The rules learned by the system, resulting in a Prolog program, were tested for analysis of simple counterpoint pieces. The system was also used for musical creation of simple two voice counterpoint pieces: given the first voice of a piece and a sequence of rules, the system was able to generate the required counterpoint notes of the second voice.

Igarashi et al. describe the analysis of respiration during musical performance by inductive logic programming [6]. Using a respiration sensor, respiration during cello performance was measured and rules were extracted from the data together with musical/performance knowledge such as harmonic progression and bowing direction. The data was obtained from four skill cello players by asking each of them to perform the same representative cello piece. Each subject repeated the performance for this piece six times. As background knowledge fifteen kinds of predicates concerned with musical structure and playing styles were defined.

Other inductive logic programming approaches to rule learning and musical analysis of music include [5] and [2]. In [5], Dovey analyzes piano performances of Rachmaniloff pieces using inductive logic programming and extracts rules underlying them. In [2], Van Baelen extended Doveys work and attempted to discover regularities that could be used to generate MIDI information derived from the musical analysis of the piece.

#### 4. Learning harmonization rules in popular music

We describe a simple inductive approach for learning rules from popular music harmonizations. As opposed to most of the existing work on harmonization in computer music which views harmonization as deriving/analyzing a four voice score for a particular voice, our view is on the sequence of chords that harmonize a melody. In this context, the process of harmonization is difficult to formalize and the way a given melody is harmonized normally varies from person to person

according to her taste and background. Thus, the approach presented here makes no attempts at providing a definite set of rules but to learn generic rules from a set of popular music examples.

The data used in our experimental investigations were collected from popular music scores with chord annotations. We used 42 scores mainly of western pop-songs. The data included information about position of a particular bar in the score (initial, middle or last), notes in and harmonization of the bar, and harmonizations of preceding bars (chords assigned to previous four bars). Using this data we applied the C4.5 decision tree learning algorithm [10] and obtained a set of classification rules directly from the decision tree the algorithm generated. We also applied the Apriori rule learning algorithm [1] to induce association rules.

We chose to provide the learning algorithms with harmonic knowledge at the bar level, as opposed to at the note level, in order to capture common chord patterns in popular music. This information would have been lost if we only analyzed the harmonization of melodies at a note level. We also structured our data by musical phrases. We manually segmented the pieces into phrases and provided harmonization knowledge for each segment. This, we believe, is closer to the process of harmonizing a melody by a musician. All the data provided to the learning algorithms was coded by hand which explains the relatively small number of musical scores considered. However, despite of the reduced number of training data some of the rules generated by the learning algorithms turn out to be of musical interest and correspond to intuitive musical knowledge. In order to illustrate the types of rules found let us consider some examples of learned rules:

RULE1:  $pos=last \Rightarrow chord=1$

*“If a bar is the last bar of the song, then it is harmonized with the tonic chord (e.g. in the key of C, it is harmonized with the C chord).”*

RULE2:  $chord1=4 \ \& \ chord2=5 \Rightarrow chord=1$

*“if the penultimate bar and last bar are harmonized with the subdominant and dominant chords respectively, then the current bar is harmonized with the tonic.”*

RULE3: chord=4  $\Rightarrow$  pos=middle

*“No piece starts or ends with a subdominant chord”*

These extremely simple rules turn out to be very general: the first rule predicts 97%, the second rule predicts 78%, and the third rule predicts 100% of all the cases. The learning algorithm used for the first two rules was C4.5 and the last rule was obtained by the Apriori algorithm. A few other interesting rules were discovered which we expect to be the basis of a system for automatically harmonize popular music melodies. These rules and their implementation in a Prolog-based system will be reported in a companion paper.

## 5. Conclusion

This paper describes several approaches to learning sets of rules and provides a representative sample of the issues involved in applying such techniques in a musical context. In particular, we presented our experimental investigations on learning rules for the harmonization of popular music melodies. We applied the C4.5 decision tree learning algorithm and the Apriori learning algorithm to data with information about position of a particular bar in the score, notes in and harmonization of the bar, and harmonizations of preceding bars to induce both classification and association rules.

**Future work:** This paper presents work in progress so there is future work in different directions. The manual segmentation and coding of training data is obviously not scalable so a (semi) automatic method to do this is necessary. We also plan to experiment with different information encoded in the training data. Extending the information in the training data and combining this extended data with background musical knowledge will very likely generate a more complete set of rules. Another issue to be considered is how to implement the rules most efficiently into a system for automatic harmonization. The interesting point here is that such system would be easily extended to include other music styles.

## References

- [1] Agrawal, R.T. (1993). Mining association rules between sets of items in large databases. International Conference on Management of Data, ACM, 207,216.
- [2] Van Baelen, E. and De Raedt, L. (1996). Analysis and Prediction of Piano Performances Using Inductive Logic Programming. International Conference in Inductive Logic Programming, 55-71.
- [3] Chen, M., Jan, J. and Yu, P.S. (1996) Data Mining: An Overview from a Database Perspective. IEEE Trans. Knowledge and Data Engineering 8(6), 866-883.
- [4] Colmenauer A. (1990). An Introduction to PROLOG-III. Communications of the ACM, 33(7).
- [5] Dovey, M.J. (1995). Analysis of Rachmaninoff’s Piano Performances Using Inductive Logic Programming. European Conference on Machine Learning, Springer-Verlag.
- [6] Igarashi, S., Ozaki, T. and Furukawa, K. (2002). Respiration Reflecting Musical Expression: Analysis of Respiration during Musical Performance by Inductive Logic Programming. Proceedings of Second International Conference on Music and Artificial Intelligence, Speinger-Verlag.
- [7] Morales, E. (1997). PAL: A Pattern-Based First-Order Inductive System. Machine Learning, 26, 227-252.
- [8] Quinlan, J.R. (1986) Induction of decision trees. Machine Learning, 1(1), 81-106.
- [9] Quinlan, J.R. (1990) Learning logical definitions from relations. Machine Learning, 5, 239-266.
- [10] Quinlan, J.R. (1993) C4.5: Programs for Machine Learning, San Francisco, Morgan Kaufmann.
- [11] Shapiro, E. (1983) Algorithmic Program Debugging. Cambridge MA, MIT Press.
- [12] Widmer, G. (2002). Machine Discoveries: A Few Simple, Robust Local Expression Principles. Journal of New Music Research 31(1), 37-50.
- [13] Widmer, G. (2002). In Search of the Horowitz Factor: Interim Report on a Musical Discovery Project. Invited paper. In Proceedings of the 5th International Conference on Discovery Science (DS’02), Lbeck, Germany. Berlin: Springer-Verlag.
- [14] Widmer, G. (2001). Discovering Strong Principles of Expressive Music Performance with the PLCG Rule Learning Strategy. Proceedings of the 12th European Conference on Machine Learning (ECML’01), Freiburg, Germany. Berlin: Springer Verlag.