Rule induction for expressive music performance modeling

Rafael Ramirez, Amaury Hazan

Music Technology Group, Pompeu Fabra University
Ocata 1, 08003 Barcelona, Spain, Tel:+34 935422165, Fax:+34 935422202
{rafael,ahazan}@iua.upf.es

Abstract. Modeling expressive music performance is one of the most challenging aspects of computer music. In this paper we investigate the use of rule induction methods for mining monophonic Jazz standards recordings by a skilled saxophone player. In particular, we propose a rule induction algorithm which produces good prediction accuracy while allowing a straightforward interpretation of the prediction model. We implement a tool for automatic expressive performance transformations of Jazz melodies using the induced model.

1 Introduction

Expressive performance is an important issue in music which has been studied from different perspectives (e.g. [15, 8, 3]). The main approaches to empirically study expressive performance have been based on statistical analysis (e.g. [14]), mathematical modelling (e.g. [19]), and analysis-by-synthesis (e.g. [7]). In all these approaches, it is a person who is responsible for devising a theory or mathematical model which captures different aspects of musical expressive performance. The theory or model is later tested on real performance data in order to determine its accuracy.

In this paper we describe an approach to investigate musical expressive performance based on inductive rule learning. Instead of manually modelling expressive performance and testing the model on real musical data, we let a computer use rule learning techniques to automatically discover regularities and performance principles from real performance data (i.e. standard Jazz example performances).

Methods for rule induction generate rules that can either be treated as ordered or unordered. In the later case, some mechanism has to be adopted for resolving conflicts among intersecting rules. Previous approaches to resolve intersecting rule conflicts include calculating class probabilities based on the union [4] or the intersection [11] of examples covered by the overlapping rules, or using naive Bayes probability. Furthermore, it is possible that there is no applicable rule when trying to classify and example. In this case, a common strategy is to classify an example as belonging to the majority class [10]. Another strategy reported in [6] is to minimally generalize the rules to include the uncovered examples.

In this paper, we focus on the problem of how to handle both overlapping rules and the absence of applicable rules when trying to classify examples in the domain of expressive music performance. We combine a greedy set cover algorithm with well-established rule learning methods, namely simple classification trees, classification trees with bagging/boosting and voting using some of these. We also explore k-nearest neighbor and SVM.

The rest of the paper is organized as follows: Section 2 describes the learning scheme we use in this paper. Section 3 describes our approach to model expressive music performance. In Section 4 related work is reported, and finally Section 5 presents some conclusions and indicates some areas of future research.

2 Learning algorithm

We are interested in finding descriptive models of categories such as for example when a performer lengthens or shortens the duration of a note. The model should characterize classes of situations which are treated similarly and the descriptions should preferably be simple (in order to be understood by a human). We do not expect the model to cover and describe all of (or even a large number of) the instances of a given class. We are interested in rule models that capture only a part of the observations but describe these in meaningful terms and differentiate them from observations belonging to other classes reasonably well.

In this context, we propose a learning scheme that is geared towards inducing sets of rules which capture general regularities in the observations corresponding to general expressive performance principles, leaving out all observations which represent *exceptions* to these general principles (exceptions in the sense that either no rule applies to the observation, or the observation generates a conflict among intersecting rules). We aim at finding simple and robust classification rules in a complex and noisy data set. The learning scheme we propose can be applied to arbitrary domains. However it has been applied only in the context of our musical research. It works as follows:

1. Apply a greedy set cover algorithm to the data set in order to induce first-order rules.
2. Collect all instances which are not covered by any of the induced rules.
3. Collect also the instances with multiple classification, i.e. instances covered by more than one overlapping rule.
4. Apply standard classification techniques to the union of 2 and 3. Among the techniques we have explored are simple classification trees, classification trees with bagging/boosting, k-nearest neighbor, SVM and voting using some of these.
5. Add the resulting model from 4 to the original first-order rules model.
6. When classifying an example, apply the extended model as follows: initially apply the first-order rule model and only apply the model resulting from 4 to handle rule overlapping or absence of applicable rules.

The goal of this procedure is to obtain a set of intelligible rules characterizing the general principles of expressive music performance, and at the same time still be able to classify reasonably accurately the exception examples which cannot be covered by the general rules. Our approach contrasts with previous approaches to resolve intersecting rule conflicts (e.g. [4, 11]) and the absence of applicable rules (e.g. [10, 6]) in that we are not interested in extending or reusing the induced rules in order to handle the problematic examples. In the context of expressive music performance, we consider these problematic examples as *exceptions* which require to be treated independently from rules representing general principles. This is achieved via the process of learning first-order logic rules which capture general principles (step 1 above), identifying the examples (exceptions) which do not represent general principles (steps 2 and 3), apply a classification algorithm targeting this smaller set of examples, i.e. exceptions (step 4), and enriching the initial set of rules by adding the classification criteria for the exceptions (step 5). In step 1 we use Aleph's induce_max set covering algorithm [17] for inducing first-order rules. For the classification algorithms in step 4 we use the Waikato Environment for Knowledge Analysis [23].

3 Modeling expressive music performance

The research reported in this paper is concerned with learning expressive performance rules from Jazz standards performances by a skilled saxophone player. Our aim is to find note-level rules which predict, for a significant number of cases, how a particular note in a particular context should be played (e.g. longer than its nominal duration). We are aware of the fact that not all the expressive transformations regarding tempo (or any other aspect) performed by a musician can be predicted at a local note level. Musicians perform music considering a number of abstract structures (e.g. musical phrases) which makes expressive performance a multi-level phenomenon. In this context, our ultimate aim is to obtain an integrated model of expressive performance which combines note-level rules with structure-level rules. Thus, the work presented in this paper may be seen as a starting point towards this ultimate aim.

The training data used in our experimental investigations are monophonic recordings (i.e. recordings composed of one note at a time) of four Jazz standards (*Body and Soul, Once I loved, Like Someone in Love* and *Up Jumped Spring*) performed by a professional musician at 11 different tempos. In order to discover expressive performance regularities at different tempos we divided the recordings into three groups: nominal, slow and fast. The recordings in the nominal group are performed at the piece nominal tempo (+/- 15%) while the recordings in the slow and fast groups are respectively performed slower or faster than the ones in the nominal group. Sound analysis and synthesis techniques based on spectral models are used for extracting high-level symbolic features from the recordings, transforming them and synthesizing a modified recording. The sound spectral model analysis techniques are based on decomposing the original signal into sinusoids plus spectral residual. From the sinusoids of a monophonic signal it is possible to extract information on note pitch, onset, duration, attack and energy, among other high-level information. This information can be modified and the result added back to the spectral representation without loss of quality. We use the software SMSTools [1] which is an ideal tool for preprocessing the signal and providing a high-level description of the audio recordings, as well as for generating an expressive audio according to the transformations obtained by machine learning methods.

After the extraction of high-level symbolic features from the recordings, each note in the training data is annotated with its corresponding class, i.e. *lengthen, shorten* or *same* (see next paragraph for details), and a number of attributes representing both properties of the note itself and some aspects of the local context in which the note appears. Information about intrinsic properties of the note include the note duration and the note metrical position, while information about its context include the note Narmour group(s) [13], duration of previous and following notes, and extension and direction of the intervals between the note and the previous and following notes. This information is provided to the rule learning algorithm as background knowledge.

In this paper, we are concerned with note-level expressive transformations, in particular transformations of note duration, onset and energy (in this paper we only report on results on note duration). For classification, the performance classes we are interested in are *lengthen, shorten* and *same*. A note is considered to belong to class *same* if it is performed within 20% of its nominal duration, i.e. its duration according to the score. A note is considered to belong to class *lengthen* if its performed duration is 20% or more longer that its nominal duration. Class *shorten* is defined analogously. We decided to set the duration boundary to 20% after experimenting with smaller ratios. The idea was to guarantee that a note classified, for instance, as lengthen was purposely lengthened by the performer and not the result of an performance inexactitude.

Using this data we applied the learning scheme described in Section 2. Despite the relatively small amount of training data some of the rules generated by the learning algorithm 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 note-duration rules (the complete set of clauses is composed of 30 clauses for lengthen, 175 clauses for same and 40 clauses for shorten):

RULE1:
```
duration(X,lengthen) :-
        succ(Y,X),
        melo(Y,4,-1,0,1,2,1,nominal).
```

"Playing at a nominal tempo, lengthen the duration of a note if the previous note has the same duration (a quarter) and both notes are in a weak (off-beat) metrical position."

RULE2: `duration(X,lengthen) :-`
```
          succ(Y,X),
          melo(Y,6,2,0,-1,-1,3,slow).
```

"Playing at a slow tempo, lengthen the duration of a note if the previous note is in a strong metrical position and has the same duration (quarter triplet)."

RULE3: `duration(X,shorten) :-`
```
          melo(X,4,0,0,2,-1,2,slow),
          succ(Y, X),
          melo(Y,4,-1,0,-1,-1,3,slow).
```

"Playing at a slow tempo, shorten the duration of a note in a medium strength metrical position (2nd or 4th beat of the bar) if the previous note has the same duration (a quarter)"

RULE4: `duration(X,same) :-`
```
          melo(X,4,0,0,1,-1,2,nominal).
```

"Playing at a nominal tempo, do not alter the duration of a note appearing in a medium strength metrical position (2nd or 4th beat of the bar) if the previous and next note have the same duration (a quarter)".

RULE5: `duration(X, same) :-`
```
          melo(X,8,_,0,_,1,1,slow).
```

"Playing at a slow tempo, do not alter the duration of a note in a weak (off-beat) metrical position if both the current note and the following note are eights".

RULE6: `duration(X, lengthen) :-`
```
          succ(X, D), succ(D, E),
          narmour(E, [F|G]), narmour(X, [F|G]).
```

"lengthen a note N if N and N+2 have the same narmour context".

The complete set of rules has an accuracy of 85%. We consider a data set of 1936 positive examples and 3872 negative examples. Among the positive examples, there are 208 examples which we consider as *exceptions*, i.e. they are covered by none or more than one of the induced first-order rules. These examples are collected and classified by different algorithms. For each of the algorithms considered, we have performed a 10-fold cross validation. Table 1 presents some results obtained from this process. All the tests were performed using the Waikato Environment for Knowledge Analysis [23] (C.C.I refers to the correctly classified instances rate, M.A.E to the mean absolute error, R.M.S.E to the Root Mean Squared Error).

Algorithm	C.C.I(%)	M.A.E	R.M.S.E
KNN	63.70	0.27	0.44
KNN with boosting	62.22	0.27	0.44
KNN with bagging	60.74	0.28	0.42
SMO (1)	59.25	0.34	0.43
SMO (2)	60.37	0.34	0.43
SMO (3)	62.96	0.33	0.42
SMO (3) with Boosting	60.74	0.28	0.43
SMO (3) with Bagging	62.96	0.34	0.41
C4.5	64.07	0.32	0.41
C4.5 with Boosting	62.59	0.32	0.40
C4.5 with Bagging	65.55	0.32	0.40
Voting	64.44	0.31	0.40

Table 1. Cross validation results of auxiliary classification models for onset duration. Support Vector Machine algorithms are annotated depending of the kernel function they use: (1) Linear kernel, (2) Exponent 2 polynomial kernel, (3) Exponent 3 polynomial kernel. The voting meta algorithm combines the results of the following models: KNN, C4.5, and Support Vector Machine (3).

Synthesis tool. We have explored different possible discretization schemes. In particular we have discretized the duration values space in 9 classes according to the degree of transformation. This is, we have defined 4 classes for lengthen and 4 classes for shorten (one for same) for different degrees of lengthening and shortening. In this way, we have obtained a set of finer-grained rules which, in addition to explaining expressive performances principles in more detail, may be also used to generate expressive performances. We have implemented a tool which transforms an inexpressive melody input into an expressive one following the induced model. The tool can either generate an expressive MIDI performance from an inexpressive MIDI description of a melody, or generate an expressive audio file from an inexpressive audio file.

4 Related work

Previous research in learning sets of rules in a musical context has included a broad spectrum of music domains. The most related work to the research presented in this paper is the work by Widmer [20, 21]. Widmer 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 information about the nominal note onsets, duration, metrical information and annotations. When trained on the data the inductive rule learning algorithm named PLCG [22] discovered a small set of 17 quite simple classification rules [20] that predict a large number of the note-level choices of the pianist. In the recordings the tempo of a performed piece is not constant (as it is in our case). In fact, of special interest to them are the tempo transformations throughout a musical piece.

Other inductive machine learning approaches to rule learning in music and musical analysis include [5], [2], [12] and [9]. In [5], Dovey analyzes piano performances of Rachmaniloff pieces using inductive logic programming and extracts rules underlying them. In [2], Van Baelen extended Dovey's work and attempted to discover regularities that could be used to generate MIDI information derived from the musical analysis of the piece. In [12], Morales reports research on learning counterpoint rules. 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. In [9], Igarashi et al. describe the analysis of respiration during musical performance by inductive logic programming. 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.

Tobudic et al. [18] describe a relational instance-based approach to the problem of learning to apply expressive tempo and dynamics variations to a piece of classical music, at different levels of the phrase hierarchy. The different phrases of a piece and the relations among them are represented in first-order logic. The description of the musical scores through predicates (e.g. `contains(ph1,ph2)`) provides the background knowledge. The training examples are encoded by another predicate whose arguments encode information about the way the phrase was played by the musician. Their learning algorithm recognizes similar phrases from the training set and applies their expressive patterns to a new piece.

5 Conclusion

We have investigated the use of rule induction methods for mining monophonic Jazz standards recordings by a skilled saxophone player. In particular, we propose a learning scheme which produces good prediction accuracy while allowing a straightforward interpretation of the prediction model. Focusing on the problem of how to handle both overlapping rules and the absence of applicable rules when trying to classify examples, we have combined a greedy set cover algorithm with well established rule learning methods. Using an induced model, we have implemented a tool for automatic expressive performance transformations of Jazz melodies.

Future work: There is future work in different directions. We plan to experimentally compare with other work on conflict resolution and uncovered examples, as well as explore the use of first-order rule learners for classifying the problematic examples, i.e. the exceptions. We are currently extending our system by considering note onset and note energy expressive transformations. In the future, we plan to increase the amount of training data as well as experiment with different information encoded in it (e.g. ornamentations, vibrato). Increasing the training data, extending the information in it and combining it with background musical knowledge will certainly generate a more complete model. As mentioned earlier, we intend to incorporate structure-level information to obtain an integrated model of expressive performance which combines note-level knowledge with structure-level knowledge.

Acknowledgments: This work is supported by the Spanish TIC project ProMusic (TIC 2003-07776-C02-01). We would like to thank Emilia Gomez, Esteban Maestre and Maarten Grachten for providing and processing the data, as well as the anonymous reviewers for their insightful comments and pointers to related work.

References

1. Amatriain, X. Bonada, J. Loscos, A. Arcos, J. Verfaille, V. (2003). 'Content-based Transformations' Journal of New Music Research Vol.32 .1
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. Bresin, R. (2000). Virtual Visrtuosity: Studies in Automatic Music Performance. PhD Thesis, KTH, Sweden.
4. Clark, P. and Boswell, R. (1991). Rule induction with CN2: some recent improvements. In Proc. European Working Session on Learning, Springer.
5. Dovey, M.J. (1995). Analysis of Rachmaninoff's Piano Performances Using Inductive Logic Programming. European Conference on Machine Learning, Springer-Verlag.
6. Eineborg M. and Bostrm H. 2001. Classifying Uncovered Examples by Rule Stretching, Proceedings of the Eleventh International Conference on Inductive Logic Programming, LNCS 2157, Springer.
7. Friberg, A. (1995). A Quantitative Rule System for Musical Performance. PhD Thesis, KTH, Sweden.
8. Gabrielsson, A. (1999). The performance of Music. In D.Deutsch (Ed.), The Psychology of Music (2nd ed.) Academic Press.
9. 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, Springer-Verlag.
10. Van Laer, W., De Raedt, L. and Dzeroski, S. (1997). On multi-class problems and discretization in inductive logic programming. In Proc. International Symposium on Methodologies for Intelligent Systems. Springer-Verlag.
11. Lindgren T. and Bostrom H. (2002). Classification with Intersecting Rules, Proc. of 13th International Conference on Algorithmic Learning Theory, Springer.
12. Morales, E. (1997). PAL: A Pattern-Based First-Order Inductive System. Machine Learning, 26, 227-252.
13. Narmour, E. (1990). The Analysis and Cognition of Basic Melodic Structures: The Implication Realization Model. University of Chicago Press.
14. Repp, B.H. (1992). Diversity and Commonality in Music Performance: an Analysis of Timing Microstructure in Schumann's 'Traumerei'. Journal of the Acoustical Society of America 104.

15. Seashore, C.E. (ed.) (1936). Objective Analysis of Music Performance. University of Iowa Press.
16. Shapiro, E. (1983). Algorithmic Program Debugging. Cambridge MA, MIT Press.
17. Srinivasan, A. (2001). The Aleph Manual.
18. Tobudic A., Widmer G. (2003). Relational IBL in Music with a New Structural Similarity Measure, Proceedings of the International Conference on Inductive Logic Programming, Springer Verlag.
19. Todd, N. (1992). The Dynamics of Dynamics: a Model of Musical Expression. Journal of the Acoustical Society of America 91.
20. Widmer, G. (2002). Machine Discoveries: A Few Simple, Robust Local Expression Principles. Journal of New Music Research 31(1), 37-50.
21. 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), Lubeck, Germany. Berlin: Springer-Verlag.
22. 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.
23. Witten, I.H. and Frank, E. (1999). Data Mining, Practical Machine Learning Tools and Techniques with Java Implementation, Morgan Kaufmann Publishers.