Modeling Expressive Music Performance in Bassoon Audio Recordings

Rafael Ramirez, Emilia Gomez, Veronica Vicente, Montserrat Puiggros, Amaury Hazan, and Esteban Maestre

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

Abstract. In this paper, we describe an approach to inducing an expressive music performance model from a set of audio recordings of XVIII century bassoon pieces. We use a melodic transcription system which extracts a set of acoustic features from the recordings producing a melodic representation of the expressive performance played by the musician. We apply a machine learning techniques to this representation in order to induce a model of expressive performance. We use the model for both understanding and generating expressive music performances.

1 Introduction

Expressive performance is an important issue in music which has been studied from different perspectives (e.g. [2]). The main approaches to empirically study expressive performance have been based on statistical analysis (e.g. [11]), mathematical modelling (e.g. [13]), and analysis-by-synthesis (e.g. [1]). 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 machine learning [7]. Instead of manually modelling expressive performance and testing the model on real musical data, we let a computer use an inductive logic programming algorithm to automatically discover regularities and performance principles from real performance data (i.e. bassoon audio performances).

The rest of the paper is organized as follows: Section 2 describes how the acoustic features are extracted from the monophonic recordings. In Section 3 our approach for learning rules of expressive music performance is described. Section 4 reports on related work, and finally Section 5 presents some conclusions and indicates some areas of future research.

D.-S. Huang, K. Li, and G.W. Irwin (Eds.): ICIC 2006, LNCIS 345, pp. 951–957, 2006.

2 Melodic Description

In order to obtain a symbolic description of the expressive audio recordings we compute descriptors related to two different temporal scopes: some of them related to an analysis frame, and some other features related to a note segment. Firstly, we deivide the audio signal into analysis frames, and a set of low-level descriptors are computed for each analysis frame. Then, we perform a note segmentation using low-level descriptor values. Once the note boundaries are known, the note descriptors are computed from the low-level and the fundamental frequency values.

The main low-level descriptors we use to characterize expressive performance are instantaneous energy and fundamental frequency. Energy is computed on the spectral domain, using the values of the amplitude spectrum. For the estimation of the instantaneous fundamental frequency we use a harmonic matching model, the Two-Way Mismatch procedure (TWM) [5]. First of all, we perform a spectral analysis of a portion of sound, called analysis frame. Secondly, the prominent spectral peaks of the spectrum are detected from the spectrum magnitude. These spectral peaks of the spectrum are defined as the local maxima of the spectrum which magnitude is greater than a threshold. These spectral peaks are compared to a harmonic series and an TWM error is computed for each fundamental frequency candidates. The candidate with the minimum error is chosen to be the fundamental frequency estimate.

Note segmentation is performed using a set of frame descriptors, which are energy computation in different frequency bands and fundamental frequency. Energy onsets are first detected following a band-wise algorithm that uses some psycho-acoustical knowledge [3]. In a second step, fundamental frequency transitions are also detected. Finally, both results are merged to obtain the note boundaries.

We compute note descriptors using the note boundaries and the low-level descriptors values. The low-level descriptors associated to a note segment are computed by averaging the frame values within this note segment. Pitch histograms have been used to compute the pitch note and the fundamental frequency that represents each note segment, as found in [6]. This is done to avoid taking into account mistaken frames in the fundamental frequency mean computation.

3 Learning the Expressive Performance Model

In this section, we describe our inductive approach for learning an expressive performance model from audio performances of bassoon pieces. 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 of 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 audio recordings of XVIII century bassoon pieces performed by a professional musician. Each piece has been recorded at 3 different tempos: for pieces indicated as *adagio* the recorded tempos are 50, 60, 100 ppm, for pieces indicated as *allegro moderato* and *affectuoso* the recorded tempos are 60, 92, 120 ppm.

In this paper, we are concerned with expressive transformations of note duration, onset, energy and trills. The note-level performance classes which interest us are: *lengthen, samedur* and shorten for note duration, advance, ontime and delay for note onset, louder, medium and *softer* for note energy, and *few*, *average* and *many* for a trilled note. A note is considered to belong to class *lengthen* if its performed duration is 20% or more longer that its nominal duration, e.g. its duration according to the score. Class *shorten* is defined analogously. A note is considered to be in class *advance* if its performed onset is 5% of a bar earlier (or more) than its nominal onset. Class *delay* is defined analogously. A note is considered to be in class *louder* if it is played louder than its predecesor and louder then the average level of the piece. Class *softer* is defined analogously. Finally, a note is considered to be in class *few*, *average* or *many* if the number of trills is less tan 4, between 5 and 9, or more than 10, respectively. For synthesizing trills, we apply a nearest neighbor algorithm which selects the most similar trill (in terms of musical context) in the training examples and adapts it to the new musical context (e.g. the key of the piece).

Each note in the training data is annotated with its corresponding class 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 includes the note's duration, pitch and metrical position, while information about its context includes the duration of previous and following notes, extension and direction of the intervals between the note and both the previous and the subsequent note, the note Narmour groups [8], and tempo of the performance.

Using this data, we apply a greedy set covering algorithm in order to induce an expressive performance model. We obtain an ordered set of first-order rules each of which chharacterises a subset of the training data. We define four predicates to be learned: `duration/4`, `onset/4`, `energy/4`, and `trills/4`. For each note of our training set, each predicate corresponds to a particular type of transformation: `duration/4` refers to duration transformation, `onset/4` to onset deviation, `energy/4` to energy transformation, and `alteration/4` refers to note alteration. For each predicate we use the complete training set and consider a background knowledge containing the note's local information (`context/6` predicate) and the Narmour structures (`narmour/2` predicate), as well as predicates for specifying an arbitrary-size context (i.e. any number of successors and predecessors) of a note (`succ/2` predicate), and auxiliary predicates (e.g. `member/3`). Once we obtain a set of rules for a particular concept, e.g. duration, we collect the ex-

amples correctly covered by each rule and apply a linear regression on the their numerical values. The numerical values of the covered examples are aproximated by a linear regerssion in the same way as a *model tree* approximates examples at its leaves. The difference with a model tree is that the induced rules do not form a tree as it is the case in model trees. The algorithm is as follows:

```
SEQ-COVERING(Target_attribute,Attributes,Examples,Threshold)
   Learned_classification_rules := {}
   Learned_regression_rules := {}
   Rule := LEARN-ONE-RULE(Target_attribute, Attributes, Examples)
   while PERFORMANCE(Rule, Examples) > Threshold do
      Learned_classification_rules := Learned_classification_rules + Rule
      Examples := Examples - {examples correctly classified by Rule}
      Rule := LEARN-ONE-RULE(Target_attribute, Attributes, Examples)
   For each Rule in Learned_classification_rules do
      collect correctly covered examples by the Rule
      approximate the examples' numerical value by linear regression LR
      Construct Rule_1 as:
         body(Rule_1) := body(Rule)
         head(Rule_1) := LR
      Learned_regression_rules := Learned_regression_rules + Rule_1
   Return Learned_regression_rules
```

`SEQ-COVERING` learns rules until it can no longer learn a rule whose performance is above the given `Threshold`. The `LEARN-ONE-RULE` subroutine generates one rule by performing a general-to-specific search through the space of possible rules in search of a rule with high accuracy. It organises the hypothesis space search in the same general fashion as the CN2 algorithm mantaining a list of k best candidates at each step. In order to handle three classes (e.g. in the case of note duration, lengthen, shorten and same) we have forced the `LEARN-ONE-RULE` subroutine to learn rules that cover positive examples of one class only. Initially, it learns rules that cover positive examples of one of the classes (e.g. lengthen) and considers the examples of the other two classes (e.g. shorten and same) as negative examples. Once the rules for the first class have been learned, `LEARN-ONE-RULE` learns rules that cover only positive examples of a second class (e.g. shorten) in the same way it did for the first class, and similarly for the third class. The `PERFORMANCE` procedure computes the function $tp^{\alpha}/(tp + fp)$ where tp is the number of true positives, fp is the number of false positives and α is a parameter which provides a trade-off between the rule's accuracy and coverage. For each type of rule, depending on the exact number of positive examples, we tuned both the parameter α and the `Threshold` to constrain the minimum number of positive examples as well as the ratio of positive and negative examples covered by the rule. This is, using α and `Threshold` we restrict the area in the coverage space [1] in which the induced rules must lie.

Inductive logic programming has proved to be an extremely well suited technique for learning expressive performance rules. This is mainly due to three reasons: Firstly, inductive logic programming allows the induction of first order logic rules. First order logic rules are substantially more expressive than the traditional propositional rules used in most rule learning algorithms (e.g. the widely used C4.5 algorithm [9]) which allows specifying musical knowledge in a more

[1] Coverage spaces are ROC spaces based on absolute numbers of covered examples.

natural manner. Secondly, Inductive logic programming allows considering an arbitrary-size note context without explicitly defining extra attributes. Finally, the possibility of introducing background knowledge into the learning task provides great advantages in learning musical concepts where often there is a great amount of available background information (i.e. music theory knowledge).

Synthesis Tool. We have implemented a tool which transforms an inexpressive melody input into an expressive one following the induced model tree. 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

Widmer [14,15] reported 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 an inductive rule learning algorithm discovered a small set of quite simple classification rules [14] that predict a large number of the note-level choices of the pianist.

Tobudic et al. [12] 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.

Ramirez [10] et al report on a system capable of generating audio expressive saxophone performances of Jazz standards. The system is based on a similar approach to the one presented here, where different acoustic features of real saxophone Jazz performances are extracted and used to induce an expressive performance model.

Lopez de Mantaras et al report on SaxEx [4], a performance system capable of generating expressive solo performances in jazz. Their system is based on case-based reasoning, a type of analogical reasoning where problems are solved by reusing the solutions of similar, previously solved problems. In order to generate expressive solo performances, the case-based reasoning system retrieve, from a memory containing expressive interpretations, those notes that are *similar* to the input inexpressive notes. The case memory contains information about metrical strength, note duration, and so on, and uses this information to retrieve the appropriate notes.

5 Conclusion

This paper describes an inductive logic programming approach for learning an expressive performance model from recordings of XVIII century bassoon pieces by a professional musician. With this aim, we have extracted a set of acoustic features from the recordings resulting in a symbolic representation of the performed pieces and then applied a rule-based algorithm to the symbolic data and information about the context in which the data appeared. In this context, the algorithm has proved to be an extremely well suited technique for learning an expressive performance model. It naturally allows background knowledge (i.e. musical theory knowledge) to play an important role in the learning process, and permits considering an arbitrary-size note context without explicitly defining extra attributes for each context extension. Currently, we are in the process of increasing the amount of training data as well as experiment with different information encoded in it. Increasing the training data, extending the information in it and combining it with background musical knowledge will certainly generate a more complete set of rules.

Acknowledgments. This work is supported by the Spanish TIC project PromMusic (TIC 2003-07776-C02-01).

References

1. Friberg, A.: A Quantitative Rule System for Musical Performance. PhD Thesis, KTH, Sweden,(1995)
2. Gabrielsson, A. The Performance of Music. In D.Deutsch (Ed.), The Psychology of Music (2nd ed.) Academic Press.(1999) *few*, *average* or *many*
3. Klapuri, A.: Sound Onset Detection by Applying Psychoacoustic Knowledge, Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP.(1999)
4. Lopez de Mantaras, R. and Arcos, J.L. AI and music, from composition to expressive performance, AI Magazine, 23-3.(2002)
5. Maher, R.C. and Beauchamp, J.W. Fundamental frequency estimation of musical signals using a two-way mismatch procedure, Journal of the Acoustic Society of America, vol. 95(1994)2254-2263
6. McNab, R.J., Smith Ll. A. and Witten I.H., Signal Processing for Melody Transcription,SIG working paper, vol. (1996)95-22
7. Mitchell, T.M.: Machine Learning. McGraw-Hill.(1997)
8. Narmour, E.: The Analysis and Cognition of Basic Melodic Structures: The Implication Realization Model. University of Chicago Press.(1990)
9. Quinlan, J.R. C4.5: Programs for Machine Learning, San Francisco, Morgan Kaufmann. (1993)
10. Ramirez, R. Hazan, A. Gómez, E. Maestre, E.: A Machine Learning Approach to Expressive Performance in Jazz Standards MDM/KDD'04, Seattle, WA, USA.(2004)
11. Repp, B.H.: Diversity and Commonality in Music Performance: an Analysis of Timing Microstructure in Schumann's 'Traumerei'. Journal of the Acoustical Society of America 104.(1992)

12. Tobudic A., Widmer G.: Relational IBL in Music with a New Structural Similarity Measure, Proceedings of the International Conference on Inductive Logic Programming, Springer Verlag.(2003)
13. Todd, N.: The Dynamics of Dynamics: a Model of Musical Expression. Journal of the Acoustical Society of America 91.(1992)
14. Widmer, G. Machine Discoveries: A Few Simple, Robust Local Expression Principles. Journal of New Music Research 31(1), (2002)37-50
15. Widmer, G.: 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.(2002)