UNDERSTANDING EXPRESSIVE TRANSFORMATIONS IN SAXOPHONE JAZZ PERFORMANCES USING INDUCTIVE MACHINE LEARNING

Rafael Ramirez, Amaury Hazan, Emilia Gómez, Esteban Maestre
Music Technology Group, Pompeu Fabra University
Ocata 1, 08003 Barcelona, Spain, Tel:+34 935422165, Fax:+34 935422202
{rafael,ahazan,egomez,emaestre}@iua.upf.es

ABSTRACT

In this paper, we describe an approach to learning expressive performance rules from monophonic Jazz standards recordings by a skilled saxophonist. We have first developed 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 machine learning techniques to this representation in order to induce rules of expressive music performance. It turns out that some of the induced rules represent extremely simple principles which are surprisingly general.

1. INTRODUCTION

Expressive performance is an important issue in music which has been studied from different perspectives (e.g. [5]). The main approaches to empirically study expressive performance have been based on statical analysis (e.g. [16]), mathematical modelling (e.g. [17]), and analysis-by-synthesis (e.g. [4]). 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 machine learning. Instead of manually modelling expressive performance and testing the model on real musical data, we let a computer use machine learning techniques [13] to automatically discover regularities and performance principles from real performance data (i.e. standard Jazz example 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.

2. MELODIC DESCRIPTION

In this section, we summarize how the melodic description is extracted from the monophonic recordings. This melodic description has already been used to characterize monophonic recordings for expressive tempo transformations using CBR [8]. We refer to this paper for a more detailed explanation.

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. All the descriptors are stored into a XML document. A detailed explanation about the description scheme can be found in [7].

The procedure for description computation is the following one. First, the audio signal is divided 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. We refer to [6, 8] for details about the algorithms.

2.1. Low-level descriptors computation

The main low-level descriptors used 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) [11].

2.2. Note segmentation

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 [10]. In a second step, fundamental frequency transitions are also detected. Finally, both results are merged to find the note boundaries.

2.3. Note descriptor computation

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 [12].

2.4. Implementation

All the algorithms for melodic description have been implemented within the CLAM framework [1]. They have been integrated within a tool for melodic description, *Melodia*. This tool is available under GPL license.

3. LEARNING EXPRESSIVE PERFORMANCE RULES IN JAZZ

In this section, we describe our inductive approach for learning expressive performance rules from Jazz standards performances by a skilled saxophone player. Our aim is to find 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) or how a melody should be altered by inserting or deleting notes. 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 aim is to obtain an integrated model of expressive performance which combines note-level rules with structure-level rules. The work presented in this paper may be seen as a starting point towards this ultimate aim. At the note-level, we consider for each note its nominal duration, the duration of previous and following notes, the extension of the pitch intervals between the note and the previous and following notes, and the tempo at which it is played. As a starting point, at the structure-level we consider notes as belonging to some basic melodic units based on Narmour I/R structures [22]. A note often belongs to more than one unit appearing in a different position in each of them. Thus, a description of a melodic phrase at this level consists of a list of overlapping Narmour units.

In [8], a parser for melodies that automatically generates I/R analyses was developed. Each Narmour group in Figure 1 represents a class of note sequences patterns. The patterns shown in Figure 1 are the prototypical patterns. For instance, the Narmour group P represents sequences of three or more ascending (or descending) notes in a regular interval. On the other hand, some of these units, e.g. P or D Narmour group, do not necessarily have a fixed number of notes. Figure 2 shows a sample analysis of a melody fragment.

In this paper, we are concerned with note-level expressive transformations, in particular transformations of note duration, onset, energy, and alterations. The note-level performance classes we are interested in are *lengthen*, *shorten* and *same* for duration transformation, *advance*, *delay*, and *same* for onset deviation, *soft*, *loud* and *same* for energy, and *consolidation*, *ornamentation* and *none* for note alteration. 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 *loud* if it is played louder than its predecessor and louder then the average level of the piece. Class *soft* is defined analogously. The last type of note transformations refer to an alteration of the score melody by adding or deleting notes. These transformations play a fundamental role of jazz interpretation and can not be considered as errors as in classical music performance analysis. These transformations may be categorized as follows:

Consolidation represents the agglomeration of multiple score notes into a single performed note.

Fragmentation represents the performance of a single score note as multiple notes

Ornamentation represents the insertion of one or several short notes to anticipate another performed note.
In our dataset the number of fragmentation examples is insufficient to be able to obtain reliable generalization rules. Hence, in the case of nore alteration we defined only classes *consolidation*, *ornamentation* and *none*.
Dataset. The training data used in our experimental investigations are monophonic recordings 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 around the nominal tempo. For each piece, the nominal tempo was determined by the musician as the most natural and comfortable tempo to interpret the piece. Also for each piece, the musician identified the fastest and slowest tempos at which a piece could be reasonably interpreted. Interpretations were recorded at regular intervals around the nominal tempo (5 faster and 5 slower) within the fastest-slowest tempo limits. The dataset is composed of 1936 performed notes. 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 the note include note duration and the note metrical position within a bar, while information about its melodic context include information on neighboring notes as well as the Narmour

[1] http://www.iua.upf.es/mtg/clam

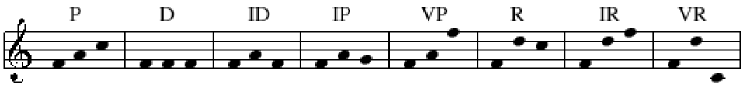

Figure 1. Basic Narmour I/R melodic units

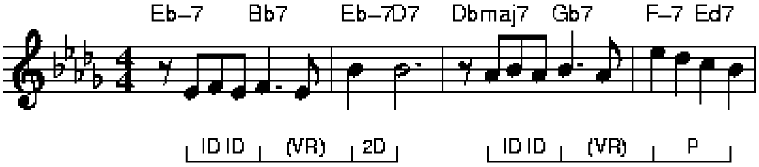

Figure 2. Narmour units parsing of the first phrase of *Body And Soul* standard

group(s) to which the note belongs to.

Using this data we applied an inductive logic programming algorithm to induce first-order rules. We applied a standard covering strategy that incrementally constructs a theory as a set of first-order rules by selecting at each step an example of the training set (covering loop) and constructing a rule that 'explains' this example (specialization loop). A strength of inductive logic programming is to allow the use of background knowledge, that is, the theory is not only induced from the training examples but the algorithm can apply user-defined concepts in order to guide the rule construction or to simply make the rule more readable.

We define 4 predicates to be learned: `stretch`, `onset`, `energy`, and `alteration`. For each note of our training set, each predicate corresponds to a particular type of transformation: `stretch` refers to duration transformation, `onset` to onset deviation, `energy` to the energy transformation, and `alteration` refers to note alteration.

For each of these predicates, the training set is composed of a set of examples that describes the performance of each note. The background knowledge contains the note-level (`melo` predicate) and structure-level (`context` predicate) of all the training excerpts as well as predicates that lead the rules to be generated regarding note successors and predecessors (`succ` predicate). We used the Aleph inductive logic programming system [21] which provides several built-in predicates to guide the rule construction.

Despite the relatively small amount of training data some of the rules generated by the learning algorithms turn out to be of musical interest. The coverage of some rules presented here highlights the role of underlying expressive transformations patterns in the performance.

The induced rules are of different types. Some focus on note-level features and depend of the performance tempo while others focus on structure-level features and are independent of the performance tempo. Note-level rules are more specific than structure-level rules as they classify a particular note in terms of the timing and pitch relationships of the note and its neighbors. Compound rules that refer to both note-level and structure-level features have also been discovered. In order to exemplify the discovered rules we present some of them next.

STRETCH RULES

S-1: [Pos cover = 12 Neg cover = 1]

```
stretch(A,B,C,lengthen) :-
    succ(C,D),
    melo(A,B,D,4,-1,-1,0,-1,1,nominal).
```

"Lengthen a note at an offbeat position if its successor is a quarter between two shorter notes, the former one being at the same pitch, the next one being lower, at a nominal tempo"

S-2: [Pos cover = 21 Neg cover = 1]

```
stretch(A, B, C, shorten) :-
    succ(C, D), succ(D, E),
    context(A, E, [nargroup(p, 1)|F]),
    context(A, C, [nargroup(p, 2)|F]).
```

"Shorten a note n if it belongs to a P Narmour group in second position and if note n+2 belongs to a P Narmour group in first position"

S-3: [Pos cover = 41 Neg cover = 1]

```
stretch(A, B, C, same) :-
    succ(C, D), succ(D, E),
    context(A, E, [nargroup(vr, 3)|F]),
    member(nargroup(p, 1), F).
```

"Do not stretch a note n if note n+2 belongs to both VR Narmour group in third position and P Narmour group in first position "

ONSET DEVIATION RULES

O-1: [Pos cover = 41 Neg cover = 2]

```
onset(A, B, C, same) :-
    succ(C, D),
    context(A, D, [nargroup(vr, 3)|E]),
    member(nargroup(d, 1), E).
```

“Play a note at the right time if its successor belongs to a VR Narmour group in third position and to a D Narmour group in first position”

O-2: [Pos cover = 10 Neg cover = 1]

```
onset(A, B, C, delay) :-
    succ(D, C),
    context(A, D, [nargroup(id, 3)|E]),
    member(nargroup(ip, 2), E).
```

“Play a note n with delay if its predecessor belongs to a ID Narmour group in third position and to a IP Narmour group in second position”

O-3: [Pos cover = 17 Neg cover = 1]

```
onset(A, B, C, advance) :-
    succ(C,D), succ(D,E),
    context(A,E,[nargroup(ip,1)|F]),
    context(A,D,[nargroup(p,3)|F]).
```

“Play a note n in advance if n+1 belongs to a P Narmour group in third position and if n+2 belongs to an IP Narmour group in first position”

O-4: [Pos cover = 3 Neg cover = 0]

```
onset(A, B, C, advance) :-
    melo(A,B,C,6,0,0,1,1,0,slow),
    context(A, C, [nargroup(p, 3)|D]).
```

“In slow interpretations, play a triplet in advance if it is between two higher triplets, if it is neither in a beat position nor an offbeat position, and if it belongs to a P Narmour group in third position”

ENERGY RULES

E-1: [Pos cover = 26 Neg cover = 0]

```
energy(A, B, C, loud) :-
    succ(D, C),
    context(A, D, [nargroup(d, 2)|E]),
    context(A, C, [nargroup(id, 1)|E]).
```

“Play loud a note if it belongs to an ID Narmour group in first position and if its predecessor belongs to a D Narmour group in second position”

E-2a: [Rule 14] [Pos cover = 34 Neg cover = 1]

```
energy(A, B, C, soft) :-
    succ(C, D),
    context(A, D, [nargroup(p, 4)|E]),
    context(A, C, [nargroup(p, 3)|E]).
```

E-2b: [Pos cover = 34 Neg cover = 1]

```
energy(A, B, C, soft) :-
    succ(D, C),
    context(A, D, [nargroup(p, 3)|E]),
    context(A, C, [nargroup(p, 4)|E]).
```

“Play soft two successive notes if they belong to a P Narmour group respectively in third and forth position”

E-3: [Pos cover = 19 Neg cover = 0]

```
energy(A, B, C, loud) :-
    succ(D, C),
    melo(A,B,D,8,0,0,-1,1,2,nominal).
```

“At nominal tempo, play loud an eight between two eights if it is on second or forth bar beat and if the 3 notes form a regular ascending scale”

E-4a: [Pos cover = 30 Neg cover = 2]

```
energy(A, B, C, same) :-
    context(A, C, [nargroup(ip, 1)|D]).
```

E-4b: [Pos cover = 34 Neg cover = 2]

```
energy(A, B, C, same) :-
    succ(D, C),
    context(A, D, [nargroup(ip, 1)|E]).
```

“Play a two notes at a normal level if the first one belongs to an IP Narmour group in first position”

ALTERATION RULES

A-1: [Pos cover = 232 Neg cover = 0]

```
alteration(A, B, C, none) :-
    context(A, C, [nargroup(p, 2)|D]).
```

“Do not perform alteration of a note if it belongs to a P Narmour group in second position”

A-2: [Pos cover = 8 Neg cover = 0]

```
alteration(A, B, C, ornamentation) :-
    succ(C, D),
    context(A, D, [nargroup(d, 2)|E]),
    member(nargroup(ip, 1), E),
    context(A, C, [nargroup(vr, 3)|F]).
```

“Ornamentate a note if it belongs to a VR Narmour group in third position and if its successor belongs to D Narmour group in second position and to IP Narmour group in first position”

A-3: [Pos cover = 4 Neg cover = 1]

```
alteration(A, B, C, consolidation) :-
    succ(C, D),
    context(A, D, [nargroup(ip, 1)|E]),
    context(A, C, [nargroup(ip, 3)|E]).
```

“Consolidate a note n with note n+1 if note n belongs to an IP Narmour group in third position and note n+1 belongs to an IP Narmour group in first position”

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 [18, 19]. 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 [20] discovered a small set of 17 quite simple classification rules [18] 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 [3], [2], [14] and [9]. In [3], 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 [14], 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.

5. CONCLUSION

This paper describes an inductive approach for learning expressive performance rules from Jazz standards recordings by a skilled saxophone player. Our objective has been to find note-level and structure-level rules which predict, for a significant number of cases, how a particular note in a particular context should be played or how a melody should be altered. In order to induce these rules, we have extracted a set of acoustic features from the recordings resulting in a symbolic representation of the performed pieces and then applied the Aleph inductive logic programming system to the symbolic data and information about the context in which the data appear.

Future work: This paper presents work in progress so there is future work in different directions. We plan to increase 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. We also plan to use our research not only for obtaining interpretable rules about expressive transformations in musical performances, but also to generate expressive performances. With this aim we will apply regression methods to derive numeric models from the data. We intend to incorporate higher-level structure information (e.g. phrase structure information) to obtain a more complete integrated model of expressive performance. Another short-term research objective is to compare expressive performance rules induced from recordings at substantially different tempos. This would give us an indication of how the musician note-level choices vary according to the tempo.

Acknowledgments: This work is supported by the Spanish TIC project ProMusic (TIC 2003-07776-C02-01). We would like to thank Maarten Grachten for providing his Narmour I/R structure parser implementation, as well as the anonymous reviewers for their insightful comments and pointers to related work.

6. 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] Dovey, M.J. (1995). Analysis of Rachmaninoff's Piano Performances Using Inductive Logic Programming. European Conference on Machine Learning, Springer-Verlag.

[4] Friberg, A. (1995). A Quantitative Rule System for Musical Performance. PhD Thesis, KTH, Sweden.

[5] Gabrielsson, A. (1999). The performance of Music. In D.Deutsch (Ed.), The Psychology of Music (2nd ed.) Academic Press.

[6] Gómez, E. (2002). Melodic Description of Audio Signals for Music Content Processing. Doctoral Pre-Thesis Work, UPF, Barcelona.

[7] Gómez, E., Gouyon, F., Herrera, P. and Amatriain, X. (2003). Using and enhancing the current MPEG-7 standard for a music content processing tool, Proceedings of the 114th Audio Engineering Society Convention.

[8] Gómez, E. Grachten, M. Amatriain, X. Arcos, J. (2003). Melodic characterization of monophonic recordings for expressive tempo transformations. Stockholm Music Acoustics Conference.

[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] Klapuri, A. (1999). Sound Onset Detection by Applying Psychoacoustic Knowledge, Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP.

[11] Maher, R.C. and Beauchamp, J.W. (1994). Fundamental frequency estimation of musical signals using a two-way mismatch procedure, Journal of the Acoustic Society of America, vol. 95 pp. 2254-2263.

[12] McNab, R.J., Smith Ll. A. and Witten I.H., (1996). Signal Processing for Melody Transcription,SIG working paper, vol. 95-22.

[13] Mitchell, T.M. (1997). Machine Learning. McGraw-Hill.

[14] Morales, E. (1997). PAL: A Pattern-Based First-Order Inductive System. Machine Learning, 26, 227-252.

[15] Quinlan, J.R. (1993). C4.5: Programs for Machine Learning, San Francisco, Morgan Kaufmann.

[16] 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.

[17] Todd, N. (1992). The Dynamics of Dynamics: a Model of Musical Expression. Journal of the Acoustical Society of America 91.

[18] Widmer, G. (2002). Machine Discoveries: A Few Simple, Robust Local Expression Principles. Journal of New Music Research 31(1), 37-50.

[19] 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.

[20] 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.

[21] Srinivasan, A. (2001). The Aleph Manual.

[22] Narmour, E. (1990). The Analysis and Cognition of Basic Melodic Structures: The Implication Realization Model. University of Chicago Press.