A Machine Learning Approach to Expressive Performance in Jazz Standards

Rafael Ramirez Amaury Hazan Emilia Gomez Esteban Maestre
Music Technology Group
Pompeu Fabra University
Ocata, 1. 08003
Barcelona, Spain
{rafael,ahazan,egomez,emaestre}@iua.upf.es

ABSTRACT

We describe an approach to perform expressive transformation in monophonic Jazz melodies. The system consists of three components: (a) a melodic transcription component which extracts a set of acoustic features from monophonic recordings, (b) a machine learning component which induce expressive transformation models from the set of extracted acoustic features, and (c) a melody synthesis component which generates expressive monophonic output (MIDI or audio) from inexpressive melody descriptions using the induced expressive transformation model. We describe and compare different machine learning methods for inducing the expressive transformation models.

Categories and Subject Descriptors

J.5 [**Computer Applications**]: Arts and Humanities Performing Arts

Keywords

Expressive Transformations, Data Mining, Audio Processing

1. INTRODUCTION

Modeling expressive music performance is one of the most challenging aspects of computer music. The focus of this paper is the study of how skilled musicians (saxophone Jazz players in particular) express and communicate their view of the musical and emotional content of musical pieces by introducing deviations and changes of various parameters like timing, dynamics, etc. The deviations and changes we consider in this paper are on note duration, note onset and note energy. The study of these variations is the basis of an inductive content-based transformation system for performing expressive transformation on musical phrases. The audio processing part of the system consists of a melodic description component and a synthesis component based on Spectral Modelling Analysis and Synthesis. For the inductive part of the system, we have explored both regression and classification machine learning techniques such as decision trees, model trees and support vector machines, among others.

The rest of the paper is organized as follows: Section 2 describes the melodic description component of the system. Section 3 describes the different approaches we have considered for the inductive part of the system and some results of a comparison among them. Section 4 briefly describes how we generate both MIDI and audio output. Section 5 reports on some related work, and finally Section 6 presents some conclusions and indicates some areas of future research.

MDM/KDD '04 August 22, 2004, Seattle, WA, USA.
.

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 [11]. 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 [10].

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 [9, 11] 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) [14]. 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. After a first test of this implementation, some improvements to the original algorithm where implemented and reported in [9].

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 [13]. 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 [15].

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. EXPRESSIVE PERFORMANCE KNOWLEDGE INDUCTION

In this section, we describe different inductive approaches to learning expressive performance models from monophonic recordings by a skilled saxophone Jazz player. Our aim is to apply and compare different machine learning techniques in order to be able to predict, for a significant number of cases, how a particular note in a particular context should be played (e.g. longer or shorter than its nominal duration and by how much). We are aware of the fact that not all the expressive transformations (e.g. tempo transformations) 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.

Data set. The training data used in our experimental investigations are monophonic recordings of three Jazz standards (*Body and Soul, Once I loved* and *Like Someone in Love*) 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 confortable 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 intevals around the nominal tempo (5 faster and 5 slower) within the fastest-slowest tempo limits. The resulting data set is composed of 1936 performed notes.

Descriptors. In this paper, we are concerned with note-level (in particular note duration, note onset and note energy) expressive transformations. Each note in the training data is annotated with its corresponding deviation 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 duration and the note metrical position, while information about its context includes duration of previous and following notes, extension and direction of the intervals between the note and the previous and following notes, and the note Narmour group(s) [18]. Narmour's Implication/Realization (I/R) model is a model of melodic structure, based on principles akin to Gestalt Theory. An I/R analysis consists of a grouping of notes and categorizing these groups into a set of predefined categories. In [11] was developed a parser for melodies that automatically generates I/R analyses. It implements most of the basic ideas from the I/R model. Fig. 1. shows the basic melodic units used by our parser.

Machine learning techniques. In order to induce predictive models for duration ratio, onset deviation and energy, variation, we have applied two types of machine learning techniques, namely regression techniques and classification techniques. On the one hand, regression methods are considered to be *black-box* in the sense that it is very difficult (or impossible) to understand the predictions they produce. Black-box statistical approaches may be good at deriving predictions from data, but formulating understandable rules from the analysis of data is something entirely different from formulating predictive models from that data. On the other hand, classification methods are good at *explaining* the predictions they provide but are restricted to a set of discrete classes as prediction space. Our problem at hand is one that requires the prediction precision of regression methods for generating accurate solutions (i.e. expressive performances) but at the same time it is highly desirable to be able to explain the system predictions. Having this in mind, we have not limited ourselves to anyone of the two techniques and instead we have explored both approaches applying a representative sample of both regression and classification methods.

3.1 Regression methods

We have explored different regression methods to induce predictive models for duration ratio, onset deviation and energy variation. The methods we have included in our research are:

- **Linear regression** is the simplest scheme for numeric prediction which has been widely used in statistical applications. The idea is to express the predicted class value as a linear combination of the attributes, with predetermined weights that are calculated from the

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

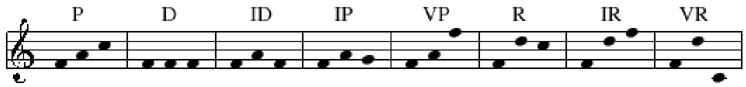

Figure 1: Basical Narmour I/R melodic units

training data, in order to minimize the overall deviation between the training values and the model. Linear regression is not the best choice to approximate non-linear functions. As confirmed by the results (Section 4) expressive performance does not seem to be a linear function on the attributes. However, we have decided to include linear regression in our experiments as a reference point to the other methods.

- **Model trees** build an induction tree containing at each leave a different linear model. Model trees behave well as each linear model at their leaves approximates a set of more specific cases than with a global linear model, i.e. model trees approximate continuous functions by linear "patches", a more sophisticated representation than simple linear regression.

- **Support Vector Machines** take great advantage of using a non linear attribute mapping that leads them to be able to predict non-linear models (thought they remain linear in a higher dimension space). Thus, they provide a more flexible prediction, but with a higher computational cost necessary to perform all the computations in the higher dimensional space. Training a Support Vector Machine requires the solution of a very large quadratic programming (QP) optimization problem. The QP resolution has been optimized in terms of speed and memory usage with the Sequential Minimal Optimization algorithm [27]. They have been applied to numerical prediction [28]. The results largely depend of the tuning of the algorithm, e.g. the choice of the kernel evaluation function, and the parameters which control the amount up to which deviations are tolerated (denoted by epsilon). The kernel function defines implicitly the higher dimensional mapping applied to the training vector. With an appropriate tuning one can control the number of support vectors that define a boundary between two classes. We were interested in having a model with a relatively reduced number of support vectors (i.e. less than a third of the training instances) in order to avoid overfitting and thus have tuned empirically 4 support vector machine with the following parameters: **(1)** Linear kernel, C=1, epsilon=0.05; **(2)** 2nd order polynomial kernel, C=1, epsilon=0.05; **(3)** 3rd order polynomial kernel, C=1, epsilon=0.05; **(4)** Radial Basis Function kernel, gamma exponent=0.95, C=10, epsilon=0.05.

3.2 Classification methods

We have also explored different classification methods. We discretized the input values into classes, and used classification techniques in order to induce predictive models for duration ratio, onset deviation and energy variation. The number of classes was determined by the distribution of the input values. We applied a fix-width discretization for duration ratio and onset deviation with 9 and 7 classes, respectively. The case of energy variation is quite different as there is no information about note energy in the score and we have to characterize each note energy in relation to the note average energy in the recordings. Thus we performed a frequency discretization i.e. each target class contains the same number of cases, and characterize soft, normal, and loud notes. Consequently the classification results presented later have to be compared with the accuracy of a random classification i.e. 11.11% for duration ratio, 14.28% for onset deviation, and 33.33% for energy variation. The methods we have included in our research are:

- The **naive Bayes classifier** is based on Bayes rule on conditional probability propagation. It is called "naive" because this rule assumes that each attribute of an instance are completely independent. This can lead to weak results when attributes have a strong correlation. The algorithm implements redundant attributes filtering as preprocess [26]. Despite being one of the simplest ML algorithm, it has outperformed more complex techniques in a significant number of cases.

- **Lazy Methods.** The notion of lazy learning subsumes a family of algorithms that store the complete set of given (classified) examples of an underlying example language and delay all further calculations until requests for classifying yet unseen instances are received. The K-Nearest Neighbor algorithm, is one of the most popular instanced-based algorithm, which handles well noisy data if the training set has an acceptable size. The main idea of this algorithm is to compare a test set with its nearest neighbors (the number is determined by the user and the computational cost largely depends of it). A major problem of the simple approach of K-NN is that the vector distance will not necessarily be suited for finding intuitively similar examples, especially if irrelevant attributes are present. We empirically found the best number of neighbors: 5 for the duration ratio model, 11 for the onset deviation model, and 1 for the energy variation model. KStar algorithms proceeds in an similar fashion as K-NN, but use an entropy similarity measure distance to find the neighbors of a test vector.

- **Tree induction algorithms** build a tree model by selecting at each node the most relevant attribute. We compare the results of C4.5 [19, 20], C4.5 with boosting, and the random forest algorithm. Boosting refers to a meta-algorithm that can improve the results of any classification algorithm by giving to each instance of the training set a particular weight proportional with the difficulty to classify such instance. That is, a first classification model is proposed giving the same weight to all the training instances. Misclassified instances with the model are then given a greater weight, and so on. After a user defined number of iterations (in our case 10 iterations) the resulting model is able to deal with "difficult" training instances. This boosting method can drastically improve the results of an inaccurate model, thought overfitting can occur. The random forest algorithm uses a bagging technique: it combines the decision of different models amalgamating the various outputs in a single prediction. The decision can be seen as a vote between the models. Each tree is built using random features selection. We used 20 random trees in our test with the random forest algorithm.

3.3 Results

We have a comparative table for each of the expressive transformation aspects we are dealing with, namely note duration (Table 1), onset (Table 2) and energy (Table 3). We performed a 10-fold cross validation for all the algorithms and all the tests were performed using the Waikato Environment for Knowledge Analisis [29]. In Table 1, 2 and 3, C.C.I refers to the correctly classified instances rate, C.C to the correlation coefficient, and R.A.E to the relative absolute error and R.R.S.E the root relative squared error.

Among the regression methods, model tree regression is consistently the most accurate method, while C4.5 (C4.5 with boosting in the case of energy variation) is the most accurate classification method. Most of the misclassified instances by C4.5 are classified into neighbor classes to the correct class. As mentioned before, linear regression performs poorly since, as expected, expressive performance is a complex and multi-level phenomenon which cannot be handled accurately by a linear model. Also as expected, support vector machines perform well only with a radial function kernel, or a 3rd order polynomial kernel or higher.

4. SYNTHESIS

We generate an expressive MIDI performance from an inexpressive description of the melody (i.e. the score description). We compute the expressive duration of a note by multiplying the predicted duration ratio and the inexpressive note duration. The expressive note onset is obtained adding the predicted onset deviation and the inexpressive onset value. The case of the energy is different as the relation between note energy and corresponding MIDI velocity (an integer between 0 and 127) is quite arbitrary. We defined the audio energy to MIDI velocity mapping as $velocity = 63 * log_{10}(energy) + 64$ where the audio energy is normalized to $1 \leq energy \leq 10$. We also generate an expressive audio from an source audio file. The process is independent of the nature of the audio source (which can either be generated by a synthesizer or be given as an audio recording). The system needs a melodic description (note onsets, durations and energies) in addition to the audio source. We use SMSTools [23] to transform the audio source (e.g. by applying a time-stretch transformation) into an expressive audio file without affecting any other perceptual feature, like pitch or spectral shape.

5. RELATED WORK

Widmer [30, 31]. 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 [32] discovered a small set of 17 quite simple classification rules [30] 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.

Tobudic et al. [24] 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.

Other inductive machine learning approaches to rule learning in music and musical analysis include [6], [3], [17] and [12]. In [6], Dovey analyzes piano performances of Rachmaniloff pieces using inductive logic programming and extracts rules underlying them. In [3], 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 [17], 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 [12], 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.

6. CONCLUSION

We have described an approach to perform expressive transformation in monophonic Jazz melodies (the deviations and changes we consider are on note duration, note onset and note energy). Our approach consists of (a) a melodic transcription component which extracts a set of acoustic features from monophonic recordings, (b) a machine learning component which induce expressive transformation models from the set of extracted acoustic features, and (c) a melody synthesis component which generates expressive monophonic phrases from inexpressive phrases using the induced expressive transformation model. For the machine learning component, we considered both classification and regression machine learning techniques. Of particular interest are the models obtained using C4.5 and model trees which are among the most accurate.

Future work: This paper presents work in progress so there is future work in different directions. We plan to increase

Algorithm	C.C.I(%)	C.C	R.A.E(%)	R.R.S.E(%)
C4.5	**74.59**	-	68.07	86.84
C4.5 with Boosting	72.99	-	72.87	87.35
RandomForest	70.51	-	**62.74**	89.85
KStar	73.71	-	63.04	**86.65**
KNN	67.15	-	72.77	95.57
Naive Bayes	59.97	-	98.18	103.07
Linear Regression	-	0.33	98.69	94.39
Least Med Square Regression	-	0.29	95.22	96.60
Model Tree Regression	-	**0.72**	**74.89**	**69.14**
SVM Regression (1)	-	0.29	95.30	96.15
SVM Regression (2)	-	0.48	89.01	88.24
SVM Regression (3)	-	0.66	76.65	75.47
SVM Regression (4)	-	0.70	81.11	71.23

Table 1: Cross validation results for duration ratio

Algorithm	C.C.I(%)	C.C	R.A.E(%)	R.R.S.E(%)
C4.5	**78.61**	-	68.47	88.12
C4.5 with Boosting	78.56	-	**57.72**	87.72
RandomForest	78.09	-	59.22	**85.73**
KStar	76.49	-	66.34	88.63
KNN	74.27	-	82.46	94.87
Naive Bayes	68.85	-	104.87	104.03
Linear Regression	-	0.17	101.12	98.41
Least Med Square Regression	-	0.01	92.50	101.32
Model Tree Regression	-	0.43	91.51	**90.16**
SVM Regression (1)	-	0.14	99.92	98.88
SVM Regression (2)	-	0.24	**89.34**	98.18
SVM Regression (3)	-	0.38	95.41	92.50
SVM Regression (4)	-	**0.44**	94.56	90.34

Table 2: Cross validation results for onset deviation

Algorithm	C.C.I(%)	C.C	R.A.E(%)	R.R.S.E(%)
C4.5	72.83	-	55.38	76.25
C4.5 with Boosting	**73.3**	-	**44.56**	**74.1**
RandomForest	70.3	-	51.7	78.04
KStar	70.92	-	52.42	75.5
KNN	58.01	-	61.91	102.67
Naive Bayes	54.8	-	80.17	91.16
Linear Regression	-	0.27	95.69	96.13
Least Med Square Regression	-	0.22	87.92	108.01
Model Tree Regression	-	**0.67**	**66.31**	**74.31**
SVM Regression (1)	-	0.25	89.28	98.57
SVM Regression (2)	-	0.47	82.53	89.4
SVM Regression (3)	-	0.56	75.47	82.95
SVM Regression (4)	-	0.64	69.28	77.23

Table 3: Cross validation results for energy variation

the amount of training data, the amount of descriptors to be extracted from it (e.g. vibrato) and combine this two with background musical knowledge. This will certainly generate a more complete model of expressive performance. 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 Maarten Grachten for providing the Narmour analysis and Ismael Mosquera for the implementation of the MIDI stream output.

7. REFERENCES

[1] Amatriain, X. de Boer, M. Robledo, E. Garcia, D. (2002). CLAM: An OO Framework for Developing Audio and Music Applications Proceedings of 17th Annual ACM Conf. on OO Programming, Systems, Languages and Applications Seattle, WA, USA.

[2] Agrawal, R.T. (1993). Mining association rules between sets of items in large databases. International Conference on Management of Data, ACM, 207,216.

[3] 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.

[4] Bresin, R. (2000). Virtual Visrtuosity: Studies in Automatic Music Performance. PhD Thesis, KTH, Sweden.

[5] Chen, M., Jan, J. and Yu, P.S. (1996). Data Mining: An Overview from a Database Perpective. IEEE Trans. Knowledge and Data Engineering 8(6), 866-883.

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

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

[10] Gomez, 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.

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

[12] 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.

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

[14] 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.

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

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

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

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

[19] Quinlan, J.R. (1986). Induction of decision trees. Machine Learning, 1(1), 81-106.

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

[21] 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.

[22] Seashore, C.E. (ed.) (1936). Objective Analysis of Music Performance. University of Iowa Press.

[23] SMSTools: http://www.iua.upf.es/sms

[24] 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.

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

[26] Langley, P.,Sage, S. (1994). Induction of selective Bayesian classifiers. Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence (pp. 399-406). Seattle, WA: Morgan Kaufmann.

[27] Platt, J. (1998). Fast Training of Support Vector Machines using Sequential Minimal Optimization.

[28] Smola, A.J., Sch?lkpof, B. (1998). A Tutorial on Support Vector Regression. NeuroCOLT2 Technical report series.

[29] Witten, I.H. (1999). Data Mining, Practical Machine Learning Tools and Techniques with Java Implementation, Morgan Kaufmann Publishers.

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

[31] 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), L?beck, Germany. Berlin: Springer-Verlag.

[32] 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.