Learning Singer-Specific Performance Rules

Maria Cristina Marinescu and Rafael Ramirez

***Abstract*—This work investigates how opera singers manipulate timing in order to produce expressive performances that have common features but also bear a distinguishable personal style. We characterize performances not only relative to the score, but also consider the contribution of features extracted from the libretto. Our approach is based on applying machine learning to extract singer-specific patterns of expressive singing from performances by Josep Carreras and Placido Domingo. We compare and contrast some of these rules, and we draw some analogies between them and some of the general expressive performance rules existing in the literature.**

***Index Terms*—Expressive performance, machine learning, timing model**

I. Introduction

In an interview that Charlie Rose took of the ``three tenors'' back in 1994, Placido Domingo was explaining how he tries to color the notes to give a song the feel that he's looking for. Josep Carreras instead talks about building each note with precision until it transmits the right emotion in the context, but gladly sacrificing precision for expressiveness. While opera singers may conceptualize the interpretation process very differently and possibly at different abstraction levels, the modifications they apply to the score may be similar. With certainty, nevertheless, there are expressive changes that they consistently apply that create their personal mark.

This work focuses on how these two specific singers manipulate timing to create expressive interpretations that have a well-defined personal style. We start with a benchmark suite consisting of CD recordings of a cappella fragments from different tenor arias - seven performed by Josep Carreras and six by Placido Domingo. Using sound analysis techniques based on spectral models we extract acoustic high-level descriptors representing properties of each note, as well as of its context. A note is characterized by its pitch and duration. The context information for a given note consists of the relative pitch and duration of the neighboring notes, as well as the Narmour [1] structures to which the note belongs.

Given that the libretto is an important part of an operatic performance which may reinforce - but may also change the expressive quality of the music - we also consider it when characterizing the notes. Each note has a syllable - occasionally a couple of syllables - associated with it. Every syllable is naturally strongly or weakly stressed. A performer may choose to accentuate weakly stressed syllables, or de-accentuate strong ones. Similarly, he can create syncopation by placing accent on weak beats, de-accentuating strong beats, or even pausing where a strong beat would normally occur. Whether inherent in the composition or introduced as expressive modifications, the performer may need to reconciliate possibly contradicting prosodic, metric, and score cues. For instance, adopting the wrong intonation or grouping the lyrics into the wrong prosodic units can ruin an otherwise good interpretation. In this work we are considering libretto descriptors such as syllable stress and whether a syllable marks the end of a prosodic unit as explained later in the paper.

Once each note in the benchmark suite is associated the corresponding acoustic and prosodic descriptors we apply machine learning techniques to understand under which conditions a performer modifies the score. Some of the most interesting rules we learn are presented in the result section. We contrast and compare some of the rules we learn from the performances of the two singers; we also compare these singer-specific rules with some of the general expressivity rules such as those proposed by Widmer [2] and the KTH [3] system. As expected, some of our rules describe similar concepts, although they are refinements of these and have lower coverage. There are also some rules for which we did not find good evidence, possibly due to the constraints on the size of our dataset and to the fact that the rules are probably sensitive to the style of music that they characterize.

The rest of the paper is organized as follows. Section II describes related work in expressive performance. Section III describes our test suite, introduces the note-level descriptors, and explains how we extract the data that is used as the input to the ML algorithms. Section IV presents the learning algorithms; Section V discusses some of the most interesting results. We conclude in Section VI.

II. Related Work

Understanding and formalizing expressive music performance is a challenging problem (e.g.[4]-[6]) which has been mainly approached via statistical analysis (e.g.[7]), mathematical modeling (e.g.[8]), and analysis-by-synthesis (e.g.[9]). In all these approaches, it is a person who is responsible for devising a theory which captures different aspects of musical expressive performance. This model is later tested on real performance data in order to determine its accuracy.

A. Machine Learning Techniques

As far as previous research addressing expressive music performance using machine learning techniques, Widmer [2] reports on the task of discovering general rules of expressive

Manuscript received March 5, 2012; revised April 6, 2012. This work was supported in part by the project TIN2010-16497 financed by the Ministry of Science and Education, Spain.

M. Cristina Marinescu is with Universidad Carlos III de Madrid, Department of Computer Science, and Leganes 28911, Spain (e-mail: mcristina@arcos.inf.uc3m.es).

R. Ramirez is with Universitat Pompeu Fabra, Music Technology Group, Barcelona 08018, Spain (e-mail: rramirez@iua.upf.edu).

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 Mozart performed by a skilled pianist in the studio. An inductive rule learning algorithm discovered a small set of quite simple classification rules that predict a large number of the note-level choices of the pianist. We will also compare some of their rules with the singer-specific rules we obtain.

Tobudic et al. [10] 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. Ramirez et al. [11] explore and compare different machine learning techniques for inducing both an interpretable and a generative expressive performance model for monophonic Jazz performances. They propose an expressive performance system based on inductive logic programming which learns a set of first order logic rules that capture expressive transformation both at an inter-note level and at an intra-note level. Based on the theory generated by the set of rules, they implement a melody synthesis component, which generates expressive monophonic output (MIDI or audio) from inexpressive MIDI melody descriptions.

Lopez de Mantaras et al. [12] report on SaxEx, 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 retrieves 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. One limitation of their system is that it is incapable of explaining the predictions it makes. Other inductive machine learning approaches to rule learning in music and musical analysis include [13]-[15].

B. Singing voice Synthesis

Most of the research in expressive music performance is concerned with instrumental music, particularly jazz and classical, and focuses on specific instruments (e.g. piano, saxophone). However, singing voice expressive performance has been much less explored. Alonso [16] describes the design of an expressive performance model focused on emotions for a singing voice synthesizer. The model is based on the rule system developed at KTH; the singing voice synthesizer is Daisy - developed at MTG at the UPF in Barcelona.

Some approaches to synthesize expressive singing use a singing performance to control pitch and timing, e.g. [17]. In these approaches, it is a singing performance which directly controls the synthesized expressive performance.

Another interesting approach is Vocalistener [18]. Their system tries to mimic a reference user voice by automatically predicting several parameters (*f0*, energy, onset and duration of notes) from the song lyrics. This approach is motivated by the fact that configuring these parameters is a time consuming and difficult task. Extending this approach to other features could be helpful for the generation of models of a particular singer, or those of artists belonging to a particular style.

There have been other approaches to modeling the control parameters using system's inputs, e.g. [19], [20]. Both works attempt to model *f0* in order to generate pitch contours mainly using second order exponential damping and oscillation models.

In addition to *f0*, energy, and timing, performers often use other expressive resources such as growl and rough voice. Loscos et al. [21] have studied roughness caused by inter-period variations of the pitch (jitter) and the period amplitude (shimmer), as well as growl which is often used as an expressive accent.

Saino [22] models singing style statistically, focusing on relative pitch, vibrato (rate and shape), and dynamics using context-dependent Hidden Markov Models. The parameters dependence on phonetics is removed and notes are considered to contain up to three regions depending on their position ('beginning', 'sustained' and 'end') which lead to up to seven patterns as a result of their combination.

The KTH [3] rule system for singing synthesis is of particular relevance to us since it can be used to synthesize opera singer's voices. Some of the rules that they apply were originally developed for instruments ([23],[24]), others have been directly created in collaboration with a violinist and conservatory music teacher. We compare a few of their musical rules with the ones we have obtained.

III. Our Training Data

Studying the singing style of well-known singers raises the issue of obtaining an extended training set. Not only there exist a small number of operatic fragments written for solo voice, but also the singer-specific expressive patterns may not transfer well across music styles. Automatic extraction of the voice from polyphonic pieces which has enough quality for our purposes is not a viable option. As a result, our training set consists of several fragments from five operas by Verdi and the recitativa *Tombe degli avi miei* from *Lucia di Lamermoore* by Donizetti. After manually eliminating those notes during which the orchestra can be heard, we are left with 841 notes in which the tenor and the orchestra do not overlap - 443 for Carreras and 398 for Domingo.

A. Acoustic and Prosodic Analysis

We use sound analysis techniques based on spectral models [25] for extracting high-level symbolic features from CD recordings. We characterize each performed note acoustically by a set of features representing both properties of the note and aspects of the musical context in which the note appears. Information about the note includes note pitch, duration, and metrical strength; information about its context includes the relative pitch, duration, and duration ratio of the neighboring notes (i.e. previous and following notes). For each musical fragment we additionally compute the actual tempo - without considering the notes annotated with fermata - and we associate it with every note in the fragment.

The metrical strength depends on the meter signature that the music is written in. For instance, for a 4/4 signature the metrical strength is *verystrong* for the first beat, *strong* for the third beat, *medium* for the second and fourth beats, *weak* for

the offbeat, and *veryweak* for any other position of the note within a bar.

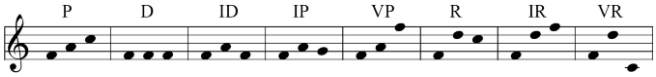


Fig. 1. Prototypical narmour structures

We parse each melody in the training data and, based on the pitch information of the neighboring notes, we automatically extract the Narmour structures to which every note belongs. This is a way to provide an abstract structure to our performance data. The Implication/Realization model proposed by Narmour is a theory of perception and cognition of melodies. The theory states that a melodic musical line continuously causes listeners to generate expectations of how the melody should continue. According to Narmour, any two consecutively perceived notes constitute a melodic interval, and if this interval is not conceived as complete, it is an implicative interval, i.e. an interval that implies a subsequent interval with certain characteristics. That is to say, some notes are more likely than others to follow the implicative interval. Based on this, melodic patterns or groups can be identified that either satisfy or violate the implication as predicted by the intervals. Fig. 1 shows prototypical Narmour structures.

Prosody can carry emotional information depending on intonational phrasing, and a skilled singer must manipulate the acoustic and prosodic parameters without transmitting conflicting messages. To begin understanding this interplay, we introduce two additional note annotations: (1) the stress naturally assigned in speech to the syllable which corresponds to the note (*strong* or *weak*), and (2) whether the note marks the end of a prosodic unit (*PU*), sub-prosodic unit (*SPU*), or of the phrase (*EPH*). For the cases in which two syllables correspond to a single note we assign it weak stress only if both syllables have naturally weak stress. A prosodic unit is a semantic unit of meaning which can be as short as a word and as long as a statement; it is a chunk of speech that may in fact reflect how the brain processes speech. Even though it isn't necessary that the prosodic units and those phrases that hold well together musically overlap, in practice this is often the case. In the case of a vocal musical piece the structural information which the singer tries to convey via expressive alterations has to do both with the structure of the score as well as with that of the libretto; we therefore expect to observe unit termination rules. We consider that a prosodic unit ends at the end of each statement and is composed of sub-prosodic units.

IV. The Learning Task

We approach our task as a regression problem to learn a model for predicting the lengthening ratio of the performed note relative to the score note. The duration of the note as prescribed by the score is computed based on the actual tempo of the piece that the note is part of. A predicted ratio greater than 1 corresponds to performing the note longer than specified in the score, while a ratio smaller than 1 corresponds to a shortened note. We use decision tree-based algorithms in Weka [26] for the learning task; specifically we use J48, REPTree, and M5. We also use Multilayer Perceptron (MlPerc) [27], as well as Bagging [28] and Gradient Boosting [29] with support vectors [30]. J48 is an implementation of the C4.5 [31] top-down decision tree algorithm. REPTree builds a decision/regression tree using information gain as the splitting criterion, and prunes it using reduce-error pruning with back-fitting. The M5 [32] algorithm generalizes decision trees to build model trees whose leaves consist of a linear regression model predicting the values of the output instances whose input values placed them on that path. Given the size of our dataset we use as example set the complete training data and we perform leave-one-out cross-validation.

V. Experimental Results

As result of applying the algorithms as described above we obtain a set of expressive performance rules. We discuss several of them in the remainder of this section. We use the notation *narmour(Y, gr$_n$)* to specify the Narmour groups to which the note belongs. Its arguments are a list of Narmour groups (*Y*) and the position of the note in the Narmour group ($n = 0,1,2$). The note duration is measured as the fraction of a beat, where a beat is a 1/4 note. Intervals are measured in semitones. We use lengthen/shorten in relative terms to the nominal values rather than as a class discriminator. A registral change (RC) is a pitch inflection point.

A. A Few Singer-specific Expressive Rules

Lengthen short note before inflection point: In general, Domingo lengthens a short note preceding RC if the tempo is fast; the faster the tempo, the more lengthening is needed to prepare for the note marking the change.

IF narmour(IP, gr0) AND Note_Dur < 0.5 AND
narmour(P, gr1) AND Tempo >= 1.07
THEN Str_Fct = 2.71 (D)

IF narmour(IP, gr0) AND Note_Dur < 0.3 AND
narmour(none, gr1) AND Tempo >= 1.35
THEN Str_Fct = 5.32 (D)

Carreras turns out to have more complex patterns when performing this lengthening transformation. If the tempo is not very fast he lengthens a short note preceding RC if the next interval is large; the larger the interval the more lengthening.

IF narmour(none, gr2) AND narmour(IP, gr0) AND
Note_Dur <= 0.5 AND Next_Int > -1 AND
Tempo <= 1.45 AND narmour(none, gr1) AND
Prev_Int > -1
THEN Str_Fct ∈ (Next_Int <= 6) ?
(1.53,2.27) : (2.27,3.01) (C)

This rather generic rule that both singers apply predicts the opposite of KTH's Leap Tone Duration (LTD) rule in the case of upward jumps for very short notes, particularly at very fast tempos. This rule shortens the first and lengthens the second note of a leap upward, and does the opposite for downward leaps. Similarly, Windmer's TL3 rule [2] also seems to contradict LTD; it may just be the case that the singer-specific rules don't predict what a general expressive model would.

As an exception, if the note preceding RC has *ExtremelyHigh* metric stress and it isn't a RC note itself, then Carreras shortens it to avoid taking emphasis away from the RC note.

IF narmour(IP, gr0) AND narmour(P, gr1) AND
Metro = ExtremelyHigh
THEN Str_Fct ∈ (-inf,-0.79) (C)

Give agogic accent to higher pitch notes:

We actually discovered, for both singers, more specific lengthening rules that apply to notes before inflection points if they are followed by a jump down in pitch. Interestingly, Domingo lengthens short notes with weak metric stress preceding those in RC position if they follow a longer note and are followed by a jump down, especially for fast tempos. This effectively emphasizes pitch accented notes that are otherwise associated with weak beats.

Carreras tends to lengthen RC notes longer than 1/12 following a large jump up of at least 4 semitones and marking the beginning of a descending sequence of intervals. This is an instance of KTH's LTD rule:

IF narmour(none, gr2) AND Note_Dur > 0.34 AND
narmour(P, gr0) AND narmour(IP, gr1) AND
Prev_Int <= -4
THEN Str_Fct = (2.27,3.01) (C)

Lengthen notes with strong syllable stress, shorten those with weak stress: For notes following RC, Carreras shortens unstressed and lengthens stressed syllables to correspondingly increase or diminish their importance:

IF narmour(ID, gr2) AND Note_Dur <= 0.5
THEN Str_Fct ∈ (Syll_Stress = strong)? (3.8,4.5) :
(-inf,-0.78) (C)

Mark SPU/PU: Both Domingo and Carreras lengthen a short note marking the end of a sub-prosodic unit. In the case of Domingo, the larger the jump following an *SPU* note - probably an upward jump – the more is the note marked by lengthening it. The intuition is that one way to mark the end of the semantic unit right before a note receiving tonic accent is to give it agogic accent. He applies a similar rule for notes marking *PU*s.

IF narmour(none, gr2) AND Note_Dur <= 0.5 AND
Phrasing = SPU
THEN Str_Fct ∈ (Next_Int <= 6) ?
(1.22,2.33) : (2.33,3.43) (D)

Carreras lengthens an *SPU* note when it precedes a shorter note marking an RC. He lengthens a *PU* note if the previous note is longer, or if it has the same/ or shorter duration but the current note is short. These transformations are consistent with the long final unit notes which acoustically characterize a prosodic unit:

IF narmour(none, gr2) AND Phrasing = SPU AND
Next_Dur <= -0.25 AND narmour(IP, gr0) AND
((Prev_Dur <= 0.25 AND Tempo > 1.13)
OR Prev_Dur > 0.25)
THEN Str_Fct ∈ (1.53,2.72) (C)

IF narmour(none, gr2) AND Phrasing = PU AND
Note_Dur <= 1.5 AND Prev_Dur > 0.25
THEN Str_Fct ∈ (2.27,3.01) (C)

Balancing neighboring note duration: The following rule has some similarity with the KTH Double Duration rule - which says that for two notes having the duration ratio 2:1, the short note will be lengthened and the long note shortened. In the absence of other context patterns, a note that is half or more than the length of the previous one is lengthened to be perceived more as being of the same length. At very fast tempos the lengthening is not that relevant due to the fact that the difference in durations is not that noticeable.

IF narmour(P, gr2) AND Note_Dur <= 0.5 AND
Prev_Ratio <= 2
THEN Str_Fct ∈ (Tempo <= 1.53) ?
(2.33,3.43) : (1.22,2.33) (D)

We observed that for the majority of the rules with no more context information than the duration ratios of neighboring notes, both singers lengthen the current note when shorter, and shorten it when longer, than the other note in the ratio pair. The following rule by Domingo is related to Widmer's TS1 rule in the following sense: it lengthens a shorter note followed by a note three or more times longer if the tempo is not very fast and the notes have the same pitch. The longer the next note is, the more lengthening is applied. This is the converse of TS1, which shortens the second longer note - for the same duration ratio - if the tempo is slow.

IF Note_Dur < 0.42 AND narmour(D, gr0) AND
Tempo < 1.35
THEN Str_Fct ∈ (Next_Ratio < 1.5) ? 3.9 : 2.26 (D)

Some of the rules we obtained for Carreras relate to Widmer's TL2a rule, which says that a note is lengthened if it is followed by a longer note and it is in a metrically weak position. Below is an example of such a rule:

IF Tempo > 0.45 AND Note_Dur <= 0.25 AND
Prev_Int > 0 AND Metro = ExtremelyLow AND
Next_Dur > 0.25
THEN Str_Fct ∈ (1.53,2.27) (C)

Tempo vs stress: One of the non-obvious dependencies that we observed is that Carreras' duration modifications depend more on tempo (especially for notes longer than 1/8) than Domingo's, while they don't seem to depend much on weak metric or syllable stress - which is true for Domingo.

Previous vs following note: In relative terms to the neighboring notes, Domingo's decision to modify the duration of a short note depends much more on the duration of the next, rather than the previous note. For notes longer than 1/8 the lengthening of the current note depends negatively on the duration of the previous note for both singers.

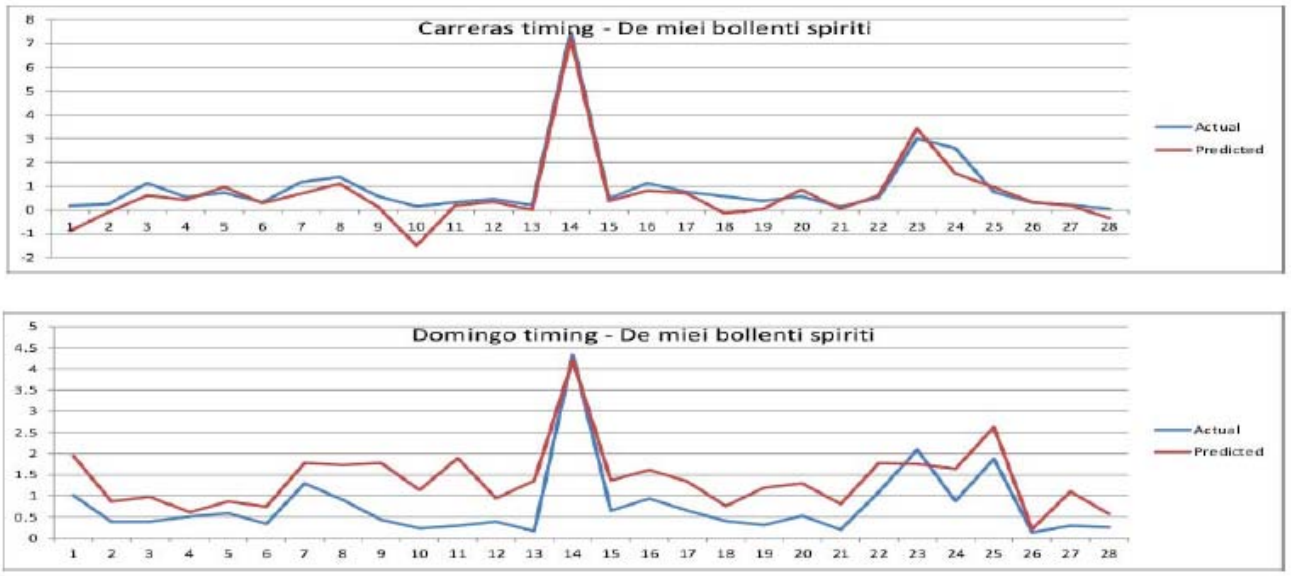


Fig. 2. Actual vs predicted note durations.

B. Correlation Coefficients

Although the purpose of this work is not generating performances that are similar in style to a singer, it is nevertheless interesting to see what are the overall correlation coefficients between the performed and predicted transformations, and how do these compare to each other throughout a musical piece. The average correlation coefficients for the three most successful algorithms are 0.65 for Carreras and 0.53 for Domingo. While these are not strong correlations it is important to note that the arias that the data comes from have widely diverse tempos and note densities, varying between 54 and 110 (tempo for Carreras), and between 56 and 98 (tempo for Domingo), while note density varies between 3.47 and 5.66 for both singers. Given this fact and the number of overall notes, a correlation of over 0.5 is better than expected. Additional experiments show that larger data sets strengthen the models such that almost every aria is better - and more consistently - predicted than in the case in which the benchmark consists of only the aria itself.

Whereas the rules that we reported on have good precision, not all have good coverage. The average precision of the rules with the largest coverage is 0.7 which corresponds to an average of 24 instances. The next cluster of rules with an average of 10.58 instances has an average precision of 0.86.

C. Correlation between predicted and performed values

A property not apparent from the correlation coefficients is the extent to which the correlation is uniformly distributed or concentrated exclusively within a particular fragment. Fig. 2 shows the note-by note duration ratio for one of the aria fragments (relative to the score duration) for Carreras and Domingo. We plot both the performed values (actual) as well as the values predicted by the best singer-specific regression model - obtained via Gradient Boosting using support vectors. These figures correspond to the fragment from the aria *De miei bollenti spiriti* from *La Traviata*. The predictions are obtained by supplying the aria as a test set and using the Multilayer Perceptron algorithm. The average correlation coefficients over all arias when validating each aria using the test set method are 0.89 for Carreras and 0.79 for Domingo. Several other algorithms also give very good predictions; of these, a k-nearest neighbor and a bagging algorithm with decision tables return predictions that are also uniformly distributed over the arias. As Fig. 2 shows, the predictions are quite uniformly distributed over the test fragment. Carreras' note durations are better predicted than Domingo's, although Domingo's model never predicts a shortening when a lengthening is performed, or vice versa; this does happen for 4 of the 28 notes for Carreras.

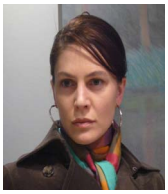
VI. Conclusion

This paper analyzes how tenors manipulate timing in order to produce expressive performances; to do this we characterize performances via parameters extracted from both the score and the libretto. We employ machine learning methods to extract singer-specific patterns of expressive singing from performances by Carreras and Domingo. We compare and contrast the rules we obtained and we draw some analogies between them and some of the general expressive performance rules extracted from the literature.

References

[1] E. Narmour, The Analysis and Cognition of Basic Melodic Structures: The Implication Realization Model, Univ. of Chicago Press, 1990.
[2] G. Widmer, "Machine Discoveries: A Few Simple, Robust Local Expression Principles," *Journal of New Music Research,* vol. 31, no. 1, pp. 37-50, 2002.
[3] G. Berndtsson and J. Sundberg, "The MUSSE DIG singing synthesis, KTH," *SMAC'93*, Royal Swedish Academy of Music no. 79, 1993. C.E. Seashore(ed.), *Objective Analysis of Music Performance*, University of Iowa Press, 1936.
[4] A. Gabrielsson, "The performance of Music," In D. Deutsch (Ed.), *The Psychology of Music* (2nd ed.), Academic Press, 1999.
[5] R. Bresin, "Virtual Visrtuosity: Studies in Automatic Music Performance," PhD Thesis, KTH, Sweden, 2000.
[6] B. H. Repp, "Diversity and Commonality in Music Performance: an Analysis of Timing Microstructure in Schumann's `Traumerei'," Journal *of the Acoustical Society of America,* vol. 92, no. 5, pp. 2546-68, 1992.
[7] N. Todd, "The Dynamics of Dynamics: a Model of Musical Expression," *Journal of the Acoustical Society of America,* vol. 91, no. 6, pp. 3540-50, 1992.

[8] A. Friberg, R. Bresin, and L. Fryden, "Music from Motion: Sound Level Envelopes of Tones Expressing Human Locomotion," *Journal of New* Music *Research,* vol. 29, no. 3, pp. 199-210, 2000.
[9] A. Tobudic and G. Widmer, "Relational IBL in Music with a New Structural Similarity Measure," in *Proceedings of the International Conference on Inductive Logic Programming*, Springer-Verlag, pp. 365-382, 2003.
[10] R. Ramirez, A. Hazan, E. Gomez, and E. Maestre, "Understanding Expressive Transformations in Saxophone Jazz Performances," *Journal of New Music Research*, vol. 34, no. 4, pp. 319-330, 2005.
[11] L. de Mantaras R. and J. L. Arcos, "AI and music, from composition to expressive performance," *AI Magazine*, vol. 23, no. 3, 2002.
[12] M. J. Dovey, "Analysis of Rachmaninoff's Piano Performances Using Inductive Logic Programming," In *ECML*, Springer, 1995.
[13] E. V. Baelen and L. de Raedt, "Analysis and Prediction of Piano Performances Using Inductive Logic Programming," in *International Conference in Inductive Logic Programming*, pp. 55-71, 1996.
[14] E. Morales, "PAL: A Pattern-Based First-Order Inductive System," *Machine Learning*, vol. 26, pp. 227-252, 1997.
[15] M. Alonso, "Expressive performance model for a singing voice synthesizer," Thesis, 2005.
[16] J. Janer, J. Bonada, and M. Blaauw, "Performance-Driven Control for Sample-Based Singing Voice Synthesis," in *Proceedings of the DAFx06*, pp. 42-44, Montreal, 2006.
[17] T. Nakano and M. Goto, "Vocalistener: A singing-to-singing synthesis system based on iterative parameter estimation," in *Proc. of the 6th* Sound *and Music Computing Conference*, pp. 343-348, Porto, 2009.
[18] T. Saitou, M. Unoki, and M. Akagi, "Development of an F0 control model based on F0 dynamic characteristics for singing-voice synthesis," *Speech Communication,* vol. 46, pp. 405-417, 2005.
[19] T. Saitou, M. Goto, M. Unoki, and M. Akagi. "Speech-to-singing synthesis: converting speaking voices to singing voices by controlling acoustic features unique to singing voices," in *Proc. of WASPAA*, pp. 215-218, 2007.
[20] A. Loscos and J. Bonada, "Emulating Rough And Growl Voice In Spectral Domain," in *Proc. of the 7th Int. Conference on Digital Audio Effects (DAFX-04)*, 2004.
[21] K. Saino, M. Tachibana, and H. Kenmochi, "A Singing Style Modeling System for Singing Voice Synthesizers," in *Proceeding of Interspeech*, Chiba, pp. 2894-2897, 2010.
[22] J. Sundberg, A. Friberg, and L. Fryden, *Common secrets of musicians and* listeners*: An analysis-by-synthesis study of musical performance. Representing musical structure*, London: Academic Press, 1991.
[23] A. Friberg, "Generative rules for music performance: A formal description of a rule system," *Computer Music Journal*, vol. 15, no.2, The MIT Press, pp. 56-71, 1991.
[24] X. Serra and S. Smith, "Spectral Modeling Synthesis: A Sound Analysis/Synthesis System Based on a Deterministic plus Stochastic Decomposition," *Computer Music Journal*, vol. 14, no. 4, 1990.
[25] Data Mining Software in Java. [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/
[26] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools* and *techniques*, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.
[27] L. Breiman, "Bagging predictors," *Machine Learning,* vol. 24, no. 2, pp.123-140, 1996.
[28] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," *Annals of Statistics*, vol. 29, no. 5, pp/ 1189-1232, 2001.
[29] A. J. Smola and B. Scholkopf, "A Tutorial on Support Vector Regression," NeuroCOLT2 Technical Report Series - NC2-TR-1998-030, 1998.
[30] J. R. Quinlan, *C4.5: Programs for Machine Learning,* Morgan Kaufmann Publishers, 1993.
[31] J. R. Quinlan, "Learning with continuous classes," in *Proceedings of AI'92, 5th* Australian *Joint Conference on Artificial Intelligence*, Adams & Sterling (eds.), World Scientific, Singapore, pp 343-348, 1992.



Maria-Cristina Marinescu received her PhD degree in computer science in 2002 from University of California, Santa Barbara, USA, and her B.S. in computer science in 1995 from "Politehnica" Institute, Bucharest, Romania. She was a Postdoctoral Fellow at the Massachusetts Institute of Technology until 2003 and a Research Staff Member at IBM T.J. Watson between 2003 and 2008. She is currently a Visiting Professor at Universidad Carlos III de Madrid, Madrid, Spain. Her research interests include machine learning applied to music, programming languages, distributed and embedded systems, and social networks. She has published in conferences and journals in design automation, distributed and embedded systems, bioinformatics, and machine learning applied to music. She holds several US patents. Dr. Marinescu is an ACM and IACSIT member and has served as a program committee member for several conferences and workshops on reconfigurable computing, software maintenance, and machine learning and applications.

Rafael Ramirez is an Associate Professor in the Department of Information and Communications Technology at the Pompeu Fabra University. He obtained a Bachelors degree in Mathematics at the National Autonomous University of Mexico, and his MSc and PhD in computer science from the University of Bristol, UK. From 1997 to 2001, he was a Lecturer in the Department of Computer Science at the School of Computing of the National University of Singapore. His research interests include machine learning and music informatics, concurrency, formal verification, and declarative programming. He has more than 45 international publications on the application of machine learning techniques to music processing.
He is the chair for the series of international Workshops on Music and Machine Learning (MML2008-ICML'08,Finland; MML2009-ECML'09, Slovenia; MML2010-ACM-MM'09, Italy, MML2011-NIPS'11, Spain). He acts as program committee member for several AI and music related conferences, and as a reviewer for several artificial intelligence, and music related journals. He has given invited seminars across Europe, Asia and America.