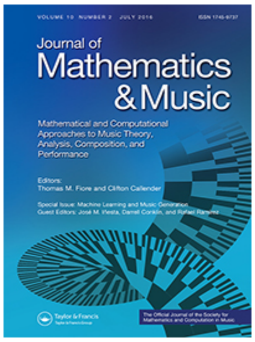


Taylor & Francis
Taylor & Francis Group

Journal of Mathematics and Music

Mathematical and Computational Approaches to Music Theory, Analysis, Composition and Performance

ISSN: 1745-9737 (Print) 1745-9745 (Online) Journal homepage: http://www.tandfonline.com/loi/tmam20

# A machine learning approach to ornamentation modeling and synthesis in jazz guitar

Sergio Giraldo & Rafael Ramírez

**To cite this article:** Sergio Giraldo & Rafael Ramírez (2016) A machine learning approach to ornamentation modeling and synthesis in jazz guitar, Journal of Mathematics and Music, 10:2, 107-126, DOI: 10.1080/17459737.2016.1207814

**To link to this article:** http://dx.doi.org/10.1080/17459737.2016.1207814

*Journal of Mathematics and Music*, 2016
Vol. 10, No. 2, 107–126, http://dx.doi.org/10.1080/17459737.2016.1207814

# A machine learning approach to ornamentation modeling and synthesis in jazz guitar

Sergio Giraldo* and Rafael Ramírez

*Music Technology Group, Universidad Pompeu Fabra, Barcelona, Spain*

(*Received 2 December 2015; accepted 26 May 2016*)

We present a machine learning approach to automatically generate expressive (ornamented) jazz performances from un-expressive music scores. Features extracted from the scores and the corresponding audio recordings performed by a professional guitarist were used to train computational models for predicting melody ornamentation. As a first step, several machine learning techniques were explored to induce regression models for timing, onset, and dynamics (i.e. note duration and energy) transformations, and an ornamentation model for classifying notes as ornamented or non-ornamented. In a second step, the most suitable ornament for predicted ornamented notes was selected based on note context similarity. Finally, *concatenative synthesis* was used to automatically synthesize expressive performances of new pieces using the induced models. Supplemental online material for this article containing musical examples of the automatically generated ornamented pieces can be accessed at doi:10.1080/17459737.2016.1207814 and https://soundcloud.com/machine-learning-and-jazz. In the Online Supplement we present an example of the musical piece *Yesterdays* by Jerome Kern, which was modeled using our methodology for expressive music performance in jazz guitar.

**Keywords:** machine learning; expressive music performance; ornamentation modeling; jazz guitar; concatenative synthesis

## 1. Introduction

*Performance actions* (PAs) can be defined as musical resources used by musicians to add expression when performing a musical piece, which consist of variations in timing, pitch, and energy. In the same context, *ornamentation* can be considered as an expressive musical resource used to embellish and add expression to a melody. In the past, music expression has been mostly studied in the context of classical music, e.g. Puiggròs et al. (2006), in particular classical piano music, e.g. Widmer and Tobudic (2003). Contrary to classical music scores, performance annotations (e.g. ornaments and *articulations* ) are seldom indicated in popular music (e.g. jazz music) scores, and it is up to the performer to include them based on his/her musical background. Therefore, in popular music it may not always be possible to characterize ornaments with the archetypical classical music conventions (e.g. trills and appoggiaturas). Several approaches have been proposed to generate expressive performances in jazz saxophone music, e.g. Arcos, De Mantaras, and Serra (1998), Ramírez and Hazan (2006), and Grachten (2006). Ramírez and Hazan (2006) describe a method to predict ornamentation (among other performance actions). Grachten (2006) detects

*Corresponding author. Email: sergio.giraldo@upf.edu ornaments of multiple notes to render expressive-aware tempo transformations. Other methods are able to recognize and characterize ornamentation in popular music, e.g. Gómez et al. (2011) and Perez et al. (2008). However, due to the complexity of free ornamentation, most of these approaches study ornamentation in constrained settings, for instance by restricting the study to one-note or notated trills ornamentations, e.g. Puiggròs et al. (2006). Based on our previous studies in expressive performance modeling in jazz music (Giraldo 2012; Giraldo and Ramírez 2014, 2015a, 2015b, 2015c, 2015d), this article presents a system for automatically predicting and synthesizing expressive jazz guitar music performances with unrestricted ornamentation.

The aim of this work is twofold: (1) to train computational models of music expression using recordings of a professional jazz guitar player, and (2) to synthesize expressive ornamented performances from inexpressive scores. The general framework of the system is depicted in Figure 1. In order to train a jazz guitar ornamentation model, we recorded a set of 27 jazz standards performed by a professional jazz guitarist. We extracted symbolic features from the scores using information on each note, information on the neighboring notes, and information related to the musical context. The performed pieces were automatically transcribed by applying note segmentation based on pitch and energy information. After performing score-to-performance alignment, using *dynamic time warping* (DTW), we calculated performance actions by measuring the deviations between performed notes and their respective parent notes in the score. For model evaluation, the data set was split using a leave-one-piece-out approach in which each piece was in turn used as test set, using the remaining pieces as training set. The expressive

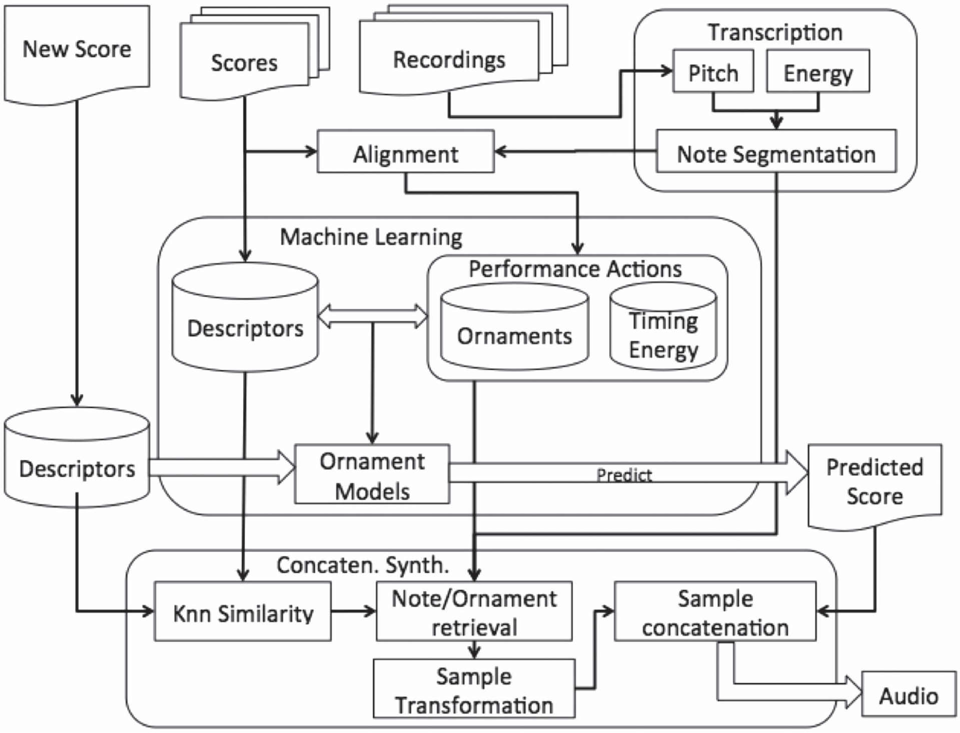


Figure 1. General framework for jazz guitar ornament modeling.

actions considered in this article were duration, onset, energy, and ornamentation transformations. *Concatenative synthesis* was used to synthesize new ornamented jazz melodies using samples of adapted notes/ornaments from the segmented audio recordings.

The rest of the article is organized as follows. Section 2 surveys related work. Section 3 describes data acquisition. Section 4 presents our machine learning approach to ornament prediction. Section 5 describes the audio synthesis method. Section 6 reports on the results, and finally, Section 7 presents some conclusions and future work.

## 2. Related work

Expressive music performance studies the micro variations a performer introduces (voluntary or involuntary) when performing a musical piece to add expression. Several studies investigating this phenomenon have been conducted, e.g. Gabrielsson (1999, 2003) and Palmer (1997). Computational approaches to studying expressive music performance have been proposed in which data are extracted from real performances and used to formalize expressive models for different aspects of performance – for an overview see Goebl et al. (2008). *Computational systems for expressive music performance* (CEMP) are often targeted at automatically generating human-like performances by introducing variations in timing, energy, and articulation (Kirke and Miranda 2013).

Two main approaches have been used to model expression computationally. On one hand, expert-based systems obtain their rules manually from music experts. A relevant example is the work of the KTH group (Bresin and Friberg 2000; Friberg, Bresin, and Sundberg 2006; Friberg 2006). Their *Director Musices* system incorporates rules for tempo, dynamic, and articulation transformations. Other examples of manually generated expressive systems are the Hierarchical Parabola Model (Todd 1989, 1992, 1995) and the work of Johnson (1991), who developed a rule-based expert system to determine expressive tempo and articulation for Bach's fugues from *The Well-Tempered Clavier*. The rules were obtained from two expert performers. On the other hand, machine-learning-based systems obtain their expressive models from real music performance data by measuring the deviations of a human performance with respect to a neutral or *robotic* performance, using computational learning tools. For example, neural networks were used by Bresin (1998) to model piano performances, and by Camurri, Dillon, and Saron (2000) to model emotional flute performances. Rule-based learning algorithms were used by Widmer (2003) to cluster piano performance rules. Other piano expressive performance systems worth mentioning are the ESP piano system by Grindlay (2005), which utilizes Hidden Markov Models, and the generative performance system of Miranda, Kirke, and Zhang (2010), which uses genetic algorithms to construct tempo and dynamic curves.

Most of the proposed expressive music systems are targeted at classical piano music. More recently, there have been several approaches to computationally modeling expressive performance in popular music by applying machine learning techniques. Arcos, De Mantaras, and Serra (1998) report on *SaxEx*, a performance system capable of generating expressive solo saxophone performances in jazz, based on case-based reasoning. Ramírez and Hazan (2006) compare different machine learning techniques to obtain jazz saxophone performance models capable of both automatically synthesizing expressive performances and explaining expressive transformations. Grachten (2006) applies dynamic programming using an extended version of edit distance and case-based reasoning to detect multiple note ornaments and render expressive-aware tempo transformations for jazz saxophone music. In previous work (Giraldo 2012; Giraldo and Ramírez 2015a, 2015b, 2015c), ornament characterization in jazz guitar performances is accomplished using machine learning techniques to train models for note ornament prediction.

## 3. Data acquisition

In this study, the data set consisted of 27 jazz standard audio recordings (resulting in a total of 1368 notes) recorded by a professional jazz guitarist, and their corresponding music scores. Each note in the score of the recorded pieces was characterized by a set of 30 descriptors. The music scores and audio recordings were analyzed as explained in Sections 3.1 and 3.2, respectively.

### 3.1. *Score analysis*

Music scores were obtained from commercially available compilations of jazz scores (The Real Book Series). Selected scores were rewritten using an open source software for music notation, and saved into *MusicXML* format. The MusicXML format allows not only information about the notes (pitch, onset, and duration) to be stored, but also other relevant information for note description such as chords, key, and tempo (among others).

#### 3.1.1. *Feature extraction*

Feature extraction was performed following an approach similar to that of Giraldo (2012), in which each note is characterized by its *nominal*, *neighboring*, and *contextual* properties.

- *Nominal* descriptors refer to the intrinsic properties of score notes (e.g. pitch, duration, and onset). Duration and onsets were described both in beats and seconds, as the duration in seconds depends on the tempo of the piece. For example the choice of ornamenting two different notes from different pieces with quarter note duration (beats) may differ if the pieces are played at slow and fast tempos. The energy descriptor refers to the loudness of the note, which in MIDI format is measured as velocity (how fast a piano key was pressed).
- Given a particular note, its *neighboring* descriptors refer to the properties of its neighboring notes, e.g. previous/next interval, previous/next duration ratio, previous/next inter-onset interval. In this work, only one previous and one following note were considered. Inter-onset distance (Giraldo and Ramírez 2015b) refers to the onset difference between two consecutive notes.
- *Contextual* descriptors refer to the musical *context* in which the note occurs, e.g. tempo, chord, and key. The phrase descriptor (Giraldo and Ramírez 2015b) refers to the note position within a phrase: initial, middle, or end. Phrase descriptors were obtained using the melodic segmentation approach of Cambouropoulos (1997), which indicates the probability of each note being at a phrase boundary. Probability values were used to decide if the note was a *boundary note*, annotated as either *initial (i)* or *ending (e)*. Non-boundary notes were annotated as *middle (m)*. The phrase descriptor was introduced based on the hypothesis that boundary notes (i.e. initial or ending phrase notes) are more prone to be ornamented than middle notes. *Note to key* and *note to chord* descriptors are intended to capture harmonic analysis information, as they refer to the interval of a particular note with respect to the key and to the chord root, respectively. *Key* and *mode* refer to the key signature of the song (e.g. key: $C$, mode: major). Mode is a binary descriptor (major or minor), whereas key is represented numerically in the circle of fifths (e.g. $B\flat = -1$, $C = 0$, $F = 1$, etc.). However, for some calculations (e.g. note to key in Table 3) a linear representation of the notes (e.g. $C = 0$, $C\sharp/D\flat = 1$, $D = 2$, etc.) is used for key. Also, it is worth noticing that the key descriptor may have 13 possible values as the extreme values (−6 and 6) corresponding to enharmonic tonalities ($G\flat$ and $F\sharp$ ). The descriptor *Is chord note* was calculated using the chord type description of Table 1 in which each of the notes of the chord is shown using the aforementioned linear note representation. If a note corresponds to any of the notes included in the chord type description it is labeled *yes*.

Table 1. Chord description list.

| Chord type | Intervals |
|---|---|
| major | 0 4 7 |
| m (minor) | 0 3 7 |
| 2 (sus2) | 0 2 7 |
| sus (4) | 0 5 7 |
| Maj7 | 0 4 7 11 |
| 6th | 0 4 7 9 |
| m7 | 0 3 7 10 |
| m6 | 0 3 7 9 |
| mMaj7 | 0 3 7 11 |
| m7♭5 | 0 3 6 10 |
| dim | 0 3 6 9 |
| 7th | 0 4 7 10 |
| 7♯5 | 0 4 8 10 |
| 7♭5 | 0 4 6 10 |
| 7sus | 0 5 7 10 |
| Maj9 | 0 2 4 7 11 |
| m9 | 0 2 3 7 11 |
| 6/9 | 0 2 4 7 9 |
| m6/9 | 0 2 3 7 9 |
| 9th | 0 2 4 7 10 |
| 7♭9 | 0 1 4 7 10 |
| 7♯9 | 0 3 4 7 10 |
| 13 | 0 2 4 7 9 10 |
| 7♭9♭13 | 0 1 4 7 8 10 |
| 7alt | 0 1 3 4 6 8 10 |

Other relevant descriptors used for this work include categorization based on the *implication–realization* (I-R) model of Narmour (1992). Grachten (2006) parses the melodies and obtains for each note the label of the I-R structure to which it belongs. The concept of closure was based on metrical position and duration. The basic Narmour structures (P, D, R, and ID) and their derivatives (VR, IR, VP, and IP) are represented in Figure 2.

The *metrical strength* concept refers to the rhythmic position of the note inside the bar (Cooper and Meyer 1960). Four levels of metrical strength were used to label notes in three common time signatures, depending on the beat at which the note occurs, as shown in Table 2.

The complete list of the 30 descriptors used for this study and its definition is summarized in Table 3.

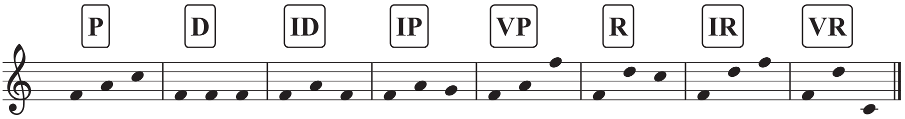


Figure 2. Basic Narmour structures P, D, R, and ID, and their derivatives VR, IR, VP, and IP.

Table 2. Metrical strength at beat occurrence, for different time signatures.

| Time signature | Very strong | Strong | Weak | Very weak |
|---|---|---|---|---|
| 4/4 | Beat 1 | Beat 3 | Beats 2 and 4 | Other |
| 3/4 | Beat 1 | None | Beats 2 and 3 | Other |
| 6/8 | Beat 1 | Beat 2.5 | Beats 1.5, 2, 3, and 3.5 | Other |

Table 3. Features extracted from music scores – in the fifth row, in the column headed ‘Formula,’ ‘*bpb*’ means *beats per bar* according to the time signature.

| | Descriptor | Abbreviation | Units | Formula | Range |
|---|---|---|---|---|---|
| Nominal | Duration | $ds_n$ | Seconds | $ds_0$ | $[0, +\infty]$ |
| | Duration | $db_n$ | Beats | $db_0$ | $[0, +\infty]$ |
| | Onset | $ons_n$ | Seconds | $os_0$ | $[0, +\infty]$ |
| | Onset | $onb_n$ | Beats | $ob_0$ | $[0, +\infty]$ |
| | Onset in bar | $obm_n$ | Beats | $ob_0\%bpb$ | $[0, +bpb]$ |
| | Pitch | $p_n$ | Semitones | $p_0$ | $[1, 127]$ |
| | Chroma | $ch_n$ | Semitones | $p_0\%12$ | $[0, 11]$ |
| | Energy | $v_n$ | MIDI vel | $v_0$ | $[1, 127]$ |
| Neighbor | Prev. duration | $pds_n$ | Seconds | $ds_{-1}$ | $[0, +\infty]$ |
| | Prev. duration | $pdb_n$ | Beats | $db_{-1}$ | $[0, +\infty]$ |
| | Next duration | $nds_n$ | Seconds | $ds_1$ | $[0, +\infty]$ |
| | Next duration | $ndb_n$ | Beats | $db_1$ | $[0, +\infty]$ |
| | Prev. interval | $pint_n$ | Semitones | $p_{-1} - p_0$ | $[-60, 60]$ |
| | Next interval | $nint_n$ | Semitones | $p_1 - p_0$ | $[-60, 60]$ |
| | Prev. inter-onset dist. | $piod_n$ | Seconds | $os_0 - os_{-1}$ | $[0, +\infty]$ |
| | Next. inter-onset dist. | $piod_n$ | Seconds | $os_1 - os_0$ | $[0, +\infty]$ |
| | Narmour | $nar1_n$ | Label | $nar(p_{-1}, p_0, p_1)$ | {P, D, R, ID, (P), (R), (ID), |
| | | $nar2_n$ | | $nar(p_{-2}, p_{-1}, p_0)$ | VR, IR, VP, IP, (VR), (IR), |
| | | | | | (VP), (IP), |
| | | $nar3_n$ | | $nar(p_0, p_1, p_2)$ | dyadic,′ monadic, none} |
| Context | Measure | $m_n$ | Bars | $m_0$ | $[0, +\infty]$ |
| | Tempo | $t_n$ | Bpm | $t_0$ | $[30, 260]$ |
| | Key | $k_n$ | Semitones | $k_0$ | $[-6, 6]$ |
| | Mode | $mod_n$ | Label | $mod_0$ | {major, minor} |
| | Note to key | $n2k_n$ | Semitones | $ch_0 - k_0$ (linear) | $[0, 11]$ |
| | Chord root | $chr_n$ | Semitones | $chr_0$ | $[0, 11]$ |
| | Chord type | $cht_n$ | Label | $cht_0$ | { + , 6, 7, 7♯11, 7♯5, 7♯9, 7alt,<br>7♭5, 7♭9, Maj7, dim, dim7,<br>m, m6, m7, m7♭5, major} |
| | Note to chord | $n2ch_n$ | Semitones | $ch_0 - chr_0$ | $[0, 11]$ |
| | Is chord note | $ichn_n$ | Boolean | $isChNote(chr_0, cht_0, ch_0)$ | {true, false} |
| | Metrical strength | $mtr_i$ | Label | $metStr_0$ | {Very strong, Strong,<br>Weak, Very weak} |
| | Phrase | $ph_n$ | Label | $phrase_0$ | {initial, middle, final} |

## 3.2. *Audio analysis*

The audio of the performed pieces was recorded from the raw signal of an electric guitar. The guitarist was instructed not to strum chords or play more than one note at a time. The guitarist recorded the pieces while playing along with prerecorded commercial accompaniment backing tracks (Kennedy and Kernfeld 2002). We opted to use audio backing tracks performed by professional musicians, as opposed to synthesized MIDI backing tracks, in order to provide a more natural and ecologically valid performance environment. However, using audio backing tracks required a preprocessing beat tracking task. Each piece’s section was recorded once (i.e. no repetitions or solos were recorded), For instance, for a piece consisting of sections *AABB*, only sections *A* and *B* were considered.

### 3.2.1. *Melodic transcription*

The monophonic audio signal recorded from the guitar was parsed to automatically obtain a *MIDI* type transcription of the notes performed by the guitarist, based on the previous work of Bantula, Giraldo, and Ramírez (2014). This representation includes the pitch, onset, duration, and energy of each note. For doing this, the audio signal was segmented based on the pitch and energy profiles obtained with the YIN algorithm (De Cheveigné and Kawahara 2002). Each segment represents a note with its corresponding information on pitch, onset, and offset (therefore duration). To minimize transcription errors, the resulting segments (notes) were filtered using heuristic rules based on human perceptual thresholds for minimum note duration and minimum note gaps. Also, rules to detect octave errors or unusual note intervals were used. To obtain temporal information on the recordings, beat tracking (Zapata et al. 2012) was used, as there was uncertainty concerning the use of a metronome in the recordings of the accompaniment backing tracks. After manual correction of beat tracking, the onset and duration information on each note was adjusted to the beat grid detected for each piece.

### 3.3. *Score to performance alignment*

Score to performance alignment was performed to correlate each performed note with its respective *parent* note in the score as depicted in Figure 3. This procedure was carried out following the approach of Giraldo and Ramírez (2015d), in which DTW techniques were used to match performance and score note sequences. A similarity cost function was designed based on pitch, duration, onset, and phrase onset/offset deviations.

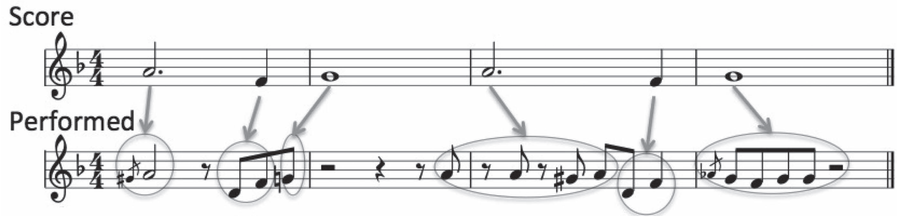


Figure 3. Score to performance alignment. Fragment of *Yesterdays* (J. Kern) as performed by Wes Montgomery.

Phrase onset and offset deviation were introduced to force the algorithm to map all the notes of a particular short ornament phrase (*lick*) to one parent note in the score. We assumed that a group of notes conforming a *lick* are played *legato*. Therefore, the performed sequence is segmented in phrases in which the time gap between consecutive notes is less than 50 ms. This threshold was chosen based on human time perception studies (Woodrow 1951).

Each note from the score and the corresponding performed sequence is represented by a five position *cost vector* as

$$cs = (p(i), ds(i), ons(i), ons(i), ofs(i)) \tag{1}$$

and

$$cp = (p(j), ds(j), ons(j), ph_{ons}(j), ph_{ofs}(j)), \tag{2}$$

respectively, where $cs$ is the *score cost vector* and $cp$ is the *performance cost vector*. Index $i$ refers to a note position in the score sequence, and $j$ refers to a note position at the performed sequence. The onset of the first note of the lick phrase in which the $j$th note of the performance sequence occurs is represented by $ph_{ons}(j)$. Similarly $ph_{ofs}(j)$ refers to the offset of the last note of the lick phrase in which the $j$th note of the performance sequence occurs.

The total cost is calculated using the *Euclidean distance* as follows:

$$cost(i,j) = \sqrt{\sum_{n=1}^{5} (cs(n)_i - cp(n)_j)^2}\,. \tag{3}$$

Notice that in equation (3) phrase onset and offset deviations are calculated when $n$ equals four and five.

Finally, we apply DTW: a similarity matrix $H_{(m \times n)}$ is defined in which $m$ is the length of the performed sequence of notes and $n$ is the length of the sequence of score notes. Each cell of the matrix $H$ is calculated as follows:

$$H_{i,j} = cost + min(H_{i-1,j}, H_{i,j-1}, H_{i-1,j-1}) \tag{4}$$

where *min* is a function that returns the minimum value of the preceding cells (up, left, and up-left diagonal). The matrix $H$ is indexed by the note position of the score sequence and the note position of the performance sequence.

A backtrack path is obtained by finding the lowest cost calculated in the similarity matrix. Starting from the last score/performance note cell, the cell with the minimum cost at positions $H_{(i-1)}$, $H_{(i,j-1)}$, and $H_{(i-1,j-1)}$ is stored in a backtrack path array. The process iterates until indexes arrive to the first position of the matrix, assigning each note in the performance to a parent note in the score.

Figure 4 presents an example of the resulting similarity matrix obtained for one of the recorded songs. The $x$-axis corresponds to the sequence of notes of the score and the $y$-axis corresponds to the sequence of performed notes. The cost of correspondence between all possible pairs of notes is depicted darker for the highest cost (less similar) and lighter for the lowest cost (most similar).

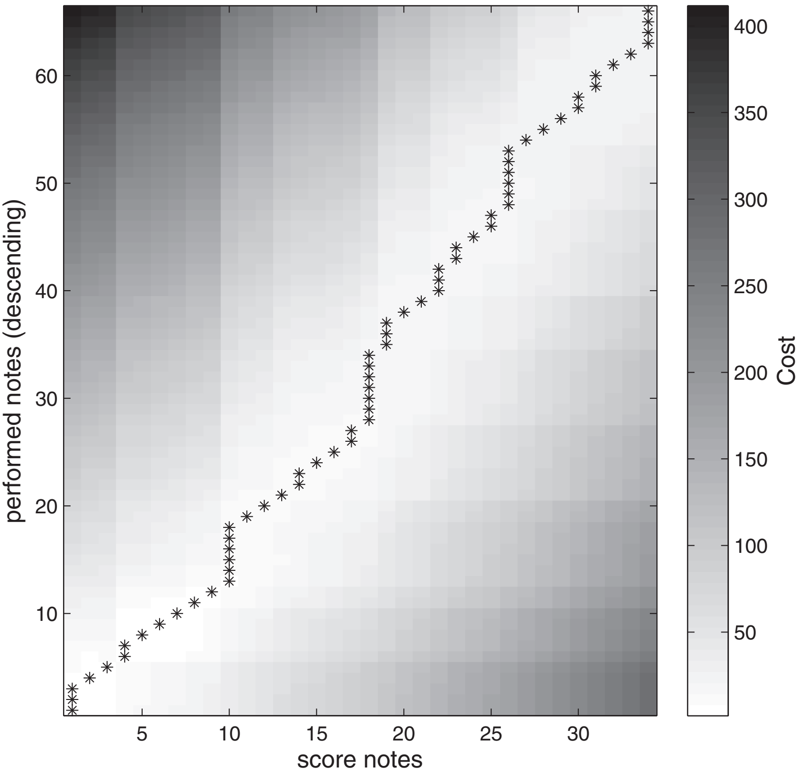


Figure 4. Similarity matrix of performed notes and score notes. Dots indicate alignment path between score and performance notes.

The dots on the graph show the backtrack path (or optimal path) found for alignment. Diagonal lines represent notes which were not ornamented, as the correspondence from the performance notes to the parent score notes is one to one. On the contrary, vertical lines represent ornaments, as two or more performed notes correspond to one parent note in the score.

Because there are no concrete rules to map performance notes to *parent* score notes, our alignment algorithm was evaluated by comparing its output with the level of agreement between five human experts who were asked to align performance and score note sequences manually. Accuracy of the system was estimated by quantifying how much each note pair produced by the algorithm agreed with the human experts, using penalty factors for high, medium, and low agreement. The results of the evaluation showed that the performance of our approach was comparable with that of the human annotators. Details of these evaluations can be found in Giraldo and Ramírez (2015d).

### 3.4. *Performance actions*

After alignment, score notes which were mapped to only one performance note were labeled as *non-ornamented*, whereas score notes mapped to several performance notes (or which were omitted in the performance) were mapped as *ornamented*. Performance actions were calculated

Table 4. Expressive performance actions calculation.

| PA | Abbreviation | Units | Formula | Range |
|---|---|---|---|---|
| Ornamentation | $Orn_n$ | Boolean | $ornament(N_n)$ | {yes, no} |
| Duration ratio | $Dr_n$ | Percentage | $\frac{db_j}{db_i} * 100$ | $[0, +\infty]$ |
| Onset deviation | $Od_n$ | Beats | $ob_j - ob_i$ | $[0, +\infty]$ |
| Energy ratio | $Er_n$ | Percentage | $\frac{v_j}{70} * 100$ | $[0, +\infty]$ |

Table 5. Example of ornament annotation for the music excerpt of Figure 3.

| Score note index ($i$) | Performance note index ($j$) | Pitch deviation (semitones) | Onset deviation (beats) | Duration ratio (beat fraction) |
|---|---|---|---|---|
| 1 | 1 | −1 | −1/2 | 1/16 |
| 1 | 2 | 0 | 0 | 2/3 |
| 2 | 3 | −3 | −1/2 | 1/2 |
| 2 | 4 | 0 | 0 | 1/2 |
| 4 | 6 | 0 | −1/2 | 1/16 |
| 4 | 7 | 0 | 1/2 | 1/16 |
| 4 | 8 | −1 | 3/2 | 1/16 |
| 4 | 9 | 0 | 2 | 1/16 |
| 5 | 10 | −3 | −1/2 | 1/2 |
| 5 | 11 | 0 | 0 | 1 |
| 6 | 12 | 1 | −1/2 | 1/8 |
| 6 | 13 | 0 | 0 | 1/8 |
| 6 | 14 | −2 | 1/2 | 1/8 |
| 6 | 15 | 0 | 1 | 1/8 |
| 6 | 16 | 0 | 3/2 | 1/8 |

for each score note, as defined in Table 4, by measuring the deviations in onset, energy, and duration. Again, indexes *i* and *j* refer to the note position at the score and the performance sequence, respectively.

### 3.5. *Database construction*

The data collected was organized, storing each note descriptors along with its corresponding *performance action*. The *pitch*, *duration*, *onset*, and *energy* deviations of each ornament note with respect to the score *parent* note were annotated as shown in Table 5.

## 4. Expressive performance modeling

Several machine learning algorithms – i.e. *artificial neural networks* (ANNs), *decision trees* (DTs), *support vector machines* (SVMs), and *k-nearest neighbor* (k-NN) – were applied to predict the ornaments introduced by the musician when performing a musical piece. The accuracy of the resulting classifiers was compared with the *baseline classifier*, i.e. the classifier which always selects the most common class. Timing, onset, and energy performance actions were modeled by applying several regression machine learning methods (i.e. ANNs, *regression trees* (RTs), SVMs, and k-NN).

Based on their accuracy, we chose the best performance model and feature set to predict the different performance actions. Each piece (used as test set) was in turn predicted based on the models obtained with the remaining pieces (used as training set) and synthesized using a concatenative synthesis approach.

### 4.1. *Algorithm comparison*

In this study we compared four classification algorithms for ornament prediction and four regression algorithms for duration ratio, onset deviation, and energy ratio. We used the implementation of the machine learning algorithms provided by the WEKA library (Hall et al. 2009). We applied $k$-NN with $k = 1$, SVM with linear kernel, ANN consisting of a fully-connected multi-layer neural network with one hidden layer, and DT/RT with post pruning.

A paired T-test with a significance value of 0.05 was performed for each algorithm for the ornamentation classification task, over all the data set with 10 runs of 10-fold cross validation scheme. Experiments results are presented in Table 8 and will be commented in Section 6.

### 4.2. *Feature selection*

Both filter and wrapper feature selection methods were applied. Filter methods use a proxy measure (e.g. information gain) to score features, whereas wrappers make use of predictive models to score feature subsets. Features were filtered and ranked by information gain values, and a wrapper with greedy search and decision trees accuracy evaluation was used to select optimal feature subsets. We used the implementation of these methods provided by WEKA library (Hall et al. 2009). Selected features are shown in Table 6, and will be commented upon in Section 6.

Learning curves on the number of features, as well as on the number of instances, were obtained to measure the learning rate of each of the algorithms. The selection of the model was based on the evaluation obtained with these performance measures.

Table 6. Features selected using the Filter and Wrapper (with J48 classifier) methods.

| Info-Gain + Ranker | Wrapper + Greedy |
|---|---|
| Duration (sec) | Duration beat |
| Duration (beat) | Prev. duration (sec) |
| Phrase | Tempo |
| Prev. duration (sec) | Phrase |
| Onset in bar | Velocity |
| Metrical strength | Onset beat |
| Prev. duration (beat) | Duration (sec) |
| Next duration (beat) | Prev. duration (beat) |
| Next duration (sec) | Next duration (beat) |
| Narmour 1 | Narmour 3 |
| Tempo | Metrical strength |
| Chord type | Onset in bar |
| Prev. interval | Narmour 1 |
| Next interval | Prev. interval |
| Narmour 2 | Narmour 2 |
| Narmour 3 | Is chord note |
| Is chord note | Onset (sec) |
| Mode | Key |
| keyMode | Chord root |
| | Next duration (sec) |
| | Pitch |
| | Note to key |
| | Note to chord |
| | Measure |
| | Next interval |
| | Chroma |
| | Chord type |

## 5. Synthesis

Predicted pieces were created in both MIDI and audio formats. A concatenative synthesis approach was used to generate the audio pieces. This process consists of linking note audio samples from real performances to render a synthesis of a musical piece. The use of this approach was possible because we had monophonic performance audio data from which onset and offset information was extracted based on energy and pitch information as described in Section 3.2.1. Therefore, it was possible to segment the audio signal into individual notes, and furthermore to obtain complete audio segments of ornaments.

Similarly to the evaluation of the different machine learning algorithms, for synthesis we followed a leave-one-piece-out scheme in which, on each fold, the notes of one piece were used as test set, whereas the notes of the remaining 26 songs were used as training set.

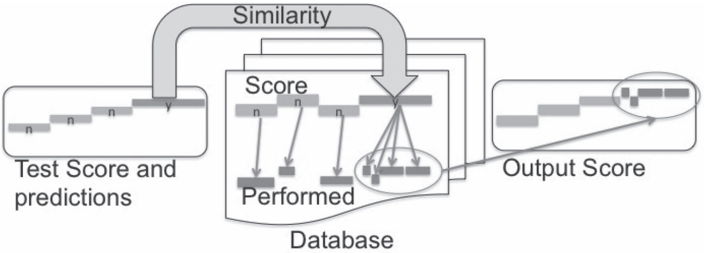
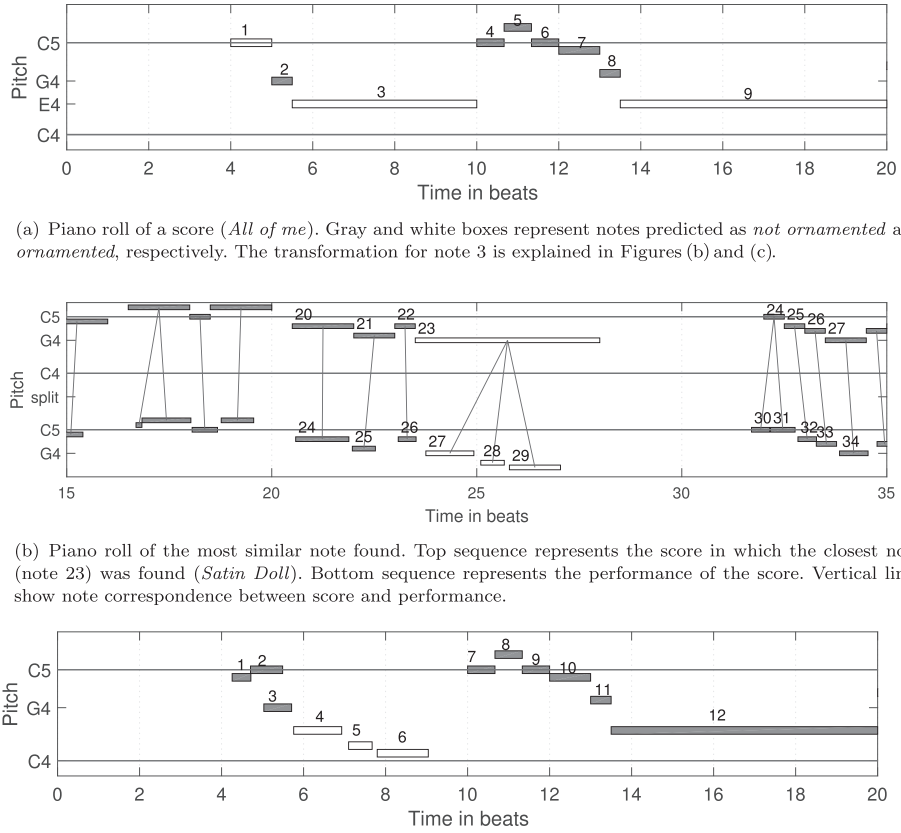


Figure 5. Concatenative synthesis approach.

### 5.1. *Note concatenation*

The note concatenation process is divided into three different stages as depicted in Figure 5.

- *Sample retrieval* For each note predicted to be ornamented, the k-NN algorithm, using a *Euclidean* distance similarity function based on note description, was applied to find the most suitable ornamentation in the database (see Section 3.5). This was done by searching for the most suitable ornament in the songs in the training set (Section 4.2).
- *Sample transformation* For each note classified as ornamented, transformations in duration, onset, energy, and pitch (in the case of ornaments) were performed based on the deviations stored in the database, as seen in Figures 6(a) and 6(b). For audio sample transformation we used the time and pitch scaling approaches of Serra (1997). Notes classified as not ornamented were simply transformed as predicted by the duration, onset, and energy models.
- *Sample concatenation* Retrieved samples were concatenated based on final onset and duration information after transformation. The tempo of the score being predicted (in BPM), was imposed on all the retrieved notes.



(a) Piano roll of a score (*All of me*). Gray and white boxes represent notes predicted as *not ornamented* and *ornamented*, respectively. The transformation for note 3 is explained in Figures (b) and (c).

(b) Piano roll of the most similar note found. Top sequence represents the score in which the closest note (note 23) was found (*Satin Doll*). Bottom sequence represents the performance of the score. Vertical lines show note correspondence between score and performance.

(c) Piano roll of a partial predicted score (*All of me*). Note number 3 of Figure (a) has been replaced by notes 27, 28, and 29 of Figure (b), obtaining notes 4, 5, and 6.

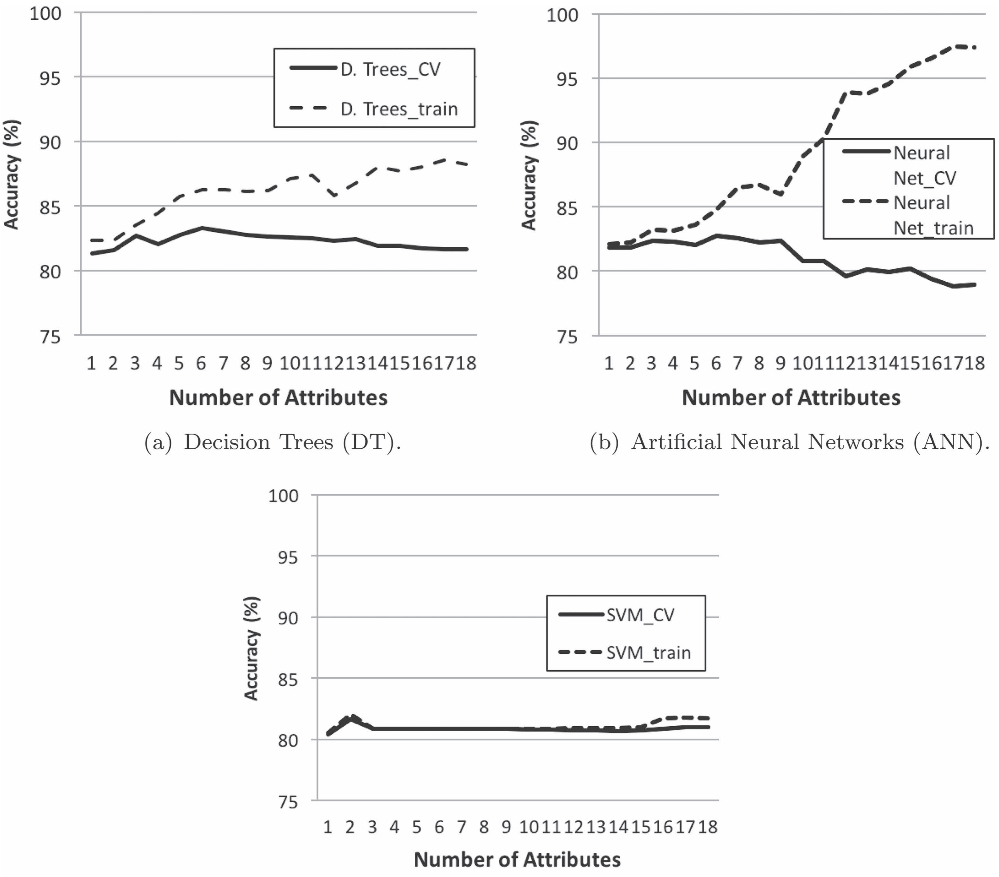
Figure 6. Sample retrieval and concatenation example.

## 6. Results

### 6.1. *Feature selection*

The most relevant features found using the two selection methods described in Section 4.2 are shown in Table 6. The average correctly classified instances percentage (C.C.I.%) obtained using the features selected by the information gain filtering and the greedy search (decision trees) wrapper methods were 78.12 and 78.60%, respectively (F-measures of 0.857 and 0.866, respectively). Given that both measures are similar, i.e. not significantly different, the smallest subset was chosen.

In Figure 7, the accuracy on increasing the number of features based on the information gain ranking (explained in Section 4.2) is presented for each of the algorithms used (SVM, ANN, DT). From the curves it can be seen that the subset with the first three features contains sufficient information, as additional features do not add significant accuracy improvement. SVM exhibits better accuracy on the cross validation scheme, and less over-fitting based on the difference between *cross validation* (CV) and *training set* (TS) accuracy curves.



(a) Decision Trees (DT).

(b) Artificial Neural Networks (ANN).

(c) Support Vector Machines (SVM).

Figure 7. Accuracies on increasing the number of attributes.

## 6.2. *Quantitative evaluation*

### 6.2.1. *Algorithm comparison results*

For the (ornament) classification problem we compared each of the algorithms (SVM, DT, ANN, and k-NN) with the baseline classifier (i.e. the majority class classifier) following the procedure explained in Section 4.1. From Table 7 it can be seen that all the algorithms present a statistically significant improvement, except k-NN. Given the accuracy results, we apply the ornamentation prediction model induced by the DT algorithm to determine whether a note is to be ornamented or not. We discarded the use of k-NN for this task due to its low accuracy, which led to larger mis-classifications of ornamented and not ornamented notes.

For the regression problems (duration, onset, and energy prediction) we applied *regression trees*, *SVM*, *neural networks*, and *k-NN*, and obtained the correlation coefficient values shown in Table 8. *Onset deviation* has the highest correlation coefficient, close to 0.5.

Table 7. *Correctly classified instances* (CCIs) (%) comparison with paired T-test for classification task.

| Dataset | Baseline classifier (CCIs) ( %) | Instance base learner (CCIs) ( %) | Design trees (CCIs) ( %) | | SVM (CCIs) ( %) | | Neural Network (CCIs) ( %) | |
|---|---|---|---|---|---|---|---|---|
| Ornament | 72.74 | 70.58 | 78.68 | ∘⋄ | 77.64 | ∘⋄ | 76.60 | ∘⋄ |

∘/• Statistically significant improvement/degradation against 'Baseline classifier.'
⋄/⋆ Statistically significant improvement/degradation against 'Instance base learner.'

Table 8. *Pearson correlation coefficient* (PCC) and *explained variance* ($R^2$) for the regression task.

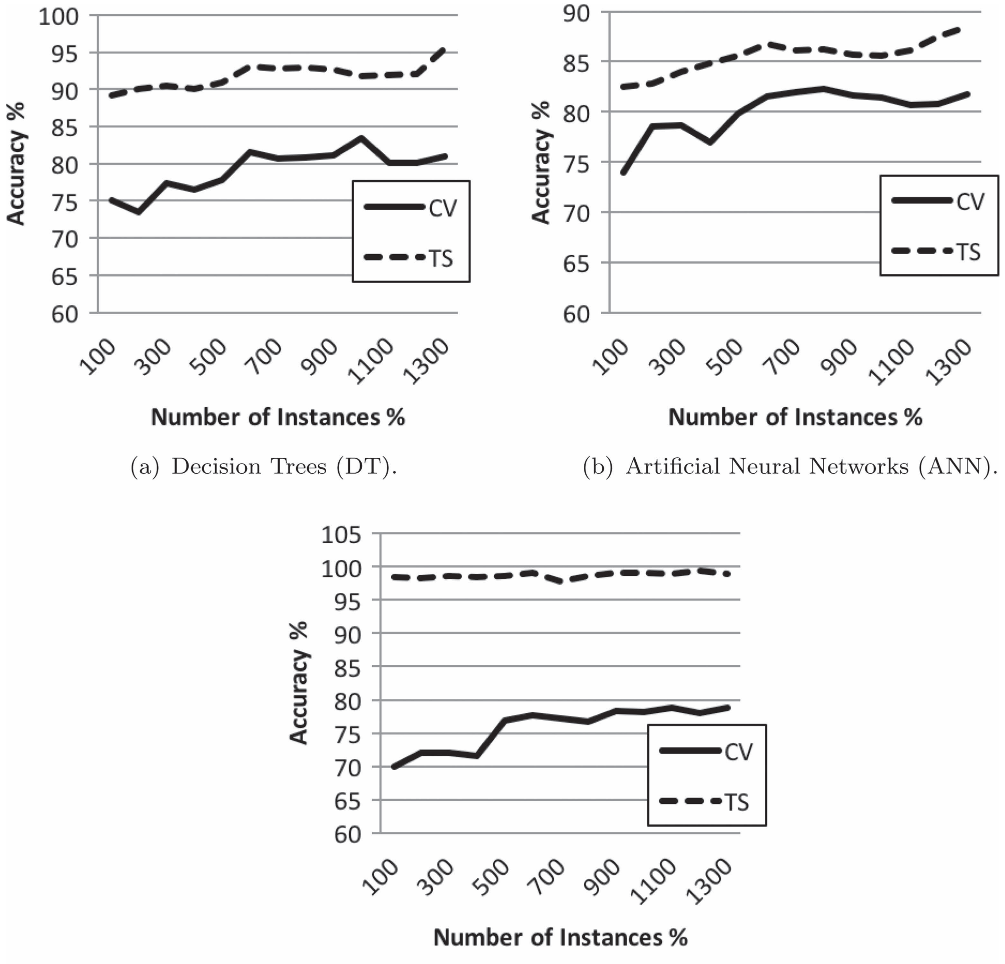
| | k-NN | | | Reg. Trees | | | Reg. SVM | | ANN | | $PA_{mean}$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Dataset | PCC | $R^2$ | | PCC | $R^2$ | | PCC | $R^2$ | PCC | $R^2$ | PCC | $R^2$ |
| Duration ratio | 0.20 | 0.04 | | 0.25 | 0.06 | | 0.17 | 0.03 | 0.19 | 0.04 | 0.20 | 0.04 |
| Energy ratio | 0.19 | 0.04 | | 0.37 | 0.14 | | 0.38 | 0.14 | 0.37 | 0.14 | 0.33 | 0.11 |
| Onset deviation | 0.41 | 0.17 | | 0.51 | 0.26 | | 0.43 | 0.18 | 0.44 | 0.19 | 0.45 | 0.20 |
| $Algorithm_{mean}$ | 0.25 | 0.08 | 0.38 | 0.15 | 0.33 | 0.12 | 0.33 | 0.12 | | | | |

For ornamentation classification using k-NN we explored several values for $k$ ($1 \leq k \leq 10$). However, all of the explored values for $k$ resulted in inferior classification accuracies when compared with decision trees and SVM. As in the case of $k = 1$, both the decision trees and SVM classifiers resulted in statistically significantly higher accuracies (based on the T-test) when compared with the classifiers for $2 \leq k \leq 10$.

## 6.3. *Learning curves*

The learning curves of accuracy improvement, for both cross validation and training sets, over the number of instances are shown in (Figure 8). The learning curves were used to measure the learning rate and estimate the level of overfitting. Data subsets of different sizes (in steps of 100 randomly selected instances) were considered and evaluated using 10-fold cross validation. In general, for the three models, it can be seen that the accuracy on CV tends to have no significant improvement above 600 instances.

Overfitting can be correlated with the difference between the accuracies of CV and TS, wherein a high difference means higher levels of overfitting. In this sense, in Figure 8(c), SVM shows a high tendency for overfitting, but seems to improve slowly over the number of instances. On the other hand, in Figures 8(a) and 8(b), ANN and DT seem to improve overfitting between 700 and 1100 instances. This could mean that adding more instances may slightly improve the accuracy of both CV and TS for the three models, and may slightly improve overfitting for SVM, but this may not be the case for ANN and DT.



(a) Decision Trees (DT).

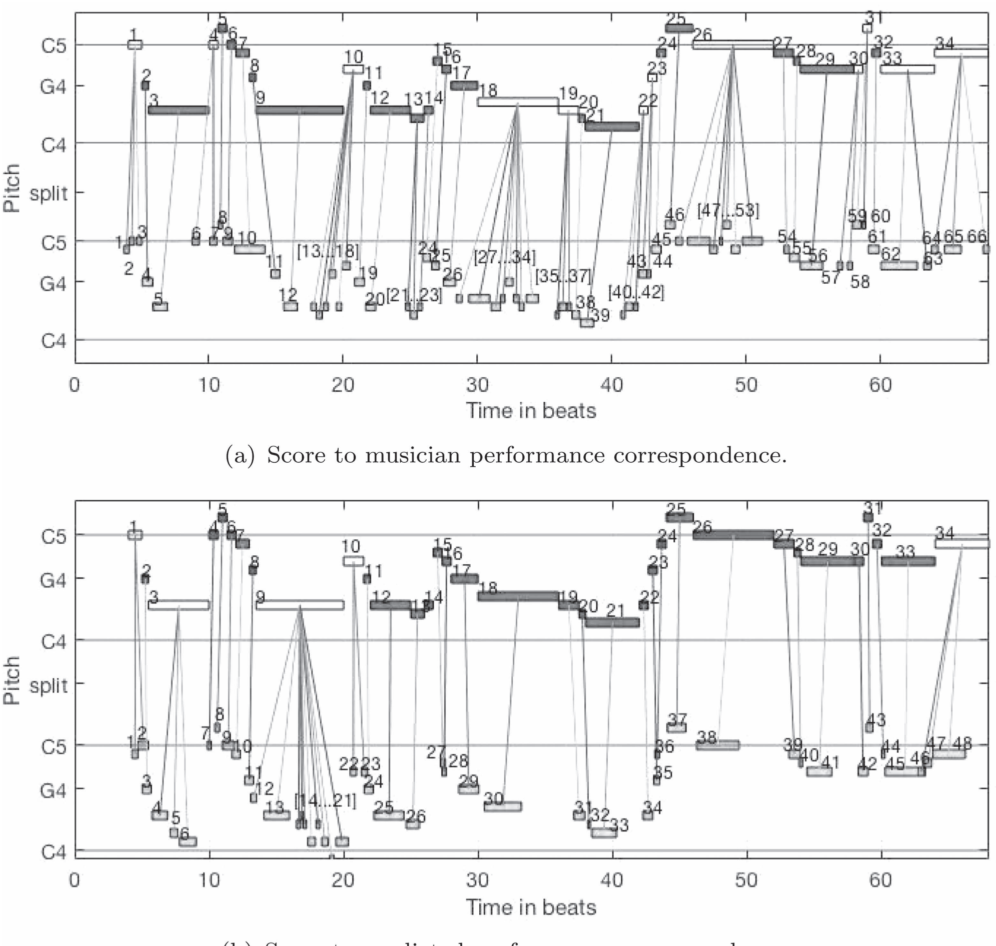
(b) Artificial Neural Networks (ANN).

(c) Support Vector Machines (SVM).

Figure 8. Accuracies on increasing the number of instances.

### 6.4. *Obtained pieces*

Figure 9 shows a MIDI piano roll of an example piece performed by a professional musician and the predicted performance obtained by the system, using a decision trees classifier. It can be noticed how the predicted piano roll follows a similar melodic structure as the one performed by the musician. For instance, for the score notes predicted correctly as ornamented (true positives), notes 1, 10, and 34 in Figure 9(a) (top sequence), the system finds ornaments of similar duration, offset, and number of notes as the musician's performance. Also, score notes 3 and 9 of Figure 9(b) (false positives), are ornamented similarly as score notes 18 and 26 (Figure 9(a)), which are in a similar melodic context.



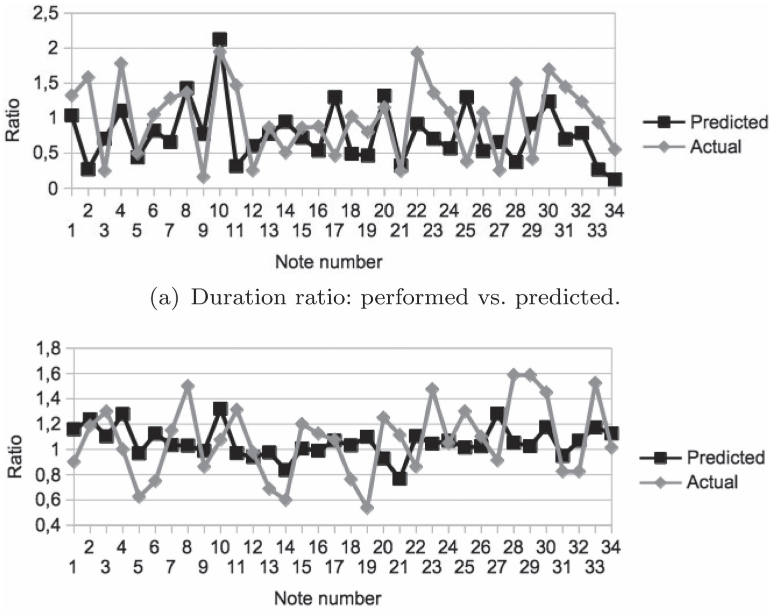
(a) Score to musician performance correspondence.

(b) Score to predicted performance correspondence.

Figure 9. Musician versus predicted performance. Top and bottom sequences represent score and performance piano roll, respectively. Vertical lines indicate score to performance note correspondence. Gray and white boxes represent notes predicted as *not ornamented* and *ornamented*, respectively.

### 6.5. *Duration and energy ratio curves*

Duration and energy deviation ratio measured in the musician performance and predicted by the system for one example piece (*All of me*) are compared in Figures 10(a) and 10(b), respectively. We obtained similar results for the other pieces in the data set. Similarity between the contour of the curves indicates that the deviations predicted by the system are coherent with the ones performed by the musician.



(a) Duration ratio: performed vs. predicted.

(b) Energy ratio: performed vs. predicted.

Figure 10. Performed versus predicted duration and energy ratio example for *All of me*. Gray and black lines represent performed and predicted ratios, respectively, for each note in the score.

### 6.6. *Musical samples*

Musical examples of the automatically generated ornamented pieces can be found in the Online Supplement (see the unnumbered section directly before the references list at the end of this article). The rendered audio of the *Yesterdays* music piece generated by the system (as test piece) has been included in this site.

## 7. Conclusions

In this article we have presented a machine learning approach for expressive performance (ornament, duration, onset, and energy) prediction and synthesis in jazz guitar music. We used a data set of 27 recordings performed by a professional jazz guitarist, and extracted a set of descriptors from the music scores and a symbolic representation from the audio recordings. In order to map performed notes to parent score notes we have automatically aligned performance to score data. Based on this alignment we obtained performance actions, calculated as deviations of the performance from the score. We created an ornaments database including the information on the ornamented notes performed by the musician. We have compared four learning algorithms to create models for ornamentation, based on performance measures, using a significance paired T-test. Feature selection techniques were employed to select the best feature subset for ornament modeling. For synthesis purposes, instance based learning was used to retrieve the most suitable ornament from the ornamentation database. A concatenative synthesis approach was used to generate expressive performances of new pieces – i.e. pieces not in the training set – automatically. A subjective perceptual evaluation based on listening tests is beyond the scope of this article. As future work, we plan to evaluate the performances generated by the system by computing the alignment distance between the system and the target performance.

## Acknowledgments

We would like to thank the two anonymous reviewers and Editor-in-Chief Professor Thomas Fiore for their constructive comments on previous versions of this article.

## Disclosure statement

No potential conflict of interest was declared by the authors.

## Funding

This project has received funding from: the European Union Horizon 2020 research and innovation programme [grant agreement No 688269]; the Spanish TIN project TIMUL [grant agreement TIN2013-48152-C2-2-R].

## Supplemental online material

Supplemental online material for this article can be accessed at doi:10.1080/17459737.2016.1207814 and https://soundcloud.com/machine-learning-and-jazz. In the Online Supplement we present an example of the generated pieces. The Online Supplement consists of a pdf description file and three audio files. The three audio files are an inexpressive (mechanical) rendering of the score, a recorded performance of the musician, and the rendered performance predicted by the system.

## References

Arcos, Josep Lluís, Ramon López De Mántaras, and Xavier Serra. 1998. “SaxEx: A Case-Based Reasoning System for Generating Expressive Musical Performances.” *Journal of New Music Research* 27 (3): 194–210.

Bantula, Helena, Sergio Giraldo, and Rafael Ramírez. 2014. “A Rule-Based System to Transcribe Guitar Melodies.” In *Proceedings of the 11th International Conference on Machine Learning and Music (MML 2014)*, 28 November 2014, Barcelona, Spain, 6–7.

Bresin, Roberto. 1998. “Artificial Neural Networks Based Models for Automatic Performance of Musical Scores.” *Journal of New Music Research* 27 (3): 239–270.

Bresin, Roberto, and Anders Friberg. 2000. “Emotional Coloring of Computer-Controlled Music Performances.” *Computer Music Journal* 24 (4): 44–63.

Cambouropoulos, Emilios. 1997. “Musical Rhythm: A Formal Model for Determining Local Boundaries, Accents and Metre in a Melodic Surface.” In *From Rhythm to Expectation*. Volume III of *Music, Gestalt, and Computing: Studies in Cognitive and Systematic Musicology*. Volume 1317 of Lecture Notes in Computer Science Series, 277–293. Berlin: Springer. doi:10.1007/BFb0034121.

Camurri, Antonio, Roberto Dillon, and Alberto Saron. 2000. “An Experiment on Analysis and Synthesis of Musical Expressivity.” *Proceedings of the 13th Colloquium on Musical Informatics (XIII CIM)*, 2–5 September 2000, L’Aquila: Italy. ftp://ftp.infomus.org/pub/Publications/2000/CIM2000CDS.PDF.

Cooper, Grosvenor, and Leonard B. Meyer. 1960. *The Rhythmic Structure of Music*. Chicago, IL: University of Chicago Press. http://www.press.uchicago.edu/ucp/books/book/chicago/R/bo24515499.html.

De Cheveigné, Alain, and Hideki Kawahara. 2002. “YIN, a Fundamental Frequency Estimator for Speech and Music.” *The Journal of the Acoustical Society of America* 111 (4): 1917–1930.

Friberg, Anders. 2006. “pDM: An Expressive Sequencer with Real-Time Control of the KTH Music-Performance Rules.” *Computer Music Journal* 30 (1): 37–48.

Friberg, Anders, Roberto Bresin, and Johan Sundberg. 2006. “Overview of the KTH Rule System for Musical Performance.” *Advances in Cognitive Psychology* 2 (23): 145–161.

Gabrielsson, Alf. 1999. “The Performance of Music.” In *The Psychology of Music,* edited by Diana Deutsch, Cognition and Perception Series, 2nd ed., 501–602. San Diego, CA: Academic Press.

Gabrielsson, Alf. 2003. “Music Performance Research at the Millennium.” *Psychology of Music* 31 (3): 221–272.

Giraldo, Sergio. 2012. “Modeling Embellishment, Duration, and Energy Expressive Transformations in Jazz Guitar.” Master’s thesis, Universidad Pompeu Fabram, Barcelona, Spain.

Giraldo, Sergio, and Rafael Ramírez. 2014. “Optimizing Melodic Extraction Algorithm for Jazz Guitar Recordings Using Genetic Algorithms.” In *Joint 42nd International Computer Music Conference and 13rd Sound & Music Computing Conference (ICMC-SMC 2014)*, 22–26 October 2014, Athens, Greece, 25–27.

Giraldo, Sergio, and Rafael Ramírez. 2015a. “Computational Generation and Synthesis of Jazz Guitar Ornaments using Machine Learning Modeling.” In *Proceedings of the 11th International Conference on Machine Learning and Music (MML 2014)*, August 2015, Vancouver, Canada, 10–12.

Giraldo, Sergio, and Rafael Ramírez. 2015b. “Computational Modeling and Synthesis of Timing, Dynamics and Ornamentation in Jazz Guitar Music.” In *11th International Symposium on Computer Music Interdisciplinary Research (CMMR 2015)*, Plymouth, UK, 806–814.

Giraldo, Sergio, and Rafael RafaelRamírez. 2015c. “Computational Modelling of Ornamentation in Jazz Guitar Music.” In *International Symposium in Performance Science*, 2 September 2015, Kyoto, Japan, 150–151.

Giraldo, Sergio, and Rafael Ramírez. 2015d. “Performance to Score Sequence Matching for Automatic Ornament Detection in Jazz Music.” In *International Conference of New Music Concepts (ICMNC 2015)*, Treviso, Italy, 8.

Goebl, Werner, Simon Dixon, Giovanni DePoli, Anders Friberg, Roberto Bresin, and Gerhard Widmer. 2008. “Sense in Expressive Music Performance: Data Acquisition, Computational Studies, and Models.” Chapter 5 in *Sound to Sense – Sense to Sound: A State of the Art in Sound and Music Computing*, 195–242. Berlin: Logos. http://iwk.mdw.ac.at/goebl/papers/Goebl-etal-2008-Sense-in-Expressive-Performance.pdf.

Gómez, Francisco, Aggelos Pikrakis, Joaquín Mora, Juan Manuel Díaz-Bánez, Emilia Gómez, and Francisco Escobar. 2011. “Automatic Detection of Ornamentation in Flamenco.” In *Fourth International Workshop on Machine Learning and Music (MML 2011)*, 20–22.

Grachten, Maarten. 2006. “Expressivity-Aware Tempo Transformations of Music Performances Using Case Based Reasoning.” PhD thesis, Universidad Pompeu Fabra: Barcelona, Spain.

Grindlay, Graham Charles. 2005. “Modeling Expressive Musical Performance with Hidden Markov Models.” Master’s thesis: University of California Santa Cruz. http://www.ee.columbia.edu/∼grindlay/pubs/UCSC_MS_thesis_2005.pdf.

Hall, Mark, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. “The WEKA Data Mining Software: An Update.” *ACM SIGKDD Explorations Newsletter* 11 (1): 10–18. http://www.cms.waikato.ac.nz/∼ml/publications/2009/weka_update.pdf.

Johnson, Margaret L. 1991. “Toward an Expert System for Expressive Musical Performance.” *Computer* 24 (7): 30–34.

Kennedy, Gary, and Barry Kernfeld. 2002. “Aebersold, Jamey.” In *The New Grove Dictionary of Jazz*. Volume 1. 2nd ed. 16–17. New York: Grove’s Dictionaries.

Kirke, Alexis, and Eduardo R. Miranda. 2013. “An Overview of Computer Systems for Expressive Music Performance.” In *Guide to Computing for Expressive Music Performance*, 1–47. London: Springer.

Miranda, Eduardo R., Alexis Kirke, and Qijun Zhang. 2010. “Artificial Evolution of Expressive Performance of Music: An Imitative Multi-Agent Systems Approach.” *Computer Music Journal* 34 (1): 80–96.

Narmour, Eugene. 1992. *The Analysis and Cognition of Melodic Complexity: The Implication–Realization Model*. Chicago, IL: University of Chicago Press.

Palmer, Caroline. 1997. “Music Performance.” *Annual Review of Psychology* 48 (1): 115–138.

Perez, Alfonso, Esteban Maestre, Stefan Kersten, and Rafael Ramírez. 2008. “Expressive Irish Fiddle Performance Model Informed with Bowing.” In *Proceedings of the International Computer Music Conference (ICMC 2008)*, Sonic Arts Research Centre: Belfast, Northern Ireland.

Puiggròs, Montserrat, Emilia Gómez, Rafael Ramírez, Xavier Serra, and Roberto Bresin. 2006. “Automatic Characterization of Ornamentation from Bassoon Recordings for Expressive Synthesis.” In *Proceedings of 9th International Conference on Music Perception and Cognition*, 22–26 August 2006, University of Bologna, Italy.

Ramírez, Rafael, and Amaury Hazan. 2006. “A Tool for Generating and Explaining Expressive Music Performances of Monophonic Jazz Melodies.” *International Journal on Artificial Intelligence Tools* 15 (4): 673–691.

The Real Book Series, Milwaukee, WI: Hal Leonard. 2013. http://www.halleonard.com/aboutUs.jsp.

Serra, Xavier. 1997. “Musical Sound Modeling with Sinusoids Plus Noise.” *Musical Signal Processing* 91–122.

Todd, Neil. 1989. “A Computational Model of Rubato.” *Contemporary Music Review* 3 (1): 69–88.

Todd, Neil P. McAngus. 1992. “The Dynamics of Dynamics: A Model of Musical Expression.” *The Journal of the Acoustical Society of America* 91 (6): 3540–3550.

Todd, Neil P. McAngus. 1995. “The Kinematics of Musical Expression.” *The Journal of the Acoustical Society of America* 97 (3): 1940–1949.

Widmer, Gerhard. 2003. “Discovering Simple Rules in Complex Data: A Meta-Learning Algorithm and Some Surprising Musical Discoveries.” *Artificial Intelligence* 146 (2): 129–148.

Widmer, Gerhard, and Asmir Tobudic. 2003. “Playing Mozart by Analogy: Learning Multi-Level Timing and Dynamics Strategies.” *Journal of New Music Research* 32 (3): 259–268.

Woodrow, Herbert. 1951. “Time Perception.” In *Handbook of Experimental Psychology*, edited by Stanley Smith Stevens, 1224–1236. New York: Wiley.

Zapata, José R., André Holzapfel, Matthew E. P. Davies, João Lobato Oliveira, and Fabien Gouyon. 2012. “Assigning a Confidence Threshold on Automatic Beat Annotation in Large Datasets.” In *13th International Society for Music Information Retrieval Conference*, 8–12 October 2012, Porto, Portugal, 157–162.