Automatic Performer Identification in Celtic Violin Audio Recordings

Rafael Ramirez, Esteban Maestre, Alfonso Perez, Xavier Serra
Information Systems and Telecommunications Department
Pompeu Fabra University
Tanger 122, 08018 Barcelona, Spain
Tel:+34 935421365, Fax:+34 935422202
rafael.ramirez@upf.edu, esteban.maestre@upf.edu,
alfonso.perez@upf.edu, xavier.serra@upf.edu

Abstract

We present a machine learning approach to the problem of identifying performers from their interpretative styles. In particular, we investigate how violinists express their view of the musical content in audio recordings and feed this information to a number of machine learning techniques in order to induce classifiers capable of identifying the interpreters. We apply sound analysis techniques based on spectral models for extracting expressive features such as pitch, timing, and amplitude representing both note characteristics and the musical context in which they appear. Our results indicate that the features extracted contain sufficient information to distinguish the considered performers, and the explored machine learning methods are capable of learning the expressive patterns that characterize each of the interpreters.

1. Introduction

Music performance plays an important role in our culture nowadays. Most people are able to distinguish between different types of expression in performances. However, there is little quantitative information about how and in which contexts expressive performance occurs. Expressive music performance research (for an overview see (Gabrielsson, 1999, 2003)) investigates the manipulation of sound properties such as pitch, timing and amplitude in an

attempt to understand and recreate expression in performances. In the past, expressive performance research has investigated manipulations of different sound properties in score-driven performances in classical music (e.g. (Widmer, 2002)) as well as different types of deviations from the score in popular music (e.g. (Lopez de Mantaras, 2002), (Ramirez, 2008)).

On the other hand, a key challenge in the area of music information is the development of efficient and reliable music search and retrieval systems. One of the main deficiencies of current music search and retrieval systems is the gap between the simplicity of the content descriptors that can be currently extracted automatically and the semantic richness in music information. It has been widely recognized that music retrieval techniques should incorporate high-level music information. The incorporation of expressive-content based performer identification in search and retrieval systems raises particularly interesting questions but has nevertheless received relatively little attention in the past.

In this paper we focus on automatic identification of violinists based on expressive content extracted from monophonic audio performances (we consider monophonic recordings in order to simplify the audio processing). In particular, we investigate how violinists add expression to performances of Celtic music pieces and how to use this information in order to automatically identify them. We extract features from the audio recordings representing both note characteristics (e.g. note duration), and aspects of the musical context in which the note appears (e.g. pitch interval with previous note). We study deviations of parameters such as pitch, timing, and amplitude. This work builds upon our previous work on interpreter identification (Ramirez, 2008b) and (Ramirez 2010) in which we present preliminary work on violin interpreter identification and an algorithm for automatically identifying saxophonists in Jazz performances, respectively. In (Ramirez, 2010), both the audio feature extraction and the classification algorithm are tuned specifically for Jazz saxophone, while here we focus on violin performances.

The rest of the paper is organized as follows: Section 2 describes related work on music performance. Section 3 describes how we process the audio recordings in order to extract information about both the internal structure of notes (i.e. intra-note information) and the musical context in which they appear (i.e. inter-note information). Section 4 describes our approach to performance-driven performer identification and applies it to the identification of violinists, and finally, Section 5 presents some conclusions and indicates some areas of future research.

2. Expressive music performance background

There has been much speculation as to *why* performances contain expression. Hypothesis include that musical expression communicates emotions (Juslin, 2001) and that it clarifies musical structure (Kendall, 1990), i.e. the performer shapes the music according to his/her own intentions (Apel, 1972). In any case, understanding and formalizing expressive music performance is an extremely challenging problem, which in the past has been studied from different perspectives. The main approaches to studying expressive performance have been based on statistical analysis (e.g. (Repp, 1992)), mathematical modeling (e.g. (Todd, 1992)), analysis-by-synthesis (e.g. (Friberg, 2006)), and machine learning techniques (e.g. Widmer, 2004).

One of the first attempts to provide a computer system with musical expressiveness is that of Johnson (Johnson, 1992). Johnson 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. A long-term effort in expressive performance modeling is the work of the KTH group (Sundberg, 1983), (Bresin, 2001), (Friberg, 2006). Their *Director Musices* system incorporates rules for tempo, dynamic and articulation transformations. The rules are obtained from both theoretical musical knowledge, and experimentally by using an analysis-by-synthesis manual approach. The rules are divided into *differentiation rules* which enhance the differences between notes, *grouping rules,* which specify what notes belong

together, and *ensemble rules* which synchronize the voices in an ensemble. Canazza et al. (Canazza, 1997, 2004) implemented a system to analyze the relationship between the musician's expressive intentions and his/her performance. The analysis reveals two expressive dimensions, one related to energy (*dynamics*), and another one related to velocity (*tempo*). Dannenberg et al. (Dannenberg, 1998) investigated the trumpet articulation transformations using manually generated rules. They developed a trumpet synthesizer, which combines a physical model with an expressive performance model. The performance model generates control information for the physical model using a set of rules manually extracted from the analysis of a collection of performance recordings.

Previous research addressing expressive music performance using machine learning techniques has included a number of approaches. Lopez de Mantaras et al. (Lopez de Mantaras, 2002) report on SaxEx, a performance system capable of generating expressive solo saxophone 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.

Ramirez et al. (Ramirez, 2006) have explored and compared diverse machine learning methods for obtaining expressive music performance models for Jazz saxophone that are capable of both generating expressive performances and explaining the expressive transformations they produce. 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 (e.g. note duration, loudness) and at an intra-note level (e.g. note attack, sustain). Based on the theory generated by the set of rules,

they implemented a melody synthesis component, which generates expressive monophonic output (MIDI or audio) from inexpressive melody MIDI descriptions.

With a few exceptions, most of the research in expressive performance using machine learning techniques has focused on solo classical piano music, e.g. (Dovey, 1995), (Van Baelen, 1996), (Widmer, 2001), (Tobudic, 2003). The modeling of expressive performance in these works has been centered on *global* timing and loudness transformations.

The use of expressive performance models (either automatically induced or manually generated) for identifying musicians has received little attention in the past. Saunders et al. (Saunders, 2004) have applied string kernels to the problem of recognizing famous pianists from their playing style. The characteristics of performers playing the same piece are obtained from changes in beat-level tempo and beat-level loudness. From such characteristics, general performance alphabets can be derived, and pianists' performances can then be represented as strings. They applied both kernel partial least squares and Support Vector Machines to these data.

Stamatatos and Widmer (Stamatatos, 2005) have addressed the problem of identifying the most likely music performer, given a set of performances of the same piece by a number of skilled candidate pianists. They proposed a set of very simple features for representing stylistic characteristics of a music performer that relate to a kind of 'average' performance. A database of piano performances of 22 pianists playing two pieces by Frédéric Chopin is used. They proposed an ensemble of simple classifiers derived by both subsampling the training set, and subsampling the input features. Experiments showed that the proposed features are able to quantify the differences between music performers.

Ramirez et al. (Ramirez, 2007, 2010) have developed a machine learning approach to identifying Jazz saxophonists by analyzing the pitch, timing, amplitude and timbre of individual notes, as well as the timing and amplitude of individual intra-note events. Their approach consists of establishing a

performer-dependent mapping from inter-note features (essentially a ‘score’ whether or not the score physically exists) to a repertoire of inflections characterized by intra-note features. Thus, their work strongly relies on the performances’ timbre content, which makes sense in performances in Jazz saxophone. In this paper we extend this work by considering violin performances in which the articulation, timing and amplitude content in the performances are central for interpreter identification.

Molina et al. (Molina, 2010) proposed an approach for identifying violinists in monophonic audio recordings. They considered a database of sonatas and partitas for solo violin by J.S. Bach, and identified performers by capturing their general expressive footprint based on a characterization of the way melodic patterns are played as a set of frequency distributions. Performances were transcribed focusing on the melodic contour, and melodic segments were tagged according to Narmour’s Implication/Realization model (Narmour, 1990).

3 Audio Analysis

In this section, we outline how we extract a symbolic description of a performed melody for monophonic recordings (for a comparison of the method reported here and other methods see (Gomez, 2003)). We use this melodic representation to provide description of the performances and apply machine learning techniques to this representation. Our interest is to obtain, for each performed note, a set of symbolic features from the audio recording.

The process is depicted in Figure 1 and described in the subsequent sections. In a first step, low-level instantaneous descriptors are extracted from the sound recordings. The main low-level descriptors used to characterize note-level expressive performance are instantaneous energy and fundamental frequency. From these low-level descriptors we perform note segmentation. Once the note boundaries are known, the note descriptors are computed from the low-level values.

3.1 Low-level descriptors computaton: energy and fundamental frequency

First of all, we perform frame-by-frame spectral analysis of sound recordings. Then, for each frame we compute a set of low-level descriptors: frame energy and an estimation of the fundamental frequency.

The instantaneous energy descriptor (for each frame) is computed on the spectral domain, using the values of the amplitude spectrum at each analysis frame. In addition, energy is computed in different frequency bands as defined in (Klapuri, 1999), and these values are used by the algorithm for note segmentation.

For the estimation of the instantaneous fundamental frequency we use a harmonic matching model derived from the Two-Way Mismatch procedure (TWM) (Maher, 1994). For each fundamental frequency candidate, mismatches between the harmonics generated and the measured partials frequencies are averaged over a fixed subset of the available partials. A weighting scheme is used to make the procedure robust to the presence of noise or absence of certain partials in the spectral data. The solution presented in (Maher, 1994) employs two mismatch error calculations. The first one is based on the frequency difference between each partial in the measured sequence and its nearest neighbor in the predicted sequence. The second is based on the mismatch between each harmonic in the predicted sequence and its nearest partial neighbor in the measured sequence. This two-way mismatch helps to avoid octave errors by applying a penalty for partials that are present in the measured data but are not predicted, and also for partials whose presence is predicted but which do not actually appear in the measured sequence. The TWM mismatch procedure has also the benefit that the effect of any spurious components or partial missing from the measurement can be counteracted by the presence of uncorrupted partials in the same frame. After a first test of this implementation, some improvements to the original algorithm where implemented to deal with some errors of the algorithm (see (Gomez, 2003) for details).

Note segmentation is performed using a set of frame descriptors, which are the previously described (i) energy in different frequency bands and (ii) fundamental

frequency. Energy onsets are first detected following a band-wise algorithm that uses some psycho-acoustical knowledge (Klapuri, 1999). In a second step, fundamental frequency transitions are also detected.

3.2 Note descriptors.

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 (McNab, 1996). This is done to avoid taking into account mistaken frames in the fundamental frequency mean computation. First, frequency values are converted into cents, by the following formula:

$$c = 1200 \cdot \log_2\left(\frac{f}{f_{ref}}\right) \qquad (1)$$

where f_{ref} = 8.176 Hz (f_{ref} is the reference frequency of the C_0). Then, we define histograms with bins of 100 *cents* and hop size of 5 *cents* and we compute the maximum of the histogram to identify the note pitch. Finally, we compute the frequency mean for all the points that belong to the histogram. The MIDI pitch is computed by quantization of this fundamental frequency mean over the frames within the note limits. Figure 1 shows an overview of the melodic description process.

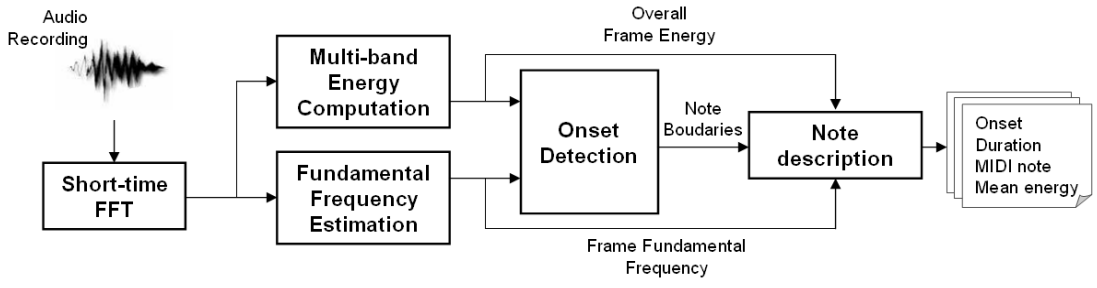

Figure 1. Schematic view of the melodic description process. Note onsets are extracted based on the study of energy and fundamental frequency.

3.3 Note Transitions

For characterizing note detachment, we also extract some features of the note-to-note transitions describing how two notes are detached. For two consecutive notes, we consider the transition segment starting at the first note's release and finishing at the attack of the following one (for details see (Maestre, 2005)). Both the energy envelope and the fundamental frequency contour (schematically represented by E and f_0 in Figure 2) during transitions are studied in order to extract descriptors related to articulation. We measure the energy envelope minimum position t_c (see also Figure 2).

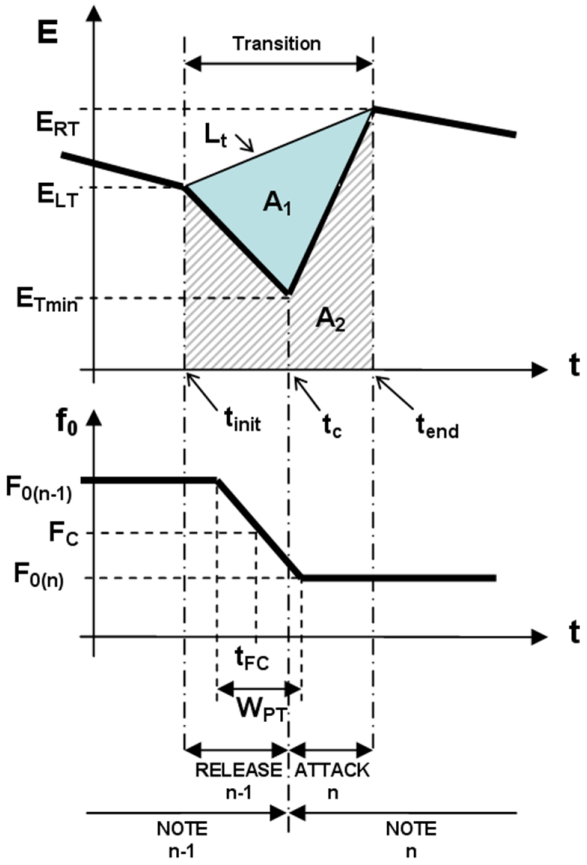

Figure 2: Schematic view of the transition segment characterization

We compute a legato descriptor as described next. First, we join start and end

points on the energy envelope contour by means of a line L_t representing the smoothest case of detachment. Then, we compute both the area A_2 below energy envelope and the area A_1 between energy envelope and the joining line L_t and define our legato descriptor as shown in Eq. (2). The relevance of this descriptor was assessed in (Maestre, 2005).

$$LEG = \frac{A_1}{A_1 + A_2} = \frac{\int_{t_{init}}^{t_{end}} (L_t(t) - E(t))dt}{\int_{t_{init}}^{t_{end}} L_t(t)dt} \qquad (2)$$

3.4 Musical Analysis.

After having computed the note descriptors as above, and as a first step towards providing an abstract structure for the recordings under study, we decided to use Narmour's theory of perception and cognition of melodies (Narmour 1990), (Narmour, 1991) to analyze the performances.

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. Any two consecutively perceived notes constitute a melodic interval. According to Narmour, 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. Two main principles recognized by Narmour concern *registral direction* and *intervallic difference*. The principle of registral direction states that small intervals imply an interval in the same registral direction (a small upward interval implies another upward interval and analogously for downward intervals), and large intervals imply a change in registral direction (a large upward interval implies a downward interval and analogously for downward intervals). The principle of intervallic difference states that a small (five semitones or less) interval implies a similarly-sized interval (plus or minus 2 semitones), and a large interval (seven semitones or more)

implies a smaller interval. Based on these two principles, melodic patterns or groups can be identified that either satisfy or violate the implication as predicted by the principles. Such patterns are called structures and are labeled to denote characteristics in terms of registral direction and intervallic difference. The prototypical Narmour structures are P, D, ID, IP, VP, R, IR and VR (see Figure 3). A note in a melody often belongs to more than one structure. Thus, a description of a melody as a sequence of Narmour structures consists of a list of overlapping structures. We have implemented a Narmour structure parser and we have parsed each melody in the training data in order to automatically generate an implication/realization analysis of the pieces. Figure 4 shows the analysis for a melody fragment.

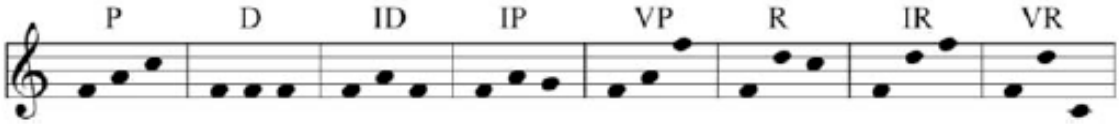

Fig. 3 Prototypical Narmour Structures

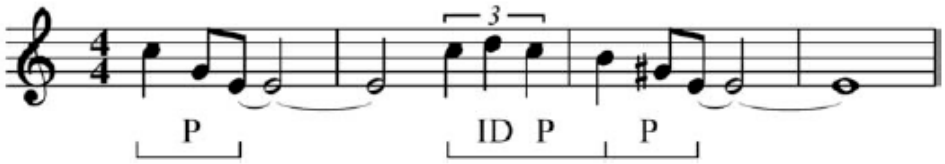

Fig. 4 Narmour analysis of a melody fragment

4. Performance-driven Interpreter Identification

In this section, we describe our approach to the task of identifying violinists from their playing style. Our approach consists of learning an expressive model for each performer, and given a new performance, identifying the performer whose model most closely matches the new performance.

4.1 Note characterization

The note features represent both properties of the note itself and aspects of the musical context in which the note appears. Information about the note includes note pitch and note duration, while information about its melodic context includes the relative pitch and duration of the neighboring notes (i.e. pitch interval and duration ratio with previous and following notes) as well as the Narmour structures to which the note belongs (Nar_1, Nar_2 and Nar_3 denote the three Narmour structures with the considered note in position 1, 2 and 3, respectively). The Narmour structures for each note are computed by performing the musical analysis described in Section 3.4. Thus, each performed note N_i is contextually characterized by the tuple

N_i = (*Pitch, Dur, PrevPitch, PrevDur, NextPitch, NextDur, Nar1, Nar2, Nar3*)

4.2 Classification algorithm

We are ultimately interested in obtaining a classifier *F* of the following form:

$$F(MelodyFragment(N_1,\ldots,N_k)) \rightarrow Performers$$

where $MelodyFragment(N_1,\ldots,N_k)$ is the set of melody fragments (composed of notes $N_1,\ldots,N_k$) and *Performers* is the set of possible performers to be identified. For each performer P_i to be identified we learn an expressive performance model M_i predicting the performer timing, energy and some expressive characteristics of note transitions (characterized by the note duration, energy mean and legato descriptors described in Section 3):

$$M_i(Note) \rightarrow PN$$

where *Note* is a note in the score represented by its context features, i.e. *Note* is represented by the tuple *(Pitch, Dur, PrevPitch, PrevDur, NextPitch, NextDur,* Nar_1, Nar_2, Nar_3*)* as described before, and *PN* is the vector *(PerfDur,PerfEner,PerfLeftTrans,PerfRightTrans)* containing the model's prediction for how Performer P_i would play the note in terms of note duration (*PerfDur*), energy (*PerfEner*), and transitions (*PerfLeftTrans,PerfRightTrans*).

For training each Model M_i we explore several machine learning techniques. The machine learning techniques considered are the following:

- *Decision Trees*. A decision tree classifier (Quinlan, 1993) recursively constructs a tree by selecting at each node the most relevant attribute. This process gradually splits up the training set into subsets until all instances at a node have the same classification. The selection of the most relevant attribute at each node is based on the *information gain* associated with each node of the tree (and corresponding set of instances). We have applied an extension of decision trees called model trees, which are able to learn regression models, as opposed to classification models.

- *Support Vector Machines (SVM)*. SVM (Cristiani, 2000) take great advantage of using a non-linear attribute mapping that allows them to be able to predict non-linear models (though they remain linear in a higher dimension space). Thus, they provide a flexible prediction, but with a higher computational cost necessary to perform all the computations in the higher dimensional space. SVM have been extended to handle regression problems. The accuracy of SVM largely depends on the choice of the kernel evaluation function and the parameters which control the amount to which deviations are tolerated (denoted by epsilon). In this paper we have explored SVM with linear and polynomial kernels (2nd, 3rd and 4th order). The results shown in Section 5.2 are those obtained with 2nd order polynomial kernel.

- *Artificial Neural Networks (ANN).*ANN learning methods provide a robust approach to approximating a target function. In this paper we apply a gradient descent back propagation algorithm (Chauvin, 1995) to tune the neural network parameters to best fit the training set. The back propagation algorithm learns the weights for a multi layer network, given a network with a fixed set of units and interconnections. We set the momentum applied to the weights during updating to 0.2 and the learning rate (the amount the weights are updated) to 0.3. We use a fully-

connected multi layer neural network with one input neuron for each attribute and one hidden layer with six neurons.

- *Lazy Methods.* Lazy Methods are based on the notion of lazy learning which 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. In this paper we have explored the *k*-Nearest Neighbor (*k*-NN) algorithm (with *k* in {1,2,3,4,7}), which is capable of handling noisy data well if the training set has an acceptable size. However, *k*-NN does not behave well in the presence of irrelevant attributes. The results shown in Section 5.2 are those obtained with k=2.

- *Ensemble Methods.* One obvious approach to making more reliable decisions is to combine the output of several different models. In this paper we explore the use of methods for combining models (called *ensemble* methods) generated by machine learning. In particular, we have explored *voting, stacking, bagging* and *boosting*. In many cases they have proved to increase predictive performance over a single model. In the *voting* method, a set of *n* different classifiers are trained on the same training data using different learning algorithms (in this paper we applied decision trees, SVM, ANN, and 1-NN), and prediction is performed by allowing all *n* classifiers to 'vote' on class prediction; the final prediction is the class that gets the most votes. *Stacking* trains *n* learning algorithms (here we applied decision trees, SVM, ANN, and 1-NN) in the same training data and also trains another learning algorithm, the 'meta-learner', (we applied decision trees) to learn to predict the class from the predictions of the base learners. *Bagging* draws *n* bootstrap samples from the training data, trains a given learning algorithm (here we consider decision trees) on each of these *n* samples (producing *n* classifiers) and predicts by simple voting of all *n* classifiers. *Boosting* generates a series of classifiers using the same learning algorithm (here we applied decision trees) but differently weighted examples from the

same training set, and predicts by weighted majority vote (weighted by accuracy) of all *n* classifiers.

All the recorded pieces are segmented into fragments representing musical phrases. Given a fragment denoted by a list of notes [$N_1,\ldots,N_m$] and a set of possible performers denoted by a list of performers [$P_1,\ldots,P_n$], classifier *F* identifies the performer as follows:

$F([N_1,\ldots,N_m], [P_1,\ldots,P_n])$
    for each performer P_i
        $Score_i = 0$
    for each note N_k
        FN_k = features(N_k)
        $(PD_k,PE_k,PLT_k,PRT_k) = Mi(FN_k)$
        for each performer P_i
            $ScoreNK_i = dist([D(N_K),E(N_k),LT(N_k),RT(N_k)] ,$
                $[PDk,PEk,PLTk,PRTk])$
        $Scorei = Scorei + ScoreNKi$
return P_i *(i in {1,...,m})* with minimum score

where

$dist([X1,X2,X3,X4], [Y1,Y2,Y3,Y4]) = sqrt((X1-Y1)^2 + (X2-Y2)^2 + (X3-Y3)^2/2 + (X4-Y4)^2/2)$

For each note in the melody fragment the classifier *F* computes the set of the note contextual features. Once this is done, for each note N_k and for each performer P_i, performance model M_i predicts the expected duration, energy, and transitions for N_k. This prediction is based on the note's contextual features. The score $Score_i$ for each performer *i* is updated by taking into account the Euclidean distance between the note's actual duration, energy and transitions, and the predicted values. Finally, the performer with the lower score (i.e. the smaller accumulated distance) is returned. Clearly, the expressive models *Mi* play a central role in the output of classifier *F*. Note that in computing this

distance we have balanced the weight of the duration, energy and transition differences by dividing by 2 the left and right transition differences. The reason for this is that we consider the left and right transitions as one prediction.

4.3 Evaluation

We evaluated the induced classifiers by performing the standard 10-fold cross validation in which 10% of the melody fragments is held out in turn as test data while the remaining 90% is used as training data. When performing the 10-fold cross validation, we leave out the same number of melody fragments per class. In order to avoid optimistic estimates of the classifier performance, we explicitly remove from the training set all melody fragment repetitions of the hold out fragments. This is motivated by the fact that musicians are likely to perform a melody fragment and its repetition in a similar way. Thus, the applied 10-fold cross validation procedure, in addition to holding out a test example from the training set, also removes repetitions of the example.

5. Violin performer identification

5.1 Training data

In this work we focus on Celtic jigs. Celtic Jigs are a form of lively folk dance, as well as the accompanying dance tune, originating in England in the sixteenth century and today most associated with Irish dance music. Celtic jigs are fast tunes but slower than reels which usually consist of eighth notes in a ternary time signature (6/8 time), with strong accents at each beat. The training data used in this research are 27 monophonic recordings. It consists of nine Celtic jigs, each performed by three professional violinists. The recordings were made explicitly for the current study. Apart from the tempo (they played following a metronome), the musicians were not given any particular instructions on how to perform the pieces.

5.2 Results

Initially, we evaluated the expressive performance model, M_1, M_2 and M_3, for each performer considered. For M_1 we obtained correlation coefficients of 0.88, 0.83, 0.65 and 0.71 for duration, energy, left transition, and right transition prediction tasks, respectively, while we obtained 0.91, 0.85, 0.70 and 0.73 for M_2, and 0.82, 0.74, 0.67 and 0.72 for M_3. These numbers were obtained by performing 10-fold cross validation on the training data. The induced models seem to capture accurately the violinists' expressive transformations. Figure 5 contrasts the note duration deviations predicted by model M_1 and the deviations performed by the violinist. Similar results were obtained for M_2 and M_3.

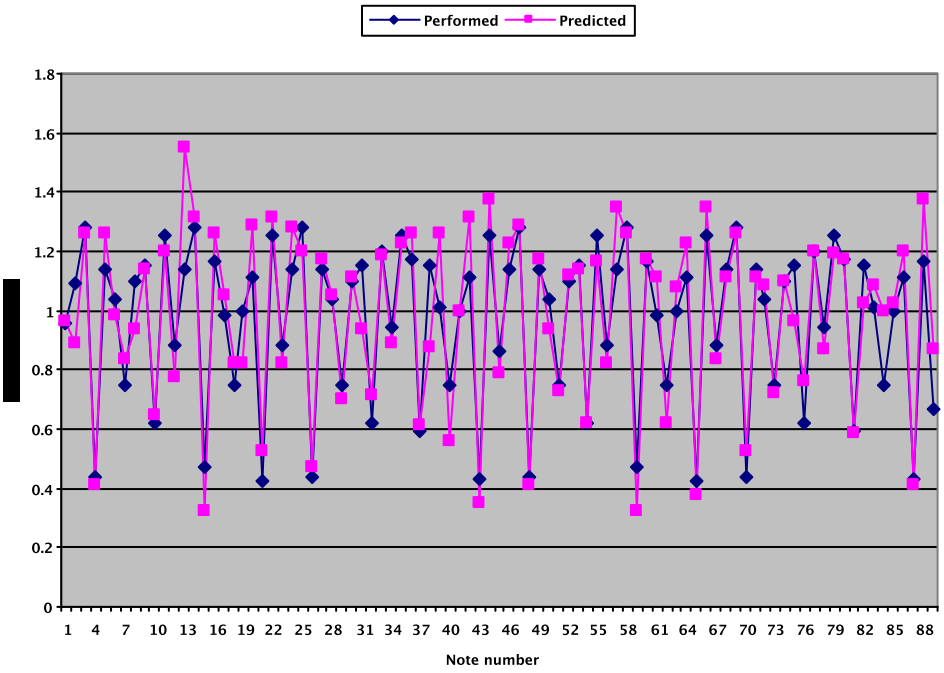

Figure 5. Note duration deviation ratio for a tune with 89 notes. Comparison between performed by P_1 and predicted by M_1

We then proceed to evaluate the classification function *F*. There were a total of 806 notes available for each performer. We segmented each of the performed pieces in phases and obtain a total of 92 phrases (between 6 and 12 notes long) for each performer. The classification accuracy of the baseline classifier (one which chooses randomly one of the three performers) is 33% (measured in correctly classified instances percentage). The average accuracy and the accuracy obtained for the most successful trained classifier was 76.1% and

81.2%, respectively. The correctly classified instances percentage for each learning method is presented in Table 1. For each algorithm, the results shown in Table 1 are obtained using the parameter settings described in Section 4.2. We also evaluated the classification accuracy when considering pairs of interpreters, i.e. combinations of two models. For M_1 vs. M_2 we obtained average accuracy and best accuracy (as correctly classified instances percentage) of 82.4% and 88.9%, respectively. Similarly, for M_1 vs. M_3 we obtained 87.9% and 92.4%, and for M_2 vs. M_3 we obtained 87.6% and 92.2%, as average and best accuracies.

The obtained results seem to indicate that it is indeed feasible to train successful classifiers to identify performers from their playing style using the considered features. It must be noted that the performances in our training data were recorded at different times with different recordings settings (the only constraint was the performance tempo). This corresponds to a general setting where recordings are obtained under very different circumstances.

Algorithm	**1-note**	**Phrase**
Decision Trees	29.1	69.3
Support Vector Machines	35.7	80.7
Artificial Neural Networks	32.9	76.5
k-Nearest Neighbor	30.6	72.9
Bagging (decision trees)	31.2	78.1
Boosting (decision trees)	30.5	74.1
Voting (decision trees, SVM, ANN, 1-NN)	32.9	76.5
Stacking (decision trees, SVM, ANN, 1-NN)	31.7	81.2

Table 1: Classification accuracy (as correctly classified instances percentage) for 1-note and short-phrases segments

5.3 Discussion

The difference between the results obtained and the accuracy of a baseline classifier indicates that the features extracted contain sufficient information to identify the studied set of performers, and that the machine learning methods explored are capable of learning performance patterns that distinguish these performers. It is worth noting that every learning algorithm investigated (decision trees, SVM, ANN, k-NN and the reported ensemble methods) produced considerably better than random classification accuracies. This supports our statement about the feasibility of training successful classifiers for the case study reported.

In order to give an indication of how difficult the same classification task is for humans, we have conducted a small number of blind listening trials. The experiment involved 7 healthy participants (4 male, 3 female). Out of these 7 participants, 4 declared to have some musical training (more than 5 years music training). For each of the three violinists, the participants were presented with 5 short-phrase fragments (i.e. in total 15 fragments) in order to get an idea of the style of each of the performers. Then the participants were asked to classify a different set of 10 short fragments (composed of 3 fragments of the first performer, 3 of the second one and 4 of the third one). The obtained average, best and worst accuracy was 75.7%, 90% and 50%, respectively. The sample of participants and musical material is to clearly too small to draw any definite conclusions but interestingly, the average accuracies obtained by the algorithm (i.e. 76.1%) and in the human experiment (i.e. 75.7%) are comparable.

We have selected two types of musical segment lengths: 1-note segments and short-phrase segments. Evaluation using 1-note segments results in poor classification accuracies, while short-phrase segments evaluation results in accuracies well above the accuracy of a baseline classifier. The poor results of the 1-note evaluation may indicate that although the extracted features are relevant, it is not sufficient to consider them in a one-note basis. Just as a human expert would have problems identifying interpreters from listening to one-note audio files, the trained classifiers are not able to identify the performers reliably given this limited information. As soon as there are more

notes involved together with the context in which they appear, the trained classifier (just as a music expert) improves its capacity to identify the interpreter.

One issue, which is not clear from the reported results, is what features are mostly responsible for the identification results. In order to investigate this we have performed an additional experiment in which we have applied each model separately. For the duration model, the obtained average and best accuracy are 46.7% and 51.1%, respectively. For the energy model, the obtained average and best accuracy are 43.0% and 47.6%, respectively. Finally for the transition model, the obtained average and best accuracy and are 41.6% and 43.9%, respectively. These results seem to indicate that there is some performer-specific information in the isolated duration, energy and transition models but the models are certainly more accurate at identifying interpreters when considered together.

An alternative approach to identify the violinists considered would be to establish a performer-dependent mapping from the note's context (i.e. its inter-note description) to a repertoire of inflections characterized by timbre features, as described in (Ramirez, 2010). While this approach has been proved to produce good results for identifying saxophonists in Jazz performances, it does not necessarily mean that it can produce similar results for violin interpreter identification. In order to investigate this, we applied the timbre-based approach to our violin data and we obtained lower classification accuracies: the average accuracy and the accuracy obtained for the most successful trained classifier was 45.3% and 50.7%, respectively. The reason for these results is that for identifying a performer, the relative importance of different expressive resources varies depending on the instrument being considered. Two clear extremes are the singing voice and the piano. While timing and dynamics are always an important expressive resources and thus, provide important information for interpreter identification, timbre information is clearly much more important for singer identification that it is for pianist identification.

6. Conclusions

In this paper, we concentrated on the task of automatic identification of violin performers based on their playing style. We have applied sound analysis techniques to monophonic audio recordings in order to extract pitch, timing, amplitude and transition features, characterising both the notes and the musical context in which they appear. We explored and compared different machine learning techniques for building style-based performer classifiers. The results obtained indicate that the extracted features contain sufficient information to identify the studied set of performers, and that the machine learning methods explored are capable of learning performance patterns that distinguish these performers. We plan to extend the number of performers, as well as the set of descriptors with relevant descriptors such as *vibrato*. We also plan to extend our approach to performance-based performer identification in polyphonic multi-instrument audio recordings.

References

(Bresin, 2001) Bresin, R. 2001. Articulation rules for automatic music performance. In Schloss, A., Dannenberg, R., & Driessen, P. (Eds.), Proceedings of the International Computer Music Conference, pp. 294-297. San Francisco, ICMA.

(Canazza, 1997) Canazza, S.; De Poli, G.; Roda, A.;and Vidolin, A. 1997 Analysis and Synthesis of Expressive Intention in a Clarinet Performance. Proceedings of the 1997 International Computer Music Conference, pp. 113-120. San Francisco, ICMA.

(Canazza, 2004) Canazza, S.; De Poli, G.; C. Drioli, Roda, A. and Vidolin, A. 2004. Modeling and control of expressiveness in music performance, Proceedings IEEE Multimedia, vol.7, no.3, no. 4, pp. 79-83

(Chauvin, 1995) Chauvin, Y., Rumelhart D.E. 1995. Backpropagation: Theory, Architectures and Applications. Lawrence Erlbaum Assoc.

(Cristiani, 2000) Cristianini N., Shawe-Taylor J. 2000. An Introduction to Support Vector Machines, Cambridge University Press

(Dannenberg, 1998) Dannenberg, R. B., and Derenyi, I. 1998. Combining Instrument and Performance Models for High-Quality Music Synthesis. Journal of New Music Research 27(3): 211-238.

(Dovey, 1995) Dovey, M.J. 1995. Analysis of Rachmaninoff's Piano Performances Using Inductive Logic Programming. European Conference on Machine Learning, pp. 279-282, Springer-Verlag.

(Friberg, 2006) Friberg, A., Bresin, R., Sundberg, J. 2006. Overview of the KTH rule system for musical performance. Advances in Cognitive Psychology, Special Issue on Music Performance, 2(2-3), 145-161.

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

(Gabrielsson, 2003) Gabrielsson, A. 2003. Music Performance Research at the Millennium. Psychology of Music, Vol. 31, No. 3, 221-272

(Gomez, 2003) Gómez, E., Klapuri, A., Meudic, B. 2003. Melody Description and Extraction in the Context of Music Content Processing. Journal of New Music Research. 32.

(Johnson, 1992) Johnson, M.L. 1992. An expert system for the articulation of Bach fugue melodies. In Readings in Computer-Generated Music, ed. D.L. Baggi, pp. 41-51, IEEE Computer Society.

(Juslin, 2001) Juslin, P. N., & Laukka, P. 2001. Impact of intended emotion intensity on cue utilization and decoding accuracy in vocal expression of emotion. Emotion, 1, pp. 381-412.

(Kendal, 1990) Kendall R. A., Carterette E. C. 1990. The communication of musical expression. *Music Percept.* 8, pp.129–64

(Klapuri, 1999) Klapuri, A. 1999. Sound Onset Detection by Applying Psychoacoustic Knowledge, Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP. vol.6, pp. 3089 - 3092

(Lopez de Mantaras, 2002) Lopez de Mantaras, R. and Arcos, J.L. 2002. AI and music, from composition to expressive performance, AI Magazine, 23(3). pp.43-57.

(Maestre, 2005) Maestre, E., and Gomez, E. 2005. “Automatic characterization of dynamics and articulation of monophonic expressive recordings” *In Proceedings of the 118th AES Convention*, pp. 44-27.

(Maher, 1994) 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.

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

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

(Molina, 2010) Molina-Solana, M., Arcos, J.L., Gomez, E. 2010 "Identifying Violin Performers by their Expressive Trends". Intelligent Data Analysis, 14(5), pp. 555-571.

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

(Narmour, 1991) Narmour, E. 1991. The Analysis and Cognition of Melodic Complexity: The Implication Realization Model. University of Chicago Press.

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

(Ramirez, 2006) Ramirez, R. Hazan, A. 2006. A Tool for Generating and Explaining Expressive Music Performances of Monophonic Jazz Melodies, International Journal on Artificial Intelligence Tools, 15(4), pp. 673-691.

(Ramirez, 2007) Ramirez, R., Maestre, E., Pertusa, A., Gomez, E., Serra, X. 2007. "Performance-based interpreter identification in saxophone audio recordings". IEEE Trans. on Circuits and Systems for Video Technology, 17(3):356-364.

(Ramirez, 2008) Ramirez, R., Hazan, A., Maestre, E., Serra, X. 2008. A Genetic Rule-based Expressive Performance Model for Jazz Saxophone, Computer Music Journal, 32(1), pp.38-50.

(Ramirez, 2008b) Ramirez, R., Perez, A., Kersten, S., Maestre, E. 2008. Performer Identification in Celtic Violin Recordings, International Society of Music Information Retrieval Conference, pp. 483-488.

(Ramirez, 2010) Ramirez, R., Maestre, E., Serra, X. 2010. Automatic performer identification in commercial monophonic Jazz performances, Pattern Recognition Letters, 31, pp. 1514-1523.

(Repp, 1992) 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 92(5), pp. 2546-68

(Saunders, 2004) Saunders C., Hardoon D., Shawe-Taylor J., and Widmer G. 2004. Using String Kernels to Identify Famous Performers from their Playing Style, Proceedings of the 15th European Conference on Machine Learning, pp. 385-395.

(Stamatatos, 2005) Stamatatos, E. and Widmer, G. 2005. Automatic Identification of Music Performers with Learning Ensembles. *Artificial Intelligence* 165(1), 37-56.

(Tobudic, 2003) Tobudic A., Widmer G. 2003. Relational IBL in Music with a New Structural Similarity Measure, Proceedings of the International Conference on Inductive Logic Programming, pp. 365-382, Springer Verlag.

(Todd, 1992) Todd, N. 1992. The Dynamics of Dynamics: a Model of Musical Expression. Journal of the Acoustical Society of America 91(6), pp.3540-3550.

(Sundberg, 1983) Sundberg J., Askenfelt A. & Frydén L. 1983. "Musical performance: A synthesis-by-rule approach," Computer Music Journal Vol. 7(1), pp. 37-43.

(Van Baelen, 1996) Van Baelen, E.; De Raedt, Luc. 1996. Analysis and prediction of piano performance using inductive logic programming, Muggleton, S (ed.), 6th International Workshop on Inductive Logic Programming, Lecture Notes in Computer Science 1314, pp. 55-71

(Widmer, 2001) 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, pp. 552-563. Springer Verlag.

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

(Widmer, 2004) Widmer, G., and Goebl, W. 2004. "Computational models of expressive music performance: The state of the art," Journal of New Music Research 33(3), pp. 203-216.