An approach to predicting bowing control parameter contours in violin performance

Esteban Maestre, Rafael Ramírez
Music Technology Group, Universitat Pompeu Fabra, Barcelona, SPAIN

June 27, 2009

Abstract

We present a machine learning approach to modeling bowing control parameter contours in violin performance. Using accurate sensing techniques we obtain relevant timbre-related bowing control parameters such as bow transversal velocity, bow pressing force, and bow-bridge distance of each performed note. Each performed note is represented by a curve parameter vector and a number of note classes are defined. The principal components of the data represented by the set of curve parameter vectors are obtained for each class. Once curve parameter vectors are expressed in the new space defined by the principal components, we train a model based on inductive logic programming, able to predict curve parameter vectors used for rendering bowing controls. We evaluate the prediction results and show the potential of the model by predicting bowing control parameter contours from an annotated input score.

1 Introduction

The interaction between a performer and his/her musical instrument is difficult to model. This is particularly true in the case of excitation-continuous musical instruments, for which variations of sound are achieved by continuous modulations of the physical actions directly involved in sound production mechanisms, i.e. instrumental gestures [3]. Because of the complex and continuous nature of gestures involved in the control of bowed-string instruments (often considered among the most articulate and expressive), analysis of bowed-string instrumental gestures has been an active and challenging topic of study for several years. Particularly for the case of violin, recent studies have taken the advantage of currently available motion tracking and force sensing techniques for providing accurate gesture data [19] [11] [6].

Violin bowing control data analysis has been approached recently by several works. Rasamimanana and coworkers [14] used bow acceleration extrema for automatic bow stroke classification. In a similar fashion, Young [18] extends the classification to different bowing techniques by extracting the principal components of raw acceleration and strain gage sensor data. None of these approaches is aligned toward a generative model able to also provide means for the automation of bowing control parameter rendering.

A first attempt to create synthetic bowing control parameter contours from an annotated score is found in the work by Chafe [4], where the author presents an algorithm for rendering a number of violin performance gesture parameter contours (including both left and right hand) by concatenating short segments following a number of hand-made rules. Following the same line, extensions dealing with left hand articulations and string changes were introduced by Jaffe and Smith [9]. Both approaches lack real performance data -driven definition of segment contours parameters. A more recent study working with real data is found in the work by Demoucron [6], where bow velocity and bow force contours of different bow strokes are quantitatively characterized and reconstructed mostly using sinusoidal segments. Flexibility limitations of the proposed contour representation may impede to easily generalize its application to other bowing techniques.

Maestre [10] points directions toward a general framework for the automatic characterization of real instrumental gesture cue contours using parametric Bézier curves, foreseeing them as a more powerful and flexible basis for contour shape representation (see their use for speech prosody modeling by Escudero and coworkers [7]). Aimed at providing means for reconstructing contours by concatenating short curve units, implied a structured representation as opposed as the work presented by Battey [1], dealing with audio perceptual attributes. Later, Maestre and coworkers [12] used Bézier concatenated curves for pursuing a model for different note-to-note articulation classes in singing voice performance.

In this paper, we present a machine-learning-based framework for modeling bowing control parameter contours (bow velocity, bow force, and bow-bridge distance) using concatenated Bézier cubic curves. We have represented in Figure 1 an overview of the system. Curve parameter extraction is carried out automatically which provides a representation usable both in gesture analysis and synthesis applications. Contour modeling is adapted to specific note classes, having notes classified in a previous step by attending to a number of musical contextual characteristics. The dimensionality of the spaces where different contour parameter vectors reside is then reduced by analyzing their principal components. We apply inductive logic programming techniques to predict those curve parameter vectors when expressed in the new space. We evaluate the prediction results and show the potential of the model by predicting bowing control parameter contours from an annotated input score.

The rest of the paper is organised as follows: in Section 2, we present the methodology followed for carrying out contour parameter automatic extraction. Section 3 gives details on the prediction model and reports on obtained results. We conclude by pointing out possible extensions and applications.

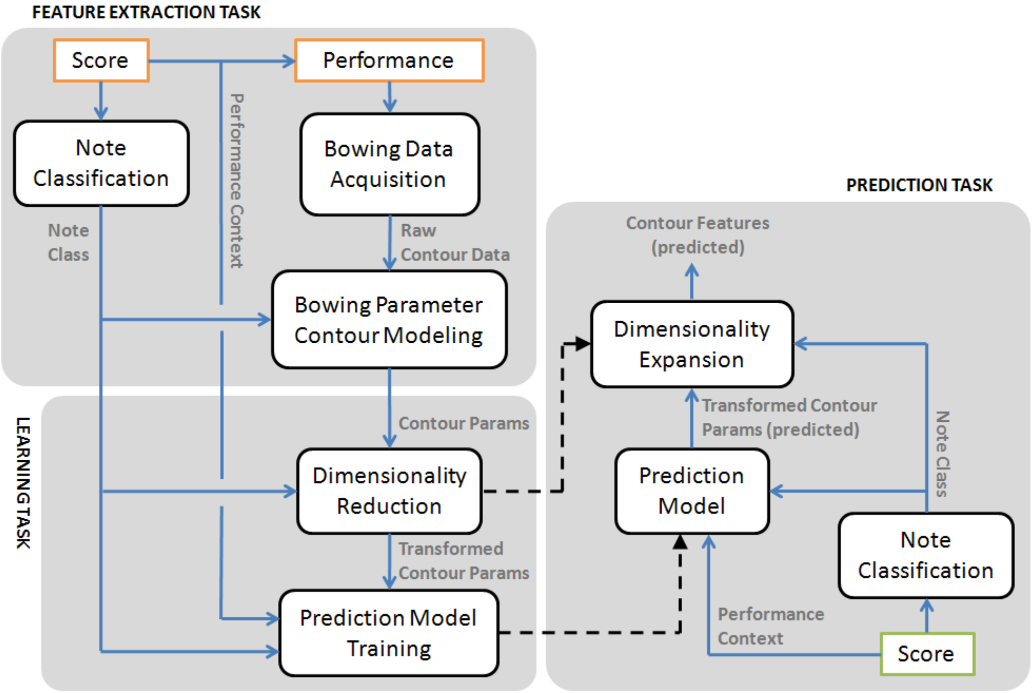


Figure 1: Overview of the proposed system, in which the learning and prediction tasks have been illustrated separately

2 Feature extraction

2.1 Bowing data acquisition

Bowing data acquisition was performed by means of a commercial electromagnetic field sensing device as reported in previous work by Maestre and coworkers [11], extracting bow force by applying the techniques presented by Guaus et al. [8] and by Demoucron [6]. Recording scripts (including both exercises and short musical pieces) were designed to cover four different articulation types (*détaché*, *legato*, *staccato*, and *saltato*), three different dynamics, and varied note durations in different performance contexts (attending to bow direction changes and rests). Score-performance alignment was carried automatically by means of a dynamic programming (based on the *Viterbi* algorithm [17]) adaptation of the procedure introduced in [11] plus manual correction when needed for ensuring the appropriate segmentation of bow velocity, bow force, and β ratio contours of around 10K notes. The β ratio is defined as the proportion between the effective string length (distance from the finger and the bridge) and the measured distance between the bow hair and the bridge (see [11]). We have depicted in Figure 2 examples of acquired raw data corresponding to the four different articulation types considered in our corpus.

2.2 Note classification

Concerning different score annotation -based characteristics of the notes in the corpus, we perform a classification that will define different classes of notes for which specific gesture models will be later constructed. The basis for classifying note samples is divided into two main groups: *intrinsic* aspects and *contextual* aspects.

Regarding *intrinsic* note characteristics, we considered three aspects. The first and most important is the *articulation type* (expressed as *[ART]*), and we have considered four different articulations: *détaché*, *legato*, *saltato*, and *saltato*. Secondly, we considered the *dynamics type* (expressed as *[DYN]*), comprising *piano*, *mezzoforte*, or *forte*. The last of this group is the *bow direction* (expressed as *BD*) including *downwards* and *upwards*.

In terms of what we call *contextual* characteristics, we considered two main aspects: which is the position of a note within a bow (e.g. in *legato* articulation, several notes are played successively without any bow direction change), and which is the position of a note with respect to *rest* segments (e.g. silences).

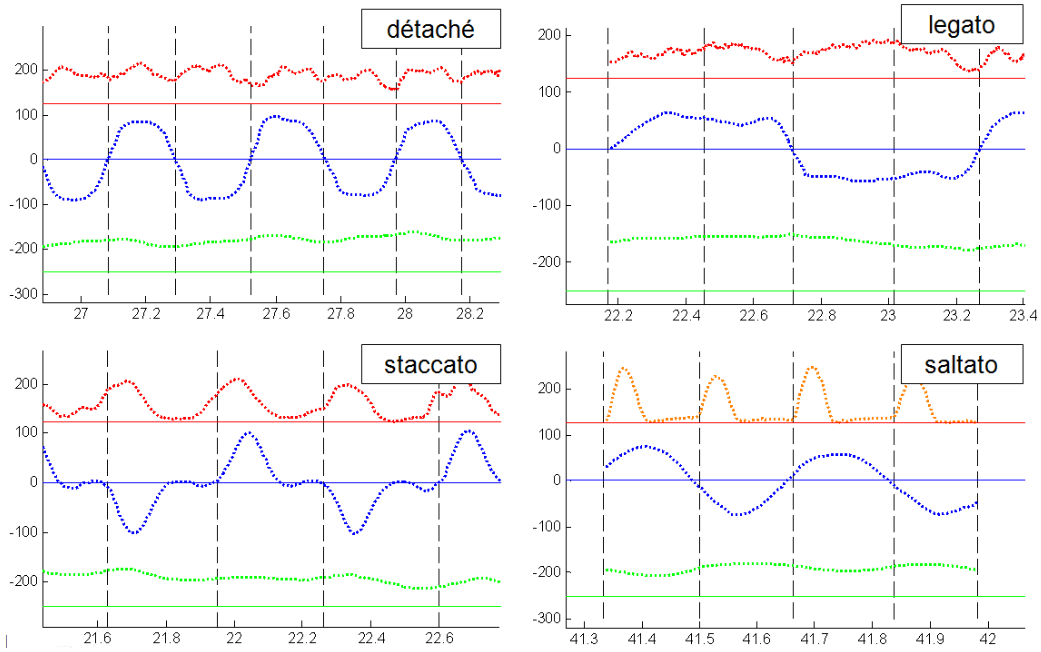


Figure 2: Captured bowing control parameter cues for each of the four articulation types we are considering. From top to bottom: bow force (expressed in $50N$ units), bow transversal velocity (cm/s), and bow-bridge distance ($25cm$ units) are depicted with dashed curves. Solid horizontal lines represent their respective zero level. Vertical dashed lines represent note onsets and offsets.

For the case of *bow context*, we classify a note as *init* when, in a succession of notes sharing the same bow direction, is played first. A note is classified as *mid* when is played neither first nor last. The class *end* corresponds to notes played in last sequence order, while notes appearing as the only notes within a bow (e.g. in *détaché* articulation) are classified as *iso*. Analogously to the case of bow context, for what we called *phrase context*, we look at successions of notes with no *rest* segments or silences in between. Classified as *init* will be those preceded by a silence and followed by another note, as *mid* those preceded by and followed by a note, as *end* those preceded by a note and followed by a silence, and as *iso* those surrounded by silences.

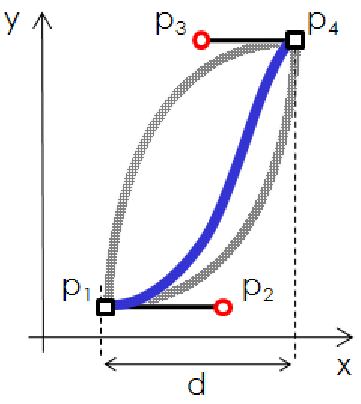
Each possible combination of the classes above represented will lead to a note gesture class C_i characterized by the tuple in (1). Note that not every possible combination of any of the classes considered within each of the five characteristics is feasible in practice, getting a final number of combinations that leads to 102 note classes.

$$C_i = [\, ART_i \;\; DYN_i \;\; BD_i \;\; BC_i \;\; PC_i \,] \tag{1}$$

2.3 Contour representation

Bowing control parameter cue contours of recorded notes are represented in this framework by sequences of a predefined number of units (e.g. lines, curve segments). For the use case presented here, we used constrained cubic Bézier curve segments, similarly as Battey in [1] used for representing perceptual audio parameter contours. In contrast to the lack of consistent score-performance relation in the organization of the representation proposed there, here we define a structured representation applying at note-level. We have represented the basic unit in Figure 3.

Even though it responds to a parametric curve defined by the x-y points p_1, p_2, p_3, p_4, the constrains found in (2) allow defining its shape by a vector $\mathbf{b} = [d\ v_s\ v_e\ r_1\ r_2]$, where d represents the segment duration, v_s represents the starting y-value, v_e represents the ending y-value, and r_1 and r_2 represent the relative x-values of the attractors p_2 and p_3 respectively. Among the reasons why we choose this as the building block for modeling bowing control parameter contours, we highlight its linearity (small changes in curve control points lead to small changes in curve contour) and its flexibility (a diverse number of shapes can be modeled by different values of r_1 and r_2, as it is illustrated by gray curves in Figure 3, which correspond to rather extreme values of r_1 and r_2).



$$\begin{aligned} p_{1y} &= p_{2y} = v_s \\ p_{3y} &= p_{4y} = v_e \\ r_1 &= \frac{p_{2x} - p_{1x}}{d} \\ r_2 &= \frac{p_{4x} - p_{3x}}{d} \\ 0 &\leq r_1 \leq 1 \\ 0 &\leq r_2 \leq 1 \end{aligned} \tag{2}$$

Figure 3: Constrained Bézier cubic segment used as the basic unit in the representation of bowing control parameter contours.

Given a time-series bowing parameter motion cue segment $s(t)$ with $t \in [0,d]$, starting value $s(0) = v_s$, and ending value $s(d) = v_e$, optimal attractor relative x-values r_1^* and r_2^* leading to an optimal approximation $\sigma^*(t)$ of the cue segment can be found via constrained optimization (see [1]).

2.4 Segmentation constrains definition

Contours of the bowing control parameter cues (bow velocity, bow force, and β ratio) corresponding to the samples present in the corpus of each note class C_i have been carefully observed in order to foresee an optimal representation scheme by using the constrained Bézier cubic curve segments presented in previous Section. When observing the data and taking the decisions on the segment sequence arrangement, we aimed at keeping the length of the sequences at a minimum while preserving the fidelity of representation.

For each of the note classes C_i, we have defined a set ρ^i of segmentation constrains composed by three different tuplas ρ_V^i, ρ_F^i, and ρ_β^i, each one defining the number of segments $N^i_{\{V,F,\beta\}}$ and the slope sequence constrain vector $\Delta s^{i*}_{\{V,F,\beta\}}$.The slope sequence constrain vectors $\mathbf{\Delta s}_{\{\mathbf{V},\mathbf{F},\beta\}}{}^{i*}$ define the expected sequence of slope changes for each of the gesture parameter cues. If each i-th segment is approximated linearly, a contour slope sequence $s = [s_i \cdots s_N]$ is obtained. Each pair of successive slope leads to a parameter δs_i that might take three different values: $\delta s_i \in \{-1, +1, 0\}$. The value $\delta s_i = 0$ will be assigned whenever there is no clear expectancy in the relationship between successive slopes s_i and

s_{i+1} (due to observations), while on the presence of a particular expectancy, the value for δs_i will be defined by equation 3. The reader can find the segmentation constrain sets used for pursuing this analysis in Section 4.

$$\delta s_i = sign(s_{i+1} - s_i) \tag{3}$$

2.5 Contour segmentation and fitting

Driven by each previously defined segmentation constrain set ρ^i for each note class C_i, the acquired bowing control parameter raw contours of each note sample are automatically segmented and approximated by appropriate sequences (see Section 2.4) of Bézier cubic curve segments as the one depicted in Figure 3. An overview of the procedure for carrying out contour segmentation and curve fitting is given next.

For every i-th segment (each one presenting a relative duration d_i), the real (aquired) bow velocity contour is denoted as q_i, while the approximated Bézier contour is denoted by σ_i. We set the problem of segmentation and fitting as the optimization task of finding an optimal relative duration vector $\mathbf{d}^* = [d_1^* \cdots d_N^*]$ such that a total cost C is minimized while satisfying that $\sum_{i=1}^{N} d_i = 1$. This is expressed in equation (4), where we defined the approximation error ξ_i for the i-th segment as the mean squared error between the real contour q_i and the optimal Bézier approximation σ_i^* (see Section 2.3), and a weight w_i applied to each ξ_i.

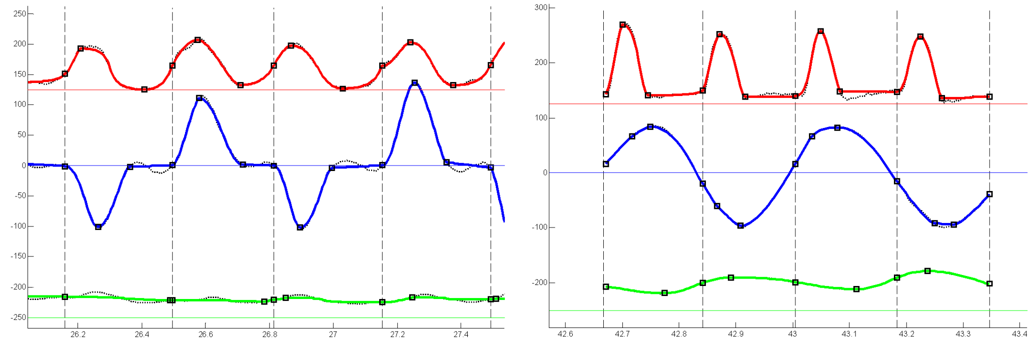
$$\mathbf{d}^* = [d_1^* \cdots d_N^*] = \underset{\mathbf{d}, \sum_{i=1}^{N} d_i = 1}{\operatorname{argmin}} \; C(\mathbf{d}) = \underset{\mathbf{d}, \sum_{i=1}^{N} d_i = 1}{\operatorname{argmin}} \; \sum_{i=1}^{N-1} w_i \xi_i + \xi_N \tag{4}$$

$$\xi_i = \frac{\int_0^{d_i} (q_i(t) - \sigma_i(t))^2 dt}{d_i} \tag{5}$$

The weight w_i applied to each of the first $N-1$ computed ξ_i will depend on the fulfillment of the slope sequence constrains defined by δs^*. Expressed in (6), the weight w_i will be set to a arbitrary value $W >> 1$ in case δs_i (computed from the slopes of the linear approximations of the i-th and $(i+1)$-th segments) does not match the sign of its corresponding δs_i^* (see Section 2.4).

$$w_i = \begin{cases} W >> 1 & \text{if } \frac{\delta s_i}{\delta s_i^*} < 1 \text{ and } \delta s_i^* \neq 0, \\ 1 & \text{otherwise.} \end{cases} \tag{6}$$

The solution for this problem is found by using dynamic programming techniques, in particular based on the *Viterbi* decoding algorithm [17]. As a result, the whole set of note samples corresponding to each of the note classes included in the corpus is analyzed, so that the set of parameters defining the Bézier curve segments that best model each of the bowing control parameter contours of each note is attached to each note sample. Some examples of the results on automatic segmentation and fitting are shown in Figure 4, where acquired bowing parameter cues are compared to their corresponding Bézier approximations for *détaché*, *legato*, *staccato*, and *saltato* articulations.



(a) *Staccato* articulation

(b) *Saltato* articulation

Figure 4: Bowing control parameter cue segmentation and fitting results. In each figure, from top to bottom: acquired bow force (expressed in $50N$ units), bow transversal velocity (cm/s), and bow-bridge distance ($25cm$ units) are depicted with think dashed curves laying behind the modeled, contours, represented by solid thick curves. Solid horizontal lines represent the respective zero levels. Junction points between successive Bézier segments are represented with black squares, while vertical dashed lines represent note onsets and offsets.

3 Prediction model

In this section we describe our approach to learning a model for predicting curve parameter vectors used for rendering bowing controls. A prediction model is built from the obtained bowing control parameter contour representation data of notes present in the database. For each note class C_i, a space with a dimensionality defined by the number of parameters needed for modeling bowing control parameter contours is defined. The bowing control parameter contours are characterised by a number of Bézier cubic segments. Each note in the database is represented by both (1) a contour parameter vector constructed from the obtained curve fitting parameters and (2) a performance context vector containing information about the characteristics of the note used for the prediction of the contour curve parameters. First, a dimensionality reduction based on principal component analysis is performed on the contour parameter vectors of each class. We apply inductive logic programming techniques to predict the contour parameter vectors in the reduced dimensionality space given the performance context vector of a note. This is, we train our prediction model with examples of the form:

$$EX_i = [ART_i, DYN_i, BD_i, BC_i, PC_i, \mathbf{s}_i, \mathbf{p}_i^*]$$

where ART_i, DYN_i, BD_i, BC_i, and PC_i are articulation type, dynamics type, bow direction, bow context, and phrase context, respectively (see Section 2 for a detailed description), and define the note class C_i to which the note belongs. The vector $\mathbf{s}_i$ (see next Section) corresponds to a set of specific score-based annotations (e.g. duration, pitch, etc.) The vector $\mathbf{p}_i^*$ is the contour parameter vector, as expressed in the reduced dimensionality space, to be predicted by the model. Thus, once the prediction model has been trained with a set of examples EX_i, the model's input consists of values for ART_i, DYN_i, BD_i, BC_i, PC_i, and $\mathbf{s}_i$ while its output (e.g. prediction) is a value for $\mathbf{p}_i^*$. Prediction results (i.e. the predicted contour parameter vector in the reduced dimensionality space) are returned back to the original space so that they can be used for rendering bowing parameter contours.

3.1 Data preparation

The procedure that follows applies to any note class C_i. The curve parameters of each note are represented as a vector $\mathbf{p}$ resulting from the concatenation of three curve parameter vectors $\mathbf{p_V}$, $\mathbf{p_F}$, and $\mathbf{p}_\beta$, corresponding to the bow velocity, bow force, and β ratio contours respectively. The dimensionality of these vectors will depend on the number of segments used for modeling each bowing control parameter contour, which is defined by the corresponding segmentation constrain (see Section 2.4). Each of the three parameter vector contains three different subvectors: a first subvector $\mathbf{p^d}$ containing the relative durations d_i/D of each of the segments, a second subvector $\mathbf{p^v}$ containing the the inter-segment y-axis values (starting or ending values $v_{s,i}$ or $v_{e,i}$ of each one of the segments), and a third subvector $\mathbf{p^r}$ containing the pairs of attractor x-value ratios $r_{1,i}$ and $r_{2,i}$.

Once each note sample has been annotated with its corresponding contour parameter vector **p**, we attach to each note a context vector **s** that will be used as the input for predicting the curve parameter vector **p** (output of the prediction model). The performance context vector **s** is defined by equation (7), where D is the note duration (seconds), L_{st} is the effective length of the string (obtained from the string being played and the pitch of the note, and expressed as relative to the total length of the string as going from the nut to the bridge), and BP_{ON} represents the starting bow transversal position of the played note (measured from the frog end of the hair ribbon to the hair ribbon position in contact with the string, and expressed as relative to the total length of the hair ribbon).

$$\mathbf{s} = [\, D \; L_{st} \; BP_{ON}] \tag{7}$$

3.2 Dimensionality reduction

In order to reduce the dimensionality of the space where our prediction model is trained, but while limiting the information loss, we used principal component analysis (PCA) [15] of the space where curve parameter vectors reside. For the reported experiments we have constrained the dimensionality reduction process to keep at least 60% information.

Applying to each note class C_i, the procedure followed for applying PCA is based on eigenvector decomposition of the covariance matrix $\mathbf{C}_{X_0}$ of the mean-subtracted data (arranged into a $M \times N$ matrix $\mathbf{X_0}$ where M represents the dimensionality of the curve parameter vector **p**, and N represents the number of examples). The first step is to subtract the mean from each dimension of the original curve parameter vector examples data matrix **X** for obtaining X_0. Then, we obtained the covariance matrix $\mathbf{C}_{X_0}$ from $\mathbf{X_0}$ as expressed in equation (8). As the covariance matrix $\mathbf{C}_{X_0}$ is orthogonally diagonizable, we obtain the $N \times N$ matrix of eigenvectors **U** by searching for a $N \times N$ diagonal variances matrix **D** accomplishing equation (9). Now, each of the orthogonal principal components defining the new space is represented by a column of **U**, having the the elements in the diagonal matrix D representing the variances of $\mathbf{X_0}$ in terms of the new dimensions defined by the columns of **U**.

$$\mathbf{C}_{X_0} = \frac{1}{N-1}\mathbf{X_0}\mathbf{X_0}' \tag{8}$$

$$\mathbf{C}_{X_0} = \mathbf{U}\mathbf{D}\mathbf{U}^T \tag{9}$$

In the next step, we searched for the $n < N$ dimensions that hold the most significant curve parameter distribution information, say above a percentage threshold t, and project the original mean-subtracted curve parameter vectors $\mathbf{X_0}$ into a new space of lower dimensionality n. For that, we used the transformation matrix $\mathbf{U}^*$ composed by the n most important eigenvectors, found by sorting the columns of $\mathbf{U}$ in decreasing order of their corresponding variance component in $\mathbf{D}$, obtaining $\mathbf{U_S}$ and $\mathbf{D_S}$. The matrix $\mathbf{U}^*$ is constructed by concatenating the first n columns of $\mathbf{U_S}$ that account for the minimum relative variance energy t_n holding $t_n > t$ computed as in equation (10). The new curve parameter examples data matrix $\mathbf{X_0}^*$ is obtained by equation (11).

$$t_n = \frac{\sum_{i=1}^{n} \mathbf{D_S}(i,i)}{\sum_{i=1}^{N} \mathbf{D_S}(i,i)} \tag{10}$$

$$\mathbf{X_0}^* = \mathbf{U_S}'\mathbf{X_0} \tag{11}$$

Different dimensionality reduction ratios were found for the 102 different note classes C_i. As an example, an original dimensionality of $N = 40$ was reduced to $n = 8$ for one of the note classes, having an average dimensionality reduction rate of 81.2% for all classes while keeping the most significant information.

3.3 Learning algorithm

As mentioned before, we apply inductive logic programming techniques to induce a model for predicting contour parameter vectors in the reduced dimensionality space (obtained from applying the PCA projection on the contour parameter data). Training examples consist of the performance context vector values (input) and the n bowing control parameters values after dimensionality reduction (prediction). Given the information preservation preservation constraint $t = 0.6$, the value of n may be different for different note classes. We apply the Tilde's top-down decision tree induction algorithm ([2]). Tilde is a first order logic extension of the C4.5 [13] decision tree algorithm in which instead of testing attribute values at the nodes of the tree, it tests logical predicates. Tilde can also be used to build multivariate regression trees, i.e. trees able to predict a vector. In our case the predicted vector is the vector consisting n values of the bowing control parameters expressed in the new space. Tilde's algorithm induces a first-order logic decision tree taking a set of examples E, a background knowledge B and a query Q as inputs. For each contour parameter group, we apply Tilde with the query $predict(I_1,I_2,I_3,I_4,I_5,I_6,I_7,I_8(O_1,\ldots,O_n))$, where I_1,I_2,I_3,I_4,I_5 are performance context attributes for articulation type, dynamics type, bow direction, bow context, phrase context, and I_6,I_7,I_8 are the components of the performance context vector $\mathbf{s}$ respectively, and $(O_1,\ldots,O_n)$ is the vector consisting of the n contour parameter values in the low dimensionality space.

3.4 Prediction results

We have evaluated the induced model both in the reduced dimensionality space and in the original contour parameter space (after returning data by applying inverse PCA transformation and adding the estimated mean in each original dimension) for each of the 102 note classes. For doing so, we have performed a 5-fold cross validation in which 20% of the training set was held out in turn as test data while the remaining 80% was used as training data.

Table 1 report results for a note class in which the principal component analysis yielded an eight-dimensional space, here n_i represent each of the dimensions. The mean absolute error (MAE) is shown for each dimension. The note class mean MAE (represented as $\overline{MAE}$ is obtained by averaging the mean absolute errors for each of the n dimensions of the principal components space. In order to provide a more meaningful error measure accounting for the relative importance of each of the n dimension's variance, we have computed a weighted average of each dimension's MAE using as a weight the inverse of the corresponding variance, and we have represented it as MAE_D. After returning data to the original contour parameter space, we have also computed the average of the mean absolute errors for each of the initial dimensions, expressed as MAE_0. In Table 2 we show the average of the computed errors for all note classes.

MAE								$\overline{MAE}$	MAE_D	MAE_0
n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8			
0.3362	0.3404	0.1475	0.1869	0.1260	0.1297	0.0881	0.0984	0.2385	0.1631	0.0913

Table 1: Prediction mean absolute errors for a particular note class.

Even though some note classes presenting only a few examples led to worse prediction results, in general we have observed reasonable errors for the majority of the note classes. When returning data to the original contour parameter space,

$avg\overline{MAE}$	$avgMAE_D$	$avgMAE_0$
0.2808	0.1862	0.1353

Table 2: Global prediction errors computed by averaging all 102 note classes.

the errors decrease significantly, mainly due to the fact that the ratio between the previously subtracted mean and the range of the predictions is of great importance for most of the original dimensions.

4 Conclusion

We have introduced a modeling framework for predicting violin bowing control parameter contours from an annotated score. Common patterns observed in bow velocity, bow force, and bow-bridge distance cues lead to the definition of a set of segmentation constrains able to dictate an automatic cue contour segmentation and fitting algorithm adapted to several articulations and contexts. Each note is represented by a high dimensional feature vector from which bowing control parameter contours can be reconstructed. In order to be able to apply a prediction model given a reduced set of note characteristics to be found in an unknown input annotated score, we have reduced the dimensionality of the vectors to be predicted by applying principal component analysis. A prediction model based on inductive logic programming has been applied to the data expressed in the reduced dimensionality space. Predicted contour parameter values have been evaluated both in the low dimensionality space to wihch they were transformed space and in their original space after applying inverse PCA transformation, obtaining very interesting results that enable us to further investigate and apply the modeling framework.

Several extensions to the general methodology presented here remain clear: adding more note articulations, including left-hand gesture analysis, considering further performance context parameters (e.g. playing closer to the tip or to the frog), etc. Also, note classification could be enriched by taking into account more contextual variables, like for instance the preceding and following articulations. Likewise, approaches for automatic definition of segmentation constrains could greatly contribute. Apart from considering the application of the modeling framework to other excitation-continuous instruments, a number of violin use-cases are to be studied. Automatic performance annotation might be useful for expressiveness or style analysis and modeling by applying a similar framework. We also plan to evaluate the system performance for different PCA information loss thresholds in order to determine an optimal prediction performance given the contour description model.

We have successfully used the predicted bowing control parameter contours for driving a violin sound synthesizer based on digital waveguides as described in [16] by using the implementation found in [5]. Although the sound synthesis we used is a very simplified model, obtained synthetic violin performances resulted highly natural-sounding, thus providing further validation of the bowing control parameter contour prediction framework presented here. The reader can listen to preliminar synthetic sound examples in[1].

Acknowledgement. This work was partially supported by Yamaha Corporation, by the European Commission 7th Framework Programme SAME project (no. 215749), and by Spainish Ministry of Education project TIN2006-14932-C01-01.

References

[1] B. Battey, "Bézier spline modeling of pitch-continuous melodic expression and ornamentation," *Computer Music Journal 28:4*, 2004.

[2] H. Blockeelm, L. De Raedtm, and J. Ramon, "Top-down induction of clustering trees," in *Proceedings of the 15th International Conference on Machine Learning*, Wisconsin, USA, 1998.

[3] C. Cadoz and C. Ramstein, "Capture, representation and composition of the instrumental gesture," in *Proc. of ICMC90*, Glasgow, 1990.

[4] C. Chafe, "Simulating performance on a bowed instrument," *CCRMA Tech. Rep. STAN-M48, Stanford Univ.*, 1988.

[5] P. Cook and G. Scavone, "The synthesis toolkit in c++ (stk)." [Online]. Available: http://ccrma.stanford.edu/software/stk

[6] M. Demoucron, "On the control of virtual violins: Physical modeling and control of bowed string instruments," *PhD. dissertation, Universite Pierre et Marie Curie (Paris 6) and the Royal Institute of Technology (KTH)*, 2008.

[7] D. Escudero, V. Cardenoso, and A. Bonafonte, "Corpus-based extraction of quantitative prosodic parameters of stress groups in spanish," in *Proc. of ICASSP02*, Orlando, 2002.

---

[1] `http://www.iua.upf.edu/~emaestre/gestureModels/bowed/violin/spm.zip`

[8] E. Guaus, J. Bonada, A. Pérez, E. Maestre, and M. Blaauw, “Measuring the bow pressing force in a real violin performance,” in *Proc. of ISMA07*, Barcelona, 2007.

[9] D. Jaffe and J. Smith, “Performance expression in commuted waveguide synthesis of bowed strings,” in *Proc. of ICMC95*, Alberta, 1995.

[10] E. Maestre, “Coding instrumental gestures: towards automatic characterization of instrumental gestures in excitation-continuous musical instruments,” *DEA Doctoral pre-Thesis work, Universitat Pompeu Fabra*, 2006.

[11] E. Maestre, J. Bonada, M. Blaauw, E. Guaus, and A. Pérez, “Acquisition of violin instrumental gestures using a commercial emf device,” in *Proc. of ICMC07*, Copenhagen, 2007.

[12] E. Maestre, J. Bonada, and O. Mayor, “Modeling voice articulation gestures in singing voice performance,” *AES 118th Convention*, 2006.

[13] J. Quinlan, “C4.5: Programs for machine learning,” *Morgan Kaufmann*, 1993.

[14] N. Rasamimanana, E. Flety, and F. Bevilacqua, “Gesture analysis of violin bow strokes,” *LNCS Vol.3881*, 2006.

[15] J. Shlens, “A tutorial on principal component analysis,” *Center for Neural Science of New York University, USA and Systems Neurobiology Laboratory within Salk Institute for Biological Studies of La Jolla, CA, USA*, 2009.

[16] J. O. Smith, “Physical audio signal processing, december 2008 edition.” [Online]. Available: http://ccrma.stanford.edu/~jos/pasp/

[17] A. J. Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,” *IEEE Trans. on Information Theory*, April 1967.

[18] D. Young, “Classification of common violin bowing techniques using gesture data from a playable measurement system,” in *Proc. of NIME08*, Genova, 2007.

[19] ——, “A methodology for investigation of bowed string performance through measurement of violin bowing technique,” *PhD Dissertation, Massachusetts Institute of Technology*, 2007.

Appendix

Here we report the detailed list of the bowing control segmentation constrain sets used when analyzing the database. Note that only 17 different constrain sets are reported, being each one corresponding to only the possible combinations of *articulation type*, *bow context* and *phrase context*. Since the corpus covers all 12 feasible combinations of *dynamics type* and *bow direction* for each of the feasible combinations of *articulation type*, *bow context* and *phrase context*, we decided (based on observation of contour shape similarities) to use a unique segmentation constrain set for all 12 combinations of the 4 articulation and the 3 dynamics, leading to a final number of 17 usable sets.

Segmentation constrains		
	[*ART BC PC*]	[*ART BC PC*]
	[*détaché iso init*]	[*détaché iso mid*]
N_V	4	4
Δs^*_V	[-1 0 -1]	[-1 0 -1]
N_F	3	3
Δs^*_F	[-1 -1]	[-1 -1]
N_β	2	2
Δs^*_β	[0]	[0]
	[*détaché iso end*]	[*détaché iso iso*]
N_V	4	4
Δs^*_V	[-1 0 -1]	[-1 0 -1]
N_F	3	3
Δs^*_F	[-1 -1]	[-1 -1]
N_β	2	2
Δs^*_β	[0]	[0]
	[*legato init init*]	[*legato init mid*]
N_V	2	2
Δs^*_V	[-1]	[-1]
N_F	2	2
Δs^*_F	[-1]	[-1]
N_β	2	2
Δs^*_β	[0]	[0]
	[*legato mid mid*]	[*legato end mid*]
N_V	2	2
Δs^*_V	[-1]	[-1]
N_F	2	2
Δs^*_F	[0]	[-1]
N_β	2	2
Δs^*_β	[0]	[0]
	[*legato end end*]	[*staccato iso init*]
N_V	2	3
Δs^*_V	[-1]	[-1 +1]
N_F	2	3
Δs^*_F	[-1]	[-1 +1]
N_β	2	2
Δs^*_β	[0]	[0]

Table 3: Bowing control parameter contour segmentation constrains, defined for each of the different combinations of *articulation* (*ART*), *bow context* (*BC*), and *phrase context* (*PC*).

Segmentation constrains (cont’d)		
	[*ART BC PC*]	[*ART BC PC*]
	[*staccato iso mid*]	[*staccato iso end*]
N_V	3	3
Δs^*_V	[-1 +1]	[-1 +1]
N_F	3	3
Δs^*_F	[-1 +1]	[-1 +1]
N_β	2	2
Δs^*_β	[0]	[0]
	[*staccato iso iso*]	[*saltato iso init*]
N_V	3	3
Δs^*_V	[-1 +1]	[-1 -1]
N_F	3	3
Δs^*_F	[-1 +1]	[-1 +1]
N_β	2	2
Δs^*_β	[0]	[0]
	[*saltato iso mid*]	[*saltato iso mid*]
N_V	3	3
Δs^*_V	[-1 -1]	[-1 -1]
N_F	3	3
Δs^*_F	[-1 +1]	[-1 +1]
N_β	2	2
Δs^*_β	[0]	[0]
	[*saltato iso iso*]	
N_V	3	
Δs^*_V	[-1 -1]	
N_F	3	
Δs^*_F	[-1 +1]	
N_β	2	
Δs^*_β	[0]	

Table 4: Bowing control parameter cue segmentation constrains, defined for each of the different combinations of *articulation* (*ART*), *bow context* (*BC*), and *phrase context* (*PC*) (cont’d).