An evaluation of score descriptors combined with non-linear models of expressive dynamics in music

Carlos Eduardo Cancino Chacón and Maarten Grachten

Austrian Research Institute for Artificial Intelligence
http://www.ofai.at/research/impml/

Abstract. Expressive interpretation forms an important but complex aspect of music, in particular in certain forms of classical music. Modeling the relation between musical expression and structural aspects of the score being performed, is an ongoing line of research. Prior work has shown that some simple numerical descriptors of the score (capturing dynamics annotations and pitch) are effective for predicting expressive dynamics in classical piano performances. Nevertheless, the features have only been tested in a very simple linear regression model. In this work, we explore the potential of a non-linear model for predicting expressive dynamics. Using a set of descriptors that capture different types of structure in the musical score, we compare the predictive accuracies of linear and non-linear models. We show that, in addition to being (slightly) more accurate, non-linear models can better describe certain interactions between numerical descriptors than linear models.

Keywords: Musical expression, Non-linear Basis Models, Artificial Neural Networks, Computational models of music performance

1 Introduction

Performances of written music by humans are hardly ever precise acoustical renderings of the notes in the score, as a computer would produce —nor are they expected to be. A natural human performance involves an interpretation of the music, in terms of structure, but also in terms of affective content [5, 22], which is conveyed to the listener by local variations in tempo and loudness, and (depending on the expressive possibilities of the instrument) the timing, articulation, and timbre of individual notes.

Musical expression is a complex phenomenon. Becoming an expert musician takes many years of training and practice, and rather than adhering to explicit rules, achieved performance skills are to a large degree the effect of implicit, procedural knowledge. That is not to say that regularities cannot be found in the way musicians perform music. Decades of empirical research have identified a number of factors that jointly determine the way a musical piece is rendered [21, 11]. For example, aspects such as phrasing [29], meter [25], but also intended emotions [20], all have an effect on expressive variations in music performances.

A better understanding of musical expression is not only desirable in its own right. The potential role of computers in music creation will also depend on accurate computational models of musical expression. For example, music software such as MIDI sequencers and music notation editors may benefit from such models in that they enable automatic or semi-automatic expressive renderings of musical scores.

Several methodologies have been used to study musical expression. Complementary to controlled experiments that investigate a single aspect of performance, data mining and machine learning paradigms set out to discover regularities in musical expression using data sets comprising musical performances [31, 23]. Given the implicit nature of expressive performance skills, the benefit of the latter approach is that it may reveal patterns that have gone as of yet unnoticed, because perhaps they do not relate in any obvious ways to existing scholarly knowledge about expressive performance.

A computational framework has been proposed in [13], to model the effect of structural aspects of a musical score on expressive performances of that score, in particular expressive dynamics (the relative intensity with which the notes are performed). This framework, referred to as the Linear Basis Model (LBM), follows the machine learning paradigm in that it estimates the parameters of a model from a set of recorded music performances, for which expressive parameters such as local loudness, tempo, or articulation, can be measured or computed.

An important characteristic of the LBM is its use of *basis functions* as a way to describe structural properties of a musical score, ranging from the metrical position of the notes, to the presence and scope of certain performance directives. For instance, a basis function for the performance directive *forte* (f), may assign a value of 1 to notes that lie within the scope of the directive, and 0 to notes outside the scope. Another basis function may assign a value of 1 to all notes that fall on the first beat of a measure, and 0 to all other notes. But basis functions are not restricted to act as indicator functions; They can be any function that maps notes in a score to real values. For example, a useful basis function proves to be the function that maps notes to (powers of) their MIDI pitch values. Given a set of such basis functions, each representing a different aspect of the score, the intensity of notes in an expressive performance is modeled simply as a linear combination of the basis functions. The resulting model has been used for both predictive and analytical purposes [13, 15].

The original formulation of the LBM used a least squares (LS) regression to compute the optimal model parameters. A probabilistic LBM using the Bayesian linear regression assuming zero mean Gaussian priors with isotropic covariance was presented in [15], and then expanded to Gaussian priors with arbitrary mean and covariance in [4].

Although the linear model produces surprisingly good results given its simplicity, a question that has not been answered until now is whether the same basis function framework can benefit from a more powerful, non-linear model. It is conceivable that *interactions* of score properties produce an effect on performance, rather than each of the properties in isolation. Moreover, it may be that

certain properties covary with musical expression, but not in a linear fashion. Therefore, in this paper, we propose a Non-Linear Basis Model (NLBM), that enables non-linear combinations of basis functions through the use of supervised Feedforward Neural Networks (FFNN). These models have been successful in a variety of tasks, ranging from handwritten digit recognition to robot control. FFNNs are powerful models for learning non-linear transformations: with enough hidden units they can represent arbitrarily complex but smooth functions.

Thus, the purpose of this paper is to investigate whether the basis-function modeling approach to expressive dynamics benefits from non-linear connections between the basis-functions and the targets to be modeled. To this end, we run a comparison of the LBM and the NLBM approaches on a data set of professional concert performances of Chopin's piano works. Apart from the predictive accuracy of both models, we present a (preliminary) qualitative interpretation of the results, by way of a sensitivity analysis of the models.

The outline of this paper is as follows: In Section 2, we discuss prior work on computational models of musical expression. In Section 3, the basis-function modeling approach for musical expression is presented in some more detail. A mathematical formulation of the presented non-linear model is provided in Section 4. In Section 5, we describe the experimental comparison mentioned above. The results of this experimentation are presented and discussed in Section 6. Conclusions are presented in Section 7.

2 Related Work

Musical performance represents an ongoing research subject that involves a wide diversity of scientific and artistic disciplines. On the one hand, there is an interest in understanding the cognitive principles that determine the way a musical piece is performed [5, 22] such as the effects of musical imagery in the anticipation and monitoring of the performance of musical dynamics [2]. On the other hand, computational models of expressive music performance attempt to investigate the relationships between certain properties of the musical score and performance context with the actual performance of the score [32]. These models can serve mainly analytical purposes [30, 33], by showing the relation between structural properties of the music and its effect in the performance of such music, mainly predictive purposes [28], i.e. the models are used to render expressive performances, or both [17, 7, 13]. Computational models of music performance tend to follow two basic paradigms: *rule based* approaches, where the models are defined through music-theoretically informed rules that intend to map structural aspects of a music score to quantitative parameters that describe the performance of a musical piece, and *data-driven* (or *machine learning*) approaches, where the models try to infer the rules of performance from analyzing patterns obtained from (large) datasets of observed (expert) perfomances [31, 14].

One of the most well-known rule-based systems for musical music performance was developed at the Royal Institute of Technology in Stockholm (referred to as the KTH model) [10]. This system is top-down approach that describes

expressive performances using a set of (music theoretically sound/cognitively plausible) performance rules that predict aspects of timing, dynamics and articulation, based on a local musical context. On the other hand, the model proposed in this paper represents a bottom-up approach that uses a lower level encoding of a musical score in order to learn how different aspects of the score contribute to generate an expressive performance of a musical piece.

Among the machine learning methods for musical expression is the model proposed by Bresin [3]. This model uses artificial neural networks (NNs) in a supervised fashion in two different contexts: 1) to learn and predict the rules proposed by the KTH model and 2) to learn the performing style of a professional pianist using an encoding of the KTH rules as inputs. As in the case of the KTH model, the NLBM proposed in this paper uses a lower level representation of the score, and makes less assumptions on how the different score descriptors contribute to the expressive dynamics.

On the other hand, Van Herwaarden et al. [18] present an unsupervised approach to modeling musical dynamics using restricted Boltzmann machines. This approach uses a piano roll representation of musical scores to explain the musical dynamics of performed piano music. In order to to predict expressive dynamics of a score, the features learned by this model are trained in a supervised fashion using LS regression. The choice of a note-centered representation of a musical score makes this system able to model harmonic context based on relative pitch, but insensitive to absolute pitch. Furthermore, this encoding of a score does not include performance directives written by the composer, such as dynamics or articulation markings (such as *piano*, staccato, etc). Both the KTH system and previous work on LBMs have shown that the encoding of pitch and dynamics/articulation markings plays an important role in the rendering of expressive performances.

A broader overview of computational models of expressive music performance can be found in [32, 14].

3 The Basis-Function Model of Expressive Dynamics

In this section, we describe the basis-function modeling (BM) approach, independent of the linear/non-linear nature of the connections to the expressive parameters. We consider a *musical score* a sequence of elements [13]. These elements include note elements (e.g. pitch, duration) and non-note elements (e.g. dynamics and articulation markings). The set of all note elements in a score is denoted by $\mathcal{X}$. Musical scores can be described in terms of *basis functions*, i.e. numeric descriptors that represent aspects of the score. Formally, we can define a basis function φ as a real valued mapping $\varphi\colon \mathcal{X} \mapsto \mathbb{R}$. In a similar way, musical expression is characterized in a quantitative way by a number of *expressive parameters*. In particular, expressive dynamics is conveyed by the MIDI velocities of the performed notes. Further expressive parameters capture aspects of note timing and local tempo (e.g. inter-onset intervals between consecutive notes), and articulation (the proportion of the duration of a note with respect to its

inter-onset interval). Although the basis-function approach can be applied without any alteration to model all of these expressive parameters, the focus in this study will be on expressive dynamics. By defining basis functions as functions of notes, instead of functions of time, the BM framework allows for modeling forms of music expression related to simultaneity of musical events, like the micro-timing deviations of note onsets in a chord, or the melody lead [12], i.e. the accentuation of the melody voice with respect to the accompanying voices by playing it louder and slightly earlier.

The BM framework relies on the simplifying assumption that given all score information, the expressive parameters for each note are independent from those of other notes. This assumption implies that temporal dependencies within parameters are not explicitly modeled. One advantage of non-linear models over previous work is that this framework allows for modeling of mutual dependencies between expressive parameters.

Figure 1 illustrates the idea of modeling expressive dynamics using basis functions schematically. Although basis functions can be used to represent arbitrary properties of the musical score (see Section 3.1), the BM framework was proposed with the specific aim of modeling the effect of *dynamics markings*. Such markings are hints in the musical score, to play a passage with a particular dynamical character. For example, a *p* (for *piano*) tells the performer to play a particular passage softly, whereas a passage marked *f* (for *forte*) should be performed loudly. Such markings, which specify a constant loudness that lasts until another such directive occurs, are modeled using a step-like function, as shown in the figure. A gradual increase/decrease of loudness (*crescendo*/*diminuendo*) is indicated by right/left-oriented wedges, respectively. Such markings are encoded by ramp-like functions. A third class of dynamics markings, such as *marcato* (i.e. the "hat" sign over a note), or textual markings like *sforzato* (*sfz*), or *forte piano* (*fp*), indicate the accentuation that note (or chord). This class of markings is is represented through (translated) unit impulse functions. In the BM approach, the expressive dynamics (i.e. the MIDI velocities of performed notes) are modeled as a combination of the basis functions, as displayed in the figure.

3.1 Groups of basis functions

As stated above, the BM approach encodes a musical score into a set of numeric descriptors. In the following, we describe various groups of basis functions, each group representing a different aspect of the score. This list should by no means be taken as an exhaustive (or accurate) set of features for modeling musical expression. It is a tentative list that encodes basic information, either directly available, or easily computable from a symbolic representation of the musical piece (such as MusicXML).

I **Dynamics markings**. Bases that encode dynamics markings, such as shown in figure 1. For each of the constant loudness markings (*p*, *pp*, *f* etc.), two additional ramp-function are included that allows for a gradual change towards the loudness level indicated by the marking. Such bases are referred

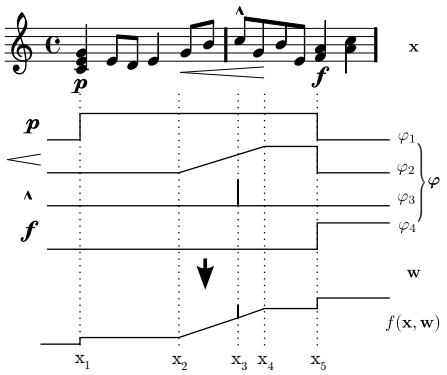

Fig. 1. Schematic view of expressive dynamics as a function $f(\mathbf{x}, \mathbf{w})$ of basis functions φ, representing dynamic annotations

to as *anticipation* functions, and we distinguish between *long* and *short* anticipations, according to how gradual is the change towards the target dynamics marking. Additionally, basis functions that describe gradual changes in loudness, such as *crescendo* and *diminuendo*, are represented through a combination of a ramp function, followed by a constant (step) function, that continues until a new constant dynamics marking (e.g. f) appears, as illustrated by φ_2 in Figure 1.

II **Polynomial pitch model**. Grachten et al. [13] proposed a third order polynomial model to describe the dependency of dynamics on pitch. This model can be integrated in the BM approach by defining each term in the polynomial as a separate basis function, i.e. "pitch", "pitch^2", and "pitch^3".

III **Vertical neighbors**. Two basis functions that evaluate to the number of simultaneous notes with lower and higher pitches, respectively.

IV **IOI**. The inter-onset-interval (IOI) is the time between the onsets successive notes; For note i, three basis functions encode the IOIs between $(i, i-1)$, $(i-1, i-2)$, and $(i-2, i-3)$, respectively.

V **Ritardando**. Encoding of markings that indicate gradual changes in the tempo of the music; Includes functions for *rallentando*, *ritardando*, *accelerando*.

VI **Slur**. Description of *legato* articulations, which indicate that musical notes are performed smoothly and connected, i.e. without silence between each note. The encoding of this bases functions is through parabolic functions that act locally where such a slur is present on the score.

VII **Duration**. A basis function that encodes the duration of a note.

VIII **Rest**. Indicates whether notes precede a rest.

IX **Metrical**. Representation of the time signature of a piece, and the position of each note in the bar. For example, the basis function labeled *4/4 beat 0* evaluates to 1 for all notes that start on the first beat in a 4/4 time signature, and to 0 otherwise.

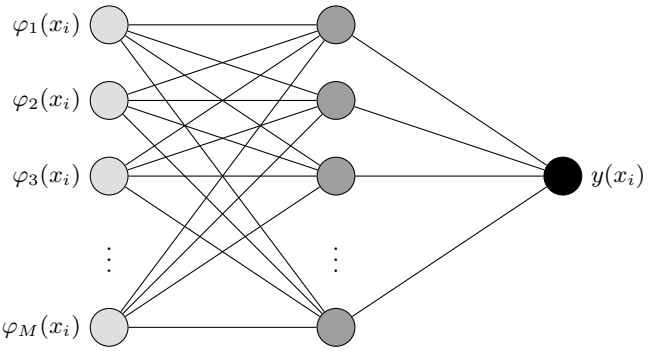

Fig. 2. The architecture of the used NLBM for modeling expressive dynamics

X **Repeat**. Takes into account repeat and ending bars, i.e. explicit markings of that indicate the structure of a piece by indicating the end of a particular section (which can be repeated), or the ending of a piece.

XI **Accent**. Accents of individual notes or chords, such as the *marcato* in figure 1.

XII **Staccato**. Encodes *staccato* markings on a note, an articulation indicating that a note should be temporally isolated from its successor, by shortening its duration

XIII **Grace notes**. Encoding of musical ornaments that are melodically and or harmonically nonessential, but have an embellishment purpose.

XIV **Fermata**. A basis function that encodes markings that indicate that a note should be prolonged beyond its normal duration.

4 Non-Linear Basis Model

In this section we provide a mathematical formulation of the Non-Linear Basis Model (NLBM) model for modeling expressive dynamics. Let $\mathbf{x} = (x_1, \ldots, x_N)^T \in \mathbb{R}^N$ be a vector representing a set of N notes in a musical score and $\mathbf{y} = (y_1, \ldots, y_N)^T \in \mathbb{R}^N$ be a vector representing of an expressive parameter for each note. In this paper, we focus on expressive dynamics, but this framework can be used for other parameters. Let $\boldsymbol{\varphi}(x_i) = (\varphi_1(x_i), \ldots, \varphi_M(x_i))^T \in \mathbb{R}^M$ be a vector whose elements are the values of the basis functions for note x_i. The influence of these basis functions in the expressive parameter can be modeled in a non-linear way using the framework of Feed Forward Neural Networks (FFNNs). These neural networks can be described as a series of functional transformations [1], i.e. a series of non-linear activations of linear combinations of the inputs. Using this formalism, we can write the parameter y as the output of a fully-connected FFNN with L *hidden layers* as

$$y(x_i, \mathbf{w}) = f^{(L)}\left(\sum_{j=1}^{D_L} w_j^{(L)} h_j^{(L-1)}(x_i) + w_0^{(L)}\right), \tag{1}$$

where $\mathbf{h}^{(l)}(x_i) \in \mathbb{R}^{D_l}$ is the activation of the l-th hidden layer, whose k-th component is given by

$$h_k^{(l)}(x_i) = f^{(l)}\left(\sum_{j=1}^{D_l} w_{kj}^{(l)} h_j^{(l-1)}(x_i) + w_{k0}^{(l)}\right), \tag{2}$$

and activation of the first hidden layer is then given as a function of the basis functions as

$$h_k^{(1)}(x_i) = f^{(1)}\left(\sum_{j=1}^{M} w_{kj}^{(1)} \varphi_j(x_i) + w_{k0}^{(1)}\right). \tag{3}$$

The set of all parameters is denoted by $\mathbf{w}$, where $\mathbf{w}^{(l)} = \{w_0^{(l)}, w_1, \ldots, w_{D_l}^{(l)}\}$ are the parameters of the l-th hidden layer[1], and $f^{(l)}$ represent the activation function of the l-th layer. Common (non-linear) activation functions are sigmoid, hyperbolic tangent, softmax and rectifier ($ReLU(x) = \max(0, x)$). Since we are using the FFNN in a regression scenario, the activation function of the last hidden layer is set to the identity function, i.e. $f(x) = x$ [1]. Figure 2 shows the scheme of an FFNN with one hidden layer.

Given a set of training data consisting of input $\mathbf{x}$ and target data $\mathbf{t}$, the model parameters can be estimated in a supervised way by minimizing a loss function, as

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}}\, \mathcal{L}(\mathbf{y}(\mathbf{x}, \mathbf{w}), \mathbf{t}). \tag{4}$$

A usual loss function for supervised regression problems is the *mean squared error* (MSE), i.e.

$$\mathcal{L}_{MSE}(\mathbf{y}, \mathbf{t}) = \frac{1}{N}\sum_i (y_i(\mathbf{x}, \mathbf{w}) - t_i)^2. \tag{5}$$

As previously stated, the NLBM is able to model mutual dependencies between the basis functions. The output of the model can be written as a linear combination of the last hidden layer, i.e.

$$y(\mathbf{h}^L, \mathbf{w}^{(l)}) = \sum_{j=1}^{D_L} w_j^{(L)} h_j^{(L-1)} + w_0^{(L)} = {\mathbf{w}^{(L)}}^T \tilde{\mathbf{h}}^{(L-1)}, \tag{6}$$

where $\tilde{\mathbf{h}}^{(L-1)} = \left(1, h_1^{(L-1)}, \ldots, h_{D_L}^{(L-1)}\right)^T$. Since $\tilde{\mathbf{h}}^{(L-1)}$ is a non-linear activation of linear combinations of the input units, it can model the dependencies and interactions of the basis functions. Therefore, we can understand the training of the NLBM as finding Least Squares solution of a non-linear encoding of the input basis functions.

[1] In the machine learning literature $\{w_1, \ldots, w_{D_l}^{(l)}\}$ and $w_0^{(l)}$ are respectively referred to as the set of *weights* and the *bias* of the l-th layer.

5 Experiments

To determine to what degree the model is able to account for expressive dynamics, encoded as MIDI velocities of performed notes (see Section 5.1), the accuracy of the predictions of the trained model was tested using a 10-fold cross validation. We report several measures to characterize the accuracy of the learned models. Firstly, we report MSE, the mean squared error of the predictions, which is the most direct measure of how close the model predictions are to their targets. Secondly the Pearson correlation coefficient (r), expresses how strongly predictions and target are correlated. Lastly, the coefficient of determination R^2, expresses the proportion of variance explained by the model.

5.1 Data Set

The Magaloff corpus [9] consists of the complete Chopin piano solo works performed by the renown Russian-Georgian pianist Nikita Magaloff (1912-1992) during a series of concerts in Vienna, Austria in 1989. These performances were recorded using a Bösendorfer SE computer-controlled grand piano, and then converted into standard MIDI format. These performances have been aligned to their corresponding musical scores. One of the unique properties of this corpus is that the hammer velocities of each performed note have been recorded in a precise way, and converted to MIDI velocities. This dataset comprises more than 150 pieces and over 300,000 performed notes, adding up to almost 10 hours of music.

5.2 Model training

We trained several NLBM models with different configurations. Of the training data in each fold, 70% was used for updating the parameters, and 30% was used as validation set. The model was trained using RMSProp [6]. This method is a mini batch variant of stochastic gradient descent that adaptively updates the learning rate by dividing the gradient by an average of its recent magnitude. In order to avoid overfitting, dropout and early stopping were used. Dropout prevents overfitting and provides a way of approximately combining different neural networks efficiently by randomly removing units in the network, along with all its incoming and outgoing connections. These methods have been effectively used to improve the results in several applications including image processing [26, 19].

The number of hidden units, activation function of the hidden layers and the hyper-parameters (learning rate, batch size and probability of dropout $p_{dropout}$) were empirically selected using a grid search. The results presented below are those of the the best model on the test set. This network has one hidden layer model with 100 $ReLU$ hidden units, and a linear output layer with a single unit, $p_{dropout} = 0.5$, a learning rate of 0.0001 a batch size of 16000 and was trained for an average of 1037 epochs. It is interesting to notice that with the current training methods, the accuracy of the model was not benefitted by the addition of more hidden layers.

Model	MSE	r	R^2
LBM	0.780	0.472	0.223
LBM (Bayesian)	0.774	0.475	0.226
LBM (best regularized)	0.771	0.477	0.228
NLBM	**0.757**	**0.492**	**0.242**

Table 1. Predictive results for MIDI Velocity, averaged over a 10-fold cross-validation on the Magaloff piano performance corpus. A smaller value of MSE is better, while larger r and R^2 means better performance.

The LBM models were trained using the original LS solution, a regularized LS that imposes a constraint in the l_2 norm on the model parameters [1] and the Bayesian LBM reported in [15]. The damping coefficient for the regularized LS was selected empirically through a grid search, and the reported results correspond to those with the lowest MSE on the test set (denoted as "best regularized" in Table 1).

6 Results and Discussion

In this section, we present and discuss the results of the cross-validation experiment. We first present the predictive accuracies, and continue with a more qualitative analysis of the results.

Table 1 shows the accuracy the LBM and the NLBM Models in the 10-fold cross-validation scenario. All three accuracy measures show that the NLBM model gives a small but consistent improvement over all LBM models. A t-test was performed over the MSE, showing that the difference between the LBM with lowest MSE (the regularized LBM, from now on referred to as the best LBM), and NLBM is statistically significant ($t(316344) = 4.64$ at $p = 3.5 \times 10^{-6}$). This may not seem surprising, since FFNNs are known to be *universal approximators*, i.e. they can uniformly approximate any continuous function on a compact input domain to arbitrary accuracy, given that the model has enough hidden units [1]. However, the limited amount of training data, and the approximate nature of the parameter optimization techniques may well limit the improvement in accuracy in practice.

Prior work has revealed that a major part of the variance explained by the LBM is accounted for by the basis functions that represent dynamic markings and pitch, respectively, whereas other basis functions had very little effect on the predictive accuracy of the model [13]. To gain a better insight into the role that different basis functions play in each of the models, the learned models must be studied in more detail. For the LBM this is straight-forward: Each of the basis-functions is linearly related to the target using a single weight, so that the magnitude of a weight is a direct measure of the impact of the corresponding basis-function on the target. In a non-linear model such as the NLBM, the weights of the model cannot be interpreted in such a straight-forward way. To accommodate for this, we use a more generic method to analyze the behavior of computational models, referred to as *sensitivity analysis*.

6.1 Sensitivity analysis

In order to account for the effects of the different basis functions, a *variance based sensitivity analysis* was performed on the trained LBM and NLBM models [24]. In this way, the sensitivity of the model as a function of the input basis functions $\boldsymbol{\varphi}$ given the parameters $\mathbf{w}$, i.e. $y = f(\boldsymbol{\varphi} \mid \mathbf{w})$ is explained through a decomposition of the variance of y into terms depending on the input basis functions and their interactions. The *first order sensitivity coefficient* S_{1_i} measures the additive effect of the basis function φ_i in the model output, while S_{T_i}, the *total effect index*, accounts for all higher order effects (interactions) of a factor φ_i. These sensitivity measures are given respectively by

$$S_{1_i} = \frac{V_{\varphi_i}(\mathbb{E}_{\boldsymbol{\varphi}\setminus\varphi_i}(y \mid \varphi_i))}{V(y)} \qquad \text{and} \qquad S_{T_i} = \frac{\mathbb{E}_{\boldsymbol{\varphi}\setminus\varphi_i}(V_{\varphi_i}(y \mid \boldsymbol{\varphi}_i))}{V(y)}, \quad (7)$$

where V_{φ_i} is the variance with respect to the i-th basis function, $\mathbb{E}_{\boldsymbol{\varphi}\setminus\varphi_i}$ is the expected value with respect to all basis functions but φ_i and $V(y)$ is the total variance of y. It can be shown that $\sum_i S_{T_i} \geq 1$, with the equality occurring if the model is linear (as is the case with LBM), and $S_{1_i} = S_{T_i}$. Both quantities are estimated using a quasi-Monte Carlo method proposed by Saltelli et al. [24], that generates a pseudo random sequence of samples using low-discrepancy (Sobol sequences) to estimate the expected values and variances in the above equations.

Table 2 lists the basis functions that contribute the most to the variance of the model, ordered according to S_{T_i} for the best LBM and the NLBM models. These results show that the polynomial model (the basis functions pitch, pitch^2, and pitch^3) and the dynamics annotations (the basis-functions for *f*, *ff*, *ff* and their anticipations, *pp* anticipation, and *sotto voce*) have the strongest impact on the predicted MIDI velocities in the LBM models. This is consistent with findings reported in [13]. The other basis functions in the LBM list pertain to time signatures that occur relatively rarely: 12/8 time signature occurs in 4 pieces; the high S_T values for those bases may well be due to an overfitting of the model to the particularities of those pieces.

The list of bases to which the NLBM model is most sensitive (Table 2, right half) shows a similar pattern, i.e. the strongest effect on the predicted dynamics come from the dynamics annotations, with a smaller contribution from the polynomial pitch model. Comparing the total effect index and the first order sensitivity coefficient shows that the non-linear effects in the NLBM model capture interactions between the certain basis functions, e.g. *diminuendo* (*dim.*) with $S_T = 0.173$ and $S_1 = 0.087$ and *crescendo* (*cresc.*) with $S_T = 0.133$ and $S_1 = 0.051$. These results also suggest an increased total effect index for gradual basis functions (like *cresc.* or *dim.*).

Figure 3 illustrates how the NLBM model can account for interactions between the *cresc.* and *dim.* These bases interact in ca. 28% of the Magaloff corpus. In this context, interaction should be understood as those instances where the value of both basis functions is non-zero at the same time, i.e. when *dim* appears after a *cresc.*, before a new constant loudness dynamics markings appear on the score (see Figure 1 and Section 3.1). The lower half of the figure shows the *cresc.*

LBM			NLBM		
basis function	S_T	S_1	basis function	S_T	S_1
pitch^3	0.187	0.187	*ff*	0.182	0.160
ff	0.112	0.112	*diminuendo*	0.173	0.087
duration	0.085	0.085	*crescendo*	0.133	0.051
pitch	0.081	0.081	*fff*	0.115	0.095
fff	0.080	0.080	*f*	0.095	0.082
f	0.044	0.044	duration	0.082	0.052
pitch^2	0.022	0.022	pitch^3	0.046	0.041
pp	0.021	0.021	*pp*	0.032	0.020
f anticipation long	0.016	0.016	pitch^2	0.017	0.015
ff anticipation long	0.015	0.015	4/4 weak beat	0.016	0.014
12/8 beat 1	0.013	0.013	*p*	0.015	0.008
4/4 weak beat	0.013	0.013	*p* anticipation short	0.014	0.013
fz	0.013	0.013	*f* anticipation long	0.013	0.010
12/8 beat 2	0.012	0.012	*ff* anticipation long	0.013	0.012
accent	0.011	0.011	*mp*	0.012	0.008
12/8 beat 7	0.011	0.011	*p* anticipation long	0.012	0.010
3/4 beat 1	0.011	0.011	pitch	0.012	0.009
12/8 beat 8	0.010	0.010	accent	0.010	0.009
p	0.010	0.010	*fz*	0.010	0.008
6/8 beat 1	0.009	0.009	*mf*	0.010	0.005

Table 2. Basis functions with the largest sensitivity coefficients for the best LBM and NLBM models; Averages are reported over the 10 runs of the cross-validation.

and *dim.* basis functions in two different contexts: *cresc.* alone and the effects of *cresc.* after *dim.* The upper leftmost figure represents the case of the dynamics predicted by the best LBM using the crescendo basis function alone. The upper center figure shows the predicted dynamics by the NLBM using only *cresc.*, while the upper rightmost figure shows the interaction of a *cresc.* after a *dim.* for both NLBM and the best LBM models. Here it is possible to see a diminished effect of the *cresc.* on predicted dynamics by the NLBM when it appears after a *dim.* On the other hand, these results also illustrate the inability of the LBM to model interactions between basis functions. These results also suggest that the NLBM model might be able to capture a more "natural" dynamics curve for basis function that represent gradual changes, like *cresc.*, and polynomial pitch model. The interaction between *cresc.* and *dim.* illustrates how the NLBM model can capture interactions between basis functions that the (simpler) LBM model is not able to describe.

The results in Table 2 suggest that some of the most important basis functions for both the LBM and NLBM correspond to certain rules in the KTH model, as is the case of the polynomial pitch model and the *High Loud* phrasing rule[2].

[2] See Table 1 in [10] for an overview of the rules of the KTH model.

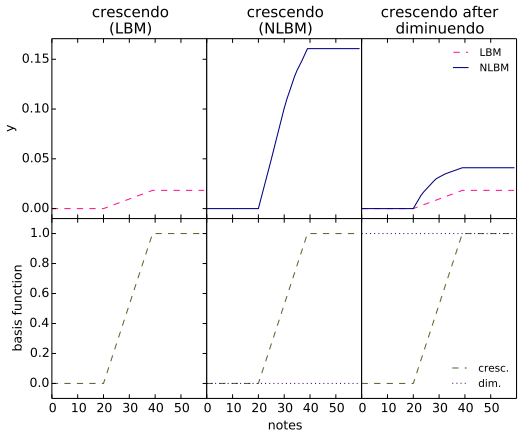

Fig. 3. Example of the effect of the interaction of *crescendo* after a *diminuendo* for both LBM and NLBM models.

7 Conclusions

In this paper, a neural-network based model for musical expression was presented. This model is shown to perform better than previous work based on linear basis models. A sensitivity analysis performed on the two models suggests that the new non-linear approach is able to capture certain interactions of basis functions that cannot be captured in linear models.

In this work, we used simple music-theoretically informed numerical descriptors to capture certain aspects of the score. The results presented above suggest that new basis functions could improve the performance of the presented model.

Additionally, the presented results suggest that the LBM model benefits from bases that contain redundant information (such as long and short anticipation and the polynomial pitch model). It would be interesting to determine wether the NLBM model can capture the similar effects, without recurring to the use of such basis functions, e.g. by using only pitch instead of pitch, pitch^2 and pitch^3. Another interesting question would be to investigate to what degree the nonlinear mappings from basis functions to targets improves the accuracy of the model for non-binary basis functions.

An interesting approach from the music-theoretic side would be the use of basis functions that encode structural (i.e. form) and harmonic information of the piece. Among these basis functions could be the use of key identification algorithms and pattern identification techniques [27].

Furthermore, it would be interesting to use a combination of unsupervised learned features (using Deep Learning) and music-theoretic-informed features for analyzing and predicting expressive music performance, expanding previous work by van Herwaarden et al. [18]. Following previous work on Bayesian LBMs [15], the presented framework can also be expanded into a fully probabilistic approach using the framework of Bayesian neural networks [1].

As stated in Section 3, neither the NLBM, nor the LBM (in both its deterministic and Bayesian formulations) allow for modeling of temporal dependencies within parameters. This issue can be addressed by using a temporal model, such as recurrent neural networks (RNNs) [16], conditional random fields (CRFs) or considering the temporal autocorrelation [8].

Acknowledgments. This work is supported by European Union Seventh Framework Programme, through the Lrn2Cre8 (FET grant agreement no. 610859) and the PHENICX (grant agreement no. 601166) projects.

References

1. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer Verlag, Microsoft Research Ltd. (2006)
2. Bishop, L., Bailes, F., Dean, R.T.: Performing Musical Dynamics. Music Perception 32(1), 51–66 (Sep 2014)
3. Bresin, R.: Artificial neural networks based models for automatic performance of musical scores. Journal of New Music Research 27 (3), 239–270 (1998)
4. Cancino Chacon, C.E., Grachten, M., Widmer, G.: Bayesian linear basis models with gaussian priors for musical expression. Tech. rep. (Oct 2014)
5. Clarke, E.F.: Generative principles in music. In: Sloboda, J. (ed.) Generative Processes in Music: The Psychology of Performance, Improvisation, and Composition. Oxford University Press (1988)
6. Dauphin, Y.N., de Vries, H., Chung, J., Bengio, Y.: RMSProp and equilibrated adaptive learning rates for non-convex optimization. arXiv 1502, 4390 (2015)
7. De Poli, G., S., C., Rodà, A., Vidolin, A., Zanon, P.: Analysis and modeling of expressive intentions in music performance. In: Proceedings of the International Workshop on Human Supervision and Control in Engineering and Music. Kassel, Germany (September 21–24 2001)
8. Eck, D.: Beat Tracking using an Autocorrelation Phase Matrix. In: IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07. pp. 1313–1316. IEEE (Jan 2007)
9. Flossmann, S., Goebl, W., Grachten, M., Niedermayer, B., Widmer, G.: The Magaloff Project: An Interim Report. Journal of New Music Research 39(4), 363–377 (2010)
10. Friberg, A., Bresin, R., Sundberg, J.: Overview of the kth rule system for musical performance. Advances in Cognitive Psychology 2(2–3), 145–161 (2006)
11. Gabrielsson, A.: Music performance research at the millennium. The Psychology of Music 31(3), 221–272 (2003)
12. Goebl, W.: Melody lead in piano performance: expressive device or artifact? Journal of the Acoustical Society of America 110(1), 563–572 (2001)
13. Grachten, M., Widmer, G.: Linear basis models for prediction and analysis of musical expression. Journal of New Music Research 41(4), 311–322 (2012)
14. Grachten, M.: Summary of the Music Performance Panel, MOSART Workshop 2001, Barcelona. In: MOSART Workshop. pp. 1–17 (Mar 2002)
15. Grachten, M., Cancino Chacón, C.E., Widmer, G.: Analysis and prediction of expressive dynamics using Bayesian linear models. In: Proceedings of the 1st international workshop on computer and robotic Systems for Automatic Music Performance. pp. 545–552 (Jul 2014)

16. Graves, A.: Generating Sequences With Recurrent Neural Networks. arXiv 1308, 850 (2013)
17. Grindlay, G., Helmbold, D.: Modeling, analyzing, and synthesizing expressive piano performance with graphical models. Machine Learning 65(2–3), 361–387 (2006)
18. van Herwaarden, S., Grachten, M., de Haas, W.B.: Predicting Expressive Dynamics using Neural Networks. In: Proceedings of the 15th Conference of the International Society for Music Information Retrieval. pp. 47–52 (Jul 2014)
19. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv 1207, 580 (2012)
20. Juslin, P.: Communicating emotion in music performance: a review and a theoretical framework. In: Juslin, P., Sloboda, J. (eds.) Music and emotion: theory and research, pp. 309–337. Oxford University Press, New York (2001)
21. Palmer, C.: Anatomy of a performance: Sources of musical expression. Music Perception 13(3), 433–453 (1996)
22. Palmer, C.: Music performance. Annual Review of Psychology 48, 115–138 (1997)
23. Ramirez, R., Hazan, A.: Rule induction for expressive music performance modeling. In: ECML Workshop Advances in Inductive Rule Learning (September 2004)
24. Saltelli, A., Annoni, P., Azzini, I., Campolongo, F., Ratto, M., Tarantola, S.: Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. Computer Physics Communications 181(2), 259–270 (Feb 2010)
25. Sloboda, J.A.: The communication of musical metre in piano performance. Quarterly Journal of Experimental Psychology 35A, 377–396 (1983)
26. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research 2014(15), 1929–1958 (Jul 2014)
27. Temperley, D.: Music and Probability. Mit Press (2007)
28. Teramura, K., Okuma, H.: Gaussian process regression for rendering music performance. In: Proceedings of the 10th International Conference on Music Perception and Cognition (ICMPC). Sapporo, Japan (2008)
29. Todd, N.: The dynamics of dynamics: A model of musical expression. Journal of the Acoustical Society of America 91, 3540–3550 (1992)
30. Widmer, G.: Machine discoveries: A few simple, robust local expression principles. Journal of New Music Research 31(1), 37–50 (2002)
31. Widmer, G.: Discovering simple rules in complex data: A meta-learning algorithm and some surprising musical discoveries. Artificial Intelligence 146(2), 129–148 (2003)
32. Widmer, G., Goebl, W.: Computational models of expressive music performance: The state of the art. Journal of New Music Research 33(3), 203–216 (2004)
33. Windsor, W.L., Clarke, E.F.: Expressive timing and dynamics in real and artificial musical performances: using an algorithm as an analytical tool. Music Perception 15(2), 127–152 (1997)