Expressive Performance with Bayesian Networks and Linear Basis Models

1. BACKGROUND

Expressive music performance is a complex task that appears to require heterogeneous information, varying from one expressive dimension to another. For example, loudness is guided to a considerable extent by annotations in the score, whereas overall performance tempo is more related to phrasing [1]. Timing and articulation on the other hand may depend more on local score information. The system we present takes a modular approach that treats dynamics, articulation, timing and global tempo in different ways.

2. SYSTEM OVERVIEW

Figure 1 shows an overview of the complete system. From a set of performed pieces (the training data), score *features* and *targets* (loudness, IOI, articulation) are extracted, and used to train the different components of the system. In order to render a new piece, features, tempo and dynamic annotations are extracted from the MusicXML data. The features are used to calculate articulation and tempo predictions, the latter of which is then combined with the tempo annotations to form the tempo of the rendered performance. The dynamic annotations, together with a subset of the features, are used to calculate the loudness of the performance.

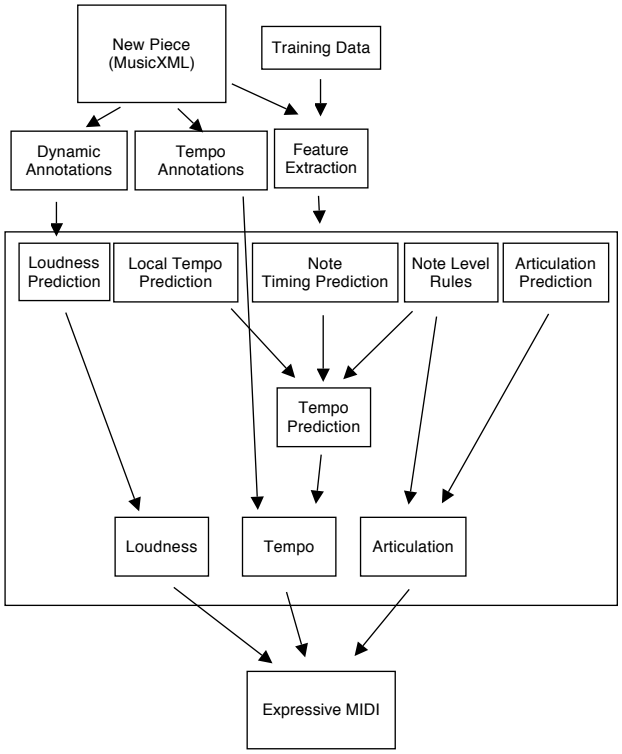

Figure 1. An Overview of the System

3. RENDERING METHOD

In the following we describe the different components of the system and how they are used to form an expressive performance from a score specification.

3.1 Tempo and Articulation Prediction

Tempo and articulation are predicted by a Bayesian Network modeling dependencies between score and performance as conditional probability distributions. The score model comprises simple score descriptors (rhythmic, melodic and harmonic) and higher-level features from the Implication-Realization (I-R) model of melodic expectation by E. Narmour [2] (I-R-labels and a derivation of I-R-closure).

The tempo prediction is consists of three components: 1). *local tempo*, a per-note prediction of long-term tempo changes; 2). *note timing*, note-to-note deviations from local tempo, and 3). *global tempo*, extracted from tempo annotations in the score (e.g. *andante*, *a tempo*). For the prediction of the *local tempo*, the Bayesian network is unfolded in time and an adapted version of the Viterbi Algorithm for Hidden Markov Models is used to predict a series of tempo changes that is optimal with respect to the complete piece. This results in predictions that take the surrounding performance context into account, instead of predictions based only on the local score context. The *note timing* is predicted using only the immediate score context and the value predicted for the previous note. The two series are then combined to form the tempo prediction. In combination with the *global tempo*, the expressive annotations from the score (initial tempo markings, ritardandi, and accelerandi), this constitutes the final tempo of the performance.

Articulation, a very localized aspect of performance, is predicted using only the immediate score context and no previously predicted articulation values. The sets of features used for the three different predictions (local tempo, note timing, and articulation) are specifically tailored to the respective target.

3.2 Note Level Rules

In 2003, Widmer developed a rule extraction algorithm for musical expression [3]. Applied to a collection of Mozart piano sonatas, this resulted in a number of simple rules suggesting expressive change under certain melodic or rhythmic circumstances. We use two of the rules to further enhance the aesthetic qualities of the rendered performances. The *staccato rule* prescribes, that, if two successive notes have the same pitch, and the second of the two is longer,

then the first note is played staccato. The *delay-next rule* states, that, if two notes of the same length are followed by a longer note, the last note is played with a slight delay.

Professional musicians tend to emphasize the melody by playing the melody notes slightly ahead of time, a phenomenon called *melody lead* [4]. To simulate this, we implemented a lead of 13 milliseconds for all melody notes.

3.3 Loudness Prediction

The algorithm used for predicting loudness is based on linear regression. It composes loudness curves by mixing basis functions according to weights learned from musical performances. The basis functions represent musical features as a function of the notes in the score. The fact that the basis functions are functions of the notes rather than functions of time, allows for separate predictions per note, rather than one prediction per time position. The primary benefit of this is that it enables prediction of different loudness values for simultaneous notes. This in turn allows for modeling phenomena like voice leading by increased loudness of the melody, and coloring of chords by varying the loudness of the constituent pitches.

To describe the algorithm, we first define a score $\mathbf{x} = (x_1, \cdots, x_n)$. An element x_i $(1 \leq i \leq n)$ is a vector containing the score onset and pitch of the i-th note, and an indicator value for each dynamic marking. For instance, the indicator for crescendo has a value of one for all notes spanned by crescendos, and zero for all other notes.

We then define a basis function as a function $\varphi_k(\mathbf{x})$ that takes the n elements of $\mathbf{x}$ as arguments to produce a real valued vector of size n. Once a set $\varphi_1(\mathbf{x}), \cdots, \varphi_m(\mathbf{x})$ of m basis functions is fixed, it can be applied to a musical score $\mathbf{x}$ to yield a matrix $\boldsymbol{\varphi}(\mathbf{x}) = (\varphi_1(\mathbf{x}), \cdots, \varphi_m(\mathbf{x}))$ of size $n \times m$, where n is the number of notes in $\mathbf{x}$.

Finally, we define a loudness function y of the score $\mathbf{x}$ and a vector of weights $\mathbf{w} = (w_1, \cdots, w_m)$, such that the loudness is a linear combination of the basis functions:

$$y(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}) \tag{1}$$

For the purpose of loudness prediction, a performance of a score $\mathbf{x}$ (of size n) is regarded as a vector of loudness values $\mathbf{y} = (y_1, \cdots, y_n)$, where y_i $(1 \leq i \leq n)$ is the loudness of the i-th note. In this way, if we have a score $\mathbf{x}$ and an example performance $\mathbf{y}$, we can adapt the model (eq. (1)) to those data by finding appropriate weights $\mathbf{w}$. This problem is known as linear regression, and we will choose $\mathbf{w}$ here as the the least squares solution:

$$\mathbf{w}_{\mathbf{x},\mathbf{y}} = \mathrm{argmin}_{\mathbf{w}} \sum_{i}^{n} (y_i - y(x_i, \mathbf{w}))^2, \tag{2}$$

3.3.1 Basis Functions

We use the basis functions to represent the following features of the score:

- Dynamic annotations in the score (*ff*, *crescendo*, etc)
- Note features: pitch, decorative role, emphasis
- Implication-Realization closure

Within the dynamic annotations we distinguish between three types of dynamics: *impulsive* (e.g. *sf*), *incremental* (e.g. *crescendo*), and *constant* (e.g. *mf*), represented by impulse, ramp, and step functions respectively. These functions are local in the sense that they are non-zero only over a limited range. The ramp and step functions drop to zero at the next annotation of type *constant*. The pitch sequence is taken as a basis function to allow for melody lead by loudness (assuming the melody is typically in higher pitch ranges). The indicator for decorative role and annotated emphasis are included to be able to play grace notes in a different way, and to play emphasized notes louder. Finally, we include a feature indicating closure in the sense of the I-R model [2], which reflects mainly rhythmic, metrical and motivic boundaries. The basis functions representing note features and I-R features are global in the sense that they are non-zero throughout the pieces.

3.3.2 Prediction of $\mathbf{w}$ *for new pieces*

The first step in prediction is finding a solution $\mathbf{w}_{\mathbf{x},\mathbf{y}}$ for each piece-performance pair $(\mathbf{x}, \mathbf{y})$ (according to eq. (2)). We use separate prediction methods to estimate the coefficients of local and global basis functions for a new piece. For global bases we take the median coefficients over the training data. Local bases can vary in number across pieces (depending on the number of dynamic annotations). We predict local coefficients by training an SVM on learned coefficients and n-grams of dynamic annotations.

3.4 Training Data

The system is trained using two corpora of real performance data: 13 complete Mozart piano sonatas, performed by R. Batik and the complete works for solo piano by Chopin, performed by N. Magaloff. All pieces were performed on a Bösendorfer computer-controlled grand piano, in total over 400.000 performed notes, and matched to symbolic representations of the scores.

Training the components and rendering the performance of a new piece is done autonomously without relying on human feedback. The complete process takes a couple of minutes.

4. REFERENCES

[1] N. P. Todd, "A computational model of rubato," *Contemporary Music Review*, vol. 3, no. 1, pp. 69–88, 1989.

[2] E. Narmour, *The Analysis and Cognition of Basic Melodic Structures: The Implication-Realization Model*. University of Chicago Press, 1990.

[3] G. Widmer, "Discovering simple rules in complex data: A meta-learning algorithm and some surprising musical discoveries," *Artificial Intelligence*, vol. 146, no. 2, pp. 129–148, 2003.

[4] W. Goebl, "Melody lead in piano performance: Expressive device or artifact?" *The Journal of the Acoustical Society of America*, vol. 110, no. 1, pp. 563–572, 2001.