Expressive Performance Rendering with Probabilistic Model

Sebastian Flossmann, Maarten Grachten, and Gerhard Widmer

Abstract. We present YQX, a probabilistic performance rendering system based on Bayesian network theory. It models dependencies between score and performance and predicts performance characteristics using information extracted from the score. We discuss the basic system that won the Rendering Contest RENCON 2008 and then present several extensions, two of which aim to incorporate the current performance context into the prediction, resulting in more stable and consistent predictions. Furthermore, we describe the first steps towards a multilevel prediction model: Segmentation of the work, decomposition of tempo trajectories, and combination of different prediction models form the basis for a hierarchical prediction system. The algorithms are evaluated and compared using two very large data sets of human piano performances: 13 complete Mozart sonatas and the complete works for solo piano by Chopin.

1 Introduction

Expressive Performance Modelling is the task of automatically generating an expressive rendering of a music score such that the performance produced sounds both musical and natural. This is done by first modelling the score or certain structural and musical characteristics of it. Then the score model is projected onto performance trajectories (for timing, dynamics, etc.) by a predictive model typically learned from a large set of example performances.

Unlike models in, for instance, rule-based or case-based approaches, the probabilistic performance model is regarded as a conditional multivariate probability distribution. The models differ in the way the mapping between score and performance model is achieved. Gaussian Processes [31], hierarchical Hidden Markov Models [9], and Bayesian networks [37] are some of the techniques used.

Aside from the central problem of mapping the score to the performance, the main challenges in the process are acquisition and annotation of suitable example performances and the evaluation of the results. The data must encompass both precise performance data and score information and must be sufficiently large to be statistically representative. The level of precision required can not yet be achieved through analysis of audio data, which leaves computer-controlled pianos, such as the Bösendorfer CEUS or the Yamaha Disklavier, as the main data source. For the training of our system we have available two datasets recorded on such a device: 13 complete Mozart sonatas, performed by the Viennese pianist R. Batik in 1990, and the complete works for solo piano by Chopin, performed by the Russian pianist N. Magaloff in several live performances at the Vienna Konzerthaus in 1989.

Judging expressivity in terms of “humanness” and “naturalness” is a highly subjective task. The only scientific environment for comparing models according

to such criteria is the annual Performance Rendering Contest RENCON [11], which offers a platform for presenting and evaluating, via listener ratings, state-of-the-art performance modelling systems. Alternatively, rendering systems can be evaluated automatically by measuring the similarity between rendered and real performances of a piece. This, however, is problematic: in some situations small differences may make the result sound unintuitive and completely unmusical, whereas in other situations a rendering may be reasonable despite huge differences.

In this chapter we discuss a prototypical performance rendering system and its different stages: The basic system was entered successfully into the RENCON 2008 rendering contest. Several extensions have been developed, that shed light on the problems and difficulties of probabilistic performance rendering.

2 Related Work

Systems can be compared in terms of two main components: the score representation and the learning and prediction model. The way expressive directives given in the score are rendered also makes a considerable difference in rendered performances, but this is beyond the scope of this article.

Score models – i.e., representations of the music and its structure – may be based either (i) on a sophisticated music theory such as Lerdahl and Jackendoff's Generative Theory of Tonal Music (GTTM) [15] and Narmour's Implication-Realization (IR) model [22] or (ii) simply on basic features capturing some local score characteristics (see, e.g., [6, 9, 10, 31, 36]). Many current models work with a combination of computationally inexpensive descriptive score features and particular structural aspects – mainly phrasal information or simplifications thereof – that are calculated via musicological models. Examples are the models of Arcos and de Mántaras [1], who partially apply the GTTM, and our system [5, 37], which implements parts of the IR model to approximate the phrasal structure of the score.

Regarding the learning and prediction models used, three different categories can be distinguished [38]: Case-based Reasoning (CBR), rule extraction, and probabilistic approaches. Case-based approaches use a database of example performances of music segments. New segments are played imitating stored ones on the basis of a distance metric between score models. Prototypical case-based performance models are *SaxEx* [1] and *Kagurame Phase II* [29]. In [32, 34], a structurally similar system is described that is based on a hierarchical phrase segmentation of the music score. The results are exceedingly good, but the approach is limited to small-scale experiments, as the problem of algorithmic phrase detection is still not solved in a satisfactory way. Dorard et al. [2] used Kernel methods to connect their score model to a corpus of performance worms, aiming to reproduce the style of certain performers.

Rule-based systems use a matching process to map score features directly to performance modifications. Widmer [36] developed an inductive rule learning algorithm that automatically extracts performance rules from piano performances;

it discovered a small set of rules that cover a surprisingly large amount of expressivity in the data. Our YQX system uses some of these rules in combination with a probabilistic approach. Ramirez et al. [25] followed a similar approach using inductive logic programming to learn performance rules for Jazz saxophone from audio recordings. Perez et al. [24] used a similar technique on violin recordings. The well-known KTH rule system was first introduced in [28] and has been extended in more than 20 years of research. A comprehensive description is given in [6]. The *Director Musices* system is an implementation of the KTH system that allows for expressive performance rendering of musical scores. The rules in the system refer to low-level musical situations and theoretical concepts and relate them to predictions of timing, dynamics, and articulation.

The performance model In probabilistic approaches, is regarded as a multivariate probability distribution onto which the score model is mapped. The approaches differ in how the mapping is achieved. The Naist model [31] applies Gaussian processes to fit a parametric output function to the training performances. YQX [5] uses Bayesian network theory to model the interaction between score and performance. In addition, a small set of note-level rules adapted from Widmer's rule-based system are applied to further enhance musical quality. Grindlay and Helmbold first proposed a Hidden Markov Model (HMM) [10], which was later extended to form a Hierarchical HMM [9], the advantage of which is that phrase information is coded into the structure of the model. All approaches mentioned above learn a monophonic performance model, predict the melody voice of the piece and, in the rendering, synchronize the accompaniment according to the expressivity in the lead voice. Kim et al. [13] proposed a model of three sub-models: local expressivity models for the two outer voices (highest and lowest pitch of any given onset) and a harmony model for the inner voices.

Mazzola follows a different concept, building on a complex mathematical theory of musical structure [16]. The *Rubato* system [17, 19] is an implementation of this model.

3 The Data

Probabilistic models are usually learned from large sets of example data. For expressive performance modelling, the data must provide information on what was played (score information) and how it was played (performance information). The richness of available score information limits the level of sophistication of the score model: the more score information is provided, the more detailed a score model can be calculated. Piano performances can be described adequately by three dimensions: tempo, loudness, and articulation (our current model ignores pedalling). However, the information cannot be extracted from audio recordings at the necessary level of precision. This leaves a computer-controlled piano, such as the Bösendorfer CEUS (or the earlier version, the Bösendorfer SE) or a Yamaha Disklavier, as the only possible data source. This, of course, poses further problems. The number of recordings made on such devices is very small.

Since such instruments are not normally used in recording studios or in public performances, the majority of available recordings stem from a scientific environment and do not feature world-class artists.

For our experiments we use two unique data collections: The *Magaloff Corpus* and a collection of Mozart piano sonatas. In Spring 1989, Nikita Magaloff, a Russian-Georgian pianist famous for his Chopin interpretations, performed the entire work of Chopin for solo piano that was published during Chopin's lifetime (op. 1 - op. 64) in six public appearances at the Vienna Konzerthaus. Although the technology was fairly new at the time (first prototype in 1983, official release 1985 [20]), all six concerts were played and recorded on a Bösendorfer SE, precisely capturing every single keystroke and pedal movement. This was probably the first time the Bösendorfer SE was used to such an extent. The collected data is presumably the most comprehensive single-performer corpus ever recorded. The data set comprises more than 150 pieces with over 320,000 performed notes. We scanned the sheet music of all pieces and transformed it into machine-readable, symbolic scores in musicXML [27] format using the optical music recognition software SharpEye[1]. The MIDI data from the recordings were then matched semi-automatically to the symbolic scores. The result is a completely annotated corpus containing precisely measured performance data for almost all notes Chopin has ever written for solo piano[2]. Flossmann et. al [3] provided a comprehensive overview of the corpus, its construction, and results of initial exploratory studies of aspects of Magaloff's performance style.

The second data collection we use for the evaluation of our models are 13 complete Mozart piano sonatas played by the Viennese pianist R. Batik, likewise recorded on a Bösendorfer computer piano and matched to symbolic scores. This data set contains roughly 104,000 performed notes. Table 1 shows an overview of the two corpora.

Table 1. Overview of the data corpora.

	Magaloff Corpus	Mozart Corpus
Pieces/Movements	155	39
Score Notes	328,800	100,689
Performed Notes	335,542	104,497
Playing Time	10h 7m 52s	3h 57m 40s

4 Score and Performance Model

As indicated above, our rendering system is based on a score model comprising simple score descriptors (the *features*) and a musicological model – the

[1] see http://www.visiv.co.uk

[2] Some of the posthumously published works were played as encores but have not yet been included in the dataset

Implication-Realization model by Narmour. Performances are characterized in three dimensions: tempo, loudness, and articulation. The way the performance characteristics (the *targets*) are defined has a large impact on the quality of the rendered pieces.

The prediction is done note by note for the melody voice of the piece only. In the Mozart sonatas, we manually annotated the melody voice in all pieces. In the case of the Chopin data, we assume that the highest pitch at any given time is the melody voice of the piece. Clearly, this very simple heuristic does not always hold true, but, in the case of Chopin, it is correct often enough to be justifiable.

4.1 Performance targets

Tempo in musical performances usually refers to a combination of two aspects: (1) the current tempo of a performance that evolves slowly and changes according to ritardandi or accelerandi; (2) the tempo of individual notes, often referred to as *local timing*, i.e., local deviations from the current tempo, used to emphasize single notes through delay or anticipation. Tempo is often measured in absolute beats per minute. We define the tempo relative to inter-onset-intervals (IOI), i.e., the time between two successive notes. A performed IOI that is longer than prescribed by the score and the current tempo implies a slowing down, while a shorter IOI implies a speeding up. Thus, the description is independent of the absolute tempo and focuses on changes.

Loudness is not measured in absolute terms but relative to the overall loudness of the performance. Articulation describes the amount of legato between two successive notes: The smaller the audible gap between two successive notes, the more legato the first one becomes; the larger the gap, the more staccato.

Formally, we define the following performance targets:

Timing: The timing of a performance is measured in *inter-onset intervals* (IOIs), i.e., the time between two successive notes. The *IOI ratio* of a note relates the nominal score IOI and the performance IOI to the subsequent note. This indicates whether the next onset occurred earlier or later than prescribed by the score, and thus also whether the previous note was shortened or lengthened. Let s_i and s_{i+1} be two successive score notes, p_i and p_{i+1} the corresponding notes in the performance, $ioi^s_{i,i+1}$ the score IOI, $ioi^p_{i,i+1}$ the performance IOI of the two notes[3], l_s the duration of the complete piece in beats, and l_p the length of the performance. The IOI ratio $ioiR_i$ of s_i is then defined as:

$$ioiR_i = \log \frac{ioi^p_{i,i+1} * l_s}{ioi^s_{i,i+1} * l_p}.$$

Normalising both score and performance IOIs to fractions of the complete score and performance makes this measure independent of the actual tempo.

[3] The unit of the duration does not matter in this case, as it cancels out with the unit of the complete duration of the performance

The logarithm is used to scale the values to a range symmetrical around zero, where $ioiR_i > 0$ indicates a prolonged IOI, i.e., a tempo slower than notated, and $ioiR_i < 0$ indicates a shortened IOI, i.e., a tempo faster than notated.

Split tempo and timing: It can be beneficial to divide the combined tempo into current tempo and local timing. The current tempo is defined as the lower frequency components of the IOI ratio time series. A simple way of calculating the low-frequency component is to apply a windowed moving average function to the curve. The residual is considered the local timing. Let $ioiR_i$ be the IOI ratio of note s_i, and $n \in \mathbb{N}$ (usually $5 \leq n \leq 10$), then the current tempo ct_i of the note s_i is calculated by:

$$ct_i = \frac{1}{n} \sum_{j=i-\frac{(n-1)}{2}}^{i+\frac{(n-1)}{2}} ioiR_j.$$

The residual high-frequency content can be considered as the local timing lt_i and, in relation to the current tempo, indicates that a note is either played faster or slower with respect to the current tempo:

$$lt_i = \frac{ioiR_j - ct_i}{ct_i}.$$

Loudness: The loudness, also referred to as velocity[4], of a performance is described as the ratio between the loudness of a note and the mean loudness of the performance. Again, the logarithm is used to scale the values to a range symmetrical around zero, with values above 0 being louder than average and those below 0 softer than average. Let $mvel_i$ be the midi velocity of note s_i. The loudness vel_i is then calculated by:

$$vel_i = \log \frac{mvel_i}{\sum_j mvel_j}.$$

Articulation: Articulation measures the amount of legato between two notes, i.e., the ratio of the gap between them in a performance and the gap between them in the score. Let $ioi^s_{i,i+1}$ and $ioi^p_{i,i+1}$ be the score and performance IOIs between the successive notes s_i and s_{i+1}, and dur^s_i and dur^p_i the nominal score duration and the played duration of s_i respectively. The articulation art_i of a note s_i is defined as

$$art_i = \frac{ioi^s_{i,i+1} * dur^p_i}{dur^s_i * ioi^p_{i,i+1}}.$$

4.2 Score features

As briefly mentioned above, there are basically two ways of modelling a musical score: using (i) sophisticated musicological models, such as implementations

[4] Computer-controlled Pianos measure loudness by measuring the velocity at which a hammer strikes a string.

of the GTTM [15] or Narmour's Implication-Realization model [22], and (ii) feature-based descriptors of the musical content. We use a combination of both approaches.

Features fall into different categories, according to the musical content they describe, : rhythmic, melodic, and harmonic descriptors.

Rhythmic features describe the relations of score durations of successive notes and their rhythmic context. In our system, we use:

Duration Ratio: Let dur_i be the score duration of note s_i measured in beats; the duration ratio $durR_i$ is then defined by:

$$durR_i = \frac{dur_i}{dur_{i+1}}.$$

Rhythmic Context: The score durations of notes s_{i-1}, s_i, and s_{i+1} are sorted and assigned 3 different labels: short (s), neutral (n) and long (l). When a rest immediately before (and/or after) s_i is longer than half the duration of s_{i-1} (and/or s_{i+1}), the respective labels are replaced with (-). The rhythmic context $rhyC_i$ of s_i is then one of the 20 possible label triplets[5].

Melodic features describe the melodic content of the score, mainly pitch intervals and contours.

Pitch interval: The interval to the next score note, measured in semi-tones. The values are cut off at -13 and $+13$ so that all intervals greater than one octave are treated identically.

Pitch contour: The series of pitch intervals is smoothed to determine the distance of a score note to the next maximum or minimum pitch in the melody. The smoothing is needed to avoid choosing a local minimum/maximum.

IR features: One category of features is based on Narmour's Implication–Realization model of melodic expectation [22]. The theory constitutes an alternative to Schenkerian analysis and is focused more on cognitive aspects than on musical analysis. A short overview is given in section 4.3. We use the labels assigned to each melody note and the distance of a melody note to the nearest point of closure as score features.

Harmonic Consonance: Harmonic features describe perceptual aspects related to melodic consonance. Using Temperley's key profiles [30], we automatically determine the most likely local harmony given the pitches at a particular onset. The consonance of a note within an estimated harmony is judged using the key-profiles proposed by Krumhansl and Kessler [14].

[5] In the case of two equally long durations, we only discriminate between long and neutral. Hence, there are no situations labelled *lsl*, *sls*, *ssl*, etc., only *lnl*, *nln*, *nnl*, etc., which reduces the number of combinations used.

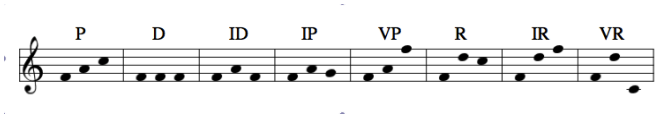

Fig. 1. Examples of eight IR-structures

4.3 Narmour's Implication-Realization (IR) model

The Implication- Realization (I-R) Model proposed by Narmour [22, 23] is a cognitively motivated model of musical structure. It tries to describe explicitly the patterns of listener expectation with respect to the continuation of the melody. It applies the principles of Gestalt theory to melody perception, an approach introduced by Meyer [18]. The model describes both the continuation implied by particular melodic intervals and the extent to which this (expected) continuation is actually realized by the following interval. Grachten [8] provides a short introduction to the processes involved.

Two main principles of the theory concern the direction and size of melodic intervals. (1) Small intervals imply an interval in the same registral direction, and large intervals imply a change in registral direction. (2) A small interval implies a following similarly-sized interval, and a large interval implies a smaller interval. Based on these two principles, melodic patterns, or *structures*, can be identified that either satisfy or violate the implications predicted by the principles. Figure 1 shows eight such structures: Process (P), Duplication(D), Intervallic Duplication (ID), Intervallic Process (IP), Registral Process (VP), Reversal (R), Intervallic Reversal (VR), and Registral Reversal (VR). The Process structure, for instance, sastisfies both registral and intervallic implications. Intervallic Process satisfies the intervallic difference principle, but violates the registral implication.

Another notion derived from the concept of implied expectations is *closure*, which refers to situations in which listeners might expect a caesura. In the IR model, closure can be evoked in several dimensions of the music: intervallic progression, metrical position, rhythm, and harmony. The accumulated degrees of closure in each dimension constitute the perceived overall closure at any point in the score. Occurrences of strong closure may coincide with a more commonly used concept of closure in music theory that refers to the completion of a musical entity, for example a phrase. Hence, calculating the distance of each note to the nearest point of closure can provide a segmentation of a piece similar to phrasal analysis.

5 YQX - The simple model

Our performance rendering system, called YQX, models the dependencies between score features and performance targets by means of a probabilistic network. The network consists of several interacting nodes representing different features and targets. Each node is associated with a probability distribution

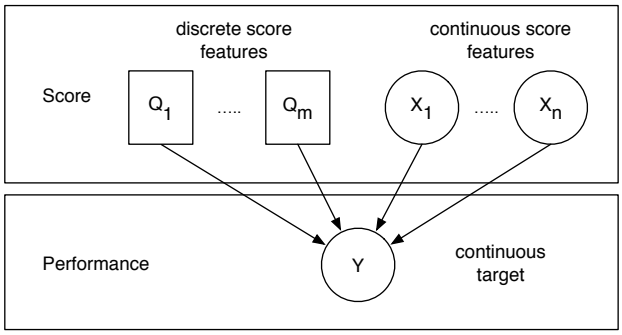

Fig. 2. The probabilistic network forming the YQX system

over the values of the corresponding feature or target. A connection between two nodes in the graph implies a conditioning of one feature or target distribution on the other. Discrete score features (the set of which we call $\mathbf{Q}$) are associated with discrete probability tables, while continuous score features ($\mathbf{X}$) are modelled by Gaussian distributions. The predicted performance characteristics, the targets ($\mathbf{Y}$), are continuously valued and conditioned on the set of discrete and continuous features. Figure 2 shows the general layout. The semantics is that of a linear Gaussian model [21]. This implies that the case of a continuous distribution parenting a continuous distribution is implemented by making the mean of the child distribution linearly dependant on the value of the condition. Sets are hereafter denoted by bold letters, and vectors are indicated by variables with superscribed arrows.

Mathematically speaking, a target y is modelled as a conditional distribution $P(y|\mathbf{Q}, \mathbf{X})$. Following the linear Gaussian model, this is a Gaussian distribution $\mathcal{N}(y; \mu, \sigma^2)$ with the mean μ varying linearly with $\mathbf{X}$. Given specific values $\mathbf{Q} = \mathbf{q}$ and $\mathbf{X} = \overrightarrow{x}$ (treating the real-valued set of continuous score features as a vector):

$$\mu = d_{\mathbf{q}} + \overrightarrow{k}_{\mathbf{q}} \cdot \overrightarrow{x},$$

where $d_{\mathbf{q}}$ and $\overrightarrow{k}_{\mathbf{q}}$ are estimated from the data by least squares linear regression. The average residual error of the regression is the variance σ^2 of the distribution. Thus, we collect all instances in the data that share the same combination of discrete feature values and build a joint probability distribution of the continuous features and targets of these instances. This implements the conditioning on the discrete features $\mathbf{Q}$. The linear dependency of the mean of the target distribution on the values of the continuous features introduces the conditioning on $\mathbf{X}$. This constitutes the training phase of the model.

Performance prediction is done note by note. The score features of a note are entered into the network as evidence $\overrightarrow{x}$ and $\mathbf{q}$. The instantiation of the discrete features determines the appropriate probability table and the parameterisation $d_{\mathbf{q}}$ and $\overrightarrow{k}_{\mathbf{q}}$, and the continuous features are used to calculate the mean of the target distribution μ. This value is used as the prediction for the specific note.

As the targets are independent, we create models and predictions for each target separately.

5.1 Quantitative Evaluation of YQX

We evaluated the model using the datasets described in section 3: the complete Chopin piano works played by N. Magaloff and 13 complete Mozart piano Sonatas played by R. Batik. The Mozart data were split into two different datasets – fast movements and slow movements – as they might reflect different interpretational concepts that would also be reproduced in the predictions. Thus, we also show the results for the Chopin data for different categories (ballades, nocturnes, etc.[6]). The quality of a predicted performance is measured by Pearson's correlation coefficient between the predicted curve and the curve calculated from the training data.

Table 2 shows the averaged results of threefold cross-validations over the datasets. For each result we chose the combination of score features with the best generalization on the test set. On the basis of predictions of local timing and current tempo, the complete IOI curve can be reassembled by reversing the splitting process described in 4.1. The column marked *ioi (r)* shows the best combined predictions for the each dataset.

The first observation is that the Chopin data generally show lower prediction quality, which implies that these data are much harder to predict than the Mozart pieces. This is probably due to the much higher variation in the performance characteristics for which the score features must account. Second, the loudness curves seem harder to predict than the tempo curves, a problem also observed in previous experiments with the model (see [5]) and [34]). Third, articulation seems to be easier to predict than tempo (with the exception of the slow Mozart movements and the Chopin scherzi, mazurkas, and waltzes, for which articulation was harder to predict than tempo). The Chopin categories show huge differences in the prediction quality for tempo (the scherzi being the hardest to predict and the waltzes the easiest), suggesting that there are indeed common interpretational characteristics within each category.

Predicting the IOI ratio by combining the predictions for local timing and current tempo seems moderately successful. Only in some cases is the best combined prediction better than the best prediction for the separate components. It must be noted though, that the combined predictions used the same set of features for both local timing and current tempo. Due to the extremely high number of possible combinations involved, experiments to find the two feature sets that lead to the best combined prediction have not yet been conducted.

[6] The category *Pieces* comprises: Rondos (op.1, op.5, op.16), Variations op.12, Bolero op.19, Impromptus (op.36, op.51), Tarantelle op.43, Allegro de Concert op. 46, Fantaisie op.49, Berceuse op.57, and Barcarolle op.61.

Table 2. Correlations between predicted and real performance for YQX. The targets shown are: IOI Ratio (*ioi*), loudness (*vel*), articulation (*art*), local timing (*timing*), current tempo (*tempo*), and reassembled IOI ratio (*ioi (r)*)

	ioi	vel	art	timing	tempo	ioi (r)
Mozart fast	0.46	0.42	0.49	0.43	0.39	0.46
Mozart slow	0.48	0.41	0.39	0.48	0.35	0.48
Chopin	0.22	0.16	0.33	0.15	0.18	0.22
Ballades	0.33	0.17	0.40	0.12	0.37	0.33
Etudes	0.17	0.15	0.17	0.09	0.20	0.16
Mazurkas	0.23	0.14	0.29	0.20	0.13	0.23
Nocturnes	0.17	0.17	0.33	0.14	0.11	0.17
Pieces	0.20	0.15	0.35	0.17	0.14	0.19
Polonaises	0.20	0.16	0.32	0.13	0.14	0.20
Preludes	0.20	0.15	0.33	0.15	0.16	0.21
Scherzi	0.33	0.23	0.26	0.16	0.30	0.33
Sonatas	0.16	0.14	0.32	0.12	0.20	0.16
Waltzes	0.35	0.16	0.29	0.22	0.35	0.35

5.2 Qualitative Evaluation of YQX

All quantitative evaluations of performances face the same problem: Although similarities between the predicted and the original curves can be measured to a certain degree, there is no computational way of judging the aesthetic qualities, or the degree of naturalness of expression, of a performance. The only adequate measure of quality is human judgement. The annual rendering contest RENCON [11] offers a scientific platform on which performance rendering systems can be compared and rated by the audience.

The system YQX participated in RENCON08, which was hosted alongside the ICMPC10 in Sapporo. Entrants to the "autonomous section" were required to render two previously unknown pieces (composed specifically for the competition) without any audio feedback from the system and within the time frame of one hour. Four contestants entered the autonomous section and competed for three awards: The Rencon award was to be given to a winner selected by audience vote (both through web and on-site voting), the Rencon technical award was to be given to the entrant judged most interesting from a technical point of view, and finally the Rencon Murao Award was to be given to the entrant that most impressed the composer Prof. T. Murao. YQX won all three prizes. While this is no proof of the absolute quality of the model, it does give some evidence that the model is able to capture and reproduce certain aesthetic qualities of music performance. A video of YQX performing at RENCON08 can be seen at `http://www.cp.jku.at/projects/yqx/yqx_cvideo2.flv`[7].

[7] The performed piece 'My Nocturne', a piano piece in a Chopin-like style, was composed by Prof. Tadahiro Murao specifically for the competition.

6 YQX - The enhanced dynamic model

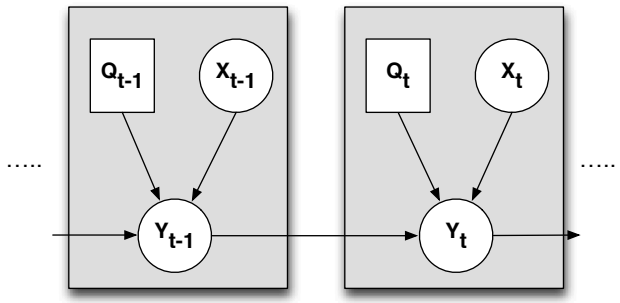

Fig. 3. The network unfolded in time

The predictions of the basic YQX system are note-wise; each prediction depends only on the score features at that particular score onset. In a real performance this is of course not the case: typically, changes in dynamics or tempo evolve gradually. Clearly, this necessitates awareness of the surrounding expressive context.

In this section we present two extensions to the system that both introduce a dynamic component by incorporating the prediction made for the preceding score note into the prediction of the current score note. Graphically, this corresponds to first unfolding the network in time and then adding an arc from the target in time-step $t-1$ to the target in time-step t. Figure 3 shows the unfolded network. This should lead to smoother and more consistent performances with less abrupt changes and, ideally, to an increase in the overall prediction quality.

The context-aware prediction can be done in two different ways: (1) Using the previous target simply as an additional parent probability distribution to the current target allows optimisation with respect to one preceding prediction. Minimal adaptation has to be made to the algorithm (see 6.1). (2) Using an adaptation of the Viterbi decoding in Hidden Markov Models results in a predicted series that is optimal with respect to the complete piece (see 6.2).

6.1 YQX with local maximisation

The first method is rather straightforward: We use the linear Gaussian model and treat the additional parent (the target y_{t-1}) to the target y_t as an additional feature that we calculate from the performance data. In the training process, the joint distribution of the continuous features, the target y_t, and the target in the previous time-step y_{t-1} given the discrete score features – in mathematical terms $P(y_{t-1}, y_t, \overrightarrow{x}_t|\mathbf{q}_t)$ – is estimated. This alters the conditional distribution of the

target y_t to $P(y_t|\mathbf{Q}, \mathbf{X}, y_{t-1}) = \mathcal{N}(y; \mu, \sigma^2)$ with[8]

$$\mu = d_{\mathbf{q},y_{t-1}} + \overrightarrow{k}_{\mathbf{q},y_{t-1}} \cdot (\overrightarrow{x}, y_{t-1}).$$

The prediction phase is equally straightforward. As in the simple model, the mean of $P(y_t|\mathbf{q}_t, \overrightarrow{x}_t, y_{t-1})$ is used as the prediction for the score note in time-step t. This is the value with the highest local probability.

6.2 YQX with global maximisation

The second approach drops the concept of a linear Gaussian model completely. In the training phase the joint conditional distributions $P(y_{t-1}, y_t, \overrightarrow{x}_t|\mathbf{q}_t)$ are estimated as before, but no linear regression parameters need to be calculated. The aim is to construct a sequence of predictions that maximises the conditional probability of the performance given the score features with respect to the complete history of predictions made up to that point.

This is calculated in similarly to the Viterbi-decoding in Hidden Markov Models, which tries to find the best explanation for the observed data [12]. Aside from the fact that the roles of evidence nodes and query nodes are switched, the main conceptual difference is that – unlike the HMM setup, which uses tabular distributions – our approach must deal with continuous distributions. This rules out the dynamic programming algorithm usually applied and calls for an analytical solution, which we present below. As in the Viterbi algorithm, the calculation is done in two steps: a forward and a backward sweep. In the forward movement the most probable target is calculated relative to the previous time-step. In the backward movement, knowing the final point of the optimal path, the sequence of predictions is found by backtracking through all time-steps.

The forward calculation Let $\overrightarrow{x}_t, \mathbf{q}_t$ be the sets of continuous and discrete features at time t, and N be the number of data points in a piece. Further, let α_t be the probability distribution over the target values y_t to conclude the optimal path from time-steps 1 to t. By means of a recursive formula, $\alpha(y_t)$ can be calculated for all time-steps of the unfolded network[9]:

$$\alpha(y_1) = p(y_1|\mathbf{x}_1, \mathbf{q}_1) \tag{1}$$

$$\alpha(y_t) = \max_{y_{t-1} \in \mathbb{R}} [p(y_t, y_{t-1}|\overrightarrow{x}_t, \mathbf{q}_t) \cdot \alpha(y_{t-1})] \tag{2}$$

This formula can be interpreted as follows: Assume that we know for all the target values y_{t-1} in time step $t-1$ the probability of being part of the optimal path. Then we can calculate for each target value y_t in time step t the predecessor that yields the highest probability for each specific y_t of being on the optimal path. In the backward movement we start with the most probable final point

[8] The construct $(\overrightarrow{x}, y_{t-1})$ is a concatenation of the vector $\overrightarrow{x}$ and the value y_{t-1} leading to a new vector of dimension $dim(\overrightarrow{x}) + 1$.

[9] We use $\alpha(y_t)$ and $p(y_t)$ as abbreviations of $\alpha(Y_t = y_t)$ and $p(Y_t = y_t)$, respectively.

of the path (the mean of the last α) and then backtrack to the beginning by choosing the best predecessors. As we cannot calculate the maximum over all $y_{t-1} \in \mathbb{R}$ directly, we need an analytical way of calculating $\alpha(y_t)$ from $\alpha(y_{t-1})$, which we derive below. We will also show that $\alpha(y_t)$ remains Gaussian through all time-steps. This is particularly important because we rely on the parametric representation using mean and variance.

We hereafter use the distribution $p(y_{t-1}|y_t, \overrightarrow{x}_t, \mathbf{q}_t) \propto \mathcal{N}(y_{t-1}; \mu_{t-1}, \sigma^2_{t-1})$ that can be calculated via conditioning from the joint conditional distribution $p(y_{t-1}, y_t, \overrightarrow{x}_t|\mathbf{q}_t)$ that is estimated in the training of the model. For details as to how this is done see, for instance, [26]. Anticipating our proof that the $\alpha(y_t)$ are Gaussian, we refer to the mean and variance as $\mu_{\alpha,t}$ and $\sigma^2_{\alpha,t}$.

The inductive definition of α (eq. 2) can be rewritten (the conditioning on $\mathbf{q}_t$, $\overrightarrow{x}_t$ is omitted for simplicity):

$$\alpha(y_t) = \max_{y_{t-1} \in \mathbb{R}} [p(y_{t-1}|y_t) \cdot \alpha(y_{t-1})] \cdot p(y_t) \tag{3}$$

Assuming that $\alpha(y_{t-1})$ is Gaussian, the result of the product in brackets is Gaussian $\mathcal{N}(y_{t-1}; \mu_*, \sigma^2_*)$ with a normalising constant z, that is Gaussian in both means of the factors:

$$\sigma^2_* = \frac{\sigma^2_{t-1} * \sigma^2_{\alpha,t-1}}{\sigma^2_{t-1} + \sigma_{\alpha,t-1}} \tag{4}$$

$$\mu_* = \sigma^2_* \left(\frac{\mu_{t-1}}{\sigma^2_{t-1}} + \frac{\mu_{\alpha,t-1}}{\sigma^2_{\alpha,t-1}} \right) \tag{5}$$

$$z = \frac{1}{\sqrt{2\pi|\sigma^2_{1-1} + \sigma^2_{\alpha,t-1}|}} e^{\left(\frac{-(\mu_{t-1} - \mu_{\alpha,t-1})^2}{2(\sigma^2_{t-1} + \sigma^2_{\alpha,t-1})} \right)} \tag{6}$$

Later, z is be multiplied with a Gaussian distribution over y_t. Hence, z must be transformed into a distribution over the same variable. By finding a y_t such that the exponent in eq. 6 equals 0 we can construct the mean μ_z and variance σ^2_z of z. Note that the variable μ_{t-1} is dependent on y_t due to the conditioning of $p(y_{t-1}|y_t)$ on y_t.

$$z \propto \mathcal{N}(y_t; \mu_z, \sigma^2_z) \tag{7}$$

$$\mu_z = -\frac{\sigma^2_t \cdot (\mu_{t-1} + \mu_{\alpha,t-1}) + \mu_t \cdot \sigma^2_{t,t-1}}{\sigma^2_{t,t-1}} \tag{8}$$

$$\sigma^2_z = \sigma^2_{t-1} + \sigma^2_{\alpha,t-1} \tag{9}$$

As z is independent of y_{t-1}, it is not affected by the calculation of the maximum:

$$\alpha(y_t) \propto \max_{y_{t-1} \in \mathbb{R}} [\mathcal{N}(y_{t-1}; \mu_*, \sigma^2_*)] \cdot \tag{10}$$

$$\mathcal{N}(y_t; \mu_z, \sigma^2_z) \cdot p(y_t)$$

$$= \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \mathcal{N}(y_t; \mu_z, \sigma^2_z) \cdot p(y_t) \tag{11}$$

The factor $\frac{1}{\sqrt{2\pi\sigma^2}}$ can be neglected, as it does not affect the parameters of the final distribution of $\alpha(y_t)$. The distribution $P(y_t)$ is Gaussian by design, and hence the remaining product again results in a Gaussian and a normalising constant. As the means of both factors are fixed, the normalising constant in this case is a single factor. The mean $\mu_{\alpha,t}$ and variance $\sigma^2_{\alpha,t}$ of $\alpha(y_t)$ follow:

$$\alpha(y_t) \propto \mathcal{N}(y_t; \mu_{\alpha,t}, \sigma^2_{\alpha,t}) \tag{12}$$

$$\sigma_{\alpha,t} = \frac{\sigma_t^2 \cdot \sigma_z^2}{\sigma_t^2 + \sigma_z^2} \tag{13}$$

$$\mu_{\alpha,t} = \sigma_{\alpha,t} \left(\frac{\mu_z}{\sigma_z^2} + \frac{\mu_t}{\sigma_t^2} \right). \tag{14}$$

Thus, $\alpha(y_t)$ is Gaussian in y_t, assuming that $\alpha(y_{t-1})$ is Gaussian. Since $\alpha(y_1)$ is Gaussian, it follows that $\alpha(y_t)$ is Gaussian for $1 \leq t \leq N$. This equation shows that the mean and variance of $\alpha(y_t)$ can be computed recursively using the mean $\mu_{\alpha,t-1}$ and variance $\sigma^2_{\alpha,t-1}$ of $\alpha(y_{t-1})$. The parameters of α_{y_1} equal μ_{y_1} and $\sigma^2_{y_1}$, which are the mean and the variance of the distribution $p(y_1|\overrightarrow{x}_1, \mathbf{q}_1)$, and are estimated from the data.

The backward calculation Once the mean and variance μ_t, σ_t^2 of $\alpha(y_t)$ are known for $1 \leq t \leq N$, the optimal sequence $y_1, ..., y_N$ can be calculated:

$$y_N = \mu_{\alpha,N} \tag{15}$$

$$y_{t-1} = \underset{y_{t-1}}{\operatorname{argmax}} \left[\mathcal{N}(y_{t-1}; \mu_*, \sigma_*^2 \right] \tag{16}$$

$$= \mu_* \tag{17}$$

6.3 Quantitative Evaluation

We evaluated the enhanced algorithms using the same datasets as for the original YQX model. As before, the correlation between predicted and human performance serves as a measure of quality. Table 3 shows the results. For comparison we also included the results for the original YQX model as presented in section 5.1.

For the Chopin data (both complete set and individual categories) the prediction quality for tempo increases in all cases and, for loudness in some cases. Prediction quality for articulation decreases compared to the original model for both local and global optimisation. This is not surprising, because articulation is a local phenomenon that does not benefit from long-term modelling. This also holds for the timing, i.e., the local tempo component: in most cases local or global optimisation does not improve the prediction quality. However, the current tempo – the low frequency component of the IOI ratio – on the other hand, does benefit from optimising the prediction globally with respect to the performance context: the prediction quality is increased in all cases (the biggest gain, almost 80%, is registered in the mazurkas).

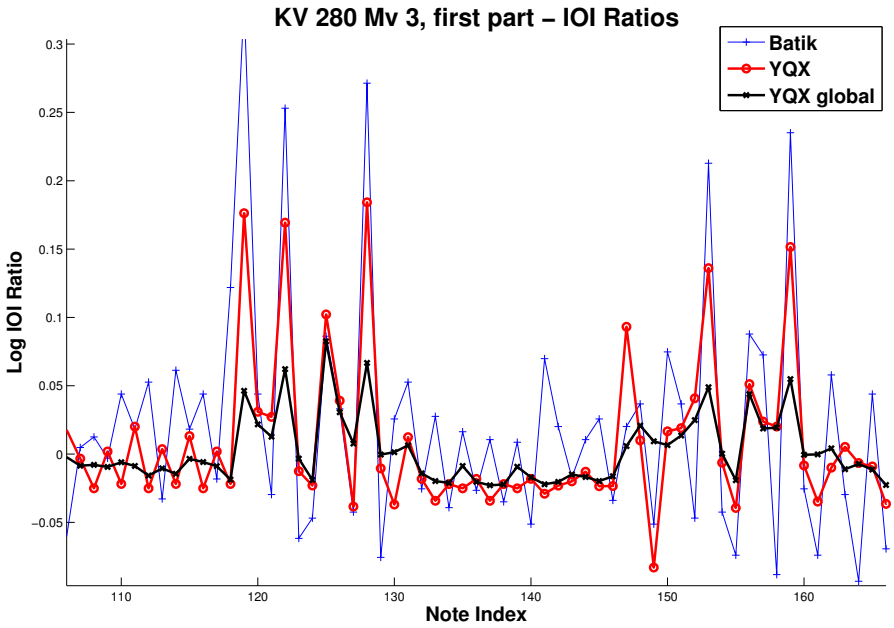

Fig. 4. IOI Ratios predicted for bars 31 - 54 of K. 280, Mv.3

Surprisingly, the Mozart data paint a different picture: None of the performance targets (with the exception of the current tempo prediction for the fast movements) benefits from including the performance context into the predictions. Previous experiments [5] showed that, given a specific, fixed set of features, local or global optimisation improves the prediction quality. However, given the freedom of choosing the best set of features for each particular target (which is the evaluation setup we chose here), feature sets exist with which the original, simple model outperforms the enhanced versions in terms of average correlation.

6.4 Qualitative Evaluation

Figure 4 shows the IOI ratio predictions for bars 31 to 54 in the third movement of Mozart Sonata K. 280. The original YQX algorithm exhibits small fluctuations that are largely uncorrelated with the human performance. This results in small but noticeable irregularities in the rendered performance. In contrast to the human performance, which is far from yielding a flat curve, these make the result sound inconsistent instead of lively and natural. The globally optimized YQX eliminates them at the expense of flattening out some of the (musically meaningful) spikes. The correlation for the movement was improved by 57.2% from 0.29 (YQX) to 0.456 (YQX global).

Table 3. Correlations between predicted and real performance for the basic YQX and the locally and globally optimized models. The targets shown are: IOI Ratio (*ioi*), loudness (*vel*), articulation (*art*), local timing (*timing*), current tempo (*tempo*), and reassembled IOI ratio (*ioi (r)*).

		ioi	vel	art	timing	tempo	ioi (r)
Mozart fast	yqx	0.46	0.42	0.49	0.43	0.39	0.46
	local	0.44	0.41	0.48	0.42	0.43	0.44
	global	0.39	0.37	0.37	0.32	0.43	0.39
Mozart slow	yqx	0.48	0.41	0.39	0.48	0.35	0.48
	local	0.46	0.39	0.38	0.48	0.42	0.47
	global	0.46	0.35	0.23	0.44	0.34	0.46
Chopin	yqx	0.22	0.16	0.33	0.15	0.18	0.22
	local	0.21	0.14	0.14	0.15	0.16	0.20
	global	0.23	0.15	0.14	0.16	0.22	0.23
Ballades	yqx	0.33	0.17	0.40	0.12	0.37	0.33
	local	0.36	0.17	0.39	0.12	0.30	0.25
	global	0.38	0.19	0.36	0.12	0.46	0.38
Etudes	yqx	0.17	0.15	0.17	0.09	0.20	0.16
	local	0.14	0.14	0.16	0.09	0.17	0.14
	global	0.22	0.15	0.15	0.13	0.26	0.23
Mazurkas	yqx	0.23	0.14	0.29	0.20	0.13	0.23
	local	0.22	0.14	0.28	0.22	0.13	0.21
	global	0.23	0.13	0.27	0.20	0.19	0.24
Nocturnes	yqx	0.17	0.17	0.33	0.14	0.11	0.17
	local	0.17	0.11	0.32	0.14	0.17	0.16
	global	0.20	0.18	0.31	0.15	0.14	0.18
Pieces	yqx	0.20	0.15	0.35	0.17	0.14	0.19
	local	0.22	0.12	0.33	0.12	0.16	0.18
	global	0.23	0.14	0.33	0.17	0.25	0.26
Polonaises	yqx	0.20	0.16	0.32	0.13	0.14	0.20
	local	0.18	0.19	0.32	0.13	0.15	0.16
	global	0.22	0.19	0.31	0.14	0.20	0.23
Preludes	yqx	0.20	0.15	0.33	0.15	0.16	0.21
	local	0.19	0.11	0.31	0.15	0.22	0.18
	global	0.22	0.14	0.28	0.14	0.23	0.22
Scherzi	yqx	0.33	0.23	0.26	0.16	0.30	0.33
	local	0.34	0.18	0.26	0.15	0.32	0.31
	global	0.34	0.18	0.25	0.13	0.36	0.34
Sonatas	yqx	0.16	0.14	0.32	0.12	0.20	0.16
	local	0.17	0.12	0.32	0.12	0.18	0.15
	global	0.21	0.15	0.32	0.09	0.28	0.22
Waltzes	yqx	0.35	0.16	0.29	0.22	0.35	0.35
	local	0.37	0.18	0.28	0.23	0.31	0.14
	global	0.38	0.24	0.29	0.22	0.44	0.38

7 Further extensions

7.1 Note-level Rules

In 2003, Widmer developed a rule extraction algorithm for musical expression [36]. Applied to the Mozart sonatas, this resulted in a number of simple rules suggesting expressive change under certain melodic or rhythmic circumstances. Some of them were used with surprising consistency [35]. We use two of the rules to further enhance the aesthetic qualities of the rendered preformances:

Staccato Rule: If two successive notes (not exceeding a certain duration) have the same pitch, and the second of the two is longer, then the first note is played staccato. In our implementation the predicted articulation is substituted with a fixed small value, usually around 0.15, which amounts to 15% of the duration in the score in terms of the current performance tempo.

Delay Next Rule: If two notes of the same length are followed by a longer note, the last note is played with a slight delay. The IOI ratio of the middle note of a triplet satisfying the condition is calculated by taken the average of the two preceding notes and adding a fixed amount.

Figure 5 shows the effect of the Delay Next rule on the IOI ratios predicted for Chopin Prelude op.28 No.18, bars 12 - 17. Instances of the Delay Next rule occur at beats $24, 24.5, 26.5, 28.5$, and 29.5, all of which coincide with local delays in Magaloff's performance.

7.2 Combined Tempo and Timing Model

As discussed briefly in section 5.1, it seems reasonable to split the tempo curve into a high- and a low-frequency component (the local and global tempo) predict the two separately, and reassemble a tempo prediction from the two curves. Considering the effect of global optimisation, as discussed in section 6.3, it also seems appropriate to use the basic model for the local timing predictions and the global optimisation algorithm for the current tempo predictions.

An obvious extension to the experiments already presented would be to use different feature sets for the two components. In previous studies [4] we have discovered a relation between the size of the context a feature describes and its prediction quality for global and local tempo changes. The low-frequency components of certain features that are calculated, for instance, via a windowed moving average, are more suitable for global tempo prediction than are the high-frequency components, and vice versa for local tempo changes. Preliminary experiments that integrate this concept in the YQX algorithms show a slight quality increase (around 5%) for the current tempo and, consequently, for the combined IOI ratio target.

Also, global tempo trends in classical music are highly correlated with the phrase structure of a piece. This fact is often discussed in research on models of expressivity, such as the kinematic models introduced by Friberg and Sundberg [7] and by Todd [33]. Instead of training a model on the tempo curve of a complete

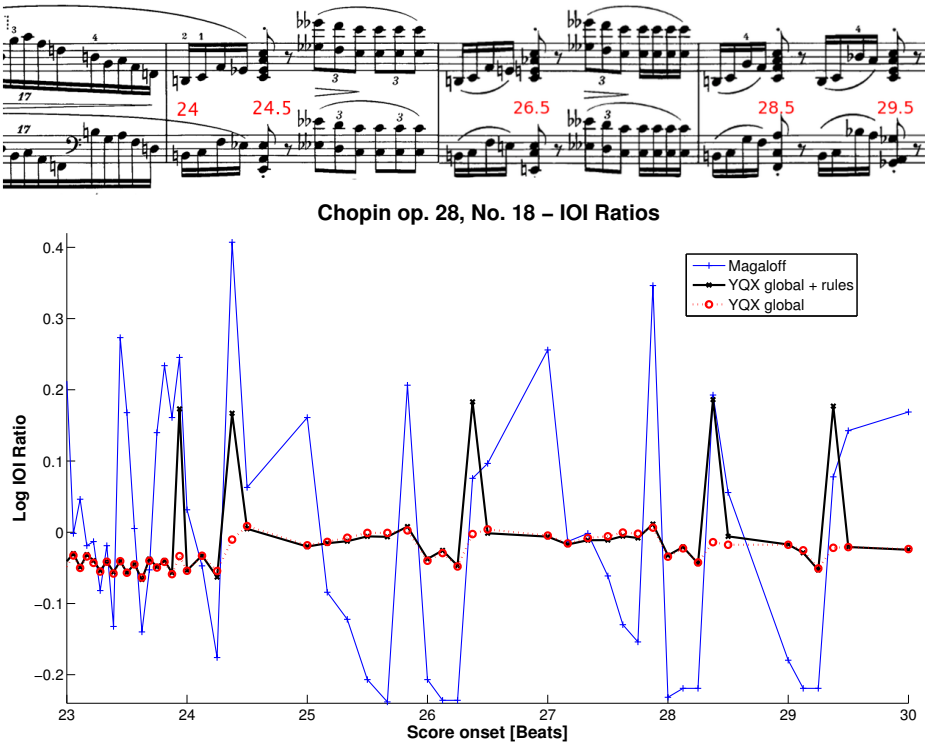

Fig. 5. Upper panel: Score of Chopin Prelude op.28 No.18, bars 13-15, the onsets in questions are marked; Lower panel: Effect of the Delay Next rule applied to the YQX prediction for Chopin Prelude op.28 No. 18, beats 22-32.

piece, a promising approach would thus be to train and predict phrases or phrase-like segments of the score. A possible, albeit simplistic, implementation would assume that tempo and loudness follow a n approximately parabolic trend – soft and slow at the beginning and end of a phrase, faster and louder in the middle. A performance would then be created by combining the local tempo predictions made by a probabilistic model with a segment-wise parabolic global tempo. To refine the segment-wise predictions of global tempo, any kind of more sophisticated model could be used – a probabilistic system, a parametric model or a case-based one (as in [34]).

7.3 Dynamic Bayesian Networks

Both models presented above, the Bayesian reasoning of YQX and the context-aware dynamic YQX, are subclasses of the general complex of Bayesian networks. The obvious generalization of the models is towards a Dynamic Bayesian Network (DBN). The main differences lie in (1) the network layout and (2) the way the model is trained.

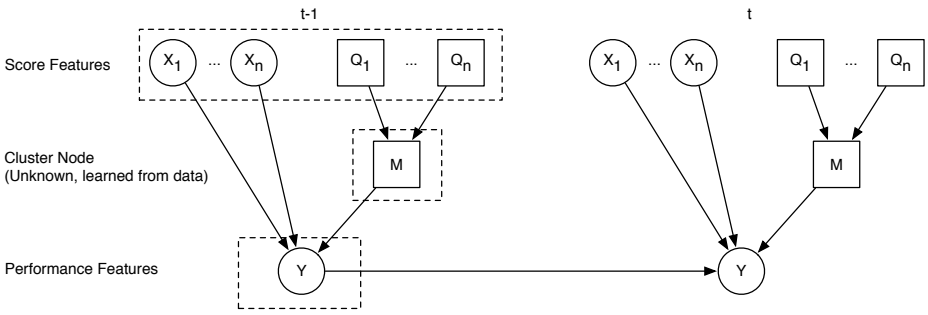

Fig. 6. Possible extension of YQX to a Dynamic Bayesian Network

We restricted our system to connections from score features to the performance targets within one timestep. For the training of the model, all features and targets had to be known in advance. Figure 6 shows what a DBN could look like for expressive performance rendering. The basic idea is the same: the performance targets are statistically dependent on the score features and the previously predicted target value. In addition, an intermediate layer (the discrete node M in figure 6) can be added that does not represent any particular score characteristic but instead functions as a clustering element for the discrete score features $Q_1, \ldots, Q_n$. This mitigates the sparsity problem caused by the huge number of possible combinations of values for the discrete features. The values of M are not known in advance, only the number of discrete states that the node can be in is fixed. The conditional probability distribution of M given the parenting nodes $Q_1, \ldots, Q_n$ is estimated in the training process of the model. The training itself is done by maximising the log-likelihood of the predicted values with an expectation-maximisation algorithm [21].

However, the most significant difference is that, instead of feeding the complete piece into the model at once, DBNs work on short segments. In theory, any trend common to all or most of the segments should also be recognizable in the predicted curves. Given a segmentation of musical works into musically meaningful fragments – ideally phrases – the network should be able to reproduce patterns of tempo or loudness that are common across phrases.

8 Conclusion

Automatic synthesis of expressive music is a very challenging task. Of particular difficulty is the evaluation of a system, as one cannot judge the aesthetic quality of a performance by numbers. The only adequate measure of quality is human judgement. The rendering system presented, passed this test in the RENCON08 and therefore constitutes a baseline for our current research. The two extensions we devised incorporate the current performance context into predictions. This proved useful for reproducing longer term trends in the data at the expense of local expressivity.

We consider this a work in progress. There is still a long way to go to a machine-generated performance that sounds profoundly musical. The main goal in the near future will be to further develop the idea of a multilevel system comprising several sub-models, each specialised on a different aspect of performance – global trends and local events. Segmentation of the input pieces will also play a significant role, as this reflects the inherently hierarchical structure of music performance.

9 Acknowledgements

We express our gratitude to Mme Irène Magaloff for her generous permission to use the unique resource that is the Magaloff Corpus for our research. This work is funded by the Austrian National Research Fund FWF via grants TRP 109-N23 and Z159 ("Wittgenstein Award"). The Austrian Research Institute for Artificial Intelligence acknowledges financial support from the Austrian Federal Ministries BMWF and BMVIT.

10 Questions

1. Aside from the central problem of mapping the score to the performance, what are the other main challenges in the process of generating a computer performance?
2. Why is evaluating automatically by measuring the similarity between rendered and real performances of a piece problematic?
3. What are the two methods on which score models (i.e., representations of the music and its structure) may be based?
4. What three different categories can be distinguished regarding the learning and prediction models used in CSEMPs?
5. In probabilistic approaches how is the performance model regarded?
6. For data used in developing an expressive performance statistical model, the data must provide information on what two elements?
7. What musicological model was selected for the YQX system?
8. In what three dimensions are performances characterized in YQX?
9. What is the difference in implementation between the local and the global maximization approaches in YQX?
10. What is the difference in results between the local and the global maximization approaches in YQX?

References

1. Arcos, J., de Mántaras, R.: An interactive CBR approach for generating expressive music. Journal of Applied Intelligence 27(1), 115–129 (2001)

2. Dorard, L., Hardoon, D., Shawe-Taylor, J.: Can style be learned? A machine learning approach towards 'performing' as famous pianists. In: Proceedings of Music, Brain & Cognition Workshop - The Neural Information Processing Systems 2007 (NIPS 2007). Whistler, Canada (2007)
3. 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)
4. Flossmann, S., Grachten, M., Widmer, G.: Experimentally investigating the use of score features for computational models of expressive timing. In: Proceedings of the 10th International Conference on Music Perception and Cognition 2008 (ICMPC '08). Sapporo, Japan (2008)
5. Flossmann, S., Grachten, M., Widmer, G.: Expressive performance rendering: Introducing performance context. In: Proceedings of the 6th Sound and Music Computing Conference 2009 (SMC '09). pp. 155–160. Porto, Portugal (2009)
6. 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)
7. Friberg, A., Sundberg, J.: Does music performance allude to locomotion? A model of final ritardandi derived from measurements of stopping runners. Journal of the Acoustical Society of America 105(3), 1469–1484 (1999)
8. Grachten, M.: Expressivity-Aware Tempo Transformations of Music Performances Using Case Based Reasoning. Ph.D. thesis, Pompeu Fabra University, Barcelona (2006)
9. Grindlay, G., Helmbold, D.: Modeling, analyzing, and synthesizing expressive piano performance with graphical models. Machine Learning 65(2-3), 361–387 (2006)
10. Grindlay, G.C.: Modeling Expressive Musical Performance with Hidden Markov Models. Master's thesis, University of California, Santa Cruz (2005)
11. Hashida, M.: RENCON - Performance Rendering Contest for computer systems. http://www.renconmusic.org/ (September 2008), `http://www.renconmusic.org/`
12. Juang, B.H., Rabiner, L.R.: Hidden Markov Models for speech recognition. Technometrics 33(3), 251–272 (1991)
13. Kim, T.H., Fukayama, S., Nishimoto, T., Sagayama, S.: Performance rendering for polyphonic piano music with a combination of probabilistic models for melody and harmony. In: Proceedings of the 7th Sound and Music Computing Conference 2010 (SMC '10). Barcelona, Spain (2010)
14. Krumhansl, C.L., Kessler, E.J.: Tracing the dynamic changes in perceived tonal organization in a spatioal representation of musical keys. Psychological Review 89, 334–368 (1982)
15. Lerdahl, F., Jackendoff, R.: A Generative Theory of Tonal Music. The MIT Press, Cambridge (1983)
16. Mazzola, G.: The Topos of Music - Geometric Logic of Concepts, Theory, and Performance. Birkhäuser Verlag, Basel (2002)
17. Mazzola, G.: Rubato software. http://www.rubato.org (2006)

18. Meyer, L.: Emotion and meaning in Music. University of Chicago Press, Chicago (1956)
19. Milmeister, G.: The Rubato Composer Music Software: Component-Based Implementation of a Functorial Concept Architecture. Ph.D. thesis, Universität Zürich, Zürich (2006)
20. Moog, R.A., Rhea, T.L.: Evolution of the keyboard interface: The Boesendorfer 290 SE recording piano and the moog multiply-touch-sensitive keyboards. Computer Music Journal 14(2), 52–60 (1990)
21. Murphy, K.: Dynamic Bayesian Networks: Presentation, Inference and Learning. Ph.D. thesis, University of California, Berkeley (2002)
22. Narmour, E.: The Analysis and Cognition of Basic Melodic Structures: The Implication-Realization Model. University of Chicago Press, Chicago (1990)
23. Narmour, E.: The Analysis and Cognition of Melodic Complexity: The Implication-Realization Model. University of Chicago Press, Chicago (1992)
24. Perez, A., Maestre, E., Ramirez, R., Kersten, S.: Expressive irish fiddle performance model informed with bowing. In: Proceedings of the International Computer Music Conference 2008 (ICMC '08). Belfast, Northern Ireland (2008)
25. Ramirez, R., Hazan, A., Gòmez, E., Maestre, E.: Understanding expressive transformations in saxophone Jazz performances using inductive machine learning in saxophone jazz performances using inductive machine learning. In: Proceedings of the Sound and Music Computing International Conference 2004 (SMC '04). Paris, France (2004)
26. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. The MIT Press (2006), `www.GaussianProcess.org/gpml`
27. Recordare: MusicXML definition. http://www.recordare.com/xml.html (2003), `http://www.recordare.com/xml.html`
28. Sundberg, J., Askenfelt, A., Frydén, L.: Musical performance. A synthesis-by-rule approach. Computer Music Journal 7, 37–43 (1983)
29. Suzuki, T.: The second phase development of case based performance rendering system "Kagurame". In: Working Notes of the IJCAI-03 Rencon Workshop. pp. 23–31. Acapulco, Mexico (2003)
30. Temperley, D.: Music And Probability. MIT Press, Cambridge, MA, USA (2007)
31. Teramura, K., Okuma, H., et al.: Gaussian process regression for rendering music performance. In: Proceedings of the 10th International Conference on Music Perception and Cognition 2008 (ICMPC '08). Sapporo, Japan (2008)
32. Tobudic, A., Widmer, G.: Relational IBL in classical music. Mach. Learn. 64(1-3), 5–24 (2006)
33. Todd, N.P.M.: The dynamics of dynamics: A model of musical expression. Journal of the Acoustical Society of America 91, 3450–3550 (1992)
34. Widmer, G., Tobudic, A.: Playing Mozart by analogy: Learning multi-level timing and dynamics strategies. Journal of New Music Research 32(3), 259–268 (2003)
35. Widmer, G.: Machine discoveries: A few simple, robust local expression principles. Journal of New Music Research 31(1), 37–50 (2002)

36. Widmer, G.: Discovering simple rules in complex data: A meta-learning algorithm and some surprising musical discoveries. Artificial Intelligence 146(2), 129–148 (2003)
37. Widmer, G., Flossmann, S., Grachten, M.: YQX plays Chopin. AI Magazine 30(3), 35–48 (2009)
38. 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)