Playing Mozart Phrase by Phrase

Asmir Tobudic[1] and Gerhard Widmer[1,2]

[1] Austrian Research Institute for Artificial Intelligence, Vienna,
[2] Department of Medical Cybernetics and Artificial Intelligence,
University of Vienna, Austria

Abstract. The article presents an application of instance-based learning to the problem of expressive music performance. A system is described that tries to learn to shape tempo and dynamics of a musical performance by analogy to timing and dynamics patterns found in performances by a concert pianist. The learning algorithm itself is a straightforward k-nearest-neighbour algorithm. The interesting aspects of this work are application-specific: we show how a complex, multi-level artifact like the tempo/dynamics variations applied by a musician can be decomposed into well-defined training examples for a learner, and that case-based learning is indeed a sensible strategy in an artistic domain like music performance. While the results of a first quantitative experiment turn out to be rather disappointing, we will show various ways in which the results can be improved, finally resulting in a system that won a prize in a recent 'computer music performance' contest.

1 Introduction

The work described in this paper is another step in a long-term research endeavour that aims at building quantitative models of expressive music performance via AI and, in particular, machine learning methods [9, 10]. This is basic research. We do not intend to engineer computer programs that generate music performances that sound as human-like as possible. Rather, the goal is to investigate to what extent a machine can automatically build, via inductive learning from 'real-world' data (i.e., real performances by highly skilled musicians), operational models of certain aspects of performance, for instance, predictive models of tempo, timing, or dynamics. In this way we hope to get new insights into fundamental principles underlying this complex artistic activity, and thus contribute to the growing body of knowledge in the area of empirical musicology (see [4] for an excellent overview).

In previous work, we managed to show that a computer can indeed find interesting regularities of musical performance. A new rule learning algorithm [12] succeeded in discovering a small set of simple, robust, and highly general rules that predict a substantial part of the note-level expressive choices of a performer (e.g., whether she will shorten or lengthen a particular note) with surprisingly high precision [11]. But these rules described only very local, low-level aspects (things a performer does to a particular note), and indeed, the

'expressive' performances produced by the computer on the basis of the learned rules were far from sounding musical.

Music performance is a highly complex activity, with performers tending to shape the music at many different levels simultaneously (see below). The goal of our current work is to complement the note-level rule model with a predictive model of musical expression at higher levels of the musical structure, e.g., the level of *phrases*. This paper presents our first steps in this direction. An instance-based learning system is described that recognizes performance patterns at various abstraction levels and learns to apply them to new pieces (phrases) by analogy to known performances. The learning algorithm itself is a straightforward k-nearest neighbour algorithm. The interesting aspects of this work are thus not so much machine-learning-specific but application-specific: we show how a complex artistic artifact like the tempo/dynamics variations applied by a musician can be decomposed into well-defined training examples for a learner, and that case-based prediction is indeed a sensible strategy in an artistic domain like music performance. While the results of a first quantitative experiment turn out to be rather disappointing, we will show various ways in which the results can be improved, finally resulting in a system that — while still far from being able to attain the musical quality of human musicians — won a prize in a recent 'computer music performance' contest.

The paper is organized as follows: Section 2 briefly introduces the reader to the notion of expressive music performance and its representation via performance curves. Section 3 then describes how the training examples for the learner are derived (by decomposing complex performance curves into elementary 'expressive shapes' that can be associated with musical phrases at different levels), and specifies our learning algorithm. Section 4 presents first results of systematic experiments. Various ways of improving these are shown in Section 5, and Section 6 briefly talks about the qualititative, musical side of the results, including the above-mentioned computer music performance contest. Current and future research plans are then discussed in the final Section 7.

2 Expressive music performance and performance curves

Expressive music performance is the art of shaping a musical piece by continuously varying important parameters like tempo, dynamics, etc. Human musicians do not play a piece of music mechanically, with constant tempo or loudness, exactly as written in the printed music score. Rather, they speed up at some places, slow down at others, stress certain notes or passages by various means, and so on. The most important parameter dimensions available to a performer (a pianist, in particular) are tempo and continuous tempo changes, dynamics (loudness variations), and articulation (the way successive notes are connected). Most of this is not specified in the written score, but at the same time it is absolutely essential for the music to be effective and engaging. As such, expressive performance is a phenomenon of central interest in contemporary (cognitively oriented) musicology.

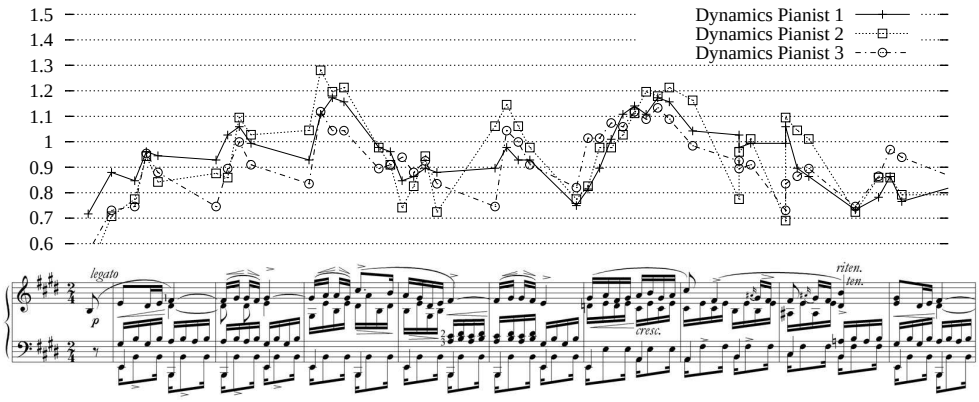

Fig. 1. Dynamics curves (relating to melody notes) of performances of the same piece (Frédéric Chopin, Etude op.10 no.3, E major) by three different Viennese pianists (computed from recordings on a Bösendorfer 290SE computer-monitored grand piano).

In the following, we will restrict ourselves to two of the most important parametric dimensions: *timing* (tempo variations) and *dynamics* (loudness variations). The tempo and loudness variations applied by a musician over the course of a piece (if we can measure them, which is a problem in its own right) can be represented as *tempo* and *loudness curves*, respectively. For instance, Figure 1 shows *dynamics curves* — the dynamics patterns produced by three different pianists in performing the same piece. Each point represents the relative loudness with which a particular melody note was played (relative to an averaged 'standard' loudness); a purely mechanical, unexpressive rendition of the piece would correspond to a perfectly flat horizontal line at $y = 1.0$. Variations in tempo can be represented in an analogous way.

Musically trained readers will notice certain high-level patterns or trends in the curves in Figure 1 that seem to correlate with lower- and higher-level phrases of the piece (e.g., a global up-down, *crescendo-decrescendo* tendency over the large phrase that covers the first four bars, and a consistent patterning of the one-bar subphrases contained in it). Extracting and learning to apply such high-level expressive patterns is the goal of the work presented here.

3 Learning Task and Algorithm

3.1 Deriving the training instances: Multilevel decomposition of performance curves

Our starting material is the scores of musical pieces plus measurements of the tempo and dynamics variations applied by a pianist in actual performances of these pieces, represented as *tempo* and *dynamics curves*. Both tempo and loudness are represented as multiplicative factors, relative to the average tempo and

dynamics of the piece. For instance, a tempo value of 1.5 for a note means that the note was played 1.5 times as fast as the average tempo of the piece, and a loudness of 1.5 means that the note was played 50% louder than the average loudness of all melody notes. In addition, the system is given information about the *hierarchical phrase structure* of the pieces, currently at four levels of phrasing. Phrase structure analysis is currently done by hand, as no reliable algorithms are available for this task.

Given a performance (dynamics or tempo) curve, the first problem is to define and extract the *training examples* for phrase-level learning. Remember that we want to learn how a performer 'shapes' phrases at different structural levels by tempo and dynamics 'gestures'. To that end, the complex curve must be decomposed into basic expressive 'gestures' or 'shapes' that represent the most likely contribution of each phrase to the overall observed performance curve.

As approximation functions to represent these shapes we decided to use the class of second-degree polynomials (functions of the form $y = ax^2 + bx + c$), because there is quite a consensus in musicology that high-level tempo and dynamics are well characterized by quadratic or parabolic functions [5, 7, 8] (but see section 5.4 below). Decomposing a given performance curve is an iterative process, where each step deals with a specific level of the phrase structure: for each phrase at a given level, we compute the polynomial that best fits the part of the curve that corresponds to this phrase, and 'subtract' the tempo or dynamics deviations 'explained' by the approximation. The curve that remains after this 'subtraction' is then used in the next level of the process. We start with the highest given level of phrasing and move to the lowest.

As by our definitions, tempo and dynamics curves are lists of multiplicative factors, 'subtracting' the effects predicted by a fitted curve from an existing curve simply means dividing the y values on the curve by the respective values of the approximation curve.

More formally, let $N_p = \{n_1, ..., n_k\}$ be the sequence of melody notes spanned by a phrase p, $O_p = \{onset_p(n_i) : n_i \in N_p\}$ the set (sequence) of relative note positions of these notes within phrase p (on a normalized scale from 0 to 1), and $E_p = \{expr(n_i) : n_i \in N_p\}$ the part of the performance curve (i.e., tempo or dynamics values) associated with these notes. Fitting a second-order polynomial onto E_p then means finding a function $f_p(x) = a^2x + bx + c$ that minimizes

$$D(f_p(x), N_p) = \sum_{n_i \in N_p} [f_p(onset_p(n_i)) - expr(n_i)]^2$$

Given an performance curve $E_p = \{expr(n_1), ..., expr(n_k)\}$ over a phrase p, and an approximation polynomial $f_p(x)$, 'subtracting' the shape predicted by $f_p(x)$ from E_p then means computing the new curve

$$E'_p = \{expr(n_i)/f_p(onset_p(n_i)) : i = 1...k\}.$$

The final curve we obtain after the fitted polynomials at all phrase levels have been 'subtracted' is called the *residual* of the performance curve [13].

To illustrate, Figure 2 shows the dynamics curve of the last part (mm.31–38) of the Mozart Piano Sonata K.279 (C major), first movement, first section. The four-level phrase structure our music analyst assigned to the piece is indicated by the four levels of brackets at the bottom of each plot. The figure shows the stepwise approximation of the performance curve by polynomials at three of the four phrase levels, as well as how much of the original curve is accounted for by the four levels of approximations, and what is left unexplained (the *residuals*).

3.2 Learning and prediction

Given performance curves decomposed into levels of phrasal shapes, the learning task is to predict appropriate tempo or dynamics shapes for new musical phrases (at any level) on the basis of examples of known phrases with associated shapes. More precisely, what is to be predicted for each example are three coefficients a, b, c that define an approximation polynomial $y = ax^2 + bx + c$. The learning algorithm is a simple nearest-neighbour algorithm [2]; we first decided to use only the one nearest neighbour for prediction, because it is not entirely clear how several predictions (triples of coefficients) should be combined in a sensible way.

The similarity between phrases is computed as the inverse of the standard Euclidean distance. For the moment, phrases are represented simply as fixed-length vectors of attribute values, where the attributes describe very basic phrase properties like the length of a phrase, melodic intervals between the starting and ending notes of the melody, information about where the highest melodic point (the 'apex') of the phrase is, the harmonic progression between start, apex, and end, whether the phrase ends with a cadential chord sequence, etc. Given such a fixed-length representation, the definition of the Euclidean distance is trivial.

At prediction time, the shapes predicted by the learner for nested phrases at different levels must be combined into a final compositive performance curve that is then evaluated (and can be used to produce a computer-generated 'expressive' performance). This is simply the inverse of the curve decomposition problem. Given a new piece to produce a performance for, the system starts with an initial 'flat' performance curve (a list of 1.0 values) and then successively multiplies the current value by the phrase-level predictions.

Formally, for a given note n_i that is contained in m hierarchically nested phrases $p_j, j = 1...m$, the expression (tempo or dynamics) value $expr(n_i)$ to be applied to it is computed as

$$expr(n_i) = \prod_{j=1}^{m} f_{p_j}(onset_{p_j}(n_i)),$$

where f_{p_j} is the approximation polynomial predicted as being best suited for the j^{th}-level phrase p_j by the nearest-neighbour learning algorithm.

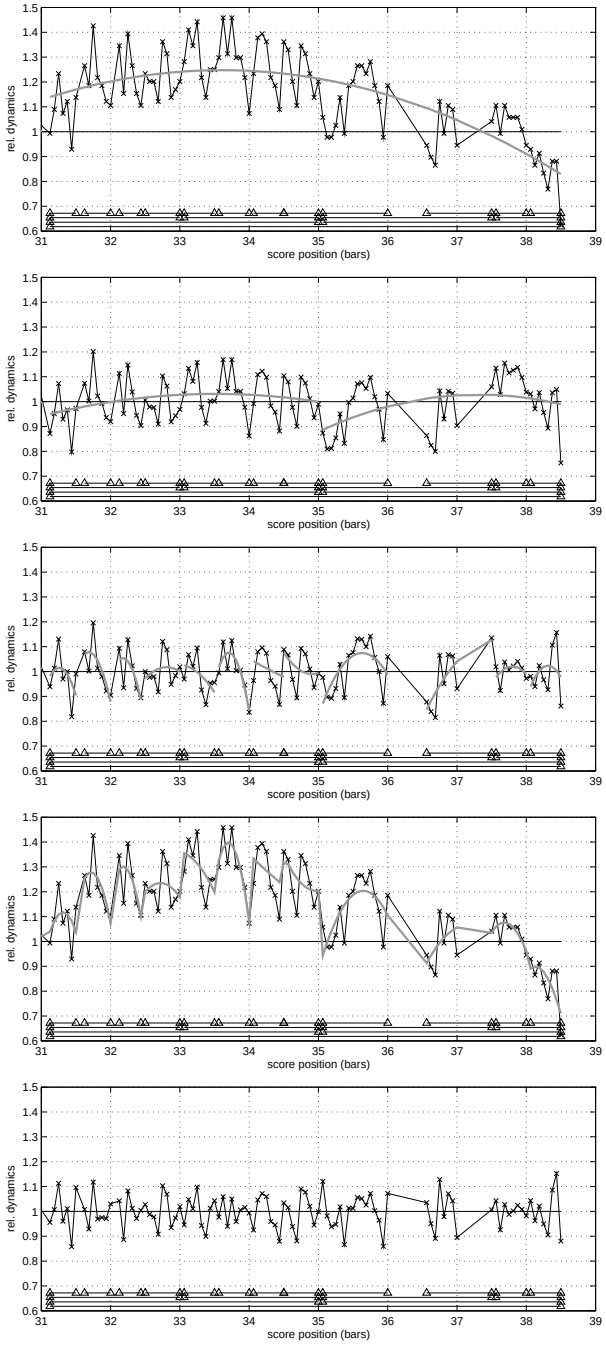

Fig. 2. Multilevel decomposition of dynamics curve of performance of Mozart Sonata K.279:1:1, mm.31–38. From top to bottom: (1) original dynamics curve (black) plus the second-order polynomial giving the best fit at the top phrase level (grey); (2+3) each show, for two lower phrase levels, the dynamics curve after 'subtraction' of the previous approximation, and the best-fitting approximations at this phrase level; (4): 'reconstruction' (grey) of the original curve by the polynomial approximations; (5): *residuals* after all higher-level shapes have been subtracted.

Table 1. Mozart sonata sections used in experiments (to be read as <sonataName>:<movement>:<section>); *notes* refers to 'melody' notes.

				phrases at level			
Piece	tempo	time sig.	notes	1	2	3	4
K.279:1:1	fast	4/4	391	50	19	9	5
K.279:1:2	fast	4/4	638	79	36	14	5
K.280:1:1	fast	3/4	406	42	19	12	4
K.280:1:2	fast	3/4	590	65	34	17	6
K.280:2:1	slow	6/8	94	23	12	6	3
K.280:2:2	slow	6/8	154	37	18	8	4
K.280:3:1	fast	3/8	277	28	19	8	4
K.280:3:2	fast	3/8	379	40	29	13	5
K.282:1:1	slow	4/4	165	24	10	5	2
K.282:1:2	slow	4/4	213	29	12	6	3
K.282:1:3	slow	4/4	31	4	2	1	1
K.283:1:1	fast	3/4	379	53	23	10	5
K.283:1:2	fast	3/4	428	59	32	13	6
K.283:3:1	fast	3/8	326	53	30	12	3
K.283:3:2	fast	3/8	558	79	47	19	6
K.332:2	slow	4/4	477	49	23	12	4
Total:			5506	714	365	165	66

4 A First Experiment

4.1 The Data

The data used in the following experiments were derived from performances of Mozart piano sonatas on a Bösendorfer SE 290 computer-controlled piano by a Viennese concert pianist. The SE 290 is a full concert grand piano with a special mechanism that measures every key and pedal movement with high precision and stores this information in a format similar to MIDI. From these measurements, and from a comparison with the notes in the written score, the tempo and dynamics curves corresponding to the performances can be computed.

A manual phrase structure analysis of some sections of these sonatas was carried out by a musicologist. Phrase structure was marked at four hierarchical levels. The resulting set of annotated pieces available for our experiment is summarized in Table 1. The pieces and performances are quite complex and different in character; automatically learning expressive strategies from them is a challenging task.

4.2 Quantitative Results

A systematic *leave-one-piece-out* cross-validation experiment was carried out on these data. Each of the 16 sonata sections was once set aside as a test piece, while the remaining 15 pieces were used for learning. The learned

Table 2. Results of piece-wise cross-validation experiment. Measures subscripted with D refer to the 'default' (inexpressive) performance, those with L to the performance produced by the learner. *Mean* is the simple mean, *WMean* the weighted mean (individual results weighted by the relative length (number of notes) of the pieces).

	dynamics					tempo				
	MSE_D	MSE_L	MAE_D	MAE_L	$Corr_L$	MSE_D	MSE_L	MAE_D	MAE_L	$Corr_L$
kv279:1:1	.0383	.0409	.1643	.1543	.6170	.0348	.0420	.1220	.1496	.3095
kv279:1:2	.0318	.0736	.1479	.1978	.4157	.0244	.0335	.1004	.1317	.2536
kv280:1:1	.0313	.0275	.1432	.1238	.6809	.0254	.0222	.1053	.1071	.4845
kv280:1:2	.0281	.0480	.1365	.1637	.4517	.0250	.0323	.1074	.1255	.3124
kv280:2:1	.1558	.0831	.3498	.2002	.7168	.0343	.0207	.1189	.1111	.7235
kv280:2:2	.1424	.0879	.3178	.2235	.6980	.0406	.0460	.1349	.1463	.4838
kv280:3:1	.0334	.0139	.1539	.0936	.7656	.0343	.0262	.1218	.1175	.5276
kv280:3:2	.0226	.0711	.1231	.2055	.4492	.0454	.0455	.1365	.1412	.3006
kv282:1:1	.1126	.0476	.2792	.1737	.7609	.0295	.0320	.1212	.1216	.3689
kv282:1:2	.0920	.0538	.2537	.1829	.6909	.0227	.0443	.1096	.1555	.2863
kv282:1:3	.1230	.0757	.2595	.2364	.6698	.1011	.0529	.2354	.1741	.8104
kv283:1:1	.0283	.0236	.1423	.1206	.5907	.0183	.0276	.0918	.1196	.2409
kv283:1:2	.0371	.0515	.1611	.1625	.4469	.0178	.0274	.0932	.1197	.1972
kv283:3:1	.0404	.0319	.1633	.1324	.5993	.0225	.0216	.1024	.1083	.4300
kv283:3:2	.0417	.0399	.1676	.1457	.5305	.0238	.0244	.1069	.1116	.3060
kv332:2	.0919	.0824	.2554	.2328	.5599	.0286	.0436	.1110	.1529	.1684
Mean:	.0657	.0533	.2012	.1718	.6027	.0330	.0339	.1199	.1308	.3877
WMean:	.0486	.0506	.1757	.1662	.5584	.0282	.0332	.1108	.1285	.3192

phrase-level predictions were then applied to the test piece, and the following measures were computed: the *mean squared error* of the learner's predicted curve relative to the actual performance curve produced by the pianist ($MSE = \sum_{i=1}^{n} (pred(n_i) - expr(n_i))^2/n$), the *mean absolute error* ($MAE = \sum_{i=1}^{n} |pred(n_i) - expr(n_i)|/n$), and the *correlation* between predicted and 'true' curve. MSE and MAE were also computed for a *default* curve that would correspond to a purely mechanical, unexpressive performance, i.e., a performance curve consisting of all 1's. That allows us to judge if learning is really better than just doing nothing. The results of the experiment are summarized in Table 2, where each line gives the results obtained on the respective test piece when all others were used for training.

At a first glance, the results look rather disappointing. We are interested in cases where the *relative errors* (i.e., MSE_L/MSE_D and MAE_L/MAE_D) are less than 1.0, that is, where the curves predicted by the learner are closer to the pianist's actual performance than a purely mechanical rendition. In the dynamics dimension, this is the case in 11 out of 16 cases for MSE, and in 12 out of 16 for MAE. Tempo seems basically unpredictable: only in 5 (MSE) and 3 (MAE) cases, respectively, did learning produce an improvement over no learning, at least in terms of these purely quantitative, unmusical measures Also, the correlations vary between 0.77 (kv280:3:1, dynamics) and only 0.17 (kv332:2, tempo).

Averaging over all 16 experiments, it seems that dynamics seems learnable under this scheme to some extent — the relative errors being $RMSE = 0.811$, $RMAE = 0.854$ (unweighted), $RMSE = 1.041$, $RMAE = 0.945$ (weighted) respectively — while tempo seems hard to predict in this way — all relative errors are above 1.0.

5 Improving the results

The above result were rather disappointing. Even keeping in mind that artistic performance of difficult music like Mozart sonatas is a complex and certainly not entirely predictable phenomenon, we had hoped that there would be something predictable about phrase-level tempo and dynamics that a learner could pick up. But the above results are not the end of the story, and in the following sections we explore ways in which they can be improved — at the end we will end up with a system that at least partly makes surprisingly good predictions and even won a prize in a performance contest (see Section 6).

5.1 More homogeneous training sets

One way of improving the results is by noting that Mozart piano sonatas are highly complex music, with a lot of diversity in character. Splitting this set of rather different pieces into more homogeneous subsets and performing learning within these subsets should make the task easier for the learner. For instance, it is known in musicology that absolute tempo has quite an impact on what performance patterns sound acceptable. And indeed, it turns out that simply separating the pieces into fast and slow ones and learning in each of these sets separately considerably increases the number of cases where learning produces an improvement over no learning, both in the dynamics and the tempo domain. Table 3 summarizes the results in terms of wins/losses between learning and no learning for both learning settings. The improvement is obvious. However, the tempo domain is still a problem, with only 7 wins out of 16 cases.

Table 3. Summary of wins vs. losses between learning and no learning; + means curves predicted by the learner better fit the pianist than a flat curve (i.e., relative error < 1), − means the opposite. First line: piece-level cross-validation over all pieces; second line: learning and testing on fast and slow pieces separately.

Training set	MSE/dynamics	MAE/dynamics	MSE/tempo	MAE/tempo
all pieces	11+/5-	12+/4-	5+/11-	3+/13-
slow / fast	14+/2-	14+/2-	7+/9-	7+/9-

5.2 Varying numbers of neighbours and phrase levels

All the results so far were produced by a k-NN learner with $k = 1$. We initially chose $k = 1$ because we could not think of a meaningful way to combine the predictions of several neighbours — simple pairwise averaging of triples of polynomial coefficients seemed not sensible. The three coefficients have a very different impact on the shape of a phrase pattern and thus on the musical effect, and they interact. But in experiments it turned out that in the absence of a more informed combination strategy, even simple averaging of several neighbours' predictions can substantially improve the quality of the predicted curves. Table 4 shows the results obtained by increasing the number k of neighbours used in the prediction. The dynamics results in particular show substantial improvement — the RMSE ($\text{MSE}_L/\text{MSE}_D$) drops from 1.041 for $k = 1$ to 0.654 for $k = 10$, the RMAE from 0.946 to 0.787, and the correlation improves. There is also some improvement in the tempo dimension, with at least the RMSE dropping below 1.0. The attendant slight drop in correlation indicates that with increasing k, the learner tends to reproduce fewer of the local tempo changes of the pianist, while improving the overall fit at higher levels.

In further experiments, it turned out that the highest level of phrasing that was marked by our musicologist — extended phrases that span several, sometimes many, bars — was not well mirrored in the performances by our pianist. Ignoring the highest phrase level and learning and predicting only at the lower three phrase levels leads to even better result, as shown in the last rows in Table 4. Finally, learning beats no learning even in the tempo dimension.

Table 4. Varying the numbers of neighbours and phrase levels. Top: errors (weighted means over all test pieces). Bottom: wins/losses relative to default.

	dynamics					tempo				
Variant	MSE_D	MSE_L	MAE_D	MAE_L	Corr_L	MSE_D	MSE_L	MAE_D	MAE_L	Corr_L
4 levels, 1NN	.0486	.0506	.1757	.1662	.5584	.0282	.0332	.1108	.1285	.3192
4 levels, 2NN	.0486	.0395	.1757	.1520	.5637	.0282	.0299	.1108	.1239	.3105
4 levels, 3NN	.0486	.0354	.1757	.1466	.5918	.0282	.0297	.1108	.1231	.2871
4 levels, 5NN	.0486	.0336	.1757	.1424	.6114	.0282	.0292	.1108	.1208	.2786
4 levels, 10NN	.0486	.0318	.1757	.1382	.6166	.0282	.0276	.1108	.1157	.2960
3 levels, 10NN	.0486	.0312	.1757	.1380	.6096	.0282	.0271	.1108	.1136	.2937

Variant	MSE/dynamics	MAE/dynamics	MSE/tempo	MAE/tempo
4 levels, 1NN	11+/5-	12+/4-	5+/11-	3+/13-
4 levels, 2NN	12+/4-	13+/3-	7+/9-	4+/12-
4 levels, 3NN	12+/4-	14+/2-	6+/10-	2+/1=/13-
4 levels, 5NN	14+/2-	14+/2-	8+/8-	5+/11-
4 levels, 10NN	14+/2-	15+/1-	10+/6-	6+/10-
3 levels, 10NN	15+/1-	15+/1-	11+/1=/4-	9+/7-

5.3 Improving the musical quality by learning local rules

As Figure 2 above shows quite clearly, hierarchically nested quadratic functions tend to reconstruct the larger trends in a performance curve quite well, but they cannot describe all the detailed local nuances added by a pianist, e.g., the emphasis on particular notes. Local nuances will be left over in what we call the *residuals* — the tempo and dynamics fluctuations left unexplained by the phrase-level polynomials. These can be expected to represent a mixture of noise and meaningful or intended local deviations.

In order to also learn a model of these intended deviations, we applied a rule learning algorithm to the residuals. The goal was to induce note-level rules that predict when the pianist will significantly lengthen or shorten a particular note relative to its context, or play it significantly louder or softer. The learning algorithm used was PLCG, which has been shown to be quite effective in distinguishing between signal and noise and discovering reliable rules when only a part of the data can be explained [12]. Combining the learned rules with a simple numeric prediction scheme again based on a k-NN algorithm produces a partial model of note-level nuances that predicts local timing and dynamics changes to be applied to some individual notes.

Combining these note-level predictions with the phrase-level predictions yields an additional slight reduction in MSE and MAE both for tempo and dynamics, but the difference is almost negligible (though consistently in favour of the combined learner). The interesting fact is that the correlation values improve significantly. For instance, combining the note-level model with the *3 levels, 10NN* learner yields (weighted mean) correlations of 0.6182 for dynamics and 0.3588 for tempo — for tempo in particular, this is significantly higher than any of the values in Table 4. Obviously, the note-level model captures some important local choices of the pianist (which also strongly contribute to the musical quality of the performance).

5.4 A fairer comparison

A final way of 'improving' the results is to note that the error measures we used so far in this paper may not be quite appropriate. What was compared was the performance (tempo or dynamics) curve produced by composing the polynomials predicted by the learner, with the curve corresponding to the pianist's actual performance. However, what the k-NN learner learned from was not the actual performance curves, but an *approximation*, namely, the polynomials fitted to the curve at various phrase levels. And maybe this approximation is not very good to begin with. This is partly confirmed by a look at Table 5, which summarizes how well the four-level decompositions (without the residuals) reconstruct the respective training curves.[1] The dynamics curves are generally better approximated by the four levels of polynomials than the tempo curves, and the difference is dramatic. That may explain in particular why our tempo results were so poor.

[1] That is, we look not at the performance of the learning system, but only at the effectiveness of approximating a given curve by four levels of quadratic functions.

Table 5. Summary of fit of four-level polynomial decomposition on the training data. Measures subscripted with D refer to the 'default' (mechanical, inexpressive) performances (repeated from table 2), those with P to the fit of the curves reconstructed by the polynomial decompositions.

	MSE_D	MSE_P	RMSE	MAE_D	MAE_P	RMAE	$Corr_P$
dynamics	.0486	.0049	.1008	.1757	.0501	.2851	.9397
tempo	.0282	.0144	.5106	.1108	.0755	.6814	.6954

Table 6. Summary of errors resulting from comparing the learner's predictions to the 'reconstructed' training curve rather than the actual performance curve produced by the pianist. Shown are weighted means over all training examples.

	dynamics					tempo				
Variant	MSE_D	MSE_L	MAE_D	MAE_L	$Corr_L$	MSE_D	MSE_L	MAE_D	MAE_L	$Corr_L$
4 levels, 1NN	.0437	.0457	.1665	.1543	.5936	.0141	.0190	.0811	.0959	.4517
4 levels, 2NN	.0437	.0345	.1665	.1394	.5995	.0141	.0158	.0811	.0919	.4361
4 levels, 3NN	.0437	.0304	.1665	.1339	.6296	.0141	.0156	.0811	.0922	.4020
4 levels, 5NN	.0437	.0286	.1665	.1292	.6522	.0141	.0151	.0811	.0894	.3829
4 levels, 10NN	.0437	.0268	.1665	.1249	.6571	.0141	.0135	.0811	.0831	.4137
3 levels, 10NN	.0437	.0262	.1664	.1246	.6489	.0141	.0130	.0811	.0806	.4155

The finding implied by Table 5 has implications for musicology, where it has hitherto been believed (but never systematically tested with large numbers of real performances) that quadratic functions are a reasonable model class for expressive timing (e.g., [7, 13]). But it also suggests that the above way of computing prediction error was not entirely fair. It would seem more appropriate to compare the predicted curves not to the actual performance curve, but to the approximation curve that is implied by the four levels of quadratic functions that were used as training examples.[2] Correctly predicting these is the best the learner could hope to achieve. Table 6 shows the error figures we obtain in this way, for all the k-NN learners described above.

As can be seen, the situation indeed now looks better for our learner (compare this to Table 4 above). Note the substantially higher correlations in the tempo domain — it is obviously easier to predict approximated curves than real curves. There is also some improvement in terms of the numbers of wins vs. losses against the default. For example, with 3 levels of phrasing and 10 nearest neighbours (last line in Table 6) we get win/loss ratios of 15+/1- for dynamics (both for MSE and MAE) and 11+/1=/4- (MSE) and 10+/6- (MAE) for tempo. That is the best we managed to obtain so far.

[2] Actually, the most direct comparison would be between the predicted and 'true' polynomial coefficients; but numeric errors and correlations at this level would be hard to interpret intuitively or musically.

Of course, this 'trick' of changing the definition of error does not change the musical quality of the results, but it gives a more realistic picture of the capabilities of nearest-neighbour learning in this domain.

6 Musical Results

The musical quality of the results is hard to describe in a paper. Generally, the quality varies strongly between pieces, and even within pieces — passages that are musically sensible are sometimes followed by rather extreme errors, at least in musical terms. One incorrect shape can seriously compromise the quality of a composite performance curve that would otherwise be perfectly musical. The quantitative measures MSE, MAE, and correlation are not necessarily indicative of the quality of the listening experience.

Figure 3 tries to give the reader an impression of how well the learning system (phrase-level + note-level) can predict how a pianist is going to play a given passage. This is a case where prediction worked quite well, especially concerning the higher-level aspects. Some of the local patterns were also predicted quite well, while others were obviously missed.

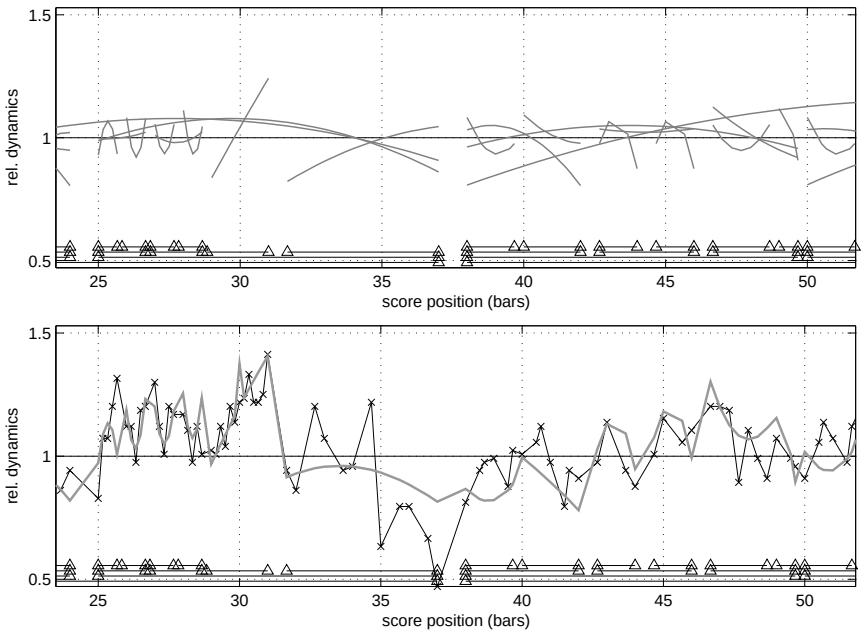

Fig. 3. Learner's predictions for the dynamics curve of Mozart Sonata K.280, 3rd movement, mm. 25–50. Top: quadratic performance shapes predicted for phrases at four levels; bottom: composite predicted dynamics curve resulting from phrase-level shapes and note-level predictions (grey, without markers) vs. pianist's actual dynamics (black, with markers).

The curve shown in Figure 3 is from a computer-generated performance of the Mozart piano sonata K.280 in F major that was produced by the 1-NN learning algorithm + rules learned from the residuals, after training on other sonatas. A recording of this performance was submitted to an International Computer Piano Performance Rendering Contest[3] (RENCON'02) in Tokyo in September 2002, where it won Second Prize behind a rule-based rendering system that had been carefully tuned by hand. The rating was done by a jury of human listeners. While this result does in no way imply that a machine will ever be able to learn to play music like a human artist, we do consider it a nice success for a machine learning system. This was an early result, and we expect further improvement by increasing the number of neighbours k, refining the strategy for combining predictions, and introducing contextual knowledge (see below). We hope to demonstrate some interesting sound examples at the conference.

7 Conclusions

To summarize, this paper has presented a system that combines case-based with rule-based learning in the difficult domain of expressive music performance. First experimental results are at least encouraging. Case-based learning for expressive performance has been proposed before in the domain of expressive phrasing in jazz [1, 6], where the promise of CBR was shown, but the evaluation was mostly qualitative and based on relatively small numbers of phrases. The work presented here thus constitutes the first large-scale quantitative evaluation of case-based learning for expressive performance (against a high-class concert pianist).

There are numerous possibilities for improvement that are currently being investigated. One obvious limitation is the propositional attribute-value representation used to characterize phrases, which does not permit the learner to refer to details of the internal structure and content of phrases. Here, we now investigate the use of first-order logic representations and ILP methods [3].

A related problem is that phrasal shapes are predicted individually and independently of the shapes associated with (or predicted for) other, related phrases, i.e., phrases that contain the current phrase, or are contained by it. Obviously, this is too simplistic. Shapes applied at different levels are highly dependent. We are now trying to introduce dependency information via the notion of *context*; very preliminary experiments with a new relational instance-based learner with a context-sensitive similarity measure indicate that this may be fruitful.

The above experiments showed that combining predictions by more than one nearest neighbour greatly improves the results. Our current stratey for combining the predictions of $k > 1$ neighbours — simple coefficient averaging — is too simplistic. A solution we are going to study is to compute the actual performance curve that would be jointly produced by the k predicted polynomials, and then again fit a single polynomial to the resulting curve to get the final, 'combined' coefficients; in other words, we will average curves instead of coefficients.

[3] yes, there is such a thing ...

A general problem with nearest neigbour learning is that it does not produce interpretable models. As the ultimate goal of our project is to contribute new insights to musical performance research, this is a serious drawback. Along these lines, we will investigate both feature selection/weighting and alternative learning algorithms.

Acknowledgments

This research is made possible by a START Research Prize by the Austrian Federal Government, administered by the Austrian *Fonds zur Förderung der Wissenschaftlichen Forschung (FWF)* (project no. Y99-INF). Additional support for our research on machine learning and music is provided by the European project HPRN-CT-2000-00115 (MOSART). The Austrian Research Institute for Artificial Intelligence acknowledges basic financial support from the Austrian Federal Ministry for Education, Science, and Culture. Thanks to Werner Goebl for performing the harmonic and phrase structure analysis of the Mozart sonatas.

References

1. Arcos, J.L. and López de Mántaras (2001). An Interactive CBR Approach for Generating Expressive Music. *Journal of Applied Intelligence* 14(1), 115–129.
2. Duda, R. and Hart, P. (1967). *Pattern Classification and Scene Analysis.* New York, NY: John Wiley & Sons.
3. Dzeroski, S. and Lavrac, N. (eds.) (2001). *Relational Data Mining: Inductive Logic Programming for Knowledge Discovery in Databases.* Berlin: Springer Verlag.
4. Gabrielsson, A. (1999). The Performance of Music. In D. Deutsch (ed.), *The Psychology of Music (2nd ed.)*, 501–602. San Diego, CA: Academic Press.
5. Kronman, U. and Sundberg, J. (1987). Is the Musical Ritard an Allusion to Physical Motion? In A. Gabrielsson (ed.), *Action and Perception in Rhythm and Music*, 57–68. Stockholm, Sweden: Royal Swedish Academy of Music No.55.
6. López de Mántaras, R. and Arcos, J.L. (2002). AI and Music: From Composition to Expressive Performances. *AI Magazine* 23(3), 43–57.
7. Todd, N. (1989). Towards a Cognitive Theory of Expression: The Performance and Perception of Rubato. *Contemporary Music Review, vol. 4*, pp. 405–416.
8. Todd, N. McA. (1992). The Dynamics of Dynamics: A Model of Musical Expression. *Journal of the Acoustical Society of America* 91, 3540–3550.
9. Widmer, G. (2001). Using AI and Machine Learning to Study Expressive Music Performance: Project Survey and First Report. *AI Communications* 14(3), 149-162.
10. Widmer, G. (2002). In Search of the Horowitz Factor: Interim Report on a Musical Discovery Project. In *Proceedings of the 5th International Conference on Discovery Science (DS'02)*, Lübeck, Germany. Berlin: Springer Verlag.
11. Widmer, G. (2002). Machine Discoveries: A Few Simple, Robust Local Expression Principles. *Journal of New Music Research* 31(1).
12. Widmer, G. (2003). Discovering Simple Rule in Complex Data: A Meta-learning Algorithm and Some Surprising Musical Discoveries. *Artificial Intelligence* (in press).
13. Windsor, W.L. and Clarke, E.F. (1997). Expressive Timing and Dynamics in Real and Artificial Musical Performances: Using an Algorithm as an Analytical Tool. *Music Perception* 15(2), 127–152.