Playing Mozart by Analogy: Learning Multi-level Timing and Dynamics Strategies

Gerhard Widmer[1,2] and Asmir Tobudic[2]
[1]Department of Medical Cybernetics and Artificial Intelligence,
University of Vienna
[2]Austrian Research Institute for Artificial Intelligence, Vienna
email: {gerhard|asmir}@ai.univie.ac.at

Abstract

The paper describes basic research in the area of machine learning and musical expression. A first step towards automatic induction of multi-level models of expressive performance (currently only tempo and dynamics) from real performances by skilled pianists is presented. The goal is to learn to apply sensible tempo and dynamics 'shapes' at various levels of the hierarchical musical phrase structure. We propose a general method for decomposing given expression curves into elementary shapes at different levels, and for separating phrase-level expression patterns from local, note-level ones. We then present a hybrid learning system that learns to predict, via two different learning algorithms, both note-level and phrase-level expressive patterns, and combines these predictions into complex composite expression curves for new pieces. Experimental results indicate that the approach is generally viable; however, we also discuss a number of severe limitations that still need to be overcome in order to arrive at truly musical machine-generated performances.

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 Artificial Intelligence and, in particular, inductive machine learning methods (Widmer 2001c). This is to be regarded as basic research. We do not intend to engineer computer programs that generate music performances that sound as human-like as possible. Rather, our 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 (e.g., predictive models of tempo, timing, dynamics, etc.). By analysing the models induced by the machine, we hope to get new insights into fundamental principles underlying the complex phenomenon of expressive music performance, and in this way contribute to the growing body of scientific knowledge about performance (see (Gabrielsson 1999) for an excellent overview of current knowledge in this area).

Previous research has shown that computers can indeed find and describe interesting and useful regularities at the level of individual notes. Using a new machine learning algorithm (Widmer 2001a), we succeeded in discovering a small set of simple, robust, and highly general rules that predict a substantial part of the note-level choices of a performer (e.g., whether he will shorten or lengthen a particular note) with high precision (Widmer 2001b). However, it became equally clear (actually, it was clear from the outset) that this low level of single notes is far from sufficient as a basis for a complete model of expressive performance, and that these note-level models must be complemented with models of expression at higher levels of musical organization (e.g., the level of phrases).

The work presented here is a first preliminary step in this direction. We describe a system that learns to predict elementary tempo and dynamics 'shapes' at different levels of the hierarchical musical phrase structure, and combines these predictions with local timing and dynamics effects predicted by learned note-level models. To do this, the learning system must first be able to decompose given expression curves into elementary patterns that can be associated with individual phrases (at different phrase levels), in order to obtain meaningful training examples for phrase-level learning, and to separate phrase-level effects from local note-level effects (which will be learned by a separate learning algorithm). Likewise, we need a strategy for combining expressive shapes predicted at different levels into one final composite expression curve.

In the following, we describe our current solution to the problems of expression curve decomposition and re-combination and present a first prototype system that combines two types of learning algorithms: a simple nearest neighbor algorithm that predicts phrase-level expressive shapes in new pieces by 'analogy' to shapes identified in similar phrases in other pieces, and a rule learning algorithm that learns prediction rules for note-level effects from the 'residuals' that cannot be attributed to the phrase structure by the expression

decomposition algorithm. Experiments with performances of various sections of Mozart piano sonatas show that the approach is viable in principle.

However, our approach still suffers from a number of severe limitations, and these will be discussed in the final section of this paper.

2 Multilevel Decomposition of Expression Curves

Input to our learning system are the scores of musical pieces plus measurements of the tempo and dynamics variations applied by a pianist in a particular performance. These variations are given in the form of *tempo* and *dynamics curves* and represent the local tempo and the relative loudness of each melody note of the piece, respectively. Both tempo and loudness are represented as multiplicative factors, relative to the average tempo and dynamics of the piece. For instance, a tempo indication 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 an expression (dynamics or tempo) curve, the learner is first faced with the problem of extracting the *training examples* for phrase-level and note-level learning. That is, the complex curve must be decomposed into basic expressive 'shapes' that represent the most likely contribution of each phrase to the overall expression curve.

As approximation functions to represent these shapes we decided to use the class of second-degree polynomials (i.e., functions of the form $y = ax^2 + bx + c$), because there is ample evidence from previous research that high-level tempo and dynamics are well characterized by quadratic or parabolic functions (Todd 1992; Repp 1992; Kronman and Sundberg 1987). Decomposing a given expression 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 approximations. 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. The rudimentary expression curve left after all levels of phrase approximations have been subtracted is called the *residual curve*.

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 expression curve (i.e., tempo or dynamics values) associated with these notes. Fitting a second-order polynomial onto E_p then means finding a function $f_p(x) = a^2x + bx + c$ such that

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

is minimal.

Given an expression curve (i.e., sequence of tempo or dynamics values) $E_p = \{expr(n_1), ..., expr(n_k)\}$ 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 expression curve.

To illustrate, Figure 1 shows the dynamics curve of the last part (mm.31–38) of the Mozart Piano Sonata K.279 (C major), 1st 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 expression curve by polynomials at these four phrase levels. The red line in level (e) of the figure shows how much of the original curve is accounted for by the four levels of approximations, and level (f) shows the *residual* that is not explained by the higher-level patterns and will be submitted to a rule learner for note-level learning.

3 Learning to Predict Tempo and Dynamics

Given expression curves decomposed into levels of phrasal shapes (approximation polynomials) and a residual curve, we apply a two-level learning strategy to these training examples. Phrase shapes for phrases in new pieces are predicted by a standard nearest-neighbor learning algorithm (see section 3.1), and the residuals are fed into an inductive rule learning algorithm that induces rules that predict low, note-level deviations (section 3.2). For prediction in new pieces, note-level and phrase-level predictions are then combined in a straightforward way (section 3.3).

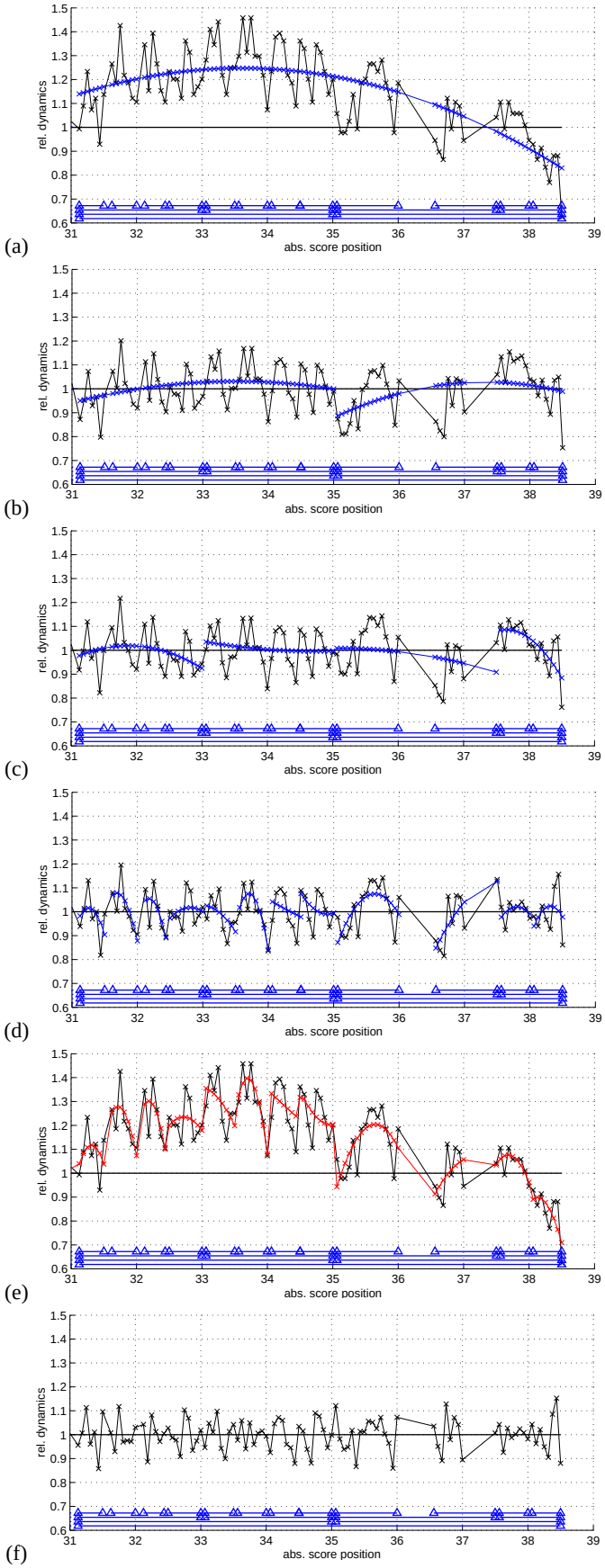

Figure 1: [best viewed in color] Multilevel decomposition of dynamics curve of performance of Mozart Sonata K.279:1:1, mm.31–38. Level (a): original dynamics curve plus the second-order polynomial giving the best fit at the top phrase level (blue); levels (b–d) each show, for successively lower phrase levels, the dynamics curve after 'subtraction' of the previous approximation, and the best-fitting approximations at this phrase level; Level (e): 'reconstruction' (red) of the original curve by the four levels of polynomial approximations; level (f): *residual* after all higher-level shapes have been subtracted.

3.1 Phrase-level learning via nearest neighbor prediction

Given a set of training performances with tempo and dynamics curves decomposed into phrasal shapes and residuals as described above, a straightforward *Nearest Neighbor* learning algorithm with one neighbor (Duda and Hart 1967) is used to predict phrase shapes (polynomials) for phrases in new pieces. Given a phrase in a new piece, the algorithm searches its memory for the most similar phrase in the known pieces (at the same phrase level) and predicts the polynomial associated with this phrase as the appropriate shape for the new phrase.

The similarity between phrases is computed as the inverse of the standard Euclidean distance between the new target phrase and a phrase retrieved from memory. 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, 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.

We have decided to use only the one nearest neighbor for prediction (instead of performing general k-NN, with $k > 1$), because what is predicted is not a scalar value, but a triple of values (the three parameters a, b, c of an approximation polynomial $y = ax^2 + bx + c$), where it is not quite clear how several predictions would be combined. Also, an obvious drawback of nearest neighbor algorithms is that they do not produce explicit, interpretable models — they make predictions, but they do not describe the data and the target classes. As a next research step, we plan to investigate the utility of other inductive learning algorithms for phrase-level learning, so that we will also get interpretable models that we can learn something from.

3.2 Rule-based learning of 'residuals'

As figure 1 shows quite clearly, the quadratic phrasal 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 shapes (see level (f) of figure 1). We would like to also learn a model of these local expressive choices.

Actually, the residuals can be expected to represent a mixture of noise and meaningful or intended local deviations. To learn reliable rules for predicting note-level expressive actions, we need a learning algorithm that is capable of effectively distinguishing between signal and noise. Nearest neighbor algorithms are not particularly suitable here. Instead, we have chosen to use PLCG (Widmer 2001a), a new inductive rule learning algorithm that has been shown to be highly effective in discovering reliable, robust rules from complex data where only a part of the data can actually be explained. PLCG also has the advantage that it learns explicit sets of prediction rules, so that we will get explicit interpretable models at least at the note level.

PLCG learns sets of classification rules for discrete classification problems. In order to apply it to the residual learning problem, we need to define discrete target classes. The simple solution adopted here, which turns out to work sufficiently well, is to assign all expression values above 1.0 to a class `above1` and all others to class `below1`. The training examples at the residual level are single notes, described via a set of attributes that represent both intrinsic properties (such as scale degree, duration, metrical position) and some aspects of the local context (e.g., melodic properties like the size and direction of the intervals between the note and its predecessor and successor notes, and rhythmic properties like the durations of surrounding notes and some abstractions thereof). An example of the kinds of rules that PLCG discovered under these definitions is shown in section 4.3 below.

To be able to predict numeric note-level expression values, PLCG has been extended with a numeric learning method — again, a nearest-neighbor algorithm: all the training examples (notes) covered by a learned rule are stored together with the rule. When predicting an expression value for a new note in a new test piece, PLCG first finds a matching rule to decide what category to apply, and then performs a k-NN search among the training examples stored with that rule, to find the k (currently 3) notes most similar to the current one. The expression value predicted for the new note is then a distance-weighted average of the values associated with the k most similar notes.

3.3 Combining phrase-level and note-level predictions

As noted above, the expression values that make up our expression curves are to be interpreted as multiplicative factors. Applying multi-level predictions made by the phrase-level and note-level learners for new pieces is thus straightforward — it 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' expression curve (i.e., a list of 1.0 values) and then successively multiplies the current value by the phrase-level predictions and the note-level prediction.

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 $exp(n_i)$ to be applied to it is computed as

$$exp(n_i) = pred_{PLCG}(n_i) \times \prod_{j=1}^{m} f_{p_j}(onset_{p_j}(n_i)),$$

where $pred_{PLCG}(n_i)$ is the note-level prediction of tempo or duration made by the 'residual rules' learned by PLCG, and f_{p_j} is the approximation polynomial predicted as being best suited for the j^{th}-level phrase p_j by the nearest-neighbor learning algorithm.

4 Experiments

4.1 The Data

In the following, we briefly present some experiments with our new approach. The data used for the experiments were derived from performances of Mozart piano sonatas by a Viennese concert pianist on a Bösendorfer SE 290 computer-controlled grand piano. The measurements made by the piano permit the exact calculation of the tempo and dynamics curves corresponding to these performances.

A manual phrase structure analysis (and harmonic 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.

sonata section		notes	phrases at level 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

Table 1: Sonata sections used in experiments (*notes* refers to 'melody' notes).

4.2 Systematic Quantitative Evaluation

A systematic *leave-one-piece-out* cross-validation experiment was carried out to quantitatively assess the results achievable with our approach. Each of the 16 sections was once set aside as a test piece, while the remaining 15 pieces were used for learning. The learned phrase-level and note-level predictions were then applied to the test piece, and the following measures were computed: the *mean squared error* of the system's predictions on the piece relative to the actual expression 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 particularly punishes curves that produce a few extreme 'errors' (i.e., deviations from what the pianist actually does). MSE and MAE were also computed for a *default* curve that would correspond to a purely mechanical, unexpressive performance (i.e., an expression 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.

As can be seen, the results are mixed. We are interested in cases where the *relative errors* (i.e., $\text{MSE}_L/\text{MSE}_D$ and $\text{MAE}_L/\text{MAE}_D$) 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 not as well predictable: only in 6 out of 16 cases (both w.r.t. MSE and MAE) does learning produce an improvement over a mechanical performance (at least in terms of these purely quantitative, unmusical measures). Also, the correlations vary between 0.78 (kv280:3:1, dynamics) and only 0.19 (kv283:1:2, tempo).

Averaging over all 16 experiments, it seems that dynamics seems learnable under this scheme to some extent (the relative errors being $RMSE = 0.799$ and $RMAE = 0.845$), while tempo seems hard to predict in this way ($RMSE \approx 1$, $RMAE = 1.075$). The correlations are quite high in most cases.

The results can be improved if we split this set of rather different pieces into more homogeneous subsets, and perform learning within these subsets. For instance, 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 improvement over no learning — again, especially in the domain of dynamics; tempo remains a problem. Table 3 summarizes the results in terms of wins/losses between learning and no learning.

Although also the tempo can be predicted quite well in some pieces, the tempo results in general seem quite disappointing. But a closer analysis reveals that part of these rather poor results for tempo can be attributed to problems with the quadratic approximations. It turns out that quadratic or parabolic approximations

	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	.0411	.1643	.1544	.6212	.0348	.0406	.1220	.1479	.3550
kv279:1:2	.0318	.0737	.1479	.1975	.4204	.0244	.0335	.1004	.1327	.2984
kv280:1:1	.0313	.0266	.1432	.1226	.7080	.0254	.0192	.1053	.1032	.5821
kv280:1:2	.0281	.0491	.1365	.1642	.4711	.0250	.0304	.1074	.1232	.4010
kv280:2:1	.1558	.0831	.3498	.2002	.7168	.0343	.0187	.1189	.1079	.7518
kv280:2:2	.1424	.0879	.3178	.2235	.6980	.0406	.0431	.1349	.1400	.5128
kv280:3:1	.0334	.0134	.1539	.0916	.7765	.0343	.0244	.1218	.1136	.5813
kv280:3:2	.0226	.0728	.1231	.2089	.4590	.0454	.0418	.1365	.1327	.3953
kv282:1:1	.1126	.0465	.2792	.1721	.7667	.0295	.0315	.1212	.1160	.4222
kv282:1:2	.0920	.0521	.2537	.1782	.6976	.0227	.0421	.1096	.1477	.3460
kv282:1:3	.1230	.0613	.2595	.2105	.7200	.1011	.0583	.2354	.1815	.6676
kv283:1:1	.0283	.0234	.1423	.1194	.6007	.0183	.0274	.0918	.1193	.2441
kv283:1:2	.0371	.0520	.1611	.1629	.4406	.0178	.0275	.0932	.1208	.1948
kv283:3:1	.0404	.0320	.1633	.1323	.6030	.0225	.0214	.1024	.1085	.4460
kv283:3:2	.0417	.0402	.1676	.1466	.5336	.0238	.0254	.1069	.1150	.2948
kv332:2	.0919	.0844	.2554	.2370	.5475	.0286	.0416	.1110	.1520	.2787
Mean:	.0657	.0525	.2012	.1701	.6113	.0330	.0329	.1199	.1289	.4232

Table 2: Results of cross-validation experiment. Measures subscripted with D refer to the 'default' (mechanical, inexpressive) performance, those with L to the performance produced by the learner.

	dynamics	tempo
Learning from all pieces:		
MSE	11+/5-	6+/10-
MAE	12+/4-	6+/10-
Learning from slow and fast pieces separately:		
MSE	14+/2-	8+/8-
MAE	14+/2-	8+/8-

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.

might not be as suitable for describing expressive timing as has hitherto been believed. When we look at how well the four-level decompositions (without the residues) reconstruct the respective training curves,[1] we find that the dynamics curves are generally better approximated by the four levels of polynomials than the tempo curves. The overall figures are given in table 4. The difference between tempo and dynamics is quite dramatic. This phenomenon definitely deserves more detailed investigations.

Generally, we must keep in mind that our current representation for phrases is extremely limited: characterizing phrases via a small number of global attributes does not give the learner access to the detailed contents of a phrase. The results might improve substantially if we had a better representation. Extensive studies in this direction are currently planned.

Another question of interest is whether the learning

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

	MSE_D	MSE_P	MAE_D	MAE_P	$Corr_P$
dyn.	.0657	.0055	.2012	.0523	.9456
tempo	.0330	.0127	.1199	.0720	.7421

Table 4: 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.

of note-level rules from the *residuals* contributes anything to the results. When we disable note-level learning and only use phrase-level learning for predicting expression curves, the results are as shown in table 5. Comparing this to table 2, we note that the note-level rules do indeed improve the quality of the results, both in terms of error and correlation. The improvement may be slight in quantitative terms, but listening tests show that the predicted residuals contribute important audible effects that improve the musical quality of the resulting performances.

	MSE_D	MSE_L	MAE_D	MAE_L	$Corr_L$
dyn.	.0657	.0533	.2012	.1718	.6027
tempo	.0330	.0339	.1199	.1308	.3877

Table 5: Results of learning at phrase-levels only (i.e., without residual predictions).

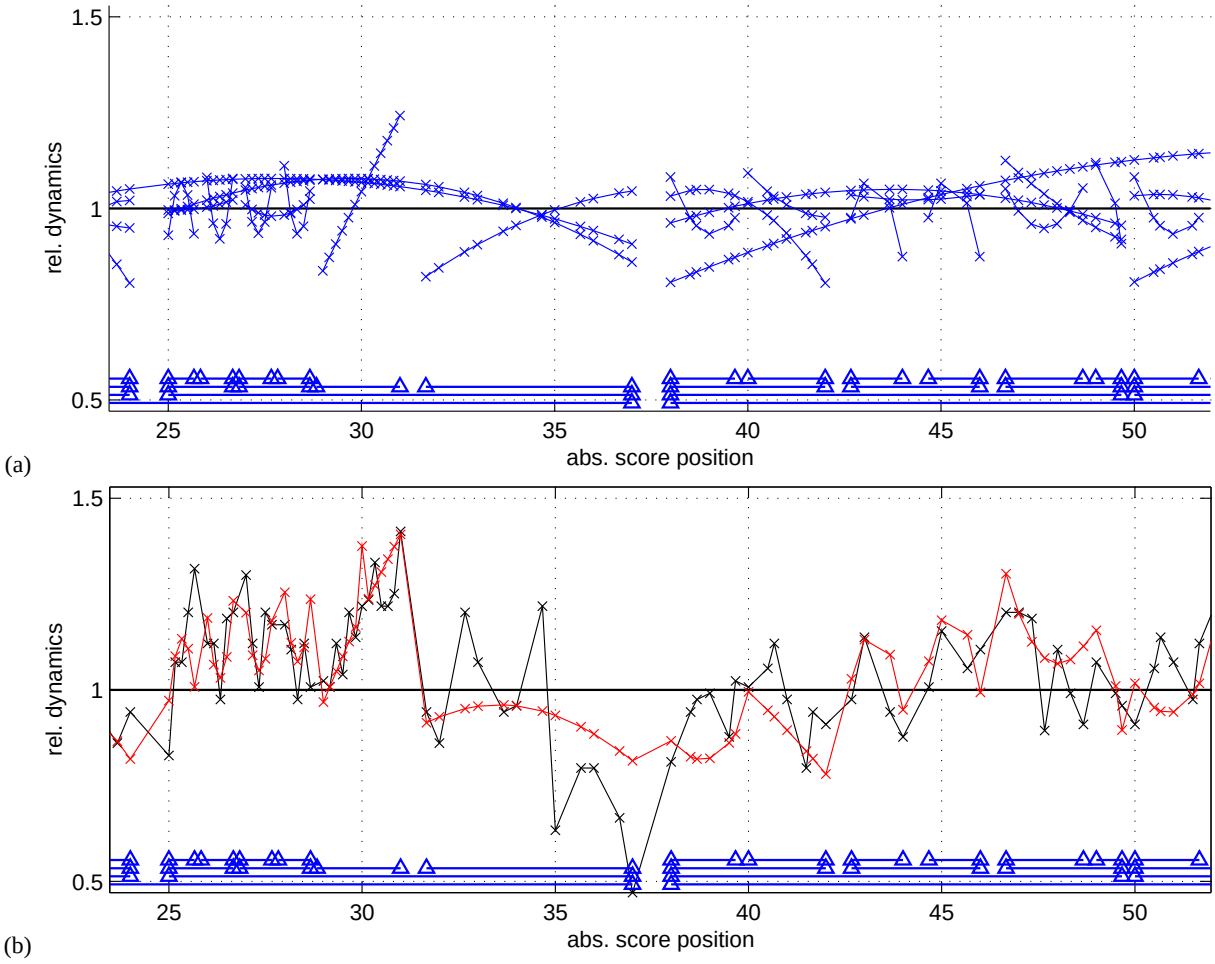

Figure 2: [best viewed in color] Learner's predictions for the dynamics curve of Mozart Sonata K.280, 1st movement, mm. 25–50. Level (a): quadratic expression shapes predicted for phrases at four levels (blue); (b): composite predicted dynamics curve resulting from phrase-level shapes and note-level predictions (red) vs. pianist's actual dynamics (black).

4.3 Qualitative Results

It is instructive to look at the expression curves produced by the learning system, and to listen to the resulting 'expressive' performances. The quality varies strongly, 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 expression curve that would otherwise be perfectly musical.

Figure 2 shows 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. The piece from which this passage was taken — the first section of movement 3 of the piano sonata K.280 — is also enclosed as a sound example (see below).

With respect to note-level learning, an analysis of the rules learned by PLCG from the residuals shows that PLCG indeed seems to discover quite general and sensible principles of local timing and dynamics. An example of a rule discovered by PLCG is

RULE TL4:
`below1 IF`
`next_dur_ratio` $\leq 1/3$ &
`dur_next` > 1

"Lengthen a note if it is followed by a substantially longer note (i.e., the ratio between its duration and that of the next note is $\leq$ 1:3) and if the next note is longer than 1 beat."

This kind of principle — slightly delaying a long note that follows short ones — has been noted before and indeed has been found to be quite a general principle, not only in Mozart performances (Widmer 2001b).

5 Notes Concerning the Enclosed Sound Examples

Enclosed with this paper is a test piece (the first section of the third movement of the Mozart piano sonata

K.280, F major), as played by the system after learning from the other sonata sections. For comparison, we also include a purely mechanical, inexpressive version produced directly from the score.

It should be kept in mind that this is purely a result of automated learning. Only tempo and dynamics were shaped by the system. Articulation and pedalling are simply ignored, so the result cannot be expected to sound truly pianist-like. Also, grace notes and other ornaments are currently inserted via an extremely simple and crude heuristic and should be made to sound much more musical, depending on the context. The only other effect that was added was a simple dynamic enhancement of the melody: melody notes were made to be 20% louder than the rest, in order to make the melody more clearly audible. This factor is roughly consistent with empirical results of a recent study on melody dynamics and melody lead (Goebl 2001).

The example demonstrates the musical potential of our system, but also exhibits some obvious problems. Still, overall the system's performance sounds quite lively and not uninteresting, with a number of quite musical developments both in tempo and dynamics, and with a closing of the piece by a nice final ritard.

6 Discussion and Future Work

In this paper, we have presented a two-level approach to learning both phrase-level and note-level timing and dynamics strategies for expressive music performance. Both qualitative and quantitative analyses show that the approach has some promise, but of course there are still some severe problems that must be solved.

One obvious limitation is the propositional attribute-value representation we are currently using to characterize phrases, which does not permit the learner to refer to details of the internal structure and content of phrases. As a next step, we will look at possibilities of using more expressive representation languages and related learning algorithms (e.g. relational learning methods from the area of Inductive Logic Programming (Lavrac and Dzeroski 1994)).

A general problem with nearest neigbor learning is that it does not produce interpretable models. As the explicit goal of our project is to contribute new insights to musical performance research, this is a serious drawback. Alternative learning algorithms will have to be investigated.

A more difficult problem is the fact that we are currently predicting phrasal shapes individually and independently of the shapes associated (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. Predicting highly dependent concepts at different levels of resolution is a new kind of scenario for machine learning, with potential applications in many domains, and we are planning to study this problem in a general way.

Acknowledgments

This research is part of the START programme Y99-INF, financed by the Austrian Federal Ministry for Education, Science, and Culture. 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

Duda, R. and P. Hart (1967). *Pattern Classification and Scene Analysis*. New York, NY: Wyley & Sons.

Gabrielsson, A. (1999). The performance of music. In D. Deutsch (Ed.), *The Psychology of Music (2nd ed.)*, San Diego, CA, pp. 501–602. Academic Press.

Goebl, W. (2001). Melody lead in piano performance: Expressive device or artifact? *Journal of the Acoustical Society of America 110*(1), 563–572.

Kronman, U. and J. Sundberg (1987). Is the musical ritard an allusion to physical motion? In A. Gabrielsson (Ed.), *Action and Perception in Rhythm and Music*, Stockholm, Sweden, pp. 57–68. Royal Swedish Academy of Music No.55.

Lavrac, N. and S. Dzeroski (1994). *Inductive Logic Programming*. Chichester, NY: Ellis Horwood.

Repp, B. (1992). Diversity and commonality in music performance: An analysis of timing microstructure in schumann's 'träumerei'. *Journal of the Acoustical Society of America 92*(5), 2546–2568.

Todd, N. M. (1992). The dynamics of dynamics: A model of musical expression. *Journal of the Acoustical Society of America 91*, 3540–3550.

Widmer, G. (2001a). Discovering strong principles of expressive music performance with the PLCG rule learning strategy. In *Proceedings of the 11th European Conference on Machine Learning (ECML'01)*, Berlin. Springer Verlag.

Widmer, G. (2001b). Inductive learning of general and robust local expression principles. In *Proceedings of the International Computer Music Conference*. International Computer Music Association.

Widmer, G. (2001c). Using ai and machine learning to study expressive music performance: Project survey and first report. *AI Communications 14*(3), 149–162.