Learning to Play Mozart : Recent Improvements

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

Abstract

These paper describes basic research on the crossroads between machine learning and musicology. Starting from a system which is able to automatically induce multi-level tempo and dynamics models of expressive performance from a large corpus of real performances by skilled pianists, we discuss several of its shortcomings and present improvements and their empirical evaluation. In particular, we show that in a such complex domain as a concert-class musical performance, one can treat the training data as noisy. Applying a standard machine learning technique for noise handling indeed significantly improve the results. We also discuss the major drawback of standard propositional k nearest neighbor algorithm in case of learning mutually dependent concepts on different levels of resolution and present our solution to these problems by introducing a new relational instance-based learning algorithm. It turns out that it is indeed able to overcome some of the weaknesses of its propositional counterpart.

1 Introduction

The work described in this paper is further 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 [13; 14]. 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 extend 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 [5] for an excellent overview).

In previous work, we showed a hybrid system which was able to combine a predictive model of musical expression at higher levels of the musical structure with a note-level rule model [16]. The first part of the system is an instance-based learner, which recognizes performance patterns at different levels of the hierarchical musical structure (level of phrases) and learns to apply them to new pieces. The learning algorithm itself is a straight-forward k-nearest neighbor algorithm. For the prediction of local timing and dynamics effects we used a new rule learning algorithm [17], which succeeds 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 [15].

Although the approach seems well grounded from the viewpoint of musicology (human performers also complement 'shaping' of higher-level musical structures with local decisions), the first experiments were rather disappointing. In these paper we explore the shortcomings of the system together with possible improving solutions. In particular, we show that treating the training data as noisy and applying a standard technique for noise handling significantly improves the results. We also discuss a major drawback of a propositional approach to learning expressive shapes of *hierarchically nested* phrases and present our current work on a novel $relational$ instance-base learner, which indeed succeeds in achieving musically more sensible results.

This paper is organized as follows: Section 2 recalls the architecture of our hybrid approach. First experiments with a huge corpus of 'real-world' data are reproduced in section 3. In these section we also discuss major drawbacks of the system and give improvement proposals together with their empirical evaluation. In the section 4 we present a new step into the world of inductive logic programming and introduce a new relational instance-based algorithm, which seems to overcome some of the shortcomings of its propositional counterpart. Conclusion and directions for a future work are given in section 5.

2 Learning Multi-level Timing and Dynamics Strategies

The aim of our work is the automatic induction of multi-level models of tempo and dynamics strategies from large amounts of real performances by high-class pianists. 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. Phrase structure analysis is currently done by hand, as no reliable algorithms are available for this task.

Our system is a mixed approach which unifies the learning of expressive 'shapes' at the phrase-level together with local timing and dynamics effects predicted by a note-level rule model. Various problems arise: obtaining meaningful *training instances* for both, phrase-level and note-level learners, learning at both levels, and combining the predictions of both learners to produce computer-generated expressive performances. The following is a brief recapitulation of our basic approach, as previously described in [16].

2.1 Deriving the training instances: Multilevel decomposition of performance curves

Given an expression (dynamics or tempo) curve, the system 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 [10; 9; 6]. 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.

2.2 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 [2] 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 phrases. 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.

In the first experiments we used one nearest neighbor for shape prediction. The way of improving results by use of general k-NN (with $k > 1$) is investigated and discussed in section 3.4 below. 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.

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.

2.3 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 [17], 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).

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.

2.4 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.

3 Systematic Experiments

3.1 The data

In the following, we report about a systematic evaluation of our learning system. 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 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 multi-level phrase structure analysis of the musical score, which was essential for our phrase-level learning, was carried out manually by a musicologist. Phrase structure was marked at four hierarchical levels, although in some of the experiments we did not use all of them (see below).

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).

3.2 Learning with one nearest neighbor

In this section we present results of our system, in which our phrase-level learning part was set to perform one nearest neighbor for prediction. This is a recapitulation from [16]. Basically, a *leave-one-piece-out* cross-validation experiment was carried out. 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 shapes along with note-level rules 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 (i.e., curve, which shows what the pianist actually does). MSE particularly punishes curves that produce a few extreme 'errors'. 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. The last line ($WMean$) shows the weighted mean performance over all pieces (individual results weighted with the relative length of the pieces).

Overall results produced with one nearest neighbor look rather disappointing. 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 the learner produces encouraging results, in 11 out of 16 cases for MSE and in 12 out of 16 for MAE is learning better then doing nothing. Tempo on the other hand seems rather unpredictable, at least with these learner setup: only in 5 (MSE) and 3 (MAE) cases did learning produce an improvement over no learning. The correlations also vary between 0.77 (kv280:3:1, dynamics) and only 0.17 (kv332:2, tempo).

Averaging over all 16 experiments shows the same behavior, dynamics seems learnable at least to some extent (the weighted relative errors being RMSE = 1.04 and RMAE = 0.95 respectively), while tempo seems unpredictable (all relative errors are above 1.0).

3.3 Homogeneous training sets

The results presented with the learner setup discussed in the last section were rather disappointing, even keeping in mind that artistic performance of complex music like Mozart piano sonatas is certainly not entirely predictable phenomenon. One reason for a such poor performance is our current oversimplified phrase representation (discussed in the section 2.2), which certainly lacks some features, that are essential for 'understanding' of musical phrase content.

For instance, it is known in musicology that absolute tempo has quite an impact on what performance patterns sound acceptable. One way of dealing with that problem would be to include such an attribute in the phrase representation (possibly with higher weight than other attributes). The problem can also be avoided by separating the training pieces into fast and slow ones and learning in each of these sets separately. That was the procedure of our next experiment. The results in terms of wins/loses are shown in table 3. The results from first experiment (table 2) also summarized in these terms are shown in the first row. Although there is an obvious overall improvement, tempo domain remains a problem, with only 7 wins out of 16 cases.

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

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 opposite. First line: piece-level cross validation over all pieces; second line: learning on fast and slow pieces separately

3.4 Is artistic performance a noisy domain?

In our experiments so far, we used one nearest neighbor for expressive shape prediction (i.e. we borrow the shape coefficients from the most 'similar' phrase). Given that fact together with the fact that we evaluate our learner by comparing produced curves with the pianists real ones (on the same piece), we implicitly assume that the pianist always plays the

	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	.1076	.0480	.2719	.1751	.7454	.0367	.0390	.1300	.1303	.3166
kv282:1:2	.0865	.0508	.2420	.1759	.6887	.0278	.0479	.1142	.1571	.2560
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	.0314	.1633	.1311	.6061	.0225	.0216	.1024	.1083	.4260
kv283:3:2	.0424	.0405	.1688	.1466	.5255	.0256	.0261	.1085	.1130	.2961
kv332:2	.0919	.0824	.2554	.2328	.5599	.0286	.0436	.1110	.1529	.1684
WMean:	.0486	.0506	.1757	.1662	.5584	.0282	.0332	.1108	.1285	.3192

Table 2: Piecewise results of the cross-validation experiments with phrase-level learner performing one nearest neighbor for prediction and note-level learner enabled. The phrases were marked at four hierarchical levels. Measures subscripted with D refer to the 'default' (mechanical, inexpressive) performance, those with L to the performance produced by the learner.

same passages (exactly) the same way. This assumption is not grounded since it is well known in musicology that the performers play even the same parts of pieces, e.g. repeats, (slightly) differently. Transformed to our learning problem, we can look at the expressive shape diversities, even within highly 'similar' phrases, as a sort of noise in our training set.

Consequently, we can use well developed techniques for noise handling in machine learning. Indeed, taking in account more than one neighbor is a procedure which makes k-NN particularly suited for noisy data. In order to test if we can treat our performance learning problem as a 'usual' noisy one, we conducted experiments with k = 1,2,3,5 and 10 neighbors. The learned shapes were simply the averages of all neighbor-shapes [1].

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. Trying not to confuse our learner with maybe unnecessary large phrases at the highest phrase level, we ignored them and repeated the same series of experiments only at the lower three phrase levels.

In all experiments described above we wanted to exam our phrase level learning approach and thus disabled our note-level learning algorithm. Investigating the question if learning of note-level rules from $residuals$ really improves the overall results, we carried out all series of experiments described above, yet with our PLCG algorithm switched on. The results of this bundle of experiments are given in Figure 2 and Table 4. Every sub-figure shows weighted mean for one type of the measures introduced in section 3.2 (i.e. last row of table 2) achieved with the learner setups described above. Horizontal lines show the 'default' weighted errors, i.e. those which would be produced by perfectly mechanical performances.

The first look at the results confirms what the former experiments already revealed: in all experiment setups, tempo is much less predictable than dynamics. In dynamics, all but one (4 levels, k = 1 NN, no residuals, figure 2 (a)) learner setups produce errors below the 'default' error lines for both MSE (a) and MAE (b). Completely different by tempo: few of our learners succeed in producing errors smaller than the absolutely mechanical performance would in the case of MSE (setups with k = 5 and k = 10, (d)) and no one in the case of MAE (e).

The experiments also seem to validate the idea of handling the data as noisy by taking more than one nearest neighbor. The learner performance gets 'monotonically' better, given more nearest neighbors, reducing both types of errors for both dynamics and tempo and increasing the correlation for dynamics ((a)-(e)). The only exception is correlation in tempo (f), which drops in setups with more neighbors [2]. The results also confirm our suspicion that the highest-level phrases were not really musically realized by our pianist: the performance of our system is in both dimensions systematically better if it learns shapes on three phrase levels only (reducing both types of errors and increasing the correlation).

Careful examination of figure 2 also reveals one further point: Errors made with the local-level rule learning switched on are almost identical to the learning setup without it-in the case of dynamics ((a) and (b)). Not so by tempo, were the errors achieved with note-level learning drop systematically for all k ((d) and (e)). It is even more dramatical in the case of correlation (notice the big leap in tempo correlation (f) between

[1] Technically, we did not average the shapes themselves, but the triples of polynomial coefficients. The equivalence between these two procedures can be shown in a one-step proof.

[2] This can be explained by the fact that averaging over more shapes the learner tends to reproduce fewer of the local pianist decisions, which are crucial in the tempo domain.

(a) (d) (b) (e) (c) (f)

Figure 2: Different types of performance-measures achieved with various learning setups. The x-axis indicates the number of neighbors used. Horizontal lines show the baseline errors, which would be produced by perfectly mechanical performances. (a) MSE for dynamics; (b) MAE for dynamics; (c) correlation for dynamics; (d) MSE for tempo; (e) MAE for tempo; (f) correlation for tempo.

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

Table 4: Summary of wins vs. losses (piecewise) between learning and no learning for various numbers of neighbors and phrase levels (note-level learner enabled).

setups without the PLCG learner and those with it). This observation has implications for musicology, since it suggests that tempo is a much more 'local' phenomenon than dynamics. It is also in accordance with our previous research, where we showed that quadratic functions are not as reasonable a model class for expressive timing as it has hitherto been believed [16; 11] [3].

Table 4 shows the results of the above described experiments in terms of wins/losses (note-level learner enabled). Highlighted are those learners, whose learning performance beats no learning for every error definition. For the best learner (3 levels of phrasing and 10 nearest neighbors), learning beats no learning even in the tempo dimension. It is also the best learner in terms of numeric error: Its weighted tempo-MSE is significantly below the 'default-performance' error line, and it is the only one getting close to the baseline in the case of MAE (figure 2, (d) and (e)).

4 Logic in the Music Domain

In the previous section we reported about experiments with various setups of phrase-level learners combined with a note-level rule learning algorithm. The obvious drawback of the first part of the system is the current propositional attribute-value representation used to characterize phrases, which does not supply the learner with sufficient information about the internal structure of the phrases. Another related problem, even more serious, is that phrasal shapes are predicted individually and independently of the shapes of related phrases, i.e., phrases that contain the current phrase, or are contained by it. Obviously, this is a major limitation, since the human performers also 'shape' phrases dependent of the musical $context$ of the current phrase, i.e. the expressive shapes applied at different levels of 'abstraction' are highly dependent.

In the current work, we try to overcome both these limitations by introducing the dependency information between phrases via a first-order logic representation and also developing the appropriate ILP algorithm with a $context$-$dependent$ similarity measure. Since the preliminary results with the new ILP algorithm are encouraging, we are going to reproduce them here. We will also describe major ideas behind the algorithm, without bothering the reader with the formal logic. For the overview of the first-order logic in general and ILP in particular see [1; 3; 7; 8].

As stated before, the major problem of the propositional approach is the lack of coordination between predicted shapes of hierarchically nested phrases: One incorrectly chosen shape (e.g. the upward $\bigcup$ shape instead of downward $\bigcap$ shape) can completely ruin the musical acceptability of a passage, which would otherwise sound entirely musical. The new relational data representation allows us to determine which phrases are 'descendants' of the current phrase, and which are its 'ancestors'. The algorithm relies-like the propositional one-on the well defined similarity measure between two phrases, with the important difference that the similarity between two phrases now depends on the similarity of their attributes' values and the similarity of the phrases $related$ to them (phrase $context$). The similarity of the related phrases in turn depends on the attribute values of these phrases and their relation to other phrases and so on. The algorithm also takes into account the order of relationship between phrases, allowing it to model the natural decay of influence of phrases further 'away' from the phrases, whose similarity we are interested in. Defining the similarity between phrases in this way, we hope to achieve mutual dependency between predicted shapes: the similarity of two higher-level phrases is partly caused through the similarity of the lower-level phrases they contain, and vice versa.

Table 5 shows the experimental results of the new relational approach in terms of wins/losses between learning and no learning. For comparison, the results of the propositional algorithm are also given. In all cases we used one nearest neighbor for prediction of phrase-level expressive shapes with rule-level learning algorithm enabled.

The overall performance achieved by the relational algorithm is considerably better than the performance of the propositional one (although the algorithm is yet far from being completely developed). Interesting is the improvement in the tempo dimension (for both MSE and MAE) with taking just one nearest neighbor for prediction. We did not produce detailed piecewise results of the new approach, but some very high correlations between the learned curves and the pianist's actual ones in the tempo dimension should be noted, e.g. a

[3] On the other hand we also showed that the hierarchically nested quadratic functions fit the data almost perfectly in the dynamics-dimension.

	dynamics		tempo	
	MSE	MAE	MSE	MAE
prop.,4 levels	12+/4-	13+/3-	7+/9-	4+/12-
prop.,3 levels	12+/4-	14+/2-	7+/9-	6+/10-
relat.,3 levels	13+/1=/2-	15+/1-	8+/8-	8+/8-

Table 5: Summary of wins vs. losses between learning and no learning for the propositional (with phrases marked at four and three levels) and the new relational approach (phrases marked at three levels). In all cases we used one nearest neighbor for prediction. Rule-based learning was enabled.

correlation of 0.68 for kv280:1:1 [4]. On the other hand, it still performs poorly on same pieces, especially on those which are very different in character from all other pieces in the training set, which puts the learner in front of a very difficult task (e.g. correlation of 0.24 for kv332:2, tempo).

5 Discussion and Future Work

In this paper we presented a two-level system for learning both phrase-level and note-level timing and dynamics strategies for expressive music performance. Various ways of improving phrase-level learning part were introduced. Systematic experiments showed that treating the high-class performance data as noisy and applying standard machine learning techniques for noise handling can significantly improve the results. The obvious limitations of the attribute-value approach for learning hierarchically nested phrase-level shapes were also discussed, together with a new step into the world of inductive logic programming. The preliminary results with a new algorithm suggest that relational instance-based learning can indeed overcome some of the limitations of its propositional counterpart.

However, a lot of space for improvements still remains. The new algorithm is in an experimental stage and should be further improved. E.g., in the results we presented here, the weights in the similarity measure which mirror the importance decay of related phrases further away from the phrase in question are set manually without any tuning. Adjusting these weights automatically (data-driven) might not only bring a further performance gain, but also give certain insight into the domain. It would be also interesting to see if taking in account more than one neighbor in the case of the relational learner would bring such a dramatic improvement as it did in the case of the propositional one. We currently investigate this question. It is also well known that attribute weighting is an important issue in propositional instance based learning [12], and it has been argued that it will be even more important for the relational IBL [4]. Further lines of investigation on our relational learner will thus concentrate on automatical attribute selection and weighting.

On the other hand, the future work on the context-sensitive similarity measure could also have an impact on musicology. If we succeeded in developing a reliable measure for the similarity between phrases, we could try to empirically prove, that a high-class concert pianist plays similar phrases in similar ways (by trying to show high correlation between expressive shapes for those phrases for which our algorithm suggest high similarity). One could also turn the question around and take the correlation between expressive shapes for those phrases which are suggested to have high similarity as a reliability proof of our similarity measure [5].

A general problem with nearest neighbor 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. Alternative learning algorithms will have to be investigated.

[4] Such a high correlation for a $learned$ tempo curve is even more surprising taking into account the fact that kv280:1:1 is a fairly long piece with over 400 melody notes and that trying to *optimally approximate* the tempo curve by four levels quadratic functions gives a correlation of 0.71 on the same piece.

[5] For sure, it would be more reliable if we had an objective judgment about similarity between two phrases from some independent third party, but there is no procedure known in the musicology which would quantify similarity between phrases

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] Cheng, S.H.N. and Ronald de Wolf (1997). *Foundations of Inductive Logic Programming*. Berlin: Springer Verlag.

[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] Emde, D. and Wettschereck, D. (1996). Ralational Instance-Base Learning. In *Proceedings of the Thirteen International Conference on Machine Learning (ICML'96)*, pages 122-130. Morgan Kaufmann, San Mateo.

[5] Gabrielsson, A. (1999). The Performance of Music. In D. Deutsch (ed.), *The Psychology of Music (2nd ed.)*, 501–602. San Diego, CA: Academic Press.

[6] 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.

[7] Muggleton, S. (1992). Inductive Logic Programming, *The MIT Encyclopedia of the Cognitive Sciences (MITECS)*. MIT press.

[8] Muggleton S (1995). Inverse Entailment and Progol. *New Gen. Comput.*, 13:245-286.

[9] 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.

[10] Todd, N. McA. (1992). The Dynamics of Dynamics: A Model of Musical Expression. *Journal of the Acoustical Society of America* 91, 3540–3550.

[11] Tobudic, A. and Widmer, G. (2003). Playing Mozart Phrase By Phrase. Submitted to *International Conference on Case-Based Reasoning (ICCBR'03)*.

[12] Wettschereck, D., Aha, D. and Mohri, T. (1996). A Review of Feature Weighting Algorithms. *Artificial Intelligence Review Journal*, 1996.

[13] Widmer, G. (2001). Using AI and Machine Learning to Study Expressive Music Performance: Project Survey and First Report. *AI Communications* 14(3), 149-162.

[14] 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.

[15] Widmer, G. (2002). Machine Discoveries: A Few Simple, Robust Local Expression Principles. *Journal of New Musical Research* 31(1).

[16] Widmer, G. and Tobudic, A. (2003), Learning Mozart by Analogy. *Journal of New Musical Research* (in press).

[17] Widmer, G. (2003). Discovering Simple Rule in Complex Data: A Meta-learning Algorithm and Some Surprising Musical Discoveries. *Artificial Intelligence* (in press).