Discovering Strong Principles of Expressive Music Performance with the PLCG Rule Learning Strategy

Gerhard Widmer

Dept. of Medical Cybernetics and Artificial Intelligence, University of Vienna, and
Austrian Research Institute for Artificial Intelligence, Vienna
gerhard@ai.univie.ac.at

Abstract. We present a new rule learning algorithm named PLCG — a kind of ensemble learning method — that can find simple, robust partial theories (sets of classification rules) in complex data where neither high coverage nor high precision can be expected. The motivating application problem comes from an interdisciplinary research project that aims at discovering fundamental principles of expressive music performance from large amounts of complex real-world data (measurements of actual performances by concert pianists). It is shown that PLCG succeeds in finding some surprisingly simple and robust performance principles, some of which represent truly novel and musically meaningful discoveries. A more systematic experiment shows that PLCG learns significantly simpler theories than more direct approaches to rule learning, while striking a compromise between coverage and precision.

1 Introduction

The research described in the present paper is part of a large, long-term interdisciplinary research project situated at the intersection of the scientific disciplines of Musicology and AI [12]. The goal is to use intelligent data analysis methods to study the complex phenomenon of *expressive music performance*. We want to understand what great musicians do when they interpret and play a piece of music, and to what extent an artist's musical choices are constrained or 'explained' by (a) the structure of the music, (b) common performance practices, and (c) cognitive aspects of music perception and comprehension. Formulating formal, quantitative models of expressive performance is one of the big open research problems in contemporary empirical musicology. Our project develops a new direction in this field: we use *inductive machine learning* to discover general and valid expression principles from (large amounts of) real performance data.

The purpose of this research is *knowledge discovery*. We search for simple, general, interpretable models of aspects of expressive music performance (such as tempo and expressive timing, dynamics, articulation). To that end, we have compiled what is most probably the largest set of performance data (precise measurements of timing, dynamics, etc. of real musical performances) ever collected in empirical performance research. Specifically, we are analyzing large

L. De Raedt and P. Flach (Eds.): ECML 2001, LNAI 2167, pp. 552–563, 2001.

sets of recordings by highly skilled concert pianists, with the goal of discovering explainable patterns in the way the music is played.

The work described in this paper represents a first major step towards this goal. In section 2, we first explain basic concepts of expressive music performance, and then report on some problems encountered when some standard machine learning algorithms were applied in a straightforward way. These experiences prompted us to develop a new, general-purpose rule learning algorithm called PLCG, which is described in section 3. The main purpose of PLCG is to find simple, robust theories (sets of classification rules) in complex data where neither high coverage nor high precision can be expected. PLCG achieves this by learning multiple theories via some standard rule learning algorithm, and then combining these theories into one final rule set via clustering, generalization, and heuristic rule selection. Section 4 demonstrates the potential of the approach by describing some extremely simple and general performance principles and rule sets discovered by PLCG — some of the learned rules represent truly novel and musically meaningful discoveries. Then, a systematic experiment is described that compares PLCG's performance to two more 'direct' rule learning methods. The results indicate that PLCG finds more compact theories than the simpler rule learners, while striking a compromise between generality and precision.

2 The Target: Expressive Music Performance

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. Rather, they speed up at some places, slow down at others, stress certain notes or passages by various means, and so on. The expressive nuances added by an artist are what makes a piece of music come alive. The most important dimensions available to a performer (a pianist, in particular) are tempo, dynamics (loudness variations), and articulation (the way successive notes are connected).

Expressive variation is more than just a 'distortion' of the original (notated) piece of music. In fact, the opposite is the case: the notated music score is but a small part of the actual music. Not every intended nuance can be captured in a limited formalism such as common music notation, and the composers were and are well aware of this. The performing artist is an indispensable part of the system, and expressive music performance plays a central role in our musical culture. That is what makes it a central object of study in the field of musicology.

Our approach to studying this complex phenomenon is to collect large corpora of performance data (i.e., exact measurements of onset and offset times and loudness of each note as played in a performance), and to apply inductive learning algorithms to find models that compactly characterize various classes of situations that are treated in a similar way by the performer (such as 'situations where the performer slows down' vs. 'situations where s/he speeds up'). At the moment, we limit ourselves to classical piano music.

In a first major study, we investigated the feasibility of inducing performance rules at the most basic musical level: the level of individual notes. The goal was to discover rules that predict how individual notes will most likely be played by a pianist (e.g., louder or softer than their predecessor). In a first suite of experiments [11], we succeeded in showing that even at that low level, there is structure in the data; learning algorithms like C4.5 [8] were able to find rule sets that predict the performer's choices with better than chance probability.

However, the improvement over the baseline accuracy was generally rather small (though statistically significant), which indicates that there are severe limits as to how much of a performer's behaviour can be explained at the note level. Moreover, the learned models were extremely complex. For instance, a decision tree discriminating between *accelerando* (speeding up) and *ritardando* (slowing down) with 58.09% accuracy had 3037 leaves (after pruning)! This is clearly not desirable if our goal is knowledge discovery.

There are good musical reasons for these difficulties; we cannot go into these here. From a machine learning perspective, the main insight is that looking for a model that completely describes the target categories is futile. Decision tree learners attempt to build a global model that fully discriminates between the members of the various classes. What we need to do instead is search for *partial* models that only explain what *can* be explained, and simply ignore those parts of the instance space where no compact characterization of the target classes seems possible. Moreover, given the nature of our data and target phenomena, we cannot expect very high levels of discriminative accuracy — we cannot assume the artist to be perfectly consistent and predictable.

In the following, we describe a rule learning algorithm named PLCG that was developed for this purpose. It will be shown that PLCG can find very simple partial models that still characterize a number of interesting subclasses of expressive performance behaviour. (Indeed, we will show that 4 simple rules are sufficient to predict 22.89% of the instances of note lengthening in our large data set, which contrasts nicely with the decision tree with 3037 leaves mentioned above.)

3 The PLCG Rule Learning Algorithm

Given the goal of learning partial models, an obvious choice is to apply rule learning algorithms of the *set covering* variety (also known as *separate-and-conquer* learners [5]), such as FOIL [7] or RIPPER [1]. These algorithms learn theories one rule at a time, in each rule refinement step selecting a literal that maximizes some measure of discrimination (e.g., information gain). A rule is specialized until a given stopping criterion (typically based on the rule's purity or precision) is satisfied, and the overall learning process stops when no more rules can be found that satisfy this purity criterion. The stopping criterion is thus the natural entry point for the user to influence the generality and precision of the induced rules. In the context of our problem, we would require rather low levels of precision. The degree of coverage of the resulting rules would then follow automatically, dictated by the data.

After some experimentation, we have chosen to pursue a more complex approach. The basic idea is to learn several models in parallel (from subsets of the data), search for groups of similar rules in these models, generalize these into summarizing rules (of varying degrees of generality), and then select those generalizations for the final model that optimize some (possibly global) user-defined criterion (which will typically be a trade-off function between coverage and precision). This strategy gives us more direct control over the overall coverage and precision of the induced models, and at the same time helps ameliorate one of the major problems of the greedy literal selection strategy of the underlying rule learner: the danger of selecting sub-optimal conditions due to the local maximization of a given discrimination measure. In this sense, our approach — let us call it the PLCG (**P**artition+**L**earn+**C**luster+**G**eneralize) strategy — is inspired by the success of *ensemble methods* in machine learning (see [2] for a good overview). The corresponding algorithm is given in more detail in figure 1.

PLCG is really a *meta-algorithm* that can be wrapped around any algorithm that learns classification rules. We are using our own implementation of a propositional FOIL-type [7] learner, with the standard information gain heuristic and with a parameterizable stopping criterion based on rule purity and mininum required rule coverage. More precisely: a single rule is grown by adding conditions until its *purity* (or *precision*) $P = p/(p+n)$ reaches or surpasses a given *minimum precision* MP_{RL}, where p and n are the numbers of positive and negative examples, respectively, covered by the rule. The outer loop of the algorithm ter-

---

Given:

- a set of training instances D
- a target concept (class) c
- a rule learning algorithm L
- a rule selection criterion C

Algorithm:

1. Separate the training examples D into n subsets D_i, $i = 1 \ldots n$ (randomly or according to a particular scheme);
2. Learn partial rule models $R_i = \{r_{ij}\}$ for class c from each of these subsets D_i separately, using the learning algorithm L.
3. Merge the rule sets R_i into one large set R: $R = \bigcup R_i$.
4. Perform a hierarchical clustering of the rules in R into a tree of clusters C_i, $i = 1 \ldots k$, of similar rules, using some hierarchical clustering algorithm and an appropriate syntactic/semantic rule similarity measure.
5. For each cluster C_i, compute the *least general generalization* of all the rules in C_i: $\hat{r}_i = lgg(\{r_{ij} | r_{ij} \in C_i\})$. The resulting tree T of rules $\hat{r}_i$ represents generalizations of various degrees of the original rules.
6. From this generalization tree T, select those rules r_i that optimize the given selection criterion C.

Fig. 1. The PLCG (**P**artition+**L**earn+**C**luster+**G**eneralize) rule learning strategy.

minates when no more rule can be found with $P \geq MP_{RL}$ and positive coverage p greater than some user-defined minimum coverage MC_{RL}.

For *rule clustering*, we use a standard bottom-up hierarchical agglomerative clustering algorithm [6] that produces a binary cluster tree, with the individual rules forming the leaves of the tree, and the root containing all the rules. The *rule similarity measure* used for clustering is simply the inverse of the number of generalization operations needed to compute the *least general generalization (lgg)* of two rules. Given our standard propositional representation of instances and rules (see 4.2 below for an example), the definition of the *lgg* is obvious.

As for the *rule selection criterion* (step 6 of the PLCG algorithm), we currently use another greedy set-covering algorithm that starts with the empty rule set and always adds the rule that has maximum purity on the as yet uncovered instances. (In fact, we use the *Laplace estimate* $L = (p+1)/(p+n+2)$, which is related to purity, but gives higher weight to rules that cover a higher number of positive examples.) Again, the selection is terminated when no rule with purity (Laplace) greater than some user-defined MP_{PLCG} and coverage greater than some minimum required coverage MC_{PLCG} can be found.

This is just one of many possible rule selection strategies. Many others are conceivable that could use different criteria for trading coverage against precision, or that might aim at optimizing other aspects of the evolving rule set (e.g., minimum overlap or a minimum number of contradictions between rules).

4 Experimental Results

4.1 Data and Target Concepts

The data used in our first experimental investigation consists of recordings of 13 complete piano sonatas by W.A. Mozart (K.279–284, 330–333, 457, 475, and 533), performed by a Viennese concert pianist on a Bösendorfer SE290 computer-monitored grand piano. The Bösendorfer SE290 is a full concert grand piano with a special mechanism that measures and records every key and pedal movement with high precision. These measurements, together with the notated score in machine-readable form, provide us with all the information needed to compute expressive variations (e.g., tempo fluctuations). The resulting dataset consists of more than 106,000 performed notes and represents some four hours of music.

The experiments described here were performed on the melodies (usually the soprano parts) only, which gives an effective training set of 41,116 notes. Each note is described by 29 attributes (10 numeric, 19 discrete) 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 etc.).

In terms of performance parameters, we are looking at (local) tempo or timing, dynamics, and articulation. We defined the following discrete target classes:

1. in the tempo dimension, a note N is assigned to class *ritardando* if the local tempo at that point is significantly ($> 2\%$) slower than the tempo at the previous note; the opposite class *accelerando* contains all cases of local speeding up;
2. in dynamics, a note N is considered an example of class *crescendo* if it was played louder than its predecessor, and also louder than the average level of the piece; class *diminuendo* (growing softer) is defined analogously;
3. in articulation, three classes were defined: *staccato* if a note was sounded for less than 80% of its nominal duration, *legato* if the proportion is greater than 1.0 (i.e., the note overlaps the following one), and *portato* otherwise; we will only try to learn rules for the classes staccato and legato.

A performed note is considered a counter-example to a given class if it belongs to one of the competing classes. (Note that due to some details of our class definitions, there will be some notes that are neither examples nor counter-examples of some concept.)

4.2 Musical Discoveries

Let us first look at some of PLCG's discoveries from a musical perspective. When run on the complete Mozart performance data set (41,116 notes) for each of the six target concepts defined above,[1] PLCG (with parameter settings $MP_{PLCG} = .7$, $MC_{PLCG} = .02$, $MP_{RL} = .9$, $MC_{RL} = .01$) selected a final set of 17 performance rules (from a total of 383 specialized rules) — 6 rules for tempo changes, 6 rules for local dynamics, and 5 rules for articulation. (Two rules were selected manually for musical interest, although they did not quite reach the required coverage and precision, respectively.) Some of these rules turn out to be discoveries of significant musicological interest. We lack the space to list all of them here (see [13]). Let us illustrate the types of patterns found by looking at just one of the learned rules:

RULE TL2:
`abstract_duration_context = equal-longer`
`& metr_strength` $\leq$ `1`
$\Rightarrow$ `ritardando`

"Given two notes of equal duration followed by a longer note, lengthen the note (i.e., play it more slowly) that precedes the final, longer one, if this note is in a metrically weak position ('metrical strength' ≤ 1*)."*

[1] In this experiment, the data were not split into subsets randomly; rather, 10 subsets were created according to global tempo (fast or slow) and time signature (3/4, 4/4, etc.) of the sonata sections the notes belonged to. We chose these two dimensions for splitting because it is known (and has been proved experimentally [11]) that global tempo and time signature strongly affect expressive performance patterns. As a result, we can expect models that tightly fit (overfit?) these data partitions to be quite different, and diversity should be beneficial to an ensemble method like PLCG.

Table 1. Fit of rule sets on training data (13 Mozart sonatas); True Positives (*TP*) = correct predictions; False Positives (*FP*) = incorrect predictions (relative to total number of positive and negative instances, respectively); Precision = $TP/(TP + FP)$.

Category	#rules	True Positives	False Positives	Precision
ritardando	4	3069/13410 (22.89 %)	1234/20551 (6.00 %)	.713
accelerando	2	397/13307 (2.98 %)	179/20550 (0.87 %)	.689
crescendo	3	1318/11629 (11.33 %)	591/18260 (3.24 %)	.690
diminuendo	3	625/9429 (6.63 %)	230/20113 (1.14 %)	.731
staccato	4	6916/22132 (31.25 %)	1089/18984 (5.74 %)	.864
legato	1	687/9256 (7.42 %)	592/31860 (1.86 %)	.537

This is an extremely simple principle that turns out to be surprisingly general and precise: rule TL2 correctly predicts 1,894 cases of local note lengthening, which is 14.12% of all the instances of significant lengthening observed in the training data. The number of incorrect predictions is 588 (2.86% of all the counterexamples). Together with a second, similar rule relating to the same type of phenomenon, TL2 covers 2,964 of the positive examples of note lengthening in our performance data set, which is more than one fifth (22.11%)! It is highly remarkable that one simple principle like this is sufficient to predict such a large proportion of observed note lengthenings in a complex corpus such as Mozart sonatas. This is a truly novel (and surprising) discovery; none of the existing theories of expressive performance were aware of this simple pattern.

A few other interesting rules were discovered, such as two pairs of timing and articulation rules that nicely characterize the pianist's consistent treatment of certain types of melodic leaps and rhythmic patterns. These discoveries and their relation to other theories of expressive performance in the musicological literature are discussed in [13].

4.3 Fit and Generality of Discovered Principles

As our primary goal is knowledge discovery, we are first of all interested in how much of the given (training) data is explained by the learned model — in other words, how well the induced models capture the pianist's performance style. Thus, contrary to more 'standard' machine learning applications, the degree of *fit* on the training set is relevant. (Of course, we will also be looking at *generalization accuracy* on unseen data — see below). As a quantification of fit, we measure the *coverage* (i.e., the number of positive examples correctly predicted) and the *precision* (the proportion of predictions that are correct) of the rule sets on the training data, separately for each prediction category. Table 1 gives the results.

A detailed discussion of the results and their musical interpretation is beyond the scope of this paper. Generally, it turns out that certain sub-classes of note lengthening (local *ritardando*), *staccato*, and to a lesser extent the local dynamics variations (*crescendo* and *diminuendo*) are surprisingly well predictable, and with extremely few (and simple) rules. On the other hand, categories like *accelerando*

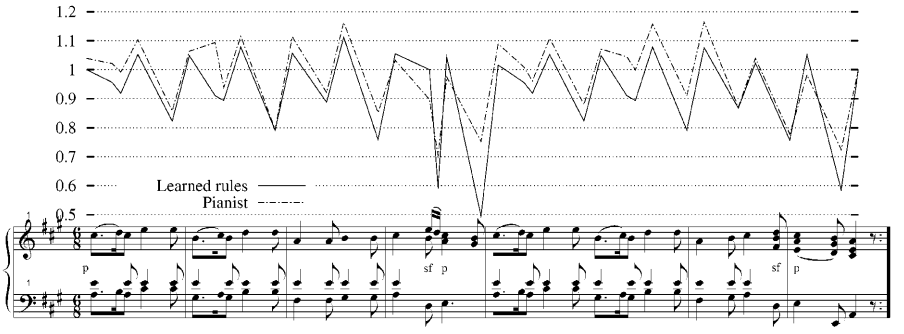

Fig. 2. Mozart Sonata K.331, 1st movement, 1st part, as played by pianist and learner. The curve plots the relative tempo at each note — notes above the 1.0 line are shortened relative to the tempo of the piece, notes below 1.0 are lengthened. A perfectly regular performance with no timing deviations would correspond to a straight line at $y = 1.0$

and *legato* seem more difficult to predict — at least at the level of individual notes. Uncovering the reasons for this will require more specialized investigations.

To give the reader an impression of just how effective a few simple rules can be in predicting a pianist's behaviour in certain cases, Figure 2 compares the tempo variations predicted by our rules to the pianist's actual timing in a performance of the well-known Mozart Sonata K.331 in A major (first movement, first section). In fact, it is just two simple rules (one for note lengthening (*ritardando*), one for shortening (*accelerando*)) that produce the system's timing curve.[2]

The next question concerns the *generality* of the discovered rules. How well do they transfer to other pieces and other performers? To assess the degree of *performer-specificity* of the rules, we tested them on performances of the same pieces, but by a different artist.[3] The test pieces in this case were the Mozart sonatas K.282, K283 (complete) and K.279, K.280, K.281, K.284, and K.333 (second movements), performed by the renowned conductor and pianist Philippe Entremont, again on a Bösendorfer SE290. The results are given in Table 2.

Comparing this to Table 1, we find no significant degradation in coverage and precision (except in category *diminuendo*). On the contrary, for some categories (*ritardando*, *crescendo*, *staccato*) the coverage is higher than on the original training set. The discriminative power of the rules — the precision — remains roughly at the same level. This (surprising?) result testifies to the generality of the discovered principles (and the merits of the PLCG rule discovery method).

[2] To be more precise: the rules predict whether a note should be lengthened or shortened; the *precise numeric amount* of lengthening/shortening is predicted by a *k-nearest-neighbor* algorithm (with $k = 3$) that uses only instances for prediction that are covered by the matching rule, as proposed in [9] and [10].

[3] The true *generalization accuracy* of the rules on music of the same style will be tested on recordings of *additional* Mozart sonatas by P.Entremont (i.e., pieces not used in training). At the time of writing, the performance measurements are still being prepared for analysis (unfortunately, that takes several person weeks!).

Table 2. Prediction results on test data (Mozart performances by P.Entremont).

Category	#rules	True Positives	False Positives	Precision
ritardando	4	596/2036 (29.27 %)	242/3175 (7.62 %)	.711
accelerando	2	90/2193 (4.10 %)	45/3013 (1.49 %)	.667
crescendo	3	210/1601 (13.12 %)	87/3055 (2.85 %)	.707
diminuendo	3	53/1598 (3.32 %)	45/2725 (1.65 %)	.541
staccato	4	861/2192 (39.28 %)	228/3996 (5.71 %)	.791
legato	1	131/2827 (4.63 %)	57/3361 (1.70 %)	.697

Table 3. Prediction results on test data (Chopin performances by 22 pianists).

Category	#rules	True Positives	False Positives	Precision
ritardando	4	1752/2537 (69.06 %)	327/2988 (10.94 %)	.843
accelerando	2	1472/2767 (53.20 %)	110/2746 (4.01 %)	.930
crescendo	3	601/2392 (25.13 %)	285/2578 (11.06 %)	.678
diminuendo	3	0/2249 (0.00 %)	0/2784 (0.00 %)	—
staccato	4	950/2932 (32.40 %)	166/2802 (5.92 %)	.851
legato	1	17/2011 (0.85 %)	27/3723 (0.73 %)	.386

Another experiment tested the generality of the discovered rules with respect to *musical style*. They were applied to pieces of a very different style (Romantic piano music), namely, the Etude Op.10, No.3 in E major (first 20 bars) and the Ballade Op.38, F major (first 45 bars) by *Frédéric Chopin*, and the results were compared to performances of these pieces by 22 Viennese pianists. The melodies of these 44 performances amount to 6,088 notes. Table 3 gives the results.

This result is even more surprising. *Diminuendo* and *legato* turn out to be basically unpredictable, and the rules for *crescendo* are rather imprecise. But the results for the other classes are extremely good, better in fact than on the original (Mozart) data which the rules had been learned from! The high coverage values, especially of the tempo rules, are remarkable. Remember also that the data represent a mixture of 22 different pianists. When looking at how well the rules fit individual pianists, we find that some of them are predicted extremely well (e.g., pianist #15: ritardando: $TP = 89/122$ (72.95%), $FP = 4/129$ (3.10%), $\pi = .957$; accelerando: $TP = 71/120$ (59.17%), $FP = 3/132$ (2.27%), $\pi = .959$). We are currently preparing recordings of a larger variety of Chopin pieces, which will permit more extensive investigations into the rules' general validity.

4.4 PLCG vs. Direct Rule Learning: A First Systematic Study

The above results show that PLCG can discover general and robust rules in complex data. To establish its advantages, if any, over the underlying rule learning algorithm, more systematic comparative experiments are needed. A first step in this direction is described here. PLCG, with parameter settings $MP_{RL} = .9$,

Table 4. Summary of 60 cross-validation results; NR = total number of rules (summed over all 60 data sets × 5 folds) and average number of rules per learning run ($NR/60/5$); TP = true positives, FP = false positives; π = precision.

	RL0.9	RL0.7	PLCG0.9/0.7
TP	12731/62017 (20.53 %)	28358/62017 (45.73 %)	18767/62017 (30.26 %)
FP	2180/102136 (2.13 %)	11710/102136 (11.47 %)	6551/102136 (6.41 %)
π	0.854	0.708	0.741
NR	2475 (8.25)	4094 (13.65)	1707 (5.69)

$MC_{RL} = .01$ for the rule learner and $MP_{PLCG} = .7$, $MC_{PLCG} = .04$ for rule selection, was compared to two versions of the base-level separate-and-conquer learner in a set of 60 cross-validation experiments: learner RL0.9 learns rules directly from the data with a required purity level of $RP = 0.9$ (i.e., RL0.9 is exactly the same algorithm as the one used within PLCG's inner loop); RL0.7 uses the more relaxed minimum purity threshold $RP = 0.7$, which corresponds to the precision level used by PLCG in its rule selection phase. The purpose of the experiment was to study whether PLCG's multiple rule learning + generalization + selection approach yields any advantage over learning rules directly from the data with the corresponding parameter settings.

60 different experimental data sets were produced by partitioning our 41,116 performed and classified notes according to the general *tempo* (slow vs. fast) and the *time signature* (3/4, 4/4, etc.) of the Mozart sonata segments they belong to. This resulted in 10 training sets each for the 6 target concepts *accelerando*, *ritardando*, *crescendo*, *diminuendo*, *staccato*, and *legato*.

On each of these 60 data sets, the three learning algorithms were compared via a 5-fold (paired) cross-validation. Within each CV run, RL0.9 and RL0.7 were applied to the combined data from the four training folds, while PLCG used the four folds to learn four separate rule sets (via RL0.9) that were then combined, generalized, and selected from. We lack the space to present the full results table with 60 × 18 entries here. Table 4 gives a summary of the results.

The results clearly reflect the expected trade-off: learning with a tighter precision threshold for individual rules (RL0.9) yields theories with higher precision, but lower coverage than learning with a lower required precision (RL0.7). PLCG, with its mixture of precision thresholds (high in the individual rule learning runs, lower in the rule selection phase) figures somewhere in between: its coverage is higher than RL0.9's and lower than RL0.7's. Conversely, it reaches a precision lower than RL0.9's and higher than RL0.7's.

The interesting result is that PLCG achieves this with significantly *fewer rules* than *either* of the two base-level learners, RL0.9 and RL0.7. In other words, PLCG covers more instances than RL0.9 with fewer (more general) rules, while still retaining a higher precision than RL0.7, which used the same precision threshold MP in its search for rules. That is indeed the desired kind of behaviour for our application, where the goal is to discover simple, general, robust (partial) theories that can be presented to and discussed with musicologists.

In general, how much precision one is willing to sacrifice for how much coverage and theory simplicity, and vice versa, will depend on the particular application. The important advantage of the PLCG approach is that it makes it easy to explore and control this trade-off via different *rule selection strategies*. In fact, one can perform the rule learning and clustering steps once and then apply any number of different rule selection algorithms on the resulting rule cluster tree.

5 Discussion

To summarize, we have presented a rule (meta-)learner that learns simple theories from complex data and offers a natural mechanism for exploring the coverage/precision/complexity tradeoff, and we have shown that PLCG is able to make interesting and surprising discoveries in a complex real-world domain.

PLCG's bottom-up generalization of classification rules is reminiscent of Domingos' RISE algorithm [3], which performs a bottom-up generalization into more and more general rules, starting from the individual training instances. What the two have in common is the idea to learn rules only for those parts of the instance space where the concepts can be easily characterized. RISE caters for the remaining space with instance-based learning, while PLCG simply ignores it (because its focus is on finding comprehensible characterizations of sub-classes of the target). On the other hand, PLCG makes it easy to explore alternative (and arbitrarily complex) strategies for rule combination and selection, which is possible because it constructs an explicit tree of rules of varying generality. Another significant difference is PLCG's use of multiple models to arrive at a more stable theory.

More directly related to PLCG are the so-called *ensemble methods* [2], which learn multiple models from subsets or modified versions of the training data, with one or several learning algorithms, and combine the resulting classifiers in some way. While the majority of known ensemble methods like bagging, boosting, stacking, etc. only combine the *predictions* of the classifiers, there are a few algorithms that try to combine the resulting multiple *models* into one coherent, comprehensible model. A prime example of this is CMM [4], a meta-learner that combines multiple models into a single theory by applying a learning algorithm to (artificially generated) training examples labeled by the n learned base models. According to the reported experimental results, CMM usually achieves higher accuracy than the base learner (C4.5RULES), but the models produced by CMM are typically 2-6 times more complex than the base learner's. In [4], it is also suggested that the accuracy/complexity tradeoff could be handled via the meta-learner's pruning parameters; again, we think PLCG's way of explicitly addressing this tradeoff via a selection procedure that can select from a set of alternative rules (of various degrees of generality) is preferable.

So far, we have compared PLCG only to one rather simple rule learner. Of course, it should be systematically compared also to more sophisticated learners like RIPPER [1]. It will be interesting to see if PLCG's ability to find simple theories matches the effect of RIPPER's complex pruning strategy. On the other hand, whether or not PLCG is better is the wrong question to ask. PLCG is a

meta-learner; it could equally well be wrapped around RIPPER, thus potentially combining the advantages of the two.

Future research will focus on trying to get a better understanding of PLCG's characteristics through systematic experiments with different underlying rule learners, and different application domains.

Acknowledgements. This research is made possible by a very generous START Research Prize (project no. Y99-INF) by the Austrian Federal Government, administered by the Austrian *Fonds zur Förderung der Wissenschaftlichen Forschung (FWF)*. The Austrian Research Institute for Artificial Intelligence acknowledges basic financial support by the Austrian Federal Ministry for Education, Science, and Culture. We are particularly grateful to the pianists Roland Batik nd Philippe Entremont for allowing us to use their performances for our investigations.

References

1. Cohen, W. (1995). Fast Effective Rule Induction. In *Proceedings of the 12th International Conference on Machine Learning*. San Francisco, CA: Morgan Kaufmann.
2. Dietterich, T. G. (2000). Ensemble Methods in Machine Learning. In J. Kittler and F. Roli (Ed.), *First International Workshop on Multiple Classifier Systems*. New York: Springer Verlag.
3. Domingos, P. (1996). Unifying Instance-Based and Rule-Based Induction. *Machine Learning* 24, 141–168.
4. Domingos, P. (1998). Knowledge Discovery via Multiple Models. *Intelligent Data Analysis* 2, 187–202.
5. Fürnkranz, J. (1999). Separate-and-Conquer Rule Learning. *Artificial Intelligence Review* 13(1), 3–54.
6. Hartigan, J. (1975). *Clustering Algorithms*. Chichester, UK: John Wiley & Sons.
7. Quinlan, J.R. (1990). Learning Logical Definitions from Relations. *Machine Learning* 5, 239–266.
8. Quinlan, J.R. (1993). *C4.5: Programs for Machine Learning*. San Francisco, CA: Morgan Kaufmann.
9. Weiss, S. and Indurkhya, N. (1995). Rule-based Machine Learning Methods for Functional Prediction. *Journal of Artificial Intelligence Research* 3, 383–403.
10. Widmer, G. (1993). Combining Knowledge-based and Instance-based Learning to Exploit Qualitative Knowledge. *Informatica* 17, 371–385.
11. Widmer, G. (2000). Large-scale Induction of Expressive Performance Rules: First Quantitative Results. In *Proceedings of the International Computer Music Conference (ICMC'2000)*. San Francisco, CA: International Computer Music Association.
12. Widmer, G. (2001). Using AI and Machine Learning to Study Expressive Music Performance: Project Survey and First Report. *AI Communications* 14 (in press).
13. Widmer, G. (2001). *Machine Discoveries: Some Simple, Robust Local Expression Principles.* Submitted.