ÖEFAI Technical Report OEFAI-TR-2003-01

RECOGNITION OF FAMOUS PIANISTS USING MACHINE LEARNING ALGORITHMS: FIRST EXPERIMENTAL RESULTS

Patrick Zanon

CSC – DEI, University of Padua
Via Gradenigo 6/a, 35131 Padova, Italy
patrick@dei.unipd.it
http://www.dei.unipd.it/~patrick

Gerhard Widmer

University of Vienna and ÖFAI
Schottengasse 3, A-1010 Vienna, Austria
gerhard@ai.univie.ac.at
http://www.oefai.at/~gerhard

ABSTRACT

The paper addresses the question whether a machine can learn to identify famous performers (pianists) based on their style of playing. A preliminary study is presented where different machine learning algorithms are applied to performance data derived from Mozart sonata recordings by several famous pianists. It is shown that the algorithms learn to recognize pianists at a level better than chance, and that some pianists seem easier to recognize than others. The study identifies a number of limitations of the current approach (regarding both data and learning algorithms) and points to a variety of fruitful directions for further research.

1. INTRODUCTION

The work presented here is part of a large investigation into the use of novel computational methods for studying basic principles of expressive music performance [9]. One of the questions we study is whether and to what extent aspects of *individual artistic style* can be quantified. And one of the possible approaches to this question is to investigate whether machines can learn to distinguish and recognize different performers based on their style of playing.

Previous research has shown that this seems indeed possible, to a certain extent [7, 8]. In a study with 22 different pianists (teachers and students of the University of Music in Vienna) a new machine learning algorithm achieved a surprising level of recognition accuracy. However, that study was limited in many respects, particularly with regard to the data that were available (recordings by 22 different pianists, but only two pieces). On the other hand, the performance measurements were extremely precise, because the recordings had been made on a Bösendorfer SE290 computer-controlled piano.

The present paper describes first steps towards generalizing this research. We extend the study towards the analysis of famous world-class pianists, and work with larger collections of recordings. At the same time, that raises a data problem, because performances of famous artists are only available as audio recordings, and it is impossible to extract the same kind of exact performance information (e.g., details of timing and articulation) from these recordings. In the experiments to be reported below, performance information will be available only at an extremely crude level (essentially, only high-level tempo and loudness changes over time), and the question is whether performer identification is still possible at this level.

The paper is organized as follows: section 2 describes in detail the experimental methodology followed, including a description of the data, a characterization of the performance features extracted from the recordings, and a list of the machine learning algorithms tested. Section 3 presents preliminary experimental results that show that the learning algorithms can at least learn to identify performers at a level better than chance. The results also point to a number of problems and prompt us to identify several promising directions for further research. The next steps to be performed in this project are then detailed in section 4.

2. METHODOLOGY

2.1. The performance data

For the experiments, commercial recordings of piano sonatas by W.A. Mozart by six different concert pianists were collected, and a sizeable number of pieces were selected for performance measuring and analysis. The pieces are listed in Table 1, and the pianists in Table 2.

ID	Sonata	Movement	Key	Time sig.
kv279_1	K.279	1st mvt.	C major	4/4
kv279_2	K.279	2nd mvt.	C major	3/4
kv279_3	K.279	3rd mvt.	C major	2/4
kv280_1	K.280	1st mvt.	F major	3/4
kv280_2	K.280	2nd mvt.	F major	6/8
kv280_3	K.280	3rd mvt.	F major	3/8
kv281_1	K.281	1st mvt.	Bb major	2/4
kv282_1	K.282	1st mvt.	Eb major	4/4
kv282_2	K.282	2nd mvt.	Eb major	3/4
kv282_3	K.282	3rd mvt.	Eb major	2/4
kv330_3	K.330	3rd mvt.	C major	2/4
kv332_2	K.332	2nd mvt.	F major	4/4

Table 1: Movements of Mozart piano sonatas selected for analysis.

From the audio recordings, rough measurements characterizing the performances were obtained. More precisely, tempo and general loudness were measured at the level of the beats, by using an interactive beat tracking program [3] to find the beat in the audio signal and computing beat-level tempo changes from the varying inter-beat intervals.

ID	Name	Recording
DB	Daniel Barenboim	EMI Classics CDZ 7 67295 2, 1984
RB	Roland Batik	Gramola 98701-705, 1990
GG	Glenn Gould	Sony Classical SM4K 52627, 1967
MP	Maria João Pires	DGG 431 761-2, 1991
AS	András Schiff	ADD (Decca) 443 720-2, 1980
MU	Mitsuko Uchida	Philips Classics 464 856-2, 1987

Table 2: Pianists and recordings.

Overall loudness of the signal at these time points was extracted from the audio signal and is taken as a very crude representation of the dynamics applied by the pianists. No more detailed information (e.g., about articulation, individual voices, or timing details below the level of the beat) is available.

These sequences of measurements can be represented as two sets of performance curves — one representing beat-level tempo, the other beat-level loudness changes — or in an integrated two-dimensional way, as trajectories over time in a 2D tempo-loudness space [6]. We have developed a graphical animation tool called the *Performance Worm* [4] that displays such performance trajectories in synchrony with the music. A part of a performance as visualized by the Worm is shown in Figure 1. Note that the display is interpolated and smoothed. For the machine learning experiments reported below, only the actually measured points were used; no interpolation or smoothing was performed.

Thus, the raw data for our experiments is tempo and overall loudness values measured at specific time points in a performance (either every beat according to the time signature or, where beat tracking was performed at lower levels, at subdivisions of the beat). For each measured time point, the following is stored: t_i (absolute time in seconds), B_i (calculated tempo in bpm), L_i (loudness level measured in sone [11]), and some bit-coded flags that represent hierarchical structural information. More precisely, $ftbb$ indicates whether the current time point coincides with a beat track point (trivially true for all), a beat, or the beginning of a bar; $fs1234$ indicates four levels of phrase structure (this information was added manually by a musicologist).

The raw data so obtained had to be refined in order to be homogeneous and usable in the learning process. In fact, data usually comes from different sources, and could have been produced by different persons, sometimes with different strategies. This introduce some noise that can affect the output of the learners. Thus, some time had to be spent in cleaning of the data in order to have the most homogeneous information representation as possible.

In our case, some of the pieces were tracked at the level of the beat (as defined by the time signature), some at the half beat level. Moreover, some pieces start in different ways, according to the performers' decisions: for example, some of them start with an upbeat, and some others with two. We decided to skip all the non-common information, by sampling the pieces with higher tracking resolution, and by discarding all the non-common up beats.

Moreover, most of the players repeated some sections, while others didn't (i.e., Gould). Also in this case, we decided to discard all the non-common sections, so that the learners would work with comparable data for all the performers, and not with over-represented sections that could affect their output.

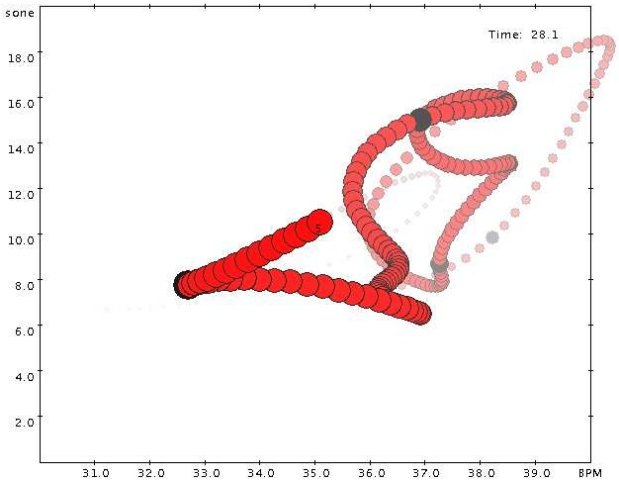

Figure 1: Snapshot of the *Performance Worm* at work: First four bars of Daniel Barenboim's performance of Mozart's F major sonata K.332, 2nd movement. Horizontal axis: tempo in beats per minute (bpm); vertical axis: loudness in *sone* [11]. Movement to the upper right indicates a speeding up (*accelerando*) and loudness increase (*crescendo*) etc. etc. The darkest points represents the current instant, while instants further in the past appear fainter.

2.2. Instances and features

Each measured time point, along with its context, is used as a *training example* for the learning algorithms. In other words, an example or *instance* for the learners is a subsegment of a tempo-loudness trajectory (see Figure 1), centered around a specific time point. Altogether, this procedure results in some 23.000 instances for all the six pianists.

The instances are represented by a set of *features* that are extracted from the raw trajectories. The features were calculated over a sliding window w_i of two bars (the context size). Thus, if in the original data there are n instances, then there will also be n windows and n sets of features. Of course, at the beginning and at the end, some of the features were calculated over a narrower window than two bars.

Notice that the number of measured points included in a window w_i may be different between pieces, since there are different tempo indications and different tempo tracking resolutions. For example, in the piece kv280_3 there is a low sampling rate (1 beat per bar), while in kv282_1 there are 8 tracked points per bar.

The sliding window method produces some redundancy in the data, since the windows were overlapping; this should be an advantage in learning. Some of the features were calculated using a window which extended beyond the boundary between two sections. This is no problem in most cases, but in some cases the two sections may be non-contiguous in the original data. We chose to allow this discontinuity in

Operation	Tempo	Loudness	Others
None	B_i	$L_i^{(1)}$	$ftbb_i$, $fs1234_i$
Average and Median	$\mu_B(w_i)$, $\nu_B(w_i)$	$\mu_L(w_i)^{(1)}$, $\nu_L(w_i)^{(1)}$	–
Standard Deviation	$\sigma_B(w_i)$	$\sigma_L(w_i)^{(1)}$	–
Min, Max and Range	$m_B(w_i)$, $M_B(w_i)$, $R_B(w_i)$	$m_L(w_i)^{(1)}$, $M_L(w_i)^{(1)}$, $R_L(w_i)^{(1)}$	–
Normalization	$b_i(w_i)$, $\sigma_b(w_i)$, $m_b(w_i)$, $M_b(w_i)$, $R_b(w_i)$	$l_i(w_i)$, $\sigma_l(w_i)$, $m_l(w_i)$, $M_l(w_i)$, $R_l(w_i)$	–
Correlation	$\Sigma_{tB}(w_i)$	$\Sigma_{tL}(w_i)$	$\Sigma_{BL}(w_i)$
Directness	$\Delta_{tB}(w_i)$	$\Delta_{tL}(w_i)$	$\Delta_{BL}(w_i)$
Derivative	$M\delta_B$, $\mu\delta_B$	$M\delta_L$, $\mu\delta_L$	–

Table 3: Complete set of features extracted from the data for each instance. $^{(1)}$indicates that the corresponding feature must not be used by the learner (if it were, it would trivially reveal some of the performers, on the basis of the CD recording level).

the data, because the number of affected instances is negligible. Thus, it was not necessary to 'instruct' the code to recognize section boundaries.

Caution had to be taken with some of the extracted performance information; In particular, the features derived from loudness had to be filtered in some way, because they can trivially reveal some of the performers. For example, Gould's CD recordings are older (1967) than the others (1980-1991), resulting in a significantly lower recording level. That would permit the learners to detect this famous performer simply by loudness difference. Thus, a normalization in the data was carried out in order to mask this information: all the relevant loudness-derived features were normalized using the actual window average $\mu_L(w_i)$.

For each instance of the original raw data, the following features were computed both for tempo and loudness: the average value within the window $\mu(w_i)$, the standard deviation $\sigma(w_i)$, the median $\nu(w_i)$, the minimum $m(w_i)$, the maximum $M(w_i)$, and the range $R(w_i) = M(w_i) - m(w_i)$. For each of these features, the corresponding normalized ones were also calculated by division by the mean. The normalized features are indicated with lower-case letters of the tempo/loudness subscripts. For example, if $\sigma_B(w_i)$ is the tempo standard deviation, then $\sigma_b(w_i) = \sigma_B(w_i)/\mu_B(w_i)$ is the corresponding normalized version. Additional features were added that represent correlations: $\Sigma_{tB}(w_i)$ is the correlation between the time and the tempo, $\Sigma_{tL}(w_i)$ is the correlation between the time and the loudness, and $\Sigma_{BL}(w_i)$ is the correlation between the tempo and the loudness. *Directness of movement* is a feature that captures aspects of the curvature of a trajectory segment: it measures the ratio between the length of a direct movement from A to B (α) and the length of the actual trajectory between the same points (β) in a bi-dimensional space (see Figure 2). These features were calculated in different spaces: in the time-tempo space ($\Delta_{tB}(w_i)$), in the time-loudness space ($\Delta_{tL}(w_i)$) and in the tempo-loudness space ($\Delta_{BL}(w_i)$). Finally, some derivatives were calculated for tempo and loudness: the maximum of the absolute value of the derivative ($M\delta(w_i)$) and the average of the absolute derivative ($\mu\delta(w_i)$).

Table 3 summarizes all the features. The loudness-derived features that cannot be used in the learning process are also shown.

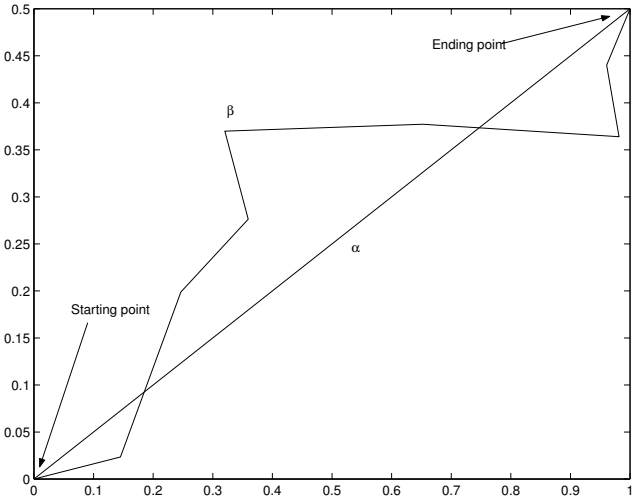

Figure 2: The definition of the directness index in an X-Y space.

2.3. The learning algorithms

For our first experiments, we selected a representative set of standard machine learning algorithms with different representations and *biases* (see Table 4). All of these are available in the Waikato Environment for Knowledge Analysis (WEKA)[1] [10], and the names given in Table 4 are the names (including parameters) by which they are called in WEKA.

The following learners were selected: J48 is a state-of-the art decision tree learner; NaiveBayes is a probability-based algorithm that applies Bayes' rule for prediction and assumes independence between the individual features; KStar is a Nearest-Neighbor classification algorithm; ClassificationViaRegression is a 'meta-learner' that induces linear discriminant functions for the individual classes and combines these into an n-class classifier by voting; VFI ('Voting Feature Intervals') is an extremely simple classifier that lets each feature vote for the class in isolation; DecisionStump learns extremely simple one-level decision trees; ConjunctiveRule is a similarly simple algorithm that

[1]The Java source code of WEKA is publicly available at `www.cs.waikato.ac.nz`

ID	WEKA Name
01	trees.j48.J48
02	bayes.NaiveBayes -K
03	lazy.kstar.KStar
04	meta.ClassificationViaRegression -W LinearRegression
05	misc.VFI
06	trees.DecisionStump
07	rules.ConjunctiveRule
08	rules.Ridor

Table 4: Learning algorithms used in the experiments.

learns one conjunctive decision rule per class; and Ridor is a algorithm for directly learning classification rules from examples.

All of these algorithms read the same data format and produce predictive models for discrete classification problems.

2.4. Testing methodology

Two kinds of aspects of the learned models are of interest: *qualitative* — do the models capture relevant and interpretable aspects of performance style? can we learn anything new about performance from them? — and *quantitative* ones — how well can the classifiers identify performers in new recordings? what is the recognition rate that can be achieved?

In the first experiments, we focused on quantitative issues, as these are easier to measure, in particular when we learn many different models with different learning algorithms. The first question to be studied was to what extent it is possible for machine learning algorithms to learn to identify performers in new recordings, and which are the most promising learning algorithms. This was tested via *cross-validation* at the level of pieces (sonata movements): each of the algorithms was trained on all of the sonata movements except one, the learned classifiers were then tested on the remaining movement, and the percentages of correct predictions were recorded. This process was repeated in a circular fashion, so that each sonata movement served as test piece exactly once for each classifier. The important thing about this procedure is that the predictivity of learned classifiers is always tested on independent data that was not used in the training phase, so that we get a realistic estimate of a classifier's expected performance.

3. RESULTS

The results of these cross-validation experiments are summarized in Table 5, which lists the *classification accuracies* achieved by the individual classifiers on each of the test pieces (after having been trained on the other pieces). Classification accuracy is defined as the percentage of instances in the test piece that were assigned the correct class by the classifier. This is to be compared to the so-called *default* or *baseline accuracy*, which is the success rate one would achieve by 'intelligent guessing', i.e., by always predicting the class that is most frequent in the training data.

At first sight, the results look disappointing. The accuracies achieved by the individual classifiers are far from the optimum of 100%; they range from 9.98% (classifier 05 on test piece kv332_2) to 34.75% (classifier 04 on kv282_2) on individual pieces, and from 16.66% to 29.41% on average. This is put into perspective by noting that 6-way classification is a difficult problem: the default accuracy would be around 16.67% (see the last column in Table 5). All the classifiers predict significantly above this baseline on average, except for classifier 05, which obviously fails to learn anything sensible. So clearly, there is significant information in the performance data that can contribute to identifying the performer, even though the data in its current form is very abstract and incomplete.

Looking at the results piece by piece reveals that performer identification may be easier in some pieces than in others. The average performance over all classifiers on a piece-by-piece basis (see penultimate column in Table 5) is above the baseline in every case, ranging from 17.89 (kv280_3) to 25.05 (kv279_1). If we remove classifier 05 from the table, the average accuracy achieved by the remaining classifiers is even higher, ranging from 18.29 (kv280_3) to 27.42 (kv279_1).

A closer look shows that not all classifiers perform well or poorly on the same pieces (see, e.g., test piece kv279_2, where classifier 02 achieves its third-poorest result, while classifier 04 achieves its second-best). That indicates that it may be fruitful to join classifiers into so-called *ensembles* [2] which combine their predictions, e.g., by voting on the class of a new test case.[2] It is known from systematic research in machine learning that classifier combination is particularly promising if the errors of the individual classifiers are highly uncorrelated [1, 8]. Experiments with ensembles of classifiers are currently on our agenda.

In order to analyze which pianists are easier or more difficult for the classifiers to recognize, Figure 3 shows a so-called *recall-precision diagram*. Recall and precision are concepts from the field of information retrieval; they give an indication of the trade-off between being able to recognize (or retrieve) as many instances of a given target class as possible (true positives), and erroneously classifying other instances as belonging to the target (false positives). More precisely, assume there are P true instances of the target class C and N instances of other classes $\overline{C}_i$, and that a classifier classifies tp instances correctly as C, fp incorrectly as C (while they really belong to one of the $\overline{C}_i$); then *recall* is defined as tp/P and *precision* as $tp/(tp + fp)$.

As Figure 3 shows, the results for different pianists occupy different regions in recall-precision space. Glenn Gould (GG) seems to the most easily recognizable pianist,[3] relatively speaking: most of the learners manage to correctly recognize between 35 and more than 60 % of the examples related to Gould, with a precision that is well above the baseline precision (between .2 and .315). Also Maria João Pires (MP) and Daniel Barenboim (DB) seem to have recognizable characteristics. On the other hand, all the classifiers have problems recognizing András Schiff. This should, however, not be misconstrued to mean that Schiff has a less individualistic style. All we can say at the

[2] Another indication for the promise of ensemble learning is the fact that classifier 04, which is in itself a kind of simple ensemble learner, performed significantly better than any of the other learners in our experiment.

[3] Not surprisingly, some might say ...

	Classifiers									
Piece	**01** [%]	**02** [%]	**03** [%]	**04** [%]	**05** [%]	**06** [%]	**07** [%]	**08** [%]	**Average** [%]	**DEF** [%]
kv279_1	30.52	34.24	23.75	32.65	16.81	20.90	20.48	29.39	26.09	16.68
kv279_2	17.91	15.97	25.15	33.96	14.10	21.27	23.21	28.81	22.55	17.24
kv279_3	32.19	27.05	26.89	26.52	18.42	17.36	16.94	24.46	23.73	16.73
kv280_1	19.30	23.87	19.85	32.61	16.83	18.84	20.27	18.96	21.32	16.71
kv280_2	20.07	30.69	19.98	25.34	21.75	20.68	22.08	24.87	23.18	16.77
kv280_3	16.68	21.42	18.44	19.67	17.12	18.00	16.68	17.12	18.14	16.68
kv281_1	17.15	13.25	15.12	24.23	15.51	17.92	21.86	21.90	18.37	16.73
kv282_1	25.81	21.93	29.63	30.57	18.34	23.57	23.40	29.22	25.31	16.93
kv282_2	21.50	28.10	24.70	34.75	17.42	24.36	24.15	25.41	25.05	16.62
kv282_3	33.50	33.66	26.45	26.95	16.54	18.76	18.10	32.27	25.78	16.71
kv330_3	19.31	15.35	17.60	28.05	16.28	23.85	21.51	18.28	20.03	16.72
kv332_2	22.88	27.69	31.97	33.54	9.98	27.80	16.67	29.94	25.06	16.67
Average	23.07	24.44	23.29	29.07	16.59	21.11	20.45	25.05	–	16.76
Weighted Average	22.85	24.42	22.88	29.41	16.66	21.17	20.68	24.76	–	16.75

Table 5: Preliminary results: classification accuracy; DEF refers to the *default (baseline) accuracy*, i.e., the accuracy one would achieve by always predicting the class that is most frequent in the training data. 'Weighted average' is the mean classification accuracy when weighted by the relative size (number of instances) of the different test pieces.

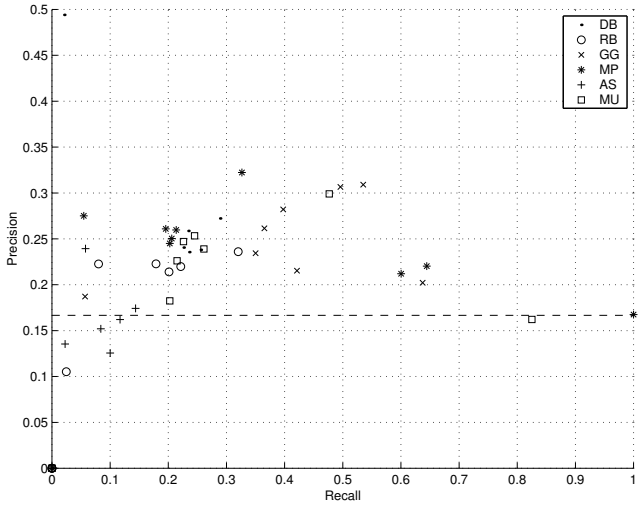

Figure 3: Recall-precision diagram showing recognition results for different pianists and all classifiers. The pianists are identified by different point types. The horizonal line at $y = .1667$ indicates the *default precision* that would result if we always predicted the artist we are interested in (which would give the optimal recall of 1.0) — note that one of the classifiers actually did this with Maria João Pires (MP).

moment (especially given the very preliminary phase of our data analysis) is that the current set of learning algorithms finds it easier to find partial discriminant descriptions of Gould's style, given the current set of features. More detailed experiments will be needed to learn something meaningful about aspects of style.

Again, the generally quite low recall and precision values may seem disappointing, but it must be kept in mind that these values derive from an n-way classification setting. The learners tried to learn a model that distinguishes between all the pianists simultaneously, rather than focusing on one particular pianist and trying to distinguish him or her from the others. We expect to get much better results if the problem is turned into n two-class discrimination tasks.

All the accuracy figures reported above refer to the classification of individual test examples (which are not entire pieces, but specific time points in a performance, with a window around them). That is, what is counted is how many of the individual instances each of the classifiers assigned to the correct pianist. In a more natural classification scenario one might be interested in the classification of an entire performance: who was the pianist behind a given recording? The simplest way to do that would be to assign the piece to the pianist who collects the most predictions over the set of instances that make up the piece. We do not have these figures ready at this moment (that will require some rewriting of the experiment scripts), but will have them available at the conference.

In summary, the first experimental results do indicate that it may be possible for a machine to recognize famous artists from their style, at least to some extent. The machine learning algorithms achieved recognition rates significantly above the baseline, which indicates that they can pick up some relevant information from the recordings. At first sight, the absolute accuracy figures look rather poor. However, our current training data (performance measurements) are extremely crude and incomplete (only beat-level tempo and beat-level overall loudness; no information about the loudness of individual voices, about articulation, about the timing of individual voices, etc.). Also, n-class iden-

tification task are notoriously difficult, and we expect to get much better results by converting the original problem into a large number of 2-way discrimination problems (one pianist against all others, or pairwise discrimination), and by employing more complex learning schemes. Some plans along these lines are listed below.

4. NEXT STEPS

This paper has presented the first preliminary experiments that study whether a machine can learn to recognize famous performers (concert pianists) based on aspects of their style of playing. Only very high-level and incomplete performance information was available for the experiments, and the results were mildly positive. Many improvements are possible, and the following next steps are currently on our research agenda:

- classification of entire pieces instead of individual instances (for instance, by voting over the instances);
- experiments with combinations of several different classifiers (*ensemble learning methods* [2]);
- reformulation of the n-class problem as n two-class problems (distinguishing one pianist from all the others) and appropriate combination of the resulting classifiers to solve the original n-class identification problem; that might also lead to learned models that tell us something about aspects of individual style;
- *round-robin* learning [5], i.e., breaking up the n-class problem into $\frac{n(n-1)}{2}$ two-class discrimination problems, one for each unique pair of artists; it has been shown that in this way classification accuracy may be significantly improved (and, surprisingly, also the efficiency of learning in terms of run-time);
- identification of those features that contribute to the discernibility of the individual pianists; by reducing the set of features to those that are truly relevant, additional improvements in classification accuracy may also be achieved;
- refinement of the performance measurements; we are working on methods to extract more detailed information regarding timing and dynamics from audio recordings.

We are confident that the current results can still be considerably improved, and that the models induced by the learning algorithms in the two-class discrimination tasks will provide some interesting insights into some of the things that make up the recognizable style of a great artist.

5. ACKNOWLEDGMENTS

This research was supported by an ERASMUS scholarship to the first author, the EC project HPRN-CT-2000-00115 MOSART, and the project Y99-INF (START Prize by the Austrian Federal Government). The Austrian Research Institute for Artificial Intelligence (ÖFAI) acknowledges basic financial support from the Austrian Federal Ministry for Education, Science, and Culture.

6. REFERENCES

[1] Bauer, E. and Kohavi, R. (1999). An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants. *Machine Learning* 36:105–169.

[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] Dixon, S., "Automatic Extraction of Tempo and Beat from Expressive Performances", Journal of the New Music Research, 30(1):39-58, 2001.

[4] Dixon, S., Goebl, W., and Widmer, G., "The Performance Worm: Real Time Visualization of Expression based on Langner's Tempo-Loudness Animation", Proc. International Computer Music Conference (ICMC 2002), Göteborg, Sweden, pp. 361-364, 2002.

[5] Fürnkranz, J. (2002). Round Robin Classification. *Journal of Machine Learning Research* 2:721–747.

[6] Langner, J. and Goebl, W. (2002). Representing Expressive Performance in Tempo-Loudness Space. *Proceedings of the ESCOM Conference on Musical Creativity*, Liège, Belgium.

[7] Stamatatos, E. (2002). Quantifying the Differences between Music Performers: Score vs. Norm. *Proceedings of the International Computer Music Conference (ICMC'2002)*, Göteborg, Sweden.

[8] Stamatatos, E. and Widmer, G. (2002). Music Performer Recognition Using an Ensemble of Simple Classifiers. *Proceedings of the 15th European Conference on Artificial Intelligence (ECAI'2002)*, Lyon, France.

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

[10] Witten, I.H. & Frank, E. (1999). *Data Mining*. San Francisco, CA: Morgan Kaufmann.

[11] Zwicker, E. and Fastl, H. (2001). *Psychoacoustics. Facts and Models*. Springer Series in Information Sciences, Vol.22. Berlin: Springer Verlag.