An assessment of learned score features for modeling expressive dynamics in music

Maarten Grachten, Florian Krebs

***Abstract*—The study of musical expression is an ongoing and increasingly data-intensive endeavor, in which machine learning techniques can play an important role. The purpose of this paper is to evaluate the utility of unsupervised feature learning in the context of modeling expressive dynamics, in particular note intensities of performed music. We use a note centric representation of musical contexts, which avoids shortcomings of existing musical representations. With that representation, we perform experiments in which learned features are used to predict note intensities. The experiments are done using a data set comprising professional performances of Chopin's complete piano repertoire. For feature learning we use Restricted Boltzmann machines, and contrast this with features learned using matrix decomposition methods. We evaluate the results both quantitatively and qualitatively, identifying salient learned features, and discussing their musical relevance.**

I. Introduction

The performance of music is a human activity that has sparked scientific interest for more than a century, with pioneering works like [1] and [2]. An important challenge has been to account for the variations in tempo, dynamics, and articulation (among other things), that are inherently present in expressive performances of a musical piece by a skilled musician. Research in this area has employed various methodologies. Some accounts of musical expression, in line with philosophy and traditional musicology, take a dialectic form, where views are put forward and disputed by authors, typically in the form of essays, where insights developed by the author are illustrated in the context of excerpts from selected musical works, as in [3]. A substantial amount of music performance research adopts methodologies more common to psychology, in which controlled experiments are carried out to test a particular hypothesis, as in [4].

More recently, music performance has been viewed from data mining and machine learning perspectives, where the aim is to take advantage of large amounts of measurement data from music performances, in order to find statistically significant patterns that can be related to principles of expressive performance . Most of the existing work in this area focuses on training computational models that link one or more aspects of musical expression (such as variations in tempo or dynamics) to underlying factors, most prominently the written musical score. Whether, and if so which, expressive patterns can be found is largely determined by the way the musical score is represented in such models. Most, if not all computational models of expression to date make use of hand-designed features to describe the musical score, based mostly on the researcher's intuitions, or those of a musical expert [5].

A strong dependence on hand-designed features has also characterized many classifiers and predictive models in image processing (notably the successful SIFT features [6]). In this field however, the past decade has witnessed a strong development of computational methods for learning features from data, rather than hand-crafting them. A notable example that has proven useful for face recognition is *nonnegative matrix factorization* (NMF) [7]. Biologically plausible visual features have also been reported by *slow feature analysis* [8]. Furthermore, the use of *deep belief networks* [9], has been proven highly effective for a variety of complex learning tasks, such as handwriting recognition [10], and object recognition in images [11]. Such architectures typically consist of stacked two-layer networks, each of which represents a generative probabilistic model of the data at a different level of abstraction.

The purpose of this paper is to evaluate the utility of unsupervised feature learning methods in the context of music expression modeling. We will limit ourselves to the prediction of note intensities in classical piano performances based on learned features. The predictive model we use to evaluate the learned features is not intended as a system, or application in itself (although successful predictive models of expression can be beneficial to tasks like automatic score-following [12]). Rather, the reported experiments are intervened as a case study of how feature learning methods can be used in computational models of musical expression.

Although it is undisputed that a minimally comprehensive model of note intensities should include notions of higher level structure and dependencies [13], our focus will for now be on learning features that describe local contexts in musical scores, comparable in scope to hand-designed features used in other work, such as [14], [15], [16], and [17]. In terms of feature learning methods, our prime interest is in the use of RBM's, and deeper learning structures based on RBM's. We compare the RBM based methods with more straightforward matrix decomposition techniques, specifically NMF and principal component analysis (PCA).

We evaluate these methods both in quantitative and in qualitative terms. For quantitative evaluation, we perform an experiment in which we use the sets of features learned by each method to train models of musical expression (in particular in the form of note intensities) and test their predictive accuracy. Because the focus is on the utility of the learned features, we use linear regression models, as the simplest sensible class of models. For qualitative evaluation, we discuss

Maarten Grachten is affiliated to the Austrian Research Institute for Artificial Intelligence (OFAI), Vienna, Austria; Florian Krebs is affiliated to the Department of Computational Perception, Johannes Kepler Universität, Linz, Austria.
E-mail: see http://www.ofai.at/~maarten.grachten/

Manuscript received ???; revised ???.

the types of features that are learned, and review their musical significance in cases where this is possible. We also use the regression coefficients of the expressive models to identify which features are relevant for predicting expression.

The paper is organized as follows: In section II we discuss related work in both music performance research, and unsupervised feature learning. We will also discuss music oriented applications of the latter. In section III, we describe the representation of musical data as input for feature learning, and subsequently we briefly introduce the feature learning methods to be used. Section IV contains a description of the musical corpus used for feature learning and evaluation and presents the feature learning and evaluation procedure in more detail. The results are presented and discussed in section V, conclusions and future work are presented in section VI.

II. Related Work

The application of unsupervised feature learning in the context of sound and music processing is relatively new, but the method is rapidly gaining popularity. Humphrey et al. [18] argue that the use of feature learning with deep learning architectures is the key to improve the state of the art in many areas of music informatics.

Previous applications of feature learning can roughly be categorized according to the nature of the input data. On the one hand, there are audio based applications. For example, phones in recorded speech can be successfully recognized using deep belief networks on MFCC's features of the audio [19]. Furthermore, music similarity can be computed competitively with mean-covariance RBM features computed from audio, using whitened, block-level, Mel-scale spectral bins [20].

Feature learning has also been applied to *symbolic* representations of music. A time recurrent specialization of RBM's has been applied to model the conditional probability of musical notes, given their preceding musical context [21]. It was shown that using the predictions of this model the accuracy of polyphonic music transcription was improved.

A similar RBM architecture has been used by Spiliopoulou and Storkey, to model temporal and tonal structure in monophonic melodies [22]. In contrast to [21], and other RBM architectures for sequence modeling [23], [24], their architecture is convolutional through time, and models the joint probability of notes with their preceding context, rather than the conditional probability.

III. Feature learning

In this section we describe how we use PCA, NMF, and (stacked) RBM's to learn features from musical material. We start by describing the way music is represented as input for feature learning.

A. Data representation

As stated in the introduction, we focus on the performance of classical piano music, in particular the piano works of Chopin (see section IV-A). This means that the musical material we deal with is mono-instrumental and polyphonic.

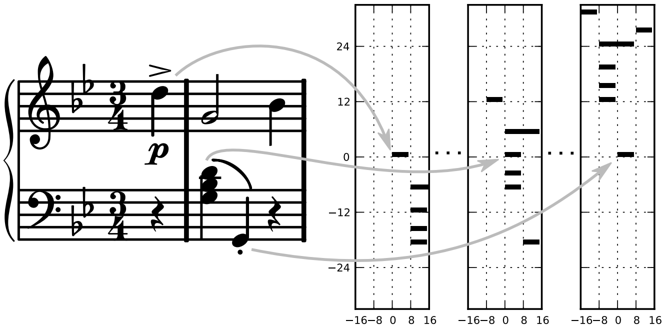

Fig. 1. Note centered piano roll representation of symbolic music (Excerpt from Chopin's Nocturne, Op. 15, No. 3)

We choose to work with the *piano roll* representation of music, a time-frequency representation roughly analogous to the spectrogram representation for audio. A musical piece can then be described as a sequence of (possibly overlapping) note configurations, by taking snapshots of parts of the piano roll, as illustrated in figure 1.

Unlike related approaches to modeling symbolic music we do not map absolute pitches [21], or chroma-like [22] attributes to input variables. The disadvantage of mapping absolute pitches is that the input is not transposition-variant. This means for example that a major triad is mapped to a different set of input variables, depending on the pitch and octave at which it is played. Using pitch chroma (the absolute pitch modulo 12) brings only octave invariance, but not pitch invariance. A chroma-like approach may be acceptable in the context of monophonic melodies, but in the case of polyphonic piano music, mapping all pitches to one or two octaves gives a severely distorted image of the musical context. This is especially true for piano music from the romantic period, where dramatic passages may span virtually the whole keyboard.

To avoid these undesired consequences, we take a *note centric* approach. This means that the context of each note is described *relative* to the centered note. Thus, in terms of pitch, a particular context note does not represent, say, an A4 pitch, or an A chroma, but rather a pitch interval, say, 5 semitones above the centered note.

Note that this approach implies that to represent a musical context where the highest and the lowest possible pitch occur simultaneously, the input needs to span twice the range of the piano keyboard. Consequently, our input representation for piano roll fragments has a vertical dimension of 174, that is 87 semitones (the typical range of a piano keyboard) above the current note, and 87 below[1]. The horizontal dimension was varied between 16, 32 and 64 units, where each unit corresponds to the duration of a 32th note. Thus, a fragment spans one, two or four beats before and after the onset of the current note (resulting in windows of two, four and eight beats respectively). The onset and offset times of all notes are quantized to the 32th grid. We refer to the horizontal dimension as the *score context size*.

For a given note, the piano roll fragment is represented as a binary matrix, where 1's indicate the presence of a context note

[1] Note that the range has been truncated in figure 1, for display purposes

at a given relative pitch and time with respect to the current note. One possibility is to indicate only the onset of each note with a 1 at the matrix cell corresponding to it relative *onset* time and pitch. Alternatively, the entire *duration* of each note can be coded by setting all cells to 1 which lie between the relative onset and offset of the note. With this latter coding however, it is not possible to distinguish between a single longer note and several consecutive notes of the same pitch where the offset of one note coincides with the onset of the next note. To avoid this ambiguity, the last matrix cell before offset of each note is left at 0, creating a gap if minimal size between consecutive notes of the same pitch.

In the rest of the paper, we will refer to the former, onset-only representation as *onset coding*, and to the latter as *duration coding*. For computation, each score fragment (a binary matrix of size 174×32 per note) is arranged linearly into a vector $\mathbf{v}$ of length m, where $m = 174 \cdot 32 = 5568$.

B. Principal Component Analysis

PCA is a frequently used tool for dimensionality reduction of data. It transforms data using a set of orthogonal (i.e., linearly uncorrelated) basis vectors. These basis vectors are selected to be the eigenvectors of the covariance matrix of the $n \times m$ data matrix V. The k basis functions that explain most of the variance in the data correspond to the k largest eigenvalues and yield the $k \times m$ projection matrix E. Using E, the data vector $\mathbf{v}$ can be transformed into the feature space using the multivariate function $f_{pca}(\mathbf{v})$ by

$$f_{pca}(\mathbf{v}) = \mathbf{v}E'. \tag{1}$$

As $k < m$, the projected data vector $f_{pca}(\mathbf{v})$ has lower dimensionality than the original data vector $\mathbf{v}$. The basis vectors in E can be interpreted as vectorized images. Therefore, we will refer to the rows of E as (PCA) *basis images*.

We compute the principal components based on randomized singular value decomposition [25].

C. Nonnegative matrix factorization

Nonnegative matrix factorization of a non-negative matrix V is the problem of finding non-negative matrices F and H such that:

$$V \approx F_{nmf} H \tag{2}$$

Note that this corresponds to equation 1 with the difference that with NMF, the matrices F_{nmf} and H are restricted to non-negative values and the basis functions H are not orthonormal. In our context, H is a $k \times m$ matrix that holds vectorized basis images as rows, and F_{nmf} is a $n \times k$ matrix that holds basis image activations of note contexts as rows.

We use a projected gradient [26] method to solve the NMF problem (2), where the minimized quantity is the euclidean norm of the difference between the target matrix and its NMF approximation.

Once a matrix of basis images H has been learned from the data, we take the activation pattern of H for a given data vector $\mathbf{v}$ as the feature description $f_{nmf}(\mathbf{v})$ of $\mathbf{v}$:

$$f_{nmf}(\mathbf{v}) = \underset{\mathbf{f}}{\operatorname{argmin}}\, ||\mathbf{v} - \mathbf{f}'H|| \tag{3}$$

D. Restricted Boltzmann machines

Boltzmann machines are stochastic neural networks, whose global state is characterized by an energy function (that depends on the activation of units, their biases and the weights between units) [27]. The probability of a unit being active depends on the difference in energy between the state where the unit is on and the state where the unit is off. When the units in the network represent the state of a set of (binary) observation variables, a Boltzmann machine with a particular set of bias and weight parameters defines a joint probability mass function over observations. The model parameters that minimize the total energy of the model on the data, are the maximum likelihood parameter estimates for the data.

Restricted Boltzmann machines (RBM's) are a special case where the network is a complete bipartite graph, such that units are divided into *visible* units and and *hidden* units. The visible units are used to represent data, and the hidden units are interpreted as factors that jointly (and non-linearly) determine the probability that visible units are activated[2]. It has been shown that RBM's can be effectively be trained to approximate the probability distribution of data using an approximate learning procedure called Contrastive Divergence [28].

A trained RBM with visible-to-hidden weights W and hidden bias b can be used as a feature extractor, where the features $f_{rbm}(\mathbf{v})$ of a data point $\mathbf{v}$ are defined as the hidden activation probabilities $p(\mathbf{h}|\mathbf{v})$:

$$f_{rbm}(\mathbf{v}) = \sigma(W'\mathbf{v} + b) \tag{4}$$

where $\sigma(x) = (1 + \exp(-x))^{-1}$. The columns of matrix W can be interpreted as basis images, analogous to those of the PCA and NMF methods.

E. Stacked Restricted Boltzmann machines

Given an RBM that extracts features from the data, it is trivial to train a subsequent RBM that takes the features of the first RBM as inputs. This stacking of RBM's can be repeated multiple times. In this way, higher level features can be learned. For a stack of l RBM's, we define the features as the activation probabilities of the top hidden layer, which are defined in terms of the activation probabilities in the lower layers:

$$f_{rbm_l}(\mathbf{v}) = \sigma(W_l' \; f_{rbm_{l-1}}(\mathbf{v}) + b_l) \tag{5}$$

$$\vdots$$

$$f_{rbm_1}(\mathbf{v}) = \sigma(W_1'\mathbf{v} + b_1). \tag{6}$$

In the case of stacked RBM's, there are multiple layers of basis images, where the basis images in the higher layers can not be interpreted with the same semantics as the input (as is the case with the other feature learning methods).

[2] Due to the bipartite structure, visible units are conditionally independent given the hidden units, and vice versa

F. Features and Basis Images

Because the matrix decomposition methods (NMF and PCA) and the RBM based methods described above are quite dissimilar, it may be helpful to be explicit on how we use them, and in particular what we mean by *features* and *basis images*.

They above methods have in common that they produce a transformation that maps data from the input space into a new (learned) space. We call the dimensions of the new space *features*. Each feature has a corresponding *basis image*, that has the dimensionality of the input space. A basis image gives a visual impression of the type of data that will "activate" the feature. A fundamental difference between NMF and PCA on the one hand, and (stacked) RBM's on the other, is that in the former, the input activates the features *linearly* through the basis image, and in the latter, the features are activated *non-linearly*. A further difference is that in the case of stacked RBM's, it is not trivial to produce basis-images[3].

IV. Methodology

A. data

For the evaluation we use the Magaloff corpus [30] – a data set that comprises live performances of virtually the complete Chopin piano works, as played by the Russian-Georgian pianist Nikita Magaloff (1912-1992). The music was performed in a series of concerts in Vienna, Austria, in 1989, on a Bösendorfer SE computer-controlled grand piano [31] that recorded the performances onto a computer hard disk. The recorded data contains highly precise measurements of the times any keys (and pedals) were pressed and released, and the intensity with which they were pressed.

Symbolic scores were obtained from scanned sheet music using optical music recognition. Performances were aligned to the score automatically using an adaptation of the edit-distance based method used in [32]. Subsequently, the alignments were corrected manually.

The data set consists of 155 pieces, adding up to over 320,000 performed notes, almost 10 hours of music.

B. Prediction of note intensities with learned features

In addition to the question how precisely the learned features sets described in section III encode musical contexts, we evaluate their utility with respect to predicting expressive dynamics, in particular note intensities. We do so by using linear regression from the feature sets to the target variable, the intensities with which score notes are performed.

For each score feature setting, we learn the features on the complete data set and then learn the prediction coefficients employing a leave-one-out evaluation approach. That is, for each of the 155 pieces in the data set, regression coefficients are computed on the remaining pieces. As the score features are learned in a purely unsupervised manner and the objective functions that are minimized in order to learn the features have no relation with the prediction task, we believe that this is a valid approach - in contrast to scenarios where unsupervised pre-training is combined with a supervised stage and the learned features are fine-tuned to optimize prediction accuracies.

In the first half of this section, we describe the different setups we use for learning features using NMF, PCA, and both single and stacked RBM's, respectively. In the second half, we describe how the learned features are used to predict note intensities.

1) Feature learning: configurations and setup: The input data is identical for all feature learning methods used. We apply each method to both onset and duration coded music, as described in subsection III-A. Furthermore, for each method, we test different numbers of features to be learned. In summary, we vary:

- Input data representation: onset coding, duration coding
- Feature learning method: NMF, PCA, RBM, stacked RBM
- Number of features learned: 50, 100, 200, 300, 400, 500, and 1000
- Score context size: 2, 4, and 8 beats

The NMF projected gradient method is run until convergence (or close to convergence in case of larger feature dimensionalities). In the case of RBM's, we always train the models for 100 epochs, although the learning typically converges after 20 to 50 epochs. Furthermore, we consider stacks of two RBM's, where the lower RBM always has 1000 hidden units.

2) Prediction of note intensities: With the feature sets learned as described above, we predict the note intensities as measured in the music performances of the Magaloff data set. In the Magaloff data set, note intensities are represented as MIDI velocities, which are roughly linearly proportional to peak sound pressure level (measured in dB) [33]. The note intensities encode what we refer to as *expressive dynamics*: intentional variations in loudness of performed notes to convey information to the listener (ignoring non-expressive, non-intentional variations due to e.g., motoric imprecisions of the performer). In this work, we address expressive dynamics exclusively, ignoring additional expressive parameters like tempo and articulation.

To predict the MIDI velocity for each note in the data set we proceed as follows: The note velocities for each piece are normalized to have zero mean, in order to be independent of the absolute velocity. After having learned the features as described in section III which yields the matrices E, H, W, W_l and the vectors b, b_l, we compute the activations $f(\mathbf{v})$ of the features for each note in the data set (feature extraction). Finally, the velocity y_i of a note i is predicted by a linear function g of the feature activations $f(\mathbf{v}_i)$ and a vector of regression coefficients $\mathbf{c}$:

$$y_i \approx g(f(\mathbf{v}_i), \mathbf{c}) = \mathbf{c}' f(\mathbf{v}_i) \tag{7}$$

The regression coefficients $\mathbf{c}$ are obtained by finding the least squares solution

$$\mathbf{c} = \underset{\mathbf{c}}{\operatorname{argmin}} \, ||y - \mathbf{c}' f(\mathbf{v})||. \tag{8}$$

[3] See [29] for some possible approaches

TABLE I
RECONSTRUCTION ERRORS OF SCORE FEATURES WITH SCORE CONTEXT SIZE OF EIGHT BEATS

#	duration coding				onset coding			
feat.	NMF	PCA	RBM	sRBM	NMF	PCA	RBM	sRBM
50	116.6	114.5	115.7	154.0	40.4	40.0	45.6	48.4
100	101.7	99.2	93.8	140.0	37.4	36.7	43.0	48.2
200	82.2	79.5	67.5	117.9	32.7	32.0	31.1	43.6
300	68.6	66.1	51.2	103.8	29.0	28.4	20.7	38.9
400	59.0	56.3	39.4	74.8	26.3	25.4	15.5	36.5
500	51.7	48.6	31.4	64.3	23.8	23.0	11.5	34.9
1000	38.9	27.3	14.4	39.4	23.5	14.6	4.0	26.2

For each of the feature sets, we predict the velocities in a leave-one-out scenario and a best-fit scenario. In the leave-one-out scenario, the coefficients $\mathbf{c}$ are computed separately for each piece, using the whole data set except the piece of interest. In the best-fit scenario, the coefficients are also computed separately for each piece, but using only the piece of interest as training data. Note that the latter yields optimal coefficients for each piece in terms of prediction error and provides an upper bound to the prediction results that can be obtained by a given feature set using linear prediction.

C. Prediction measure

We quantify the prediction results in terms of the coefficient of determination R^2, which measures the proportion of variance in target that is explained by the linear model.

V. Results and discussion

A. Reconstruction of the input data with learned features

In table I we show the reconstruction errors, i.e., the squared distance between the original data V and its "estimate" $\tilde{V}$. In the case of NMF and PCA, the reconstruction of an image is obtained projecting its feature activations back linearly into the input space, through the basis images. As expected, PCA shows a smaller reconstruction error than NMF. This is because by definition, its objective is to minimize the reconstruction error.

In the case of RBM's, the reconstruction of an image is a non-linear projection of the feature activations into the input space. The comparatively high error for the stacked RBM's can be explained by the fact that the first level hidden unit activations are not determined only by the data to be reconstructed, but also by the second layer of hidden units, which serves as a prior over the first hidden layer. This prior can make reconstructions more robust when the input is degraded, but in case the input data is presented as is, it tends to distort the input.

B. Prediction of note intensities

The prediction results as measured in terms of R^2 are shown in figure 2 in four different plots. Note that the data shown in the plots is the same in all four plots, only the x-axis and the color/shape coding differs, to highlight different trends. Each point in the plots represents the average R^2 value of predicted note intensities over all performed notes of the 155 musical pieces, where the note intensities for one piece are predicted using a regression model trained on the 154 other pieces. Furthermore, by *number of features* we refer to the number of basis functions in NMF, the number of principal components in PCA, number of hidden units in RBM's, and to the number of *top-level* hidden units in stacked RBM's.

From the overall range of the R^2 values it becomes clear that roughly 5 to 15% of the variance in note intensity is explained by the models. This may appear rather low, but it is important to bear in mind that musical expression is a phenomenon known to be much more complex than can possibly be captured in terms of local contexts of the musical score, as described in the introduction. Despite that, the results reveal interesting information, both about the feature learning methods, and about expressive dynamics.

There are several trends to be observed from the results. Firstly, there is a positive correlation between the size of the score context modeled by the features and prediction accuracy (figure 2a), indicating that note intensities can be modeled better on features that describe larger time spans of the music. On average, best results are obtained for a the largest score context computed in this experiment (8 beats), which corresponds to two bars of music in a 4/4 time signature. This result is in line with the common idea that the expressive dynamics of performed notes does not only depend on the immediate context, but also involves longer range dependencies.

A second clear trend is the increase of predictive accuracy with increasing numbers of features (figure 2b). Using less than 200 features to represent score contexts is detrimental to the suitability of the learned feature space for predicting note intensities. Again, the best results are obtained for the largest feature space dimensionality considered in this experiment. Larger dimensionalities might improve results further, but the trend visible in the plot suggests that the improvement will be only marginal.

The input coding method (figure 2c) has no clear effect on prediction accuracy. This seems to suggest that for predicting note intensities, knowing both onset times and durations of notes has no benefit over knowing just onset times. This result is surprising, since only when the durations of notes are known, it is possible to determine which notes sound simultaneously. Particular constellations of notes may sound very different depending on whether (dissonant) notes overlap or not, and it may be expected that this has an influence on the intensities with which notes are played.

Figure 2d shows the results grouped by feature learning method. It shows an advantage of the RBM based methods over the matrix decomposition methods, irrespective of the input coding, and the number of features. It is conceivable that this discrepancy is caused by the fact that the NMF and PCA features depend linearly on the inputs, whereas the RBM based features involve the non-linear sigmoid function (see subsection III-D), which potentially increases the flexibility and robustness of the features in the light of deformations in the input. Furthermore, from the results it appears to be no clear advantage of stacked RBM's over single RBM's. In

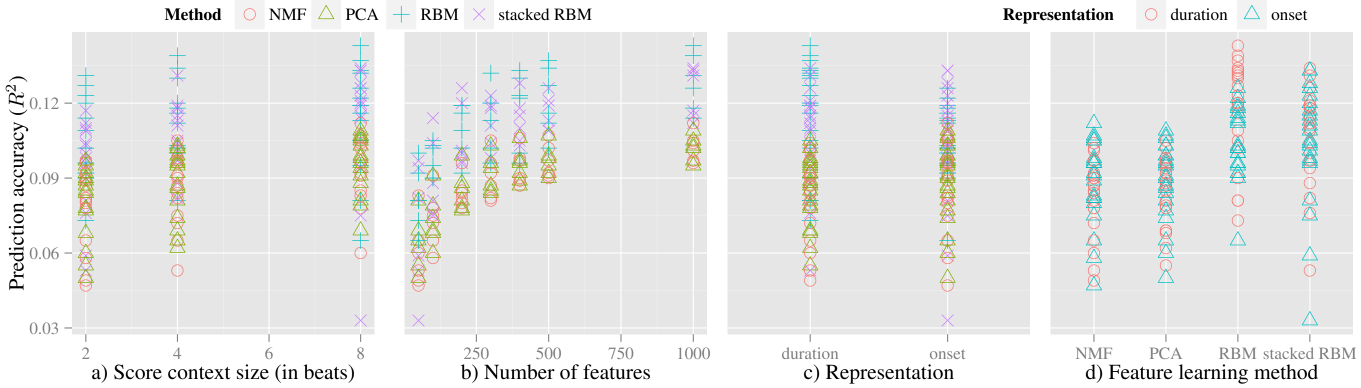

Fig. 2. Prediction accuracy (R^2) of note intensities using linear regression on learned features, as a function of different parameters; In plots a-c, the color/shape coding represents the feature learning method; In plot d, the color/shape coding represents the representation of notes

some cases the stacked RBM's improve the results, in other cases RBM's perform better. This result is consistent with other comparisons of RBM's with stacked RBM's (e.g. [20]). For the success of stacked RBM's it may be necessary to fine-tune the learned features using supervised learning [34]. This seems plausible, because as the learned features of deeper networks grow more abstract, there may be an increased need to "ground" the features in some specific task (such as predicting note intensities).

Figure 2d also reveals that although on average duration and onset coding perform similarly, the feature learning methods behave differently on both codings. In particular, the best results for the matrix decomposition methods are obtained for onset coding, whereas the RBM-based methods work best for duration coding. This may be an indication that RBM's are more capable of exploiting the harmonic structure of the music that is implicit in the duration coding, but it is not clear which characteristics of the feature learning methods account for this difference.

To get an impression of the type of information that features capture, it is helpful to inspect their corresponding basis images. A selection of basis images is shown for NMF, PCA, and RBM (all having size 500, and spanning a score context of 4 beats), in figure 3, top, center, and bottom, respectively. Each figure shows results for both duration and onset coding. Figure 3-NMF shows that NMF, most likely due to the non-negativity constraint, learns very sparse features, that often represent only one or a few notes. Interestingly, the recurring diagonal structures produced with onset coding are not present in the features learned on duration coding.

The features learned by PCA are much less sparse, and have basis images that are both positive and negative. Duration based features tend to emphasize harmonic relationships (the horizontal structures in figure 3-PCA-b,c) and in some cases even harmonic progressions (figure 3-PCA-e). Onset based features on the other hand, represent mainly rhythmical structures. Nevertheless, the structure in the PCA features is not very localized in pitch and time. Rather, it spans the central pitch region in a rather homogeneous way across time.

The RBM feature set (figure 3-RBM) also contains both harmony and rhythm related features, but these are distributed more evenly across duration and onset based features. The RBM examples also have a more diverse and localized character, with some features being sensitive only to the harmonic structure in a single beat unit (figure 3-RBM-a), whereas others are sensitive to the presence to notes in a specific *region* of the musical context, irrespective of the precise pitch and time (figure 3-RBM-e, i).

A question of special interest is whether it is possible to identify learned features that are helpful in predicting note intensities. To this end, we correlate the activation of each feature with note intensity. For the RBM 500 feature sets, this yields approximately 40 features with an r value over 0.1. Of those features, the few features with highest correlations have r values around 0.2. Figure 4 shows a some of those features.

Even if some harmonic structure is visible in some features (4c,e,h), it is evident that the features with strongest correlations to note intensity tend to be sensitive mainly to rather fuzzy regions above and below the center note. A light region above the center activates the feature when a note is located below other notes, which is typically the case for bass/accompaniment notes. Moreover, a dark region below the center inhibits the feature in the presence of notes below the current notes. Features with opposite characteristics (a light region below the center, and a dark region above), are most strongly activated for notes having neighboring notes below, but not above, as is usually the case with melody notes. Thus, features a, d, j in figure 4, and to a lesser extent e and h, can be interpreted as bass/accompaniment note detectors, and features b, c, f, g, i as melody note detectors.

In this light, it can be observed by means of the r values below the features, that the bass/accompaniment note features are negatively correlated with note intensity, and and the melody note features positively. This finding is in accordance with results reported in [35]. In that study, note intensity was modeled using a third degree polynomial function of pitch, yielding a prediction accuracy of $R^2 = 0.149$ on the same data set as is used in the current experiment. This result is slightly over the best results we report here. The polynomial pitch model, in combination with other hand-crafted features, and loudness annotations from the score, gives a maximal prediction accuracy of $R^2 = 0.188$.

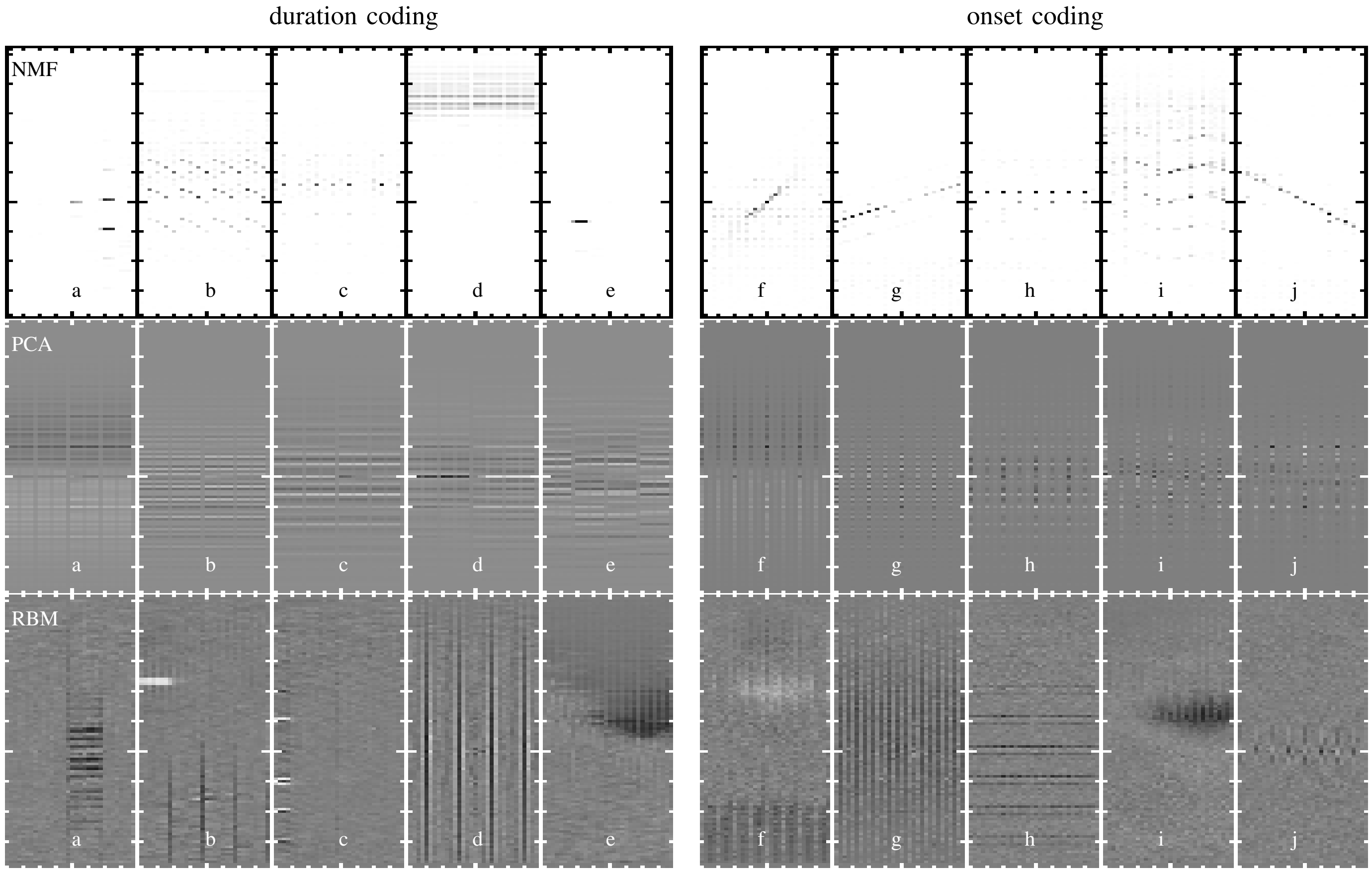

Fig. 3. Example basis images for duration coding (left) and onset coding (right), produced by NMF (top row), PCA (middle row), and RBM (bottom row); The center note of the context is indicated with a major tic at the border of each image; Vertical tics indicate octaves; horizontal tics indicate 8th notes. In the top row, white corresponds to zero values, black to positive values; in the center and bottom rows, light and dark colors correspond to positive and negative values, respectively

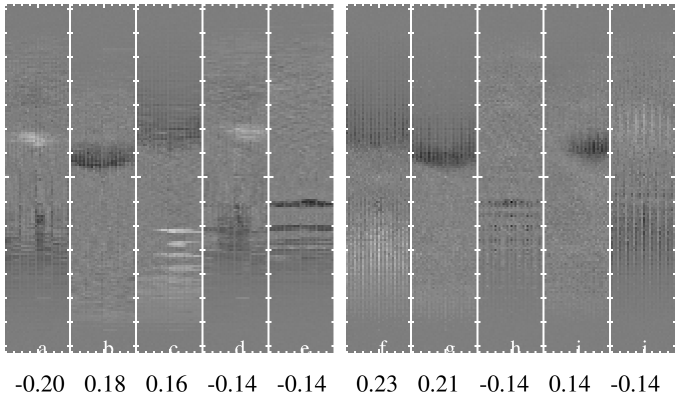

Fig. 4. RBM basis images for duration coding (left) and onset coding (right) with strongest note intensity correlation; Correlation coefficients (r) are printed below each filter

VI. Conclusions and future work

A crucial issue in expressive music performance research is the question how musicians shape the dynamics of their performance as a function of the musical material they are playing. Machine learning methods are used increasingly to model this relationship, but to date most methods rely on hand-designed features for representing musical scores. Recent developments in unsupervised feature learning have proven successful in image processing and other domains, but modeling symbolic music is a relatively unexplored application domain for unsupervised feature learning methods.

In this paper, we propose a novel input representation for musical context, that allows for learning a variety of different features from musical context, including harmonic, and rhythmic characteristics. The learned features are evaluated in the context of predicting expressive dynamics, in particular note intensities. Several non-supervised feature learning methods have been evaluated in this way. The results show that note intensities can be better modeled by features that model longer time ranges. Furthermore, predictive accuracy for note intensities is improved by learning a larger number of features. The results reported here are close to hand-designed features for modeling note intensities tested in [35].

The experiments reported here include only features learned in an unsupervised way, that have not been fine-tuned in any way to model note intensities explicitly. It is to be expected that such a fine-tuning can improved the results further, especially in the case of deep belief networks.

Acknowledgment

Parts of this work were supported by the Austrian Science Fund (FWF) in the context of the projects Z159 “Wittgenstein Award” and TRP-109, and by the European Union Seventh

Framework Programme FP7 through the PHENICX project (grant agreement no. 601166).

References

[1] A. Binet and J. Courtier, "Recherches graphiques sur la musique," L'année Psychologique (2), 201–222, 1896.

[2] C. E. Seashore, *Psychology of Music*. New York: McGraw-Hill, 1938, (Reprinted 1967 by Dover Publications New York).

[3] P. Kivy, *The Corded Shell: Reflections On Musical Expression*. Princeton, N. J.: Princeton University Press, 1980.

[4] J. Sundberg and V. Verrillo, "On the anatomy of the retard: A study of timing in music," *Journal of the Acoustical Society of America*, vol. 68, no. 3, pp. 772–779, 1980.

[5] J. Sundberg, A. Friberg, and L. Frydén, "Threshold and preference quantities of rules for music performance," *Music Perception*, vol. 9, pp. 71–92, 1991.

[6] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[7] D. D. Lee and H. S. Seung, "Learning the Parts of Objects by Non-negative Matrix Factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[8] P. Berkes and L. Wiskott, "Slow feature analysis yields a rich repertoire of complex cell properties," *Journal of Vision*, vol. 5, no. 6, pp. 579–602, Jul. 2005.

[9] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[10] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, 1998, pp. 2278–2324.

[11] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, vol. 25. MIT Press, 2012.

[12] A. Arzt, G. Widmer, and S. Dixon, "Automatic page turning for musicians via real-time machine listening." in *ECAI*, 2008, pp. 241–245.

[13] N. Todd, "The dynamics of dynamics: A model of musical expression," *Journal of the Acoustical Society of America*, vol. 91, pp. 3540–3550, 1992.

[14] G. Widmer, "Discovering simple rules in complex data: A meta-learning algorithm and some surprising musical discoveries," *Artificial Intelligence*, vol. 146, no. 2, pp. 129–148, 2003.

[15] G. Widmer, S. Flossmann, and M. Grachten, "YQX plays Chopin," *AI Magazine (Special Issue on Computational Creativity)*, vol. 30, no. 3, pp. 35–48, 2009.

[16] A. Friberg, R. Bresin, and J. Sundberg, "Overview of the kth rule system for musical performance," *Advances in Cognitive Psychology*, vol. 2, no. 2–3, pp. 145–161, 2006.

[17] A. Hazan, R. Ramirez, E. Maestre, A. Perez, and A. Pertusa, "Modelling expressive performance: A regression tree approach based on strongly typed genetic programming," in *Proceedings on the 4th European Workshop on Evolutionary Music and Art*, Budapest, Hungary, 2006, pp. 676–687.

[18] E. J. Humphrey, J. P. Bello, and Y. Lecun, "Moving beyond feature design: Deep architectures and automatic feature learning in music informatics," in *Proceedings of the 13th International Society for Music Information Retrieval Conference*, Porto, Portugal, October 8-12 2012.

[19] A. Mohamed, T. Sainath, G. Dahl, B. Ramabhadran, G. Hinton, and M. Picheny, "Deep belief networks using discriminative features for phone recognition," in *ICASSP-2011*, 2011.

[20] J. Schlüter and C. Osendorfer, "Music Similarity Estimation with the Mean-Covariance Restricted Boltzmann Machine," in *Proceedings of the 10th International Conference on Machine Learning and Applications (ICMLA 2011)*, Honolulu, USA, 2011.

[21] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription," in *Proceedings of the Twenty-nine International Conference on Machine Learning (ICML'12)*. ACM, 2012.

[22] A. Spiliopoulou and A. Storkey, "Comparing probabilistic models for melodic sequences," in *Proceedings of the 2011 European conference on Machine learning and knowledge discovery in databases - Volume Part III*, ser. ECML PKDD'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 289–304.

[23] I. Sutskever and G. Hinton, "Learning multilevel distributed representations for high-dimensional sequences," in *Proceedings of AISTATS*, 2007.

[24] A. J. Lockett and R. Miikkulainen, "Temporal convolution machines for sequence learning," Department of Computer Sciences, the University of Texas at Austin, Tech. Rep. AI-09-04, 2009.

[25] N. Halko, P. G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions," Applied & computational mathematics, California Institute of Technology, Tech. Rep., 2009.

[26] C.-J. Lin, "Projected gradient methods for non-negative matrix factorization," *Neural Computation*, vol. 19, pp. 2756–2779, 2007.

[27] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for boltzmann machines*," *Cognitive Science*, vol. 9, no. 1, pp. 147–169, 1985.

[28] G. E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, pp. 1527–1554, 2006.

[29] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, "Visualizing higher-layer features of a deep network," Dept. IRO, Université de Montréal, Tech. Rep., 2009.

[30] S. Flossmann, W. Goebl, M. Grachten, B. Niedermayer, and G. Widmer, "The Magaloff Project: An Interim Report," *Journal of New Music Research*, vol. 39, no. 4, pp. 369–377, 2010.

[31] R. A. Moog and T. L. Rhea, "Evolution of the Keyboard Interface: The Bösendorfer 290 SE Recording Piano and the Moog Multiply-Touch-Sensitive Keyboards," *Computer Music Journal*, vol. 14, no. 2, pp. 52–60, 1990.

[32] M. Grachten, J. L. Arcos, and R. López de Mántaras, "Evolutionary optimization of music performance annotation," in *Computer Music Modeling and Retrieval*, ser. Lecture Notes in Computer Science. Springer, 2004.

[33] W. Goebl and R. Bresin, "Measurement and reproduction accuracy of computer controlled grand pianos," *Journal of the Acoustical Society of America*, vol. 114, no. 4, pp. 2273–2283, 2003.

[34] J. Schlüter, "Unsupervised Audio Feature Extraction for Music Similarity Estimation," Master's thesis, Technische Universität München, Munich, Germany, October 2011.

[35] M. Grachten and G. Widmer, "Linear basis models for prediction and analysis of musical expression," *Journal of New Music Research*, vol. 41, no. 4, pp. 311–322, 2012.

Maarten Grachten holds a Ph.D. degree in computer science and digital communication (2006, Pompeu Fabra University, Spain). He is a former member of the Artificial Intelligence Research Institute (IIIA, Spain), the Music Technology Group (MTG, Spain), the Institute for Psychoacoustics and Electronic Music (Belgium), and the Department of Computational Perception (Johannes Kepler University, Austria). Currently, he is a senior researcher at the Austrian Research Institute for Artificial Intelligence (OFAI, Austria). Grachten has published in and reviewed for international conferences and journals, on topics related to machine learning, music information retrieval, affective computing, and computational musicology.

Florian Krebs received the Diploma degree in Electrical Engineering - Audio Engineering from University of Technology and University of Music and Dramatic Arts (Graz, Austria) in 2010. He is currently a Ph.D. candidate at the Department of Computational Perception of the Johannes Kepler University (Linz, Austria). His work focuses on the automatic analysis of music, including onset detection, beat tracking, tempo estimation and expressive performance analysis with interest in probabilistic graphical models.