FEATURE DISCOVERY FOR SEQUENTIAL PREDICTION OF MONOPHONIC MUSIC

Jonas Langhabel[1] **Robert Lieck**[2,3] **Marc Toussaint**[2] **Martin Rohrmeier**[3,4]
[1] Department of Computer Science, TU Berlin, Germany
[2] Machine Learning and Robotics Lab, University of Stuttgart, Germany
[3] Systematic Musicology and Music Cognition, TU Dresden, Germany
[4] Digital and Cognitive Musicology Lab, EPFL, Switzerland
langhabel@campus.tu-berlin.de, robert.lieck@ipvs.uni-stuttgart.de,
marc.toussaint@ipvs.uni-stuttgart.de, martin.rohrmeier@epfl.ch

ABSTRACT

Learning a model for sequential prediction of symbolic music remains an open challenge. An important special case is the prediction of pitch sequences based on a corpus of monophonic music. We contribute to this line of research in two respects: (1) Our models improve the state-of-the-art performance. (2) Our method affords learning interpretable models by discovering an explicit set of relevant features. We discover features using the *PULSE* learning framework, which repetitively suggests new candidate features using a generative operation and selects features while optimizing the underlying model. Defining a domain-specific generative operation allows to combine multiple music-theoretically motivated features in a unified model and to control their interaction on a fine-grained level. We evaluate our models on a set of benchmark corpora of monophonic chorales and folk songs, outperforming previous work. Finally, we discuss the characteristics of the discovered features from a musicological perspective, giving concrete examples.

1. INTRODUCTION

Predictive processing and the formation of expectancies are core capacities of human cognition, that are closely tied to the perception and interaction with our environment and to survival and fitness in an evolutionary perspective. Apart from its role in most cognitive domains, predictive processing has also been understood to play a fundamental role in music perception [22, 29]. The formation of musical expectancies is essential for goal directed processes at different musical time-scales, for musical interaction and synchronization as well as for the play with emotional effects in music [10, 17], and particularly, musical tension [8, 13]. Musical expectancy has also been understood to be culture- and style-dependent and to be grounded in musical knowledge that is acquired through processes of implicit or statistical learning [10, 31, 32]. Modeling of human predictive processing in music is thus fundamental for computational cognitive models of music as well as for models of musical interaction or generation.

Musical expectancy has been studied in terms of melody, harmony and rhythm. While the task involves the prediction of the next note, onset, chord or combinations thereof given the past events in the sequence, there are many different types of underlying models one can posit to compute predictions and event probabilities. While the setting of predicting the next event is difficult to define accurately and to tackle in the general case of polyphonic music, many past approaches have simplified the problem to tackle a single stream of events, such as melodic notes or chord events. Because this context is similar to the linguistic case, one frequent approach has been to take into account language models as commonly used in computational linguistics, particularly, n-gram models and derived models, which are discussed in Sec. 1.1. More recently, connectionist models have also become increasingly popular, as discussed in Sec. 1.3.

Apart from n-gram and connectionist models musical expectancy and melodic structure have been modelled by other kinds of approaches that we do not discuss in detail. These involve, most notably, hidden Markov models for melodic structure (e.g. [15, 16]) and dynamic Bayesian networks (e.g. [19, 20, 27]). More generally, a large variety of latent structure models beyond Markovian approaches may be adequate to characterize prediction and to compute sequential event probabilities.

1.1 n-gram Models

Markovian and similar approaches have been applied since decades for the modeling of music (see [22] for a review). n-gram models track the number of times a particular contiguous sub-sequence of events occurs in the data. In simple n-gram models the predictive distribution is computed by normalizing these frequency counts for a fixed context length n. Such n-gram models have been applied particularly in the modelling of melody [5, 21, 24] and harmony

[26, 30, 33, 36, 37], classification [4, 9], or applications such as style description or identification [18, 28].

A major problem with this approach is that short context lengths are unable to capture longer patterns while longer context lengths overfit on the training data by assigning zero probability to unseen sequences. Two common methods to overcome this problem are (1) *escaping* strategies, which explicitly assign non-zero weights to unseen sequences and (2) *smoothing* methods, which combine multiple n-gram models of different, possibly unbounded context length (see [24] for an extensive overview).

Extending common n-gram models, Conklin and Witten [5] proposed the notion of multiple viewpoint systems, which combine n-gram models over different basic and derived features in order to improve melodic prediction taking into account correlations between different features. Pearce [21] extended this idea forming the basis of IDyOM, a cognitive model of melodic prediction and generation, which was evaluated with human psychological data [23].

1.2 Long-Term and Short-Term Model

Conklin and Witten [5] proposed the distinction between a long-term model (LTM) and a short-term model (STM). The LTM is trained on a corpus of data and is supposed to capture piece-independent characteristics of the corresponding style, epoch, or genre. The STM, on the other hand, is supposed to capture properties of a single piece like motives or repetitions and is trained online at prediction time for each piece separately. LTM and STM are combined at prediction time (see Sec. 2.6 for details).

1.3 RTDRBM (Connectionist Models)

Recently, Cherla et al. [3] used a *recurrent temporal discriminative restricted Boltzmann machine* (RTDRBM) as LTM, improving over the to-date best performing n-gram LTMs. RTDRBM are state-full connectionist models similar to recurrent neural networks (RNNs), which have the potential to capture long-term dependencies in time series data. This renders them particularly suited for the LTM.

2. THE *PULSE* FRAMEWORK FOR MUSIC

We employ the *PULSE* learning framework [14] for discovering relevant musical features. *PULSE* performs a guided search through an infinitely large feature space and thereby allows to discover features in spaces that are too large for classical feature selection approaches. In doing so, *PULSE* iteratively performs (forward) feature expansion and (backward) feature selection, resulting in a framework similar to evolutionary algorithms. We will first describe the general principles of discovering features with *PULSE* in Sec. 2.1 before going into details about our specific implementation.

2.1 Discovering Features with *PULSE*

PULSE addresses the ubiquitous machine learning problem that, on the one hand, we need to include task-specific prior knowledge to efficiently solve a learning task but, on the other hand, explicitly specifying a set of features might neither be possible nor desirable for a number of reasons: (a) We may lack the explicit knowledge required to specify good features. (b) The specified features may be too specific and "overfit" on a single problem instance. (c) Explicitly specifying features is tedious work to be done by experts, which we might want to automate.

More precisely, instead of explicitly specifying features, in *PULSE* we specify a generative operation N^+ that suggests new candidate features based on the current feature set. N^+ may inject new features as well as mutate and recombine existing features, analogous to an evolutionary algorithm. After expanding the feature set by including all candidate features suggested by the N^+ operation, *PULSE* shrinks the feature set by optimizing the underlying model and selecting features based on the model performance. Again this is akin to evolutionary algorithms with the difference that *PULSE* defines an objective based on the whole feature set and features are thus not scored individually but selected based on how much they contribute to the fitness of the whole population. For learning an optimal set of features and parameters, *PULSE* repetitively expands the feature set by applying N^+ and selects a subset of features by optimizing the model parameters Θ and removing features with zero weight.

As the underlying model, *PULSE* uses a conditional random field [12], which defines a conditional probability distribution $p(x|y)$ as

$$p(x|y) = \frac{1}{Z(y)} \exp \sum_{f \in \mathcal{F}} \theta_f f(x, y) \tag{1}$$

$$Z(y) = \sum_{x' \in \mathcal{X}} \exp \sum_{f \in \mathcal{F}} \theta_f f(x', y) \,, \tag{2}$$

where $y \in \mathcal{Y}$ is known at prediction time, $x \in \mathcal{X}$ is to be predicted, $\mathcal{F}$ is the set of features with weights $\Theta = \{\theta_f \in \mathbb{R} \mid f \in \mathcal{F}\}$, and the partition function $Z(y)$ ensures correct normalization of the conditional distribution. The features $f \in \mathcal{F}$ may be arbitrary real-valued functions of x and y, $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$. When modeling sequential data, $x \in \mathcal{X}$ is the next event, $y \in \mathcal{Y} \equiv \mathcal{X}^*$ is the sequence of past events, and $\mathcal{X}$ is called the symbol space or the alphabet. The parameters are optimized by performing (stochastic) gradient descent on the negative log-likelihood of the data

$$\ell(\Theta; D) = -\sum_{(x,y) \in D} \log p(x|y) + \rho(\Theta) \,, \tag{3}$$

where $\rho(\Theta)$ comprises any regularization terms, most notably an L_1-regularization to enforce a sparse feature set. Note that since ρ implements a prior over the model parameters, specifying additional regularization terms is another means to inject task-specific knowledge in addition to N^+ (also see Sec. 2.5). For optimization we use *AdaGrad* [7] combined with the approach described by Tsuruoka, Jun'ichi Tsujii, and Ananiadou [35] for implementing the L_1-regularization. We will interchangeably speak of maximizing the data likelihood or minimizing the model cross-entropy as both objectives are equivalent.

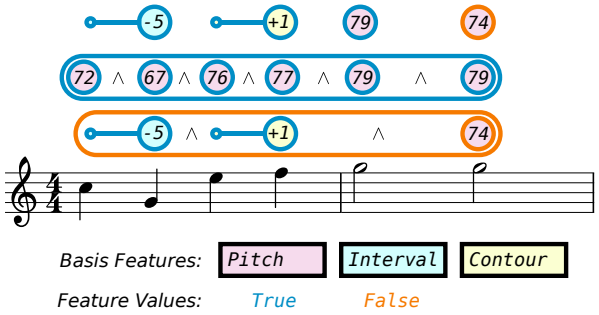

Figure 1. Visualization of generalized n-gram features constructed from three different viewpoints with basis features (top row), a classical n-gram feature (middle row), and a generalized n-gram feature (bottom row). See text for detailed explanation.

We use *PULSE* to discover two kinds of features: Viewpoint features (Sec. 2.2), which generalize the concept of classical n-grams, and Anchor features (Sec. 2.3), which incorporate the concepts of tonic and mode (key) that is common in tonal music.

2.2 Viewpoint Features

Viewpoint or n-gram features indicate the presence of a specific sequence of events. Compared to classical n-grams, the approach described in the following defines a more general set of features, namely ones that also include partial sequences (i.e. sequences with "holes") and sequences that mix different viewpoints. We sometimes differentiate between these generalized and classical non-generalized n-gram features.

Any viewpoint ξ implies a particular alphabet $\mathcal{X}_\xi$. We define an associated set of basis features B_ξ such that feature $b_{(x,t)}(s) \in B_\xi$ indicates whether symbol $x \in \mathcal{X}_\xi$ occurred at time $T-t$ in the given sequence $s \in \mathcal{X}_\xi^*$ of length T. The feature $b_{(x,t)}$ thus looks t steps into the past, indicating the occurrence of event x at that time, with $b_{(x,0)}$ referring to the next event to be predicted. In the upper part of Tab. 1 we list all viewpoints that we used for constructing n-gram features.

Picking up on the construction method suggested in [14] we construct more complex features via logical conjunction of multiple basis features. Features constructed in this way are more general than classical n-grams in two respects: (1) They do not necessarily contain a contiguous sequence of basis features, which allows to generalize over events at a specific time by leaving a "hole". (2) They may be composed of basis features derived from different alphabets/viewpoints, which allows to define a specific combination of viewpoints at one point in a sequence and ignore some of the viewpoints at others. In Sec. 2.4 we describe the specific generative N^+ operations that we use in detail.

An illustration of possible features constructed in this way is given in Fig. 1. Here, we used pitch, interval and contour (P, I, C in Tab. 1) as viewpoints. The last tone in the sequence, the rightmost G5, is to be predicted and has time index $t = 0$. As indicated by the colored border, features evaluate to `true` (or 1) if they match the data and to `false` (or 0) otherwise. In the top row, the basis features $b_{(I=-5,t=4)}$, $b_{(C=+1,t=2)}$, $b_{(P=79,t=1)}$ and $b_{(P=74,t=0)}$ are depicted. Note that interval and contour features not only depend on the pitch at time t but additionally on the previous pitch at time $t-1$. By concatenating basis features from a single viewpoint we can construct classical n-gram features, as illustrated for a pitch n-gram feature in the middle row, which indicates the pitch sequence $72, 67, 76, 77, 79, 79$. If we combine basis features from different viewpoints in a non-contiguous way, we end up with a generalized n-gram feature, as shown in the bottom row. This feature indicates a sequence that starts with a 5-semitone step down, followed by an arbitrary tone from which it rises by an arbitrary interval, again followed by an arbitrary tone, and finally terminates on a D5. As the actual sequence terminates on a G5 this feature evaluates to `false`/0 in this specific case.

2.3 Anchor Features

Anchor features allow to incorporate the concept of tonic and mode, that is key, into our model. They essentially are interval features where the value is not defined with respect to the previous tone but with respect to an anchor tone that may be computed based on any information available at prediction time.

We use three kinds of anchor features that introduce an increasing amount of prior knowledge about tonal music, as listed in the bottom part of Tab. 1. The F_i features use the i^{th} tone of the current piece as reference tone, which is trivial to compute, does not change during the piece and ignores the mode. In many cases the tonic is among the first tones of a piece. A more sophisticated approach is to estimate the tonic based on all tones heard so far using the key-finding algorithm by Krumhansl [11] with parameters from [34]. This is realized in the T features, which may thus change during a piece (even though a change usually only occurs within the first couple of tones) but still ignore the mode. As the employed key-finding algorithm also estimates whether the piece is in major or minor mode it is straightforward to include this distinction, which is realized in the K features. It is interesting to note that in contrast to n-gram features, which have an arbitrary yet fixed length, anchor features incorporate information from the entire history dating back to the very first tone in a piece.

Just as with viewpoint features it is possible to form logical conjunctions of anchor features (anchored generalized n-gram features), however, in this work we confine ourselves to including only single (unigram) anchor features.

2.4 N^+ Operation

In this section we describe the different generative N^+ operations we use to search through the space of generalized n-gram features. The role of the N^+ operation is to suggest new candidate features that are included in the feature set if they improve the model (see Sec. 2.1). Our N^+ operations will inject new basis features (unigrams) and sug-

	Abbrev.	Name	Value Range	Description
Viewpoint Features	P	pitch	$\mathcal{P}$	MIDI pitch of the current note
	I	interval	$\mathcal{I}$	pitch difference between current and previous note
	C	contour	$\{-1, 0, 1\}$	sign of the interval
	X	extended contour	$\{-2, -1, 0, 1, 2\}$	like C but ± 2 for intervals larger than ± 5 steps
Anchor Features	F_i	i^{th} in piece	$\mathcal{I}$	pitch differences to the i^{th} tone in the current piece
	T	tonic	$\{0, \ldots, 11\}$	octave invariant pitch difference to the tonic
	K	key	$\{\mathrm{maj}, \mathrm{min}\} \times \{0, \ldots, 11\}$	like T but separate for major and minor keys

Table 1. Set of employed basis features. $\mathcal{P}$ corresponds to the prediction alphabet, that is, the set of possible MIDI pitches. $\mathcal{I} = \{a - b \,|\, a, b \in \mathcal{P}\}$ is the set of all possible intervals in $\mathcal{P}$.

gest new candidates by taking existing features and adding a new basis feature via logical conjunction. In general, we will apply a combination of multiple N^+ operations to learn our models.

More precisely, we write ξ to refer to an N^+ operation that adds all basis features $b_{(x,0)} \in B_\xi$ of the corresponding viewpoint for time zero to the feature set. Whether the symbol ξ refers to the viewpoint or the corresponding N^+ operation should be clear from context or is explicitly stated otherwise. Likewise, the operators F_i, T, and K add the corresponding anchor features to the feature set. We write ξ^* for an N^+ operation that expands existing features by adding another basis feature from the corresponding viewpoint. The ξ^* operation ignores features corresponding to viewpoints other than ξ. When applying the ξ^* operation we implicitly assume that the ξ operation is also applied (after ξ^*) without explicitly stating it.

Our ξ^* operations come in two versions, as *backwards expansion* for the LTM and as *forwards expansion* for the STM.

For *backwards expansion* (LTM), in the i^{th} iteration of feature expansion, ξ^* expands all existing features with all basis features $b_{(x,i)} \in B_\xi$ with time i. That is, if we were not to remove any features from the set, after n iterations of *backwards expansion*, ξ^* would have constructed all possible generalized n-gram features (with and without "holes") for alphabet $\mathcal{X}_\xi$.

For *forwards expansion* (STM), ξ^* first shifts all existing features by one time step to the past and then expands them with all basis features $b_{(x,0)} \in B_\mathcal{X}$ for time zero. If we were not to remove any features, after n iterations of *forwards expansion*, ξ^* would have constructed all possible *non-generalized* n-gram features (only those without "holes") for alphabet $\mathcal{X}_\xi$.

Backwards and forwards expansion take into account the different learning scenarios for LTM and STM. For the LTM all data is known from the beginning and backwards expansion successively suggests n-gram features of increasing context length until the model stops improving. In contrast, for the STM new data keeps coming in and we construct n-gram features on-the-fly by performing forward expansion once per time step (see Sec. 3). This ensures that (1) short n-gram features can be rebuilt from scratch to account for new data and (2) if an existing n-gram feature captures a motive in the piece, all possible continuations are considered as new candidate features in the next time step.

2.5 Regularization

The purpose of the regularization terms $\rho(\Theta)$ in the *PULSE* objective is twofold: (1) It limits growth of the feature set via L_1-regularization. (2) It implements a prior/bias, which shapes the model characteristics and is a means to prevent overfitting. We use a regularization of the form

$$\rho(\Theta) = \sum_{f \in \mathcal{F}} \Big[|\theta_f| \rho_{L_1}(f) + |\theta_f|^2 \rho_{L_2}(f) \Big] \,, \qquad (4)$$

where $\rho_{L_1}(f)$ and $\rho_{L_2}(f)$ compute the L_1 and L_2 regularization independently for each feature f. For the L_1-regularization in our LTM we follow the rationale that the further back a note lies in time, the less impact it has on the prediction of the current note. This means that longer context lengths risk to overfit on the training data and should be regularized more strongly. We did not observe a significant improvement from applying an additional L_2-regularization in the LTM and use

$$\text{LTM:} \quad \rho_{L_1}(f) = \lambda_1 \, e^{\tau_f / \epsilon} \qquad (5)$$

$$\rho_{L_2}(f) = 0 \,, \qquad (6)$$

where τ_f is the temporal extent of feature f (i.e. the maximum time index of the basis features), ϵ determines how quickly the regularization kicks in for increasing temporal extent, and λ_1 determines the overall regularization strength. For the STM we use

$$\text{STM:} \quad \rho_{L_1}(f) = \lambda_1 \, e^{-t/r_1} \, e^{\tau_f / \epsilon} \qquad (7)$$

$$\rho_{L_2}(f) = \lambda_2 \, e^{-t/r_2} \,, \qquad (8)$$

which implements the same idea with two crucial modifications: (1) The overall regularization strength decays exponentially as more data becomes available, where $r_{1/2}$ are the decay rates and t is the current time index in the song during online training of the STM. (2) We use an additional L_2-regularization, which impedes sparsity but was found to improve the STM performance especially in the initial phase.

The structure of these regularization functions was the result of preliminary runs. Parameters are chosen as described in Sec. 3.

2.6 Combining LTM and STM

LTM and STM are combined by computing a weighted arithmetic mean of their predictive distributions [25]

$$p(x|y) \propto \frac{\sum_{m \in M} w_m p_m(x|y)}{\sum_{m \in M} w_m} \tag{9}$$

where M is the set of available models, which in principle may contain more than just two models. The weights are computed based on how "certain" a given model is, as measured by the entropy of its predictive distribution

$$w_m = \left[\frac{-\sum_{x \in \mathcal{X}} \log p_m(x|y)}{\log |\mathcal{X}|} \right]^{-b}, \tag{10}$$

where division by $\log |\mathcal{X}|$ (the maximum possible entropy) ensures that weights are in $[0, 1]$, and $b \geq 0$ is a bias parameter that allows to shift weights towards models with lower entropy.

3. EXPERIMENTS

We evaluate our models on a corpus of eight symbolic music datasets, as used in [1–3, 24]. The corpus consists of 1009 monophonic folk melodies from different countries and styles as well as the soprano lines of 185 Bach chorales. The data is parsed from the ***kern* format using the Python toolkit *music21* [6]. As input for our model, the melodies are represented as monophonic chromatic pitch sequences with ties being merged. Performance is indicated by the model *cross-entropy* measured in bits. We perform 10-fold cross-validation on each corpus separately using the same folds as [2, 24]. The overall model performance is computed by first computing the cross-entropy for the test set in each cross-validation fold, then averaging over the folds within one corpus, and finally averaging over the different corpora (this is the same approach as in previous work).

Optimization of the feature weights is done using *AdaGrad* [7] with a constant learning learning rate of $\eta = 1$ and initial gradient squared accumulators of $g = 10^{-10}$.

LTM: We evaluate our LTM using the different N^+ operations listed in Tab. 2. Our best performing LTM is compared to the state-of-the-art (see Tab. 3). The feature set is expanded until less than 1% of the features change. The LTM hyperparameter ϵ was fixed to $\epsilon = 1/\ln(2) \approx 1.44$ in preliminary runs while λ_1 is optimized for every cross-validation fold by leaving out 10% of the training data as validation set and performing a Gaussian process based optimization for $\lambda_1 \in [10^{-9}, 10^{-6}]$ using the framework Scikit-Optimize [1].

STM: For the STM we combine the N^+ operations P, I* and F_1. The feature set is expanded once per time step followed by an optimization of the feature weights until convergence. The hyperparameters were set to the following values based on preliminary runs: $\lambda_1 = 10^{-5}$, $r_1 = 100$, $\epsilon = 1/\ln(1.2) \approx 5.48$, $\lambda_2 = 10^{-2}$, $r_2 = 8$.

[1] `https://scikit-optimize.github.io`

PULSE-LTM				
P	I*	C	–	2.701
		X*	–	2.692
		C*	–	2.692
			F_1	2.620
			$F_1F_2F_3$	2.602
			T	2.586
			K	**2.547**

Table 2. Performance of *PULSE*-LTM for different configurations.

	LTM	STM	Hybrid
PULSE	**2.547**	**3.094**	**2.395**
RTDRBM [3]	2.712	3.363	2.421
n-gram [24]	2.878	3.139	2.479

Table 3. Comparison of best performing *PULSE*, RTDRBM, and n-gram models.

Hybrid: For the hybrid model we combine our (PI*C*K)-LTM with our (PI*F_1)-STM. We also test the combination of our LTM with an (C*I) n-gram STM model from the IDyOM-framework [21]. The bias parameter b was determined over the grid $b \in \{0, 1, 2, 3, 4, 5, 6, 16, 32\}$ on the training set of each cross-validation fold.

4. RESULTS

The chief results are that

1. Our *PULSE*-LTM outperforms the current state-of-the-art, RTDRBM [2].
2. Our *PULSE*-STM outperforms the current state-of-the-art, X*UI n-gram [24].
3. Our LTM/STM-hybrid model outperforms the current state-of-the-art, RTDRBM/n-gram [3].
4. The discovered features and learned weights provide musically interpretable insights into the model.

We will now discuss these results in more detail.

4.1 LTM Configurations

In Tab. 2 we list the results for our different LTM configurations. In preliminary runs we identified PI*C to be the minimum setup for outperforming previous work. Performance is improved by expanding contour features (C → C*) in addition to intervals, enabling the model to learn melody contours in addition to transposition invariant motifs. Interestingly, the distinction of small and large intervals using extended contour features, X*, which is considered relevant in music theory, did not result in further improvement.

As expected, incorporating an increasing amount of prior knowledge about tonic and key via F_1, $F_1F_2F_3$, T,

and K, respectively, led to significant improvements. Our best LTM configuration, PI*C*K, significantly improves over the current state-of-the-art, with the performance gain being of the same magnitude as was achieved by the current state-of-the-art RTDRBM [2] versus n-grams [24].

It is interesting to note that the algorithm for computing T and K involves a combination of music-theoretical insights and empirical tone profiles. An important future research question is how this can be generalized and made accessible to learning from corpus data.

4.2 STM Performance

In Tab. 3 we compare the PI*F_1 *Pulse*-STM with the best RTDRBM [3] and n-gram STM [24]. To our knowledge we present the first model that outperforms the well-established n-gram STM for the task of sequential prediction of symbolic monophonic music. We assume that an even better performance is achievable by performing a more thorough (yet very expensive) optimization of the STM hyperparameters.

4.3 Hybrid

Combining the *Pulse*-LTM with the *Pulse*-STM gives a performance boost, outperforming current state-of-the-art hybrids (see Tab. 3). The combination of our *Pulse*-LTM with an C*I n-gram STM from the IDyOM-framework yields an interesting result: While the n-gram STM alone performs worse (3.152 bits) than our *Pulse* STM the corresponding hybrid displays a better performance of 2.365 bits. We conjecture that the n-gram STM has complementary properties to our *Pulse*-based model and therefore is able to contribute valuable missing information. The best performing LTM/STM-hybrid on this corpus thus is the combination of our PI*C*K *Pulse*-LTM and the C*I n-gram STM.

4.4 Discussion of Features

While an extensive discussion of all features for the different corpora is beyond the scope of this paper, we show a qualitative plot of the weights for a subset of features learned by our PI*C*K *Pulse*-LTM from the Bach chorale corpus in Fig. 2. First, we see that the pitch features (P) describe a general preference for tones in the middle register. For the interval features (I) we restrict ourselves to length-one features, which show a preference of small (especially descending) steps over large steps. This is in accordance with general music-theoretic principles of voice leading. Note that a tritone step (±6 semitones) is particularly discouraged. The anchor features (K) model separate tone profiles for major (M) and minor (m) modes. We empirically confirm a preference for relevant in-scale tones: tonic (0), major third (4)/minor third (3) and fifth (7), whereas the out-of-scale tones minor second (1), minor third(3)/major third (4), tritone (6), and minor seventh (10)/major seventh (11) are discouraged.

During training of the model, a total of 5851 features was temporally included from which 322 remained in the final model. This underlines the relevance of performing both feature expansion *and* selection, which allows *Pulse* to scale to very large feature spaces.

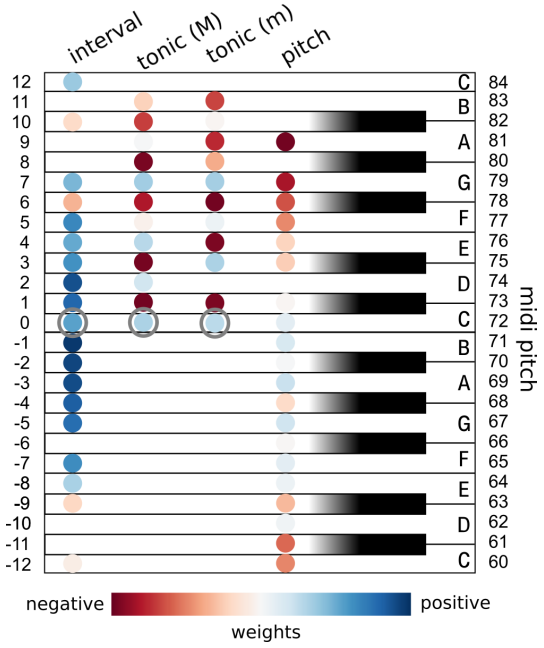

Figure 2. Qualitative plot of the feature weights for P and I features of length one as well as K features, learned based on the Bach chorales. For I and K features middle C is chosen as reference tone as marked by the circles.

5. CONCLUSION

We applied the *Pulse* framework to the problem of learning a model for sequential prediction of symbolic monophonic music. Our models outperform the current state-of-the-art for long-term, short-term and hybrid models on a standard benchmark corpus of folk melodies and Bach chorales. At the same time our approach affords interpretable models that use an explicit set of musically relevant features. The size of the processed feature spaces are challenging for classical feature expansion methods and our method has the potential to scale to even larger spaces. This becomes particularly relevant for an application to polyphonic music and modeling of harmony as well as for including more complex viewpoints.

This is the first application of the *Pulse* framework for modelling music, which provides excellent results and opens up a number of possible directions for further investigation. We therefore consider *Pulse* to be a promising framework for the development of a unified architecture for modelling music.

6. REFERENCES

[1] Srikanth Cherla et al. “A Distributed Model For Multiple-Viewpoint Melodic Prediction”. In: *International Society for Music Information Retrieval (ISMIR)*. 2013, pp. 15–20.

[2] Srikanth Cherla et al. “Discriminative learning and inference in the Recurrent Temporal RBM for melody modelling”. In: *Neural Networks (IJCNN), International Joint Conference on*. IEEE. 2015, pp. 1–8.

[3] Srikanth Cherla et al. “Hybrid Long-and Short-Term Models of Folk Melodies.” In: *International Society for Music Information Retrieval*. 2015, pp. 584–590.

[4] Darrell Conklin. “Multiple viewpoint systems for music classification”. In: *Journal of New Music Research* 42.1 (2013), pp. 19–26.

[5] Darrell Conklin and Ian H. Witten. “Multiple Viewpoint Systems for Music Prediction”. In: *Journal of New Music Research* 24.1 (1995), pp. 51–73.

[6] Michael Scott Cuthbert and Christopher Ariza. “music21: A toolkit for computer-aided musicology and symbolic music data”. In: *International Society for Music Information Retrieval (ISMIR)*. 2010, pp. 637–642.

[7] John Duchi, Elad Hazan, and Yoram Singer. “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization”. In: *Journal of Machine Learning Research* 12.Jul (2011), pp. 2121–2159.

[8] Morwaread M Farbood. “A parametric, temporal model of musical tension”. In: *Music Perception: An Interdisciplinary Journal* 29.4 (2012), pp. 387–428.

[9] Ruben Hillewaere, Bernard Manderick, and Darrell Conklin. “Global Feature Versus Event Models for Folk Song Classification.” In: *International Society for Music Information Retrieval (ISMIR)*. 2009.

[10] David Huron. *Sweet Anticipation: Music and the Psychology of Expectation*. Cambridge, Massachusetts: MIT Press, 2006.

[11] Carol L. Krumhansl. *Cognitive Foundations of Musical Pitch*. Oxford Psychology 17. New York / Oxford: Oxford University Press, 1990.

[12] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. “Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data”. In: *Proceedings of the Eighteenth International Conference on Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 282–289.

[13] Moritz Lehne et al. “The influence of different structural features on felt musical tension in two piano pieces by Mozart and Mendelssohn”. In: *Music Perception: An Interdisciplinary Journal* 31.2 (2013), pp. 171–185.

[14] Robert Lieck and Marc Toussaint. “Temporally Extended Features in Model-Based Reinforcement Learning with Partial Observability”. en. In: *Neurocomputing* 192 (2016), pp. 49–60.

[15] Panayotis Mavromatis. “A hidden Markov model of melody production in Greek church chant”. In: *Computing in Musicology* 14 (2005), pp. 93–112.

[16] Panayotis Mavromatis. “Minimum description length modelling of musical structure”. In: *Journal of Mathematics and Music* 3.3 (2009), pp. 117–136.

[17] L B Meyer. *Emotion and Meaning in Music*. London: University of Chicago Press, 1956.

[18] M. Ogihara and T Li. “N-Gram chord profiles for composer style identification”. In: *International Society for Music Information Retrieval (ISMIR)*. 2008, pp. 671–676.

[19] Jean-Francois Paiement, Samy Bengio, and Douglas Eck. “Probabilistic models for melodic prediction”. In: *Artificial Intelligence* 173.14 (2009), pp. 1266–1274.

[20] Jean-Francois Paiement, Yves Grandvalet, and Samy Bengio. “Predictive models for music”. In: *Connection Science* 21.2-3 (2009), pp. 253–272.

[21] Marcus T. Pearce. “The Construction and Evaluation of Statistical Models of Melodic Structure in Music Perception and Composition”. PhD thesis. Department of Computing, City University, London, UK, 2005.

[22] Marcus T Pearce and Geraint A Wiggins. “Auditory expectation: The information dynamics of music perception and cognition”. In: *Topics in cognitive science* 4.4 (2012), pp. 625–652.

[23] Marcus T Pearce and Geraint A Wiggins. “Expectation in melody: The influence of context and learning”. In: *Music Perception: An Interdisciplinary Journal* 23.5 (2006), pp. 377–405.

[24] Marcus T. Pearce and Geraint A. Wiggins. “Improved Methods for Statistical Modelling of Monophonic Music”. In: *Journal of New Music Research* 33.4 (2004), pp. 367–385.

[25] Marcus Pearce, Darrell Conklin, and Geraint Wiggins. “Methods for Combining Statistical Models of Music”. In: *International Symposium on Computer Music Modeling and Retrieval*. Springer, 2004, pp. 295–312.

[26] Dan Ponsford, Geraint Wiggins, and Chris Mellish. “Statistical learning of harmonic movement”. In: *Journal of New Music Research* 28.2 (1999), pp. 150–177.

[27] Stanislaw Raczynski et al. “Multiple pitch transcription using DBN-based musicological models”. In: *International Society for Music Information Retrieval (ISMIR)*. 2010, pp. 363–368.

[28] M. Rohrmeier and I. Cross. “Statistical Properties of Harmony in Bach’s Chorales”. In: *10th International Conference on Music Perception and Cognition (ICMPC)*. 2008, pp. 619–627.

[29] Martin A Rohrmeier and Stefan Koelsch. “Predictive information processing in music cognition. A critical review”. In: *International Journal of Psychophysiology* 83.2 (2012), pp. 164–175.

[30] Martin Rohrmeier and Thore Graepel. “Comparing feature-based models of harmony”. In: *Proceedings of the 9th International Symposium on Computer Music Modelling and Retrieval*. Citeseer. 2012, pp. 357–370.

[31] Martin Rohrmeier and Patrick Rebuschat. “Implicit learning and acquisition of music”. In: *Topics in cognitive science* 4.4 (2012), pp. 525–553.

[32] Jenny R Saffran et al. “Statistical learning of tone sequences by human infants and adults”. In: *Cognition* 70.1 (1999), pp. 27–52.

[33] R. Scholz, E. Vincent, and F. Bimbot. “Robust modelling of musical chord sequences using probabilistic n-grams”. In: *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2009, pp. 53–56.

[34] David Temperley. “What’s Key for Key? The Krumhansl-Schmuckler Key-Finding Algorithm Reconsidered”. In: *Music Perception: An Interdisciplinary Journal* 17.1 (1999), pp. 65–100.

[35] Yoshimasa Tsuruoka, Jun’ichi Tsujii, and Sophia Ananiadou. “Stochastic Gradient Descent Training for L1-Regularized Log-Linear Models with Cumulative Penalty”. In: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. Suntec, Singapore: Association for Computational Linguistics, 2009, pp. 477–485.

[36] Raymond P Whorley and Darrell Conklin. “Music generation from statistical models of harmony”. In: *Journal of New Music Research* 45.2 (2016), pp. 160–183.

[37] R. Whorley et al. “Multiple Viewpoint Systems: Time Complexity and the Construction of Domains for Complex Musical Viewpoints in the Harmonisation Problem”. In: *Journal of New Music Research* 42 (2013), pp. 237–266.