A data-driven model of tonal chord sequence complexity

Di Giorgi, B; Dixon, S; Zanoni, M; Sarti, A

A data-driven model of tonal chord sequence complexity

Bruno Di Giorgi, Simon Dixon, Massimiliano Zanoni, and Augusto Sarti, *Senior Member, IEEE,*

***Abstract*—We present a compound language model of tonal chord sequences, and evaluate its capability to estimate perceived harmonic complexity. In order to build the compound model we trained three different models: prediction by partial matching, a hidden Markov model and a deep recurrent neural network, on a novel large dataset containing half a million annotated chord sequences. We describe the training process and propose an interpretation of the harmonic patterns that are learned by the hidden states of these models. We use the compound model to generate new chord sequences and estimate their probability, which we then relate to perceived harmonic complexity. In order to collect subjective ratings of complexity, we devised a listening test comprising two different experiments. In the first, subjects choose the more complex chord sequence between two. In the second, subjects rate with a continuous scale the complexity of a single chord sequence. The results of both experiments show a strong relation between negative log probability, given by our language model, and the perceived complexity ratings. The relation is stronger for subjects with high musical sophistication index, acquired through the GoldMSI standard questionnaire. The analysis of the results also includes the preference ratings that have been collected along with the complexity ratings; a weak negative correlation emerged between preference and log probability.**

***Index Terms*—Harmonic complexity, Chord sequences, Language models**

I. Introduction

THE estimation of complexity of musical content is among the various tasks of Music Information Retrieval (MIR). Numerous studies about complexity hypothesized a tight relationship between complexity and preference [1], [2], [3], arousal [4] and cultural evolution of music [5].

However, the automatic estimation of perceived complexity from music is an ill-conditioned problem. In fact, while everyone can think of relatively complex or simple songs, assigning a precise meaning to the word "complexity" is as hard as describing our mental representation of music. Nevertheless, it is reasonable to address different musical facets separately, assuming that listeners resort to many concurrent and hierarchical models to globally represent the perceived acoustic signal [6]. In this work we focus on harmony and propose a data-driven language model of tonal chord sequences, showing its ability to automatically estimate the perceived complexity and preference. In particular, we show that the probability of a chord sequence, given by a language model trained on a sufficiently large dataset, can be related to these perceptual quantities. In order to quantitatively evaluate our hypothesis we collected the ratings of complexity and preference obtained from a listening test. Since we deal with the harmonic complexity of tonal chord sequences, we provide hereafter a definition of tonal harmony and a hierarchy of the main interpretations of harmonic complexity.

Bruno Di Giorgi Massimiliano Zanoni and Augusto Sarti are with the Dipartimento di Elettronica ed Informazione, Politecnico di Milano, 20133, Milan, Italy (e-mail: bruno.digiorgi@polimi.it; augusto.sarti@polimi.it)

Simon Dixon is with the Centre for Digital Music, School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4NS, U.K. (e-mail: s.e.dixon@qmul.ac.uk).

Tonal Harmony (TH) has been the main theoretical framework for harmony in Western music since the baroque period (around 1650). It implies that there is a certain pitch class, called the tonic, acting as a referential point. The building blocks of this framework are the chords, whose function can be established by the interval between their root and the tonic [7]. It is also important to stress that it has been shown that even musically untrained listeners possess a kind of unconscious natural model of tonal harmony [8], [9].

A taxonomy of different sources of harmonic complexity is proposed by Temperley [6] and includes harmonic rhythm, harmonic dissonance and harmonic evolution.

Harmonic rhythm consists of the rate of chord changes as well as their position within the meter. Both quantities are assumed to correlate with tension [6].

Harmonic dissonance is a property of the intervals between simultaneously sounding pitches. Weiss dealt with similar approaches, proposing a number of dissonance features in order to analyze different sections of Beethoven's sonatas [10] and classify four historical music periods [11]. In [12] a chroma complexity measure is used to predict the year of a set of last century top hits.

Harmonic evolution encompasses the dynamic properties of harmony, and in particular harmonic priming, by which the listener develops expectations of what is to come next, given a harmonic context. If the expectation is confirmed by the following harmony, perception will be easier and quicker [4], [13]. Models belonging to this class may be further subdivided based on how the expectation is formed: the *sensory expectation* is formed using low-level properties of the context, e.g. implying temporal continuity on the pitch class profiles, while the *cognitive expectation* is formed from using a higher level (more abstract) representation of the context, e.g. using rules of a given musical idiom such as tonal harmony [14].

Methods for estimating harmonic complexity based on sensory expectations rely on distance metrics between adjacent Pitch Class Profiles (PCPs, also called chroma). For example, Mauch [5] introduces and applies a structural change measure directly to chroma. The relationship of structural change with complexity is hypothesized but left for future work. Streich

[15] evaluates the distance between a PCP with short time span and one with longer time span. The comparison is computed with both the Pearson correlation coefficient and city block distance. Among other PCP similarity metrics used for estimating chord expectations is the spectral pitch distance described in [16], that implements ideas of perceptual affinity from the model of Parncutt [17]. In [18], chord expectations are formed favoring temporal continuity of chord voices, considering multiple temporal scales.

Methods focusing on cognitive harmonic expectations estimate the amount of surprise using rules from music theory or machine learning models. In [19], [20] the surprise is estimated as the number of applications of two rules, which comply with music theory, to transform the previous chord into the next chord. This measure is then tested on a genre classification problem. In [21], the authors train and compare different machine learning models on a corpus of Jazz chord sequences. In [22], a multiple viewpoint model is used to model prediction in a four part chorale setting. In [23], a machine learning approach is used to compute predictions and a set of chord substitution rules is also inferred from data. However, in [21], [22] and [23], no claim is made about the relationship between surprise and harmonic complexity.

It is also important to mention contributions coming from the field of psychology about harmonic priming. In [8], the authors perform three listening tests with a list of ad hoc chord sequences. Among the other results they prove that reaction times are quicker for expected chords and that priming is independent from musical training. In [14], the prevalence of cognitive priming over sensory priming is experimentally proved, even with very fast harmonic rhythm.

In this article, we design the architecture of a language model of tonal chord sequences that we use to model cognitive expectation. Then, we evaluate its capability to estimate the perceived harmonic complexity. The model is trained on a novel dataset containing approximately half a million chord sequence annotations, therefore being the largest such dataset, to the best of the authors knowledge. The dataset has been collected from the website ultimateguitar.com and contains annotations uploaded by users. Although these data may have questionable accuracy, we believe that the benefits of such a large data set outweigh any effect of the noise in the data. Apart from transcription errors, the differences from high quality annotations may include chord simplification, substitution using similar chord functions and reharmonization, but we cannot quantify them. It can be argued that with such an amount of data, the models would learn the typical chord substitutions, while filtering out the more random errors.

It is known from music theory that we can formulate much better hypotheses on chords when we know the key of the piece, because of the strong relationship between the two entities. In fact, the key is often interpreted as a latent variable in probabilistic models of chords [24], [25], [26], [27]. Therefore, we estimate the tonic as a pre-processing step, and represent chords using a relative notation (i.e. Cmaj as 0maj when the tonic is C). Estimating only the tonic, but not the specific scale or mode, allows the model to learn common chord progression patterns, such as modal interchange and secondary dominants, that exploit notes from different scales or modes. Furthermore, this way we ensure a maximally efficient usage of the models, avoiding a 12-times augmentation of the dataset with the transposed versions of each song.

The machine learning language models used in this work are: Prediction by Partial Matching (PPM) [28], discrete Hidden Markov Model (HMM) [29] and Recurrent Neural Network (RNN) [30]. We investigate to find the optimal combination of these three models, which is the final approach described here. We will then evaluate, with two perceptual tests, that the probability of a chord sequence evaluated by our model is strongly correlated to the high level concept of harmonic complexity.

The remainder of the paper is organized as follows: in Section II we review the background of the tonic identification task and the language models used; Section III discusses the dataset and our tonic identification algorithm; Section IV details the process of training and combination of the chord language models; in Section V we provide the specification and discuss the results of the two perceptual tests; in Section VI we present the conclusions and possible extensions of the work.

II. Background

In this section we provide an overview of the three language models, Prediction by Partial Matching (PPM), Hidden Markov Model (HMM) and Recurrent Neural Network (RNN), used for computing the prediction probability $p(x_i|x_{i-n}^{i-1})$, hence the expectation of a chord x_i, given the sequence of the previous n chords. Such models can compute a distribution over chord sequences, given by:

$$p(x) = p(x_0) \prod_{i>0} p(x_i|x_0^{i-1}) \tag{1}$$

The overview is focused on the prediction task and uses a consistent notation in order to highlight the differences among the models. For each model, we also provide some references on music related works.

A. Prediction by Partial Matching

PPM [28] was originally developed as a method for data compression, assuming a Markov model of the source. The nth-order Markov model is a probabilistic language model where an element depends only on the previous n elements:

$$p(x_i|x_0^N) = p(x_i|x_{i-n}^{i-1}), \tag{2}$$

where x_i^j is the sequence of symbols from i to j. $(n-1)$th-order Markov model sources are modeled using n-grams where the probability of observing a particular symbol x_i is estimated by counting the occurrences in the set of training sequences:

$$p(x_i|x_{i-(n-1)}^{i-1}) = \frac{c(x_i|x_{i-(n-1)}^{i-1})}{\sum_{x\in\mathcal{X}} c(x|x_{i-(n-1)}^{i-1})} \tag{3}$$

where the function $c(g_n|g_1^{n-1})$ counts the occurrences of the n-gram g_1^n in the training set and $\mathcal{X}$ is the alphabet. The

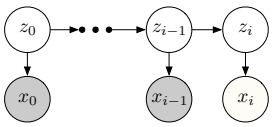


Fig. 1. In this article we use Hidden Markov models for prediction of the next symbol x_i given the previous ones x_0^{i-1}. The observed variables are shaded.

Equation (3) is called the Maximum Likelihood estimate and can lead to poor performance, particularly in the case that some n-grams never occur in the training set, known as the *zero frequency problem*. In PPM, the prediction probability is estimated using also the lower order models. This allows to cope with the sparsity of higher order models, i.e. if a particular n-gram $x_i|x_{i-(n-1)}^{i-1}$ is novel to the $(n-1)$th-order model, the symbol $x_{i-(n-1)}$ is dropped and the $(n-1)$-gram $x_i|x_{i-(n-2)}^{i-1}$ is considered using the $(n-2)$th-order model. This procedure of combining different order models is known as back-off smoothing, and it solves the zero-frequency problem [31] of fixed order n-gram models. A general framework for back-off smoothing [32] is expressed by the following equations:

$$p(x_i|x_{(i-n)+1}^{i-1}) = \\ = \begin{cases} \alpha(x_i|x_{(i-n)+1}^{i-1}) & \text{if } c(x_i|x_{(i-n)+1}^{i-1}) > 0 \\ \gamma(x_{(i-n)+1}^{i-1}) p(x_i|x_{(i-n)+2}^{i-1}) & \text{otherwise,} \end{cases} \tag{4}$$

where $\alpha(x_i|x_{(i-n)+1}^{i-1})$ is an estimate of the probability of an already seen n-gram and the *escape probability* $\gamma(x_{(i-n)+1}^{i-1})$ represents the probability mass assigned to all symbols that are novel in the current context $x_{(i-n)+1}^{i-1}$, in the training set:

$$\sum_{x_i \in \mathcal{X}} \alpha(x_i|x_{(i-n)+1}^{i-1}) + \gamma(x_{(i-n)+1}^{i-1}) = 1. \tag{5}$$

A series of heuristics have been proposed to compute the functions α and γ. As stated in [31], [28], there seems to be no theoretical basis for choosing any particular heuristic. In our experiments (see Sect. IV-A) we tried several approaches [28], [33], [34], [35], as well as some common enhancement techniques such as exclusion [28] and interpolation [36].

In the realm of music, PPM has been used in [37], [22] and [21] as part of a multiple viewpoint model, respectively for extrapolating melody, harmonizations and chord progressions. Fixed order n-grams and n-grams with simpler smoothing techniques have been used in [38] for predicting the next chord of a progression.

B. Hidden Markov Models

HMMs abstract the sequence dependences from the observed sequence x to a hidden sequence z, called the state sequence. The hidden sequence is modeled as a first order Markov model, while each observed element x_i depends only on the current state z_i. Using the d-separation criterion, it is possible to show that the predictive distribution $p(x_i|x_0^{i-1})$ for observation x_i given all previous observations does not exhibit any conditional independence properties [39, p. 372–383]. Therefore, HMMs overcome the Markov model limitation, rendering the prediction of x_i dependent on all previous observations.

HMMs are fully described by three probability distributions: the state transition probability $p(z_i|z_{i-1})$, the emission probability $p(x_i|z_i)$ and the initial state probability $p(z_0)$. In our case these are all categorical distributions, and the maximum likelihood value of their parameters can be learned from a training set of observed sequences using the iterative Expectation-Maximization (EM) algorithm. The hidden states $\mathcal{Z}$ are abstract classes and do not have a semantic meaning. They represent different distributions over the chord space, described by their emission probability. In Sect. IV-A we show that after training by EM, the states converge to model meaningful contexts about chord progressions.

Once the model parameters have been learned, the prediction of x_i, given the sequence x_0^{i-1}, is obtained marginalizing over the hidden variables z_i and z_{i-1}:

$$p(x_i|x_0^{i-1}) = \sum_{z_{i-1}} p(z_{i-1}|x_0^{i-1}) \sum_{z_i} p(z_i|z_{i-1}) p(x_i|z_i). \tag{6}$$

z_{i-1}, in turn, depends on the whole sequence x_0^{i-1}:

$$p(z_{i-1}|x_0^{i-1}) = \frac{p(x_0^{i-1}, z_{i-1})}{p(x_0^{i-1})}. \tag{7}$$

The denominator of Eq. (7) is a normalization factor, and the joint probability $p(x_0^{i-1}, z_{i-1})$ can be obtained by the following recursive relations:

$$p(x_0^i, z_i) = p(x_i|z_i) \sum_{z_{i-1}} p(x_0^{i-1}, z_{i-1}) p(z_i|z_{i-1}) \tag{8}$$

$$p(x_0, z_0) = p(x_0|z_0) p(z_0), \tag{9}$$

known as the forward part of the forward-backward algorithm, since its computation propagates forward from the start of the sequence (Eq. 9) up to the previous frame (Eq. 8), iteratively marginalizing over the hidden variables z_0^{i-2}.

The equations (6), (8) and (9) effectively marginalize over all the hidden nodes of the graphical model of Figure 1. Eq. (8) and (9) are also used in [40] to demonstrate the exponential forgetting rate behavior of this model.

An HMM has been used as a language model for chord prediction in [21]. The common usage for chord recognition [41] is fundamentally different because of the chord sequence being the hidden variable with audio features, commonly the chromagram, as the observed variable. In that case, the HMM can be interpreted as a chromagram language model, using a meaningful hidden state: chords. In our case HMM functions as a chord language model, with an abstract hidden state.

C. Recurrent Neural Networks

RNNs extend standard feed-forward deep neural networks (FNNs), adding the capability of modeling sequential data. FNNs are biologically inspired machine learning models used for classification and regression, where the input data x undergoes a series of non-linear transformations:

$$h_{l+1} = f(W_l h_l + b_l), \tag{10}$$

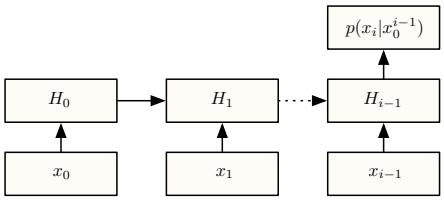


Fig. 2. A Recurrent Neural Network used for the prediction of the next symbol x_i given the previous ones x_0^{i-1}. H represents the stack of the hidden layers $h_l : 1 \leq l \leq L$, which is the state of the model.

where $h_0 = x$; for each of the layers $0 \leq l \leq L$, h_l is the vector of hidden units, W_l is the weight matrix and b_l is the bias vector; f is a non-linear function applied element-wise. For the task of classification with K alternative classes, the last layer h_L is a vector of K units, which is transformed by a softmax non-linearity to yield the probability distribution of each class $p(y|x)$, given the input. The problem with standard FNNs is that no state is maintained between successive inputs.

RNNs introduce the notion of time by including connections between adjacent time steps:

$$h_{l+1}^t = f(W_l^f h_l^t + W_l^r h_l^{t-1} + b_l), \tag{11}$$

where W_l^f is the matrix of weights for the forward propagation of input while W_l^r is the matrix of weights for the recurrent connection between the same layers of units. The set of parameters of the RNN includes W_l^f, W_l^r and b_l for each layer and is trained using the algorithm called back-propagation through time, introduced in [42].

While for FNNs only fixed-length sequences can be used to predict the probability of observing the next symbol, for RNNs there is no such limitation. Since RNNs explicitly model the concept of time, sequences of arbitrary length can be used as a context, similarly to HMMs. The model is trained using *sequences* of (input, target) pairs of examples of the form (x_i, x_{i+1}). Although there is only one symbol as input for each example pair, the sequential nature of the data and the particular model structure ensure that the model will learn to estimate the probability $p(x_i|x_0^{i-1})$ of a given symbol as the next symbol after a given sequence of previous symbols. This is a consequence of the autoregressive nature of the model, meaning that the state is function of the current input and the previous state. Therefore the state used to predict the current output is a representation of the current and all the previous inputs [43].

Once the model is trained, this estimation is performed inputting one symbol at a time and considering only the last output of the network, as shown in Figure 2.

In comparison to HMMs, the state of an RNN is distributed, which allows a larger and more flexible memory capacity [44]: the number of distinct states that can be represented in a hidden layer of nodes grows exponentially with the number of nodes in the layer [30]. An in depth review of RNNs used as language models is provided by De Mulder [43].

RNNs have been previously used in [45], [46] as language models for audio chord recognition. In those works, usage of the RNN is fundamentally different from ours, in terms of purpose (smoothing rather than prediction) and domain of input variables.

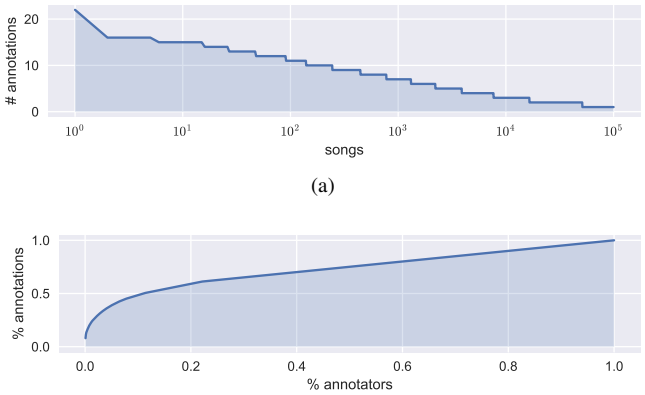


Fig. 3. The top figure shows the number of annotations per song (a) on the entire dataset. The most annotated song is "The A Team" by Ed Sheeran, which is the only one counting 22 annotations. 51k songs have more than one annotation. The bottom figure (b) shows the percentage of annotations created by a percentage of annotators and is computed on the 2575 annotations for which it was possible to extract the name of the annotator. Approximately 20% of the annotators created 60% of the annotations.

III. Dataset

The dataset used in this work contains all the chord annotations of the website *ultimateguitar.com*. Overall, it involves 26545277 chord occurrences stemming from 412924 annotations, referring to approximately 328k songs by 40k authors [1]. 51k songs have been annotated more than once, as showed in Fig. 3a. Moreover, for 2575 annotations it has been possible to extract the name the annotator. For these annotations, approximately 20% of the annotators created 60% of the annotations as showed in Fig. 3b. We expect a similarly shaped distribution for the entire dataset.

Style tags show a significant prevalence of Rock, Pop and Alternative music. 373 different chord type labels were found and manually mapped to 4, thus obtaining an alphabet of 48 symbols: 12 pitch classes for the root (discarding enharmonic information) and 4 chord types. The chosen chord types are `maj`, `min`, `7` and `(5)`, expressed using the syntax proposed in [47]. The first three were chosen because they are, as expected, the most frequently occurring in the dataset (89% of all occurrences), while `(5)` was retained because the mapping to any of the three previous types is not possible without any prior information about the tonality. Generally, the mappings just delete the extensions, for example transforming `maj7` into `maj` and `9` into `7`. In Table I we show the most common chord types found in the annotations, their proportion and the mapping to the chosen alphabet. Some of the mappings required a stronger decision, e.g. turning `dim` into `min` and `aug` into `maj`; regarding these choices, we used the the interval of third as a decision factor, because it gave us fewer and more significant categories than using the fifth.

Approximately 2% of the chords of the dataset have originally been annotated with a bass note that is different from

[1]The dataset is available at http://ispg.deib.polimi.it/mir-software.html

TABLE I
MAIN CHORD TYPE MAPPINGS AND DISTRIBUTION

source	target	%	source	target	%
`Z`	`Z`	64.73	`Z:min/X`	`Z:min`	0.25
`Z:min`	`Z:min`	21.60	`Z:maj6`	`Z`	0.24
`Z:7`	`Z:7`	3.45	`Z:(4,5,b7)`	`Z:7`	0.09
`Z:min7`	`Z:min`	1.99	`Z:dim`	`Z:min`	0.09
`Z:(5)`	`Z:(5)`	1.88	`Z:min9`	`Z:min`	0.07
`Z/X`	`Z`	1.74	`Z:min6`	`Z:min`	0.07
`Z:maj7`	`Z`	0.84	`Z:min7/X`	`Z:min`	0.05
`Z:(3,5,9)`	`Z`	0.83	`Z:maj9`	`Z`	0.05
`Z:sus4`	`Z:(5)`	0.56	`Z:7/X`	`Z:7`	0.04
`Z:(2,3,5)`	`Z:(5)`	0.47	`Z:11`	`Z:7`	0.03
`Z:9`	`Z:7`	0.28	`Z:aug`	`Z`	0.02

the root (for example `C/E`, `A:min/G`). We decided to drop the bass note, assuming that the information contained in it was not substantial enough to justify a much larger alphabet.

A. Tonic identification

In our dataset there is no information about chord onset times or durations. This fact undermines our ability to evaluate the relative importance of the different chords within a sequence. Furthermore, to the best of our knowledge, there exists no method for tonic-identification from chords that deals with the absence of time information. Therefore we designed a simple method for tonic identification tailored for our needs.

The tonic $t \in \{0, \dots, 11\}$, representing pitch classes {'C', 'C#', ..., 'B'} is found by maximizing the Pearson correlation coefficient ρ between the sum X of the pitch class profiles of all the chords in the sequence and a key template $Y(t, m)$ obtained as a cyclic permutation of a mode template φ_m:

$$t = \arg\max_t \left(\max_m \left(\rho(X, Y(t, m)) \pi_m \right) \right) \tag{12}$$

$$X = \sum_{i=0}^{N-1} \beta_{x_i} \tag{13}$$

$$Y_i(t, m) = \varphi_m((t + i) \mod 12) \qquad i = 0, \dots, 11 \tag{14}$$

$$\tag{15}$$

where $\beta_x \in \{0, 1\}^{12}$ is the pitch class profile of a chord x, N is the number of chords in the song, φ_m is the mode template, i.e. the pitch class profile of the mode (see Table II), π_m is a prior distribution over the set of considered modes (the seven diatonic modes). Maximizing over the mode m has the desired effect of considering only the best fitting mode for each tonic t. With reference to Table II, the parameters of the model are the weight τ of the tonic in the mode profiles ($\tau > 1$), and the parameters a, b, c of the distribution π_m, with $2a+2b+2c+d = 1$, (the factor 2 is due to a, b and c being applied to two modes each). Based on empirical estimates of the relative occurrence of the modes in popular music, we set $\tau = 2$, $a = 0.02$, $b = 0.33$, $c = 0.15$, $d = 0$.

In order to qualitatively evaluate the accuracy of our method, we compared it with our implementation of the Temperley model in [48], which is also based on key template matching and achieved the best performance in the symbolic key estimation task at the Music Information Retrieval Evaluation eXchange (MIREX) contest in the 2005 edition, the last one hosting this task. [48] learns key templates from the Kostka-Payne corpus and then correlates them with binary pitch class profile vectors obtained from 1.2 second long frames of a polyphonic representation (Musical Instrument Digital Interface, MIDI). Frame-wise key probabilities are then smoothed using dynamic programming. We obtained the polyphonic representation that [48] requires by aggregating adjacent chords within a moving frame, using max function, and assuming that all chords have the same duration. We noticed negligible difference changing the frame duration (i.e. the number of chords to aggregate), the aggregation function or the overlap between successive frames.

TABLE II
PITCH CLASS PROFILES OF THE DIATONIC MODES

mode name	φ_m	π_m
lydian	$[\tau, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1]$	a
ionian	$[\tau, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1]$	b
mixolydian	$[\tau, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0]$	c
dorian	$[\tau, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0]$	c
aeolian	$[\tau, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0]$	b
phrygian	$[\tau, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0]$	a
locrian	$[\tau, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0]$	d

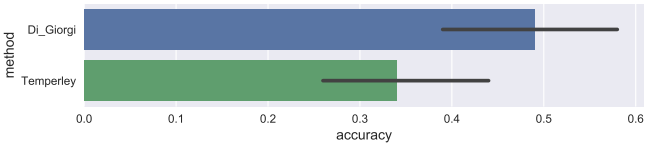


Fig. 4. The mean accuracy of our tonic identification algorithm, compared to our modified implementation of [48]. The modification concerns the computation of the required polyphonic representation starting from the sequence of chords, which are assumed to have constant duration. The results are based on a set of 100 songs where they disagree. 95% confidence intervals are obtained by 1000 iterations of bootstrapping.

For 70% of the songs, our algorithm and [48] agree, estimating the same pitch class as tonic. We randomly selected 100 songs from the remaining set and manually annotated the tonic. An estimation of the mean accuracy of both algorithms in this "disagreement set" revealed a meaningful advantage in using our method for tonic-identification (Figure 4). The same test was repeated using the original and the reduced alphabet of chords, obtaining no noticeable difference in the results. This is due to the predominance of the four chord types included in the reduced alphabet, and generally the invariance of the tonic estimation process to the chord type mappings described in Table I.

After estimating the tonic of every song, we can represent all the chord symbols relatively to the tonic, in terms of the interval (in semitones) between the tonic and the chord's root. A relative notation is usually found in harmony textbooks using roman numerals indicating scale degree ("I", "ii") and brings some advantages with respect to absolute notation (`C:maj`, `D:min`), as also argued in [49], [38]. First, it is closer to our perception of tonal music because similar sounding progressions such as [`C:maj, A:min, D:min, G:7`] and [`G:maj, E:min, A:min, D:7`] are represented by the same symbols: [`0:maj, 9:min, 2:min, 7:7`]. Second, it changes the distribution of the symbols in our dataset (Fig.

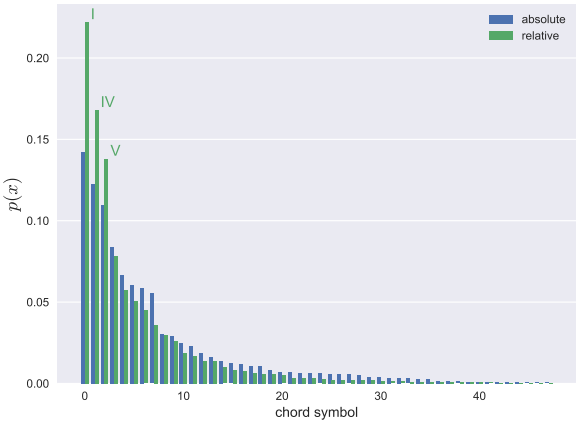


Fig. 5. The distribution of chord symbols in our dataset changes when we represent chords absolutely (e.g. `C:maj`) or relatively to the tonic of each song (e.g. `0:maj`). The figure shows the two distributions, where the 48 chord symbols are independently sorted from the most probable to the less probable. Using the relative notation, a fraction of the probability mass concentrate on just three symbols, which unsurprisingly are the three most common chords in tonal harmony theory: `0:maj` (I), `5:maj` (IV) and `7:maj` (V).

TABLE III
ESCAPE METHODS

method	escape if	ξ	$\hat{c}(x)$	$\sum_{x\in\mathcal{X}}\hat{c}(x)$
A [28]	$c(x)=0$	1	$c(x)$	$\mathcal{N}$
B [28]	$c(x)\leq 1$	$\mathcal{N}_{2+}$	$c(x)-1$	$\mathcal{N}-\mathcal{N}_1-\mathcal{N}_{2+}$
C [33]	$c(x)=0$	$\mathcal{N}_{1+}$	$c(x)$	$\mathcal{N}$
D [34]	$c(x)=0$	$.5\mathcal{N}_{1+}$	$c(x)-.5$	$\mathcal{N}-.5\mathcal{N}_{1+}$
AX [35]	$c(x)=0$	$\mathcal{N}_1+1$	$c(x)$	$\mathcal{N}$

5), reducing its entropy from 4.30 to 3.77 bits/symbol. We use the interval in semitones rather than scale degree so that we can represent chords built on non-scale notes more clearly.

IV. LANGUAGE MODELS TRAINING

In this section we describe the process of training the three language models (PPM, HMM and RNN). The evaluation metric that is commonly used for language models is the cross-entropy:

$$H_p(T)=\frac{1}{|T|}\sum_{x\in T}-\log_2 p(x), \tag{16}$$

where T represents the test data and H_p is measured in bits/symbol. Therefore, the lower the cross-entropy, the better we can expect the language model to predict the next symbol. Furthermore, we evaluate the variability of our cross-entropy estimates using 5-fold cross-validation.

A. PPM

With reference to Eq. 4, different heuristics have been proposed to compute $\alpha(x_i)$ and γ, where the dependence on the context x_{i-n}^{i-1} has been omitted to simplify the notation. $\alpha(x_i)$ and γ can be represented in terms of the smoothed counts $\hat{c}(x)$ and the virtual count ξ attributed to chord symbols that are novel in the current context:

$$\alpha(x_i)=\frac{\hat{c}(x_i)}{\xi+\sum_{x\in\mathcal{X}}\hat{c}(x)} \tag{17}$$

$$\gamma=\frac{\xi}{\xi+\sum_{x\in\mathcal{X}}\hat{c}(x)} \tag{18}$$

A summary of the different methods to obtain $\hat{c}(x)$ and ξ is given in Table III, where the following definitions have been used:

- $c(x)$ is the count of occurrences of the chord x
- $\mathcal{N}$ is the total number of occurrences
- $\mathcal{N}_k$ is the number of chord symbols occurring exactly k times
- $\mathcal{N}_{k+}$ is the number of chord symbols appearing at least k times

in the training set. All the afore-mentioned variables depend on the current context and are related by the following equations:

$$\mathcal{N}=\sum_{k=1}^{\infty}k\mathcal{N}_k=\sum_{x\in\mathcal{X}}c(x) \tag{19}$$

$$\mathcal{N}_{j+}=\sum_{k=j}^{\infty}\mathcal{N}_k. \tag{20}$$

The computation of $p(x_i|x_{i-n}^{i-1})$ is carried out by decreasing the order of the model until the recursion in Eq. (4) terminates. If a chord symbol is not found with a 0-order model (i.e. with 1-grams), a uniform distribution on the alphabet is used.

Two common enhancement techniques that we tested are exclusion and interpolation. Exclusion removes the symbols found at higher order models (i.e. with longer contexts) from the computation of $\sum\hat{c}(x)$ in equations (17) and (18). Interpolated smoothing is an alternative approach to backoff smoothing Eq. (4) and is described by the following equation:

$$p(x_i|x_{(i-n)+1}^{i-1})=\max\left(\alpha(x_i|x_{(i-n)+1}^{i-1}),0\right)+\gamma(x_{(i-n)+1}^{i-1})p(x_i|x_{(i-n)+2}^{i-1}), \tag{21}$$

where it can be noticed that interpolated smoothing uses lower order models even for n-grams with non-zero count.

The cross-entropy results achieved by the methods in Table III, with backoff and interpolated smoothing, are shown in Figure 6. All methods used the exclusion technique because it consistently led to better results. The variation in H_p observed with 5-fold cross validation is two orders of magnitude smaller than the data, meaning that our estimates are stable and independent of the particular choice of the training and test sets. Backoff smoothing seem to plateau or even overfit for high order models, due to the increasing sparsity of the occurrences for longer contexts. In particular, if a symbol is found in a high order model with a sparse distribution, the probability estimation might be a bad approximation and would benefit from exploiting the lower order models. Such a claim is fostered by the fact that there is no such trend for interpolated models. In the latter category, only method A does not benefit from increasing the order, probably due to its small escape probability that leads to small differences between backoff and interpolated smoothing.

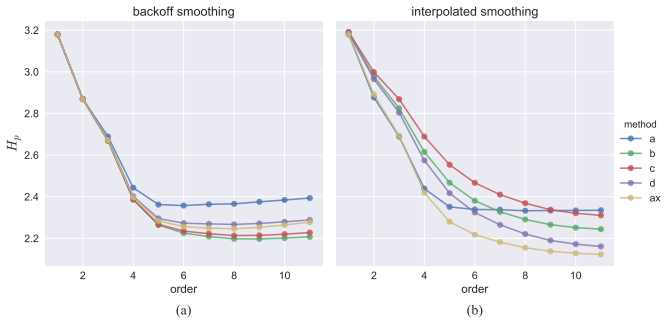


Fig. 6. Five smoothing methods for the PPM language model are compared using (a) backoff and (b) interpolated smoothing. Order 0 (the 1-gram model) is not shown in the figure and achieves the same value of cross-entropy (3.77 bits/symbol) independently of the smoothing method. Error bars computed by 5-fold cross validation are not visible, being two orders of magnitude smaller than the data.

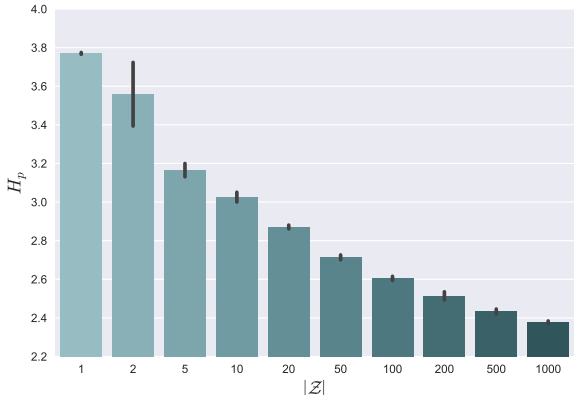


Fig. 7. The cross-entropy metric H_p obtained by the HMM with different hidden node alphabet sizes $|\mathcal{Z}|$. As expected, with increasing $|\mathcal{Z}|$ the model becomes more flexible and computes more accurate predictions. Error bars are computed using 5-fold cross validation. No overfitting behavior is observed for the sizes that we tested ($|\mathcal{Z}| \leq 1000$).

B. HMM

We trained a series of HMMs varying the size of the alphabet $\mathcal{Z}$ of the hidden nodes z logarithmically from 1 to 1000, using 100 iterations of the EM algorithm with random parameter initialization. The distributions used by the model contain a total number of parameters equal to:

$$(|\mathcal{Z}| - 1) + |\mathcal{Z}|(|\mathcal{X}| - 1) + |\mathcal{Z}|(|\mathcal{Z}| - 1). \qquad (22)$$

As expected, results (Figure 7) show that cross-entropy H_p decreases with increasing numbers of parameters. When $|\mathcal{Z}| = 1$ the model just learns one distribution $p(x_i|z_i = 1)$, effectively reducing to the 1-gram model. For the model with $|\mathcal{Z}| = 2$, the cross validation leads to much larger variance than the models with $|\mathcal{Z}| > 2$. Arguably the variance is caused by the different local maxima found by EM, due to the random initialization of the parameters. In order to get a better understanding of the model and interpret the results, we briefly analyze the meaning of the latent variable z.

If we focus on the Markov model on z and consider only transition probabilities $p(z_i|z_{i-1})$ larger than a certain threshold, it is possible to create a graph of hidden node

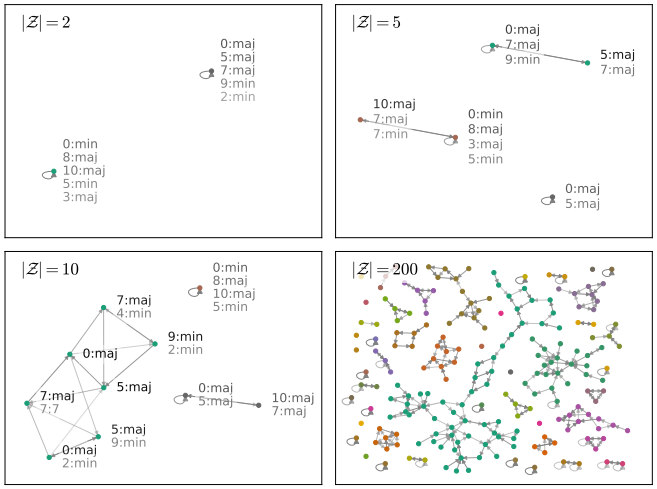


Fig. 8. The figure represents four HMM models with varying hidden node alphabet sizes $|\mathcal{Z}|$. The representation is an extension of a state diagram where points are hidden states and the arrows represent transitions, having opacity proportional to the transition probability $p(z_i|z_{i-1})$. Labels of the most probable chord symbols emitted are shown near each state, with opacity proportional to the emission probability $p(x_i|z_i)$. The two connected components of the state diagram learned by the simplest model with $|\mathcal{Z}| = 2$ (top left) exactly represent major (ionian) and natural minor (aeolian) modes. The two modes are also learned by the model with $|\mathcal{Z}| = 5$ (top right), this time with connected components including two states each. The subdivision of chord emissions between the two states of the minor mode, is also tonally relevant, because it highlights the tonal function of the dominant. For $|\mathcal{Z}| = 10$ (bottom left) a more structured component represents the major mode, and a single state component the minor mode. Interestingly, a third component includes a chromatic chord, `10:maj`, among other chords from the major mode. The bottom right figure simply shows the approximate state diagram for a much richer model $|\mathcal{Z}| = 200$.

connections, similar to a state diagram. Adding the most probable chord symbols emitted by each hidden node (i.e. thresholding the emission probability $p(x_i|z_i)$), it becomes clear how the different connected components of the graph identify different modes or chord progressions (Figure 8). The models with fewer hidden nodes can capture only few of those modes or progressions, and the choice is greatly influenced by the random initialization of the model parameters.

Discussion: In order to experiment with models having a high number of parameters ($|\mathcal{Z}| > 100$), given the size of our dataset, we had to develop a custom optimized implementation of the EM algorithm. Our implementation [2] the dataset is stored in memory-efficient data structures, each iteration of the Expectation step is split among multiple threads, and all computations use Single Instruction, Multiple Data (SIMD) parallelism whenever possible. Although this granted a speedup of three orders of magnitude and memory savings of two orders of magnitude with respect to the other implementations that we experimented with, training the largest model ($|\mathcal{Z}| = 1000$) for 100 EM iterations with our implementation took approximately five days.

C. RNN

We experimented with the training of several RNNs, varying the number of hidden layers and the number of units that

[2]https://github.com/brunodigiorgi/fdhmm

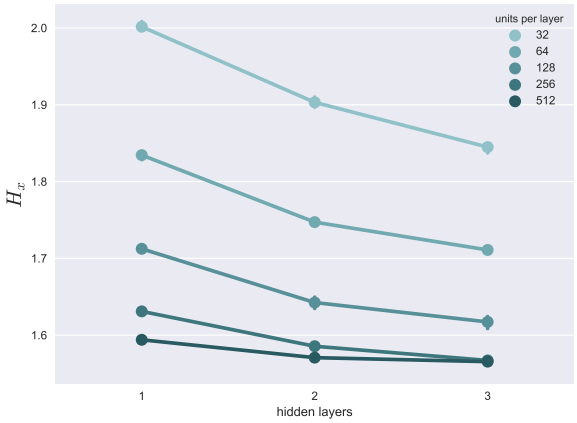


Fig. 9. The cross-entropy achieved by different RNN configurations, varying the number of hidden layers and the number of units for each layer.

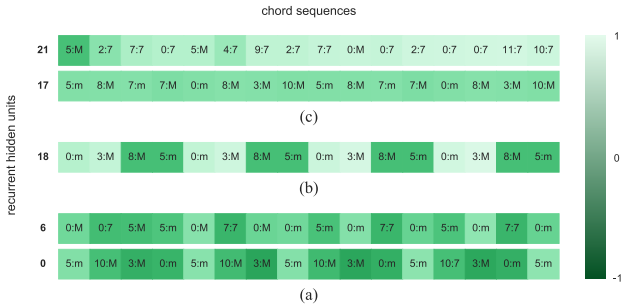


Fig. 10. Some explanatory patterns of activations of different hidden units in the RNN. Each row represents a particular hidden unit (its number is shown with a bold label at the left). The input chord sequences are represented as strips of chord labels. Chord symbol format is `r:m` with `r` being the root, relative to the tonic, and `t` being the type of the chord (`:M` for `:maj` and `:m` for `:min` for compactness). The hidden state of the unit is a continuous value in $[-1, 1]$, and is represented by the background color of each chord label. Some units (a) are suddenly activated by specific chords, e.g. unit 0 is excited by `5:m`, unit 6 by `0:M` and `0:m`. Other units (b) seem to recognize transitions or short sequences, e.g. unit 18 is excited by `0:m` followed by `3:M`. Still others (c) are activated more gradually and seem to recognize longer contexts or chord types, e.g. unit 17 recognizes the aeolian mode, unit 21 is activated by the `:7` chord type.

they contain. Before analyzing the performance, we list some additional details of the models.

- The *1-of-K* representation is required as the input of this model, therefore each chord symbol x is mapped to a binary vector of $\mathcal{X} = 48$ entries, with only the corresponding entry being equal to one.
- We adopted an extension of the basic RNN called Long Short Term Memory (LSTM) [50], in which units themselves contain recurrently connected subnets, leading to a longer effective context.
- After the recurrent layers, a fully-connected layer with a softmax non-linearity is used to compute the output probabilities.
- The gradients are computed using full BPTT, which back propagates error gradients along the full network unrolled in time.
- We used the RMSProp optimizer, a variation of Gradient Descent, where the learning rate is rescaled for each parameter, exploiting the history of the gradients for that parameter.
- The early stopping criterion was set to automatically terminate the training loop as soon as the validation error does not decrease for 5 consecutive epochs, with a tolerance equal to 1% the mean of the validation error of the last 5 epochs.

As expected, models with more units per layer are more flexible and achieve lower cross-entropy values (Fig 9). Adding more layers improves the performance, but not as much as doubling the number of units of a single hidden layer. In an RNN, each recurrent layer learns a dense latent representation of sequences of its input; the results suggest that while chord sequences are very well described by such a representation, there is less structure in higher level representations.

In Figure 10, we visualize what has been learnt by the hidden units of the network. Considering the simplest model, which includes only one hidden recurrent layer of 32 units, we chose some input chord sequences for which some units responded in a peculiar way. We noticed that different units learn patterns of different lengths and respond with different speed. Some are suddenly activated by single chords, others are activated more gradually by longer patterns. Some units learn latent invariances, for example they are excited by chords that share the same root (e.g. `0:maj` and `0:min`) or the same type (e.g. `0:7` and `2:7`). A more in depth analysis is hard because the states are recurrent and interconnected, i.e. each hidden cell is fed by the inputs and by all hidden cells of the same hidden layer at the previous time step.

The space of hidden states is much larger than that of the HMM, because it is exponential in the number of hidden units. Moreover, units are continuous-valued, which makes the space even larger. For this reason the RNN exhibits a much better predictive power than the HMM. A fair comparison with PPM would require the RNN and HMM to be constrained to have finite memory, which is what we do in the next section in order to create a compound model.

D. Compound Model

The three models described in the previous sections will be used as generative models to create short chord sequences to be used in two perceptual experiments. As we detail in Sect. V, the chord sequences used in the perceptual experiments contain 4 chords.

In order to generate such chord progressions, we created a compound model as a weighted average of the three models:

$$p(x_i|x_0^{i-1}) = \sum_{m \in \mathcal{M}} \pi_m p_m(x_i|x_0^{i-1}), \tag{23}$$

where $\mathcal{M}$ is the set of models, including a representative model for each model class (PPM, HMM and RNN), and the weights π_m sum to 1. Each model class is represented by the best performing model in the previous evaluations:

- For PPM, we chose method C with backoff smoothing (which performed best at order 3).
- For HMM, we chose $|\mathcal{Z}| = 1000$

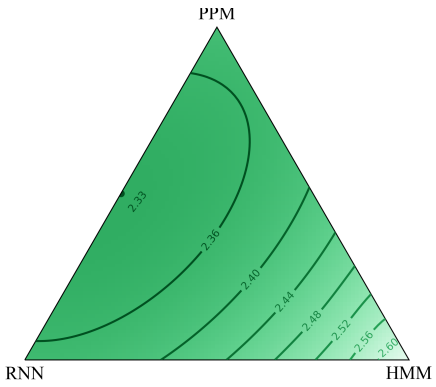


Fig. 11. Cross entropy of the compound model consisting of the weighted average of three models based on PPM, HMM and RNN. The cross entropy values are evaluated on the 2-simplex formed by the weights π_m, with $\sum_m \pi_m = 1$. These values are reported in the contour labels and are expressed in bits/sample.

- For RNN, we chose the network with 3 layers of 128 units.

We maximize the prediction accuracy, measured with cross-entropy, with a grid search over the weights π_m. Since we need to create 4-chord progressions, we evaluate cross-entropy of the compound model using a fixed-memory context of 3 chords. We use data held out from previous training and validation sets to optimize the compound model. As shown in Figure 11, the best model is the average between the PPM and the RNN models, and obtains a cross-entropy value of 2.33 bits/sample. In the separate evaluations RNN obtained lower entropy than PPM; however, the comparison was biased since RNN (as also HMM) used unbounded context (i.e. all the chords since the start of the song) when predicting the next symbol. The use of a fixed and short context explains the different results obtained here.

V. Experiments

In this section we present the design and the results of the listening test, which includes two perceptual experiments. The goal of both the experiments is to evaluate how the probability of a chord sequence, given by our language model using Eq. (1), is related to the subjective evaluation of complexity.

The first experiment is a pairwise comparison between two chord sequences, where the subject is asked to decide which is more complex and which he/she liked the most. In this experiment, we decided to only ask for binary evaluations in order to limit the cognitive load. Chord sequences used in this experiment belong to K separate groups, where sequences in the same group share similar values of probability. The second experiment is a graded evaluation of complexity and preference on a single chord sequence.

Since the result of the test depends on the meaning that is subjectively assigned to the word "complexity", we never mention words such as "surprising" or "unexpected" in the instructions in order not to bias the subjects toward a specific meaning or interpretation. 23 test cases are presented in each experiment; this number has been chosen to make the whole test last approximately 30 minutes, estimated as a reasonable time span for maintaining focus.

The generation of the chord progressions is explained in Sect. V-A and the creation of the audio files is presented in Sect. V-B. In Sect. V-C we describe the procedure for profiling the participants. The results of the tests are presented in Sect. V-D and V-E.

A. Sampling the chord sequences

The goal of our experiments is to analyze the relation between chord sequence probability (Eq. (1)) and perceived complexity. Therefore we need to generate sequences having different probabilities, to be used in the listening test.

The usual way to use a language model generatively is to sample one chord at a time: x_0 from $p(x)$, then x_1 from $p(x|x_0)$, and x_i from $p(x|x_0^{i-1})$. However, using this method we cannot control the final probability of the sequence. We used two ways to sample sequences with a desired value of probability: *range sampling* and *uniform sampling*, detailed below.

1) Range sampling: With range sampling, every chord x_i of the sequence is sampled from a subset $\hat{\mathcal{X}}_k(x_0^{i-1}) \subset \mathcal{X}$ containing chords with constrained probability:

$$\hat{\mathcal{X}}_k(x_0^{i-1}) = \left\{x | \alpha_k \leq p(x|x_0^{i-1}) < \beta_k\right\}, \tag{24}$$

where $[\alpha_k, \beta_k)$, is one of K non-overlapping intervals of probability values, with $k = 0, \ldots, K-1$.

We choose the intervals relatively to the actual values of $p(x_i|x_0^{i-1})$. In particular, we set intervals of percentiles as follows:

$$f(k) = 1 - \left(\frac{k}{K}\right)^{\eta} \tag{25}$$

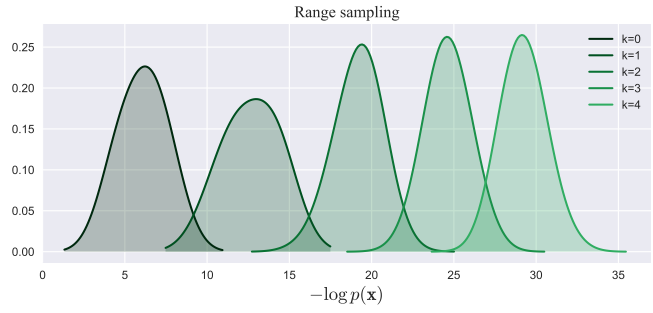
$$\tilde{\alpha}_k = f(k+1) \tag{26}$$

$$\tilde{\beta}_k = f(k), \tag{27}$$

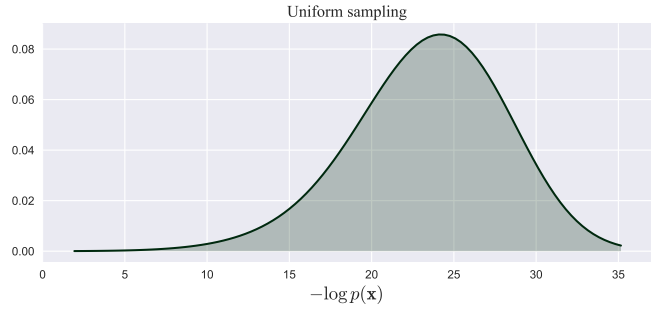
where the parameter η controls the progression of the interval sizes, e.g. fixed for $\eta = 1$ and linearly increasing with k for $\eta = 2$. We assign the $100\tilde{\alpha}_k$ percentile and the $100\tilde{\beta}_k$ percentile of $p(x|x_0^{i-1})$ to α_k and β_k respectively. Using relative boundaries ensures that the subset of candidate next chords $\hat{\mathcal{X}}_k(x_0^{i-1})$ is non empty for every contexts and intervals. In particular, $\hat{\mathcal{X}}_k(x_0^{i-1})$ always contains $|\mathcal{X}|\left(\tilde{\beta}_k - \tilde{\alpha}_k\right)$ chords.

Now, for any two sequences $\mathbf{x}_j$ and $\mathbf{x}_l$ sampled using $k = j$ and $k = l$, it is very likely that $p(\mathbf{x}_j) > p(\mathbf{x}_l)$ if $j < l$, although not true in general. For the first experiment we used $K = 5$ groups of chord sequences and set $\eta = 1.8$ to distribute them evenly in the log probability domain, as shown in Figure 12a.

2) Uniform sampling: Uniform sampling simply consists of sampling a large number of sequences by choosing each chord of the sequence from the uniform distribution over $\mathcal{X}$. The procedure results in the distribution over sequence probabilities shown in Figure 12b. For our second experiment we use uniform distribution just as a starting point to generate a large number of chord sequences, then we chose 40 sequences with evenly spaced log probability values.



(a)



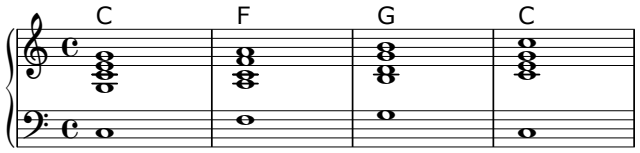
(b)

Fig. 12. Estimated density of the probability of the chord sequences generated with (a) range sampling and (b) uniform sampling. (a) shows how the parameter k controls the sampled sequence probability in range sampling. Range sampling produces good separation between different groups, and has therefore been chosen for generating the chord sequences for the first experiment. The densities in the figures are estimated by generating 10000 sequences for each configuration and using a Gaussian kernel with unit size.

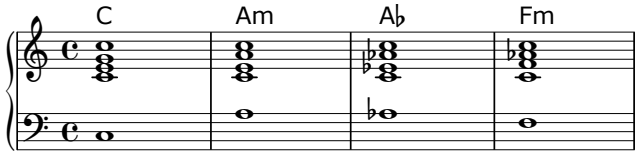
B. Creation of the audio excerpts

The chord sequences need to be rendered in audio format in order to perform the listening test. This conversion from symbolic to audio domain raises questions about tonic, tempo, instrumentation and chord voicing (i.e. the octave and the doubling of the pitch classes contained in the chord) that we addressed with the goal of narrowing the focus of the subject on the only relevant aspect: the chord sequence.

We chose the note `C` as the tonic and `C:maj` as the first chord of each sequence. Fixing the tonic and the first chord establish the same reference point for each chord progression listened during the test session, therefore eliminating a possible hidden source of noise in the survey. We decided to start only with a major chord in order to reduce the parameter space and the test duration. Regarding the tempo, we chose to change chord every 1.5 seconds, corresponding to 4 beats at 160 beats per minute (BPM) or 2 beats at 80 BPM. For the instrumentation and arrangement, we limited to the grand piano playing sustained chords. In particular, to avoid introducing any bias with human performances, we rendered the chord sequences from automatically created MIDI files, using sounds from a commercial grand piano sample library. Regarding the voicings, we chose to only use chords in root position, i.e. where the lowest note plays the root of the chord. We used four upper voices, enough to contain all the notes of the 7 chord, the richest one in our reduced alphabet. Another important issue regarding the voicings lies in the choice of the upper voices and how they evolve during chord changes.



(a)



(b)

Fig. 13. Sheet music representation of two examples of progressions used in the test. Our simple voice leading model promotes continuity within the voices, apart from the lowest note, fixed to the root of the chord. Progression (a) has higher probability than progression (b): $\log p(\mathbf{x}_{(a)}) = -4.7$, $\log p(\mathbf{x}_{(b)}) = -15.7$.

The musical theory of counterpoint deals with this problem, and it is known that moving all voices by parallel motion (i.e. in the same direction and of the same interval) sounds boring and dull. Although we did not implement a comprehensive counterpoint model, we focused on voice leading and tried to address specific aspects, such as minimizing jumps in the melodic lines of the voices.

Our simple voice leading model takes a sequence of chords and outputs a sequence of voicings (Fig. 13) ready to be synthesized to audio. The process is performed in three steps:

1) each chord generates a list of possible voicings and associates a score to each one.
2) transition probabilities among every pair of successive sets of voicings are computed using voice leading rules (i.e. enforcing continuity of the voices).
3) the optimal sequence of voicings is found by the Viterbi algorithm, which jointly maximizes the score of each voicing and the transition probabilities.

C. Subject profiling

We chose to profile the musical attitudes of participants using the self-report questionnaire of the Goldsmiths Musical Sophistication Index (Gold MSI v1.0) [51]. The test includes 38 questions with seven-point scale answers, ranging from complete agreement to complete disagreement, that can be combined to form 5 sub-factors and one general factor called General Musical Sophistication (GMS). The average GMS factor for the 56 subjects who participated in the experiment is 75. This agrees with the study in [52], where groups of subjects with different socio-economic status have averages around 80.

We also added a question specifically addressing harmony skills, with a five-scale answer. The correlation between GMS and the answers to this question is positive, strong and statistically significant ($r = .71$, $p < .001$), therefore we will just refer to the GMS value hereafter.

D. Experiment 1

In the first experiment, the subject is required to listen to two chord sequences, then select the most complex and the one he/she found more likable. The subject evaluates 23 such test cases, of which the initial three are used to familiarize with the user interface and are not recorded. During the setup phase, the sequences are generated with range sampling with $K = 5$ groups and $n = 16$ chord sequences per group. This set is then divided into two round-robin subsets (with $n_{\mathrm{r}} = 8$ chord sequences per group). Round-robin means that the system will select one of the two subsets on each session, in alternating fashion. Then, each time a new subject begins the experiment, the system:

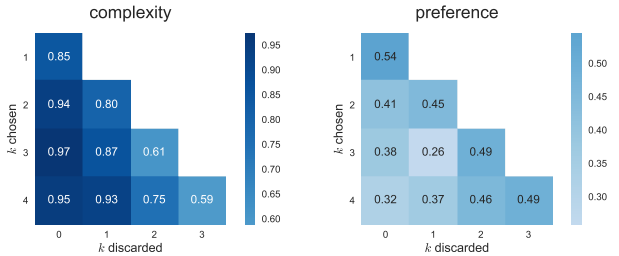
1) selects one of the two round-robin sets
2) shuffles the sequences within each of the K groups, in order to randomize the pairs and their presentation order
3) creates $2\binom{K}{2} = 20$ pairwise comparisons
4) inserts three dummy test cases at the beginning of the test, containing sequences from the unused round-robin set.

These steps result in the comparison between every pair of groups being performed twice, but no chord sequence being heard in more than one test case.

The results for complexity and preference are shown in Figure 14 as two lower triangular matrices. For the complexity matrix $\mathbf{C}$, the entry $C_{i,j}$ contains the number of times a chord progression from group i was selected as more complex when compared to a chord sequence from group j, divided by the number of comparisons between the two groups. Only the lower triangular part is shown, because $C_{i,j} + C_{j,i} = 1$. The same holds for the preference matrix $\mathbf{P}$.

As expected, if $i > j$, then $C_{i,j} > .5$, meaning that the majority of the subjects evaluated as more complex, a sequence having lower probability. Another interesting result is that $C_{i,j} > C_{i,l}$ if $j < l$ and $C_{i,j} > C_{l,j}$ if $l < i$, which means that moving along rows or columns, the agreement in choosing the more complex sequence increases by increasing the distance between the two groups; with only one exception for $C_{3,0} = .97$ and $C_{4,0} = .95$. Furthermore, results show that with fixed distance between the groups (i.e. moving along the diagonals) $C_{i,i-l} > C_{j,j-l}$ if $i > j$, meaning that there is more agreement for less complex groups. This last result might derive from the sequences of some groups (e.g. groups 3 and 4) having such low probability that they are difficult to differentiate.

The preference matrix shows the opposite behavior: $P_{i,j} < .5$ (with the exception of $P_{1,0}$), meaning that the majority of subjects preferred simpler chord sequences. However, the exception is meaningful since it may indicate that chord sequences in group 1 are more complex than group 0 ($C_{1,0} > .5$) but also more interesting ($P_{1,0} > .5$), due to the usage of less predictable chords. Analogously to the results for complexity, moving along rows or columns, the agreement in choosing the preferred sequence increases by increasing the distance between the two groups: $P_{i,j} < P_{i,l}$ if $j < l$ and $P_{i,j} < P_{l,j}$ if $l < i$ (only exception: $P_{3,1} = .26$). Moreover, the preference results show balance in the diagonal $i-j = 1$ where $P_{i,j} \approx .5$,



(a) (b)

Fig. 14. The results of the first experiment are summarized in this figure by the two lower triangular matrices. For each test case the subjects were asked to chose (a) the more complex; and (b) the preferred, between two chord sequences belonging to different groups. Chord sequences that belong to the same group k have similar probability values, where small k means higher probability. The value of the (i, j) entry correspond to the number of times the group i has been chosen when compared with the group j, normalized by the total number of such comparisons.

meaning that there is no agreement about preference for chord sequences belonging to adjacent groups.

We also recorded the number of times the subject listened to the two chord sequences in a test case before advancing to the next test case. Only one listening is required, but the system allows the subject to play the audio again if needed. For most of the cases one listening was enough for people to provide their ratings. However, the comparisons between the group pairs (2, 3), (2, 4) and (3, 4) required on average one extra listening. This confirms that those groups are difficult to discern, as it is also evident in $\mathbf{C}$ and $\mathbf{P}$: in fact the agreement $|\mathbf{C} - .5|$ and $|\mathbf{P} - .5|$ is minimized for those same pairs.

E. Experiment 2

In the second experiment, the subjects are asked to listen to a chord sequence and then rate its complexity and how much they liked it on a continuous scale. As in the first experiment, the subjects evaluate 23 such test cases, of which the initial three are used to familiarize with the user interface and are not recorded. During the setup phase, the sequences are generated with uniform sampling, by generating 100000 chord sequences and then picking 40 sequences with linearly spaced log probability. As in the first experiment, the 40 sequences are subdivided into two round-robin subsets, with interleaved splitting to maintain the linear spacing and the range of the original set. Each time a new subject starts the experiment, the system:

1) selects one of the two round-robin sets
2) shuffles the sequences in the set
3) inserts three dummy test cases at the beginning of the test, containing sequences from the unused round-robin set. Specifically, to set the scale for the ratings, we pick the three sequences with minimum, maximum and mean log probability values.

Results of the experiment are shown in Figure 15 and confirm the existence of a relationship between the probability of the sequence given our model and the subjects' ratings of complexity. For each of the 40 chord sequences we collected

TABLE IV
PEARSON R CORRELATION COEFFICIENT VS MUSICAL SOPHISTICATION

	GMS: high	GMS: low	GMS: all
complexity	0.65	0.54	0.60
preference	0.20	0.25	0.21

28 ratings of complexity and preference. The negative log probability exhibits a strong positive correlation with the complexity ratings ($r = .6$, $p < .001$) and a weak negative correlation with preference ratings ($r = -.21$, $p < .001$).

In order to better analyze those relations, we tried different polynomial regression models, with the negative log probability as input and the average ratings as targets. These regressors have been evaluated using the coefficient of determination R^2 on 200 iterations of shuffle and split cross-validation, each time using 20% of the examples in the test set. As shown in Figure 16, the relation between complexity ratings and negative log probability is best explained by a 2nd-order polynomial ($R^2 = 0.81$), while the relation between preference and negative log probability is linear ($R^2 = 0.14$).

Such regression curves are plotted in Figure 15, where we separated the ratings given by participants with lower and higher than average musical sophistication, given by the GMS factor. For both categories the complexity ratings seem to saturate for chord sequences with low probability. This confirms the intuition that for such chord sequences it is difficult to discern the harmonic complexity, as also shown by the results of the first experiment. The saturation effect is less evident for subjects with high musical sophistication, whose complexity ratings tend to maintain a more linear behavior. The relation with preference highlights the tendency to prefer the chord sequences with high probability. However, this relation is weaker and independent of the musical sophistication factor. The Pearson correlation coefficients between the polynomial regression models and the ratings of subjects in a specific GMS category is reported in Table IV, where all the values are statistically significant ($p < .001$).

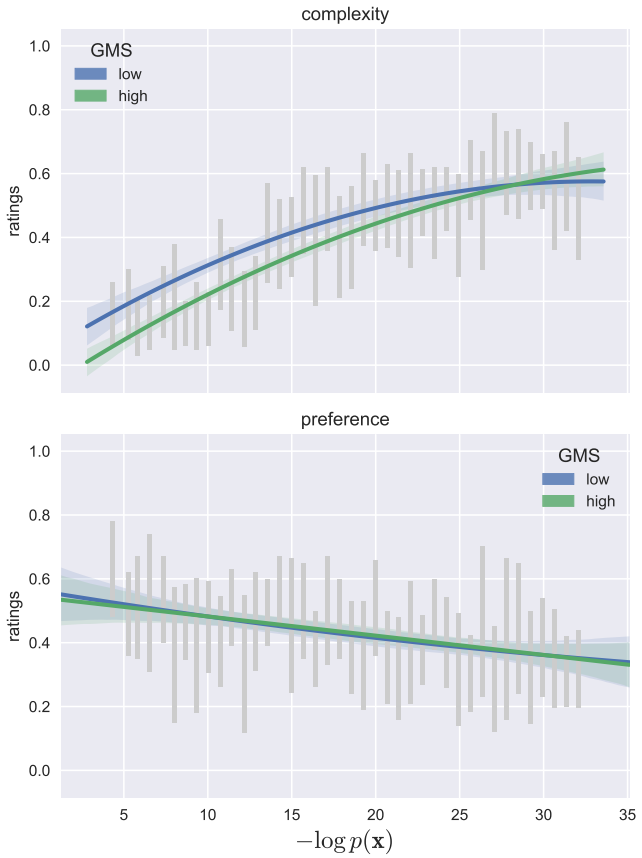


Fig. 15. The figure shows the ratings of complexity and preference obtained in the second experiment, versus the negative log probability of the rated chord sequences. The ratings are shown in the original scale of [0, 1] used in the test. Regression models are overlaid, with different colors for for the ratings of subjects with low and high GMS factor (i.e. lower and higher than the average). Vertical gray bars are the interquartile ranges of the ratings for each chord sequence. The confidence intervals of the regression lines are obtained through 1000 bootstrap iterations.

VI. CONCLUSION AND FUTURE WORK

In this paper, we design the architecture of a language model of tonal chord sequences and evaluate its capability to estimate the perceived harmonic complexity. We build and train three different language models: prediction by partial matching, a hidden Markov model and a recurrent neural network, analyzing several variations of them. For training we use the largest available annotated dataset, to the best of the authors knowledge, containing the chords of half a million songs. We combine the best configurations of the three models into a more powerful model, which is then used to generate a set of chord sequences and evaluate their probability. We devise a listening test, comprising two experiments, that allows us to collect many perceptual ratings related to complexity and preference of the generated sequences. By analyzing the results of the test, we show the existence of a strong positive and statistically significant correlation between the log probability of the chord sequences, computed by our language model, and the complexity ratings.

However, some issues remain open for future investigations.

First, it should be relatively straightforward to complete the exploration of the parameter space, including in a similar listening test chord sequences starting from a minor chord and generated with the different models without averaging predictions. A fifth tonic chord could be forced at the end of the progressions in order to avoid drifting away from tonality, especially for the less probable sequences.

Second, since the results show that the sequences with very low probability are difficult to distinguish in terms of complexity, it should be worthwhile to focus on the most probable sequences, in order to better analyze the relation with complexity and to see if a stronger relationship with preference emerges. The correlation between complexity and preference is something that we find worth investigating more; in particular, we could not verify the effect of over-familiarity on the less complex chord progression, which would result in the famous inverted U-shaped function. Possible reasons are the bias generated by querying at the same time complexity and preference ratings, and the brevity of progressions: too short to exhibit meaningful internal structure. Besides, the more complex progressions often obfuscate the tonality,

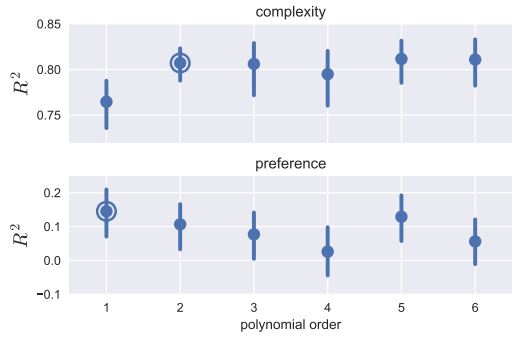


Fig. 16. The values of the coefficient of determination R^2 obtained by different polynomial regression model used to fit the log probability of the sequences to the average ratings of the second experiment. These values are obtained through 200 iterations of shuffle and split cross-validation, splitting the examples into 80% train set and 20% test set. The models that we chose to represent the relations of log probability with complexity and preference are circled.

producing an interesting sound for someone used to complex music. Further experiments that address specifically these issues would provide an important complement to our results.

Third, it would be interesting to leverage the existing annotated datasets of chords used for automatic chord recognition research, which cumulatively count approximately two thousand annotated songs. The quality of those annotations is superior, on average, to those used in this work, and this could be exploited by weighting the contribution of different datasets in the training of the language model.

Finally, the dataset used here contains genre tags, opening the possibility of analyzing chord sequence complexity across different genres.

ACKNOWLEDGMENT

The authors would like to thank Dr Marcus Pierce for his assistance with the design of the experiments.

REFERENCES

[1] P. C. Vitz, "Preferences for rates of information presented by sequences of tones," *Journal of Experimental Psychology*, vol. 68, no. 2, p. 176, 1964.

[2] W. F. Thompson, *Music, thought, and feeling: Understanding the psychology of music*. Oxford; New York: Oxford University Press, 2008.

[3] R. G. Heyduk, "Rated preference for musical compositions as it relates to complexity and exposure frequency," *Perception & Psychophysics*, vol. 17, no. 1, pp. 84–90, 1975.

[4] D. E. Berlyne, *Aesthetics and psychobiology*. Appleton-Century-Crofts, 1971.

[5] M. Mauch and M. Levy, "Structural change on multiple time scales as a correlate of musical complexity," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2011, pp. 489–494.

[6] D. Temperley, *The cognition of basic musical structures*. MIT press, 2004.

[7] S. Kostka, D. Payne, and B. Almen, *Tonal Harmony*, 7th ed. McGraw-Hill Education, 2012.

[8] E. Bigand and M. Pineau, "Global context effects on musical expectancy," *Perception & Psychophysics*, vol. 59, no. 7, pp. 1098–1107, 1997.

[9] S. Koelsch, T. Gunter, A. D. Friederici, and E. Schröger, "Brain indices of music processing: "nonmusicians" are musical," *Journal of Cognitive Neuroscience*, vol. 12, no. 3, pp. 520–541, 2000.

[10] C. Weiss and M. Müller, "Quantifying and visualizing tonal complexity," in *Proceedings of the Conference on Interdisciplinary Musicology (CIM). Berlin*, 2014.

[11] C. Weiss and M. Müller, "Tonal complexity features for style classification of classical music," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 688–692.

[12] K. K. Jensen and D. Hebert, "Predictability of harmonic complexity across 75 years of popular music hits," in *Proceedings of the International Symposium on Computer Music Multidisciplinary Research (CMMR)*. Springer, 2015.

[13] D. B. Huron, *Sweet anticipation: Music and the psychology of expectation*. MIT press, 2006.

[14] E. Bigand, B. Poulin, B. Tillmann, F. Madurell, and D. A. D'Adamo, "Sensory versus cognitive components in harmonic priming," *Journal of Experimental Psychology: Human perception and performance*, vol. 29, no. 1, p. 159, 2003.

[15] S. Streich, "Music complexity: a multi-faceted description of audio content," Ph.D. dissertation, Universitat Pompeu Fabra, 2006.

[16] A. J. Milne, "A computational model of the cognition of tonality," Ph.D. dissertation, The Open University, 2013.

[17] R. Parncutt, *Harmony: A psychoacoustical approach*. Springer Science & Business Media, 1989.

[18] L. Corentin and F. Bimbot, "Description of chord progressions by minimal transport graphs using the system & contrast model," in *Proceedings of the International Computer Music Conference (ICMC)*, 2016, pp. 345–350.

[19] L. Maršík, J. Pokornỳ, and M. Ilcík, "Towards a harmonic complexity of musical pieces," in *Procedings of the Annual International Workshop on Databases, Texts, Specifications, and Objects (DATESO)*, 2014, pp. 1–12.

[20] L. Maršík, J. Pokornỳ, and M. Ilcík, "Improving music classification using harmonic complexity," in *Proceedings of the Conference Information Technology - Applications and Theory (ITAT)*, 2014, pp. 13–17.

[21] M. Rohrmeier and T. Graepel, "Comparing feature-based models of harmony," in *Proceedings of the International Symposium on Computer Music Modelling and Retrieval*, 2012, pp. 357–370.

[22] R. P. Whorley, G. A. Wiggins, C. Rhodes, and M. T. Pearce, "Multiple viewpoint systems: Time complexity and the construction of domains for complex musical viewpoints in the harmonization problem," *Journal of New Music Research*, vol. 42, no. 3, pp. 237–266, 2013.

[23] F. Pachet, "Surprising harmonies," *International Journal of Computing Anticipatory Systems*, vol. 4, pp. 139–161, 1999.

[24] M. Mauch and S. Dixon, "Simultaneous estimation of chords and musical context from audio," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1280–1289, 2010.

[25] B. Catteau, J.-P. Martens, and M. Leman, "A probabilistic framework for audio-based tonal key and chord recognition," in *Advances in Data Analysis: Proceedings of the Annual Conference of the Gesellschaft für Klassifikation e.V., Freie Universität Berlin, March 8–10, 2006*. Springer, 2007, pp. 637–644.

[26] K. Lee and M. Slaney, "Acoustic chord transcription and key extraction from audio using key-dependent hmms trained on synthesized audio," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 291–301, 2008.

[27] B. Di Giorgi, M. Zanoni, A. Sarti, and S. Tubaro, "Automatic chord recognition based on the probabilistic modeling of diatonic modal harmony," in *Proceedings of the International Workshop on Multidimensional Systems (nDS)*, 2013, pp. 145–150.

[28] J. Cleary and I. Witten, "Data compression using adaptive coding and partial string matching," *IEEE Transactions on Communications*, vol. 32, no. 4, pp. 396–402, 1984.

[29] L. Rabiner and B. Juang, "An introduction to hidden Markov models," *IEEE ASSP magazine*, vol. 3, no. 1, pp. 4–16, 1986.

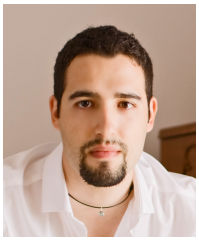
[30] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," *arXiv preprint arXiv:1506.00019*, 2015.

[31] I. H. Witten and T. C. Bell, "The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression," *IEEE Transactions on Information Theory*, vol. 37, no. 4, pp. 1085–1094, 1991.

[32] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1995, pp. 181–184.

[33] A. Moffat, "Implementing the PPM data compression scheme," *IEEE Transactions on Communications*, vol. 38, no. 11, pp. 1917–1921, 1990.

[34] P. G. Howard, "The design and analysis of efficient lossless data compression systems," Ph.D. dissertation, Brown University, Providence, RI, USA, 1993, uMI Order No. GAX94-06956.
[35] A. Moffat, R. M. Neal, and I. H. Witten, "Arithmetic coding revisited," *ACM Trans. Inf. Syst.*, vol. 16, no. 3, pp. 256–294, 1998.
[36] S. Bunton, "Semantically motivated improvements for PPM variants," *The Computer Journal*, vol. 40, no. 2 and 3, pp. 76–93, 1997.
[37] M. T. Pearce, "The construction and evaluation of statistical models of melodic structure in music perception and composition," Ph.D. dissertation, City University London, 2005.
[38] R. Scholz, E. Vincent, and F. Bimbot, "Robust modeling of musical chord sequences using probabilistic n-grams," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2009, pp. 53–56.
[39] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc., 2006.
[40] L. Shue, D. Anderson, and S. Dey, "Exponential stability of filters and smoothers for hidden Markov models," *IEEE Transactions on Signal Processing*, vol. 46, no. 8, pp. 2180–2194, 1998.
[41] H. Papadopoulos and G. Peeters, "Large-scale study of chord estimation algorithms based on chroma representation and HMM," in *International Workshop on Content-Based Multimedia Indexing*. IEEE, 2007, pp. 53–60.
[42] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.
[43] W. De Mulder, S. Bethard, and M.-F. Moens, "A survey on the application of recurrent neural networks to statistical language modeling," *Computer Speech & Language*, vol. 30, no. 1, pp. 61–98, 2015.
[44] A. Graves, G. Wayne, and I. Danihelka, "Neural Turing machines," *arXiv preprint arXiv:1410.5401*, 2014.
[45] S. Sigtia, N. Boulanger-Lewandowski, and S. Dixon, "Audio chord recognition with a hybrid recurrent neural network," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2015, pp. 127–133.
[46] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Audio chord recognition with recurrent neural networks," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2013, pp. 335–340.
[47] C. Harte, M. B. Sandler, S. A. Abdallah, and E. Gómez, "Symbolic representation of musical chords: A proposed syntax for text annotations," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2005, pp. 66–71.
[48] D. Temperley, "A Bayesian approach to key-finding," in *Music and Artificial Intelligence*. Springer, 2002, pp. 195–206.
[49] J. Pauwels and J.-P. Martens, "Integrating musicological knowledge into a probabilistic framework for chord and key extraction," in *Audio Engineering Society Convention 128*, 2010.
[50] S. Hochreiter and J. Schmidhuber, "Bridging long time lags by weight guessing and long short-term memory," *Spatiotemporal Models in Biological and Artificial Systems*, vol. 37, pp. 65–72, 1996.
[51] D. Müllensiefen, B. Gingras, L. Stewart, and J. Ji, "Goldsmiths musical sophistication index (gold-msi) v1," Goldsmiths University of London, Tech. Rep., 2013.
[52] D. Müllensiefen, B. Gingras, J. Musil, and L. Stewart, "The musicality of non-musicians: an index for assessing musical sophistication in the general population," *PloS one*, vol. 9, no. 2, 2014.



Bruno Di Giorgi Bruno Di Giorgi received his M.S. degree in sound engineering and his Ph.D. degree in music information retrieval from Politecnico di Milano, Milano, Italy, in 2013 and 2017, respectively. Between 2007 and 2013 he worked as live guitarist and soundtrack composer, and still teaches music composition. His research focuses on the automatic estimation of beats and chords, and the analysis of the perception of complexity related to such musical structures.

Simon Dixon Prof. Simon Dixon is Director of Graduate Studies and Deputy Director of the Centre for Digital Music at Queen Mary University of London. He has a PhD in Computer Science (Sydney) and LMusA diploma in Classical Guitar. His research is in music informatics, including high-level music signal analysis, computational modelling of musical knowledge, and the study of musical performance. Particular areas of focus include automatic music transcription, beat tracking, audio alignment and analysis of intonation and temperament. He was President (2014-15) of the International Society for Music Information Retrieval (ISMIR), is founding Editor of the Transactions of ISMIR, and member of the Editorial Board of the Journal of New Music Research (since 2011), and has published over 160 refereed papers in the area of music informatics.

Massimiliano Zanoni Massimiliano Zanoni is post-doctoral researcher in the Image and Sound Processing Group (ISPG) at the Department of Electronics, Information and Bioengineering (DEIB) of Politecnico di Milano. He received a Master degree in computer science from Alma Mater Studiorum University of Bologna and a Ph.D. degree in 2013 from Politecnico di Milano. His main research interests include music information retrieval, music emotion Recognition, ontology-based information management for modeling musical instruments knowledge and feature-based analysis of musical instruments.

Augusto Sarti Augusto Sarti (M04SM13) received his M.S. degree in electronic engineering and his Ph.D. degree in information engineering from the University of Padova, Padova, Italy, in 1988 and 1993, respectively. His graduate studies were in a joint graduate program with the University of California at Berkeley, Berkeley, CA, USA. In 1993 he joined the Politecnico di Milano, Milan, Italy, where he is currently a Full Professor. In 2013, he also joined the University of California at Davis, Davis, CA, USA, as an Adjunct Professor. He coordinates the activities of the Musical Acoustics Laboratory and the Sound and Music Computing Laboratory of the Politecnico di Milano. He has promoted/coordinated and/or contributed to numerous European projects. He has coauthored over 250 scientific publications on international journals and congresses and numerous patents in the multimedia signal processing area. His research interests include the area of multimedia signal processing, with particular focus on sound analysis, synthesis, and processing; space-time audio processing; geometrical acoustics; and music information extraction. He is an active member of the IEEE Technical Committee on Audio and Acoustics Signal Processing, and the chairman of the EURASIP Special Area Team (SAT) on Acoustic, Speech and Music Signal Processing (ASMSP). He is on the Editorial Board of the IEEE.