Automatic Phrase Continuation from Guitar and Bass-guitar Melodies [1]

Srikanth Cherla

Music Informatics Research Group
A309 (College Building)
City University London
10 Northampton Square
London EC1V 0HB, United Kingdom
abfb145@city.ac.uk

Hendrik Purwins

Neurotechnology Group
Berlin Institute of Technology
Facultät IV, MAR 4-3
Marchstr. 23
D-10587 Berlin, Germany
Sound and Music Computing Group
Department of Architecture, Design & Media Technology
Aalborg University Copenhagen
A.C. Meyers Vænge 15, DK-2450 Copenhagen SV, Denmark
hpurwins@gmail.com

Marco Marchini

Music Technology Group
Universitat Pompeu Fabra

[1] *Preprint* from: Srikanth Cherla, Hendrik Purwins, and Marco Marchini: *Automatic Phrase Continuation from Guitar and Bass Guitar Melodies*, Computer Music Journal, 37 (3), Pages 68-81, 2013.

Roc Boronat, 138
08018 Barcelona, Spain
marco.marchini@upf.edu

Abstract

A framework is proposed for generating interesting, and musically similar variations of a given monophonic melody. The focus is on rock/pop guitar and bass-guitar melodies with the aim of eventual extensions to other instruments and musical styles. It is demonstrated here how learning musical style from segmented audio data can be formulated as an unsupervised learning problem to generate a symbolic representation. A melody is first segmented into a sequence of notes using onset detection and pitch estimation. A set of hierarchical, coarse-to-fine symbolic representations of the melody is generated by clustering pitch values at multiple similarity thresholds. The *Variance Ratio Criterion* is then used to select the appropriate clustering levels in the hierarchy. Note onsets are aligned with beats, considering the estimated meter of the melody, to create a sequence of symbols that represent the rhythm in terms of onsets/rests and the metrical locations of their occurrence. A joint representation based on the cross-product of the pitch cluster indices and metrical locations is used to train the prediction model - the variable-length Markov chain. The melodies generated by the model were evaluated through a questionnaire by a group of experts, and received an overall positive response.

Introduction

Research in algorithmic music aims to create interesting music using mathematical models, with the aid of computers for its generation and synthesis. The idea is to view music as a deterministic or stochastic process and to program computers to create new music according to the process. Here, the composer is often only involved to the extent of specifying certain rules or an overall structure that the composition is expected to follow. One such scenario, which is the focus of the present work, is music generation according to style. It involves training models on aspects of musical style such as note patterns, rhythm evolution and overall structure, and using these trained models to generate stylistically similar music. Probably the most popular example where computers are made to imitate musical style is David Cope's system called *Experiments in Musical Intelligence* (EMI). EMI analyses the score structure of a MIDI sequence in terms of recurring patterns (a signature), creates a database of the meaningful segments, and learns the style of a composer, given a certain number of pieces (Cope 1996).

What has come to be known as Evolutionary Music is based on the fundamental idea of how a genetic algorithm works. The generation process starts with some initial music data (a piece, melody, or loop in audio or symbolic representation), which is initialized either randomly or based on human input. Then through the repeated application of computational steps analogous to biological selection, recombination and mutation, the aim is to produce more music data of the same nature. *GenJam* (Biles 1994) is one such system developed for composing Jazz solos.

A dictionary-based prediction for automatic composition is discussed in (Assayag et al. 1999). Two dictionaries, namely, the *Motif Dictionary* and the *Continuation Dictionary* are used to represent and continue a given melody. A generation algorithm is used for continuation of a (so far) predicted sequence. A context variable is maintained which determines the maximum previous sequence to consider while making the prediction. The prediction is based on whether the context matches any of the motifs in the motif dictionary. The continuation dictionary gives the probabilities of various continuations and is used to choose the next symbol.

Among different models that have been applied to learning musical style, Markov chains have been very popular in research owing to the fact that they directly incorporate sequential information into music prediction. Ever since one of the first and well-known applications of this model to music generation in the *Illiac Suite* of Hiller & Isaacson (Hiller and Isaacson 1959), there have been several others attempting to do the same in a variety of contexts (Ames 1987). In one of the more recent approaches (Allan 2002), HMMs are employed for harmonizing Bach chorales. Here, the visible states are melody notes and the hidden states are a sequence of chords that would suggest possible harmonizations. The model predicts up to three voices at each time-step.

A more elaborate application of HMMs to style-specific music generation is the work of Paiement (Paiement 2008), where, a total of three HMMs are used to different ends. The first one models the underlying rhythm of a MIDI melody, the second the intervallic variations using simplified Narmour features (Schellenberg 1996) given the rhythm, and the third - an Input/Output HMM, predicts pitches that satisfy constraints imposed by an input chord progression and the intervals predicted by the second HMM.

It is, however, well-known that training fixed-order Markov chains requires a prohibitive amount of data with increase in the order. A solution to this problem is employed in *The Continuator* (Pachet 2003). This is a “collaborative musical instrument" which operates mainly on short melodic phrases. A reduction function interprets a given input phrase in MIDI. Sequences of symbols thus interpreted are parsed using an incremental parsing algorithm to train a variable-length Markov chain that maintains various possible sequences of symbols and their probabilities of occurrence. The system progressively learns new phrases from a musician to eventually develop a more accurate representation of her/his style.

In a similar approach, (Marchini and Purwins 2010) developed a system for the analysis of structure and style in a percussive audio sequence with the aim of generating an “arbitrarily long musically meaningful and interesting sound sequence with the same stylistic characteristics as the original". By applying several clustering thresholds simultaneously on the values of a *regularity measure* of MFCC features extracted from segmented percussive sounds, a multi-level discrete representation of the sound sequence is obtained. Periodic events are then used to estimate tempo, and meter information.

One thing to note about a majority of these approaches is their use of symbolic representations of music such as MIDI or text for reading and representing musical information (Pachet 2003), (Paiement 2008), (Biles 1994), (Cope 1996), (Allan 2002), (Conklin and Witten 1995). A MIDI-based input is assumed and hence, the problem of audio segmentation prior to developing a representation is not considered in them. MIDI is preferred because one may avoid issues related to possible inaccuracy in segmenting audio data, and focus solely on the stylistic aspect of music. Moreover, the availability of different MIDI instruments for melody and percussion also makes these approaches feasible. However, it must be noted that, at the same time, the flexibility of such systems is also very limited to only these MIDI instruments. There have been a few recent approaches that perform analysis directly on audio data for music generation. The work by (Marchini and Purwins 2010) is one such approach for generating rhythmic variations of percussive audio. This system was flexible and robust to audio recorded from several percussive sources such as drums, beat-box, etc. In the same spirit, Jehan (Jehan 2005) uses an intermediate minimal data representation directly obtained from the audio signal based on perceptual listening, for analysis and synthesis. The music/audio structure analysis system *Audio Oracle*, also directly works on audio for analysis and generation (Dubnov et al. 2007).

The framework proposed here (Figure 1) takes as input, directly, an audio signal (monophonic) and segments it into atomic components that serve as a basis for feature extraction. The focus is on rock/pop guitar and bass-guitar melodies with the aim of eventual extensions to other instruments and musical styles. The melody is first segmented into a sequence notes using onset detection and pitch estimation. As no prior assumption of a tuning system, the number and spacing of the scale notes has been made here, hierarchical agglomerative clustering is applied to the estimated pitches that underlie the scale structure, producing a set of multi-level, coarse-to-fine symbolic representations of the melody. This is followed by a novel application of the *Variance Ratio Criterion* (Calinski and Harabasz 1974) to select the appropriate clustering levels. Note onsets are aligned with beats, considering the estimated meter of the melody to create a sequence of symbols that represent the rhythm in terms of onsets/rests and the metrical locations of their occurrence. These symbols are evenly spaced in time (time-homogeneous). A joint representation based on the cross-product of the pitch cluster indices and metrical locations is used to train the prediction model - the variable-length Markov chain. An efficient implementation of this model such as the one in (Pachet 2003) is employed here. A group of experts evaluate the musical output of the system.

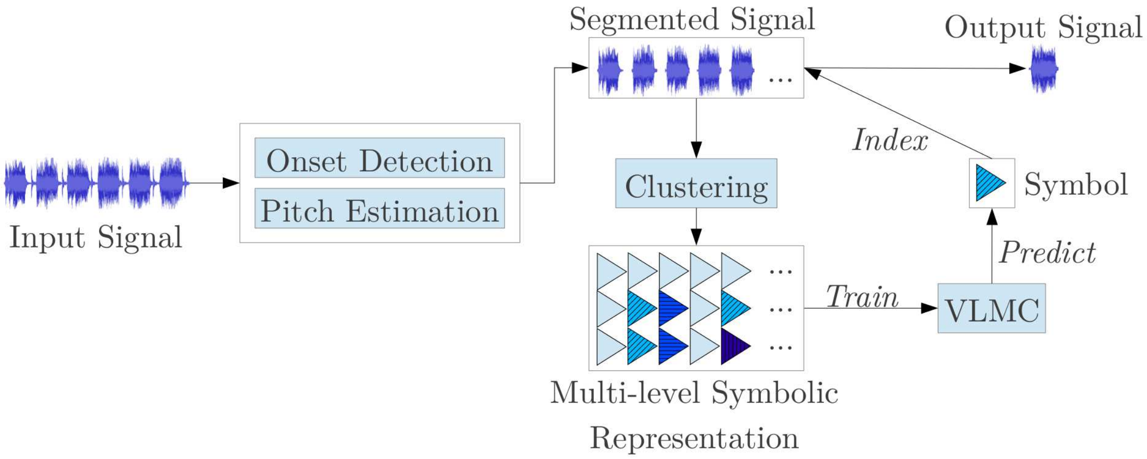


Figure 1. An overview of the proposed method. The input audio melody is first segmented into a sequence of notes. A multi-level symbolic representation, based on pitch and metrical location, is generated for each note. Sequences at each level are learned using Variable-length Markov chains (VLMC). The VLMC is used to iteratively predict the next symbol, given a previous context symbol sequence. With the predicted symbol at hand, the next segment in the generation is sampled from the audio segments corresponding to that symbol.

Segmentation

The segmentation step converts the input audio data into a sequence of mid-level features that serve as the basis for further analysis. This section describes the steps involved in obtaining an initial description of individual notes that make up the melody. This is achieved through onset detection and pitch estimation.

Onset Detection

A melody played by an electric/bass guitar is first segmented into a sequence of notes or comparably short segments which form a set of atomic units that would be used for generation. A detailed review and comparative study of different algorithms for onset detection is available in (Bello et al. 2005), and the implementation of some of these algorithms, in the *Aubio* toolbox (Brossier 2006). In the present work, the *VAMP* plugin version of this toolbox is used through the *sonic-annotator* command-line interface (Cannam 2013). On experimenting with the different methods available for onset detection, it was observed that *complex-domain* onset detection (Duxbury et al. 2003) performs consistently better than the rest, namely, *high-frequency content*, *phase* and *spectral difference* methods. Additional post-processing steps are then applied over the detected onsets to filter out false-positives incurred by Aubio. Firstly, all those onsets that occur in succession with less than 150ms between them are removed and replaced by the first one in the series. Secondly, those segments between consecutive onsets with average energy less than 40% that of the entire signal are joined with segments that respectively precede them. The particular threshold value has been found to work for the instruments employed here. These steps are illustrated in Figure 2.

Pitch Estimation

The onset detection step provides possible candidates for notes. As only monophonic melodies are dealt with here, musical pitch is the chosen feature. The *YIN* pitch estimation algorithm (de Cheveigne and Kawahara 2002) was employed for this. Once again, the implementation of this algorithm available in Aubio was used. Pitch estimation is applied to each segment between two consecutive onsets, yielding one pitch value per segment, using the *aubio-notes* function. Only one pitch is assigned to an entire segment even if the segment contains glissandi. Table 1 lists the parameter values used for onset detection and pitch estimation.

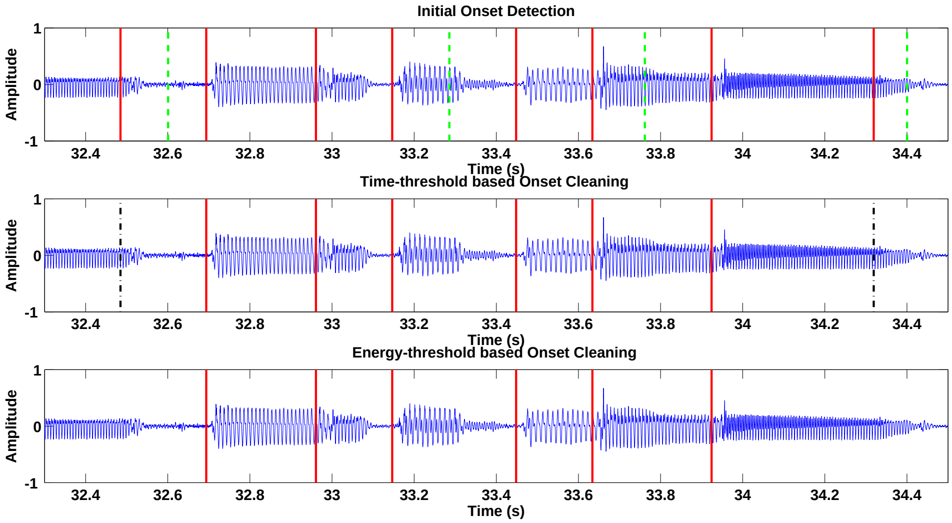


Figure 2. Different stages involved in onset detection and cleaning for an example segment of an audio signal. The first row shows the unprocessed onset detection output. The onsets denoted by the dashed lines are removed using the time-threshold. The second row shows the onset locations after this time-threshold based cleaning. The onsets denoted by the dash-dotted lines are removed using the energy-threshold. The third row shows the final onsets, denoted by the solid lines, after filtering.

Representation

In this work, melody is encoded using pitch and time representation. Firstly, the sequence of estimated pitch values is quantized using clustering, and secondly, the sequence of onset and rest times are symbolized by alignment with a beat sequence. Symbol sequences from each of these representations are then combined together to obtain a final sequence that is used to train the VLMC.

Parameter	**Value**	**Parameter**	**Value**
Pitch Type	1 (*"yinfft"*)	Onset Type	1 (*"complexdomain"*)
Step Size	512	Peak Pick Threshold	0.5
Block Size	2048	Silence Threshold	-65
Max Pitch	12,543 Hz	Wrap Range	0
Min Pitch	8 Hz	Avoid Leaps	1

Table 1. The final set of parameters used for note segmentation with aubionotes.

Pitch Clustering

In this section, we will introduce a new method to generate a pitch representation by unsupervised clustering. Although previous approaches have applied clustering to pitch Marxer et al. (2007), the novel feature of this work is that the number of clusters (pitches) is determined automatically. Following pitch estimation for each inter-onset segment using the YIN algorithm, a sequence of pitch values (in Hz) are available. These values now need to be grouped together based on similarity. This is required, as any set of frequency values corresponding to the same note are not necessarily identical but are often very close. Moreover, no prior assumption has been made about temperament, instrument tuning, number and pitch of scale notes or octave information of the given frequency values (in the case of equal temperament for example, a B, E, $F\#$, etc.), thereby making it applicable to any (non-Western) number and tuning of scale notes.

Firstly, frequency is transformed into a logarithmic scale (of base 2) in order to apply a linear distance measure for clustering pitches. Grouping of similar *log-frequency* values is realized using agglomerative single-linkage clustering. This yields a dendrogram representing their nested grouping and levels at which groupings change. That is to say, at the bottom of the dendrogram, each leaf-node corresponds to an individual log-frequency value. This corresponds to the finest similarity threshold value. At the root node, this value is maximum and all pitches are grouped into a single cluster. This gives a coarse-to-fine (top-to-bottom in the dendrogram) cluster representation of the pitch data. As the log-frequency values are scalar, the absolute value of their difference is used as the distance measure for linkage. A review of the clustering algorithm employed here and a more detailed background on the topic can be found in (Jain et al. 1999).

Such a clustering method yields clusters at multiple distance-threshold levels. Often these are too many in number, and a subset of these has to be chosen. The task of determining the best number of clusters for a given data distribution is one that has received much attention over the years in a variety of research areas. The performance of 30 different criteria for this purpose is reviewed in (Milligan and Cooper 1985). The variance ratio criterion (VRC) (Calinski and Harabasz 1974), which was verified as one of the most effective criteria in their analysis is chosen to be applied in the present case. This method estimates “the best sum-of-squares split" of the dendrogram using the Within-Group Scatter Sum (WGSS) and Between-Group Scatter Sum (BGSS). The idea is to have clusters that are well-separated (high BGSS) and, at the same time, compact (low WGSS).

An explanation of the VRC for scalar-valued data is as follows. Suppose that there are n log-frequency values $\mathbf{p} = (p_1, p_2, \ldots, p_n)$ corresponding to n segments. Then, the clustering of these values will be given by the partition of $\mathbf{p}$. Without loss of generality, it may be assumed that the mean of the total n pitches is zero. Thus, the total scatter of the n points is given by

$$t = \mathbf{p}^T \mathbf{p} = \sum_{i=1}^{n} p_i^2$$

Now, suppose the n log-frequency values are partitioned into g groups with $n_1, n_2, \ldots, n_g$ values in each group, such that $n = \sum_{i=1}^{g} n_i$. Then, for the k^{th} group, a subset of $\mathbf{p}$, p_{lk} (for $l = 1, \ldots, n_k$) represent the log-frequency values in group $\mathcal{G}_k$. One can now define the scatter for each group $\mathcal{G}_k$ with center of gravity c_k by

$$w_k = \sum_{l=1}^{n_k} (p_{lk} - c_k)^2.$$

The pooled WGSS is defined by

$$w = \sum_{k=1}^{g} w_k.$$

The BGSS is defined by

$$b = \sum_{k=1}^{g} n_k c_k^2.$$

Hence, for each partition (at each clustering level) of the n log-frequency values into g partitions, there exists the identity $t = w + b$ (Friedman and Rubin 1967).

Since the total scatter t is fixed, and a scalar, a natural criterion for grouping is to minimize w. This is equivalent to maximizing b. The Variance Ratio Criterion (VRC) uses the values of b and w at each clustering level to find "the best sum of squares split" of the dendrogram by evaluating

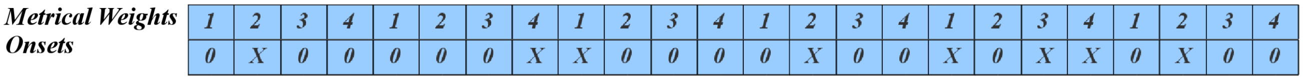
$$VRC = \frac{b/(g-1)}{w/(n-g)}.$$

The VRC is a decreasing function of the number of clusters which tends to form a local maximum when data points are grouped into natural clusters, with small within-cluster variation. It is suggested that those numbers of clusters g (at certain clustering levels) be chosen for which the VRC has an absolute or local maximum, or at least a comparatively rapid increase (Calinski and Harabasz 1974). Depending on the case, either situation might occur. And also that if there exist several local maxima, the most economical choice would correspond to the smallest number of clusters g. Following this suggestion, in the present case, these conditions are evaluated in an order such that the local maxima are given the first preference starting with the level containing the least number of clusters, followed by the slope (only if sufficient local maxima don't occur).

The cluster-levels that are the VRC maxima are sorted in increasing order of the number of clusters (or slope, if that be the case) and the top C levels are chosen for training C VLMCs. In the experiments, a value of $C = 4$ was used. This is done, firstly, as there is no a priori knowledge of the correct number of pitch-clusters that actually occur in the melody and as the VRC only provides an estimate of the best clustering levels. Secondly, selecting multiple cluster-levels also provides us with more patterns at different (coarse-to-fine) levels to learn from the data. It is not necessarily the case that each obtained cluster contains frequencies corresponding to a single note. The similarity threshold corresponding to each selected level determines the frequency precision of clustering. For instance, among the selected levels, the one that contains the least number of clusters is more likely to have grouped a wider frequency range together (that may even correspond to different notes) into the same cluster. Similar to how adjacent frequencies are quantized into “notes" in different musical traditions, even if the note instances slightly deviate from the ideal note frequency, clustering provides one such quantized representation at each level (henceforth referred to as *note-unit*). The number of note-units, depending on the melody, typically varied between 3 at the coarsest level up to 30 in the finest level.

Metrical Analysis

Every melody has an underlying rhythm that gives it a certain structure according to a beat or meter. Some of the prior approaches model the rhythmic structure of a melody explicitly to use the information in generating continuations (Paiement 2008; Pachet 2003). In the present work, a time-homogeneous sequence of symbols (symbols that occur at regular intervals of time), which serves as a representation of rhythm, is derived from the set of detected onsets. Each symbol is a beat in a beat sequence and is represented as the cross-product of metrical locations (for example, from the set $\mathcal{M} = \{1, 2, 3, 4\}$, for a metrical unit with 4 equal-length subdivisions) and $\{0, 1\}$ indicating whether an onset exists (1) or not (0) (cf. Figure 3). Such a time-homogeneous representation ensures that only onsets with the same metrical locations are treated as equivalent in the Markov chain. The system does not have to detect the beginning of a measure ($\mathcal{M} = 1$) correctly, as long as metrical locations are assigned consistently (possibly wrong) throughout the piece. The method for metrical analysis presented here was adapted to melodies from Marchini and Purwins (2010), where it was first introduced for drum sounds.



Metrical Weights	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Onsets	0	X	0	0	0	0	0	X	X	0	0	0	0	X	0	0	X	0	X	X	0	X	0	0

Figure 3. The underlying rhythmic structure of an arbitrary melody where the 'X's indicate onset locations.

The task at hand is to assign appropriate metrical locations to each of the onsets and rests in the lower row of Figure 3. Using a beat-detection algorithm, an initial beat sequence (at an arbitrary metrical level) can be obtained. Some of the onsets would coincide with the beats in this sequence, which corresponds to the metrical location "1". The procedure from here on is to progressively halve the inter-beat-interval (IBI) until at least 90% of onsets coincide with beats. In each iteration, the metrical location of the onset that coincides with a beat is noted. A rest (or equivalently, the continuation of a sustained note) is assumed at every beat where there is no coincidence with an onset. In this representation, a symbol is generated corresponding to every beat (at the final metrical level where at least 90% of the onsets match with beats) irrespective of whether there exists an onset there or not.

At a homogeneous time instant i, given a metrical location $m_i \in \mathcal{M}$, and a onset-type symbol value $v_i \in \mathcal{V}$, the cross-product rhythm symbol r_i of these two can be written as the ordered pair

$$r_i = (m_i, v_i)$$

This symbolization scheme is implemented at multiple meter-length levels, namely 1, 2 and 4. At each level, the number of symbols for representing the rhythm is twice as many as the meter length itself, in order to be able to represent either an onset or a rest at each metrical location.

Combined Representation

The final representation is also a cross-product between the symbols generated for pitches and those generated for the homogeneous rhythm representation described in the previous section. Consider a set of symbols $\mathcal{R} = \{r_1, \ldots, r_R\}$ that represent R homogeneous metrical symbols. And let $\mathcal{N} = \{n_1, \ldots, n_N\}$, a set of N elements that represents the note-units. The combined representation essentially involves generating the set of levels $\mathcal{S} = \{s_1, \ldots, s_S\}$ such that $\mathcal{S} = \mathcal{N} \times \mathcal{R}$. This would result in a total of $S = N \cdot R$ symbols.

Statistical Modeling

A variable-length Markov chain (VLMC) defines a probability for a symbol $s_j \in \mathcal{S}$, following a symbol sequence $s_{j-1}, \ldots, s_{j-k+1}$ (the *context*), given by the probability distribution $p(s_j|s_{(j-k+1)},\ldots,s_{j-1})$ where k is the maximal context-length that can influence the prediction of s_j. Given a symbol sequence, probabilities of symbol subsequences (of varying lengths) are estimated by their respective frequency counts. These counts are stored in a suffix tree. Starting from a random segment of the original sequence, segments can be iteratively appended by random sampling from the suffix tree, selecting a segment assigned to symbol s_j, following the defined context. Several VLMC are employed in parallel for the statistical analysis of the $C = 4$ combined representation symbol sequences. In (Ron et al. 1996; Bühlmann and Wyner 1999), a general method for inferencing long sequences is described. For faster computation, a simplified implementation is presented in (Pachet 2003). A suffix tree can be generated in real-time for each level of the combined representation. Each node of the tree represents a specific context that has occurred in the past. In addition, each node carries a list of continuation indexes corresponding to segment indexes matching the *context*. In every iteration of Incremental Parsing (IP) (Assayag et al. 1999), the minimum length symbol sequence that has not appeared so far generates only one new node. In contrast, in the VLMC, for all symbol sequences at least as long as this minimal length up to its maximal length, a new node including a frequency count is generated. Therefore, in the VLMC, long contexts are not automatically split up into substrings as done in IP, but taken as they are. In the VLMC, the generation respects contexts of that proper length. In the beginning, the VLMC builds up a larger tree and therefore provides more detailed information to generate sequences from a short music excerpt.

For audio, a different approach has been applied in (Dubnov et al. 2007). This method does not require an event-wise symbolic representation since it employs the factor oracle algorithm. To the best of the authors' knowledge, the VLMC has so far been applied for audio data only in (Marchini and Purwins 2010), owing to the absence of an event-wise symbolic representation prior to it.

If a particular level is fixed, the continuation indexes are drawn according to a posterior probability distribution determined by the longest context found. But the question arises as to which level has to be chosen. A trade-off exists between the level and the number of choices that it makes available. Lower levels tend to present fewer continuation choices resulting in a replication of subsequences in the original melody, while on the other hand, higher levels tend to result in random generations owing to excessive number of choices. Selecting a lower level at which a context of at least $\hat{l}$ exists (for a predetermined fixed $\hat{l}$, usually $\hat{l}$ equal to 6 or 8) works quite well for the examples. But in some cases a context of that length does not exist and the system often reaches the higher level where too many symbols are provided inducing too random generations.

In order to increase recombination of symbols and still provide good continuation, some heuristics are employed taking into account the multiple levels available for the prediction. A recombination value p, in the range $[0, 1]$ is also set. The following heuristics are used to generate the continuation at each step:

1. Set a maximal context-length $\hat{l}$ and compute the list of indexes for each level using the appropriate suffix tree. Store the achieved length of the context for each level.

2. Count the number of indexes provided by each level. Select only the levels that provide less than 75% the total number of symbols in order to guarantee a minimum of prediction specificity.

3. Among these level candidates, select only the ones that have the longest context.

4. Merge all the continuation indexes across the selected levels and remove the trivial continuation (the next segment).

5. In case there is no level providing such a context and the current block is not the last, use the next segment as a continuation.

6. Otherwise, decide randomly with probability p whether to select the next segment or rather to generate the actual continuation by selecting randomly between the

available indexes.

Evaluation

Evaluation of computational musical creativity is an issue that has, in the past, been dealt with in a variety of ways (Conklin 2003). A possible reason for this is that each system realized for music generation highlights one among many views of what may be considered "creative" or "good". Moreover, the existence of a plethora of musical styles, often a subset of which are handled by different approaches has made it difficult to establish a standard to quantitatively compare these approaches. An evaluation based purely on "accuracy", as in the case of (Paiement 2008), favors exact repetition. Such an evaluation measure is not suitable here because instead of merely reproducing the original, we rather aim at creating variations of the same. (Allan 2002) and (Dubnov et al. 2007) evaluate a generative style model employing it as a classifier: Trained for a particular style, each generative model assigns a generation probability to a given music excerpt. The style is then determined by the model with maximal probability. The aforementioned measures are objective criteria that measure the stylistic coherence or musical characteristics of the outcome. However, it is not clear to what extent they measure the aesthetic quality of the generation. For the assessment e.g. of musical interestingness, a subjective measure is more suitable. A quasi-Turing test can be employed that determines to what extent a listener can be led to believe that a melody actually generated by the system is that played by a human (or vice versa). This was employed in evaluating The Continuator (Pachet 2003). As sound synthesis is not currently the focus of this work, the generated melodies often contain certain artifacts that would make the answer to this question too obvious. Due to the absence of a reference or benchmark dataset for music generation which would make the performance of different generation systems quantitatively comparable, an alternate evaluation, based on the feedback of a group of musical experts was conducted here. It was decided that those aspects of the system that required evaluation could be focused on through a questionnaire. Experts - individuals with an educational background in music or extensive music performance experience - were approached for providing feedback on the quality of the generated melodies. A group of 10 experts (6 male, 4 female between the ages 21 and 42), each of whom is either a composer/session musician and/or a diploma-holder in classical music/jazz were consulted.

Database

The analyzed database consists of a variety of recorded pop/rock excerpts from the authors' personal collection. The main focus was on bass-lines with some importance given to guitar melodies. Bass-lines usually had a riff-like structure in which the same melodic segment repeats over a short period of time (typically 2s to 5s) with minor (if not any) variations in each repetition. Two solo guitar melodies and four accompanied bass guitar excerpts have been considered[2]. Short excerpts from each melody in the database are shown in the left column of Table 2

Onset Detection

A precision-recall based method was used to determine the performance of onset detection. This measures how accurate, and at the same time, how exhaustive a certain retrieval operation was (Brossier 2006). In this context, the data to be retrieved is the set of onsets of the original melody. Informally, a high recall would mean that nothing has been missed but there may be a lot of irrelevant results to sift through (which would imply low precision). High precision means that everything returned was a relevant result, but all the relevant items may not have been found (which would imply low recall). The f-measure is defined as the harmonic mean of the precision and recall.

A detection nearest to a ground-truth onset, within 150 ms of it was considered to be a match. Table 3 shows the evaluation of onset detection according to the described measures, with an overall precision of 91.25%, recall of 87.45% and f-measure of 88.91%.

[2] Due to copyright issues, only the guitar melodies are available at
`https://soundcloud.com/freakanth/sets/melody-prediction-examples-1`

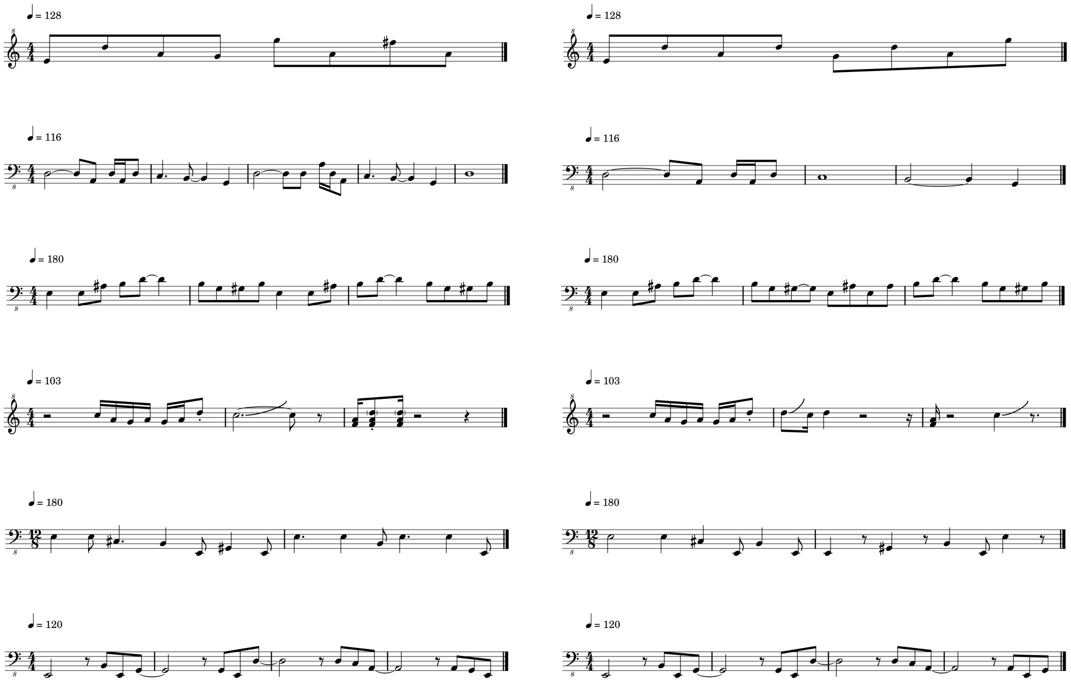


Table 2. Transcribed excerpts from the six input melodies (1st + 4th line: solo guitar, the rest: bass guitar) on the left and their corresponding generations on the right.

Sequence No.	1	2	3	4	5	6	Overall
f-measure (before)	0.755	0.899	0.749	0.716	0.731	0.606	0.742
f-measure (after)	0.951	0.954	0.894	0.905	0.776	0.851	0.889
Onset Count	60	31	54	65	111	20	331

Table 3. Table illustrating the performance of the onset detection on the evaluated examples before and after post-processing.

Expert Evaluation

A questionnaire (cf. Figure 4) was prepared and presented to each of the experts as a part of an evaluation package. The evaluation package contained six folders, each with a copy of the questionnaire, an audio file of the original music excerpt and another that is generated from it. In the case of multi-track excerpts, both the original and the generation, which in all cases were bass-lines, were overlaid on other tracks of the original recording. In each case, it was specified in the instructions that the bass-line was to be focused on. The solo guitar excerpts were played in isolation without accompaniment. The experts were first asked to listen to the original excerpt any number of times until they developed a fair idea of the melody (in terms of tempo, melodic and rhythmic patterns, structure, etc.). Following this, they were asked to listen to the generated melody and answer the questionnaire.

Responses to questions 2 through 6 of the questionnaire have been summarized in Figure 5. Question 1 is not included here as its result was the same ("Yes") across all experts and melodies. On the whole, the responses indicated that the generations were fairly interesting and coherent with the original in terms of style, motivic material, tempo, and metrical structure. In general, there was agreement in responses from different experts.

One response that was consistently the same across all the experts for all the excerpts was the observability of short note patterns in the generations that also occurred in the original (Question 1). This is encouraging and demonstrates the efficacy of the system in effectively capturing recurring recognizable melodic segments or motifs. However, when it came to the occurrence of original patterns in the generated melody (Question 2), it was pointed out that note accents changed on some occasions, for example in excerpts 1 and 5. That is, certain notes which originally occurred on a strong-beat shifted to a weak-beat and vice versa. This can be explained as follows. The beat detection algorithm generated a sequence of beat locations that occurred between approximately equal time intervals. Metrical levels corresponding to these beats at higher resolutions were generated by linear interpolation of these detected beat locations which also occurred between approximately equal time intervals. Since, at a higher resolution, the interpolated beats were very close together, on certain occasions a note-onset was matched wrongly with a generated beat adjacent to the actual beat that it corresponded to. During generation these beats were reproduced accordingly at, what was considered by some experts as, inappropriate metrical locations.

In the case of the tempi of the generated melodies (Question 3), nearly three-fourths of all the responses agreed that they were the same as those of their respective originals. It is interesting to see that even in those cases where the tempo of a generation was considered not to be the same as that of the original, it was either something else (regular, but not the same) or not determinable (irregular). This happened with excerpts 3 and 5. Also, in those cases where the tempo of the generation and the original were not considered to be the same, the experts pointed out that melodic patterns were occurring in inappropriate metrical locations (or broken riffs) in the generation. As pointed out earlier, one of the reasons for this could be the shift in note accents. Also, some of the generated melodies suffered from a drawback that note-onsets differed from the accompaniment by a slight temporal offset. This is, once again due to the approximate regularity of the beat detection algorithm. It may, therefore, have been the case that due to the delayed/premature reproduction of certain onsets, keeping track of the tempo in case of some of the melodies was not straightforward. This was the case particularly with the generations for melodies 3 and 5, both of which were considered to lack synchrony with their respective accompaniments (Question 6).

The question on overall similarity between the generation and the original received a fairly good feedback with all responses being either "very similar" or "somewhat similar". Although, it must be noted that in certain cases (excerpts 2 & 6), the generations were found to be too similar to the original. On revisiting generation parameters for these melodies, it was found that the generation for excerpt 2 used a very long context (32 symbols) as a result of which there were few note choices left following such a long context. Despite the long context still some degree of variation occurred. In the case of excerpt 6, the riff itself was composed of few notes and with minimal variations. Moreover, the riff structure and rhythmic evolution of the generation in this case are almost identical to that of the original, and when played with the accompaniment could have created such an impression.

The generation process used here, namely the variable-length Markov chain is more suitable in the case of short melodic segments rather than long solos. The main problem with longer melodies is that elements reflecting long-term coherence such as repetition of segments from time to time, logical transitions between shorter segments of a melody, are not evident. This was highlighted in some of the comments received for excerpts 4 and 5 that, respectively, contained a minute long guitar solo and a walking bass-line without a very obvious repetitive structure. As the temporal scope considered by the system was limited to a relatively short duration, it does not take into account the long-term evolution and structure of these melodies.

In general, the experts found those generations interesting, which had a regular rhythm and noticeable pitch variations from the original. It was often the case that the same variation was considered to be inappropriate or abrupt by one expert, and interesting by another. The generations for excerpts 3 and 4 received relatively better feedback and ratings than the others consistently from all the experts. It would help to use these excerpts as a reference in the future. It was also appreciated that the system was able to reproduce even silent pauses from the original melody on several occasions. The average interestingness rating for excerpts 1 through 6 were 2.3, 3.2, 3.7, 3.4, 2.0 and 3.3 (out of 5.0) respectively. All of the experts found at least two generations out of the six genuinely interesting and expressed that with some minor improvements in synthesis and overall structure of the generations, the others could also sound much better.

Conclusions & Future Work

The present work addresses the much ignored problem of generating stylistically similar melodies directly from audio, instead of symbolic data (MIDI, MusicXML, etc.). This is, in general, a more difficult problem due to the occurrence of segmentation errors that tend to propagate into any symbolic representation that is in use for generating music. A multi-level representation, similar in spirit to that of (Marchini and Purwins 2010) is employed here for the symbolic representation of notes in the melody and their metrical locations.

In particular, the representation adapts to the style of the melodic excerpt. Neither a particular scale (e.g. C major) is assumed, nor a particular temperament (e.g. equal temperament), nor the usage of a particular subset of the diatonic scale of 7 notes. The employed numbers of note-units are chosen automatically using the clustering level selection method by (Calinski and Harabasz 1974). Unlike a music generation system based on audio-to-MIDI transcription as the first stage, this system could be made to meaningfully adapt to non-equal temperaments such as in Greek, Ottoman, Arab, or African music. Moreover, the multi-level representation provides a coarse-to-fine abstraction of the melody and enables different levels of precision while choosing a continuation pitch. It allows the system to simplify the melodic structure of the example and to generate musical continuation even from short melodic examples.

While a pitch-based representation for symbolizing audio data is used here, others, like those based on intervals, or melodic contours may be incorporated, with possibly different musical output. It is worth noting that the present method for symbolizing rhythm does not take into account other than binary (e.g. triple, quintuple) and compound meters, and changing tempo (accelerando/ritardando). Introducing other than binary metrical divisions and applying the method locally would yield a more general method to handle these cases. It was also observed during evaluation that there did not exist a single value of context-length that would suit all the examples. A possible cue, it seemed, that could be explored to automatically estimate its value, is the tempo of the melody. The method, which seemed more suitable for short, riff-type melody generation, may be improved to handle longer solo-type melodies by incorporating methods that segment the melody into regions based on higher-level similarity that would help reproduce its global evolution (Mozer 1994), and use what could be short motifs that are generated by the VLMC locally. Another improvement over the current system would be the possibility to automatically determine the number of selected clustering-levels dynamically depending on the melody. With these improvements, the present approach can also be eventually extended to melodies of other musical styles played by different instruments.

The present generative model for bass and guitar melodies sheds light into some aspects of musical creativity but it is only a small step towards the goal of a comprehensive understanding of human musicianship.

Acknowledgements

H. P. was supported in part by the German Bundesministerium für Forschung und Technologie (BMBF), Grant No. Fkz $01GQ0850$.

References

Allan, M. 2002. “Harmonizing Chorales in the Style of Johann Sebastian Bach.” Master thesis, School of Informatics, University of Edinburgh.

Ames, C. 1987. “Automated Composition in Retrospect: 1956-1986.” *Leonardo Music Journal* 20(2):169–185.

Assayag, G., S. Dubnov, and O. Delerue. 1999. “Guessing the Composer’s Mind: Applying Universal Prediction to Musical Style.” In *Proc. International Computer Music Conference*. pp. 496–499.

Bello, J. P., L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler. 2005. “A Tutorial on Onset Detection in Musical Signals.” *Speech and Audio Processing, IEEE Transactions on* 13(5):1035–1047.

Biles, J. 1994. “GenJam: A genetic algorithm for generating jazz solos.” In *International Computer Music Conference*. pp. 131–137.

Brossier, P. 2006. “Automatic Annotation of Musical Audio for Interactive Applications.” Ph.D. thesis, Centre for Digital Music, Queen Mary, University of London.

Bühlmann, P., and A. J. Wyner. 1999. “Variable Length Markov Chains.” *The Annals of Statistics* 27(2):480–513.

Calinski, T., and J. Harabasz. 1974. “A Dendrite Method for Cluster Analysis.” *Communcations in Statistics - Theory and Methods* 3:1–27.

Cannam, C. 2013. ““Sonic-Annotator".” http://omras2.org/SonicAnnotator.

Conklin, D. 2003. “Music Generation from Statistical Models.” In *AISB 2003 Symposium on Artificial Intelligence and Creativity in the Arts and Sciences, In Proceedings of*. pp. 30–35.

Conklin, D., and I. H. Witten. 1995. “Multiple viewpoint systems for music prediction.” *Journal of New Music Research* 24(1):51–73.

Cope, D. 1996. *Experiments in Musical Intelligence*. A-R Editions, Madison, Wisconsin.

de Cheveigne, A., and H. Kawahara. 2002. “YIN, a fundamental frequency estimator for speech and music.” *The Journal of the Acoustic Society of America* 11(4):1917–1930.

Dubnov, S., G. Assayag, and A. Cont. 2007. “Audio Oracle: A New Algorithm for Fast Learning of Audio Structures.” In *International Computer Music Conference*. pp. 224–228.

Duxbury, C., J. P. Bello, M. Davies, and M. B. Sandler. 2003. “Complex domain Onset Detection for Musical Signals.” In *6th Conference on Digital Audio Effects (DAFx-03)*.

Friedman, H. P., and J. Rubin. 1967. “On Some Invariant Criteria for Grouping Data.” *Journal of the American Statistical Association* 62(320):1159–1178.

Hiller, L. A., and L. M. Isaacson. 1959. *Experimental Music: Composition with an Electronic Computer*. McGraw-Hill Book Co., New York.

Jain, A. K., M. N. Murty, and P. J. Flynn. 1999. “Data Clustering: A Survey.” *ACM Computing Surveys* 31(3):264–323.

Jehan, T. 2005. “Creating Music by Listening.” Ph.D. thesis, MIT Media Lab, Massachusetts Institute of Technology.

Marchini, M., and H. Purwins. 2010. "Unsupervised Generation of Percussion Sound Sequences from a Sound Example." In *Sound and Music Computing Conference*. pp. 477–484.

Marxer, R., P. Holonowicz, P. Purwins, and A. Hazan. 2007. "Dynamical Hierarchical Self-Organization of Harmonic, Motivic, and Pitch Categories." In *Workshop Music, Brain and Cognition, held at NIPS)*.

Milligan, G. W., and M. C. Cooper. 1985. "An Examination of Procedures for Determining the Number of Clusters in a Data Set." *Psychometrika* 50(2):159–179.

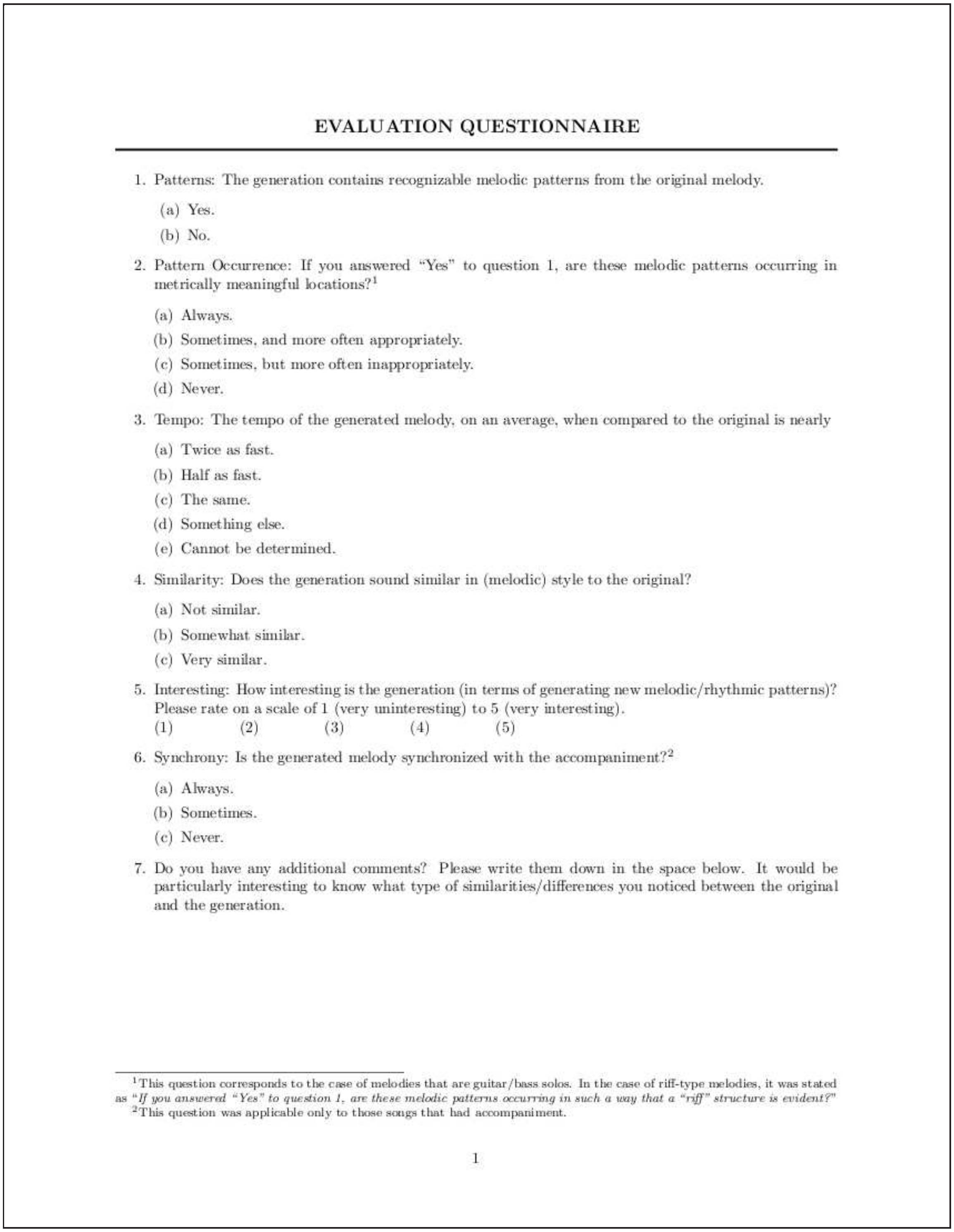
Mozer, M. C. 1994. "Neural network music composition by prediction: Exploring the benefits of psychoacoustic constraints and multi-scale processing." *Connection Science* 6(2–3):247–280.

Pachet, F. 2003. "The Continuator: Musical Interaction With Style." *Journal of New Music Research* 32(3):333–341.

Paiement, J.-F. 2008. "Probabilistic Models for Music." Ph.D. thesis, Idiap Laboratory, Ecole Polytechnique Federale de Lausanne.

Ron, D., Y. Singer, and N. Tishby. 1996. "The Power of Amnesia: Learning Probabilistic Automata with Variable Memory Length." *Machine Learning* 25(2–3):117–149.

Schellenberg, E. G. 1996. "Expectancy in melody: tests of the implication-realization model." *Cognition* 58:75–125.



EVALUATION QUESTIONNAIRE

1. Patterns: The generation contains recognizable melodic patterns from the original melody.
   (a) Yes.
   (b) No.
2. Pattern Occurrence: If you answered "Yes" to question 1, are these melodic patterns occurring in metrically meaningful locations?[1]
   (a) Always.
   (b) Sometimes, and more often appropriately.
   (c) Sometimes, but more often inappropriately.
   (d) Never.
3. Tempo: The tempo of the generated melody, on an average, when compared to the original is nearly
   (a) Twice as fast.
   (b) Half as fast.
   (c) The same.
   (d) Something else.
   (e) Cannot be determined.
4. Similarity: Does the generation sound similar in (melodic) style to the original?
   (a) Not similar.
   (b) Somewhat similar.
   (c) Very similar.
5. Interesting: How interesting is the generation (in terms of generating new melodic/rhythmic patterns)? Please rate on a scale of 1 (very uninteresting) to 5 (very interesting).
   (1) (2) (3) (4) (5)
6. Synchrony: Is the generated melody synchronized with the accompaniment?[2]
   (a) Always.
   (b) Sometimes.
   (c) Never.
7. Do you have any additional comments? Please write them down in the space below. It would be particularly interesting to know what type of similarities/differences you noticed between the original and the generation.

[1] This question corresponds to the case of melodies that are guitar/bass solos. In the case of riff-type melodies, it was stated as "*If you answered "Yes" to question 1, are these melodic patterns occurring in such a way that a "riff" structure is evident?*"

[2] This question was applicable only to those songs that had accompaniment.

1

Figure 4. The questionnaire presented to the experts for their feedback. This questionnaire was to be answered for each of the original/generated melody pair.

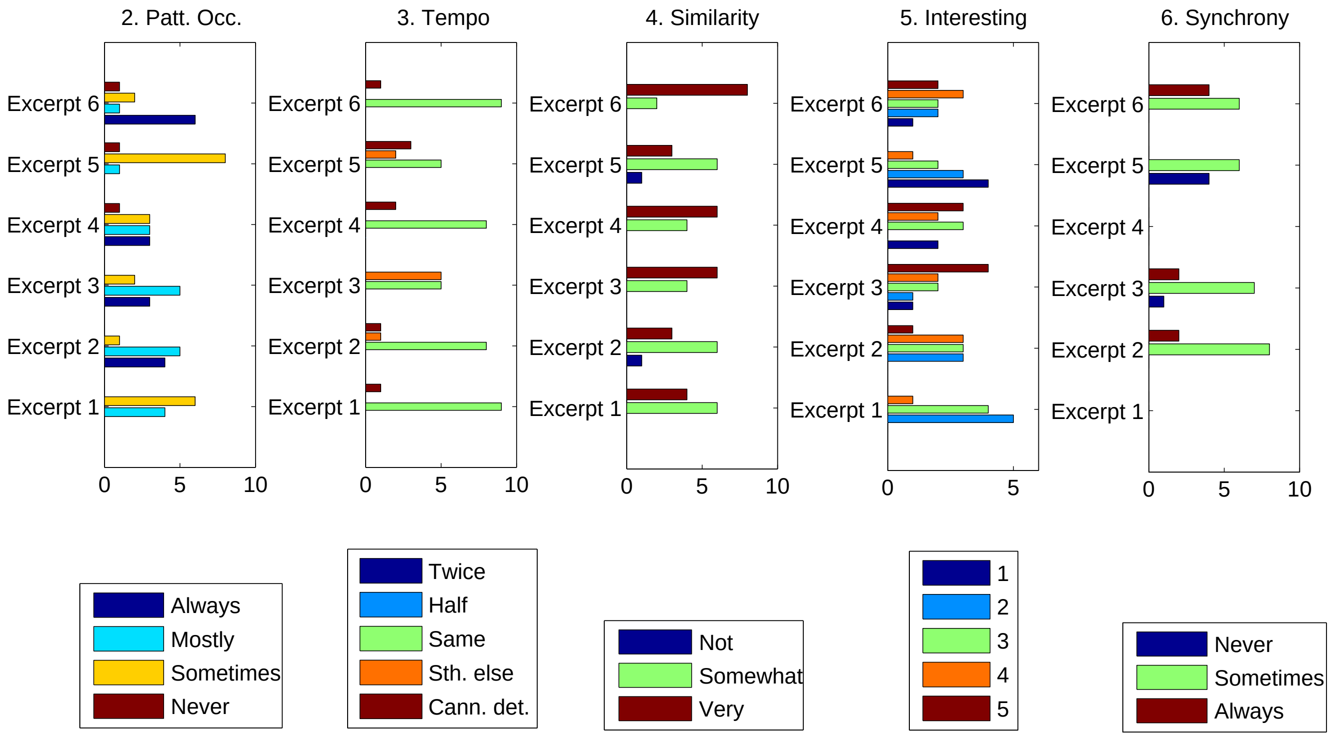


Figure 5. Summary of the 10 experts' responses to the Evaluation Questionnaire (Figure 4). Question 1 has been left out as all the responses to it were the same (a "Yes").