DISCOVERY OF REPEATED THEMES AND SECTIONS WITH VARIABLE MARKOV ORACLE

Cheng-i Wang, Jennifer Hsu and Shlomo Dubnov
Music Department
University of California, San Diego
{chw160, jsh008, sdubnov}@ucsd.edu

ABSTRACT

This abstract describes the motif discovery algorithm based on Variable Markov Oracle (VMO) for the MIREX 2015 Discovery of Repeated Themes and Sections task. VMO is a suffix automaton capable of indexing multivariate time series. The algorithm identifies repeated sub-clips from an audio recording by returning the ending points and durations for found repeated sub-clips. Chroma is used as the input to the algorithm. The implementation for this algorithm is publicly accessible.

1. INTRODUCTION

Automatic discovery of musical patterns (motifs, themes, sections, etc.) is a task defined as identifying salient musical ideas that repeat at least once within a piece [2, 9] with computational algorithms. The patterns found here may overlap with each other and may not cover the entire piece. In addition, the occurrences of these patterns could be inexact in terms of harmonization, rhythmic pattern, melodic contours, etc. Lastly, hierarchical relations between motifs, themes and sections are also desired outputs of the pattern discovery task.

Two major approaches for symbolic representations are the string-based and the geometric methods. A string-based method treats a symbolic music sequence as a string of tokens and applies string pattern discovery algorithms on the sequence [1, 13]. A geometric method views musical patterns as shapes appearing on a score and enables inexact pattern matching as similar shapes imply different occurrences of one pattern [3, 12]. For audio representations, geometric methods for symbolic representations have been extended to handle audio signals by multi $F0$-estimation with beat tracking techniques [4]. Approaches adopted from music segmentation tasks using self-similarity matrices and greedy search algorithms are proposed in [14, 15].

The algorithms described in this abstract can be seen as a string-based method in which input features are symbolized. The algorithms consists of two blocks: 1) feature extraction with post-processing routines and 2) the pattern finding algorithm. For audio representations, chroma features are extracted and post-processed based on musical heuristics, such as modulation, beat-aggregation, etc. The core of the pattern finding algorithm is a *Variable Markov Oracle* (*VMO*). A *VMO* is a data structure capable of symbolizing a signal by clustering the observations in a signal, and is derived from the *Factor Oracle* (*FO*) [11] and *Audio Oracle* (*AO*) [7] structures.

2. FEATURE EXTRACTION

The routines for extracting the chromagram from an audio recording used in this algorithm is as follows. For a mono audio recording sampled at 44.1 kHz, the recording is first downsampled to 11025 Hz. Next, a spectrogram is calculated using a Hann window of length 8192 with 128 samples overlap. Then the constant-Q transform of the spectrogram is calculated with frequency analysis ranging between $f_{min} = 27.5$ Hz to $f_{max} = 5512.5$ Hz and 12 bins per octave. Finally, the chromagram is obtained by folding the constant-Q transformed spectrogram into a single octave to represent how energy is distributed among the 12 pitch classes.

To achieve the pattern discovery on a music metrical level, the chroma frames are aggregated with a median filter according to the beat locations found by a beat tracker [8] conforming to the music metrical grid. For finer rhythmic resolution, each beat identified is spliced into two subbeats before chroma frame aggregation. Last, the sub-beat-synchronous chromagram is whitened with a log function. Whitening boosts the harmonic tones implied by the motifs so that the difference between the same motif with and without harmonization is reduced.

3. VARIABLE MARKOV ORACLE

In this section, we briefly introduce VMO. Technical details and pseudo codes could be found in [17, 19]. FO is a suffix automaton devised for retrieving patterns from a symbolic sequence [11]. AO is the signal extension of FO capable of indexing multi-variate timeseries, and has been applied to audio query [5] and audio structure discovery [6]. As mentioned earlier, FO tracks the longest repeated suffix of every "letter" along a symbolic sequence by constructing an array, `sfx`, storing the position of where the

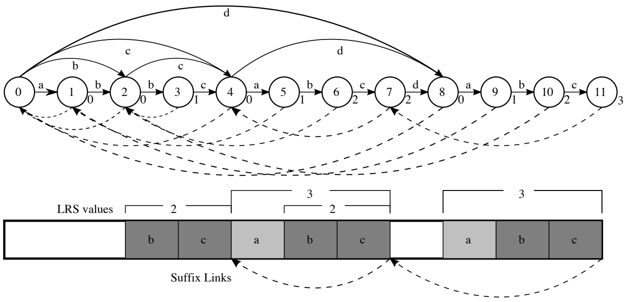

Figure 1. (Top) A VMO structure with symbolized signal $\{a, b, b, c, a, b, c, d, a, b, c\}$, upper (solid) arrows represent forward links with labels for each frame and lower (dashed) are suffix links. Values outside of each circle are the `lrs` value for each state. (Bottom) A visualization of how patterns $\{a, b, c\}$ and $\{b, c\}$ are related to lrs and sfx.

longest repeated suffix happened, and an `lrs` array storing the length for the corresponding longest repeated suffix. AO extends FO by implicitly symbolizing each incoming observation of a multi-variate time series. VMO combines FO and AO in the sense that the symbolization of AO is made explicit in VMO. The explicit symbolization is done by assigning labels to the frames linked by suffix links. As a result, VMO is capable of symbolizing a signal by clustering the feature frames in the signal and keeps track of where and how long the longest repeated suffix is for each observation frame.

Since VMO stores the length and positions of the repeated suffixes within a time series, using VMO to find motifs within a time series is then straight forward. In [19], a motif finding algorithm using VMO for multi-variate time series is proposed. The algorithm sweeps through the length and location of longest repeated suffixes for each frame and filters out the repeated suffixes that are non-trivial (sub-motifs enclosed by longer motifs are considered trivial) and above a minimum length. Fig. 1 shows an example of a constructed *VMO* and how `lrs` and `sfx` are related to motif discovery.

4. MOTIF DISCOVERY ALGORITHM WITH VMO

For the specific task of repeated themes discovery, a pattern discovery algorithm is devised based on *VMO* and shown in Algorithm 1. The idea behind the algorithm is to track patterns by following `sfx` and `lrs`. `sfx` provides the locations of repeated suffixes and `lrs` contains the length for these repeated suffixes. In line 5 of Algorithm 1, state i is checked to make sure no redundant patterns are recognized and the lengths of patterns are larger than a user-defined minimum L. From line 6 to 10, the algorithm recognizes occurrences of established patterns and from line 11 to 15 it detects new patterns and stores them into $Pttr$ and $PttrLen$. Algorithm 1 returns $Pttr$, $PttrLen$ and K. $Pttr$ is a list of lists with each $Pttr[k], k \in \{1, 2, \ldots, K\}$, a list containing the ending indices of different occurrences of the kth pattern found. K is the total number of patterns found. $PttrLen$ has K values representing the length of

Algorithm 1 Pattern Discovery using *VMO*

Require: constructed *VMO*, V, of length T and a minimum pattern length L.
Ensure: $\texttt{sfx}, \texttt{rsfx}, \texttt{lrs} \in V$
1: Initialize $Pttr$ and $PttrLen$ as empty lists.
2: Initialize $prevSfx = -1, K = 0$
3: **for** $i = T : L$ **do**
4:   $pttrFound = False$
5:   **if** $i - \texttt{lrs}_V[i] + 1 > \texttt{sfx}_V[i] \wedge \texttt{sfx}_V[i] \neq 0 \wedge \texttt{lrs}_V[i] \geq L$ **then**
6:     **if** $\exists k \in \{1, \ldots, K\}, \texttt{sfx}[i] \in Pttr[k]$ **then**
7:       Append i to $Pttr[k]$
8:       $PttrLen[k] \leftarrow \min(lrs[i], PttrLen[k])$
9:       $pttrFound = True$
10:     **end if**
11:     **if** $prevSfx - \texttt{sfx}[i] \neq 1 \wedge pttrFound == False$ **then**
12:       Append $\{\texttt{sfx}[i], i, \texttt{rsfx}[i]\}$ to $Pttr$
13:       Append $\min\{\texttt{lrs}[\{\texttt{sfx}[i], i, \texttt{rsfx}[i]\}]\}$ to $PttrLen$
14:       $K \leftarrow K + 1$
15:     **end if**
16:     $prevSfx \leftarrow \texttt{sfx}[i]$
17:   **else**
18:     $prevSfx \leftarrow -1$
19:   **end if**
20: **end for**
21: **return** $Pttr, PttrLen, K$

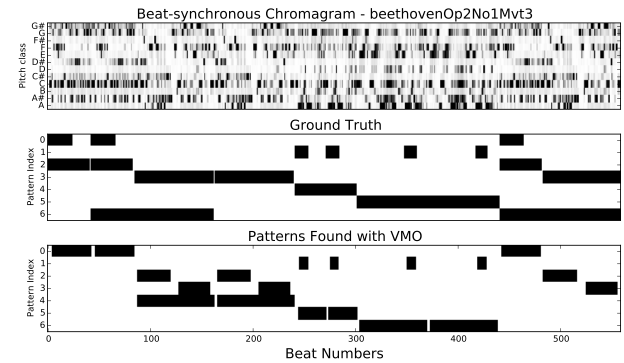

Figure 2. (Top) Beat-synchronous Chromagram, (Middle) Ground truth from JKU dataset. (Bottom) Found patterns by Algorithm 1.

the kth pattern in $Pttr$.

After the feature sequence O is extracted from the audio recording as described in the section 2, thresholds $\theta \in \{0.0, 0.001, 0.002, \ldots, 2.0\}$ are used to construct multiple *VMO*s with O, then the one *VMO* with the highest *IR* is fed into Algorithm 1 with L set to 5 empirically to find repeated themes and their occurrences. The result for finding repeated themes in one of the audio recordings from the dataset is shown in the bottom plot of Fig 2.

To consider transposition (moving patterns up or down by a constant pitch interval), the distance function used for *VMO* structures is a cost function with transposition invariance. For a transposition invariant cost function, a cyclic permutation with offset k on an n-dimensional vector $\mathbf{x} = (x_0, x_1, \ldots, x_{n-1})$ is defined as

$$cp_k(\mathbf{x}) := \{x_i \rightarrow x_{(i+k \bmod n)}, \forall i \in (0, 1, \ldots, n-1)\},$$

and the transposition invariant dissimilarity d between two vectors x and y is defined as, $d = \min_k\{\|x - cp_k(y)\|_2\}$. $n = 12$ for the chroma vector, and the cost function is used during the *VMO* construction.

5. REFERENCES

[1] Emilios Cambouropoulos, Maxime Crochemore, Costas S Iliopoulos, Manal Mohamed, and Marie-France Sagot. All maximal-pairs in step–leap representation of melodic sequence. *Information Sciences*, 177(9):1954–1962, 2007.

[2] Tom Collins. Discovery of repeated themes and sections. *Retrieved 4th May, http://www.music-ir.org/mirex/wiki/2013:Discovery_of_Repeated_Themes_&_Sections*, 2013.

[3] Tom Collins, Andreas Arzt, Sebastian Flossmann, and Gerhard Widmer. SIARCT-CFP: Improving precision and the discovery of inexact musical patterns in point-set representations. In *ISMIR*, pages 549–554, 2013.

[4] Tom Collins, Sebastian Böck, Florian Krebs, and Gerhard Widmer. Bridging the audio-symbolic gap: The discovery of repeated note content directly from polyphonic music audio. In *Audio Engineering Society Conference: 53rd International Conference: Semantic Audio*. Audio Engineering Society, 2014.

[5] Arshia Cont, Shlomo Dubnov, Gérard Assayag, et al. Guidage: A fast audio query guided assemblage. In *International Computer Music Conference*, 2007.

[6] Shlomo Dubnov, Gérard Assayag, and Arshia Cont. Audio oracle analysis of musical information rate. In *Semantic Computing (ICSC), 2011 Fifth IEEE International Conference on*, pages 567–571. IEEE, 2011.

[7] Shlomo Dubnov, Gerard Assayag, Arshia Cont, et al. Audio oracle: A new algorithm for fast learning of audio structures. In *International Computer Music Conference*, 2007.

[8] Daniel P. W. Ellis. Beat tracking by dynamic programming. *Journal of New Music Research*, 36(1):51–60, 2007.

[9] Berit Janssen, W. Bas de Haas, Anja Volk, and Peter Kranenburg. Discovering repeated patterns in music: potentials, challenges, open questions. In *10th International Symposium on Computer Music Multidisciplinary Research*. Laboratoire de Mécanique et d'Acoustique, 2013.

[10] Arnaud Lefebvre and Thierry Lecroq. Compror: online lossless data compression with a factor oracle. *Information Processing Letters*, 83(1):1–6, 2002.

[11] Arnaud Lefebvre, Thierry Lecroq, and Joël Alexandre. An improved algorithm for finding longest repeats with a modified factor oracle. *Journal of Automata, Languages and Combinatorics*, 8(4):647–657, 2003.

[12] David Meredith, Kjell Lemström, and Geraint A. Wiggins. Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. *Journal of New Music Research*, 31(4):321–345, 2002.

[13] Oriol Nieto and Morwaread Farbood. Perceptual evaluation of automatically extracted musical motives. In *Proceedings of the 12th International Conference on Music Perception and Cognition*, pages 723–727, 2012.

[14] Oriol Nieto and Morwaread Farbood. MIREX 2013: Discovering musical patterns using audio structural segmentation techniques. *Music Information Retrieval Evaluation eXchange, Curitiba, Brazil*, 2013.

[15] Oriol Nieto and Morwaread Farbood. Identifying polyphonic patterns from audio recordings using music segmentation techniques. In *The 15th International Society for Music Information Retrieval Conference*, 2014.

[16] Greg Surges and Shlomo Dubnov. Feature selection and composition using pyoracle. In *The 9th Artificial Intelligence and Interactive Digital Entertainment Conference*, 2013.

[17] Cheng-i Wang and Shlomo Dubnov. Guided music synthesis with variable markov oracle. In *The 3rd International Workshop on Musical Metacreation, 10th Artificial Intelligence and Interactive Digital Entertainment Conference*, 2014.

[18] Cheng-i Wang and Shlomo Dubnov. Variable markov oracle: A novel sequential data points clustering algorithm with application to 3d gesture query-matching. In *International Symposium on Multimedia*, pages 215–222. IEEE, 2014.

[19] Cheng-i Wang and Shlomo Dubnov. Pattern discovery from audio recordings by variable markov oracle: A music information dynamics approach. In *Acoustics, Speech, and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015.