DISCOVERY OF REPEATED THEMES AND SECTIONS WITH PATTERN CLUSTERING

Tsung-Ping Chen
Institute of Information Science,
Academia Sinica, Taipei, Taiwan
tearfulcan-on@iis.sinica.edu.tw

Li Su
Institute of Information Science,
Academia Sinica, Taipei, Taiwan
lisu@iis.sinica.edu.tw

ABSTRACT

Two versions (symbolic-monophonic version and symbolic-polyphonic version) of the pattern discovery algorithms were submitted to the 2017 MIREX competition on Discovery of Repeated Themes and Sections. Both the submitted algorithms find patterns along with repeated sections through a geometric approach, and picks out theme-like patterns by the knowledge of repeated sections.

1. INTRODUCTION

Pattern discovery algorithm which adopts geometric approach like variants of SIA described by Meredith et al. [1], which takes multidimensional representation of a music as input and finds patterns through searching all displacement vectors of each note translating to others, can efficiently find out all repeated patterns in a musical piece. However, the conventional approach of SIA suffers from two issues. First, as the result of a brute-force-search strategy, usually tens of thousands of patterns would be found in a single piece, and most of them are of little musical interest. Though the variants of SIA attempt to extract only the interesting repeated patterns, it turns out that the output amount is still huge for a theme discovery task. Second, SIA uses an iterative way to find the set of occurrences for a given pattern, and thus has the time complexity $O(kn^3)$, as reported by Meredith.

Instead of using iterative strategy for finding all the occurrences of each pattern, the proposed algorithm utilizes the clustering method which can directly get all sets of occurrences for all patterns. Then, the output is constrained with the assumption that theme-like patterns should occur at the beginning of each repeated section.

2. INPUT DATA

The submitted algorithm assumes that the music to be analyzed is represented as a set of note events, in which each event indicates the information about the note, such as onset, pitch, duration, and so on. Both the submissions take the JKU-PDD CSV format files as input, in which onset (in crotchet beats), pitch (in MIDI note number); morphetic pitch number, duration (in crotchet beats), and staff number are included.

3. PATTERN DISCOVERY

The main idea of SIA is to inspect all the displacements, i.e. onset differences and pitch differences, between every two notes within the piece. Suppose there is a pattern occurring twice in the space, say A and B. Then, the distances from all notes in A to their counterparts in B should be the same. Thus, the goal of this algorithm is to find all group of displacement vectors with equal value.

In practice, however, it is hard to find patterns only by searching groups of vectors with “equal” displacement, since the onset differences and pitch differences of some notes in a pattern sometime will vary slightly, due to its transposition, modulation, or variation of the pattern comparing to its other occurrences. As a result, the task of grouping vectors should allow some tolerance for both onset difference and pitch difference. Moreover, it is possible that the origins of some vectors in a group are distant from that of the other vectors. In this case, those vectors with “remote” origins should not belong to the group and need to be split up into another group.

4. PATTERN CLUSTERING

The repeated patterns found in Section 3 are then clustered in order to put all the occurrences of a pattern together. The feature for clustering is the chroma vector of each pattern. To compute the transposition-invariant similarity between these patterns, all the chroma vectors are transposed to a certain key according to a selected chroma by examining to which key the chroma is closest to the selected chroma. The purpose of transposition is to ensure that transposed patterns may not be excluded from original one.

5. OUTPUT CONSTRAINT

Although it is hard to tell whether a pattern is theme or not, it seems useful to assume that a theme tends to appear along with large repeated patterns. For example, the theme(s) of a Sonata occur just at the opening of Exposition, which is usually a repeated section in Sonata Form. Thus, if there are large repeated patterns or repeated sections in a musical piece, the output patterns can be constrained to those in the large repeated patterns or repeated sections.

	P_{est}	R_{est}	$F1_{est}$	P_{occ}	R_{occ}	$F1_{occ}$	P_3	R_3	TLF1
				c = 0.75					
Bach	0.94444	0.62963	0.75556	0.91702	0.91702	0.91702	0.64206	0.42804	0.51365
Beethoven	0.52779	0.74308	0.61720	0.84615	0.84615	0.84615	0.45540	0.71850	0.55747
Chopin	0.31250	0.93750	0.46875	0.93750	0.93750	0.93750	0.20866	0.62598	0.31299
Gibbons	0.33217	0.21609	0.26184	0.00000	0.00000	0.00000	0.33694	0.21014	0.25884
Mozart	0.38182	0.95455	0.54545	0.95455	0.95455	0.95455	0.39048	0.97619	0.55782
Average	0.49974	0.69617	0.52976	0.73104	0.73104	0.73104	0.40671	0.59177	0.44015

Table 1. Evaluation of monophonic version on the JKU-PDD.

	P_{est}	R_{est}	$F1_{est}$	P_{occ}	R_{occ}	$F1_{occ}$	P_3	R_3	TLF1
				c = 0.75					
Bach	0.62063	0.47919	0.54082	0.68750	0.73214	0.70912	0.23991	0.19378	0.21439
Beethoven	0.47483	0.71154	0.56957	0.98000	1.00000	0.98990	0.36732	0.55418	0.44181
Chopin	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
Gibbons	0.40630	0.27460	0.32772	0.00000	0.00000	0.00000	0.24432	0.18877	0.21298
Mozart	0.44286	0.85714	0.58399	1.00000	1.00000	1.00000	0.50000	0.91667	0.64706
Average	0.38892	0.46449	0.40442	0.53350	0.54643	0.53980	0.27031	0.37068	0.30325

Table 2. Evaluation of polyphonic version on the JKU-PDD.

6. EVALUATION

For the evaluation of the outputs to the ground truth, the metrics provided by Collins and Meredith [2] are used. The results of the evaluation on the JKU-PDD [3] are shown in Table 1 and Table 2 (for monophonic version and polyphonic version respectively). In the monophonic version, the average runtime is around 2.5 seconds using Intel i7-6700 processor @ 3.4 GHz, while the average memory usage is 125 MB; in polyphonic version, the average runtime is around 18 seconds using Intel i7-6700 processor @ 3.4 GHz, while the average memory usage is 290 MB.

7. REFERENCES

[1] David Meredith, Kjell Lemström, and Geraint A. Wiggins. Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. In *Cambridge Music Processing Colloquium*, 2003.

[2] Tom Collins. 2017: Discovery of Repeated Themes & Sections, 2017. http://www.music-ir.org/mirex/wiki/2017:Discovery_of_Repeated_Themes_%26_Sections. Accessed on 8 September 2017.

[3] Tom Collins. JKU Patterns Development Database, 2013. Available at https://dl.dropbox.com/ u/11997-856/ JKU/JKUPDD-Aug2013.zip.