COMPRESSION-BASED SIMILARITY MEASURES IN SYMBOLIC, POLYPHONIC MUSIC

Teppo E. Ahonen, Kjell Lemström, Simo Linkola
Department of Computer Science
University of Helsinki
{teahonen,klemstro,slinkola}@cs.helsinki.fi

ABSTRACT

We present a novel compression-based method for measuring similarity between sequences of symbolic, polyphonic music. The method is based on mapping the values of binary chromagrams extracted from MIDI files to tonal centroids, then quantizing the tonal centroid representation values to sequences, and finally measuring the similarity between the quantized sequences using Normalized Compression Distance (NCD). The method is comprehensively evaluated with a test set of classical music variations, and the highest achieved precision and recall values suggest that the proposed method can be applied for similarity measuring. Also, we analyze the performance of the method and discuss what should be taken into consideration when applying the method for measurement tasks.

1. INTRODUCTION

Measuring similarity between symbolically encoded music has been studied extensively, with several approaches existing. Similarity measuring between pieces of polyphonic music is far from trivial, but the extensive amount of applications (for example, query by example music retrieving) that require such measuring motivates to explore novel techniques for the task.

Here, we present an approach that is based on mapping the pitches present in a given time frame to tonal centroid vectors, quantizing the tonal centroid values, and representing the obtained information as a sequence of characters. For measuring similarity between sequences, we apply normalized compression distance (NCD) [2], which is a parameter-free, quasi-universal similarity metric [2].

We believe that NCD is applicable to polyphonic, symbolic music similarity measuring, since there are already several existing, well-performing classification and clustering approaches utilizing NCD as the similarity metric. However, the previously developed NCD-based methods all seem to rely on rather crude representations of music, such as skyline reduction or melodic contour description, both which lose a significant amount of tonal information. We wish to keep as much of the tonal information included as possible but the amount of parallel pitch values may be large, even if the octave information is ignored and values are reduced into a 12-dimension chromagram (also known as pitch class profile). Therefore, dimension reduction is needed, and for this, we use a method with musical knowledge. This is where the tonal centroid representation [4] seems a feasible solution, as it turns a 12-dimension chromagram into a 6-dimensional representation, still holding most of the harmonic information and also some of the melodic information.

To see how well our approach performs in a particular similarity measuring task, we use it to determine whether a given piece of music is a variation of an original theme included in the training data. This task is challenging and a very suitable way to evaluate our method, as it is objective (in comparison to, for example, genre classification), and in order to be successful, the method must retain tonal information and still be able to measure the essential musical similarity without too much complexity. We evaluate our approach with a set of 18 classical compositions and their variations, and based on the results, discuss the remarks we discovered in our evaluations, suggesting several issues that need to be addressed when using NCD for similarity measuring with polyphonic music, and present ideas for future research work.

The rest of this paper is organized as follows. In Section 2 we review methods of similarity measuring between symbolic, polyphonic pieces of music. In Section 3 we present the methods used for extracting tonal information from MIDI files and representing the information in a format suitable for compressor-based similarity measurement. Experiments on the method are presented in Section 4 and conclusions in Section 5.

2. RELATED WORK

In the early days of similarity measuring of symbolically encoded music, linear string representation together with string matching methods was most often used. The approach, however, does not work with polyphonic music in general, except for some very specific cases (see e.g. [3,7]). Recently several authors have applied geometric modeling of music for the task (see e.g. [9, 13, 16]). Many of these algorithms are based on computing translation vectors, which makes them transposition-invariant and also allows for extra intervening notes that appear in one of the pieces of music under consideration but not in the other. Recent geometric methods have also challenged timing problems; the pieces of music under consideration may be either time-scaled [5, 13] or time-warped [6] copies of each other.

Interesting alternative approaches can be found in [10, 17]. In his PhD thesis [17], Rizo introduces a tree representation for polyphonic music and shows how to apply different tree matching algorithms for various similarity cases including variation recognition. He also shows how to use his generic representation for implementing and illustrating schenkerian reduction. In [10], Marsden concentrates on schenkerian reduction in recognizing polyphonic variations of the classical era. To this end, he divides polyphony in three 'voices': melody, middle and bass. Melody and bass contain the highest and lowest notes, respectively, while middle voice contains all the notes belonging to neither of the previous two. Using such a reduction of polyphony he studies whether a method based on schenkerian reduction would outperform another method based just on surface analysis, but found no evidence to support that hypothesis.

In the literature, one can also find several compression-based approaches for similarity measuring in symbolic music. In [2], NCD is used as the similarity measure for composer- and genre-based clustering experiments. The method extracts key-invariant melodic contours from the MIDI files, and constructs a distance matrix for the clustering algorithm using NCD as the similarity metric. In [8], Kolmogorov complexity is estimated as the size of the dictionary produced by the LZ78 compression algorithm. Based on this estimation, k-NN classification is applied for melodies represented as both absolute and relative values. In [11], NCD is applied as similarity measure for string representations of music, obtained by converting symbolic music with a graph structure representation. In a recent study, NCD is used as one of the possible similarity metrics for measuring similarity between bass lines [15]. The bass line melodic interval histogram similarities are used as a feature for genre classification. In [1], NCD is used for measuring similarity between MIDI pieces. The polyphonic MIDI melodies are converted into monophonic versions by taking only the highest pitch value present in a time slice (also known as the skyline representation).

Another method based on measuring the amount of similar information between pieces of polyphonic music is presented in [12]. Their work is based on using Kullback-Leibler divergence to measure similarity between chromagram sequences. Based on the chromagrams, a 24-chord lexicon (all the major and minor triad chords in the western tonal scale) is used to create a probability distribution, and then Kullback-Leibler divergence is applied to measure the similarity between the distributions of the query and target models. The method is evaluated with classical variation recognition.

3. METHODOLOGY

Let us next introduce the methods that we will use for extracting tonal information from the MIDI files and how to represent the information in an appropriate format for the compression-based similarity measurement. Figure 1 depicts a blueprint of the system components and the data processing.

3.1 Binary Chroma Representation

Chroma vector is a 12-dimension representation of notes, stripped from octave information, that are present in a given time frame. We encode MIDI files as binary chroma vector sequences by first chopping the piece of music into slices of $\frac{1}{4}$ the length of the duration of a quarter note, then concatenating these slices to form the sequences and, finally, mapping each sequence with notes playing in that particular time frame.

3.2 Tonal Centroid Representation

To reduce the variation of $2^{12} = 4096$ possible chromagram vectors, the method explained in [4] is used to transform chroma vectors into tonal centroid vectors. As the tonal centroid vector has values ranging in $-1 \leq k \leq 1$, $k \in \mathbb{R}$, for all six dimensions, we quantize each value to 0 or 1 using the median of all the possible values of chroma vectors in the particular dimension of tonal centroid representation as a threshold, thus effectively lessening the alphabet to $2^6 = 64$.

The tonal centroid vector for time frame t is given by formula:

$$\zeta_t(d) = \frac{1}{||c_t||} \sum_{p=0}^{11} \Phi(d,p) c_t(p) \qquad (1)$$

where $0 \leq d \leq 5$ and $0 \leq p \leq 11$. $||c_t||$ denotes the L_1-norm of chroma vector c_t, p is the pitch class index in c_t, d represents which of the dimensions of tonal centroid is being calculated and $\Phi = [\phi_0, \phi_1, \ldots, \phi_{11}]$ is the transformation matrix where

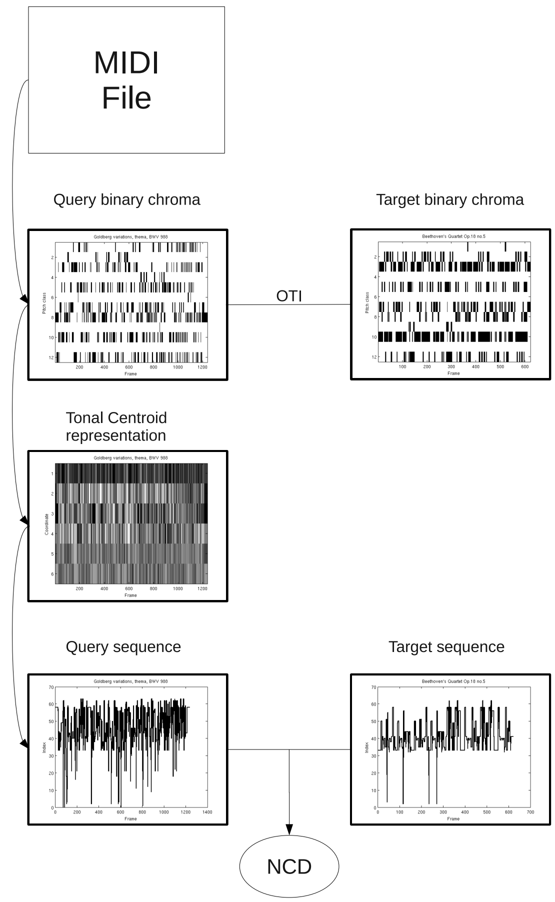

Figure 1. Blueprint of the components.

$$\phi_p = \begin{pmatrix} \Phi(0,p) \\ \Phi(1,p) \\ \Phi(2,p) \\ \Phi(3,p) \\ \Phi(4,p) \\ \Phi(5,p) \end{pmatrix} = \begin{pmatrix} \sin p\frac{7\pi}{6} \\ \cos p\frac{7\pi}{6} \\ \sin p\frac{3\pi}{2} \\ \cos p\frac{3\pi}{2} \\ \frac{1}{2}\sin p\frac{2\pi}{3} \\ \frac{1}{2}\cos p\frac{2\pi}{3} \end{pmatrix}, 0 \leq p \leq 11. \qquad (2)$$

3.3 Normalized Compression Distance

To measure the similarity of two different pieces of music we use NCD to see how close the quantized 6-dimensional tonal centroid vectors are to each other. The NCD is shown to be a quasi-universal similarity metric in [2] as it approximates normalized information distance (NID) up to an error depending on the quality of the compressor that is used in the calculation.

Normalized information distance is based on Kolmogorov complexity of the given object and is a universal metric in the sense that it uncovers all the similarities of the objects at the same time. Kolmogorov complexity of an object is the length of the shortest binary program that outputs the object on a universal computer. If $K(x)$ denotes Kolmogorov complexity of x, then $K(x|y)$ denotes conditional Kolmogorov complexity of x given y as an input. NID for x and y is given by the formula

$$NID(x,y) = \frac{\max\{K(x|y), K(y|x)\}}{\max\{K(x), K(y)\}}. \qquad (3)$$

As Kolmogorov complexity of an object is non-computable, we cannot calculate NID. However, we can approximate it with standard compression algorithms.

Let C be a lossless data-compression algorithm which satisfies the requirements of reference compressor mentioned in [2]. C can be used to approximate K. $C(x)$ is used to denote the length of x compressed with C and $C(xy)$ to denote the length of concatenated x and y compressed with C. $C(x|y)$ can also be defined as $C(x|y) = C(xy) - C(y)$, which tells us the amount of bits of information in x related to y. Now the normalized compression distance can be given by the formula

$$NCD(x,y) = \frac{C(xy) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}}, \qquad (4)$$

which is shown in [2] to approximate the NID formula mentioned in Equation 3.

4. EXPERIMENTS

4.1 Test Data

We use the polyphonic classical music variation dataset described in [17] for our experiments. It is a fairly extensive collection of classical themes and their variations, ranging over different values in terms of instrumentation, lengths, eras and numbers of voices. In addition, we also included two more compositions: the Haydn variations by Brahms, Op. 56 (nine variations of a theme), and the Piano Sonata number 11 by Mozart, KV 331 (six variations of a theme). The total size of our dataset is 18 themes and 84 variations, totalling 102 pieces of music, with the highest number of variations for a theme being 30 and lowest being 1.

4.2 Evaluations and results

For evaluations, we used the original themes as the training data and the variations as queries. We ran the classification tests with several different parameters. For each different evaluation, the overall accuracy and average precision, average recall, and average f-measure are reported.

First, we had to select the data-compression algorithm. We ran the classification test with bzip2 and prediction by

Compressor	bzip2	PPMZ
Accuracy	0.393	0.536
Precision	0.526	0.632
Recall	0.391	0.474
F-measure	0.448	0.542

Table 1. Comparison between used compression algorithms.

partial matching (PPMZ) compression algorithms. The results for the two algorithms are presented in Table 1. Since using the PPMZ algorithm produced slightly better results, we conducted the rest of the evaluations using it as the selected compression algorithm.

In addition to the assumption that variations are performed in the same key as the original, we wanted to be able to measure similarity between pieces of music in different keys. In order to transpose two chroma sequences into the same key, we used Optimal Transposition Index (OTI) [14], where the most likely transposition between two chromagrams is calculated by measuring the dot product between all 12 possible transpositions of the chromagrams summed over time and normalized. Having calculated OTI, we rotated the query binary chromagram according to the OTI value before making the tonal centroid transformation and writing the transformed sequence to a file.

The sequences produced by our method have occasional short sections of outliers, caused by transitional anomalies produced by our time-slicing MIDI-extraction method and resulting in overall noisy sequences. Such noise can be harmful for compression-based similarity measuring, since noise in data reduces the compressibility, thus resulting possibly in a lower performance in classification. However, the anomalies could also be significant distinguishing features in the sequences. An illustration of changes in the sequence indices is depicted in Figure 2, with several noisy spikes clearly visible. To get rid of the transitions and make the sequences smoother, we experimented with median filtering, and ran median filter of order 5 to the sequences before writing them into files. An example of a median-filtered sequence is depicted in Figure 3.

The results for the evaluations with OTI and median-filtered sequences are presented in Table 2.

4.2.1 Clustering experiment

In addition to the supervised classification, we carried out an experiment with unsupervised machine learning, and for this, we ran a k-medians clustering for the whole dataset, using the NCD values as the distances between the objects when forming the clusters.

We ran the NCD-based k-medians clustering by setting $K = 18$ (there are 18 different original themes and their variations). As the k-medians algorithm selects the initial cluster centroids randomly, we ran the evaluation 5 times. The results reported here are averaged over the evaluation runs.

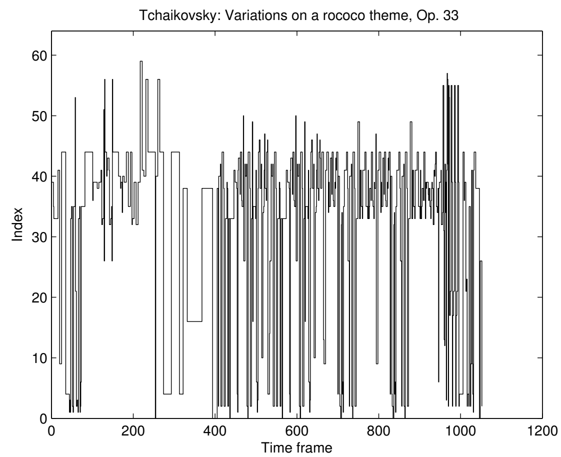

Figure 2. Sequence illustration of theme of *Variations on a rococo theme, Op. 33* by Tchaikovsky.

We did not expect the 18 different themes and their variations to group into clean clusters, but to evaluate the performance, we measured the number of different clusters (i.e. clusters that have centroids that are not original versions or variations of the centroids of the other clusters) and the number of correct clusterings (i.e. cases where the piece of music is clustered into a cluster with a centroid that is the original version or a variation of the piece). The number of different clusters was 11.8, and the number of correctly clustered compositions was 68. Thus, although not every theme and variations family of the dataset results in a single, separate cluster, a significant number of the compositions is still clustered correctly.

4.3 Discussion

Based on the results of the previous subsection, the proposed method does seem to have potential for measuring similarity between polyphonic, symbolic pieces of music. For comparison, the results reported for three different methods in [17] all have precision and recall values ranging from 0.4 to 0.5, with even a slightly smaller test data set (16 themes and 70 variations). The highest performance of our method is on a par with these results.

When using NCD, the most crucial choice is the data representation. Having fixed the representation, the next important choice to be made is to select an appropriate data-compression algorithm. Considering the idea that using a more efficient compressor algorithm yields a better approximation of Kolmogorov complexity, it would seem trivial to

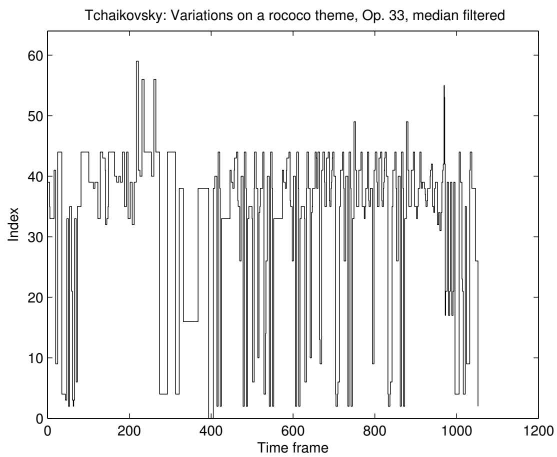

Figure 3. Sequence illustration of theme of *Variations on a rococo theme, Op. 33* by Tchaikovsky, with a median filtering of order 5.

Processing	none	OTI	MF	OTI & MF
Accuracy	0.536	0.250	0.291	0.190
Precision	0.632	0.231	0.409	0.165
Recall	0.474	0.152	0.393	0.193
F-measure	0.542	0.184	0.401	0.178

Table 2. Classification results with no additional processing applied, with OTI applied, with median filtering (MF) applied, and with both OTI and median filtering applied. All evaluations are conducted using PPMZ as compression algorithm.

use the most efficient compressor. However, there is no way of knowing how well the compression algorithm actually approximates Kolmogorov complexity, and thus, selecting only an efficient compressor does not necessarily guarantee that the NCD used (Equation 4) is actually a valid approximation of NID (Equation 3). As stated in [2], it is theoretically possible that when the compression gets more efficient, the NCD value disentangles from the NID value.

In our experiments, the more efficient PPMZ algorithm did eventually yield better results. We suppose that this happens due to the statistical nature of PPMZ, where the differences in file lengths is a lesser problem than with other compression approaches, and the effect of normalization in NCD is more likely to happen. Keeping in mind that the sequences we operate with are relatively short, the differences between longest and shortest files can potentially cause bias, as the longer x becomes, the better $C(x)$ approximates $K(x)$.

The key of the performances is an important factor when measuring similarity between pieces of music. In our evaluations the results are better when OTI is not calculated, as most of the variations are in the same key as the original theme. This is clearly a problem when considering to apply the method for other similarity measuring tasks. It is possible that the OTI algorithm, although very useful with audio-extracted chromagrams, might not be a suitable method when approximating the tonal similarity between two binary chromagrams extracted from MIDI data, and some other key-estimation algorithm could perform better for the task. It is also noteworthy that even though several variations are in different keys they are still classified correctly, possibly because our quantization method of the tonal centroid vectors maps several combinations of notes into the same characters, assuming they are in nearby keys.

The noise in the sequences, caused by transients of the time-slice chopping in the MIDI extraction method, may seem like an identification-distracting feature. Based on the results, however, it seems that the median filtering, although making the sequences smoother, does not provide better classification accuracy. This suggests that noise itself is not a hindrance as long as a suitable compression algorithm is used, and over-reducing the sequences loses important information that could be rather useful for distinguishing.

It should be noted that the variation database is somewhat unevenly distributed, with several themes having only a single variation included. The average precision of our system is slightly biased due to the high success rate of correctly classifying such variations, but the accuracy still supports that the method can be used for the selected task. Also, an interesting notion is that in some cases, there is confusion between different pieces of music by the same composer: The Goldberg Variations and English Suites by Johann Sebastian Bach are occasionally confused. This suggests that the NCD-based similarity measuring could be used for composer identification, as some stylistic information of the composer seems to be captured with the method.

5. CONCLUSIONS

We have presented a method for measuring similarity between symbolic, polyphonic pieces of music. Our method takes the MIDI data, extracts a binary chromagram out of it, maps the binary chromagram to tonal centroid representation and finally quantizes it, casting the original MIDI data into a sequence of characters comprising an alphabet of size 64. Then, the similarity between character sequences is measured using a compression-based similarity metric.

We experimented with both supervised and unsupervised machine learning with a dataset consisting of classical themes and their variations. The classification yielded results that are comparable with the state-of-the-art results with the same dataset. The clustering method used was a compression-based variant of k-medians, and using this novel method produced rather clean, well-structured clusters.

We intend to develop our approach to a more general similarity measuring technique suitable for different classification, clustering and identification tasks. In order to be more successful, several issues need to be reconsidered. One of the challenges is caused by the possible differences in keys between the pieces of music. Here, we solved the transposition problem by using OTI, which has been successfully used with audio data. However, applying OTI caused inferior results, and we need to either examine what could be done by using OTI with symbolic data, or select another method or representation to allow key-invariant similarity measuring.

6. ACKNOWLEDGEMENTS

The work of Teppo E. Ahonen was supported by Helsinki Graduate School in Computer Science and Engineering (Hecse). The authors also wish to thank David Rizo for his polyphonic variation collection.

7. REFERENCES

[1] Z. Cataltepe, Y. Tsuchihashi, and A. Sonmez. Music genre classification using midi and audio features. *EURASIP Journal on Advances in Signal Processing*, 24(1), 2007.

[2] R. Cilibrasi and P. Vitanyi. Clustering by compression. *IEEE Transactions on Information Theory*, 51(4):1523–1545, April 2005.

[3] M.J. Dovey. A technique for “regular expression” style searching in polyphonic music. In *Proc. ISMIR’01*, pages 179–185, 2001.

[4] C. Harte, M. Sandler, and M. Gasser. Detecting harmonic change in musical audio. In *Proc. 1st ACM Workshop on Audio and Music Computing Multimedia*, pages 21–26, 2006.

[5] K. Lemström. Towards more robust geometric content-based music retrieval. In *Proc. ISMIR’10*, pages 577–582, 2010.

[6] K. Lemström and M. Laitinen. Transposition and time-warp invariant geometric music retrieval algorithms. In *Proc. ADMIRE’11, Third International Workshop on Advances in Music Information Research*, 2011.

[7] K. Lemström and J. Tarhio. Transposition invariant pattern matching for multi-track strings. *Nordic Journal of Computing*, 10(3):185–205, 2003.

[8] M. Li and R. Sleep. Melody classification using a similarity metric based on Kolmogorov Complexity. *Sound and Music Computing*, 2004.

[9] A. Lubiw and L. Tanur. Pattern matching in polyphonic music as a weighted geometric translation problem. In *Proc. ISMIR’04*, pages 289–296, 2004.

[10] A. Marsden. Recognition of variations using automatic schenkerian reduction. In *Proc. ISMIR’10*, pages 501–506, 2010.

[11] B. Mokbel, A. Hasenfuss, and B. Hammer. Graph-based representation of symbolic musical data. In *Proc. 7th IAPR-TC-15 International Workshop on Graph-Based Representations in Pattern Recognition*, GbRPR ’09, pages 42–51, 2009.

[12] J. Pickens, J. Bello, G. Monti, M. Sandler, T. Crawford, M. Dove, and D. Byrd. Polyphonic score retrieval using polyphonic audio queries: A harmonic modeling approach. *Journal of New Music Research*, 32(2):223–236, 2003.

[13] C.A. Romming and E. Selfridge-Field. Algorithms for polyphonic music retrieval: The hausdorff metric and geometric hashing. In *Proc. ISMIR’07*, pages 457–462, 2007.

[14] J. Serra, E. Gómez, and P. Herrera. Transposing chroma representations to a common key. In *Proc. IEEE CS Conference on The Use of Symbols to Represent Music and Multimedia Objects*, pages 45–48, 2008.

[15] U. Simsekli. Automatic music genre classification using bass lines. In *Proc. 20th International Conference on Pattern Recognition*, pages 4137–4140, 2010.

[16] E. Ukkonen, K. Lemström, and V. Mäkinen. Geometric algorithms for transposition invariant content-based music retrieval. In *Proc. ISMIR’03*, pages 193–199, 2003.

[17] D. Rizo Valero. *Symbolic music comparison with tree data structures*. PhD thesis, Universidad de Alicante, August 2010.