A Framework for Analysis of Music Similarity Measures
Jensen, Jesper Højvang; Christensen, Mads Græsbøll; Jensen, Søren Holdt

Published in:
Proceedings of the 15th European Signal Processing Conference

Publication date:
2007

Document Version
Publisher's PDF, also known as Version of record

Link to publication from Aalborg University

Citation for published version (APA):

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

? Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
? You may not further distribute the material or use it for any profit-making activity or commercial gain
? You may freely distribute the URL identifying the publication in the public portal

Take down policy
If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from vbn.aau.dk on: januar 09, 2019
A FRAMEWORK FOR ANALYSIS OF MUSIC SIMILARITY MEASURES

Jesper Højvang Jensen, Mads Græsbøll Christensen, and Søren Holdt Jensen

Department of Electronic Systems
Aalborg University, Denmark
email: {jhj, mgc, shj}@es.aau.dk

ABSTRACT

To analyze specific properties of music similarity measures that the commonly used genre classification evaluation procedure does not reveal, we introduce a MIDI based test framework for music similarity measures. We introduce the framework by example and thus outline an experiment to analyze the dependency of a music similarity measure on the instrumentation of a song compared to the melody, and to analyze its sensitivity to transpositions.

Using the outlined experiment, we analyze music similarity measures from three software packages, namely Marsyas, MA toolbox and Intelligent Sound Processing toolbox. The tested timbral similarity measures perform instrument recognition relatively well, although they are sensitive to transpositions and differences between sound fonts. The beat/rhythm/melody similarity measures are not always capable of recognizing the same melody played with different instruments.

1. INTRODUCTION

As sound compression has matured and storage has become cheap, digital music collections, e.g. in the mp3 format, have grown very large. Navigation in such collections is limited by the metadata, such as title and artist, that is associated with the songs. Appropriate music similarity measures could help navigating in such collections, and could also be used for music recommendation systems in online music stores. By music similarity measure, we mean a quantitative measure of the similarity (or distance) between some musical aspects of two songs. Most music similarity measures are divided into a feature extraction part that extracts features that compactly describe some musical aspects of a song, and a distance measure that computes the distance between songs from the features. Much work has already been done on music similarity and on the related task of genre classification, e.g. [1–9]. Genre classification is often used to evaluate music similarity measures since it simplifies evaluation compared to the numerous user evaluations that are otherwise needed.

While genre classification provides a good first estimate of the performance of a music similarity measure, it does not provide details of its inner workings. For example, a commonly used feature that performs relatively well in genre classification when combined with a classifier is the mel-frequency cepstral coefficients (MFCCs), e.g. [3–8]. The MFCCs have their origins in speech recognition, where they are used to model the spectral envelope of a single speaker while suppressing the fundamental frequency. This is consistent with the common notion in music similarity of MFCCs as a timbral descriptor. Timbre is defined as "the auditory sensation in terms of which a listener can judge that two sounds with the same loudness and pitch are dissimilar" [10]. Since the temporal envelope and the spectral envelope play key roles to the perception of timbre [11], one would expect the MFCCs to mainly depend on the instrumentation of a song, and one would expect them to perform genre classification by matching songs with similar instrumentation. It is a tempting conclusion, but there are a number of uncertainties. For instance, in music several notes are often played at once, and it is not obvious how this mixing affects the spectral envelope. Furthermore, it is well-known that MFCCs are not completely independent of the fundamental frequency (e.g. [12]). Unfortunately, the good performance in genre classification does not reveal to which extent the MFCCs reflect the instrumentation, and to which extent they reflect the harmonies and key of a song. In this paper, we take the first steps towards an answer by introducing a test framework based on MIDI synthesis that supplements the genre classification results.

In Section 2, we describe an experiment to evaluate the dependency of a similarity measure on instrumentation compared to melody and an experiment to evaluate the sensitivity of a similarity measure to transpositions. In Section 3 and Section 4, we briefly describe some similarity measures we have evaluated using the experimental setups and present the results, respectively. In Section 5, we discuss the results as well as how the experiments can be modified to analyze other aspects of music similarity measures.

2. ANALYSIS FRAMEWORK

To analyze how a music similarity measure depends on the instrumentation compared to the notes and how it is affected by transpositions, we use a MIDI file1 setup. We take a number of MIDI files, manipulate the instrumentation and key using the MATLAB MIDI-toolbox [13], and then use a software synthesizer to generate waveform signals that can be used in a nearest neighbor classifier. Using MIDI files might bias the results, since the synthesized signal will be more homogeneous than recordings of real musicians would be. The advantage is that it allows us to manipulate instrumentation, tempo and melody in a flexible, reproducible way. In what follows, we first introduce an experiment to test the dependency of a musical similarity measure on instrumentation compared to the dependency on the notes, i.e., the melody and harmonies. Second, we introduce an experiment to eval-

This research was supported by the Intelligent Sound project, Danish Technical Research Council grant no. 26-04-0002, and the Parametric Audio Processing project, Danish Research Council for Technology and Production Sciences grant no. 274-06-0521.

1 A MIDI file contains information about fundamental frequency, instrumentation and onset/duration of all notes.
ute the dependency on transpositions of a song, i.e., how shifting all notes of a song a number of semitones affects the similarity measure.

2.1 Instrumentation versus notes

The dependency on instrumentation is tested by taking \( M \) different MIDI songs and choosing \( N \) instruments. We choose songs of different musical styles to ensure that the songs will have very different harmonic and melodic characteristics. For each \( m \), from 1 to \( M \), and for each \( n \), from 1 to \( N \), we do the following:

1. Read MIDI file \( m \).
2. Remove all percussive instruments.
3. Let all notes be played by instrument \( n \).
4. Synthesize a waveform signal.
5. Extract the feature vector \( v_{mn} \) from the waveform signal.

In the following, we will use the term “melody” to denote the instrument and key-independent part of a song, i.e., we will say that all songs created from MIDI file \( m \), which share the exact same melody, harmonies and timing, share the same melody no matter what the instrumentation or key is. After computing features for all \( M \times N \) songs, we find the nearest neighbor of \( v_{mn} \) according to the distance \( d(\cdot,\cdot) \) associated with the feature:

\[
(p,q) = \arg \min_{lk} d(v_{mn}, v_{lk})
\]

Let \( v_{np} \) be a given query, and let the nearest neighbor among the target songs be \( v_{np} \). If \( p = m \), then the nearest neighbor has the same melody as the query. We define the melody classification accuracy by the fraction of the \( M \times N \) queries where the nearest neighbor has the same melody. Similarly, we define the instrument classification accuracy as the fraction of queries where the nearest neighbor uses the same instrument.

2.2 Transpositions

A human listener does not consider the same song played in different keys as different songs. Similarly, an instrument playing two different notes is still considered the same instrument. For most similarity measures it is of interest to know how sensitive they are to transpositions. This is what this experiment investigates. It is similar to the previous experiment; the differences being that the tonal range is normalized and the song is transposed. The tonal range is normalized by transposing each individual track of a song (such as bass or melody) by an integer number of octaves, such that the average note being played in a track is as close as possible to the C4 note (middle C on the piano). The constraint of transposing tracks an integer number of octaves ensures that the harmonic relationships are not changed. As before, let \( m \) and \( n \) denote melody and instrument number, and let \( s \) denote the number of semitones a song is transposed. Features \( v_{mn}^{(s)} \) are computed for different values of \( s \). When evaluating (1), a query that has not been transposed is always used, i.e. the minimization is over \( d(v_{mn}^{(0)}, v_{lk}^{(s)}) \). Melody and instrument classification rates are computed for all values of \( s \).

2.3 Implementation of the framework

In this section, the tested similarity measures are described. Music similarity measures from three different publicly available software packages have been tested: Marsyas [16], the MA toolbox [17], and the Intelligent Sound Processing toolbox (see http://isound.kom.auc.dk/). Since not all of the similarity measures incorporate a distance measure between individual songs, some ad hoc distance measures have been introduced. These are also described below.

3. MUSIC SIMILARITY MEASURES

In this section, the tested similarity measures are described. Music similarity measures from three different publicly available software packages have been tested: Marsyas [16], the MA toolbox [17], and the Intelligentsound Processing toolbox (see http://isound.kom.auc.dk/). Since not all of the similarity measures incorporate a distance measure between individual songs, some ad hoc distance measures have been introduced. These are also described below.

3.1 Marsyas

From Marsyas v. 0.1, five feature sets are tested. The feature sets are thoroughly described in [3], where they were used with a probabilistic classifier that was trained on features from an entire genre. For this reason, a distance measure between feature vectors from individual songs does not exist. For all but the beat feature, which performed better with ordinary Euclidean distance, we therefore use the weighted Euclidean distance, \( d_{W}(u,v) = (u−v)^T W (u−v) \), where \( W \) is a diagonal matrix with positive elements. For the timbre features, \( W \) was chosen to maximize the difference between the average distance between all vectors and the average distance between vectors from songs with the same instrument, subject to \( \|W\|_F = 1 \), where \( \| \cdot \|_F \) is the Frobenius norm:

\[
W = \arg \max_{W} \frac{1}{M^2N^2} \sum_{ij} \sum_{kl} d_{W}(v_{ij}, v_{kl}) - \frac{1}{M^2N} \sum_{ij} \sum_{kl} d_{W}(v_{ij}, v_{kl})
\]

Before computing \( W \), all feature dimensions were normalized to have unit variance. For the pitch feature, the average distance between songs of the same melody was minimized instead. The weights \( W \) were computed from one of the 30 \times 30 subsets from the experiment in Section 2.1 where both query
Another approach to describe fluctuation pattern (FP)

Distance measure: Euclidean.

Concatenation of the timbre and cepstral coefficients (MFCCs) [3]. Distance measure: Weighted Euclidean.


Beat: Based on a histogram of prominent beats. Consists of the amplitudes and periods of the two first peaks in the histogram, the ratio between these two peaks, and the sum of all peaks [3]. Distance measure: Euclidean.

Pitch: Derived from a histogram of pitches in the signal. Contains among others periods and amplitudes of some of the most prominent peaks on both a full semitone scale and on an octave-independent (modulo 12) scale [3]. Distance measure: Weighted Euclidean.

3.2 MA toolbox

From the MA toolbox [17], five features were tested. The distance measures recommended in [17] are used.

MFCC: MFCCs are estimated in short windows, and a Gaussian mixture model (GMM) is trained to model them. Distance measure: Approximated, symmetrized Kullback-Leibler [7].

Sone: In a number of frequency bands distributed according to the Bark scale, the loudness measured in sone is computed. A GMM is trained on the loudness values. Distance measure: Approximated, symmetrized Kullback-Leibler [7].

Spectrum histogram (SH): A derivative of the raw sone features where the number of times each loudness level has been exceeded in each frequency band is counted [17]. Distance measure: Euclidean.

Periodicity histogram (PH): A description of periodic beats [17]. Distance measure: Euclidean.

Fluctuation pattern (FP): Another approach to describe periodicities in a signal. Distance measure: Euclidean.

3.3 Intelligent Sound Processing toolbox

Two similarity measures from the Intelligent Sound Processing (ISP) toolbox were tested:

MFCC: Similar to the MA toolbox MFCC, but with different parameters, such as the number of dimensions. Distance measure: Approximated, symmetrized Kullback-Leibler.

MAR: A multivariate autoregressive model that captures temporal correlation of MFCCs over 1 s segments [18]. A feature vector is produced for every 1 s of audio. Distance measure: For each vector in the query song, the other songs are compared to the vector. Distance measure: Approximated, symmetrized Kullback-Leibler.

The results of the two experiments are plotted in Figures 1 and 2. As is seen, some of the results are highly dependent on whether the query features are synthesized from the same sound font as the target features or not. However, the results are largely independent of which of the two sound fonts is used as query and which is used as target. Therefore, only results for Fluid 3 as both query and target, and results for Fluid 3 as query and SGM 180 as target are shown.

When query and target are from the same sound font, the timbral similarity measures perform well. The Marsyas Timbre+MFCC, the MA toolbox MFCC and Sone, and the ISP toolbox MFCC all have average instrument classification ratios in the range from 83% to 92%. The Marsyas MFCC, MA toolbox spectrum histogram and ISP toolbox MFCC-MAR also have relatively good performance, ranging from 75% to 79%. The Marsyas timbre performs worst of the timbral features with 55%. However, when query and target are from different sound fonts, the average instrument classification accuracy never exceeds 30%. Since the difference between the same instrument synthesized with different sound fonts is clearly audible, this is understandable, although still undesirable. According to [19], temporal characteristics such as attack transients contribute significantly to human perception of timbre. Timbre similarity measures that better incorporate this short-time temporal development might be less sensitive to the use of different sound fonts.

With respect to melody classification, three similarity measures are noteworthy: Marsyas beat, and MA toolbox periodicity histogram and fluctuation pattern with average classification accuracies of 51%, 78% and 62%, respectively. They are all practically independent of the combination of sound fonts used. The Marsyas pitch feature performs surprisingly bad, probably due to the inherently difficult problem of estimating multiple fundamental frequencies. Interestingly, the fluctuation pattern from the MA toolbox also performs better than random for instrument classification. Since in this laboratory setup neither the melody, rhythm, ac-

Figure 1: Instrument and melody classification accuracies. The number 1 denotes that the same sound font has been used for both query and target, while 2 denote that different sound fonts were used for the query and target. The whiskers on the plot denote the 95% confidence intervals.

1. Same sound font
2. Different sound fonts

1-Instrument
2-Melody

©2007 EURASIP
companiment nor timing changes, it ought to be possible to classify all melodies correctly. We therefore see much room for improvement.

The second experiment shows that all the timbral similarity measures behave similarly when exposed to transpositions. Accuracy is approximately halved when transposing 12 semitones (one octave). When transposing 24 semitones (two octaves), almost no instruments are recognized. An interesting detail is that the behavior of the MFCC feature from the MA toolbox is more similar to the sone feature from the same toolbox than to the MFCC feature from the ISP toolbox. The reason might be that the former two use the same statistical model. The features that performed well in melody recognition, namely MA toolbox periodicity histogram and fluctuation pattern and Marsyas beat, are all practically unaffected by transpositions. The Marsyas pitch feature is sensitive to transpositions, but since it contains information about the most dominant pitch, this is not surprising.

5. DISCUSSION

From the experiments we observed that the timbral similarity measures did not generalize well to different sound fonts. We therefore hypothesize that timbral similarity measures that also rely on the temporal envelope will better reflect the human sound perception where certain smooth spectral changes, such as adjusting the bass and treble, do not significantly alter the perception of timbre. We also observed that there is room for improvements with melody recognition.

These results could not have not been obtained from a genre classification experiment alone. By using MIDI files, we have effectively separated the effect of instrumentation and melody, and a signal modification that would have been difficult or cumbersome to introduce directly in a waveform signal, namely transposition, has been introduced.

Although in this paper we have only tested the sensitivity of similarity measures to transpositions, it would also be relevant to measure the dependency on tempo, combinations of instruments, bandwidth and audio compression. We strongly recommend the use of such tests as a simple, yet insightful supplement to genre classification.

REFERENCES


