Music Identification with Persistent Homology

Ekaterina Ivshina

November 15, 2020

1 Objectives

In this paper, we aim to find a representation of songs as a time series of topological fingerprints, with a metric to compare pairs of time-varying shapes. We then develop and test an algorithm that can account for noise distortions of input audio clips to identify music.

2 Theory

2.1 Homology

In this section, we review the basics of homology theory.

Definition 2.1. A chain complex C is a sequence of finite-dimensional vector spaces C_k over field $\mathbb{F}$ with linear maps $\partial_k : C_k \to C_{k-1}$, called boundary operators, such that $\partial_k \circ \partial_{k+1} = 0$. The elements of C_k are called chains.
Thus we have a chain complex: $0 \xrightarrow{\partial_{m+1}} C_m \xrightarrow{\partial_m} C_{m-1} \xrightarrow{\partial_{m-1}} ... \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0 \xrightarrow{\partial_0} 0$ For the purposes of this paper, we use the field $\mathbb{F} = \mathbb{R}$.

Definition 2.2. Given a chain complex C, the k^{th} homology group is the quotient $H_k(C) = ker(\partial_k)/Im(\partial_{k+1})$. The elements of $ker(\partial_k)$ are called cycles, and the elements of $Im(\partial_{k+1})$ are called boundaries. The homology groups consist of cycles modulo boundaries.

We know that two finite-dimensional vector spaces are isomorphic if and only if they have the same dimension. Hence dimension is an invariant for $H_k(C)$. To describe the homology of C, we define the *Betti numbers* as $\beta_k = \dim H_k(C)$. For interpretations, we can think of indices k as the dimension of objects from which we determine H_k, β_k, etc.

Example 2.1. For a concrete example, let's consider a surface with triangulation T and a chain complex $C(T)$: $0 \xrightarrow{\partial_3} C_2 \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0 \xrightarrow{\partial_0} 0$. $C_0(T)$ is the set of all linear combinations of the vertices of T. Because $\text{Im}\partial_0 = \{\bar{0}\}$, ∂_0 is the zero map.

Next, assign each edge of T an orientation; the orientation on each simplex is to be induced from an enumeration of the vertices. $C_1(T)$ is the set of all linear combinations of these edges of T. Given an oriented edge e that starts from vertex a and ends with vertex b, define $\partial_1(e) = b - a$. Then ∂_1 is defined on all of $C_1(T)$. Note that $\partial_0 \circ \partial_1 = 0$, as required. Let $C_2(T)$ be spanned by the oriented 2-simplices (faces). Because ∂_3 has $\{\bar{0}\}$ as its domain, ∂_3 is the zero map. Now, by analogy, given a 2-simplex σ, we define $\partial_2(\sigma)$ as a linear combination of edges e_i that form the boundary of σ. The sign of each edge is determined based on the relative orientation of σ and e_i. For example, for the 2-simplex in Figure 1, we have $\partial_2(\sigma) = e_1 + e_2 - e_3$. We can also check that $\partial_1 \circ \partial_2 = 0$. For the case below, $\partial_1(\partial_2(\sigma)) = (b - a) + (c - b) - (c - a) = 0$.

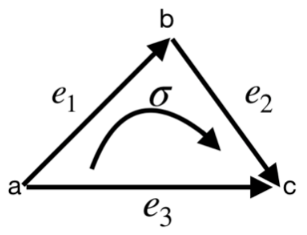

Figure 1: Example 2-simplex.

When considering manifolds or simplicial complexes of higher dimensions, the orientation of objects should be defined in terms of even/odd permutations with respect to the order on vertices.
Note that a chain lies in $\ker\partial_1$ if and only if it closes up (e.g. $e_1 + e_2 - e_3$). We can think of T as an electric circuit, then $ker\partial_1$ consists of all closed loops in the circuit.

Proposition 2.1. Euler characteristic is the alternating sum of the Betti numbers.

Proof. Let's now see the connection between the Euler characteristic and Betti numbers in our example. Notice that since $H_k(C) = \ker(\partial_k)/\text{Im}(\partial_{k+1})$, we have $\dim H_k(C) = \dim ker(\partial_k) - \dim Im(\partial_{k+1})$. By the rank-nullity theorem, $\dim C_k = \dim Im(\partial_k) + \dim ker(\partial_k)$. In particular, $\dim C_0 = \dim ker(\partial_0)$. Notice that in our example, $\dim C_0$ is the number of vertices, $\dim C_1$ is the number of edges, $\dim C_2$ is the number of faces. Thus, the Euler characteristic is $\chi(T) = F - E + V = dimC_2 - \dim C_1 + \dim C_0 = (\dim ker(\partial_2) + \dim Im(\partial_2)) - (\dim ker(\partial_1) + \dim Im(\partial_1)) + \dim ker(\partial_0) = (\dim ker(\partial_2) - \dim Im(\partial_3)) - (\dim ker(\partial_1) - \dim Im(\partial_2)) + (\dim ker(\partial_0) - \dim Im(\partial_1)) = \beta_2 - \beta_1 + \beta_0$.

□

In general, β_0 is the number of connected components. Any connected closed orientable surface has $\beta_2 = 1$, and any non-orientable surface has $\beta_2 = 0$. For an orientable surface, β_1 is twice the genus. For a non-orientable surface, β_1 is the genus minus one (Figure 2).

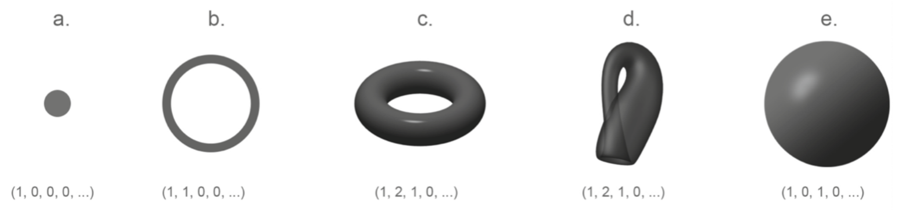

Figure 2: Betti number provide signature of the underlying topology (shape).

2.2 Persistent homology

We can now discuss persistent homology - a way of computing topological features of data at different scales [1]. In short, the idea is to build Vietoris–Rips simplicial complexes $R_\chi(\epsilon) := \{\sigma \subset \chi : d(x_i, x_j) < 2\epsilon, \forall x_i, x_j \in \sigma\}$ from the input data using different ϵ values and calculate Betti numbers for each simplicial complex. Thus, persistent homology is a way of computing Betti numbers of a topological space at different spatial resolutions.

Next, a filtration F is a collection of subspaces approximating the data points at different resolutions.

Definition 2.3. A *filtration* $F = \{F_a\}_{a\in\mathbb{Z}}$ is a collection of sets with $a \leq b$ implying $F_a \subset F_b$.

For a filtration F and for each $k \in \mathbb{N}_0 = \mathbb{N} \cup \{0\}$, the associated persistent homology $PH_K F$ is a collection of k-th dimensional homology of each subset in F.

Definition 2.4. Let F be a filtration and let $k \in \mathbb{N}_0$. The associated k-th *persistent homology* $PH_K F$ is a collection of vector spaces $\{H_k F_a\}_{a\in\mathbb{Z}}$ equipped with homomorphisms $\{\iota_k^{a,b}\}_{a\leq b}$, where $H_k F_a$ is the k-th dimensional homology of F_a and $\iota_k^{a,b} : H_k F_a \to H_k F_b$ is the homomorphism induced by the inclusion $F_a \subset F_b$.

For the k-th persistent homology $PH_K F$, the set of filtration levels at which a specific homology class appears is always an interval $[b, d) \subset [-\infty, \infty]$, i.e. a specific homology class is formed at some filtration value b and dies when the inside hole is filled at another value $d > b$ [2].

Definition 2.5. Let F be a filtration and let $k \in \mathbb{N}_0$. The corresponding k-th persistence diagram $Dgm_k(F)$ is a finite multiset of $(\mathbb{R} \cup \{\infty\})^2$, consisting of all pairs (b, d) where $[b, d)$ is the interval of filtration values for which a specific homology class appears in $PH_K F$. b is called a birth time and d is called a death time.

2.3 Bottleneck distance

We can now define a metric on the space of persistent diagrams.

Definition 2.6. The Wasserstein distance between two persistent diagrams X and Y is given by

$$W_p[L_q](X,Y) = \inf_{\phi : X \to Y} [\sum_{x \in X} ||x - \phi(x)||_q]^{\frac{1}{p}}$$

where $1 \leq p, q \leq \infty$ and ϕ ranges over bijections between X and Y.
For $p = \infty$,

$$W_\infty[L_q](X,Y) = \inf_{\phi : X \to Y} \sup_{x \in X} ||x - \phi(x)||_q,$$

and we call $W_\infty[L_\infty]$ the *bottleneck distance.*

3 Algorithm

3.1 Extracting signatures

Task: for a given input music clip, find its closest match in the database and return the name of the song. Thus, we first need to find suitable representation of musical data, extract signatures, then create a database of the signatures with labels - names of the songs, and finally apply the search algorithm to new noisy samples.
We will start by discussing the way we extract signatures from music clips. The first step is to produce constant-Q chromograms from 2-seconds clips of the songs. To produce such chromograms, we apply constant-Q transform, which transforms a time series to the frequency domain. This transformation is related to the Fourier transform and is said to be well suited for musical data, as it outputs amplitude against log frequency. The constant-Q transform of $x[n]$ is defined as follows:

$$X[k] = \frac{1}{N[k]} \sum_{n=0}^{N[k]-1} W[k,n]x[n]e^{\frac{-j2\pi Qn}{N[k]}}$$

where $W[k,n] = \alpha - (1-\alpha)cos\frac{2\pi n}{N[k]-1}$, $\alpha = \frac{25}{46}, 0 \leq n \leq N[k]-1$, and $N[k] = Q\frac{f_s}{f_k}$. f_s/f_k is equal to the number of samples that are processed per cycle at frequency f_k, $Q = \frac{f_k}{\sigma f_k}$; σf_k is the filter width, and f_s is the sample rate. Example constant-Q chromogram can be found in Figure 3.

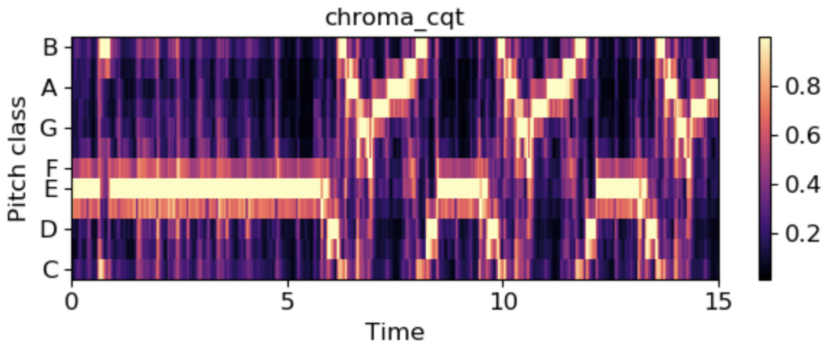

Figure 3: Example of a constant-Q chromogram.

3.2 Deforming Tonnetz

Next, we project the chromogram onto Tonnetz and deform it by defining a height function. In music theory, Tonnetz is a lattice diagram that represents pitch space, allowing to capture certain harmonic relationships in musical data (Figure 4). It was first described by Leonhard Euler in 1739. When studying harmony, Leonard Euler put notes on a torus such that in the horizontal direction, notes are separated by perfect 5th, in a diagonal direction (from left to right), notes are separated by major 3rd, and in another diagonal direction (from right to left), notes are separated by minor 3rd.

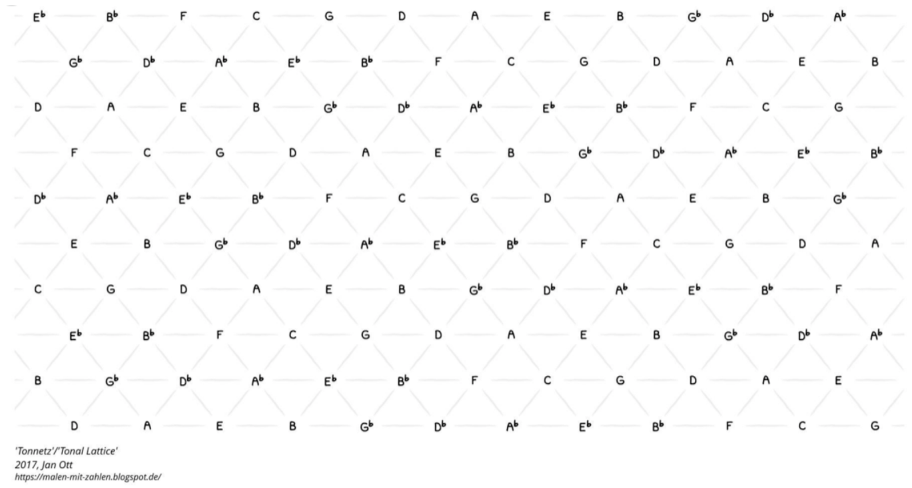
E♭ B♭ F C G D A E B G♭ D♭ A♭
G♭ D♭ A♭ E♭ B♭ F C G D A E B
D A E B G♭ D♭ A♭ E♭ B♭ F C G
F C G D A E B G♭ D♭ A♭ E♭ B♭
D♭ A♭ E♭ B♭ F C G D A E B G♭
E B G♭ D♭ A♭ E♭ B♭ F C G D A
C G D A E B G♭ D♭ A♭ E♭ B♭ F
E♭ B♭ F C G D A E B G♭ D♭ A♭
B G♭ D♭ A♭ E♭ B♭ F C G D A E
D A E B G♭ D♭ A♭ E♭ B♭ F C G
'Tonnetz'/'Tonal Lattice'
2017, Jan Ott
https://malen-mit-zahlen.blogspot.de/

Figure 4: Tonnetz based on Neo-Riemann theory.

The set of pitch classes is $\mathbb{R}/12\mathbb{Z}$ (i.e. we have 12 semitones in total, and we identify notes that are an octave apart). We place each pitch class onto the Tonnetz as a vertex. We then define a height function $h : V \to \mathbb{R}$ on the set of vertices of the space $\mathbb{R}/12\mathbb{Z}$ (each vertex is a pitch class) by associating to each vertex a height that corresponds to the amplitude of a given pitch class in the music clip. After that, the 2-dimensional Tonnetz is deformed, having the values of the height function as its third dimension. Figure 5 is an example of the Tonnetz deformed by 2-seconds clip of blues.

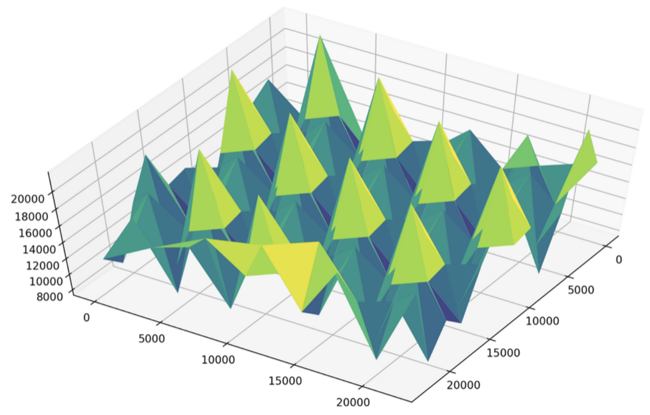

Figure 5: Deformed Tonnetz (2 seconds of blues).

3.3 Persistent diagrams

After finding a way to represent musical data as a topological object, we then apply persistent homology to the point cloud of the deformed Tonnetz to produce persistent diagrams (Figure 6).

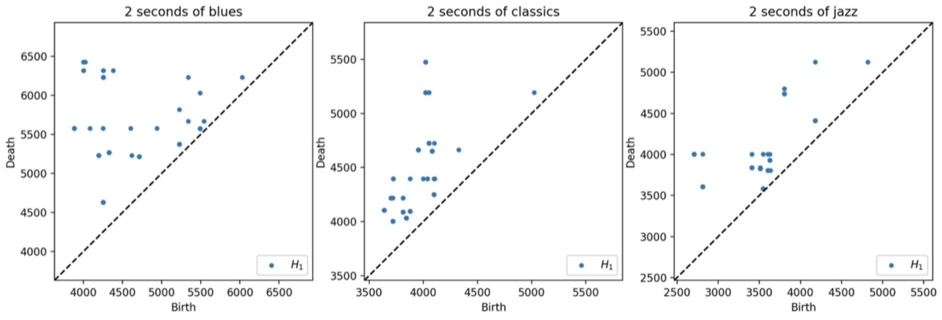

Figure 6: Persistent diagrams of three 2-seconds clips.

In general, persistent homology is useful because, for example, changing the speed of a recording doesn't alter persistent diagrams that are based on lower-star filtration that much (so we can identify DJ remixes of a song). Figure 7 illustrates this fact. Given a simplicial complex K and a real-valued function f defined on its vertices, define $K_a = \{\sigma \in K | max_{v \in \sigma} f(v) \leq a\}$ to be the lower-star filtration.

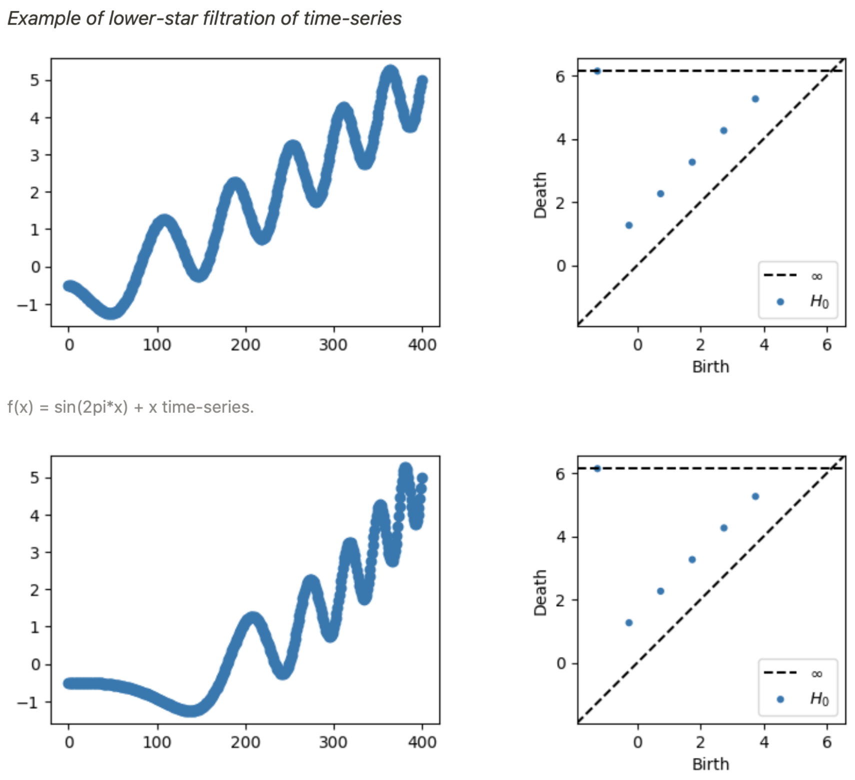

Figure 7: Reparameterizing time-series does not change the original persistent diagram.

3.4 Matching

After producing persistent diagrams, we calculate the bottleneck distances between the persistent diagrams to find the closest match of a given song. Under mild assumptions on the function, the authors of [3] showed that the persistence diagram of a function on a topological space is stable. Persistent homology is thus claimed to be stable under small changes in the input filtration (so that such changes lead to small perturbations in the bottleneck distance).

4 Results

We compiled a database of 50 songs from 10 genres, with 5 songs per genre. As input to the algorithm, we use .WAV files containing 1, 2, 3, 4, 5-seconds clips of songs. To test the algorithm, we added noise from a normal distribution so that the song clips have SNR = 10, 20, 30. We then ran the algorithm to identify music within each genre. The accuracy as a function of clip duration for different SNR is reported in Figure 8. From [4], it follows that Shazam computes about 50,000-250,000 fingerprints per song. We compute 1 persistent diagram per clip, so we produce about 10,000-20,000 signatures per song.

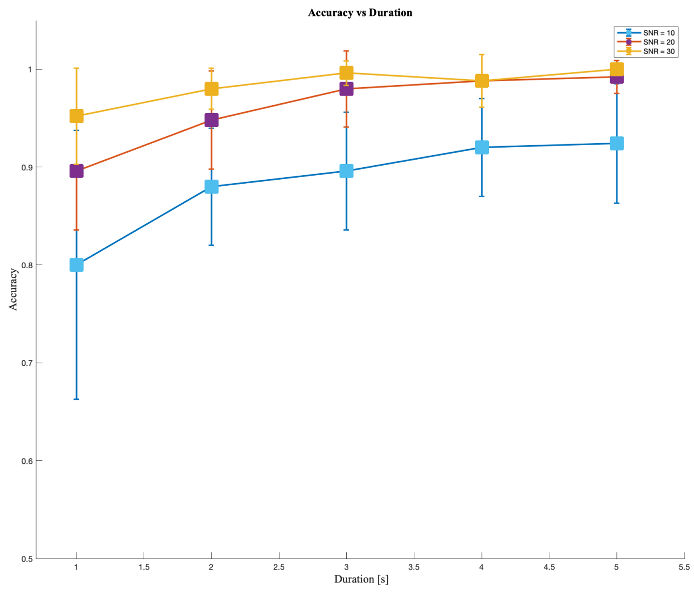

Figure 8: Accuracy as a function of clip duration for SNR = 10, 20, 30. The errorbars represent the standard deviation.

5 Possible improvements

The height function we defined above is the simplest one. We could also use discrete Gaussian curvature as our height function to reflect different geometries of the Tonnetz. Furthermore, we can capture additional harmonic information using a consonance function as our height function, which would give information on tensions/resolutions of chords in a clip. Next, while in this paper we only utilized the vertical structure of music (i.e. pitch classes), it is important to consider the horizontal structure as well, i.e. chords' time progression, tempo, etc. (after all, it is important in what order we hear the chords/notes). Additionally, the rhythmic part of musical pieces is important, and we could capture this additional information using audio novelty function.

References

[1] Herbert Edelsbrunner and John Harer. *Computational Topology: An Introduction.* 01 2010.

[2] Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete and Computational Geometry*, 33:249–274, 02 2005.

[3] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete Computational Geometry*, 37:103–120, 2007.

[4] Avery Wang. An industrial strength audio search algorithm. 01 2003.