이학석사 학위논문

Time series analysis using the persistent homology and sliding window method

(퍼시스턴트 호몰로지와 슬라이딩 윈도우 기법을 이용한 시계열 해석)

2020년 2월

서울대학교 대학원

수리과학부

최 경 제

Time series analysis using the persistent homology and sliding window method

지도교수 Otto van Koert

이 논문을 이학석사 학위논문으로 제출함

2019년 12월

서울대학교 대학원

수리과학부

최경제

최경제의 이학석사 학위논문을 인준함

2020년 2월

위 원 장 국 웅 (인)

부 위 원 장 Otto van Koert (인)

위 원 임 선 희 (인)

Time series analysis using the persistent homology and sliding window method

by

KyungJea Choi

A DISSERTATION

Submitted to the faculty of the Graduate School
in partial fulfillment of the requirements
for the degree Master of Science
in the Department of Mathematics
Seoul National University
February 2020

Abstract

We consider in this paper time series analysis using the persistent homology and sliding window method. Specifically, we look into the 1-dimensional max persistences of trigonometric functions with the sliding window method. And from this max persistence, we can think relations between values of sliding window (embedding dimension, step) which are at 1-dimensional max persistence and frequencies of the functions. Furthermore, by this process and some Fourier Analysis, We can estimate frequencies of functions from time series sample data.

Keywords : time series analysis, persistent homology, sliding window, max persistence, trigonometric function, frequency
Student number : 2017-29039

Contents

1 Introduction

Time series are often used in various fields and Time series analysis means a variety of methods for analyzing time series data in order to extract meaningful statistics and other properties of the data.
Meanwhile, Persistent homology is an algebraic method for computing topological features of space at different spatial resolutions. More persistent features are detected over a wide range of spatial scales and are deemed more likely to artifacts of sampling, noise, or the particular choice of parameters.
In particular, 1-dimensional persistence can be used at the Time series analysis and we look into some time series cases with the method which uses 1-dimensional persistence and Sliding Window (also known as time-delay reconstruction). Through this method, we can know (1) when 1-dimensional 'max' persistence occurs, and (2) then Point cloud figures by the Sliding Window are roundest (of course, we only see this in Dim = 2, 3) and (3) we can think some relation between the embedding dimension, Time-delay, and frequency of some trigonometric functions at that time.
So, through this method, we can practically find frequencies of time series from the given discrete sample data.

2 Preliminaries

This chapter introduces some definitions and theorems which are required in the next chapters.

2.1 TDA and Persistent homology

Definition 2.1. Let S be a simplicial complex and F be a field. Then a **simplicial k-chain** is a finite formal sum $\sum_{i=1}^{N} c_i\sigma_i$, where each c_i is an element of F and σ_i is an oriented k-simplex and the group of k-chains on S is denoted by $C_k(S)$.

By the above definition, all oriented k-simplices in S compose the basis of $C_k(S)$ and $C_K(S)$ is an F-vector space which has a basis in one-to-one correspondence with the set of k-simplices in S. To define a basis explicitly, we have to choose an orientation of each simplex. One standard way to do is to choose an ordering of all the vertices and give each simplex the orientation corresponding to the induced ordering of its vertices.

Definition 2.2. Let $\sigma = (v_0, ..., v_k)$ be an oriented k-simplex, viewed as a basis element of C_k. The **boundary operator** $\partial_k : C_k \to C_{k-1}$ is the homomorphism defined by $\partial_k(\sigma) := \sum_{i=1}^{k}(-1)^i(v_0, ..., \widehat{v_i}, ..., v_k)$, where the oriented simplex $(v_0, ..., \widehat{v_i}, ..., v_k)$ is the i-th face of σ obtained by deleting its i-th vertex.

Definition 2.3. Let ∂_k, ∂_{k+1} be the boundary operations as above defined. Then elements of the subgroups $Z_k := ker\partial_k$ and $B_k := im\partial_{k+1}$ of C_k are called **cycles** and **boundaries**, respectively and k-th **homology group** H_k

of S is defined to be $H_k(S) := Z_k/B_k$. Furthermore, the rank of the k-th homology group, the number $\beta_k := dim(H_k(S))$ is called the k-th **Betti number** of S.

As the above definition, we know (1) dim$H_k(S)$ = dim$Z_k(S)$ - dim$B_k(S)$. So we can find dim$H_k(S)$ by computing the null space and range of the boundary operator. (2) Generally speaking, this concept, 'homology' of S, is the measurement of those cycles that cannot be filled in by boundaries. (3) dim$H_0(S)$ means the number of components of S. & dim$H_1(S)$ means the number of holes of S. & dim$H_2(S)$ means the number of voids (or cavities), i.e. 2-dimensional holes, of S. So, in summary, for arbitrary k, dim$H_k(S)$ means the number of k-dimensional holes (or rooms) of S.

Definition 2.4. Let K be a simplicial complex. A **subcomplex** of K is a subset of its simplices that is closed under the face relation. A **filtration** of K is a nested sequence of subcomplexes that starts with the empty complex and ends with the complete complex, $\emptyset = K_0 \subset K_1 \subset ... \subset K_m = K$. A homology class α is **born** at K_i if it is not in the image of the map induced by the inclusion $K_{i-1} \subset K_i$. If α is born at K_i, we say that it **dies entering** K_j if the image of the map induced by $K_{i`1} \subset K_{j-1}$ does not contain the image of α but the image of the map induced by $K_{i-1}K_j$ does. The **persistence** of α is j - i.

Definition 2.5. Let (M, d) be a metric space. Then the **Vietoris-Rips complex** of (M, d) at the parameter $\epsilon > 0$ is the simplicial complex VR_ϵ(M) satisfying (1) Its vertices are the points in M. (2) For some distinct points $x_0, ..., x_k$ in M, if $d(x_i, x_j) < \epsilon$ for each 0 $\leq$ i, j $\leq$ k, then $x_0, ..., x_k$ spans a k-simplex.

Definition 2.6. (1) Let K be a simplicial complex. Then the **Persistence diagram** $dgm(n)(K)$ of the homology filtration induced from the Vietoris-Rips filtration which is made by the Vietoris-Rips complex of K is the multiset

of points (i, j) in the upper half-plane for all n-th homology class that is born at K_i and dies entering K_j along with infinitely many copies of the points on the diagonal $\triangle := \{(x, x) : x \geq 0\}$.
We usually write dgm instead of $dgm(n)(K)$ because n and K are clear in context.

(2) Let (x,y) $\in$ dgm. Then we define **pers**(x,y) := y - x for (x,y) $\in \mathbb{R}^2$, and as ∞ otherwise. And we define **maximum persistence** mp(dgm) of dgm by mp(dgm) := $\max_{(x,y)\in dgm}$ pers(x,y).

Definition 2.7. A **persistence module** $\mathcal{F}_A$ is a family $\{F_\alpha\}_{\alpha\in A}$ of vector spaces indexed by $A \subset \mathbb{R}$, together with a family of homomorphisms $\{f_\alpha^\beta : F_\alpha \to F_\alpha\}_{\alpha\leq\beta}$ satisfying $f_\alpha^\gamma = f_\beta^\gamma \circ f_\alpha^\beta$ and $f_\alpha^\alpha = id_{F_\alpha}$ for every $\alpha, \beta, \gamma \in A$.

We want to decompose a persistence module $\mathcal{F}_A$ as a direct sum of indecomposable persistence modules, say $\mathcal{F}_A \cong \bigoplus_{j\in J} I_j$, where each indecomposable summand I_j is an **interval module** $I[\alpha, \beta]$ over a field k, defined as follows

$$\underbrace{0 \xrightarrow{0} \cdots \xrightarrow{0} 0}_{i<\alpha} \xrightarrow{0} \underbrace{\boldsymbol{k} \xrightarrow{\mathbb{1}} \cdots \xrightarrow{\mathbb{1}} \boldsymbol{k}}_{[\alpha\,,\,\beta\,]} \xrightarrow{0} \underbrace{0 \xrightarrow{0} \cdots \xrightarrow{0} 0}_{i>\beta}\,.$$

Theorem 2.8. ***(Interval Decomposition Theorem)*** *[10]*
Every persistence module whose index set is finite or whose vector spaces are finite-dimensional decomposes uniquely as a direct sum of interval modules.

At that time, the decomposition is independent of the choice of base field k.

2.2 Time series and Fourier analysis

Definition 2.9. Let f be a function defined on the real numbers and choose a natural number m and a real number τ. Then a **sliding window embedding** of f based at t is the point

$$SW_{m,\tau}f(t) := \begin{pmatrix} f(t) \\ f(t+\tau) \\ \vdots \\ f(t+(m-1)\tau) \end{pmatrix}.$$

A collection of these points is called a **sliding window point cloud** for f and the value $(m-1)\tau$ is the **window size**.

This definition is motivated by the following theorem.

Theorem 2.10. ***(Takens' Embedding Theorem)*** *[8]*
Let M be a compact manifold of dimension k, $\phi : M \to M$ a smooth diffeomorphism, $f : M \to \mathbb{R}$ a smooth function. Then if $m > 2k$, then it is a generic property that the delay observation map $D_{\phi,f} : M \to \mathbb{R}^m$ defined by $D_{\phi,f}(x) = (f(x), f(\phi(x)), ..., f(\phi^{m-1}(x)))$ is an embedding .

The next thing is about the Fourier Analysis which is essentially needed in our examinations.

Remark 2.11. When we find the frequencies or some information from the sampling data set, we don't know the correct function surely and even the number of sampling data set is usually finite. So here, we need to consider approximating a function f by its Fourier polynomials. Let $f(t) = S_N f(t) + R_N f(t)$, where $S_N f(t) = \sum_{n=0}^{N} a_n cos(nt) + b_n sin(nt) = \sum_{n=-N}^{N} \hat{f}(n) e^{int}$ is the N-truncated Fourier series expansion of f, $R_N f$ is the remainder, $\hat{f}(n) = \frac{1}{2}a_n - \frac{i}{2}b_n$ (n > 0), $\frac{1}{2}a_{-n} + \frac{i}{2}b_{-n}$ (n < 0), or a_0 (n $= 0$). The next theorems guarantees the justification of our examinations.

Theorem 2.12. *Let C(X, Y) be the set of continuous functions from X to Y with the sup norm and* $\mathbb{T} = \mathbb{R}/2\pi\mathbb{Z}$. *Then for every* $m \in \mathbb{N}$ & $\tau > 0$, $SW_{m,\tau} : C(\mathbb{T}, \mathbb{R}) \to C(\mathbb{T}, \mathbb{R}^m)$ *is a bounded linear operator with* $\| SW_{m,\tau} \| \leq \sqrt{m}$.

Proof. We easily know that $SW_{m,\tau}$ is linear from the definition. To show its boundedness, letting $f \in C(\mathbb{T}, \mathbb{R})$ and $t \in \mathbb{T}$, $\| SW_{m,\tau} f(t) \|_{\mathbb{R}^m}^2 = |f(t)|^2 + |f(t+\tau)|^2 + ... + |f(t+(m-1)\tau)|^2 \leq m\|f\|_\infty^2$. □

Theorem 2.13. *Let* $k \in \mathbb{N}$ *and* $\mathbb{T} = \mathbb{R}/2\pi\mathbb{Z}$. *If* $f \in C^k(\mathbb{T}, \mathbb{R})$, *then* $\| SW_{m,\tau} R_N f(t) \|_{\mathbb{R}^m}$ $= \| SW_{m,\tau} f(t) - SW_{m,\tau} S_N f(t) \|_{\mathbb{R}^m} \leq \sqrt{4k-2} \| R_N f^{(k)} \|_2 \frac{\sqrt{m}}{(N+1)^{k-\frac{1}{2}}}$.

Proof. Let $k \in \mathbb{N}$ and $f \in C^k(\mathbb{T}, \mathbb{R})$. Then the integration by parts gives the following identity $|\widehat{f^{(k)}}(n)| = |n|^k |\widehat{f}(n)|$ for the length of $\widehat{f^{(k)}}(n)$, the n-th complex Fourier coefficient of $f^{(k)}$, $n \in \mathbb{Z}$. Therefore for every $t \in \mathbb{T}$, the Cauchy-Schwartz inequality, Young's inequality and Parseval's theorem together imply that

$$|R_N f(t)| \leq \sum_{n=N+1}^{\infty} \frac{|\widehat{f^{(k)}}(n)| + |\widehat{f^{(k)}}(-n)|}{n^k}$$
$$\leq \Big(\sum_{n=N+1}^{\infty} (|\widehat{f^{(k)}}(n)| + |\widehat{f^{(k)}}(-n)|)^2 \Big)^{1/2} \Big(\sum_{n=N+1}^{\infty} \frac{1}{n^{2k}} \Big)^{1/2}$$
$$\leq \Big(2 \sum_{|n| \geq N+1} |\widehat{f^{(k)}}(n)|^2 \Big)^{1/2} \Big(\int_{N+1}^{\infty} \frac{1}{x^{2k}} dx \Big)^{1/2} = \sqrt{2} \| R_N f^{(k)} \|_2 \frac{\sqrt{2k-1}}{(N+1)^{k-\frac{1}{2}}}.$$
$$= \| R_N f^{(k)} \|_2 \frac{\sqrt{4k-2}}{(N+1)^{k-\frac{1}{2}}}.$$

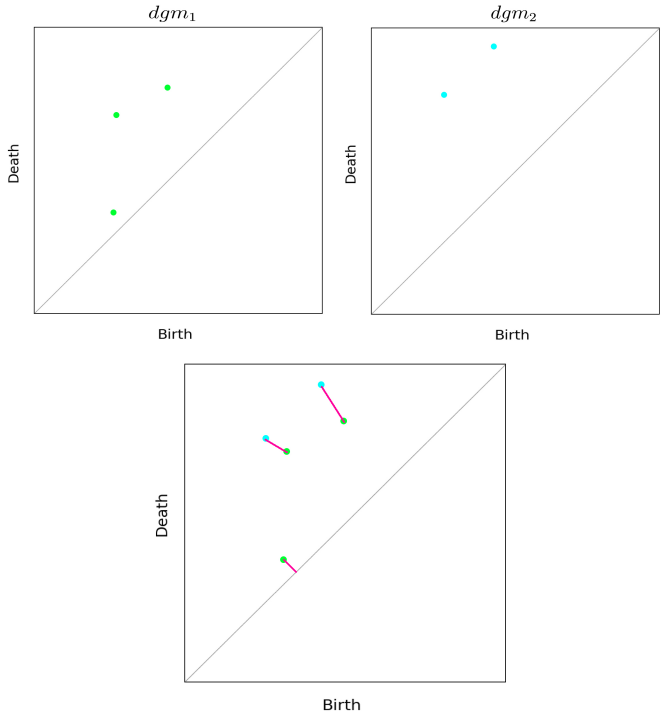
and thus by Theorem 2.9.

$$\| SW_{m,\tau} R_N f(t) \|_{\mathbb{R}^m} \leq \sqrt{m} \| R_N f \|_\infty \leq \sqrt{4k-2} \| R_N f^{(k)} \|_2 \frac{\sqrt{m}}{(N+1)^{k-\frac{1}{2}}}.$$ □

Definition 2.14. (1) Let X and Y be two nonempty subsets of a metric space (M, d). Then the **Hausdorff distance** between X and Y is defined by $d_H(X, Y) := \max \{ \sup_{x \in X} \inf_{y \in Y} d(x,y), \sup_{y \in Y} \inf_{x \in X} d(x,y) \}$.

(2) Let dgm_1, dgm_2 be persistence diagrams. Then the **Bottleneck distance** between dgm_1 and dgm_2 is defined by $d_B(dgm_1, dgm_2) := \inf_f \sup_{x \in dgm_1} \|x - f(x)\|_\infty$ for any bijection f : $dgm_1 \to dgm_2$.

In (2) of **Definition 2.12.**, such bijection f always exists because we put infinitely many points on the diagonal in **Definition 2.6.**. Our definition of the bottleneck distance is a somewhat special version, not usual but it's sufficient for our cases. For example, let dgm_1, dgm_2 be persistence diagrams such that the number of not-diagonal points of dgm_1 and dgm_2 is different, say 3 and 2, respectively. Then the Bottleneck distance between dgm_1 and dgm_2 can be measured as the following pictures.



The definition of Hausdorff distance is equivalent to $d_H(X, Y) = \inf\{\epsilon \geq 0 : X \subset Y^\epsilon \ \& \ Y \subset X^\epsilon\}$, where $X^\epsilon := \bigcup_{x \in X} \{a \in M : d(a, x) \leq \epsilon\}$.

We know $d_B(dgm(X), dgm(Y)) \leq 2d_H(X, Y)$. **(Stability Theorem)** [4]

Theorem 2.15. ***(Approximation)***

Let $\mathbb{T} = \mathbb{R}/2\pi\mathbb{Z}$, $T \subset \mathbb{T}$, $f \in C^k(\mathbb{T}, \mathbb{R})$, $X = SW_{M,\tau}f(T)$ *and* $Y = SW_{M,\tau}S_Nf(T)$. *Then*

(1) $d_H(X,Y) \leq \sqrt{4k-2} \parallel R_N f^{(k)} \parallel_2 \frac{\sqrt{m}}{(N+1)^{k-\frac{1}{2}}}$.

(2) $|mp(dgm(X)) - mp(dgm(Y))| \leq 2d_B(dgm(X), dgm(Y))$.

(3) $d_B(dgm(X), dgm(Y)) \leq 2\sqrt{4k-2} \parallel R_N f^{(k)} \parallel_2 \frac{\sqrt{m}}{(N+1)^{k-\frac{1}{2}}}$.

Proof. (1) Let $\epsilon > \sqrt{4k-2} \parallel R_N f^{(k)} \parallel_2 \frac{\sqrt{m}}{(N+1)^{k-\frac{1}{2}}}$. Then we easily know $X \subset Y^\epsilon$ & $Y \subset X^\epsilon$. Then it implies $d_H(X,Y) \leq \epsilon$. Now, letting ϵ approach $\sqrt{4k-2} \parallel R_N f^{(k)} \parallel_2 \frac{\sqrt{m}}{(N+1)^{k-\frac{1}{2}}}$, it holds.

(2) Note that mp(dgm) = $2d_B$(dgm, $dgm_\triangle$), where $dgm_\triangle$ is the diagram with the diagonal as underlying set, each point endowed with countable multiplicity. Then by the triangular inequality, $|mp(dgm(X)) - mp(dgm(Y))| =$ $|2d_B(dgm(X), dgm_\triangle)$ - $2d_B(dgm(Y), dgm_\triangle)| \leq 2d_B(dgm(X), dgm(Y))$.

(3) By the Stability Theorem, $d_B(dgm(X), dgm(Y)) \leq 2d_H(X,Y)$ and then by (1), $d_B(dgm(X), dgm(Y)) \leq 2d_H(X,Y) \leq 2\sqrt{4k-2} \parallel R_N f^{(k)} \parallel_2 \frac{\sqrt{m}}{(N+1)^{k-\frac{1}{2}}}$.

□

It follows that the persistent homology of the sliding window point cloud of a function f $\in C^k(\mathbb{T}, \mathbb{R})$ can, in the limit, be understood in terms of that of its truncated Fourier series.

2.3 Perea-Harer Conjecture

Remark 2.16. Let L $\in \mathbb{N}$ and f(t) = cos(Lt).

$$\text{Then } SW_{m,\tau}cos(Lt) := \begin{pmatrix} cos(Lt) \\ cos(Lt + L\tau) \\ \vdots \\ cos(Lt + L(m-1)\tau) \end{pmatrix}$$

$$= \begin{pmatrix} cos(Lt) \\ cos(Lt)cos(L\tau) - sin(Lt)sin(L\tau) \\ \vdots \\ cos(Lt)cos(L(m-1)\tau) - sin(Lt)sin(L(m-1)\tau) \end{pmatrix}$$

$$= cos(Lt) \begin{pmatrix} 1 \\ cos(L\tau) \\ \vdots \\ cos(L(m-1)\tau) \end{pmatrix} \text{ - } sin(Lt) \begin{pmatrix} 0 \\ sin(L\tau) \\ \vdots \\ sin(L(m-1)\tau) \end{pmatrix}$$

= cos(Lt)**a** - sin(Lt)**b**.

So, we can know that if **a** and **b** are linearly independent, then $SW_{m,\tau}cos(Lt)$ is a planar curve in $\mathbb{R}^m$ with its winding number L. Now, we research that as L & m & τ change, how the shape of this curve changes. Let A be the 2 x 2 matrix defined by $A := \begin{pmatrix} \| \mathbf{a} \|^2 & -\langle \mathbf{a}, \mathbf{b} \rangle \\ -\langle \mathbf{a}, \mathbf{b} \rangle & \| \mathbf{b} \|^2 \end{pmatrix}$. Then the following can be computed by the Lagrange trigonometric formula :

$$\langle \mathbf{a}, \mathbf{b} \rangle = \frac{1}{2} \sum_{n=0}^{m-1} sin(2Ln\tau) = \frac{sin(Lm\tau)sin(L(m-1)\tau)}{2sin(L\tau)},$$

$$\| \mathbf{a} \|^2 + \| \mathbf{b} \|^2 = \text{m},$$

$$\| \mathbf{a} \|^2 - \| \mathbf{b} \|^2 = \sum_{n=0}^{m-1} sin(2Ln\tau) = \frac{sin(Lm\tau)sin(L(m-1)\tau)}{2sin(L\tau)}.$$

Then A is positive semi-definite. Thus, the eigenvalues of A are non-negative and real number, $\lambda_1 \geq \lambda_2 \geq 0$ and there is a 2 x 2 orthogonal

matrix B so that $A = B^T \Lambda^2 B,\ where\ \Lambda = \begin{pmatrix} \sqrt{\lambda_1} & 0 \\ 0 & \sqrt{\lambda_2} \end{pmatrix}$.

Therefore, if x(t) = $\begin{pmatrix} cos(Lt) \\ sin(Lt) \end{pmatrix}$, then $\parallel SW_{m,\tau} cos(Lt) \parallel^2 = \langle\, x(t), Ax(t) \,\rangle$.

If the angle of rotation matrix B is α, then $SW_{m,\tau} cos(Lt) \mapsto \begin{pmatrix} \sqrt{\lambda_1} cos(Lt+\alpha) \\ \sqrt{\lambda_2} sin(Lt+\alpha) \end{pmatrix}$ is distance preserving, i.e. It is an isometry.

In summary, $SW_{m,\tau} cos(Lt)$ is an ellipse on the plane $span\{\mathbf{a}, \mathbf{b}\}$ and its axes are determined by the square roots of the eigenvalues $\sqrt{\lambda_1}$, $\sqrt{\lambda_2}$ of A. Then $\sqrt{\lambda_1} = \frac{m+|\frac{sin(Lm\tau)}{sin(L\tau)}|}{2}$ and $\sqrt{\lambda_2} = \frac{m-|\frac{sin(Lm\tau)}{sin(L\tau)}|}{2}$. Thus, when $\sqrt{\lambda_2}$ is maximum

($\Leftrightarrow\ Lm\tau \equiv 0\ mod\ \pi$), then the ellipse is roundest. With this conclusion, Perea and Harer conjecture one equation, $Lm\tau = 2\pi$ [**Perea-Harer Conjecture**] and we apply it to the practical experiments next chapters.

3 Implementation and Numerical Experiments

I primarily use the python 3.7.4 and GUDHI library for TDA in it to make the code of these examinations. (More specifically, I use the GUDHI library of python to make the Vietoris-Rips complex and its filtration for the persistent homology and compute the 1-dimensional max persistence.)

3.1 Implementation and Pseudo code for maximizing

Now, we look into some examinations using the sliding window method. The steps we use as follows (three bold texts are the parameter values at our examinations) :

(1) Set the function we examine.
(2) We assume our domain I = $[0, 2\pi]$ for convenience, divide I into n points evenly including 0, 2π and make new discrete domain $\bar{I}$ with these n points.
(**n : the number of sample data**)
(3) Sliding Window Method for m=2 or m=3.
(**m : the embedding dimension / t : sliding window step**)
(4) Because the embedding dimension m=2 or m=3, then we can plot the point cloud figure in $\mathbb{R}^2$ or $\mathbb{R}^3$.
(5) With the point cloud, we can find parameter t when the 1-dimensional persistence is maximum and measure the 1-dimensional max persistence.

Algorithm 1: Pseudo code

Input

: the number of sample data = n, the embedding dimension = m

Output

: 1-dimensional max persistence, and then window step & window size & mτ.

- making the sliding window = l

Define x by the list having n points from 0 to 2π. (use linspace)

For (1. add enough values to x about making sliding window 2. make the sliding window l for input dimension & every window step)

Substitute this nested list l to the function.

- computing 1-dimensional max persistence using GUDHI for each window step

1. make Vietoris-Rips complex and filtration
2. compute 1-dimensional persistence

- finding the max persistence

For each window step, find the 1-dimensional max persistence.

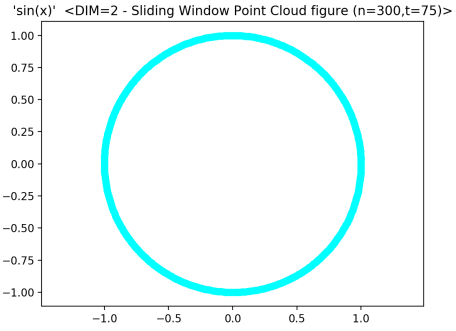
Through the First experiment, we focus only on m=2, m=3 where it can be visible directly. So, in these cases, we look into the shapes of point cloud figures using the sliding window method for our Second Experiment.

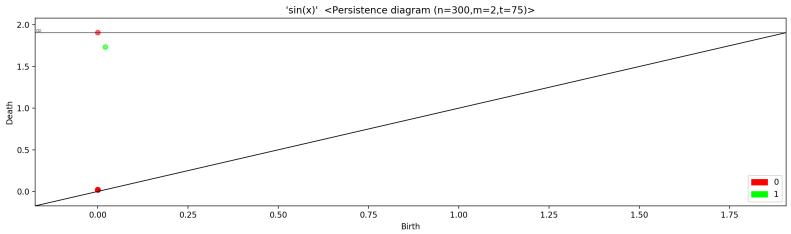
3.2 Basic results

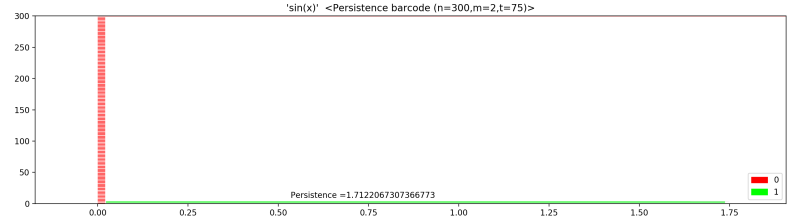
case 1 : sin(t), n=300, m=2

Persistence Sequence (descending order) = {[Persistence, m, t]} = {[1.7122, 2, 224], [1.7122, 2, 75], [1.7117, 2, 223], [1.7117, 2, 76], [1.7108, 2, 74], [1.7108, 2, 225], [1.7100, 2, 73], [1.7100, 2, 226], ... ,[0.0315, 2, 2], [0.0238, 2, 148], [0.0238, 2, 151], [0.0035, 2, 298], [0.0035, 2, 1], [0.0035, 2, 149], [0.0035, 2, 150]}

1-dimensional Max Persistence = 1.7122 (EMBEDDING DIMENSION(m) = 2 , t = 75 , STEP/DELAY(τ) = 1.5760 , WINDOW SIZE($(m-1)*\tau$) = 1.576 , $m*\tau$ = 3.1520)

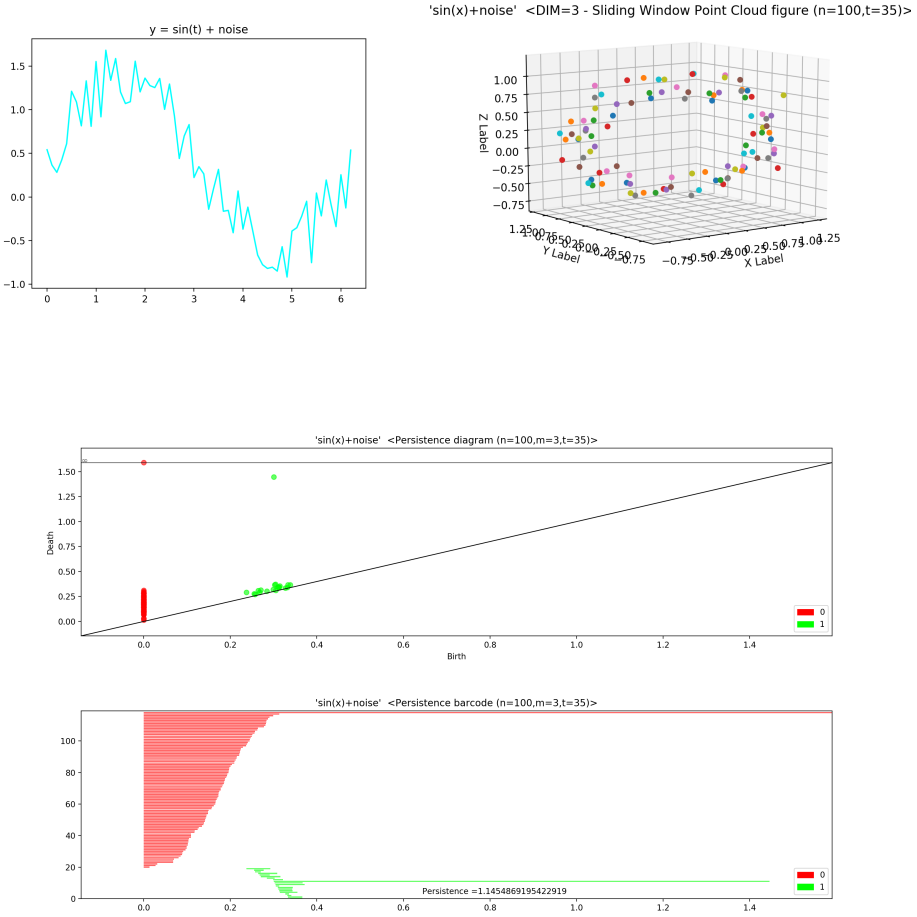






case 2 : sin(t) + noise, n=100, m=3

1-dimensional Max Persistence = 1.1454 (EMBEDDING DIMENSION(m) = 3 , t = 35 , STEP/DELAY(τ) = 2.2213 , WINDOW SIZE((m-1)*τ) = 4.4426 , m*τ = 6.6639)



In this case, the data has some noise and this noise occurs almost every time at the practical data. But as we can see, the sliding window point cloud figure is somewhat nice to apply our method to it because it has a meaningful figure and thus leads to meaningful persistence.

3.3 Normalization

Before the Second experiment, we first have to comment about 'normalizing' to compare different embedding dimensions.

Now, given the n points sample data, we don't wonder 1-dimensional max persistence 'value' at some embedding dimension m & window step t. (This is that as the embedding dimension m goes up, 1-dimensional max persistence value increases.) For each embedding dimension m, there is a corresponding window step t when 1-dimensional persistence is max. (equivalently, point cloud figure is the roundest.) We ultimately wonder given the n points sample data, point cloud figure is the roundest 'at what m & t' compared to the roundness of other m & t. (i.e. We find when the roundest occurs.) Now, we actually need when the 'roundest' occurs, not 'max persistence'. But, because we find it through when max persistence occurs and then max persistence increases as the radius increases (Note that the radius increases as m increases.), we have to normalize it so that the radius is considered the same for each m.

Remark 3.1. When the embedding dimension is m, then 1-dimensional max persistence occurs at $\tau = \frac{\pi}{m}$ and then point cloud figure is the roundest and the center is the origin. Now, we find the radius at each embedding dimension m and divide the point cloud vectors by the radius. To find the radius at each embedding dimension m, we only need one point in the point cloud because the center of point cloud figure is the origin. The next are some results to find the radius at the sine function.

(1) m=2, $\tau = \frac{\pi}{2}$: The first point = (sin0, sin$\frac{\pi}{2}$) = (0, 1) and hence radius = $\|(0, 1)\| = 1 = \frac{\sqrt{4}}{2}$.
(2) m=3, $\tau = \frac{\pi}{3}$: The first point = (sin0, sin$\frac{\pi}{3}$, sin$\frac{2\pi}{3}$) = (0, $\frac{\sqrt{3}}{2}$, $\frac{\sqrt{3}}{2}$) and hence radius = $\|(0, \frac{\sqrt{3}}{2}, \frac{\sqrt{3}}{2})\| = \frac{\sqrt{6}}{2}$.
(3) m=4, $\tau = \frac{\pi}{4}$: The first point = (sin0, sin$\frac{\pi}{4}$, sin$\frac{2\pi}{4}$, sin$\frac{3\pi}{4}$) = (0, $\frac{\sqrt{2}}{2}$, 1, $\frac{\sqrt{2}}{2}$) amd hence radius = $\|(0, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2})\| = \frac{\sqrt{8}}{2}$.

This induction holds for every m. So, when m = k, then $\tau = \frac{\pi}{k}$ and r = $\frac{\sqrt{2k}}{2}$. And we can normalize every cloud point dividing by the radius r = $\frac{\sqrt{2k}}{2}$.

3.4 Dependence for the dimension

Now, we don't fix the embedding dimension m, find 1-dimensional max persistence for all embedding dimension m & sliding window step τ, and look into then what the embedding dimension m & sliding window step τ are.

The steps we use as follows (three bold texts are the parameter values at our examinations) :

(1) We assume our domain I = $[0, 2\pi]$ for convenience, divide I into n points evenly including 0, 2π and make new discrete domain $\bar{I}$ with these n points.
(**n : the number of sample data**)
(2) Sliding Window Method
(**m : the embedding dimension / t : sliding window step**)
(3) Given the parameter n, we can find 1-dimensional max persistence and what the parameters m and t are at that time.

At the Second examination, because the 1-dimensional max persistence doesn't usually occur when m=2 or m=3, we cannot look into the point cloud figures directly unlike the First examination. But because we already looked into the aspects of point cloud figure at the First examination, we focus on finding when 1-dimensional max persistence occurs for every embedding dimension m and sliding window step t. And we check [Perea-Harer] conjecture equation, Lm$\tau = 2\pi$ (L : frequency).

At this experiment, because our algorithm calculates for all embedding dimension & window step to find 1-dimensional max persistence and then it takes a long time to find it, we put the somewhat less number of sample data compared to the first experiment.

case 1 : sin(t), n=50

Max Persistence = 1.6450
(EMBEDDING DIMENSION(m) = 7, Sliding Window Step(t) = 7, STEP/DELAY(τ) = 0.8975, WINDOW SIZE((m-1)τ) = 5.3855 , mτ = 6.2831)

At this case, Lmτ = 2π.

case 2 : sin(2t), n=51

Max Persistence = 1.5589
(EMBEDDING DIMENSION(m) = 5, Sliding Window Step(t) = 5, STEP/DELAY(τ) = 0.6283, WINDOW SIZE((m-1)τ) = 2.5132 , mτ = 3.1415)

At this case, Lmτ = 2π.

case 3 : sin(t) + cos(3t), n=50

Max Persistence = 1.6285
(EMBEDDING DIMENSION(m) = 7, Sliding Window Step(t) = 7, STEP/DELAY(τ) = 0.8975, WINDOW SIZE((m-1)τ) = 5.3855 , mτ = 6.2831)

At this case, Lmτ = 2π.

case 4 : sin(3t) + cos(7t), n=55

Max Persistence = 0.9598
(EMBEDDING DIMENSION(m) = 6, Sliding Window Step(t) = 3, STEP/DELAY(τ) = 0.3490, WINDOW SIZE((m-1)τ) = 1.7453 , mτ = 2.0943)

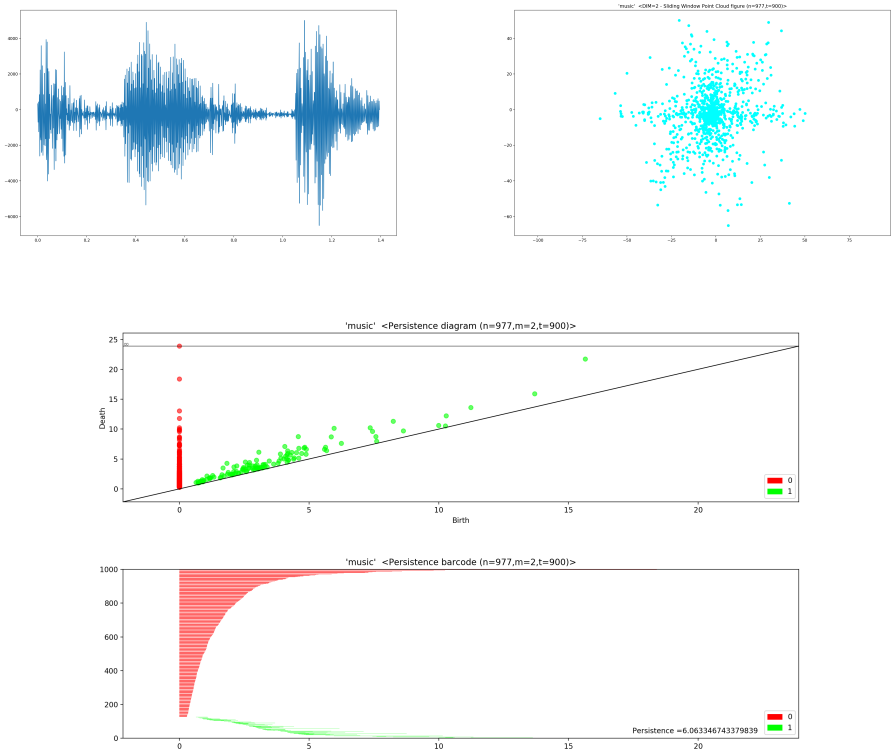
At this case, Lmτ = 2π.

As you can see, [Perea-Harer] conjecture equation holds well at this experiment in spite of the somewhat less number of sample data. But because these functions are a bit simple compared to the practical data. So we need to apply our method to that.

3.5 Practical Application : Music

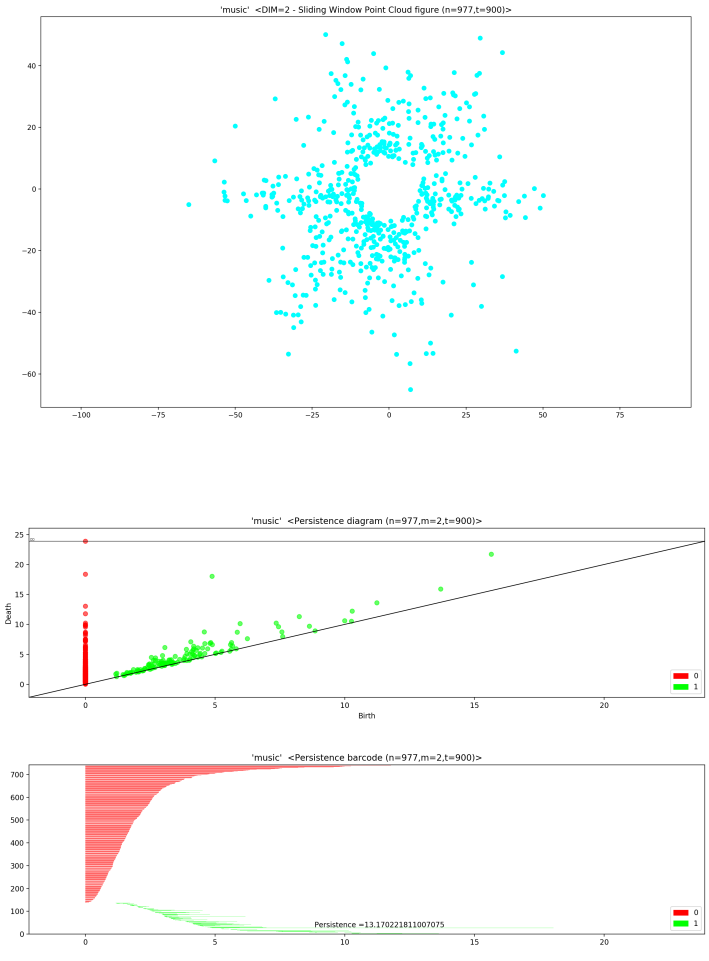
Until now, we found the frequencies of some time-series by using the persistent homology & sliding window method. Now, we apply this method to practical data. More specifically, after extracting time-series at the real pop music wav. file, we do our first, second experiment.

We use the pop music wave file whose (1) sample rate = 700Hz (2) Time ≈ 1.39(s). So the number of sample data is 977.



In this experiment, we can hardly find the meaningful frequency of this music because there are so many soft and loud amplitudes together in this music as you could see in the time-series figure (left) and sliding window figure (right). Since there are always so many different amplitudes together like this in the music sound, and then it's hard to find the significant 1-dimensional loops in the sliding window figure, so applying our method to the practical music sound is not easy.

Above the experiment, we can not find meaningful results because of the existence of much different amplitude data. More specifically, there exists a lot of noise in it. So then we can try to revise this data, removing some soft amplitudes, 'noise' from our same music data.



As a result, now a meaningful sliding window point cloud occurs as we can expect. Therefore, we know that if we remove these some music noise, we can apply our method to it properly in the same way.

3.6 Comparison with FFT

In the previous our experiments, the following two things are important : (1) **Accuracy** (2) **Speed**. But, when we compare our method that uses persistent homology & the sliding window method with Fast Fourier Transform (FFT) that is commonly used these days at the time-series analysis, FFT is far superior at all cases we looked into in terms of (1) Accuracy (2) Speed.

4 Summary and Future works

In this paper, we saw that the frequency of time-series can be found using persistent homology and sliding window method. To put it concretely, (1) in the first experiment, we looked into the 1-dimensional max persistence (and then the embedding dimension, window step, window size) & the sliding window point cloud figure. Observing it directly in the embedding dimension = 2 or 3, we could understand intuitively the reason why the sliding window cloud figure is roundest when the 1-dimensional persistence is max and the shape of visualization in the situation as the window step changes. In particular, even in the sine wave including noise, our method could be seen as positive. (2) Before the second experiment, we considered the necessity of 'normalizing' and that concrete method we use. (3) In the second experiment, we found the 1-dimensional max persistence of all the embedding dimension & window step and confirmed whether [Perea-Harer] conjecture equation holds well. (4) And we applied this method to the practical music wave file. (5) Lastly, we commented on the comparison with FFT. By (4) & (5), we can see that our method is much inferior to FFT.

FFT is certainly a very useful method but it also has some problems. So, we could solve it by improving our topological method. And in our most experiments, [Perea-Harer] conjecture equation is somewhat useful but as in previous some experiments, when complex data are given, it is less accurate than FFT. Thus, this equation also needs to be supplemented.

Bibliography

[1] A. Hatcher, Algebraic Topology, Cambridge Univ. Press, England, 2002.

[2] Jose A. Perea and John Harer, Sliding Windows and Persistence: An Application of Topological Methods to Signal Analysis, 2013.

[3] G. Carlsson, Topology and data, 2009.

[4] H. Edelsbrunner and J. Harer, Computational Topology : An Introduction, American Mathematical Soc., 2010.

[5] Munch E, Applications of persistent homology to time varying systems, Duke University, 2013.

[6] Fred H. Croom, Basic Concepts of Algebraic Topology, 1978.

[7] Frédéric Chazal and Bertrand Michel, An introduction to Topological Data Analysis: fundamental and practical aspects for data scientists, 2017.

[8] Huke, J. P., Embedding Nonlinear Dynamical Systems : A Guide to Taken's Theorem, 2006.

[9] M. A. Pinsky, Introduction to Fourier Anlysis and Wavelets, The Brooks/Cole Series in Advanced Mathematics, USA, 2003.

[10] Steve Y.Oudot, Persistence Theory - From Quiver Representations to Data Analysis, 2015.

[11] J. A. Perea, A. Deckard, S. B. Haase and J. Harer, SW1PerS: Sliding Windows and 1Persistence Scoring; Discovering Periodicity in Gene Expression Time Series Data, preprint, 2013.

[12] A. J. Blumberg, I. Gal, M. A. Mandell and M. Pancia, Persistent homology for metric measure spaces, and robust statistics for hypothesis testing and confidence intervals, 2012.

국문초록

이 논문에서 우리는 펄시스턴트 호몰로지와 슬라이딩 윈도우 기법을 이용한 시계열 해석에 대해 살펴 볼 것이다. 구체적으로, 우리는 슬라이딩 윈도우 기법으로 삼각 함수들의 일차원 펄시스턴스의 최댓값에 관해 조사할 것이다. 그리고 이 일차원 펄시스턴트의 최댓값인 상황에서 우리는 슬라이딩 윈도우 기법의 변수인 엠베딩 차원과 스텝, 그리고 함수의 주파수의 관계에 대해 생각해볼 수 있다. 더불어, 이런 과정과 푸리에 해석의 내용에 의해 우리는 시계열 샘플 데이터로 부터 함수의 주파수를 예측해 볼 수 있다.

주요어휘 : 시계열 분석, 펄시스턴트 호몰로지, 슬라이딩 윈도우, 펄시스턴스의 최댓값, 삼각 함수, 주파수
학번 : 2017-29039

감사의 글

이 논문이 완성되기까지 많은 분들의 도움이 있었습니다. 도와주신 모든 분들께 이 자리를 빌려 진심으로 감사의 말씀을 전하고 싶습니다.
가장 먼저, 제가 이 자리까지 오게 해주신 아버지, 어머니께 진심으로 감사드립니다. 부모님의 사랑과 믿음이 있었기에 그동안 힘을 내올 수 있었습니다. 몸 건강히 오래도록 평안하시길 바라며, 제가 돕겠습니다.
다음으로 제 지도 교수님이신 Otto van Koert 교수님께 감사의 말씀을 드립니다. 많이 미흡한 제자인 탓에 교수님의 노고가 많이 크셨을 텐데도 항상 열정적인 교수님의 관심과 가르침 덕분에 이 논문이 완성될 수 있었습니다. 죄송한 마음과 감사한 마음이 참 깊습니다. 수학에 대한, 그리고 지도 학생들을 가르침에 대한 교수님의 지속적인 깊은 열정에 많이 배우고, 큰 존경을 표합니다.
그리고 같은 연구실 소속 최필립, 백태진, 정효진 누나에게 감사의 말씀을 전합니다. 별다른 이유 없이 항상 저를 잘 대해주고 좋은 기운을 준 이분들 덕분에 그동안 잘 지내올 수 있었습니다. 특히나 이번 논문에 관해 큰 도움을 주었던 최필립 연구원, 본인의 지식을 진심으로 아낌없이 나누어 도와준 그 따뜻한 마음에 다시 한번 깊은 감사를 표합니다.
또한, 같은 소속은 아니지만 같은 공간에서 함께 지내 온 박상준에게도 감사의 말을 전하고 싶습니다. 역시나 별다른 이유 없이 늘 저를 잘 대해주고, 적적한 시기의 좋은 벗이면서 수학적인 많은 도움을 그에게서 얻었습니다.
그리고 수학 외의 제 일탈 낙이었던 바운스 팩토리, 그 친구들에게 감사의 말씀을 전합니다. 여러분들 덕분에 힘내서 해나갈 수 있었습니다. 함께한 추억이 참 많은 이 시절 좋은 추억들, 평생 잊지 못할 것입니다.
도움의 시기를 더 거슬러 올라가, 제가 이 자리까지 수학을 해올 수 있는 기반의 도움을 줬던 고성길, 김세훈, 이병찬에게도 감사의 말씀을 전합니다. 세 사람 덕분에 아무것도 모르는 시기의 제가 이 자리까지 오는 추진을 낼 수 있었습니다. 마찬가지로 그 시기의 가장 많은 도움을 주셨던, 늘 따뜻하게 저를 잘 대해주시고 도와주셨던 문현숙 교수님, 그리고 항상 좋은 말씀으로 앞길을 도와주셨던 임정욱 교수님, 김필수 교수님, 김은섭 교수님, 양승엽 교수님께 이 자리를 빌려 진심으로 감사의 말씀을 전합니다.
마지막으로 제 친구 차승철에게 심심한 감사를 전하며 이 지면에 언급하지 못했지만 그동안 제 수학 길을 도와주신 모든 분들께 다시 한번 진심으로 감사의 말씀을 전합니다. 수학과 함께여서 참 즐거웠습니다. 감사합니다.