ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Sliding Windows and Persistence: Topological Analysis of Time Series Data

Master Thesis

Luisa S. Meinecke

18.03.2025

Advisor: Dr. S. Kališnik Hintz

Department of Mathematics, ETH Zürich

Abstract

In this thesis we will investigate the application of topological data analysis (TDA) to time series data. The focus will be on the method sliding windows and persistence. By converting time series data into point clouds, we can use persistent homology to detect underlying periodicity and geometric patterns. To ensure the robustness of this method, we introduce and prove the approximation and stability theorem. We will see that changing the embedding dimension and window size has an impact on the geometric properties and structure of the sliding window. The higher the embedding dimension, the higher the level of detail we can capture through the embedding. When the window size corresponds to the underlying frequency of the signal, then maximum persistence functions as a level of roundness of the point cloud. This thesis is concerned with highlighting the advantages of a topological analysis of time series data compared to classical techniques. We mainly follow the paper by Jose A. Perea and John Harer named *Sliding Windows and Persistence: An Application of Topological Methods to Signal Analysis* [25].

Acknowledgements

I would like to sincerely thank Dr. Sara Kalisnik for supervising my master's thesis. In particular, I thank her for her helpful comments on the early drafts of my thesis, for her help in selecting my main paper, and for her support in developing its content. I would also like to thank my family and friends for supporting me throughout the process. [1]

[1] Moreover, I am grateful that I could use the thesis template provided by CADMO.

Contents

Chapter 1

Introduction

In this thesis, the goal is to topologically analyze time series data. A time series is a set of observations x_t, where every x_t is recorded at a specific time t. With real world data, we always have a finite number of time points for a time series, so we can write it as

$$\{x_t | t = 0, \ldots, T\}.$$

We also use the notation $x(t)$ for time series. Time series can be found in many different fields. Here are some examples of real world time series:

- Weather data: Temperature recorded every hour. Here x_t is the temperature at time t.
- Financial data: Monthly inflation rates.
- Physical data: Voltage levels in an electrical circuit over time.
- Economic data: Daily traffic flow on a highway.

Another example, on which we demonstrate how topological methods work, are heart rate measurements from a patient, for example EKG signals. For the background in EKG signals, we follow an article by Accardi et al. [1] and one by Becker [2]. An EKG measures the electrical activity of the heart muscle fibers and is quasi-periodic. This means that the signal never perfectly repeats as when it was periodic, but it still has a structured, oscillatory nature. The analysis of EKG signals helps predicting whether an individual is likely to suffer from a disease in the future and it helps detect the worsening of a patient's health condition from heart rate data.

The EKG data is distorted by noise. This noise can be caused by physiological differences or measurement inaccuracies in the devices, among other things. Although classic Fourier-based methods can estimate the average heart rate, they fail to recognize small variations in the signal. Time series analysis is particularly important and efficient when we want to filter out significant properties from complex, noisy or non-linear signals.

Let us take a look at any EKG time series and label it $x(t)$. This time series represents the measured voltage of the electrical activity of the heart over time. Depolarization is a process in which a rapid change in electrical charge occurs. A typical EKG wave has the following three main components:

- P waves represent the depolarization of the atrium[1].
- QRS complexes (characterized by a sharp peak) represent ventricular[2] depolarization.
- T waves (characterized by a broader peak) represent ventricular depolarization.

An EKG time series signal is quasi-periodic. This means that although the rough waveform is repeated, fluctuations occur from beat to beat, for example due to cardiac abnormalities.

A classical Fourier transform would extract the dominant frequency component, corresponding to the heart rate. However, it ignores the geometric structure of the waveform and cannot distinguish between normal and abnormal beats based on shape alone. Instead, we apply the sliding window embedding to reconstruct the topological structure of the EKG signal in a higher-dimensional space. Given the time series $x(t)$, the embedding is defined as:

$$SW_{M,\tau}x(t) = \begin{bmatrix} x(t) \\ x(t+\tau) \\ \vdots \\ x(t+M\tau) \end{bmatrix} \in \mathbb{R}^{M+1},$$

where M is the embedding dimension and τ is the delay parameter. For an EKG signal with a heart beat of 60 beats per minute (bpm), the typical period is $T \approx 1$ second. We generate a synthetic EKG-like time series, that is displayed in Figure 1.1. Choosing the dimension to be $M = 2$ and the delay parameter to be $\tau = 50$ ms yields a 3-dimensional representation of the EKG waveform:

$$SW_{2,50}x(t) = \begin{bmatrix} x(t) \\ x(t+50\text{ms}) \\ x(t+100\text{ms}) \end{bmatrix} \in \mathbb{R}^3,$$

see Figure 1.2.

To study this point cloud topologically, we build a family of Vietoris-Rips complexes on it. Simplicial complexes are combinatorial objects build of

[1] The upper two chambers of the heart.

[2] Refers to the ventricles, the lower two chambers of the heart.

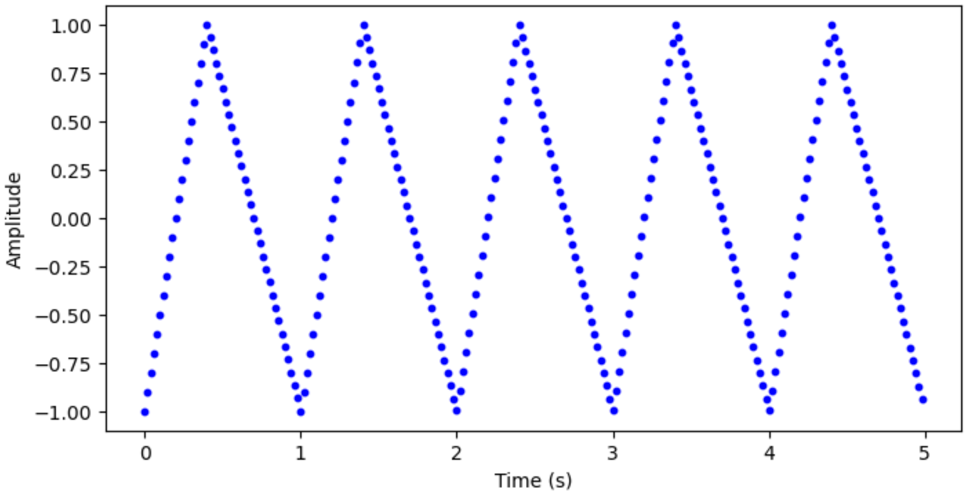


Figure 1.1: Synthetic EKG signal.

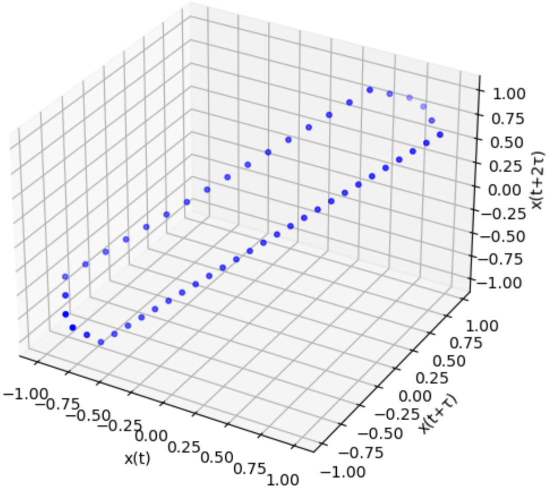


Figure 1.2: 3-dimensional sliding window embedding.

simplices. A 0-simplex is a point, a 1-simplex is a line, a 2-simplex is a triangle, a 3-simplex is a tetrahedron and so on. We now get a simplicial complex by gluing simplices of different dimensions along common faces. The Vietoris-Rips complex is defined the following way: For some point cloud X and some radius r, $k+1$ points in our point cloud form a k-simplex if and only if the pairwise distances between the points are smaller than a given radius r. The collection of all these simplices is called the **Vietoris-Rips complex**.

To demonstrate how this works, we look at the 2-dimensional projection of the sliding window point cloud and build the Vietoris-Rips complex on it, see Figure 1.3. We now have a topologically interpretable object, that forms a closed trajectory. The resulting 3-dimensional trajectory is a geometric representation of the heartbeat cycle:

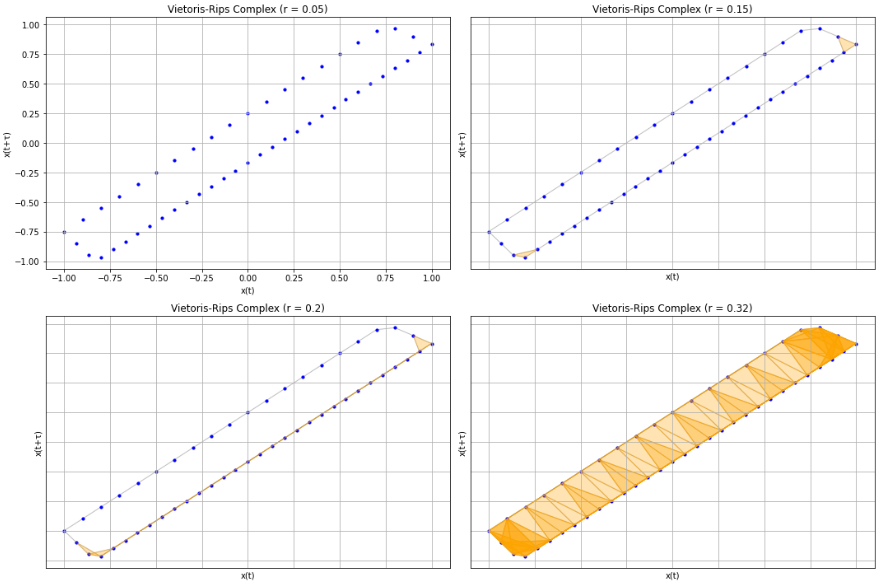


Figure 1.3: Vietoris-Rips complexes for different values of r.

- EKG signals form a closed trajectory that represents the periodicity of the heart cycles.
- With real world data, there are often small distortions of the data, for example due to arrhythmia beats, which lead to a distortion of the trajectory. These anomalies can be recognized with persistent homology.

We quantify these features, with persistent homology and display it in Figure 1.4. Intuitively, a persistence diagram keeps track of loops and higher dimensional cavities in the family of Rips-complexes. The persistence diagram reveals a dominant 1-dimensional feature (a loop) corresponding to the cardiac cycle. For real world data, there would be additional transient features that may indicate irregular beats, skipped beats, or arrhythmias.

The code for Figures 1.1, 1.2 and 1.4 can be found in the Appendix, Figure A.1, the one for Figure 1.3 in Appendix, Figure A.2.

In this thesis we investigate how changes in the window size $M\tau$, the embedding dimension M and the delay parameter τ, when we apply sliding windows to a signal, affect the topological structure of the embedded signal. We will establish and prove the stability theorem (Theorem 2.35), which gives us the guarantee that small perturbations in the signal do not significantly change the persistence diagram. This is very important because real world

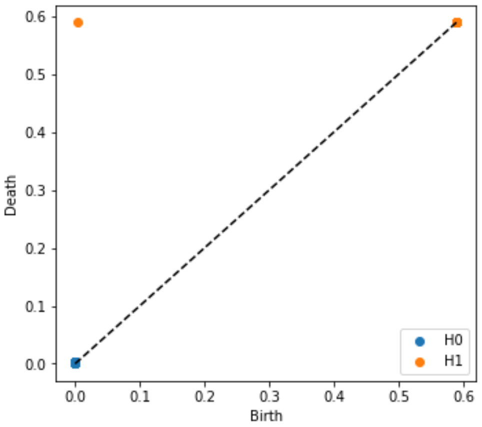


Figure 1.4: Persistence diagram for the sliding window of the EKG time series. The x-axis shows the birth time and the y-axis the death time. So the upper left orange dot represents a 1-dimensional structure, for example a loop, that births at time 0 and dies at time 0.6 approximately.

data, such as EKG data, often contains small deviations, such as noise. By proving stability, as discussed in the stability theorem, we see that persistent homology is reliable in analyzing time series data. This is in the sense that it is not overly sensitive to small fluctuations. By proving the approximation theorem (Theorem 5.5), we can show that the topological features of the trajectory obtained by sliding windows capture the periodicity of a signal more robustly than classical methods. This is particularly important because traditional methods may have problems analyzing irregularly sampled or non-linear time series. By adopting topological approaches, this thesis provides a method to recognize periodic structures well even in complex signals.

Chapter 2

Persistent Homology

Persistent homology, a core tool in topological data analysis (TDA), provides a way to quantify features such as connected components, loops, and higher dimensional cavities in a dataset. For general background in homology we refer the reader to the book „Algebraic Topology“ by A. Hatcher [15]. For this chapter we follow our main reference [25] and two articles by Carlsson [5], [6].

2.1 Simplices and Simplicial Complexes

The goal of this section is to represent complex datasets in a structured way, enabling the study of their topological features. In TDA, we analyze data points in relation to their neighbors. Simplices provide a way to encode these relationships by generalizing the notion of connectivity.

Definition 2.1 *Let $X = \{x_0, x_1, \dots, x_n\}$ a subset of an Euclidean space $\mathbb{R}^k$. This set is in **general position** if it is not contained in an affine hyperplane of $\mathbb{R}^k$ of dimension m such that $m < n$.*

Definition 2.2 *For a set X in general position, we define the **n-simplex** σ to be the convex hull of X. Note that the x_i are called **vertices** and the simplices that are spanned by a subset $Y \subseteq X$ for $|Y| = i$ are called **i-faces**.*

Definition 2.3 *A **geometric simplicial complex** is a finite collection $\mathcal{X}$ of simplices in an Euclidean space, such that the following two conditions are satisfied:*

- *For any face f in a simplex σ of $\mathcal{X}$, it holds that $f \in \mathcal{X}$.*
- *For any two simplices σ and τ of $\mathcal{X}$, $\sigma \cap \tau$ is a simplex of $\mathcal{X}$ and a face of both, σ and τ.*

Definition 2.4 *An **abstract simplicial complex** is a pair (V, Σ), where V denotes some finite set and Σ is a collection of non-empty subsets of V, such that for a simplex $\sigma \in \Sigma$ and some $\tau \subseteq \sigma$, it holds that $\tau \in \Sigma$.*

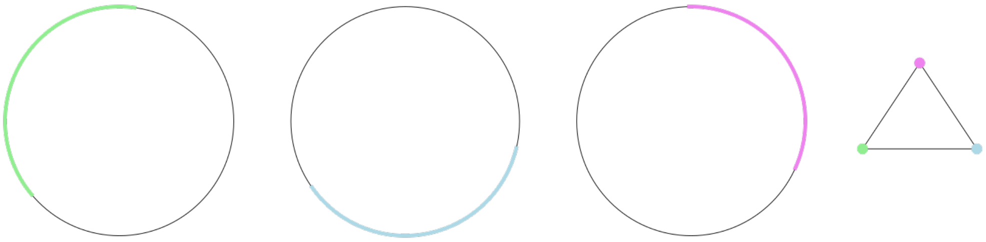


Figure 2.1: Here we have a covering of the sphere S^1 by three open sets U_1 (green), U_2 (blue) and U_3 (violet). So for the construction of the nerve we have three vertices v_0 (green), v_1 (blue) and v_3 (violet). The three open sets overlap pairwise, so our vertices form three 1-simplices, displayed at the right.

2.2 Nerves, Čech and Vietoris-Rips Complexes

Given a dataset as a set of points X in a metric or topological space, a primary challenge is defining a simplicial complex that captures the underlying topological structure of the data. This section introduces key constructions: nerve, Čech, and Vietoris-Rips complexes, that allow for systematic ways of building these complexes.

Let now X be any topological space and $\mathcal{U} = \{U_i\}_{i \in I}$ be a covering of X. For this part we follow a paper by Carlsson [5].

Definition 2.5 *The **nerve** of $\mathcal{U}$, denoted by $N(\mathcal{U})$, is the simplicial complex with vertices I and where a set of $k+1$ vertices $\{i_0, \dots i_k\}$ span a k-simplex if and only if $U_{i_0} \cap \dots \cap U_{i_k} \neq \emptyset$.*

To see that the so defined nerve $N(\mathcal{U})$ is homotopy equivalent to the underlying space X, we need the so-called Nerve Theorem [5].

Theorem 2.6 *Suppose that the covering $\mathcal{U}$ of X consists of open sets and is numerable. Suppose further that if $\emptyset \neq J \subseteq I$, then $\bigcap_{j \in J} U_j$ is contractible or empty. Then $N(\mathcal{U})$ is homotopy equivalent to X.*

Example 2.7 *We see from Figure 2.1 that the nerve in our example is indeed homotopy equivalent to our underlying space S^1.*

We now want to look at two different methods to generate coverings, so that we get new complexes. Suppose now that X is a metric space. Then we can construct a covering by placing small balls around every $x \in X$; $\mathcal{B}_r(X) = \{B_r(x)\}_{x \in X}$, where r is the radius > 0.

Definition 2.8 *Let $Y \subseteq X$, such that $X = \bigcup_{y \in Y} B_r(y)$. The **Čech complex** of X attached to Y and r, denoted by $\check{C}(Y, r)$, is the nerve of the covering $\{B_r(y)\}_{y \in Y}$.*

Example 2.9 *In Figure 2.2 we see a Čech complex on $n = 17$ vertices. With increasing radius we see three holes appear and disappear.*

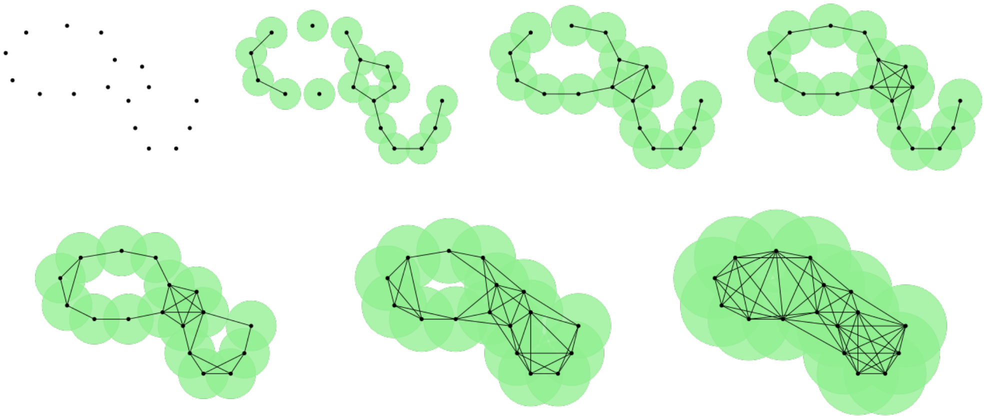


Figure 2.2: Growing union of balls and the corresponding Čech complex.

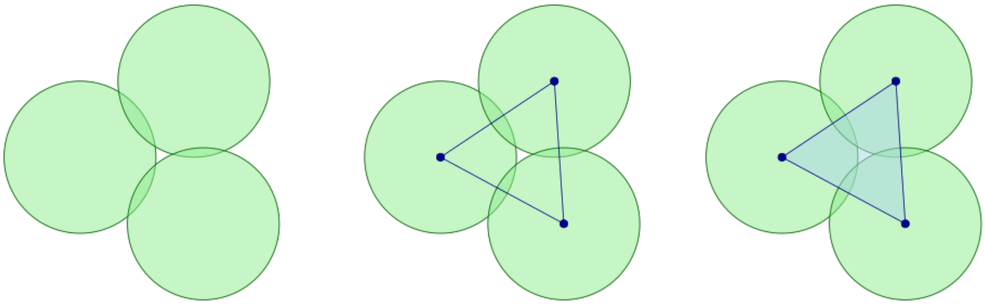


Figure 2.3: left: covering, middle: Čech complex, right: Vietoris-Rips complex

The construction of the Čech complex comes with high computational costs, because it needs to store the information of simplices of various dimensions. One way to solve this problem is a variant of the Čech complex, the Vietoris-Rips complex. This construction only uses the information from the edges. We again take X to be a metric space and denote by d the corresponding metric.

Definition 2.10 *The* ***Vietoris-Rips complex****, $VR(X, r)$, is the simplicial complex on the vertex set X and where a set of $k+1$ vertices $\{x_0, \dots, x_k\}$ span a k-simplex if and only if $d(x_i, x_j) \leq r$ for all i, j between 0 and k.*

Example 2.11 *We want to illustrate the differences between the Čech complex and the Vietoris-Rips complex. For that, look at Figure 2.3. In the left, we see a covering of some space. In the middle is the corresponding Čech complex, which is also the nerve of the covering and on the right we see the Vietoris-Rips complex, a 2-simplex.*

2.3 Persistence Objects and Classification Theorem

This section introduces the concept of persistence objects within the framework of category theory and explores persistence in the context of simplicial complexes and homology. A key part of the section is the Classification Theorem for finitely generated $F[t]$-modules.

In the following let now X be a subspace of $\mathbb{R}^n$. We suppose that there is a method to sample points from X. These sampled points may contain noise. Let $\mathbb{X}$ be such a sample. When constructing complexes like Čech or Vietoris-Rips complexes, it may not be clear what radius r we should use. Different r can lead to different topological features and it might not have a single "right" r. So instead we take 1-parameter families of simplicial complexes, that are indexed by r. This is formalized in a concept named persistence objects. To define these let $\underline{\mathcal{C}}$ be a category and $\mathcal{P}$ a partially ordered set. We can look at $\mathcal{P}$ as a category by setting the object set $ob(\underline{\mathcal{P}}) = \mathcal{P}$ and a unique morphism $m \colon x \to y$ if $x \leq y$.

Definition 2.12 *A $\mathcal{P}$-**persistence object** in $\underline{\mathcal{C}}$ is a functor $\Phi \colon \underline{\mathcal{P}} \to \underline{\mathcal{C}}$. So it is a family of objects of $\underline{\mathcal{C}}$, $\{c_x\}_{x \in \mathcal{P}}$, together with morphisms $\phi_{xy} \colon c_x \to c_y$ for $x \leq y$, such that $\phi_{yz} \circ \phi_{xy} = \phi_{xz}$. We denote the **category of $\mathcal{P}$-persistence objects** in $\underline{\mathcal{C}}$ by $\mathcal{P}_{pers}(\underline{\mathcal{C}})$.*

When analyzing data using homology, not all topological features are equally important. Some persist over a wide range of time, while others appear only briefly and can be attributed to noise. Persistent homology assigns a birth and death time to each feature, providing a systematic way to quantify its significance.

Let S be some simplicial complex gained from a point cloud and let p denote a prime number. We work over $\mathbb{F}_p$.

Definition 2.13 *A **subcomplex** of a simplicial complex S is a subset of its simplices, that is closed under the face relation.*

Definition 2.14 *Let S be some simplicial complex. A **filtration** of S is a sequence of several subcomplexes S_i, where the first one is the empty set and the last one is S, of the form*

$$\emptyset = S_0 \subseteq S_1 \subseteq \ldots \subseteq S_m = S.$$

Example 2.15 *Filtrations are an example for persistence objects. When applied to a Čech complex we get a Čech filtration and similarly when we apply it to a Vietoris-Rips complex, we get a Rips filtration.*

Applying homology with coefficients in some field F to all the subcomplexes gives us the following sequence of maps, induced by the inclusion maps

$$0 = H_k(S_0) \to H_k(S_1) \to \cdots \to H_k(S_m) = H_k(S).$$

For a solid background in homology, we refer the reader to a book by Allen Hatcher [15].

Remark 2.16 *If $f\colon \mathcal{P} \to \mathcal{Q}$ is a partial order preserving map, then we can get an evident functor $f^*\colon \mathcal{Q}_{pers}(\underline{\mathcal{C}}) \to \mathcal{P}_{pers}(\underline{\mathcal{C}})$, which is defined by $f^*(\psi) = \psi \circ \underline{f}$, where $\underline{f}\colon \underline{\mathcal{P}} \to \underline{\mathcal{Q}}$.*

Let $\mathbb{R}$ and $\mathbb{N}$ be the partially ordered sets of the real numbers and the natural numbers respectively. We can do the constructions from the previous sections analogously and get an $\mathbb{R}$-persistence simplicial complex attached to $\mathbb{X}$. Using this to construct chain complexes and homology groups, we get $\mathbb{R}$- chain complexes and $\mathbb{R}$-persistent homology groups. Homology is very useful to distinguish topological spaces, because there is a classification theorem for finitely generated abelian groups. If we had a similar theorem for $\mathbb{R}$-persistent abelian groups, then it would tell us about the behavior of the homology of all complexes $\check{C}(X, r)$. But we do not have such a theorem. However, there is a similar statement for a subcategory of the category of $\mathbb{N}$-persistence F-vector spaces for a field F. $\mathbb{N}$-persistence abelian groups can easily be viewed as graded modules over graded rings. This association works as follows. Suppose we have a $\mathbb{N}$-persistence abelian group $\{G_n\}$. Now we define a graded module associated to this abelian group.

Definition 2.17 *Let $\mathbb{Z}[t]$ be the graded polynomial ring. The* ***graded module*** *associated to $\{G_n\}$ is*

$$\theta(\{G_n\}) = \bigoplus_{s\geq 0} G_s, \tag{2.1}$$

where the n^{th} graded part is the group G_n.

The action of t, which generates $\mathbb{Z}[t]$, is defined as

$$t \cdot \{\alpha_n\} = \{\beta_n\}, \text{ where } \beta_n = \psi_{n-1,n}(\alpha_{n-1}). \tag{2.2}$$

Here the morphisms $\psi_{a,b}$ are given by multiplication with t^{b-a}. Let now F be any field, then we can state a classification theorem for finitely generated $F[t]$-modules.

Theorem 2.18 (Classification Theorem) *Let M_* be a finitely generated non-negatively graded $F[t]$-module. Then there exist three tuples of integers $\{i_1, \ldots, i_m\}$, $\{j_1, \ldots, j_n\}$, $\{k_1, \ldots, k_n\}$ and an isomorphism*

$$M_* \cong \bigoplus_{s=1}^{m} F[t](i_s) \oplus \bigoplus_{t=1}^{n} (\frac{F[t]}{t^{k_t}})(j_t). \tag{2.3}$$

This decomposition is unique up to permutation of factors.

Here we use the following notation; $M_*(s)_k = N_{k-s}$, which is an upward dimension shift.

Definition 2.19 *An $\mathbb{N}$-persistence F-vector space $\{V_n\}_n$ is called* ***tame****, if the following two conditions hold:*

- *V_n is finite dimensional for all n,*
- *$\psi_{n,n+1} \colon V_n \to V_{n+1}$ is an isomorphism for large enough n.*

As a corollary from this statement for $\mathbb{N}$-persistence F-vector spaces, we have the following.

Proposition 2.20 *$\theta(\{V_n\}_n)$ is a finitely generated $F[t]$-module if and only if $\{V_n\}_n$ is tame.*

Proof First we prove the implication **$\theta(\{V_n\}_n)$ is a finitely generated $F[t]$-module $\implies$ $\{V_n\}_n$ is tame**.
If $\theta(\{V_n\}_n)$ is a finitely generated $F[t]$-module, then Theorem 2.18 implies that it decomposes as follows

$$\theta(\{V_n\}_n) \cong \bigoplus_{s=1}^{m} F[t](i_s) \oplus \bigoplus_{t=1}^{n} (\frac{F[t]}{t^{k_t}})(j_t), \tag{2.4}$$

where the first summand corresponds to free $F[t]$-modules and the second summand corresponds to torsion $F[t]$-modules. From this decomposition, we have that for sufficiently large n, $\{V_n\}_n$ stabilizes, so $\psi_{n,n+1}$ is an isomorphism. This follows directly from the fact that the torsion modules $\frac{F[t]}{t^{k_t}}(j_t)$ are finitely supported and the free modules $F[t](i_s)$ correspond to shifts in stable dimensions (dimension of vs remains constant). Moreover, since the graded components of $\theta(\{V_n\}_n)$ are finite-dimensional, V_n is finite dimensional for all n. Therefore $\{V_n\}_n$ satisfies all conditions for being tame.

Now onto the second implication; **$\{V_n\}_n$ is tame $\implies$ $\theta(\{V_n\}_n)$ is a finitely generated $F[t]$-module**. By definition we know that $\psi_{n,n+1}$ is an isomorphism for large enough n, say for $n \geq N$ and that V_n is finite-dimensional for all n. Define now

$$W = \bigoplus_{n=0}^{N} V_n. \tag{2.5}$$

Note that for $n \geq N$, the structure of $\theta(\{V_n\}_n)$ is determined by the stabilization of $\psi_{n,n+1}$. Moreover, the module $\theta(\{V_n\}_n)$ is generated by the components $V_0, V_1, \ldots, V_N$, since any V_n for $n \geq N$ is reached through stabilization. Therefore, $\theta(\{V_n\}_n)$ is finitely generated as an $F[t]$-module. □

We can now easily translate our classification result in Theorem 2.18. For that we define an $\mathbb{N}$-persistence F-vector space $U(a,b)$ with $0 \leq a \leq b$ by setting

$$\begin{aligned} U(a,b)_t &= 0, \text{ for } t < a \text{ and } t > b, \\ U(a,b)_t &= F, \text{ for } a \leq t \leq b, \\ \psi_{s,t} &= id_F, \text{ for } a \leq s \leq t \leq b. \end{aligned}$$

Proposition 2.21 *A tame $\mathbb{N}$-persistence F-vector space $\{V_n\}_n$ can be decomposed as*

$$\{V_n\}_n \cong \bigoplus_{i=0}^{N} U(a_i, b_i), \tag{2.6}$$

where the a_i are non-negative integers and the b_i are non-negative integers or ∞. This decomposition is unique.

Remark 2.22 *The fact that the decomposition is unique is to be understood in the sense that the collection $\{(a_i, b_i)\}_i$ is unique up to the ordering of the factors.*

Proof (Proof of Proposition 2.21) With Theorem 2.18 and Proposition 2.20, we get that $\theta(\{V_n\}_n)$ decomposes as in Equation (2.4). Here, the free modules, $F[t](i_s)$ correspond to the summands $U(a_i, \infty)$, as they persist infinitely. And the torsion modules $\frac{F[t]}{t^{k_t}}$ correspond to the summands $U(a_i, b_i)$ with finite intervals. We now want to find a way to get from the decomposition of $\theta(\{V_n\}_n)$ to the decomposition of $\{V_n\}_n$. Observe that

1. $F[t](i_s)$ corresponds to a sequence $\{F\}_{n \geq i_s}$, where the vector space exists at all indices $n \geq i_s$.
2. $\frac{F[t]}{t^{k_t}}(j_t)$ corresponds to a sequence $\{F\}_{i_t \leq n \leq i_t + k_t - 1}$, where the vector space only exists in a finite range.

We will now look closer at these two correspondences.

1. $F[t](i_s)$ means that the module has a copy of F in all degrees $n \geq i_s$, with no upper bounds on persistence. So this corresponds to $U(a_j, \infty)$, where $a_j = i_s$ and we have

$$U(a_j, \infty)_n = \begin{cases} F & \text{if } n \geq a, \\ 0 & \text{otherwise.} \end{cases} \tag{2.7}$$

The morphisms $\psi_{s,t}$ are identity maps for $t \geq s \geq a$, reflecting the persistence of this feature.

2. $\frac{F[t]}{t^{k_t}}(j_t)$ means the modules is supported in degrees $j_t \leq n \leq j_t + k$, with no module elements outside this range. So this corresponds to $U(a_i, b_i)$, where $a_i = j_t$, $b_i = j_t + k_t - 1$ and

$$U(a_i, b_i)_n = \begin{cases} F & \text{if } a \leq n \leq b, \\ 0 & \text{otherwise.} \end{cases} \tag{2.8}$$

Combining these, we can get a decomposition of $\{V_n\}_n$ into summands of the form $U(a_i, b_i)$, where a_i is either i_s or j_t and where $b_i = \infty$ for free modules and $b_i = i_t + k_t - 1$ for torsion modules.

For uniqueness we use the uniqueness from Theorem 2.18 and the fact that θ is a functor and therefore preserves the structure. □

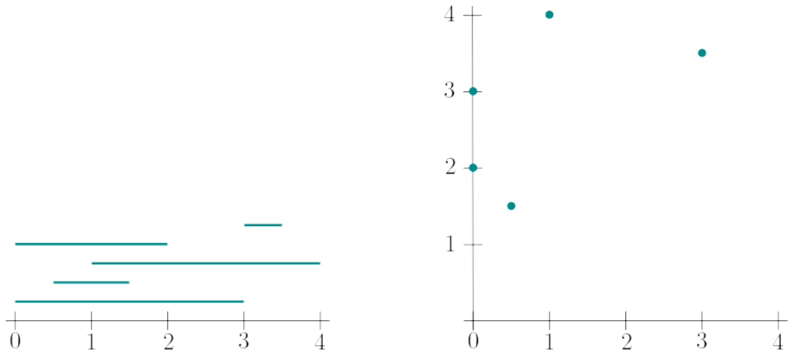


Figure 2.4: At the left we see the barcode and at the right the persistence diagram from Example 2.25.

2.4 Barcodes, Persistence Diagrams and Maximum Persistence

Definition 2.23 *[3] A **barcode** is a multiset of intervals of the form $[i, j)$, where i stands for birth of a homology class at S_i and j stands for death of a homology class at S_j.*

An other way to visualize these birth and death times are the persistence diagrams.

Definition 2.24 *The k-th **persistence diagram** is now generated by plotting every point (i, j), denoting birth at S_i and death at S_j of a homology class, with multiplicity $\mu_k^{i,j}$. We name this diagram* $\mathrm{dgm}(\mathbf{k})$.

Example 2.25 *For a simplicial complex with birth times of homology classes at $\{0, 0, 0.5, 1, 3\}$ and corresponding death times $\{3, 2, 1.5, 4, 3.5\}$ the barcode and persistence diagram are displayed in Figure 2.4.*

Note that in the diagram, all persistent classes are represented by dots, where the dots denote the birth and death times of topological features. To simplify notation we will write dgm instead of $\mathrm{dgm}(k)$. To every diagram, we adjoin the diagonal $\Delta = \{(x, x) | x \geq 0\}$ and endow each point on the diagonal with countable multiplicity.

Definition 2.26 *The **persistence** is the vertical distance from a point to the diagonal.*

To actually compute and plot these persistence diagrams there is a function called ”ripsDiag” in the R-TDA package. More on that can be found in a paper by Fasy et al. [12].

When discussing persistent homology it is interesting to look at the lifetime of the most persistent feature in a dataset. Therefore, we define a notion of the maximum persistence of some persistence diagram.

Definition 2.27 *Let* $(x, y) \in \mathrm{dgm}$ *and define*

$$\mathrm{pers}(x, y) = \begin{cases} y - x & \textit{for } (x, y) \in \mathbb{R}^2, \\ \infty & \textit{otherwise.} \end{cases} \tag{2.9}$$

Then we define the ***maximum persistence of dgm*** *to be*

$$mp(\mathrm{dgm}) = \max_{\mathbf{x} \in \mathrm{dgm}} \mathrm{pers}(\mathbf{x}). \tag{2.10}$$

Definition 2.28 *For two persistence diagrams* dgm_1 *and* dgm_2 *the* ***Bottleneck distance*** $d_B(\mathrm{dgm}_1, \mathrm{dgm}_2)$ *is defined as*

$$d_B(\mathrm{dgm}_1, \mathrm{dgm}_2) = \inf_f \{ \sup_{x \in \mathrm{dgm}_1} \{ \|x - f(x)\|_\infty \} \}, \tag{2.11}$$

where f *ranges over all bijections from* dgm_1 *to* dgm_2*.*

Example 2.29 *Let* $\mathrm{dgm}_1 = \{(1, 3), (2, 5)\}$ *and* $\mathrm{dgm}_2 = \{(1.2, 3.1), (2.5, 4.8)\}$ *be two persistence diagrams. Each point represents the birth and death of a feature in persistent homology. Pairing the closest points yields*

- $(1, 3)$ *pairs with* $(1.2, 3.1)$*, so the distance is* $\max(|1 - 1.2|, |3 - 3.1|) = 0.2$.
- $(2, 5)$ *pairs with* $(2.5, 4.8)$*, so the distance is* $\max(|2 - 2.5|, |5 - 4.8|) = 0.5$.

Taking the maximum over all pairs, we get $d_B(\mathrm{dgm}_1, \mathrm{dgm}_2) = 0.5$*. This example is illustrated in Figure 2.5.*

Lemma 2.30 *Let* dgm_Δ *denote the diagram, where we only have the diagonal* $\Delta = \{(c, c) | c \geq 0\}$ *in the diagram and where every point is endowed with countable multiplicity. Then we have that*

$$mp(\mathrm{dgm}) = 2 d_B(\mathrm{dgm}, \mathrm{dgm}_\Delta). \tag{2.12}$$

Proof First we prove that $mp(\mathrm{dgm}) \leq 2 d_B(\mathrm{dgm}, \mathrm{dgm}_\Delta)$. For that, we let $\phi \colon \mathrm{dgm} \to \mathrm{dgm}_\Delta$ be any bijection and let $x \in \mathrm{dgm}$. Then the following holds

$$\begin{aligned} \|\mathbf{x} - \phi(\mathbf{x})\|_\infty &= \max_{c \in \mathbb{R} \text{ with } \phi(\mathbf{x}) = (c,c)} \{ |x_1 - c|, |x_2 - c| \} \\ &= |x_1 - c| \geq \frac{1}{2} |x_1 - x_2| = \frac{1}{2} \mathrm{pers}(\mathbf{x}), \end{aligned}$$

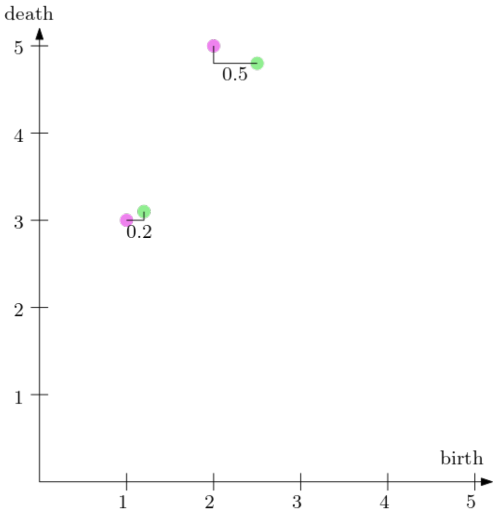


Figure 2.5: The violet dots represent dgm_1 and the green dots represent dgm_2 from Example 2.29.

where $\mathbf{x} = (x_1, x_2)$ and $\phi(\mathbf{x}) = (c, c)$, because it lies on the diagonal and we can assume without loss of generality that $|x_1 - c| \geq |x_2 - c|$. We have an equality in the case where $\phi(x_1, x_2) = \left(\frac{x_1+x_2}{2}, \frac{x_1+x_2}{2}\right)$. Thus

$$\max_{\mathbf{x} \in \text{dgm}} \|\mathbf{x} - \phi(\mathbf{x})\| \geq \frac{1}{2} mp(\text{dgm})$$

and therefore

$$d_B(\text{dgm}, \text{dgm}_\Delta) = \min_{\phi} \max_{\mathbf{x} \in \text{dgm}} \|\mathbf{x} - \phi(\mathbf{x})\| \geq \frac{1}{2} mp(\text{dgm}).$$

For the second inequality, see that the map

$$(x_1, x_2) \mapsto \left(\frac{x_1 + x_2}{2}, \frac{x_1 + x_2}{2}\right)$$

extends to a bijection of multisets $\phi_0 \colon \text{dgm} \to \text{dgm}_\Delta$, such that for all $\mathbf{x} \in \text{dgm}$ we have $\|\mathbf{x} - \phi_0(\mathbf{x})\|_\infty = \frac{1}{2} \text{pers}(\mathbf{x})$. □

2.5 Stability Theorem, Completeness and Separability

This section discusses the robustness of persistent homology under small perturbations. The key result is the Stability Theorem, which states that small changes in input data lead to small changes in the persistence diagram.

To state this result on stability, we need the definitions of some distances. For this part, we follow an article by Cohen-Steiner, Edelsbrunner and Harer [9] and one by Carlsson [6].

Definition 2.31 *Let Z be any metric space and X and Y two compact subsets of Z. The **Hausdorff distance** $d_H(X,Y)$ between X and Y is defined as follows.*

$$d_H(X,Y) = \max\{\sup_{x\in X}(\inf_{y\in Y}(d_Z(x,y))), \sup_{y\in Y}(\inf_{x\in X}(d_Z(x,y)))\}, \tag{2.13}$$

Example 2.32 *Let us take two sets in the Euclidean plane, $X = \{(0,0),(1,1)\}$ and $Y = \{(0,1),(1,2)\}$. We compute $\inf_{y\in Y} d(x,y) = 1$ for $(0,0)$ and $(1,1)$ and similar for $\inf_{x\in X}$. Therefore $d_H(X,Y) = 1$.*

Definition 2.33 *Let now X and Y be any two metric spaces and let $I_{X,Y}$ be the family of all simultaneous isometric embeddings of X and Y. An element of $I_{X,Y}$ is of the form (Z, i_X, i_Y) for a metric space Z and isometric embeddings $i_X\colon X \hookrightarrow Z$, $i_Y\colon Y \hookrightarrow Z$. The **Gromov-Hausdorff distance** $d_{GH}(X,Y)$ is defined by*

$$d_{GH}(X,Y) = \inf_{I_{X,Y}}\{d_H(i_X(X), i_Y(Y)\}. \tag{2.14}$$

Example 2.34 *The Gromov-Hausdorff distance compares metric spaces, not just subsets of a common space. It finds the smallest Hausdorff distance after embedding both spaces into a larger space. Consider the two metric spaces $X = \{0,2\}$ and $Y = \{a,b\}$ with Euclidean distance and where $d(a,b) = 3$. If we now embed both into a common space, the real line $\mathbb{R}$, the best embedding could be such that $i_X(0) = 0$, $i_X(2) = 2$, $i_Y(a) = 0.5$ and $i_Y(b) = 3.5$. Then we have $d_{GH}(X,Y) = 1$.*

Theorem 2.35 (Stability Theorem) *[7] Let X, Y be two point clouds in the same Euclidean space. Then*

$$d_B(dgm(X), dgm(Y)) \leq 2d_{GH}(X,Y) \leq d_H(X,Y). \tag{2.15}$$

For the rest of this chapter, we follow a paper by Mileyko [20] and one by Blumberg [3].

Definition 2.36 *The degree p **total persistence** of a persistence diagram* dgm *is defined as*

$$\text{Pers}_p(\text{dgm}) = 2^p \sum_{x\in\text{dgm}} (\text{pers}(x))^p. \tag{2.16}$$

*We then define the **space of persistence diagrams** to be*

$$D_p = \{\text{dgm} \mid \text{Pers}_p(\text{dgm}) < \infty\}. \tag{2.17}$$

*Define the **Wasserstein metric***

$$W_p(\text{dgm}, \text{dgm}_\Delta)^p = \sum_{x\in\text{dgm}} \text{pers}(x)^p. \tag{2.18}$$

Recall that a barcode $\{I_\alpha\}$ is a multiset of intervals. We now want to define a metric on the set of barcodes.

Definition 2.37 *Let $I_1 = [a_1, b_1)$ and $I_2 = [a_2, b_2)$ be two non-empty intervals in the barcode. We define the* ***Bottleneck distance*** *between them to be*

$$d_\infty(I_1, I_2) = \|(a_1, b_1) - (a_2, b_2)\|_\infty = \max(|a_1 - a_2|, |b_1 - b_2|). \tag{2.19}$$

To be able to prove some convergence results on persistence diagrams in Chapter 7 we have to ensure that the space of persistence diagrams behaves well as a metric space, namely that it is complete and separable with respect to the Wasserstein and Bottleneck distance. We need both of these distances, because

- Working with the Bottleneck distance ensures theoretical stability, but it is not accurate enough for detailed comparisons.
- And working with the Wasserstein distance ensures that we can do precise comparisons of persistence diagrams, but it has not as strong a stability guarantee as the Bottleneck distance.

We have the following proposition with the Definition 2.36:

Proposition 2.38 *[20] D_p is complete in the metric W_p.*

The set of barcodes $\mathcal{B}$ is the set of multi-sets of intervals A such that $|A| < \infty$. We now let $\mathcal{B}_N$ denote the set of multi-sets of intervals A with $|A| \leq N$ for $N \geq 0$. These sets are complete and separable under the Bottleneck metric, whereas the space $\mathcal{B} = \bigcup \mathcal{B}_N$ is not complete under the Bottleneck metric. We therefore introduce a new set.

Definition 2.39 *Let $\overline{\mathcal{B}}$ be the space of multi-sets A of intervals with the property that the cardinality of the multi-sets of A of those intervals of length more than ε has finite cardinality, for all $\varepsilon > 0$.*

We can now state the following theorem:

Theorem 2.40 *$\overline{\mathcal{B}}$ is the completion of $\mathcal{B} = \bigcup \mathcal{B}_N$ in the Bottleneck metric. In particular it is complete and separable.*

This theorem establishes that the space of persistence diagrams can be completed by allowing diagrams with countably many points of at most countable multiplicity, satisfying a natural finiteness condition.

Chapter 3

Time Series and their Transformation

In this chapter we will define the notion of time series and we will look at how we can transform time series data into point clouds, so that it can be analyzed using persistent homology.

3.1 Time Series Data

For the background in time series, we follow a book by Brockwell and Davis [4].

Definition 3.1 *Let $\mathbb{R}^n$ be an Euclidean space. A **dynamical system** in $\mathbb{R}^n$ is a continuous $\phi\colon \mathbb{R}^n \to \mathbb{R}^n$ (for discrete time) or a vector field X on $\mathbb{R}^n$ (for continuous time).*

Example 3.2 *An example of a dynamical system is the Lotka-Volterra system that can take, for example, the following form:*

$$\begin{aligned} \frac{dx}{dt} &= 0.8x - 0.4xy \\ \frac{dy}{dt} &= -y + 0.6xy. \end{aligned}$$

We will now look into why this is in fact a dynamical system by defining the vector field X on $\mathbb{R}^2$, because the Lotka-Volterra is continuous valued. The vector field X is given by

$$X(x,y) = (0.8x - 0.4xy, -y + 0.6xy),$$

where x and y are the populations of prey and predator species. This vector field assigns to each point (x,y) a velocity, describing how the populations evolve over time according to the given differential equations.

Definition 3.3 *A **numerical time series** is a set of observations x_t, where every x_t is recorded at a specific time t from a dynamical system.*

Example 3.4 *An example for a numerical time series is the measured temperature over a day. Say, we measure the temperature every hour. Then we have* $T = 24$ *and our time series is of the form* $\{x_0 = 5, x_1 = 4, x_2 = 2, \dots, x_7 = 4, \dots, x_{12} = 13, \dots, x_{15} = 18, \dots, x_{18} = 15, \dots, x_{21} = 11, \dots, x_{24} = 7\}$. *Here the index represents the time, on which the temperature was measured.*

Definition 3.5 *A time series is called* ***continuous*** *if the observations are recorded continuously over some time interval and* ***discrete*** *if the set of times is discrete.*

In real world cases, we always have a finite number of observations and therefore the time series can be written as

$$\{x_t | t = 0, \dots, T\}. \tag{3.1}$$

But this can not be analyzed with persistent homology directly, so we need a way to transform this data into point clouds. Before we motivate, how we can transform our data, we want to look at an other form of time series and then make a concrete example.

Definition 3.6 ***Categorical-valued time series*** *are characterized by taking values on a qualitative range consisting of a finite number of categories, which is referred to as* ***ordinal range****, if the categories exhibit a natural ordering, or* ***nominal range****, otherwise.[29]*

Example 3.7 *(Chen et al. [8]) We want to study activity-travel data. There are three possibilities for an individual's travel activity; 1) activity at home, 2) activity outside of home and 3) traveling. For a total of N individuals and an observation period of T minutes, we create a categorical-valued time series with three categories as follows. Let* $n = 1, \dots, N$ *be the individuals and* $t = 1, \dots, T$ *be the minutes, we define*

$$x_{n,t} = \begin{cases} 0 & \textit{if respondent is at Home} \\ 1 & \textit{if respondent is on Travel} \\ 2 & \textit{if respondent is Out of Home} \end{cases}. \tag{3.2}$$

The survey was taken during several years and in Figure 3.1 we can see the proportions of the three categories measured. The survey was taken over a time period of 24 *hours, therefore the Home-category has the highest proportion in the beginning and end. We can also look at the activity-travel profile of some individuals during the day, see Figure 3.2. The x-axis shows the time in minutes and the y-axis shows the three different categories. For example, the corresponding time series for individual* 3 *would be* $x_{3,t} = \{0, \dots, 0, 3, \dots, 3, 0, \dots, 0\}$, *where* 3 *first appears approximately at* $t = 970$.

In the remainder of this thesis we will work with numerical time series.

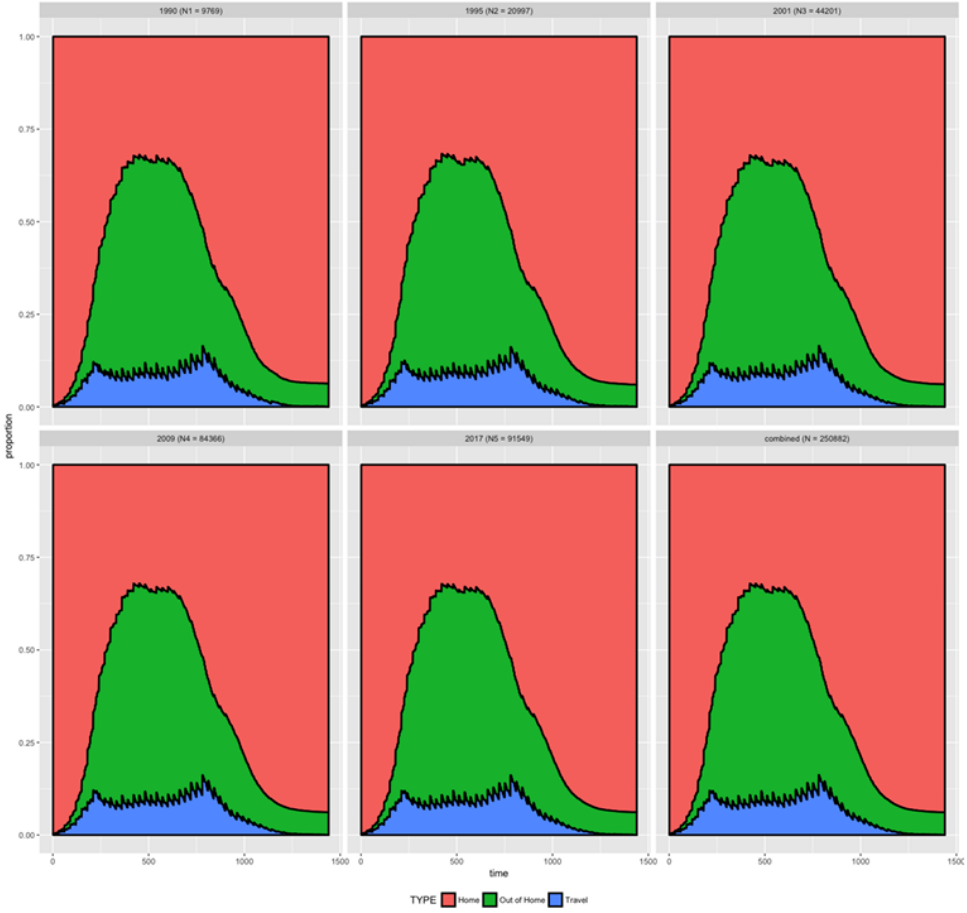


Figure 3.1: Three categories of the activity-travel survey.[8]

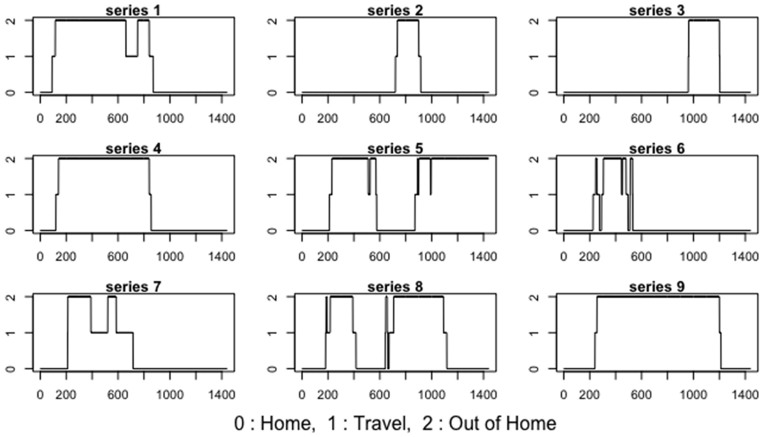


Figure 3.2: Activity-travel of nine individuals.[8]

3.2 Transformation of Data

3.2.1 Taken's Embedding

One way to transform numerical time series data into point clouds in Euclidean space, is the Taken Embedding Theorem [27]. We will present another embedding with sliding windows in the next chapter.

Theorem 3.8 (Taken's Embedding Theorem) *Let M be a compact manifold of dimension m. For pairs (ϕ, y), where $\phi\colon M \to M$ is a smooth (at least C^2) diffeomorphism and $y\colon M \to \mathbb{R}$ a smooth function, called* ***observable****, we have the following generic property: The map $\Phi_{\phi,y}\colon M \to \mathbb{R}^{2m+1}$, defined by*

$$\Phi_{\phi,y}(x) = (y(x), y(\phi(x)), \ldots, y(\phi^{2m}(x))) \tag{3.3}$$

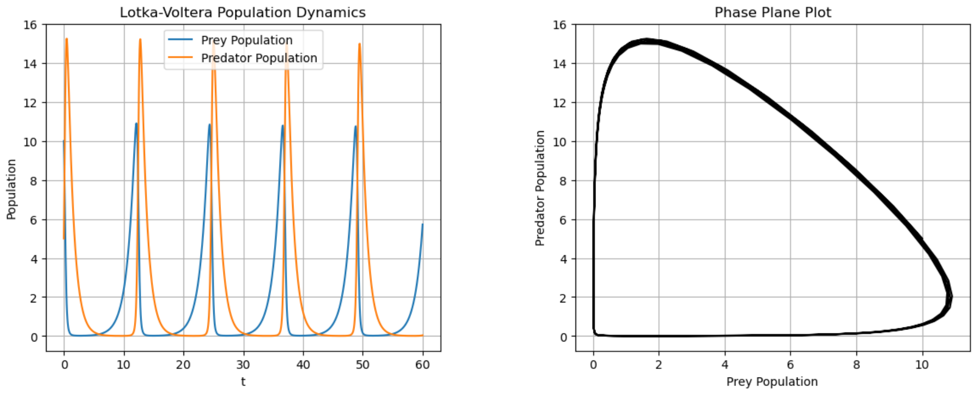
is an embedding.

So let us look again at our time series $\{x_t | t = 1, \ldots T\}$. Taken's embedding transforms this data into a point cloud consisting of the points $\mathbf{v}_i = (x_i, x_{i+\tau}, \ldots, x_{i+(d-1)\tau})$. The parameter d is the dimension of the points in our cloud and τ is the delay parameter. There are several approaches to determine τ and d.

Example 3.9 *We want to make an example of a dynamical system and apply the Taken's embedding to it. Therefore we take the following Lotka-Volterra system, already used in Example 3.2,*

$$\frac{dx}{dt} = 0.8x - 0.4xy$$
$$\frac{dy}{dt} = -y + 0.6xy.$$

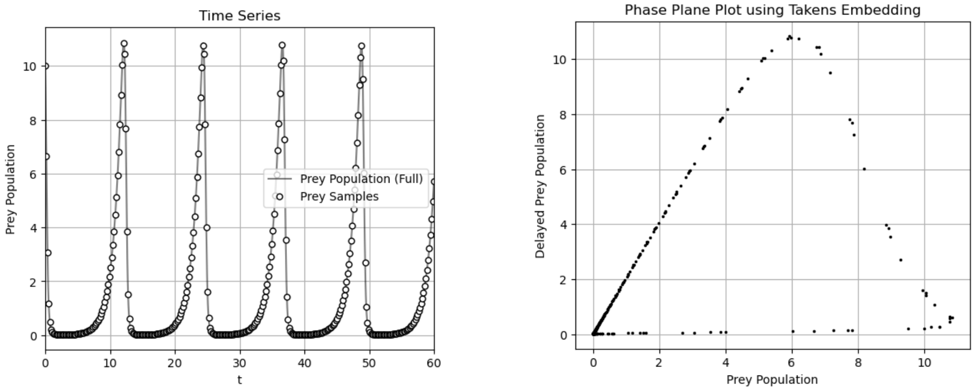
Note that here, x represents the population of a prey species and y represents the one of a predator species. We assume, that in the beginning there are 10 individuals of the prey species and 5 individuals of the predator species. The evolution of the population of this system is modeled in Figure 3.3a, the phase-plane plot in Figure 3.3b. In steps of distance 3, we take samples from the data of the prey species population to construct a time series, displayed in Figure 3.4a. From there we apply the Taken's embedding and get the new phase plane in Figure 3.4b. With the risper package in python, we compute a Vietoris-Rips filtration and plot the persistence diagrams, see Figure 3.5. The code for all the diagrams in this example can be found in the Appendix, Figure A.3.



(a) Population of prey and predator species over time.

(b) Phase-plane plot of the Lotka-Volterra system.

Figure 3.3: Plots of the raw system of Example 3.9.



(a) Time series sampled from the prey population.

(b) Phase plane plot after applying Taken's embedding with parameters $d = 2$ and $\tau = 5$.

Figure 3.4: Time series and its Phase plane plot, gained from the prey population of 3.9.

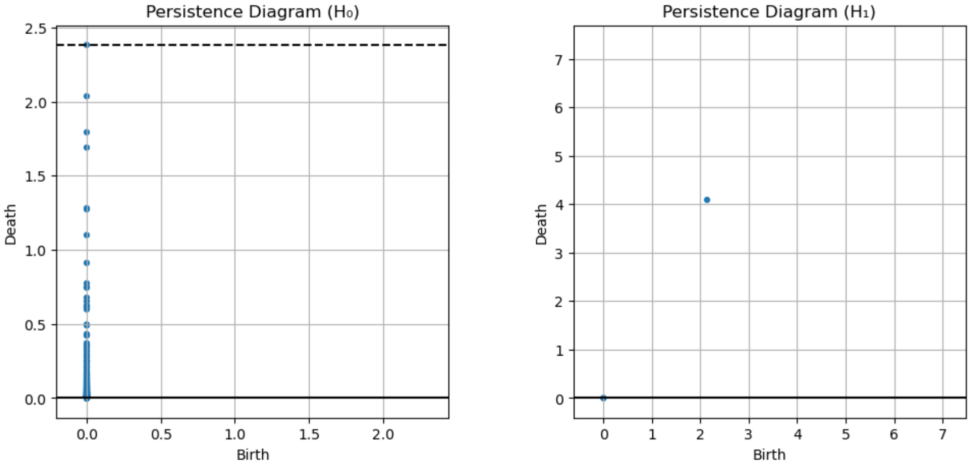


Figure 3.5: Persistence diagrams in dimension 0 and 1 of the phase plane after applying Taken's embedding.

Chapter 4

The Sliding Window Embedding

Using Taken's Theorem is one way to convert time series data into a point cloud. Another way is using the Sliding Windows and 1-Persistence Scoring (SW1PerS) method as introduced in a paper by Perea et al. [23], which we will follow in this section.

4.1 Sliding Window Embedding

This method converts time series into a point cloud (a high-dimensional set of points) and interprets the periodicity of the time series as "circularity" of the point cloud.

Let g be a function defined on some interval $I \subseteq \mathbb{R}$. Now we choose some integer M and a real number τ, both greater than 0.

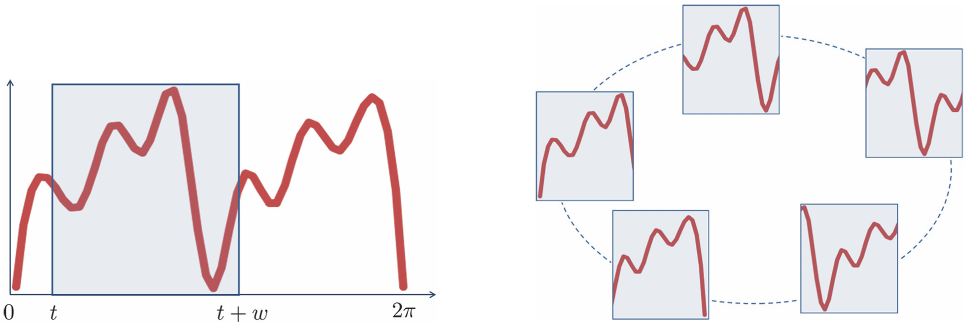
Definition 4.1 *The* ***sliding window embedding*** *of g at $t \in \mathbb{R}$ into $\mathbb{R}^{M+1}$ is defined as follows*

$$SW_{M,\tau}g(t) = \begin{bmatrix} g(t) \\ g(t+\tau) \\ \vdots \\ g(t+M\tau) \end{bmatrix} \tag{4.1}$$

Definition 4.2 *The* ***sliding window point cloud*** *for g is the collection of several sliding window embeddings for different values of t. The parameter $M\tau$ is called the* ***window size****.*

Let $\{x_t | t = 0, \ldots, T\}$ be a given, denoised time series. By mapping the interval $[0, T]$ linearly to $[0, 2\pi]$ and applying an interpolation technique called cubic splining [28], we get some continuous function

$$g\colon [0, 2\pi] \to \mathbb{R}, \text{ such that } g(0) = x_0 \text{ and } g(2\pi) = x_T. \tag{4.2}$$



(a) g along with a window of length w starting at $t \in [0, 2\pi - w]$

(b) Arrangements of snippets according to similarity

Figure 4.1: Example sliding windows [23].

For a fixed window size $0 < w < 2\pi$ and for every time $t \in [0, 2\pi - w]$, we look at the plot of the function g when restricted to $[t, t + w]$. To understand the sliding process more profoundly, we will look at the same example, displayed in Figure 4.1 as used in our reference paper by Perea et al. [23]. In the Figure 4.1a we see a prototypical function g with a window of length w.

Definition 4.3 ***Sliding*** *this window means that we let t go from* 0 *to* $2\pi - w$*.*

This yields a snippet of g for each $t \in [0, 2\pi - w]$. If we now arrange these snippets according to their degree of similarity, then we get something as in Figure 4.1b. The degree of similarity between two windows is the percentage, where they coincide. If a pattern in the plot is repeated, then it corresponds to one circle in the similarity arrangement. The distinctiveness of the pattern is pictured by the size of the hole in the middle of the arrangement.

Definition 4.4 *The* ***period length*** *of a function is the smallest parameter ℓ, such that the pattern of the functions repeats when shifted by ℓ.*

Example 4.5 *Let* $g(t) = \sin(nt)$*, where* $n \in \mathbb{N}$*. By definition of the sliding window, we get*

$$\begin{aligned} SW_{M,\tau} \sin(nt) &= \begin{bmatrix} \sin(nt) \\ \sin(nt + n\tau) \\ \vdots \\ \sin(nt + nM\tau) \end{bmatrix} \\ &= \begin{bmatrix} \sin(nt) \\ \cos(n\tau)\sin(nt) + \cos(nt)\sin(n\tau) \\ \vdots \\ \cos(nM\tau)\sin(nt) + \cos(nt)\sin(nM\tau) \end{bmatrix} \end{aligned} \tag{4.3}$$

$$
\begin{aligned}
&= \sin(nt)\begin{bmatrix}1\\ \cos(n\tau)\\ \vdots\\ \cos(nM\tau)\end{bmatrix} + \cos(nt)\begin{bmatrix}0\\ \sin(n\tau)\\ \vdots\\ \sin(nM\tau)\end{bmatrix}\\
&= \sin(nt)\mathbf{u} + \cos(nt)\mathbf{v}.
\end{aligned}
$$

Here we used that $\sin(x+y) = \cos(y)\sin(x) + \cos(x)\sin(y)$ *and denoted*

$$
\mathbf{u} = \begin{bmatrix}1\\ \cos(n\tau)\\ \vdots\\ \cos(nM\tau)\end{bmatrix} \textit{ and } \mathbf{v} = \begin{bmatrix}0\\ \sin(n\tau)\\ \vdots\\ \sin(nM\tau)\end{bmatrix}.
$$

Notice that the map $t \mapsto SW_{M,\tau}\sin(nt)$ represents a planar curve in $\mathbb{R}^{M+1}$ with winding number n, when the vectors $\mathbf{u}$ and $\mathbf{v}$ are linearly independent. We want to now look at the way the shape of this curve changes when changing the parameters M, n and τ. Changes in the curve can be measured through the eigenvalues of the shape matrix:

$$
A = \begin{bmatrix}||\mathbf{u}||^2 & \langle\mathbf{u},\mathbf{v}\rangle\\ \langle\mathbf{u},\mathbf{v}\rangle & ||\mathbf{v}||^2\end{bmatrix}. \tag{4.4}
$$

The eigenvalues of A determine the axes lengths of the ellipse traced by the sliding window embedding. We compute the components using again the identity above and the fact that $\cos(x)^2 - \sin(x)^2 = cos(2x)$*.*

$$
\begin{aligned}
\langle\mathbf{u},\mathbf{v}\rangle &= \sum_{m=0}^{M}\cos(nm\tau)\sin(nm\tau) = \frac{1}{2}\sum_{m=0}^{M}\sin(2nm\tau)\\
&= \frac{\sin((M+1)n\tau)\sin(nM\tau)}{2\sin(n\tau)}\\
||\mathbf{u}||^2 - ||\mathbf{v}||^2 &= \sum_{m=0}^{M}(\cos(nm\tau)^2 - \sin(nm\tau)^2) = \sum_{m=0}^{M}\cos(2nm\tau)\\
&= \frac{\sin((M+1)n\tau)\cos(nM\tau)}{\sin(n\tau)}.
\end{aligned} \tag{4.5}
$$

Finally with the identity $\cos(x)^2 + \sin(x)^2 = 1$ *we get*

$$
||\mathbf{u}||^2 + ||\mathbf{v}||^2 = \sum_{m=0}^{M}(\cos(nm\tau)^2 + \sin(nm\tau)^2) = \sum_{m=0}^{M}1 = M+1.
$$

From these calculation, we see directly that A is positive semi-definite. So the eigenvalues of A are non-negative real numbers λ_1, λ_2 and there exists a 2×2 orthogonal matrix B such that

$$
A = B^T\Lambda^2 B, \textit{ where } \Lambda = \begin{bmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{bmatrix} \textit{ and } B = \begin{bmatrix}\cos(\phi) & -\sin(\phi)\\ \sin(\phi) & \cos(\phi)\end{bmatrix}.
$$

We set $\mathbf{x}(t) = \begin{bmatrix}\sin(nt) & \cos(nt)\end{bmatrix}^T$ *and calculate*

$$\begin{aligned}\|SW_{M,\tau}\sin(nt)\|^2 &= \left\| \begin{bmatrix} | & | \\ \mathbf{u} & \mathbf{v} \\ | & | \end{bmatrix} \mathbf{x}(t) \right\|^2 \\ &= \|\mathbf{u}\|^2 \sin(nt)^2 + 2\langle \mathbf{u}, \mathbf{v}\rangle \cos(nt)\sin(nt) + \|\mathbf{v}\|^2 \cos(nt)^2 \\ &= \langle \mathbf{x}(t), A\mathbf{x}(t)\rangle \\ &= \langle \mathbf{x}(t), B^T\Lambda^2 B\mathbf{x}(t)\rangle \\ &= \langle \Lambda B\mathbf{x}(t), \Lambda B\mathbf{x}(t)\rangle. \end{aligned} \tag{4.6}$$

In summary, this step provides a bridge between the algebraic manipulation of the sliding window embedding and its geometric interpretation as an ellipse. This insight is essential for analyzing how the embedding behaves as the parameters M, n and τ change. With ϕ the angle by which the matrix B rotates, we see that the map

$$SW_{M,\tau}\sin(nt) \mapsto \begin{bmatrix}\sqrt{\lambda_1}\sin(nt+\phi) \\ \sqrt{\lambda_2}\cos(nt+\phi)\end{bmatrix} \tag{4.7}$$

is an isometry. So our sliding window embedding of $g(t) = \sin(nt)$ is an ellipse on the plane spanned by $\mathbf{u}$ and $\mathbf{v}$. The exact shape of the ellipse is mainly determined by the square root of the eigenvalues of A. With the values from above, we can compute these to be

$$\lambda_{1,2} = \frac{M+1 \pm \left|\frac{\sin((M+1)n\tau)}{\sin(n\tau)}\right|}{2}. \tag{4.8}$$

The ellipse is most similar to a cycle when λ_2 is maximal. This is the case if and only if $(M+1)n\tau \equiv 0 \bmod \pi$. This is the case for example for $M\tau = (\frac{M}{M+1})\frac{2\pi}{n}$. Here, the window size comes close to the length of the period $\frac{2\pi}{n}$ of $g(t)$. So the sliding window point cloud for $g(t) = \sin(nt)$ is the roundest when the window size is close to the frequency.

Example 4.6 *We will now look at an example for explicit values of n, M and τ, where we also show how we compute the eigenvalues of A. Let us take $n = 2$, $M = 2$ and $\tau = 1$. So our window size is 2. By plugging in these values, we see from Equation (4.3) that*

$$\mathbf{u} = \begin{bmatrix} 1 \\ \cos(2) \\ \cos(4) \end{bmatrix}, \mathbf{v} = \begin{bmatrix} 0 \\ \sin(2) \\ \sin(4) \end{bmatrix}. \tag{4.9}$$

So to figure out, what the values of the matrix A are, we compute the following

$$\begin{aligned}
\|\mathbf{u}\|^2 + \|\mathbf{v}\|^2 &= M+1 = 3, \\
\|\mathbf{u}\|^2 - \|\mathbf{v}\|^2 &= \frac{\sin((M+1)n\tau)\cos(nM\tau)}{\sin(n\tau)} \\
&= \frac{\sin((2+1)\cdot 2\cdot 1)\cos(2\cdot 2\cdot 1)}{\sin(2\cdot 1)} = \frac{\sin(6)\cos(4)}{\sin(2)}, \qquad (4.10) \\
\langle \mathbf{u}, \mathbf{v}\rangle &= \frac{\sin((M+1)n\tau)\sin(nM\tau)}{2\sin(n\tau)} \\
&= \frac{\sin((2+1)\cdot 2\cdot 1)\sin(2\cdot 2\cdot 1)}{2\sin(2\cdot 1)} = \frac{\sin(6)\sin(4)}{2\sin(2)}.
\end{aligned}$$

By computing $\|\mathbf{u}\|^2 = \frac{1}{2}((\|\mathbf{u}\|^2 + \|\mathbf{v}\|^2) + (\|\mathbf{u}\|^2 - \|\mathbf{v}\|^2))$ *and also computing* $\|\mathbf{v}\|^2 = \frac{1}{2}((\|\mathbf{u}\|^2 + \|\mathbf{v}\|^2) - (\|\mathbf{u}\|^2 - \|\mathbf{v}\|^2))$*, we get*

$$\begin{aligned}
\|\mathbf{u}\|^2 &= \frac{3 + \frac{\sin(6)\cos(4)}{\sin(2)}}{2}, \\
\|\mathbf{v}\|^2 &= \frac{3 - \frac{\sin(6)\cos(4)}{\sin(2)}}{2}.
\end{aligned} \qquad (4.11)$$

Now onto the determinant of A, $\det(A) = \|\mathbf{u}\|^2\|\mathbf{v}\|^2 - \langle \mathbf{u}, \mathbf{v}\rangle^2$*. By plugging in the results from above, we get*

$$\det(A) = \frac{3 + \frac{\sin(6)\cos(4)}{\sin(2)}}{2} \cdot \frac{3 - \frac{\sin(6)\cos(4)}{\sin(2)}}{2} - \left(\frac{\sin(6)\sin(4)}{2\sin(2)}\right)^2. \qquad (4.12)$$

Now using the fact that $(a+b)(a-b) = a^2 - b^2$*, we have*

$$\frac{9 - \left(\frac{\sin(6)\cos(4)}{\sin(2)}\right)^2}{4} - \frac{\sin(6)^2\sin(4)^2}{4\sin(2)^2}. \qquad (4.13)$$

We are now ready to compute the eigenvalues:

$$\lambda_{1,2} = \frac{\mathrm{Tr}(A) \pm \sqrt{\mathrm{Tr}(A)^2 - 4\det(A)}}{2}. \qquad (4.14)$$

First we compute the trace of A.

$$\mathrm{Tr}(A) = \|\mathbf{u}\|^2 + \|\mathbf{v}\|^2 = M+1 = 3. \qquad (4.15)$$

We simplify:

$$\begin{aligned}
\mathrm{Tr}(A)^2 - 4\det(A) &= 9 - 4\det(A) \\
&= 9 - 4\left(\frac{9 - \left(\frac{\sin(6)\cos(4)}{\sin(2)}\right)^2}{4} - \frac{\sin(6)^2\sin(4)^2}{4\sin(2)^2}\right) \qquad (4.16) \\
&= \left(\frac{\sin(6)\cos(4)}{\sin(2)}\right)^2 + \frac{\sin(6)^2\sin(4)^2}{\sin(2)^2}.
\end{aligned}$$

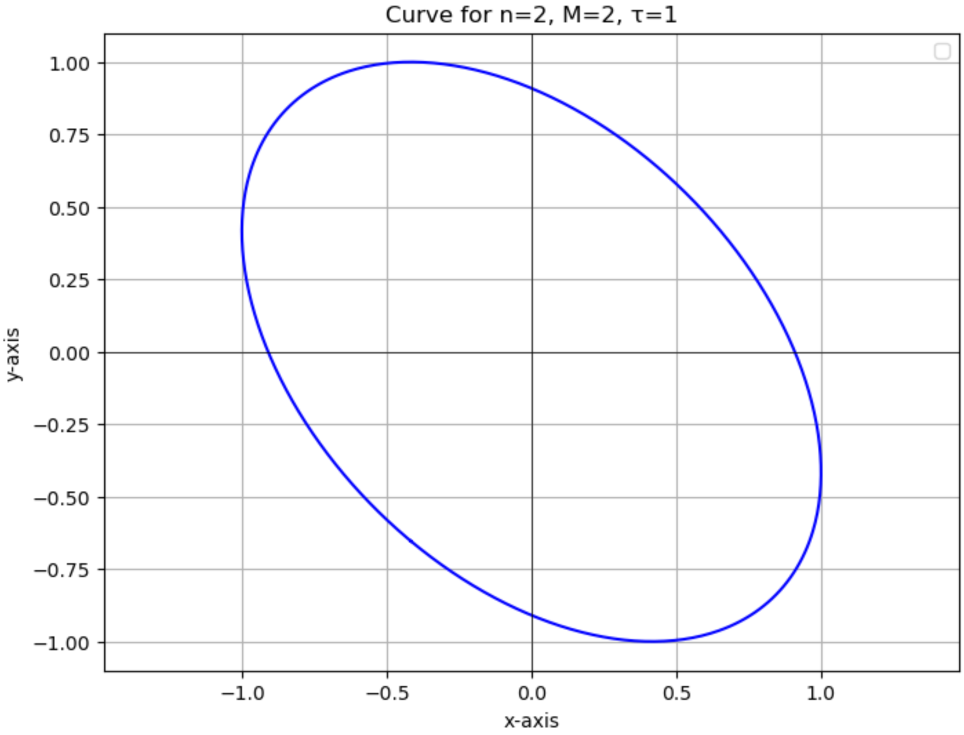


Figure 4.2: The plotted ellipse from Example 4.6. Here the x-axis represents the cosine component and the y-axis the sine component. For the code, see Appendix, Figure A.4.

And therefore we get the following for the eigenvalues

$$\begin{aligned}\lambda_1 &= \frac{3+\sqrt{\left(\frac{\sin(6)\cos(4)}{\sin(2)}\right)^2+\frac{\sin(6)^2\sin(4)^2}{\sin(2)^2}}}{2},\\ \lambda_2 &= \frac{3-\sqrt{\left(\frac{\sin(6)\cos(4)}{\sin(2)}\right)^2+\frac{\sin(6)^2\sin(4)^2}{\sin(2)^2}}}{2}.\end{aligned} \tag{4.17}$$

In the following chapters we explore the relationship between period length and window size $M\tau$. Moreover, the maximal persistence of a 1-dimensional persistence diagram of the Vietoris-Rips filtration of a finite sample is a measure of roundness of the sliding window point cloud.

Chapter 5

The Approximation Theorem

In Example 4.5 we saw that we can compute the sliding window of $\sin(nt)$ relatively easily. The same holds for $\cos(nt)$. We now want to study the sliding window and persistence of any function $f \in L^2(\mathbb{T})$, where $\mathbb{T} = \mathbb{R}/(2\pi\mathbb{Z})$. For that we use the Fourier series approximation of f. Using the Fourier series, we can reformulate $SW_{M,\tau}f$ through $SW_{M,\tau}\sin(nt)$ and $SW_{M,\tau}\cos(nt)$. To do this, we need to show that $SW_{M,\tau}$ behaves well under approximations as with Fourier series and that these approximations work in the framework of stability for persistence diagrams.

In this context we will state and prove an important result called the Approximation Theorem 5.5. It shows that, in the limit, the persistent homology of the point cloud generated by $SW_{M,\tau}f$ for $f \in C^k(\mathbb{T}, \mathbb{R})$ can be effectively analyzed through the persistent homology of its truncated Fourier series.

Let now $C(X, Y)$ be the set of continuous functions $X \to Y$, with the sup norm. The sliding window embedding induces a mapping

$$SW_{M,\tau}\colon C(\mathbb{T}, \mathbb{R}) \to C(\mathbb{T}, \mathbb{R}^{M+1}). \tag{5.1}$$

This map has the following property.

Proposition 5.1 *For all $M \in \mathbb{N}$ and $\tau > 0$, the map $SW_{M,\tau}$ is a bounded linear operator with norm $\|SW_{M,\tau}\| \leq \sqrt{M+1}$.*

Proof 1. **The map SW$_{M,\tau}$ is bounded**: Let $f \in C(\mathbb{T}, \mathbb{R})$ and let $t \in \mathbb{T}$. Then

$$\begin{aligned} \|SW_{M,\tau}f(t)\|^2_{\mathbb{R}^{M+1}} &= |f(t)|^2 + |f(t+\tau)|^2 + \ldots + |f(t+M\tau)|^2 \\ &\leq \sup_t |f(t)|^2 + \sup_t |f(t)|^2 + \ldots + \sup_t |f(t)|^2 \\ &= (M+1)\|f\|^2_\infty. \end{aligned} \tag{5.2}$$

2. **The map SW$_{M,\tau}$ is linear**: Follows directly from the definition of the sliding window embedding. □

Example 5.2 *(Example 4.6 continued) We check the computation in Equation (5.2) for our example $f(t) = \sin(2t)$ with values $M = 2$ and $\tau = 1$. The sliding window is*

$$SW_{2,1}\sin(2t) = \begin{bmatrix} \sin(2t) \\ \sin(2(t+1)) \\ \sin(2(t+2)) \end{bmatrix}. \tag{5.3}$$

So the norm is given by

$$\begin{aligned} \|SW_{2,1}\sin(2t)\|^2_{\mathbb{R}^3} &= |\sin(2t)|^2 + |\sin(2(t+1))|^2 + |\sin(2(t+2))|^2 \\ &\leq 1+1+1 = 3. \end{aligned} \tag{5.4}$$

Where we used the fact that $\sin$ *takes values between* -1 *and* 1*, so the absolute value is smaller equal than* 1.

We now want to look at the Fourier series of some function f and see how the sliding window behaves in this context. Let

$$f(t) = S_N f(t) + R_N f(t), \tag{5.5}$$

where $S_N f(t)$ is the Fourier expansion of f and $R_N f(t)$ is the remainder. The remainder is the difference between $f(t)$ and the Fourier expansion $S_N f(t)$. The Fourier expansion is given by

$$S_N f(t) = \sum_{n=0}^{N} a_n \cos(nt) + b_n \sin(nt) = \sum_{n=-N}^{N} \hat{f}(n) e^{int}, \tag{5.6}$$

where

$$\begin{aligned} a_n &= \frac{1}{\pi}\int_{-\pi}^{\pi} f(t)\cos(nt)dt \\ b_n &= \frac{1}{\pi}\int_{-\pi}^{\pi} f(t)\sin(nt)dt. \end{aligned}$$

Here $\hat{f}(n)$ is given by

$$\hat{f}(n) = \begin{cases} \frac{1}{2}a_n - \frac{i}{2}b_n & \text{if } n > 0, \\ \frac{1}{2}a_{-n} + \frac{i}{2}b_{-n} & \text{if } n < 0, \\ a_0 & \text{if } n = 0. \end{cases} \tag{5.7}$$

We recognize the function $\sin(nt)$ from our Example 4.5 and the function $\cos(nt)$ can be treated similarly. By setting

$$\mathbf{u}_n = SW_{M,\tau}\cos(nt)|_{t=0} \text{ and } \mathbf{v}_n = SW_{M,\tau}\sin(nt)|_{t=0}, \tag{5.8}$$

we can compute the sliding window of $f(t)$ using these notions:

$$\begin{aligned} SW_{M,\tau}f(t) &= SW_{M,\tau}S_N f(t) + SW_{M,\tau}R_N f(t) \\ &= \sum_{n=0}^{N} (a_n SW_{M,\tau}\cos(nt) + b_n SW_{M,\tau}\sin(nt)) + SW_{M,\tau}R_N f(t) \\ &= \sum_{n=o}^{N} (a_n(\cos(nt)\mathbf{u}_n + \sin(nt)\mathbf{v}_n) + b_n(\cos(nt)\mathbf{v}_n - \sin(nt)\mathbf{u}_n)) \\ &= \sum_{n=0}^{N} (\cos(nt)(a_n\mathbf{u}_n + b_n\mathbf{v}_n) + \sin(nt)(b_n\mathbf{u}_n - a_n\mathbf{v}_n)) \\ &\qquad + SW_{M,\tau}R_N f(t). \end{aligned} \tag{5.9}$$

Note that the vectors $\mathbf{u}_n$ and $\mathbf{v}_n$ form a basis. to simplify the notation, we name the Fourier expansion term of the sliding window of $f(t)$,

$$\phi_\tau(t) = \sum_{n=0}^{N} (\cos(nt)(a_n\mathbf{u}_n + b_n\mathbf{v}_n) + \sin(nt)(b_n\mathbf{u}_n - a_n\mathbf{v}_n)). \tag{5.10}$$

If all the variables are clear from the context, we just write $\phi_\tau = SW_{M,\tau}S_N f(t)$.

Example 5.3 *(Example 4.6 continued) We will now look at our example from the previous sections and compute the steps above for $f(t) = \sin(2t)$. First, observe that*

$$S_N f(t) = \sum_{n=-N}^{N} \hat{f}(n)e^{int}, \tag{5.11}$$

where $\hat{f}(n)$ are the Fourier coefficients, that are calculated in the following way for $f(t) = \sin(2t)$:

$$\hat{f}(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \sin(2t)e^{-int}dt. \tag{5.12}$$

As our function is sinusoidal, only the values $n = \pm 2$ yield nonzero Fourier coefficients.

$$\hat{f}(n) = \begin{cases} \frac{i}{2} & \text{if } n = -2, \\ -\frac{i}{2} & \text{if } n = 2, \\ 0 & \text{otherwise.} \end{cases} \tag{5.13}$$

We can now plug these values into the remaining equations above, to see what the sliding window is.

Now that we expressed the Fourier expansion part through well-known functions, we are interested in studying the remainder term $SW_{M,\tau}f(t)$. With the proposition below, we find bounds for each derivative $f^{(k)} = \frac{d^k f}{dt^k}$, whenever the derivatives exist and are continuous.

Proposition 5.4 *Let $k \in \mathbb{N}$. For some $f \in C^k(\mathbb{T}, \mathbb{R})$ and for all $t \in \mathbb{T}$ we have*

$$\|SW_{M,\tau}f(t) - \phi_\tau(t)\|_{\mathbb{R}^{M+1}} \leq \sqrt{4k-2}\left\|R_N f^{(k)}\right\|_2 \cdot \frac{\sqrt{M+1}}{(N+1)^{k-\frac{1}{2}}}. \tag{5.14}$$

Proof Note that by applying integration by parts k-times to $f^{(k)}$, we obtain the following relationship between the Fourier coefficients of a function and its derivatives.

$$|\widehat{f^{(k)}}(n)| = |n|^k |\hat{f}(n)|, \text{ for all } n \in \mathbb{Z}. \tag{5.15}$$

For our computations, we need the Cauchy-Schwarz inequality, Young's inequality and Parseval's Theorem. Let us have a look at them. The **Cauchy-Schwarz inequality** states, that for two sequences a_n and b_n, we have

$$\left(\sum_n a_n b_n\right)^2 \leq \left(\sum_n a_n^2\right)\left(\sum_n b_n^2\right). \tag{5.16}$$

Taking squares leaves us with

$$\sum_n a_n b_n \leq \left(\sum_n a_n^2\right)^{1/2}\left(\sum_n b_n^2\right)^{1/2}. \tag{5.17}$$

Young's inequality states that for p, q with $\frac{1}{p} + \frac{1}{q} = 1$, we have

$$|ab| \leq \frac{a^p}{p} + \frac{b^q}{q}. \tag{5.18}$$

And the **Parseval Theorem** relates the L^2-norm of a function f to the ℓ^2-norm of its Fourier coefficients:

$$\|f\|_2^2 = \sum_{n\in\mathbb{Z}} |\hat{f}(n)|^2. \tag{5.19}$$

Similarly, we get the following for the k-th derivative

$$\|f^{(k)}\|_2^2 = \sum_{n\in\mathbb{Z}} |\widehat{f^{(k)}}(n)|^2. \tag{5.20}$$

We compute

$$\begin{aligned}
|R_N f(t)| &\leq \sum_{n=N+1}^{\infty} \frac{|\widehat{f^{(k)}}(n)| + |\widehat{f^{(k)}}(-n)|}{n^k} \\
&\leq \left(\sum_{n=N+1}^{\infty}\left(|\widehat{f^{(k)}}(n)| + |\widehat{f^{(k)}}(-n)|\right)^2\right)^{1/2} \cdot \left(\sum_{n=N+1}^{\infty}\frac{1}{n^{2k}}\right)^{1/2} \\
&\leq \left(2\sum_{|n|\geq N+1}|\widehat{f^{(k)}}(n)|^2\right)^{1/2} \cdot \left(\int_{N+1}^{\infty}\frac{1}{x^{2k}}dx\right)^{1/2} \\
&= \sqrt{2}\left\|R_N f^{(k)}\right\|_2 \cdot \frac{\sqrt{2k-1}}{(N+1)^{k-\frac{1}{2}}}.
\end{aligned} \tag{5.21}$$

The second line is obtained by applying the Cauchy-Schwarz inequality. For the third line, we simplify the first term using absolute values and convert the second term using Young's inequality implicitly. In the last step, we used Parseval's Theorem and computed the integral from the second term. So with Proposition 5.1 we see that

$$\begin{aligned}\|SW_{M,\tau}f(t) - \phi_\tau(t)\|_{\mathbb{R}^{M+1}} &\leq \sqrt{M+1}\|R_N f\|_\infty \\ &\leq \sqrt{4k-2}\|R_N f^{(k)}\|_2 \cdot \frac{\sqrt{M+1}}{(N+1)^{k-\frac{1}{2}}}. \end{aligned} \tag{5.22}$$

□

All the results from this section can be summarized in the so-called Approximation Theorem, that we will state and prove below.

Theorem 5.5 (Approximation Theorem) *Let $T \subseteq \mathbb{T}$, a function $f \in C^k(\mathbb{T}, \mathbb{R})$, $X = SW_{M,\tau}f(T)$ and $Y = SW_{M,\tau}S_N f(T)$. Then we have the following three statements:*

1. $d_H(X,Y) \leq \sqrt{4k-2}\left\|R_n f^{(k)}\right\|_2 \frac{\sqrt{M+1}}{(N+1)^{k-\frac{1}{2}}}$
2. $|mp(\text{dgm}(X)) - mp(\text{dgm}(Y))| \leq 2d_B(\text{dgm}(X), \text{dgm}(Y))$
3. $d_B(\text{dgm}(X), \text{dgm}(Y)) \leq 2\sqrt{4k-2}\left\|R_n f^{(k)}\right\|_2 \frac{\sqrt{M+1}}{(N+1)^{k-\frac{1}{2}}}$

Proof The bounds we computed above directly lead to estimates for the Hausdorff distance between the sliding window point clouds of f and $S_N f$. Let X be the image of some $T \subseteq \mathbb{T}$ under $SW_{M,\tau}f$ and Y be the image of T under ϕ_τ. If now our $f \in C^k(\mathbb{T}, \mathbb{R}$ and

$$\varepsilon > \sqrt{4k-2}\|R_N f^{(k)}\|_2 \cdot \frac{\sqrt{M+1}}{(N+1)^{k-\frac{1}{2}}}, \tag{5.23}$$

then $X \subseteq Y^\varepsilon$ and $Y \subseteq X^\varepsilon$. And therefore we see that $d_H(X,Y) \leq \varepsilon$. By letting ε go to its lower bound and using the Stability Theorem 2.35 that relates d_H to d_B, we get

$$d_B(\text{dgm}(X), \text{dgm}(Y)) \leq 2\sqrt{4k-2}\|R_N f^{(k)}\|_2 \cdot \frac{\sqrt{M+1}}{(N+1)^{k-\frac{1}{2}}}. \tag{5.24}$$

□

It follows that in the limit we can describe the persistent homology of the sliding window point cloud of a function $f \in C^{(k)}(\mathbb{T}, \mathbb{R})$ in terms of the persistent homology of the truncated Fourier series of f. As outlined in the previous sections, the maximum persistence of $\text{dgm}(X)$ will be used to quantify the periodicity of f when analyzed using sliding windows of length $M\tau$.

Our calculations and results do not only hold for functions in $C^{(k)}(\mathbb{T}, \mathbb{R})$, but also for functions in the set of Hölder continuous functions with an exponent $\alpha \in (\frac{1}{2}, 1)$. For the definition of Hölder continuous function we follow a book by Renato Fiorenza, see [13].

Definition 5.6 *For a real number $\alpha \in (0, 1]$, we say that a function f is a* ***Hölder continuous function*** *with exponent α in the set X, when there exists an $H \geq 0$ such that $|f(x_1) - f(x_2)| \leq H|x_1 - x_2|, \forall x_1, x_2 \in X$.*

The above holds indeed, take such an f and consider the Fejér approximation

$$\sigma_n f(t) = \sum_{|n| \leq N} \left(1 - \frac{|n|}{N+1}\right) \hat{f}(n) e^{int}, \tag{5.25}$$

and consider the following Theorem from [26]:

Theorem 5.7 *Suppose that f is Lipschitz continuous with constant K. Then*

$$\|\sigma_n(f) - f\|_\infty \leq C_1 K \frac{\log(n)}{n}, \tag{5.26}$$

where C is an absolute constant. More generally, we can say that if f satisfies a Hölder condition with $\alpha < 1$ and Hölder constant K_α, then

$$\|\sigma_n(f) - f\|_\infty \leq C_\alpha K_\alpha n^{-\alpha}, \tag{5.27}$$

where C_α is a constant depending solely on α.

We then get a new version of Proposition 5.4:

Proposition 5.8 *(Proposition 5.4) For every $t \in \mathbb{T}$, we have that*

$$\|SW_{M,\tau} f(t) - SW_{M,\tau} \sigma_N f(t)\|_{\mathbb{R}^{M+1}} \leq C_\alpha K_\alpha \frac{\sqrt{M+1}}{N^\alpha}. \tag{5.28}$$

Similarly we can reformulate the Approximation Theorem 5.5.

Chapter 6

The Geometric Structure of $SW_{M,\tau}S_N f$

In this chapter, we look at the geometric structure of the sliding window point cloud when applied to the truncated Fourier series of a periodic function. More precisely we analyze its dependence on τ, N and M.

In most related work, the window size $M\tau$ is estimated using the autocorrelation function, which measures how well a signal correlates with itself at different time shifts. But this approach fails for nonlinear systems [18]. Also M is mostly estimated using the method of false nearest neighbors [17]. The idea behind this method is to identify points in the dataset that are neighbors in the embedding space but should not be, as their future temporal evolution differs significantly. The false nearest neighbors method detects when increasing the embedding dimension reduces false neighbors, meaning that points close in the embedding space remain close in the original system.

In contrast to these approaches, we propose a geometric approach to estimating $M\tau$ and M, based on the Structure Theorem 6.8, which formally describes the geometric properties of the sliding window embedding.

6.1 Dimension of the Sliding Window Embedding

As seen in preceding chapters, the dimension of the sliding windows embedding is $M+1$. One interpretation of the dimension is as a level of detail we hope to capture through the sliding window representation. On one hand, we wish to have as large a dimension as possible, to get more detailed information on the data. But on the other hand, regarding computational costs and the fact that we want to compute the persistent homology of these objects, we do not want the dimension to be too large. Especially because otherwise the size of the Rips complex would be too big for our computational resources and the calculation of the persistent homology would be impossible.

It is crucial to understand how much information is preserved given the computational limitations on the embedding dimension. In our case, with trigonometric polynomials we have the following statement:

Lemma 6.1 *No information is lost if and only if the embedding dimension $M+1$ is greater than twice the maximum frequency.*

Here we see the link to the computations we made in Example 4.5 in Chapter 4.

Proof (of Lemma 6.1) Recall the linear decomposition from the sections above:

$$SW_{M,\tau}S_Nf(t) = \sum_{n=0}^{N} \cos(nt)(a_n\mathbf{u}_n + b_n\mathbf{v}_n) + \sin(nt)(b_n\mathbf{u}_n - a_n\mathbf{v}_n), \tag{6.1}$$

where $\mathbf{u}_n$ and $\mathbf{v}_n$ are given by

$$\mathbf{u}_n = SW_{M,\tau}\cos(nt)|_{t=0}\ , \mathbf{v}_n = SW_{M,\tau}\sin(nt)|_{t=0} \tag{6.2}$$

and the a_n, b_n are defined as in Equation (5.7). In Example 4.5 we saw that the angles between and the norms of $\mathbf{u}_n$ and $\mathbf{v}_n$ can be computed using only M and τ. Therefore S_Nf can be received from $SW_{M,\tau}S_Nf$ if the $\mathbf{u}_n$ and the $\mathbf{v}_n$ are linearly independent. Indeed, if $\mathbf{u}_n$ and $\mathbf{v}_n$ are linearly independent for all n, then we can uniquely determine the coefficients a_n and b_n. This is because for each n, we can solve the system

$$a_n\mathbf{u}_n + b_n\mathbf{v}_n \text{ and } b_n\mathbf{u}_n - a_n\mathbf{v}_n$$

to obtain a_n and b_n. Since $S_nf(t)$ is expressed as

$$S_Nf(T) = \sum_{n=0}^{N} a_n\cos(nt) + b_n\sin(nt),$$

once we determine a_n and b_n, we immediately recover S_Nf. And this is what we mean, when we say, that no information is lost. □

These explanations result in the following Proposition.

Proposition 6.2 *Let $M\tau < 2\pi$. Then we have that $\mathbf{u}_0, \mathbf{u}_1, \mathbf{v}_1, \dots, \mathbf{u}_N, \mathbf{v}_N$ are linearly independent if and only if $M \geq 2N$.*

Example 6.3 *Going back to our Example 4.5 with $M = 2$ and $\tau = 1$, we calculate $\mathbf{u}_0, \mathbf{u}_1, \mathbf{v}_1$, so $N = 1$.*

$$\begin{aligned}\mathbf{u}_0 &= SW_{M,\tau}\cos(0t)|_{t=0} = \cos(0t)\mathbf{u}|_{t=0} - \sin(0t)\mathbf{v}|_{t=0}\\ &= \begin{bmatrix}1\\ \cos(0)\\ \cos(2\cdot 0)\end{bmatrix} - \mathbf{0} = \begin{bmatrix}1\\1\\1\end{bmatrix},\end{aligned}$$

$$\begin{aligned}
\mathbf{u}_1 &= SW_{M,\tau}\cos(t)|_{t=0} = \cos(t)\mathbf{u}|_{t=0} - \sin(t)\mathbf{v}|_{t=0} \\
&= \begin{bmatrix} 1 \\ \cos(1) \\ \cos(2\cdot 1) \end{bmatrix} - \mathbf{0} = \begin{bmatrix} 1 \\ \cos(1) \\ \cos(2) \end{bmatrix},
\end{aligned}$$

$$\begin{aligned}
\mathbf{v}_1 &= SW_{M,\tau}\sin(t)|_{t=0} = \sin(t)\mathbf{u}|_{t=0} + \cos(t)\mathbf{v}|_{t=0} \\
&= \mathbf{0} + \begin{bmatrix} 0 \\ \sin(1) \\ \sin(2\cdot 1) \end{bmatrix} = \begin{bmatrix} 0 \\ \sin(1) \\ \sin(2) \end{bmatrix}.
\end{aligned}$$

These three vectors are linearly independent and $M = 2 \geq 2 * N = 2$ is satisfied.

Proof (of Proposition 6.2) We begin with the **"if" direction** and prove it by contraposition. So we assume that our vectors $\mathbf{u}_0, \mathbf{u}_1, \mathbf{v}_1, \ldots, \mathbf{u}_N, \mathbf{v}_N$ are linearly dependent. We want to show that from that it follows that $2N > M$. According to our assumption scalars $\gamma_0, \delta_0, \ldots, \gamma_N, \delta_N \in \mathbb{R}$ exist, where $\delta_0 = 0$ and not all are 0, such that

$$\gamma_0\mathbf{u}_0 + \gamma_1\mathbf{u}_1 \ldots + \gamma_N\mathbf{u}_N + \delta_N\mathbf{v}_N = \mathbf{0}.$$

With the definitions of $\mathbf{u}$ and $\mathbf{v}$ this translates to

$$0 = \sum_{n=0}^{N} \gamma_n \cos(nm\tau) + \delta_n \sin(nm\tau) = Re\left(\sum_{n=0}^{N}(\gamma_n - i\delta_n)e^{inm\tau}\right).$$

Now let $\xi_m = e^{im\tau}$ and define polynomials

$$\begin{aligned}
p(z) &= \sum_{n=0}^{N}(\gamma_n + i\delta_n)z^n, \\
\bar{p}(z) &= \sum_{n=0}^{N}(\gamma_n - i\delta_n)z^n \text{ and} \\
q(z) &= z^N \cdot \left(\bar{p}(z) + p(\frac{1}{z})\right).
\end{aligned}$$

We see that $q(z)$ is a non-constant complex polynomial of degree $\leq 2N$ and that for $m = 0, \ldots, M$ we have $0 = Re(\bar{p}(\xi_m))$. So we can compute

$$\begin{aligned}
q(\xi_m) &= (\xi_m)^N \cdot \left(\bar{p}(\xi_m) + p\left(\frac{1}{\xi_m}\right)\right) \\
&= (\xi_m)^N \cdot \left(\bar{p}(\xi_m) + p\left(\overline{\xi_m}\right)\right) \\
&= (\xi_m)^N \cdot 2Re\left(\bar{p}(\xi_m)\right) = 0.
\end{aligned}$$

Note that the second equality only holds because e^{imt} is a unit and the last equality holds by our conclusion from above. With $q(\xi_m) = 0$ we can conclude that $\xi_0, \dots, \xi_M$ are the roots of $q(z)$. With the fact that $M\tau < 2\pi$ it follows that $\xi_0, \dots, \xi_M$ are distinct and we have

$$M + 1 \leq degree(q(z)) \leq 2N$$

and therefore $2N > M$ as we wanted.
The **"only if" direction** is much more easy. Assume that $\mathbf{u}_0, \mathbf{u}_1, \mathbf{v}_1, \dots, \mathbf{u}_N, \mathbf{v}_N$ are linearly independent. These are $2N + 1$ vectors. It follows directly that the dimension $M \geq 2N + 1$, implying $M \geq 2N$. □

In the remainder of this work, we will always assume that $N \in \mathbb{N}$, that $M = 2N$ and that $\tau > 0$ satisfies $M\tau < 2\pi$.

6.2 Window Size and Underlying Frequency

In this section we want to study the relation between the window size and the frequency more profoundly to understand the structure of the sliding window point cloud geometrically.

For that recall that we saw in the beginning of Chapter 4 that the sliding window point cloud for $\sin(nt)$ describes a planar ellipse which is most likely to a cycle when $\|\mathbf{u}\| - \|\mathbf{v}\| = \langle \mathbf{u}, \mathbf{v} \rangle = 0$, or equivalently $(M+1)n\tau \equiv 0 \pmod{\pi}$. From this we get that the maximum persistence of the sliding window point cloud of $\sin(nt)$ is largest when the window size $M\tau$ is proportional to the frequency $\frac{2\pi}{n}$. The proportionality constant in this case is $\frac{M}{M+1}$.

Our goal for this section is to show that when looking at the truncated Fourier series $S_N f$ of a periodic function f and if the same proportionality relation between window size and underlying frequency holds, then the expression

$$SW_{M,\tau} S_N f(t) = \sum_{n=0}^{N} \cos(nt)(a_n \mathbf{u}_n + b_n \mathbf{v}_n) + \sin(nt)(b_n \mathbf{u}_n - a_n \mathbf{v}_n) \qquad (6.3)$$

becomes a linear decomposition into pairwise orthogonal vectors. To start our observations, we look at the case of the restriction to $Span\{\mathbf{u}_n, \mathbf{v}_n\}$.

Proposition 6.4 *Let $n \geq 1$. Then $\langle \mathbf{u}_n, \mathbf{v}_n \rangle = \|\mathbf{u}_n\|^2 - \|\mathbf{v}_n\|^2 = 0$ if and only if $(M+1)n\tau \equiv 0$ (mod π).*

Intuitively this proposition means:

- The orthogonality between $\mathbf{u}_n$ and $\mathbf{v}_n$ ensures that the components in the Fourier decomposition do not "mix".
- The condition $(M+1)n\tau \equiv 0 \pmod{\pi}$ ensures that the relationship between window size and frequency maintains this orthogonality.

This is crucial because it guarantees that the structural analysis of the signal remains geometrically and mathematically consistent, and therefore it avoids distortions in the decomposition.

Proof (of Proposition 6.4) First we compute $\langle \mathbf{u}_n, \mathbf{v}_n \rangle$ and $\|\mathbf{u}_n\|^2 - \|\mathbf{v}_n\|^2$.

$$\begin{aligned}\langle \mathbf{u}_n, \mathbf{v}_n \rangle &= \sum_{m=1}^{M} \cos(nm\tau)\sin(nm\tau) = \frac{1}{2}\sum_{m=1}^{M} \sin(2nm\tau) \\ &= \frac{1}{2} Im\left(\sum_{m=1}^{M} z_{2n\tau}^{m}\right), \text{ where } z_\varphi = e^{i\varphi} \\ &= \frac{1}{2} Im\left(z_{2n\tau} \cdot \frac{1 - z_{2n\tau}^{M}}{1 - z_{2n\tau}}\right) = \frac{1}{2} Im\left(\frac{z_{2n\tau} - z_{2n\tau}^{M+1}}{1 - z_{2n\tau}}\right) \\ &= \frac{1}{2} Im\left(\frac{1 - z_{2n\tau}^{M+1} - (1 - z_{2n\tau})}{1 - z_{2n\tau}}\right) \\ &= \frac{1}{2} Im\left(\frac{1 - z_{2n(M+1)\tau}}{1 - z_{2n\tau}} - 1\right) = \frac{1}{2} Im\left(\frac{1 - z_{2n(M+1)\tau}}{1 - z_{2n\tau}}\right),\end{aligned}$$

$$\begin{aligned}\|\mathbf{u}_n\|^2 - \|\mathbf{v}_n\|^2 &= \sum_{m=0}^{M} \cos(nm\tau)^2 - \sin(nm\tau)^2 = \sum_{m=0}^{M} \cos(2nm\tau) \\ &= Re\left(\frac{1 - z_{2n(M+1)\tau}}{1 - z_{2n\tau}}\right),\end{aligned}$$

where the last step follows similarly to the computations of $\langle \mathbf{u}_n, \mathbf{v}_n \rangle$. These two calculations imply that

$$4\langle \mathbf{u}_n, \mathbf{v}_n \rangle^2 + (\|\mathbf{u}_n\|^2 - \|\mathbf{v}_n\|^2)^2 = \left\| \frac{1 - z_{2n(M+1)\tau}}{1 - z_{2n\tau}} \right\|^2. \tag{6.4}$$

In conclusion this means that $\langle \mathbf{u}_n, \mathbf{v}_n \rangle = \|\mathbf{u}_n\|^2 - \|\mathbf{v}_n\|^2 = 0$ if and only if $z_{2n(M+1)\tau} = 1$, which is if and only if $n(M+1)\tau \equiv 0 \pmod{\pi}$. □

So we see that $(M+1)n\tau \equiv 0 \pmod{\pi}$ implies that $a_n\mathbf{u}_n + b_n\mathbf{v}_n$ is perpendicular to $b_n\mathbf{u}_n - a_n\mathbf{v}_n$ for all $a_n, b_n \in \mathbb{R}$. To be able to extend this result to components from different harmonics, we need the following definition:

Definition 6.5 *For $L \in \mathbb{N}$, a function f is* ***L-periodic*** *on $[0, 2\pi]$, if for all t we have*

$$f\left(t + \frac{2\pi}{L}\right) = f(t). \tag{6.5}$$

Lemma 6.6 *Let f be an L-periodic function. Then a_n and b_n are its n-th real Fourier coefficients. We let $a_n + ib_n = r_n e^{i\alpha\alpha_n}$, with $\alpha_n = 0$ when $r_n = 0$. Then $r_n \neq 0$ implies $n \equiv 0$ (mod L).*

Proof Let us look at the function $g(t) = f(t/L)$. This is 1-periodic, because $g(t+1) = f(\frac{t+1}{L}) = f(\frac{t}{L} + \frac{1}{L})$ with the L-periodicity of f this results in $g(t)$ again. The Fourier series expansion of $g(t)$ is

$$g(t) = \sum_{r=0}^{\infty} a'_r \cos(rt) + b'_r \sin(rt).$$

So we compute

$$f(t) = g(tL) = \sum_{r=0}^{\infty} a'_r \cos(rLt) + B'_r \sin(rLT) = \sum_{n=0}^{\infty} a_n \cos(nt) + b_n \sin(nt)$$

for almost every t. With the uniqueness of the Fourier series expansion in $L^2(\mathbb{T})$, our result follows. □

The following proposition makes it possible to make the potentially non-zero terms in the linear decomposition of $SW_{M,\tau}S_N f$ (Equation (6.3)) orthogonal to one another. For that we have to choose the window size proportional to the underlying frequency, with constant $\frac{M}{M+1}$.

Proposition 6.7 *Let the function f be L-periodic and let $\tau = \frac{2\pi}{L(M+1)}$. Then all vectors in the set*

$$\{\mathbf{u}_n, \mathbf{v}_n \mid 0 \leq n \leq N, n \equiv 0 \textit{ (mod } L)\} \tag{6.6}$$

are pairwise orthogonal, and $\|\mathbf{u}_n\| = \|\mathbf{v}_n\| = \sqrt{\frac{M+1}{2}}$ for $n \equiv 0$ (mod L).

Proof We define $k = pL$ and $n = qL$. For $k = n$ and $\langle \mathbf{u}_n, \mathbf{v}_n \rangle = 0$ we see that, due to orthogonality and symmetry,

$$\begin{aligned} \|\mathbf{u}_n\|^2 = \|\mathbf{v}_n\|^2 &= \frac{\|\mathbf{u}_n\|^2 + \|\mathbf{v}_n\|^2}{2} \\ &= \frac{1}{2} \sum_{m=0}^{M} \cos(nm\tau)^2 + \sin(nm\tau)^2 = \frac{M+1}{2}. \end{aligned}$$

In the last equality we used that $\cos(x)^2 + \sin(x)^2 = 1$. Let us now assume that $p \neq q$. Note that if $\langle \mathbf{u}_n, \mathbf{v}_n \rangle = 0$, then $\cos(nM\tau)\sin(nM\tau) = 0$, which is if and only if $\frac{1}{2}\sin(2nM\tau) = 0$

If we again set $z_\varphi = e^{i\varphi}$ for $\varphi \in \mathbb{R}$, then

$$\begin{aligned}
\langle \mathbf{u}_n, \mathbf{u}_k \rangle &= \sum_{m=0}^{M} \cos(nm\tau)\cos(km\tau) \\
&= \frac{1}{2}\sum_{m=0}^{M} \cos((n-k)m\tau) + \cos((n+k)m\tau) \\
&= \frac{1}{2} Re \left(\frac{1 - z_{(n-k)(M+1)\tau}}{1 - z_{(n-k)\tau}} + \frac{1 - z_{(n+k)(M+1)\tau}}{1 - z_{(n+k)\tau}} \right) \\
&= \frac{1}{2} Re \left(\frac{1 - z_{(q-p)2\pi}}{1 - z_{(n-k)\tau}} + \frac{1 - z_{(q+p)2\pi}}{1 - z_{(n+k)\tau}} \right) = 0.
\end{aligned}$$

The second equality follows because of the two trigonometric identities

$$\begin{aligned}
\cos(x+y) &= \cos(x)\cos(y) - \sin(x)\sin(y) \\
\cos(x-y) &= \cos(x)\cos(y) + \sin(x)\sin(y),
\end{aligned}$$

that lead to $\cos(x)\cos(y) = \frac{1}{2}(\cos(x+y) + \cos(x-y))$. In the forth equality, we simply compute

$$\begin{aligned}
(n-k)(M+1)\tau &= (q-p)L(M+1)\tau \\
&= (q-p)L(M+1)\frac{2\pi}{L(M+1)} = (q-p)2\pi.
\end{aligned}$$

The others follow similarly. We want to check, which condition guarantees that the denominators are never zero. We compute

$$1 - z_{(n-k)\tau} = 0 \iff 1 - e^{i(n-k)\tau} = 0 \iff 1 = e^{i(n-k)\tau}.$$

Similarly the other denominator vanishes if $1 = e^{i(n+k)\tau}$. These imply $(n-k)\tau = 2\pi\ell$ and $(n+k)\tau = 2\pi\ell$ for $\ell \in \mathbb{Z}$ respectively. So we need $(n-k)\tau \neq 2\pi\ell$ and $(n+k)\tau \neq 2\pi\ell$.

- We ensure $n-k, n+k \neq 0$ by setting $0 < \min\{|n-k|, |n+k|\}$.
- To ensure that $(n-k)\tau, (n+k)\tau \neq 2\pi\ell$, observe that a key bound is $M < \frac{2\pi}{\tau}$. Since M represents the maximum possible value of $|n-k|$ or $|n+k7$, and $\tau = \frac{2\pi}{L(M+1)}$, we know:

$$(n-k)\tau < M\tau < \frac{2\pi}{L(M+1)} \cdot M < 2\pi.$$

Thus $(n-k)\tau$ and $(n+k)\tau$ are strictly less than 2π, ensuring they cannot be an integer multiple.

- Setting $\max\{|n-k|, |n+k|\} \leq 2N \leq M$ ensures that all possible values of $n-k$ and $n+k$ remain bounded within the non-problematic region.

So our condition for the denominators not to be zero is the following:

$$0 < \min\{|n-k|, |n+k|\} \leq \max\{|n-k|, |n+k|\} \leq 2N \leq M < \frac{2\pi}{\tau}.$$

We check that the vectors in the set of Equation (6.6) are indeed orthogonal:

$$\begin{aligned}
\langle \mathbf{u}_n, \mathbf{v}_k \rangle &= \sum_{m=1}^{M} \cos(nm\tau)\sin(km\tau) \\
&= \frac{1}{2}\sum_{m=1}^{M} \sin((n+k)m\tau) - \sin((n-k)m\tau) \\
&= \frac{1}{2} Im \left(\frac{1 - z_{(q+p)2\pi}}{1 - z_{(n+k)\tau}} - \frac{1 - z_{(q-p)2\pi}}{1 - z_{(n-k)\tau}} \right) = 0
\end{aligned}$$

$$\langle \mathbf{v}_n, \mathbf{u}_k \rangle = \frac{1}{2} Im \left(\frac{1 - z_{(p+q)2\pi}}{1 - z_{(k+n)\tau}} - \frac{1 - z_{(p-q)2\pi}}{1 - z_{(k-n)\tau}} \right) = 0$$

$$\langle \mathbf{v}_n, \mathbf{v}_k \rangle = \frac{1}{2} Im \left(\frac{1 - z_{(q-p)2\pi}}{1 - z_{(n-k)\tau}} - \frac{1 - z_{(q+p)2\pi}}{1 - z_{(n+k)\tau}} \right) = 0 \qquad \square$$

We sometimes gain advantage in pointwise centering and normalizing our data set in order to compute the persistent homology. We will look at a theorem that shows the outcome of such operations on the sliding window point cloud for $SW_{M,\tau}S_N f$ for f L-periodic and for $L(M+1)\tau = 2\pi$. It is called the Structure Theorem, because it gives us a clear geometric picture of the centered and normalized point cloud.

Theorem 6.8 (Structure Theorem) *Let $C\colon \mathbb{R}^{M+1} \to \mathbb{R}^{M+1}$ be the centering map, for which holds*

$$C(\mathbf{x}) = \mathbf{x} - \frac{\langle \mathbf{x}, \mathbf{1} \rangle}{\|\mathbf{1}\|^2}, \text{ where } \mathbf{1} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^{M+1}. \tag{6.7}$$

If f is L-periodic, $L(M+1)\tau = 2\pi$ and $\phi_\tau = SW_{M,\tau}S_N f$, then the following three statements hold:

1. $\phi_\tau(t) = \hat{f}(0) \cdot \mathbf{1} + C(\phi_\tau(t))$
2. $\|C(\phi_\tau(t))\| = \sqrt{M+1}\left(\|S_N f\|_2^2 - \hat{f}(0)^2\right)^{1/2}$

3. *There exists an orthogonal set*

$$\{\widetilde{\mathbf{x}}_n, \widetilde{\mathbf{y}}_n \in \mathbb{R}^{M+1} | 1 \leq n \leq N, n \equiv 0 \text{ (mod } L)\}, \tag{6.8}$$

such that

$$\varphi_\tau := \frac{C(\phi_\tau(t))}{\|C(\phi_\tau(t))\|} = \sum_{\substack{n=1 \\ n\equiv 0 \text{ (mod } L)}}^{N} \widetilde{r}_n(\cos(nt)\widetilde{\mathbf{x}}_n + \sin(nt)\widetilde{\mathbf{y}}_n), \tag{6.9}$$

where

$$\widetilde{r}_n = \frac{2|\hat{f}(n)|}{\sqrt{\|S_N f\|_2^2 - \hat{f}(0)^2}}. \tag{6.10}$$

Proof 1. We know that $\phi_\tau(t) = SW_{M,\tau}S_N f$ so

$$\phi_\tau(t) = \begin{bmatrix} S_N f(t) \\ S_N(f+\tau) \\ \vdots \\ S_N f(t+M\tau) \end{bmatrix}. \tag{6.11}$$

For the function $S_N f(t)$ we have the following:

$$S_N f(t) = \hat{f}(0) + \sum_{n=1}^{N} \left(\hat{f}(n)e^{int} + \hat{f}(-n)e^{-int} \right),$$

where the term $\hat{f}(0)$ is the mean of $S_N f(t)$, while the rest are oscillatory terms. Substituting this in Equation (6.11), we get

$$\phi_\tau(t) = \begin{bmatrix} \hat{f}(0) + \sum(\text{oscillatory terms}) \\ \hat{f}(0) + \sum(\text{oscillatory terms at } t+\tau) \\ \vdots \\ \hat{f}(0) + \sum(\text{oscillatory terms at } t+M\tau) \end{bmatrix}.$$

Applying the centering map C, removes the constant mean term and we get

$$C(\phi_\tau(t)) = \phi_\tau(t) - \frac{\langle \phi_\tau(t), \mathbf{1} \rangle}{\|\mathbf{1}\|^2}\mathbf{1}.$$

Here we have that $\langle \phi_\tau(t), \mathbf{1} \rangle = (M+1)\hat{f}(0)$, since the constant vector $\mathbf{1}$ aligns perfectly with the mean term $\hat{f}(0)$. Furthermore, we have that $\|\mathbf{1}\|^2 = M+1$. Substituting these results, we get

$$C(\phi_\tau(t)) = \phi_\tau(t) - \hat{f}(0) \cdot \mathbf{1},$$

which leads to

$$\phi_\tau(t) = \hat{f}(0) \cdot \mathbf{1} + C(\phi_\tau(t)).$$

2. Notice that by the Pythagorean theorem, the norm of the centered embedding $C(\phi_\tau(t)) = \phi_\tau(t) - \hat{f}(0) \cdot \mathbf{1}$ satisfies

$$\begin{aligned}\|C(\phi_\tau(t))\|^2 &= \|\phi_\tau(t)\|^2 - \|\hat{f}(0) \cdot \mathbf{1}\|^2 \\ &= \|S_N f\|_2^2 - (M+1) \cdot \hat{f}(0)^2,\end{aligned}$$

which, taking the square roots, gives

$$\|C(\phi_\tau(t))\| = \sqrt{M+1} \left(\|S_N f\|_2^2 - \hat{f}(0)^2\right)^{1/2}.$$

3. For the last statement we first look at the explicit Fourier decomposition of $\phi_\tau(t)$. Since we have an L-periodic function on $[0, 2\pi]$ and know that $L(M+1)\tau = 2\pi$, we can use Lemma 6.6 and Proposition 6.7 to show that for all $t \in \mathbb{R}$,

$$\begin{aligned}\phi_\tau(t) &= \sum_{\substack{n=0 \\ n \equiv 0 \pmod{L}}}^{N} \cos(nt)(a_n \mathbf{u}_n + b_n \mathbf{v}_n) + \sin(nt)(b_n \mathbf{u}_n - a_n \mathbf{v}_n) \\ &= \sum_{\substack{n=0 \\ n \equiv 0 \pmod{L}}}^{N} r_n(\cos(nt)\mathbf{x}_n + \sin(nt)\mathbf{y}_n)\end{aligned}$$

is a linear combination of pairwise orthogonal vectors $\mathbf{x}_n$ and $\mathbf{y}_n$, where $\mathbf{x}_n = \cos(\alpha_n)\mathbf{u}_n + \sin(\alpha_n)\mathbf{v}_n$ and $\mathbf{y}_n = \sin(\alpha_n)\mathbf{u}_n - \cos(\alpha_n)\mathbf{v}_n$, and where α_n is from Lemma 6.6. From Proposition 6.7, we see that whenever $n \geq 1$ is such that $n \equiv 0 \pmod{L}$, then $\|\mathbf{x}_n\| = \|\mathbf{y}_n\| = \sqrt{\frac{M+1}{2}}$. We define the normalized vectors

$$\widetilde{\mathbf{x}}_n = \frac{\mathbf{x}_n}{\|\mathbf{x}_n\|}, \widetilde{\mathbf{y}}_n = \frac{\mathbf{y}_n}{\|\mathbf{y}_n\|}$$

and compute

$$\begin{aligned}\phi_\tau(t) = {} & (a_0\sqrt{M+1})\frac{\mathbf{1}}{\|\mathbf{1}\|} \\ & + \sum_{\substack{n=0 \\ n \equiv 0 \pmod{L}}}^{N} \sqrt{\frac{M+1}{2}} r_n(\cos(nt)\widetilde{\mathbf{x}}_n + \sin(nt)\widetilde{\mathbf{y}}_n).\end{aligned}$$

This is a linear decomposition of $\phi_\tau(t)$ in terms of the orthonormal set

$$\left\{\frac{\mathbf{1}}{\|\mathbf{1}\|}, \widetilde{\mathbf{x}}_n, \widetilde{\mathbf{y}}_n \mid 1 \leq n \leq N, n \equiv 0 \pmod{L}\right\}.$$

So we compute

$$C(\phi_\tau(t)) = \sum_{\substack{n=0 \\ n \equiv 0 \pmod{L}}}^{N} \sqrt{\frac{M+1}{2}} r_n(\cos(nt)\widetilde{\mathbf{x}}_n + \sin(nt)\widetilde{\mathbf{y}}_n)$$

and therefore

$$\begin{aligned}\varphi_\tau(t) &:= \frac{C(\phi_\tau(t))}{\|C(\phi_\tau(t))\|}\\ &= \sum_{\substack{n=0\\ n\equiv 0 \pmod L}}^{N} \frac{r_n}{\sqrt{r_1^2+\ldots+r_N^2}}(\cos(nt)\widetilde{\mathbf{x}}_n+\sin(nt)\widetilde{\mathbf{y}}_n),\end{aligned}$$

which can be written as

$$\varphi_\tau(t) = \sum_{\substack{n=0\\ n\equiv 0 \pmod L}}^{N} \widetilde{r}_n(\cos(nt)\widetilde{\mathbf{x}}_n+\sin(nt)\widetilde{\mathbf{y}}_n)\ ,\ \sum_{n=1}^{N}\widetilde{r}_n^2 = 1.$$

This is exactly what we want considering the fact that
$r_n = 2|\hat{f}(n)| = |\hat{f}(n)| + |\hat{f}(-n)|$ for $n \geq 1$. □

The centered and normalized sliding window point cloud for $S_N f$ is given by the expression in Equation (6.9). We want to check this with a small example of the circle with radius r around zero, $S^1(r) \subseteq \mathbb{C}$.

Example 6.9 *We look at the map $t \mapsto \varphi_\tau(t)$, that then can be regarded as the curve in the N-torus $\mathcal{T} = S^1(\widetilde{r}_1) \times \ldots \times S^1(\widetilde{r}_N)$. If we now project this onto $S^1(\widetilde{r}_n)$ for an $\widetilde{r}_n > 0$, it goes around n times at a constant speed. But if we project it with the projection map*

$$R^N = \mathbb{R} \times \ldots \times \mathbb{R} \to (\mathbb{R}/(\widetilde{r}_1\mathbb{Z})) \times \ldots \times (\mathbb{R}/(\widetilde{r}_N\mathbb{Z})),$$

our image is the line segment in $\mathbb{R}^N$ joining $(0, 0, \ldots, 0)$ and $(\widetilde{r}_1, 2\widetilde{r}_2, \ldots, N\widetilde{r}_N)$. Figure 6.1 shows the curve $\varphi_\tau(t)$ from Theorem 6.8 for this scenario inside the 3-torus $\mathcal{T}$.

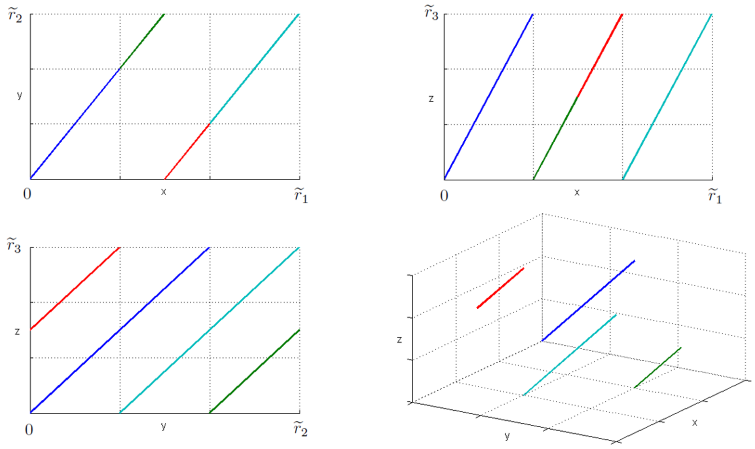


Figure 6.1: Curve $\varphi_\tau(t)$ when modeled in the coordinates $(t, 2t, 3t) \in (\mathbb{R}/(\tilde{r}_1\mathbb{Z})) \times (\mathbb{R}/(\tilde{r}_2\mathbb{Z})) \times (\mathbb{R}/(\tilde{r}_3\mathbb{Z}))$. In the bottom right, we see $\varphi_\tau(t)$ in its fundamental domain $[0, \tilde{r}_1) \times [0, \tilde{r}_2) \times [0, \tilde{r}_3)$. The three other diagrams show $\varphi_\tau(t)$ as projection onto the xy-, xz- and yz-plane. [25]

Chapter 7

The Persistent Homology of $SW_{M,\tau}f$ and φ_τ

With the background from our previous section over the structure of the sliding window point cloud and our approximation results, we can now get to the part of understanding the persistent homology of the image of φ_τ more precisely. In fact we also will have a closer look at its relationship with the persistent homology of $SW_{M,\tau}f$.

7.1 Theorems on Convergence of Persistence Diagrams

The goal of this section is to state and prove two important theorems dealing with the convergence of different persistence diagrams. We will state them here and then come back to them later.

Theorem 7.1 (1. Convergence Theorem) *Let $f \in C^1(\mathbb{T})$ be an L-periodic function, $N \in \mathbb{N}$, $\tau_N = \frac{2\pi}{L(2N+1)}$, $T \subseteq \mathbb{T}$ and $\overline{Y}_N$ be the set, that we get when pointwise centering and normalizing the point cloud*

$$SW_{2N,\tau_N} S_N f(T) \subseteq \mathbb{R}^{2N+1}.$$

Furthermore let $\overline{X}_N$ be the set, that we get when pointwise centering and normalizing the point cloud

$$SW_{2N,\tau_N} f(T) \subseteq \mathbb{R}^{2N+1}.$$

Then for any field of coefficients, the sequence of persistence diagrams $\mathrm{dgm}(\overline{X}_N)$ *is a Cauchy sequence with respect to the Bottleneck distance d_B, and*

$$\lim_{N\to\infty} \mathrm{dgm}(\overline{X}_N) = \lim_{N\to\infty} \mathrm{dgm}(\overline{Y}_N) = \mathrm{dgm}_\infty(f, T, w), \tag{7.1}$$

where $\mathrm{dgm}_\infty(f, T, w)$ *denotes the limit in the Bottleneck distance of the sequence* $\mathrm{dgm}(\overline{Y}_N)$ *and $w = \frac{2\pi}{L}$.*

The idea behind the notation $\mathrm{dgm}_\infty(f,T,w)$ is that there exists a limiting diagram from the sequence of pointwise centered and normalized versions of $SW_{M,\tau}f(T)$, when letting M go to infinity, and keeping the window size $M\tau = \frac{M}{M+1}w \approx w$.

This first theorem ensures that for every choice of T in $\mathbb{T}$, we get some limiting persistence diagram $\mathrm{dgm}_\infty(f,T,w)$ by letting N go to ∞ in the diagrams of either the sliding window point cloud of the function or the sliding window point cloud of the Fourier series of the function. Furthermore it validates the above interpretation of $\mathrm{dgm}(f,T,w)$.

Example 7.2 *To demonstrate Theorem 7.1, we choose the 2π-periodic function $f(t) = \sin(t)$, so $L = 2\pi$. Therefore $\tau_N = \frac{1}{2N+1}$. The code for the following computations and illustrations can be found in the Appendix, Figure A.5. We choose T to consist of 100 points on $\mathbb{T}$. We compute the point clouds $SW_{2N,\tau_N}S_Nf(T)$ and $SW_{2N,\tau_N}f(T)$. Then we pointwise center and normalize these point clouds. We then compute the persistence diagrams for different values of N, using the risper package in python, to show convergence of the two diagrams, see Figure 7.1.*

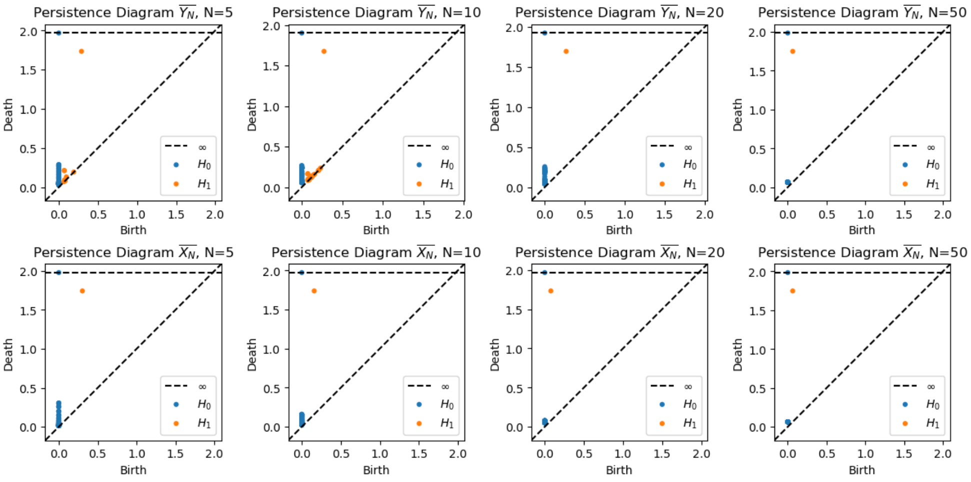


Figure 7.1: For $N = 5$ we see that the persistence diagrams differ significantly, but they are getting closer as N increases and are nearly the same at $N = 50$.

Our second convergence theorem deals with the case where we let T go to $\mathbb{T}$ and shows that there will also be a convergence in the persistence diagrams.

Theorem 7.3 (2. Convergence Theorem) *Let $T, T' \subseteq \mathbb{T}$ be finite, and let a function $f \in C^1(\mathbb{T})$ be L-periodic with modulus of continuity $\omega\colon [0,\infty] \to [0,\infty]$*[1]*. If we set the window size to be $w = \frac{2\pi}{L}$, then*

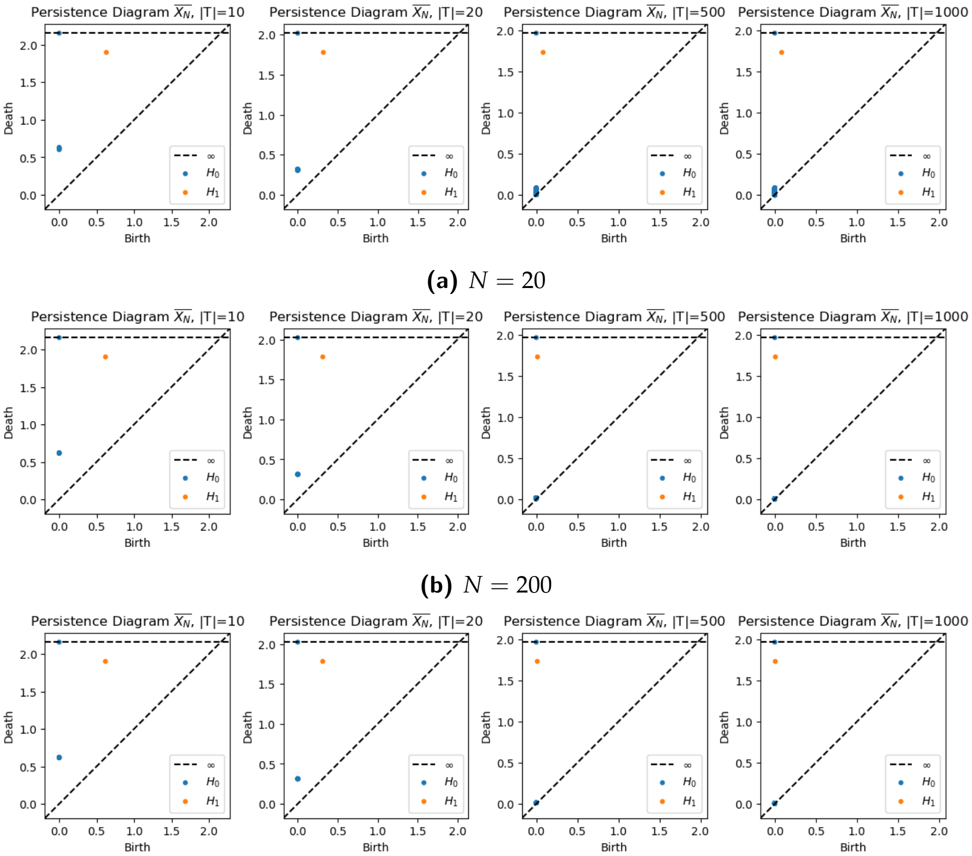
$$d_B(\mathrm{dgm}_\infty(f,T,w), \mathrm{dgm}_\infty(f,T',w)) \leq 2\left\|f - \hat{f}(0)\right\|_2 \omega(d_H(T,T')),$$

[1] A function f has modulus of continuity ω, if $|f(x) - f(y)| \leq \omega(|x-y|)$.

and therefore there exists a persistence diagram $\mathrm{dgm}_\infty(f, w)$*, such that*

$$\lim_{T \to \mathbb{T}} \mathrm{dgm}_\infty(f, T, w) = \mathrm{dgm}_\infty(f, w). \tag{7.2}$$

Example 7.4 *To understand Theorem 7.3 more profoundly we will take again* $f(t) = \sin(t)$ *as in Example 7.2. We follow the same steps, but only for the sliding window point cloud of the function itself, because we already proved that the diagrams converge. To approximate* $\mathbb{T}$*, we let the size of T increase. Here we have* $T = 10, 20, 500, 1000$*. We also run the code, displayed in the Appendix, Figure A.6, for different values of N to demonstrate the convergence from Theorem 7.1.*



(a) $N = 20$

(b) $N = 200$

(c) $N = 2000$

Figure 7.2: The persistence diagrams converge as the size of T increases.

We will have to do some prework to prove these theorems. Notice that Proposition 5.4 gives an upper bound for the distance between $\phi_\tau(T)$ and $SW_{M,\tau}f(T)$. If now N gets lagers and therefore also $M = 2N$, the denominator grows faster and the distance gets smaller with respect to the Hausdorff metric on subspaces of $\mathbb{R}^\infty$.

Definition 7.5 $\mathbb{R}^\infty$ *denotes the set of sequences* $x = (x_k)_{k\in\mathbb{N}}$, *where* $x_k \in \mathbb{R}$, *such that* $x_n = 0$ *for all* $n \geq N_0$ *and some* $N_0 = N_0(x) \in \mathbb{N}$.

We want to be able to look at our sliding window embedding in this space. Therefore, we endow $\mathbb{R}^\infty$ with the L^2-metric and define for $t \in T$

$$SW_{M,\tau}f(t) = (f(t), f(t+\tau), \ldots, f(t+M\tau), 0, 0, \ldots) \in \mathbb{R}^\infty. \tag{7.3}$$

Even though we get better approximations of the form

$$SW_{M,\tau}S_N f(T) \approx SW_{M,\tau}f(T)$$

when increasing the sliding window embedding dimension $M+1$, we also have to deal with the fact that the approximated object, $SW_{M,\tau}f(T)$ changes. We cannot assume that this process of approximating converges or stabilizes, because $(\mathbb{R}^\infty, \|\cdot\|_2)$ is not complete and there are samplings at different rates from the same window. Moreover, even though the metric completion $\overline{\mathbb{R}^\infty} = \ell^2(\mathbb{R})$, the space of square-summable sequences, is well understood, its vastness makes tracking global geometric features a nontrivial task. In this kind of situations, the method of a persistence diagram, working as a succinct and informative summary, is crucial.

However, the space of persistence diagrams is also not complete with respect to the Bottleneck distance. To find a completion, we need some extra work with two propositions from [20] and [3], done in Section 4, see Proposition 2.38 and Theorem 2.40. So we see that the space of persistence diagrams can be completed by allowing diagrams with countably many points with at most countable multiplicity, satisfying a natural finiteness condition. Furthermore, features like maximum persistence can be easily tracked, with no ambiguity in comparing diagrams, such as those from $SW_{M,\tau}f(T)$ and $SW_{3M,\frac{\tau}{3}}f(T)$ that seem kind of similar.

The following proposition ensures that persistence diagrams from sliding window embeddings remain consistent as embedding dimensions and resolutions change, providing a bound on their differences in terms of Fourier approximation error. This demonstrates the stability and robustness of persistence diagrams, making them reliable for analyzing signals at varying resolutions.

Proposition 7.6 *Let* f *be an L-periodic function and choose* $N < N'$, $M = 2N$, $M' = 2N'$ *and*

$$\tau = \frac{2\pi}{L(M+1)}, \tau' = \frac{2\pi}{L(M'+1)}.$$

If now $T \subseteq \mathbb{T}$ *is finite,* $Y = SW_{M,\tau}S_N f(T)$ *and* $Y' = SW_{M',\tau'}S_{N'}f(T)$, *then we get the following bound*

$$d_B\left(\frac{\mathrm{dgm}(Y)}{\sqrt{M+1}}, \frac{\mathrm{dgm}(Y')}{\sqrt{M'+1}}\right) \leq 2\|S_N f - S_{N'}f\|_2, \tag{7.4}$$

where $\lambda \cdot \mathrm{dgm}(A) := \{(\lambda x, \lambda y) | (x,y) \in \mathrm{dgm}(A)\}$ *for some* $\lambda \geq 0$.

Proof Remember that we defined $\mathbf{u}_n$ and $\mathbf{v}_n$ in Equation (5.8). We simplify our notation as follows: $\mathbf{u}_n = \mathbf{u}_n(M,\tau)$, $\mathbf{v}_n = \mathbf{v}_n(M,\tau)$, $\mathbf{u}'_n = \mathbf{u}_n(M',\tau')$ and $\mathbf{v}'_n = \mathbf{v}_n(M',\tau')$. We can now define two linear maps, that are well defined by Proposition 6.2:

$$\begin{aligned} P\colon \mathbb{R}^{M'+1} &\to \mathbb{R}^{M'+1} \\ \sum_{n=0}^{N'} x_n\mathbf{u}'_n + y_n\mathbf{v}'_n &\mapsto \sum_{n=0}^{N} x_n\mathbf{u}'_n + y_n\mathbf{v}'_n, \end{aligned}$$

$$\begin{aligned} Q\colon Im(P) &\to \mathbb{R}^{M+1} \\ \mathbf{u}'_n &\mapsto \sqrt{\frac{M'+1}{M+1}}\mathbf{u}_n \\ \mathbf{v}'_n &\mapsto \sqrt{\frac{M'+1}{M+1}}\mathbf{v}_n. \end{aligned}$$

With Proposition 6.7, we see that P is an orthogonal projection that restricts the higher-dimensional sliding window point cloud Y' (with embedding dimension $M'+1$) onto a subspace corresponding to the lower-dimensional point cloud Y (with embedding dimension $M+1$). The idea is to remove contributions from Fourier terms of higher frequencies ($> M$) present in Y', which do not exist in Y. Q is a scaling map that adjusts the norms of the basis vectors $\mathbf{u}_n$ and $\mathbf{v}_n$ to ensure consistency between the embeddings of $S_N f$ and $S_{N'} f$. Specifically, it rescales the vectors such that their norms are proportional to the dimensions $M+1$ and $M'+1$. Again with Proposition 6.7, we see that Q is an isometry on $P(Y')$. Notice that for $y' \in Y'$, the error introduced by projecting y' onto the lower-dimensional space is entirely due to the excluded Fourier terms. These terms contribute their respective amplitudes r_n^2, where r_n is defined as in Lemma 6.6, scaled by $\frac{M'+1}{2}$, to the squared norm of $y' - P(y')$. This leads to

$$\|y' - P(y')\| = \sqrt{\frac{M'+1}{2}\sum_{n=N+1}^{N'} r_n^2}$$

and therefore

$$d_H(Y', P(Y')) \leq \sqrt{\frac{M'+1}{2}\sum_{n=N+1}^{N'} r_n^2}.$$

Since $Q \circ P(Y') = \sqrt{\frac{M'+1}{M+1}} Y$ and $\mathrm{dgm}(\cdot)$ is invariant under isometries, we can compute

$$\begin{aligned}\sqrt{M'+1} \cdot d_B\left(\frac{\mathrm{dgm}(Y')}{\sqrt{M'+1}}, \frac{\mathrm{dgm}(Y)}{\sqrt{M+1}}\right) &= d_B(\mathrm{dgm}(Y'), \mathrm{dgm}(Q \circ P(Y'))) \\ &= d_B(\mathrm{dgm}(Y'), \mathrm{dgm}(P(Y'))) \\ &\leq 2d_h(Y', P(Y')) \\ &\leq \sqrt{2(M'+1) \sum_{n=N+1}^{N'} r_n^2}.\end{aligned}$$

This is equivalent to

$$d_B\left(\frac{\mathrm{dgm}(Y')}{\sqrt{M'+1}}, \frac{\mathrm{dgm}(Y)}{\sqrt{M+1}}\right) \leq \sqrt{2 \sum_{n=N+1}^{N'} r_n^2}.$$

With the identity $r_n = 2|\hat{f}(n)| = |\hat{f}(n)| + |\hat{f}(-n)|$ for $n \geq 1$, the result follows. □

Taking this result, the Structure Theorem 6.8 and the fact that for $N \to \infty$, $\|f - S_N f\|_2 \to 0$, we can state the following corollary.

Corollary 7.7 *Let $f \in L^2(\mathbb{T})$ be L-periodic, $N \in \mathbb{N}$, $\tau_N = \frac{2\pi}{L(2N+1)}$, $T \subseteq \mathbb{T}$ finite and let $\overline{Y}_N$ be the set we get from pointwise centering and normalizing the point cloud*

$$SW_{2N,\tau_N} S_N f(T) \subseteq \mathbb{R}^{2N+1}.$$

The the sequence $\mathrm{dgm}(\overline{Y}_N)$ of persistence diagrams is Cauchy with respect to the Bottleneck distance for every field of coefficients.

To be able to prove the first convergence Theorem 7.1, we need a statement that allows us to use the Approximation Theorem 5.5 to bound the Bottleneck distance between persistence diagrams at different resolutions. This is done by the following proposition, that ensures that the norm of the sliding window embedding stabilizes.

Proposition 7.8 *Let $f \in C(\mathbb{T})$ be L-periodic, $N \in \mathbb{N}$ and τ_N as in Corollary 7.7. Then*

$$\lim_{N\to\infty} \frac{\|C(SW_{2N,\tau_N} f(t)\|}{\sqrt{2N+1}} = \|f - \hat{f}(0)\|_2 \tag{7.5}$$

uniformly in $t \in \mathbb{T}$.

Proof We first look at the easy case, where f is constant. Then the sliding window embedding consists of one point, therefore the centered version becomes the 0-vector. With that the scaled norm in the limit is 0, which agrees with the right hand side of the equation, because for a constant function we have $f - \hat{f}(0) = 0$.

So let us assume that $f \neq \hat{f}(0)$ and let

$$g(t) = \frac{f(t) - \hat{f}(0)}{\|f - \hat{f}(0)\|_2}.$$

$g \in C(\mathbb{T})$ is L-periodic, because f is and we have

$$\hat{g}(0) = \frac{1}{2\pi} \int_0^{2\pi} g(t)dt = 0 \text{ and } \|g\|_2 = \frac{1}{\sqrt{2\pi}} \left(\int_0^{2\pi} |g(t)|^2 dt \right)^{1/2} = 1.$$

Using the Riemann sum argument $L(2N+1)\tau_N = 2\pi$, and the fact that g is L-periodic, it follows that if

$$c_N(t) = \frac{1}{2N+1} \sum_{m=0}^{2N} g(t + m\tau_N),$$

then for all $t \in \mathbb{T}$

$$\begin{aligned} \lim_{N\to\infty} c_N(t) &= \lim_{\tau_N \to 0} \frac{L}{2\pi} \sum_{m=0}^{2N} \tau_N g(t + m\tau_N) \\ &= \frac{L}{2\pi} \int_t^{t+\frac{2\pi}{L}} g(r)dr \\ &= \frac{1}{2\pi} \int_0^{2\pi} g(r)dr \\ &= 0. \end{aligned}$$

The uniform convergence of $c_n(t) \to 0$ follows from the periodicity and uniform continuity of $g(t)$. Since $g(t)$ is continuous and $\tau_N \to 0$, the spacing between sampled points in the Riemann sum becomes arbitrarily small, ensuring that the sum converges uniformly to the integral, which is zero. Moreover, the fact that g is uniformly continuous implies that $c_N(t)$ is uniformly equicontinuous. This means that for every $\varepsilon > 0$, there exists a $\delta > 0$, that is independent of N, such that for every $t, t' \in \mathbb{T}$ and all $N \in \mathbb{N}$

$$|t - t'| < \delta \implies |c_N(t) - c_N(t')| < \frac{\varepsilon}{2}.$$

For some $t \in \mathbb{T}$, let $N_t \in \mathbb{N}$ such that $N \geq N_t$ implies that $|c_N(t)| < \frac{\varepsilon}{2}$. So we can conclude with these two inequalities that if $N \geq N_t$ and $|t - t'| < \delta$, then $|c_N(t')| < \varepsilon$.

To conclude to proof, we have to show that the convergence $c_N(t) \to 0$ is uniform and that the Equation (7.5) also is uniformly. Let us choose a finite open cover of $[0, 2\pi]$ with intervals of length δ and let $N_0 := \max\{N_t\}$, where the maximum is taken corresponding to their centers. We get that $N \geq N_0$ implies $|c_N(t)| < \varepsilon$ for all $t \in \mathbb{T}$. This shows that the convergence $c_N(t) \to 0$ is uniform.

Similarly we compute that

$$\begin{aligned}\lim_{N\to\infty} \frac{\|C(SW_{2N,\tau_N} f(t)\|^2}{2N+1} &= \lim_{\tau_N \to 0} \frac{L}{2\pi} \sum_{m=0}^{2N} \tau_N (g(t + m\tau_N) - c_N(t))^2 \\ &= \frac{1}{2\pi} \int_0^{2\pi} g(r)^2 dr = 1\end{aligned}$$

converges uniformly in $t \in \mathbb{T}$. Replacing g by $\frac{f(t)-\hat{f}(0)}{\|f-\hat{f}(0)\|_2}$, yields the desired result. □

With these results now stated, we can go on and prove the convergence theorems.

Proof (Proof of Theorem 7.1) We will prove this theorem by using the Approximation Theorem 5.5 to show that

$$\lim_{N\to\infty} d_B \left(\text{dgm}(\overline{X}_N), \text{dgm}(\overline{Y}_N)\right) = 0.$$

Then we can use Corollary 7.7 to conclude the desired result.

Without loss of generality, we can assume that f satisfies $\hat{f}(0) = 0$, meaning the average value of f over its domain is zero and assume that $\|f\|_2 = 1$, meaning f has been normalized in the L^2 sense. Now consider X_N as the set of points obtained from pointwise centering the sliding window embedding of f, $SW_{2N,\tau_N} f(T)$ and Y_N as the set obtained similarly but using the truncated Fourier function $S_N f$, so $SW_{2N,\tau_N} S_N f(T)$. Thus X_N and Y_N represent point clouds derived from the original function f and its truncated Fourier approximation $S_N f$ respectively. From the uniform convergence result in Proposition 7.8, we obtain

$$\lim_{N\to\infty} d_H \left(\overline{X}_N, \frac{X_N}{\sqrt{2N+1}}\right) = 0.$$

Here the scaling factor $\sqrt{2N+1}$ normalizes X_N in a way that makes it converge in shape. Since X_N scales with denominator $\sqrt{2N+1}$, the point cloud shrinks as $N \to \infty$, ensuring that the Hausdorff distance between the centered version and its rescaled form vanishes.

Additionally, we use the fact that

$$\lim_{N\to\infty} \|S_N f\|_2 = \|f\|_2 = 1$$

from our assumptions. Substituting this into our previous limits gives

$$\lim_{N\to\infty} d_H\left(\frac{X_N}{\sqrt{2N+1}}, \frac{X_N}{\sqrt{2N+1}\|S_N f\|_2}\right) = 0.$$

Notice that since $\|S_N f\|_2 \to 1$, dividing by $\|S_N f\|_2$ asymptotically has no effect on the limit and the distance between these two rescaled versions of X_N tends to zero.

From point 2 in the Structure Theorem 6.8, we get the identity

$$\overline{Y}_N = \frac{Y_N}{\sqrt{2N+1}\|S_N f\|_2}.$$

From the first statement in the Approximation Theorem 5.5 with $k = 1$ and the fact that centering is a distance non-increasing operation, we conclude

$$\lim_{N\to\infty} d_H\left(\frac{X_N}{\sqrt{2N+1}\|S_N f\|_2}, \overline{Y}_N\right) \leq \lim_{N\to\infty} \frac{\sqrt{2}\cdot\|R_N f^{(1)}\|_2}{\|S_N f\|_2 \cdot \sqrt{N+1}} = 0.$$

With these three Hausdorff distances going to 0, we can use the triangle inequality, to conclude that

$$\lim_{N\to\infty} d_H(\overline{X}_N, \overline{Y}_N) = 0.$$

Finally, by the stability of the Bottleneck distance d_B with respect to the Hausdorff distance d_H, we conclude

$$\lim_{N\to\infty} d_B\left(\mathrm{dgm}(\overline{X}_N), \mathrm{dgm}(\overline{Y}_N)\right) = 0.$$

This follows because persistence diagrams are stable under small perturbations in the underlying point clouds. Thus, combining this with Corollary 7.7, we get our result. □

Proof (Proof of Theorem 7.3) We want to show that the Bottleneck distance between the persistence diagrams of the function f, over two different domains T and T', can be controlled by the Hausdorff distance between these two domains.

Let us fix two points $t \in T$ and $t' \in T'$. Define the sliding window embeddings for these points as follows: $\mathbf{x}_N = SW_{2N,\tau_N} f(t)$ and $\mathbf{x}'_N = SW_{2N,\tau_N} f(t')$. Here, the sliding window embedding creates a point in $\mathbb{R}^{2N+1}$ by sampling function values at time steps determined by the window size and the sampling interval. This interval is $\tau_N = \frac{2\pi}{L(2N+1)}$ and we define $\lambda = \|f - \hat{f}(0)\|_2$. Then by expanding the normed difference, we have

$$\left\| \frac{C(\mathbf{x}_N)}{\|C(\mathbf{x}_N)\|} - \frac{C(\mathbf{x}'_N)}{\|C(\mathbf{x}'_N)\|} \right\| \leq \left\| \frac{C(\mathbf{x}_N)}{\|C(\mathbf{x}_N)\|} - \frac{\lambda C(\mathbf{x}_N)}{\sqrt{2N+1}} \right\| + \frac{\lambda\|C(\mathbf{x}_N) - C(\mathbf{x}'_N)\|}{\sqrt{2N+1}}$$
$$+ \left\| \frac{C(\mathbf{x}'_N)}{\|C(\mathbf{x}'_N)\|} - \frac{\lambda C(\mathbf{x}'_N)}{\sqrt{2N+1}} \right\|.$$

Notice that we can rewrite the term

$$\begin{aligned}\left\|\frac{C(\mathbf{x}_N)}{\|C(\mathbf{x}_N)\|}-\frac{\lambda C(\mathbf{x}_N)}{\sqrt{2N+1}}\right\| &= \|C(\mathbf{x}_N)\|.\left|\frac{1}{\|C(\mathbf{x}_N)\|}-\frac{\lambda}{\sqrt{2N+1}}\right|\\ &= \frac{\|C(\mathbf{x}_N)\|}{\sqrt{2N+1}}\cdot\left|\frac{\sqrt{2N+1}}{\|C(\mathbf{x}_N)\|}-\lambda\right|\end{aligned}$$

and similarly we can rewrite the same term with $\mathbf{x}'_N$. So with Proposition 7.8, these terms go to zero as $N\to\infty$.

Since the convergence is uniformly, and given a fixed $\varepsilon>0$, we can choose a $N_0\in\mathbb{N}$, such that $N\geq N_0$ implies

$$\begin{aligned}\left\|\frac{C(\mathbf{x}_N)}{\|C(\mathbf{x}_N)\|}-\frac{C(\mathbf{x}'_N)}{\|C(\mathbf{x}'_N)\|}\right\| &\overset{(a)}{\leq}\frac{\varepsilon}{2}+\frac{\lambda\|C(\mathbf{x}_N)-C(\mathbf{x}'_N)\|}{\sqrt{2N+1}}\\ &\overset{(b)}{\leq}\frac{\varepsilon}{2}+\frac{\lambda\|\mathbf{x}_N-\mathbf{x}'_N\|}{\sqrt{2N+1}}\\ &\overset{(c)}{=}\frac{\varepsilon}{2}+\lambda\left(\sum_{n=0}^{2N}\frac{|f(t+n\tau_N)-f(t'-n\tau_N)|^2}{2N+1}\right)^{1/2}\\ &\overset{(d)}{\leq}\frac{\varepsilon}{2}+\lambda\omega(|t-t'|).\end{aligned}$$

(a): The term $\frac{\varepsilon}{2}$ stands for a small perturbation bound, whereas the key equality used here is a form of Lipschitz continuity. The fraction $\frac{1}{\sqrt{2N+1}}$ comes from the normalization of the embedding.

(b): Here, the norm of the centered embeddings $C(\mathbf{x}_N)$ is replaced with the norm of the original embeddings $\mathbf{x}_N$. This works because the centering transformation preserves distances up to somee bound.

(c): Here, we expand the norm $\|\mathbf{x}_N-\mathbf{x}'_N\|$ in terms of the differences between function values at sampled points.

(d): Since the sum in the square root is a discrete approximation of an integral, we bound it using $\omega(|t-t'|)$.

Let now $\overline{X}_N$ and $\overline{X'}_N$ be the centered and normalized sets $SW_{2N,\tau_N}f(T)$ and $SW_{2N,\tau_N}f(T')$ respectively. By the result of Proposition 7.8, all the estimates above are uniform in t and t'. From that it follows that for all $N\geq N_0$, we have

$$d_H(\overline{X}_N,\overline{X'}_N)\leq\frac{\varepsilon}{2}+\lambda\omega(d_H(T,T')),$$

where d_H on the left hand side is with respect to $\mathbb{R}^{2N+1}$ and the one on the right hand side is with respect to $\mathbb{T}$. The Stability Theorem 2.35 tells us

$$d_B(\operatorname{dgm}(\overline{X}_N),\operatorname{dgm}(\overline{X'}_N))\leq\varepsilon+2\lambda\omega(d_H(T,T')).$$

If we now let $N \to \infty$ in this equation and apply the first Convergence Theorem 7.1, we get

$$d_B(\mathrm{dgm}_\infty(f,T,w), \mathrm{dgm}_\infty(f,T',w)) \leq \varepsilon + 2\lambda\omega(d_H(T,T')).$$

We proved this for every ε, therefore we can let $\varepsilon \downarrow 0$ and get the first part of the theorem. As we know that the sequence of generalized persistence diagrams $\mathrm{dgm}_\infty(f,T,w)$ is Cauchy, we see that it must have a limit, and therefore the set of generalized persistence diagrams is complete with respect to d_B. The existence of $\mathrm{dgm}_\infty(f,w)$ follows. □

7.2 A Lower Bound for Maximum Persistence

This section establishes a lower bound for the maximum persistence of a function's persistence diagram, ensuring that the sliding window embedding retains significant topological features even under small perturbations.

The following theorem ensures that the sliding window embedding retains meaningful topological information about the function f, even under small perturbations (here expressed in δ and κ_N), which appear in real world data. It provides a stability guarantee that connects persistence diagrams with the geometric properties of the function and its derivatives. The theorem also shows that the death times of persistent homology classes are bounded from below, meaning that significant topological features cannot disappear too quickly.

Theorem 7.9 *Let $f \in C^2(\mathbb{T})$ be an L-periodic function, $N \in \mathbb{N}$, $M \geq 2N$, $L(M+1)\tau = 2\pi$ and $T \subseteq \mathbb{T}$ finite. Additionally, we assume that $d_H(T,\mathbb{T}) < \delta$ for some δ, such that*

$$0 < \delta < \max_{1 \leq n \leq N} \frac{\sqrt{3}\tilde{r}_n}{\kappa_N}, \text{ where } \kappa_N = \frac{2\sqrt{2}\|S_N f'\|_2}{\|S_N(f - \hat{f}(0))\|_2}.$$

For easier notation, we set $\overline{Y} = \overline{Y}_N$ to be the set that results from pointwise centering and normalizing the point cloud $SW_{M,\tau}S_N f(T) \subseteq \mathbb{R}^{M+1}$. Let $p > N$ be prime. If $\mathrm{dgm}(\overline{Y})$ is the 1-dimensional $\mathbb{F}_p$-persistence diagram for the Rips filtration on $\overline{Y}$, then φ_τ gives us an element $\mathbf{x}_\varphi \in \mathrm{dgm}(\overline{Y})$ with the following properties:

1. $\mathrm{birth}(\mathbf{x}_\varphi) \leq \delta\kappa_N$
2. $\mathrm{death}(\mathbf{x}_\varphi) \geq \sqrt{3} \max_{1 \leq n \leq N} \tilde{r}_n$.

Therefore we have the following lower bound for maximum persistence:

$$mp\left(\mathrm{dgm}(\overline{Y})\right) \geq \left(\sqrt{3} \max_{1 \leq n \leq N} \tilde{r}_n\right) - \delta\kappa_N. \tag{7.6}$$

Proof From the third point in the Structure Theorem 6.8 and with respect to the orthogonal set $\{\widetilde{\mathbf{x}}_n, \widetilde{\mathbf{y}}_n \in \mathbb{R}^{M+1} | 1 \leq n \leq N, n \equiv 0 \pmod{L}\}$, we have a linear decomposition of φ_τ :

$$\varphi_\tau(t) = \sum_{\substack{n=1 \\ n \equiv 0 \pmod{L}}}^{N} \widetilde{r}_n(\cos(nt)\widetilde{\mathbf{x}}_n + \sin(nt)\widetilde{\mathbf{y}}_n).$$

If follows that the following map

$$\begin{aligned} P_n \colon \overline{Y} &\to \mathbb{C} \\ \varphi_\tau(t) &\mapsto \tilde{r}_n e^{int} \end{aligned}$$

is the orthogonal projection from $\mathbb{R}^{M+1}$ to $Span\{\tilde{\mathbf{x}}_n, \tilde{\mathbf{y}}_n\}$, when restricted to $\overline{Y}$. With the fact that orthogonal projections are linear and norm-non-increasing, we get

$$\|P_n(\mathbf{x}) - P_n(\mathbf{y})\| \leq \|\mathbf{x} - \mathbf{y}\|$$

for all $\mathbf{x}, \mathbf{y} \in \overline{Y}$. So if $S^1(\tilde{r}_n) = \{\tilde{r}_n e^{int} | t \in T\}$, then P_n induces simplicial maps

$$\begin{aligned} P_{n_\#} \colon R_\varepsilon(\overline{Y}) &\to R_\varepsilon(S^1(\tilde{r}_n)) \\ [\mathbf{x}_0, \ldots, \mathbf{x}_k] &\mapsto [P_n(\mathbf{x}_0), \ldots, P_n(\mathbf{x}_k)] \end{aligned}$$

for every $\varepsilon > 0$. These simplicial maps give us the homomorphisms

$$P_{n_*} \colon H_k\bigg(R_\varepsilon(\overline{Y}); \mathbb{F}_p\bigg) \to H_k\bigg(R_\varepsilon(S^1(\tilde{r}_n); \mathbb{F}_p\bigg),$$

that are homomorphisms of $\mathbb{F}_p$-vector spaces at homology level. What we now want to show is that through the homomorphisms P_{n_*}, the maximum 1-dimensional persistence of $\overline{Y}$ can be bounded below by that of $S^1(\tilde{r}_n)$.
so let now $\varepsilon_1, \varepsilon_2 > 0$ be such that

$$\delta\kappa_N < \varepsilon_1 < \varepsilon_2 < \sqrt{3}\tilde{r}_m,$$

for $m = \arg\max\{\tilde{r}_n | 1 \leq n \leq N\}$. Remember that $d_H(T, \mathbb{T}) < \delta$, and if we take $T = \{t_0 < t_2 < \ldots < t_J\}$, it follows that $|t_j - t_{j-1}| < 2\delta$ for all

$j = 1, \ldots, J$. Therefore we can compute

$$\begin{aligned}
\|\varphi_\tau(t_j) - \varphi_\tau(t_{j-1})\|^2 &= \sum_{\substack{n=1 \\ n\equiv 0 \ (\mathrm{mod}\ L)}}^{N} 2\widetilde{r}_n^2 \Big(1 - \cos(n(t_j - t_{j-1})) \Big) \\
&\leq \sum_{\substack{n=1 \\ n\equiv 0 \ (\mathrm{mod}\ L)}}^{N} \widetilde{r}_n^2 (n(t_j - t_{j-1}))^2 \\
&= (t_j - t_{j-1})^2 \sum_{n=1}^{N} \frac{4n^2 |\hat{f}(n)|^2}{\|S_N f\|_2^2 - \hat{f}(0)^2} \\
&= \frac{(t_j - t_{j-1})^2}{\|S_N(f - \hat{f}(0))\|_2^2} \sum_{1 \leq |n| \leq N} 2|\hat{f}'(n)|^2 \\
&\leq 8\delta^2 \frac{\|S_N f'\|_2^2}{\|S_N(f - \hat{f}(0))\|_2^2} = (\delta\kappa_N)^2.
\end{aligned}$$

Notice that the first inequality is due to the Taylor expansion for $\cos(x)$ around zero. Therefore

$$\nu = [\varphi_\tau(t_0), \varphi_\tau(t_1)] + \ldots + [\varphi_\tau(t_{J-1}), \varphi_\tau(t_J)] + [\varphi_\tau(t_J), \varphi_\tau(t_0)]$$

is a 1-dimensional cycle on $R_{\varepsilon_1}(\overline{Y})$, because every summand is a 1-simplex and we get the homology class

$$P_{m_*}([\nu]) \in H_1(R_{\varepsilon_1}(S^1(\tilde{r}_m)); \mathbb{F}_p).$$

Let now $\{\theta_0 < \theta_1 < \ldots < \theta_{J_m}\} = \{t \bmod \frac{2\pi}{m} | t \in T\}$ and let $c_j = \tilde{r}_m e^{im\theta_j}$. By performing a similar calculation as above, we get

$$\|c_j - c_{j-1}\|^2 \leq (\theta_j - \theta_{j-1})^2 \frac{4|\hat{f}'(m)|^2}{\|S_N(f - \hat{f}(0))\|_2^2} \leq (\delta\kappa_N)^2$$

and therefore the 1-dimensional cycle

$$\mu = [c_0, c_1] + \ldots + [c_{J_{m-1}}, c_{J_m}] + [c_{J_m}, c_0]$$

with $[mu] \in H_1(R_{\varepsilon_1}(S^1(\tilde{r}_m)); \mathbb{F}_p)$ satisfies $i_*([\mu]) \neq 0$. Here i_* is the homomorphism induced by the inclusion

$$i \colon R_{\varepsilon_1}(S^1(\tilde{r}_m)) \hookrightarrow R_{\varepsilon_2}(S^1(\tilde{r}_m)).$$

Notice that $P_{m_*}([\nu]) = m[\mu]$. With $1 \leq m \leq N < p$ implying that m is invertible in $\mathbb{F}_p$, we get that $i_* \circ P_{m_*}([\nu]) \neq 0$. From the commutativity of the diagram

$$\begin{tikzcd}
H_1(R_{\varepsilon_1}(\overline{Y}); \mathbb{F}_p) \arrow[r, "i_*"] \arrow[d, "P_{m*}"] & H_1(R_{\varepsilon_2}(\overline{Y}); \mathbb{F}_p) \arrow[d, "P_{m*}"] \\
H_1(R_{\varepsilon_1}(S^1(\tilde{r}_m)); \mathbb{F}_p) \arrow[r, "i_*"] & H_1(R_{\varepsilon_2}(S^1(\tilde{r}_m)); \mathbb{F}_p)
\end{tikzcd}$$

we see that $i_*([\nu]) \neq 0$ and therefore $[\nu]$ gives us an element $\mathbf{x}_\varphi \in \mathrm{dgm}(\overline{Y})$ such that

$$\mathrm{birth}(\mathbf{x}_\varphi) \leq \varepsilon_1 \text{ and } \mathrm{death}(\mathbf{x}_\varphi) \geq \varepsilon_2.$$

Because we chose our epsilons $\delta\kappa_N < \varepsilon_1 < \varepsilon_2 < \sqrt{3}\tilde{r}_m$, the result of the theorem follows by letting $\varepsilon_1 \downarrow \delta\kappa_N$ and letting $\varepsilon_2 \uparrow \sqrt{3}\tilde{r}_m$. □

Notice that in this proof, we could have also worked with the field of rational numbers $\mathbb{Q}$ instead of $\mathbb{F}_p$. With the lower bound on maximum persistence (7.6) and the two convergence theorems (7.1 and 7.3), we can state the following corollary:

Corollary 7.10 *Let $f \in C^1(\mathbb{T})$ be an L-periodic function, such that $\hat{f}(0) = 0$, $\|f\|_2 = 1$ and let $T \subseteq \mathbb{T}$ be finite with $d_H(T, \mathbb{T}) < \delta$ for some*

$$0 < \delta < \frac{\sqrt{3}}{\sqrt{2}\|f'\|_2} \max_{n \in \mathbb{N}} |\hat{f}(n)|.$$

Then the 1*-dimensional persistence diagram* $\mathrm{dgm}_\infty(f, T, w)$ *with coefficients in* $\mathbb{Q}$ *satisfies*

$$\frac{1}{2} mp\Big(\mathrm{dgm}_\infty(f, T, w)\Big) \geq \sqrt{3} \max_{n \in \mathbb{N}} |\hat{f}(n)| - \sqrt{2}\delta\|f'\|_2,$$

and therefore

$$mp\Big(\mathrm{dgm}_\infty(f, w)\Big) \geq 2\sqrt{3} \max_{n \in \mathbb{N}} |\hat{f}(n)|. \tag{7.7}$$

This result connects persistent homology to classical Fourier analysis, showing that the key topological feature (the longest-lived loop in persistent homology) corresponds to the most dominant frequency in the function.

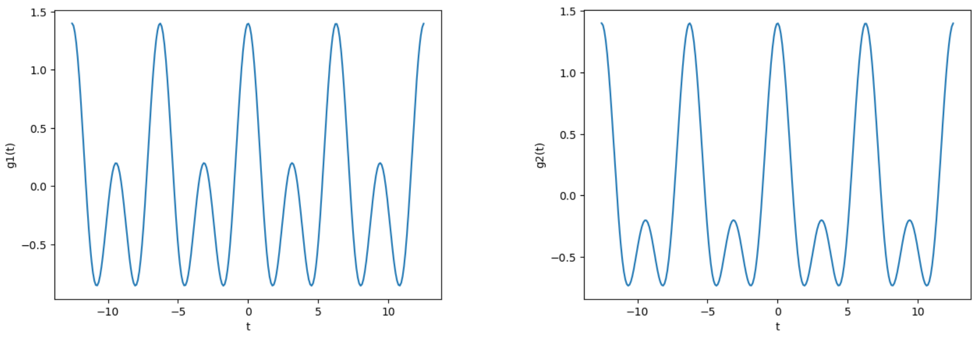
7.3 Dependence on the Field of Coefficients

This section explores how the choice of coefficient field affects persistent homology computations. A key observation is that the embedding forms a Möbius strip-like structure rather than a simple torus.

In Theorem 7.9 we worked over the field $\mathbb{F}_p$. Now an interesting question would be whether the lower bound established in the theorem is, in fact, dependent on the field of coefficients. Moreover, if the persistence diagram changes for different fields. To study these questions we look at the two functions

$$g_1(t) = 0.6\cos(t) + 0.8\cos(2t)$$
$$g_2(t) = 0.8\cos(t) + 0.6\cos(2t),$$

displayed in Figure 7.3. For the construction of their sliding window point



(a) The function $g_1(t)$.

(b) The function $g_2(t)$.

Figure 7.3

clouds, we set our parameters to be the following: $M = 4$, $\tau = \frac{2\pi}{5}$ and $T = \{\frac{2\pi k}{150} | k = 0, 1, \ldots, 150\}$. The next step is to pointwise center and normalize $SW_{M,\tau}g_1(T)$ and $SW_{M,\tau}g_2(T)$. This can be done by a fast implementation of 1-dimensional persistent homology, for some detail, see [21] and [24]. The results of this implementation can be seen in Figure 7.4, where we work over the fields $\mathbb{F}_2$ and $\mathbb{F}_3$. The main difference we see, is between the 1-persistence diagram of g_1 over $\mathbb{F}_2$ and the one over $\mathbb{F}_3$. Over $\mathbb{F}_2$ there is a persistence feature that births at approximately 1.2 and dies at 1.6, whereas it is completely absent over $\mathbb{F}_3$.

This example shows that the persistence diagrams can differ for different fields of coefficients, and therefore also the persistent homology differs. We will now discuss this in more detail. Fix $(r_1, r_2) \in \mathbb{R}^2$ such that $r_1^2 + r_2^2 = 1$ and $r_1 r_2 \neq 0$. Then we can use the third point of the Structure Theorem 6.8 to see that if $\alpha_1, \alpha_2 \in [0, 2\pi]$ and $g(t) = r_1 \cos(t - \alpha_1) + r_2 \cos(2t - \alpha_2)$, then for all $t \in [0, 2\pi]$, for $M \geq 4$ and for $\tau = \frac{2\pi}{M+1}$, we get that

$$\varphi_\tau(t) = \frac{C(SW_{M,\tau}g(t))}{\|C(SW_{M,\tau}g(t))\|}$$

can be isometrically identified with the curve in $\mathbb{C}^2$

$$\tilde{\varphi}(t) = (r_1 e^{it}, r_2 e^{2it}) \in \mathbb{C}^2.$$

This embedding captures the periodic structure by representing the function as a trajectory in a high-dimensional space.

If we now replace φ_τ through $\tilde{\varphi}$ for the persistent homology computation, we see that the image of this embedding can be understood as the boundary of a Möbius strip. To see that, define the map

$$\begin{aligned} \mathcal{M} \colon [0, \pi] \times [-1, 1] &\to \mathbb{C}^2 \\ (t, s) &\mapsto (-s r_1 e^{it}, r_2 e^{2it}). \end{aligned}$$

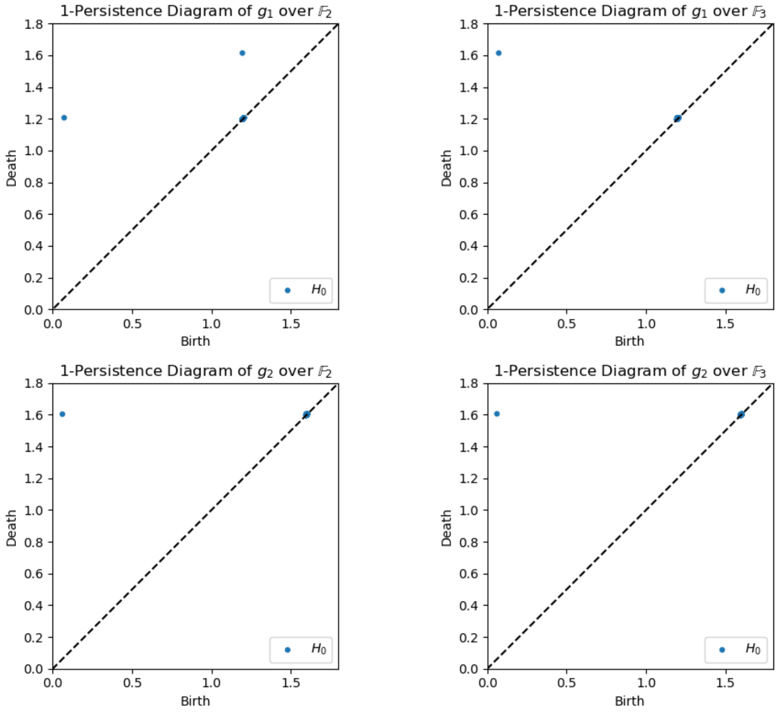


Figure 7.4: Here we see the 1-dimensional persistence diagrams of the centered and normalized sliding window point clouds of g_1 and g_2 over the different fields of coefficients $\mathbb{F}_2$ and $\mathbb{F}_3$. The x-axis denotes the birth-time and the y-axis the death-time. Code for this figure: Appendix, Figure A.7.

This construction reveals that the embedding is not simply a torus but instead has a twist, leading to Moöbius-like behavior. Because we assumed that $r_1 r_2 \neq 0$, $\mathcal{M}$ is a continuous injection onto $[0,\pi) \times [-1,1)$. Moreover, it descends to an embedding of the quotient space

$$\widetilde{\mathcal{M}} \colon ([0,\pi] \times [-1,1]/\sim) \to \mathbb{C}^2,$$

where $(0,s) \sim (\pi,-s)$ for every $s \in [-1,1]$. Notice that the Möbius strip is normally represented as $[0,\pi] \times [-1,1]/\sim$. Furthermore, we see that $\partial(Im(\widetilde{\mathcal{M}})) = Im(\tilde{\varphi})$.

The function $g(t)$ generates a periodic embedding in high-dimensional space, forming a closed loop in homology. The goal is to show that this loop is the boundary of a 2-chain, meaning it can be "filled in" with simplices when using $\mathbb{F}_2$. For that let $T = \{t_0 < t_1 < \ldots < t_J\}$ be a δ-dense subset of $[0,2\pi]$ and $X = \tilde{\varphi}(T)$. Furthermore, for $r > 4\delta$, let $[\nu] \in H_1(R_r(X);\mathbb{F}_2)$ be the homology class of the 1-cycle

$$\nu = [\tilde{\varphi}(t_0), \tilde{\varphi}(t_1)] + \ldots + [\tilde{\varphi}(t_{J-1}), \tilde{\varphi}(t_J)] + [\tilde{\varphi}(t_J), \tilde{\varphi}(t_0)].$$

Since T is δ-dense, the set of points can be used to approximate the full embedding. We define

$$V = \Big\{(t,s)|(t,s) \in (T \cap [0,\pi)) \times \{-1\} \text{ or } (t+\pi,s) \in (T \cap (\pi,2\pi]) \times \{1\}\Big\}.$$

Since T is δ-dense in $[0,2\pi]$, the points in V provide a fine enough sampling for triangulation. The image of $\widetilde{\mathcal{M}}$ forms a surface that can be decomposed into simplices using $\widetilde{\mathcal{M}}(V)$ as vertices. Indeed, $Im(\widetilde{\mathcal{M}})$ resembles a twisted cylinder or an annulus-like surface in the embedding space and a cylinder can always be triangulated by connecting neighboring points with triangles in a structured way. So if we take coefficients in $\mathbb{F}_2$, the formal sum of these triangles yields a 2-chain Σ with $\partial_2(\Sigma) = \nu$. Since T is δ-dense in $[0,2\pi]$ we can bound the variation in the embedding due to discretization for all $t \in [0,\pi]$:

$$\begin{aligned}
||\widetilde{\mathcal{M}}(t,-1) - \widetilde{\mathcal{M}}(t \pm \delta,1)||^2 &= 2[r_1^2(1+\cos(\delta)) + r_2^2(1-\cos(2\delta))] \\
&\leq 2\left[r_1^2\left(2 - \frac{\delta^2}{2}\right) + 2r_2^2\delta^2\right] \\
&= r_1^2(4-5\delta^2) + 4\delta^2.
\end{aligned}$$

So if $\delta > 0$ is small, then we can choose our 2-chain Σ such that

$$\Sigma \in C_2(R_{r'}(X);\mathbb{F}_2) \text{ , where } r' = r_1\sqrt{4-5\delta^2} + 2\delta.$$

The term r' accounts for small perturbations. To summarize this we can say that if

$$r_1\sqrt{4-5\delta^2} + 2\delta < \sqrt{3}r_2, \tag{7.8}$$

then

$$\text{death}([\nu]) \begin{cases} \leq r_1\sqrt{4-5\delta^2} + 2\delta & \text{for coefficients in } \mathbb{F}_2, \\ > \sqrt{3}r_2 & \text{for coefficients in } \mathbb{F}_p \text{ for any prime } p \geq 3. \end{cases}$$

With $\mathbb{F}_2$ coefficients and assuming that Equation (7.8) holds, for example for g_1, the first edge spanning the Möbius band gives rise to a new homology class associated with the equator $t \mapsto \widetilde{\mathcal{M}}(t,0) = (0, r_2e^{2it})$ of the embedded Möbius strip. This class persists until $\sqrt{3}r_2$. However, with $\mathbb{F}_3$ coefficients, the equatorial and boundary classes merge into the same persistence class once all the 2-simplices within the Möbius band are included. Consequently, the later-born class, represented by the equator, vanishes.

Chapter 8

Examples: Quantifying Periodicity of Sampled Signals

Our goal in this chapter is to apply the theoretical framework developed in the sections above, namely the sliding window embeddings and persistent homology, to practical problems with periodic and non-periodic signals.

Definition 8.1 *A **signal** is defined as a physical quantity that varies over time, space or other independent variables.*

Mathematically, we describe a signal as a function of one or more independent variables.

Example 8.2 *The functions*

$$S_1(t) = 2t$$
$$S_2(t) = 2t^3$$

are both signals in one variable.

Many real world signals show periodic or quasi-periodic behavior, and understanding this structure is very important in time series analysis. In particular, we look at two experiments to verify the calculations from the previous chapters. The first one, discussed in Section 8.1, aims to rank signals based purely on their periodicity, independent of the specific shape of their repeating pattern. This means that we do not rely on assumptions about waveform characteristics such as being sinusoidal or non-sinusoidal. In contrast the method evaluates periodicity regarding the shape, using topological techniques. The second one, discussed in Section 8.2, deals with the accurate classification of a signal as periodic or non-periodic at different noise levels.

For that we associate a real valued function f_S to every sampled signal of the form $S = [s_1, \ldots, s_J]$ by applying cubic spline interpolation, then constructing its centered and normalized sliding window point cloud X_S. We let

$$\frac{mp(\text{dgm}(X_S))}{\sqrt{3}} = Score(S) \tag{8.1}$$

be its periodicity score, a measure designed to quantify how periodic a given sampled signal is. These scores are then being compared to the ones obtained with different algorithms from other papers.

8.1 Ranking Signals Based on Periodicity

For this fist experiment, we want to see the differences that we get in score for the different algorithms and for different functions. Therefore, we take ten function, that do not look much alike.

1. 2-periodic pure cosine-curve
2. 2-periodic cosine-like function with variance at 25% of the signal's amplitude
3. 2-periodic cosine-like function with variance at 50% of the signal's amplitude
4. 2-periodic cosine-like function with variance at 75% of the signal's amplitude
5. noisy sawtooth-function with noise level at 25% of the signal's amplitude
6. function $\cos(\phi(t))$ for $\phi(t) = e^{at+b}$
7. noisy damped cosine-like curve with three periods
8. spiky signal with three periods
9. noisy square wave with two periods
10. 1-periodic function $Re\left(\sum_{n=1}^{5} \hat{f}(n)e^{2int}\right)$ for $\hat{f}(n)$ drawn randomly and uniformly from the unit disk in $\mathbb{C}$

Each of these ten functions is then evaluated at 50 evenly spaced time points. This gives us the sampled signals $[s_1, \ldots, s_{50}]$, that we can put in the following four algorithms:

- JTK_CYCLE [16]: This is an algorithm that efficiently identifies and characterizes cycling variables in large datasets. It is very efficient in differentiating between rhythmic and non-rhythmic transcripts and measures the period, phase and amplitude of the transcripts with high accuracy.

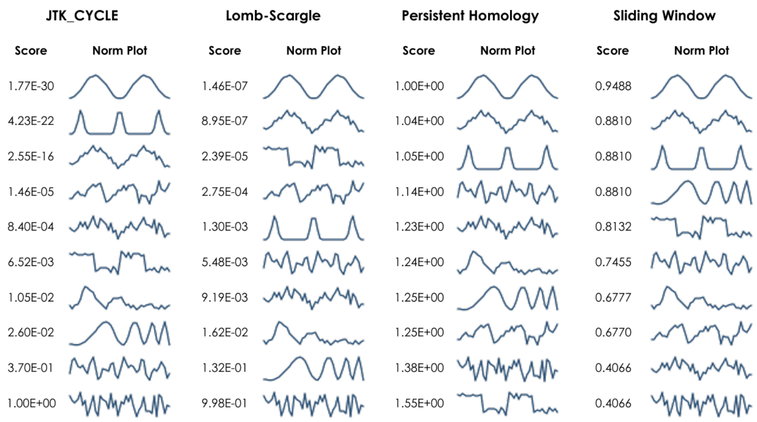


Figure 8.1: Signals are ranked by periodicity in Sliding windows going from highest (top) to lowest (bottom). For every algorithm and signal, the periodicity score and normed plot are displayed. [25]

- Lomb-Scargle periodogram [14]: This algorithm can effectively find periodic gene expression profiles in microarray data. It particularly performs well when a significant proportion of data is missing or when the data is collected at arbitrary time points.
- Total Persistent Homology [10]: This algorithm also searches for periodicity in data. Total persistence denotes the sum of p-th powers of persistence.
- Sliding window (discussed in this paper, based on [25]): With parameters $N = 50$, coefficients in $\mathbb{F}_{11}$, $L = 2, 3, 4$, such that always the best score 8.1 is reported.

If not stated otherwise, parameters for the algorithms are set to suggested or default values. The results for the different functions and algorithms are displayed in Figure 8.1.

Notice that for the JTK_CYCLE, Lomb-Scargle and Persistent Homology algorithms, the scores get bigger from top to bottom and the scores differ relatively more than in the Sliding window algorithm. This is due to the fact that for example JTK_CYCLE and Lomb-Scargle define their periodicity score in terms of p-values, that are difficult to interpret. In contrast to these algorithms, our sliding windows method has a clear geometric interpretation.

8.2 Classification of Periodic and Non-Periodic Signals

The goal of this section is to evaluate and compare different algorithms' ability to distinguish periodic from non-periodic signals. We want to test this on synthetic data. We use the following six signals:

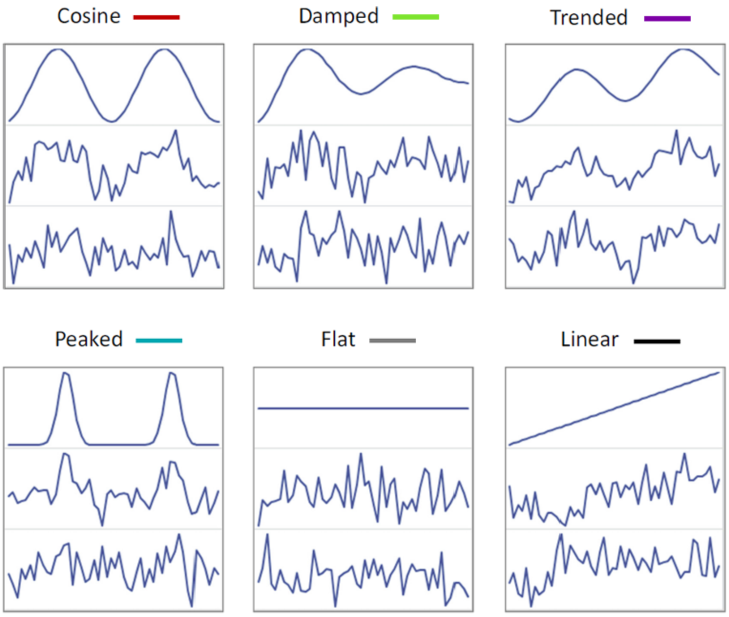


Figure 8.2: Examples of the six different periodic and non-periodic functions. For each signal, the upper third shows the signal with 0%, the middle third the one with 25% and the bottom third the one with 50% Gaussian noise. [25]

1. cosine (periodic, over two periods)
2. cosine with damping (periodic, over two periods)
3. cosine with trending (periodic, over two periods)
4. cosine with increased peak steepness (periodic, over two periods)
5. constant function (non-periodic)
6. linear function (non-periodic)

We will refer to the periodic signals as **positive cases** and to the non-periodic signals as **negative cases**. Then 100 profiles are generated of each shape, by varying the phase. For example, the 100 profiles of the cosine are

$$f_j(t) = \cos\left(2t - \frac{jt}{50}\right), \text{ for } j = 0, \ldots, 99.$$

So in total we get 600 profiles and sample these at 50 evenly spaced time points $t \in [0, 2\pi]$. Then we add a Gaussian noise with standard deviation at 0%, 25% and 50% of the signal's amplitude. See the examples in Figure 8.2.

Remark 8.3 *We treat constant functions as non-periodic, due to two main reasons.* ***Biological Relevance:*** *The primary application of SW1PerS is to identify genes that are relevant and that have a periodic pattern with respect to time. Here,* *by relevant we mean that changes in the expression translate into physiological phenomena.* ***Topological Justification****: The method detects periodicity when there are* 1*-dimensional homology classes in the sliding window point cloud. For a constant function, the sliding window embedding results in a single point. Therefore there is no* 1*-dimensional homology and we can interpret that the point cloud arises from a non-periodic function.*

We use Receiver Operator Characteristic (ROC) plots to assess the performance of various methods, including the Sliding Windows and 1-Persistent Scoring (SW1PerS) approach. By using the True Positive Rate (TPR) and False Positive Rate (FPR), the ROC curve provides a quantitative metric for classification success. Note that the TPR is the proportion of correctly identified positive cases out of all positives and the FPR is the proportion of negative cases incorrectly identified as positives out of all the negatives. The diagonal, where TPR=FPR, represents the case of a random guess. So the higher the ROC curve is above this line, the better it performs in classification. The best case would therefore be a ROC curve, that passes through TPR= 1 and FPR= 0. So the bigger the area under the ROC curve is, the better the method performs. We name this area the Area Under the Curve (AUC). It serves as a summary statistic to compare different methods.

We now have to choose parameters for the Sliding Windows approach. We let $N = 10$, $L = 2$ and use coefficients in $\mathbb{F}_{11}$. To deal with eventual noise, we include a layer of moving average at the sampled signal level and one iteration of mean-shift [11] at the sliding window point cloud level.

Definition 8.4 ***Mean-Shift*** *takes a pointwise mean-centered and normalized point cloud, lying on the surface of the unit sphere in* $\mathbb{R}^{M+1}$.[1] *It then measures the distance between two points* $\mathbf{x}, \mathbf{y}$ *in the point cloud via the angle between them. Two points are now considered* ***close****, if* $\measuredangle(\mathbf{x}, \mathbf{y}) < \frac{\pi}{16}$*. It then replaces each point by the average of its close neighbors.*

To smooth out small fluctuations and to reduce noise in the time series data, we apply a moving average filter with a window size of 7 data points. After denoising the signal, we fit a cubic spline onto it to populate the point cloud. A cubic spline is a polynomial, that is used to smoothly interpolate data points. It is called cubic because every term of the spline is a polynomial of order three, so it has the form:

$$S(x) = ax^3 + bx^2 + cx + d.$$

This spline representation is then used to construct the sliding window point cloud. We apply mean-shift to the centered and normalized sliding window point cloud X_S. This is done as follows: Given a point $x \in X_S$, we compute

[1] M is a positive integer, closer defined in the following section

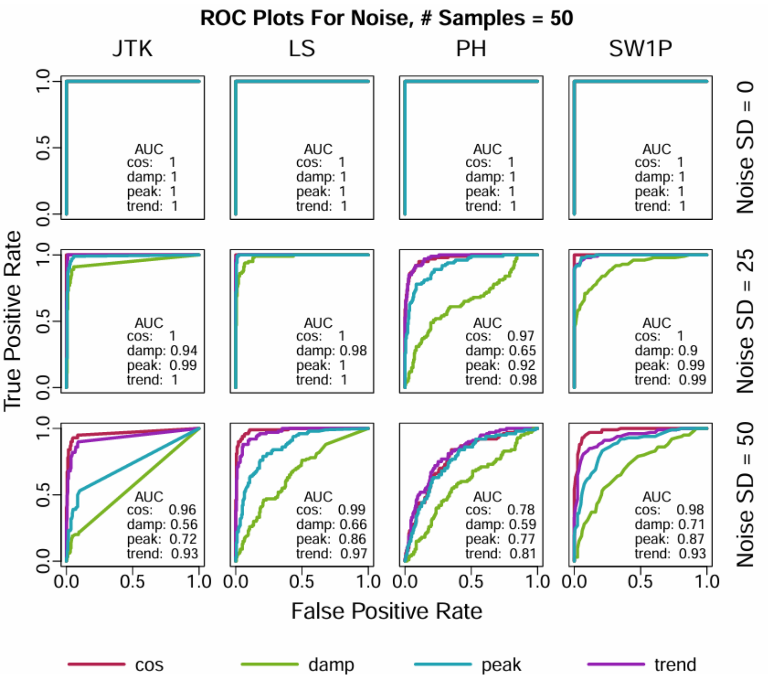


Figure 8.3: The ROC plots for the four different algorithms and the four periodic signals. The FPR is displayed at the x-axis, the TPR at the y-axis. The AUC is reported at the bottom right of every plot.[25]

its mean shift using its cosine-similarity neighbors [19]. Cosine similarity measures how similar two vectors are based on the angle between them. It is defined as:

$$\text{cosine similarity}(x, y) = \frac{x \cdot y}{\|x\| \cdot \|y\|}.$$

Specifically, we define the set of neighboring points as

$$\{y \in X_S | 1 - (x \cdot y) < \varepsilon\}\text{ , where } \varepsilon = \cos\left(\frac{\pi}{16}\right).$$

This means that we consider all points whose cosine similarity to x is above a certain threshold. We define a new representative point $\overline{x}$ for x as the mean of its ε-neighborhood. To ensure uniform scaling, we normalize each mean-shifted point by dividing it by its norm and get the mean-shifted point cloud

$$\overline{X}_S = \left\{\frac{\overline{x}}{\|\overline{x}\|} | x \in X_S\right\}.$$

We can now use this point cloud for the persistent homology computation. The results of the various ROC plots can be found in Figure 8.3.

We see that all signals perform the same with no noise, but we see significant differences in noise levels 25% and 50%. Notice, that the Lomb-Scargle periodogram is known for being one of the best approaches to detect

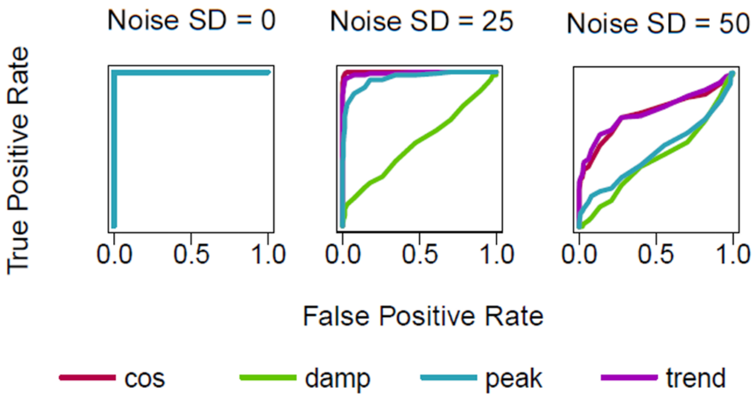


Figure 8.4: ROC plots for SW1PerS with non-denoised data. [25]

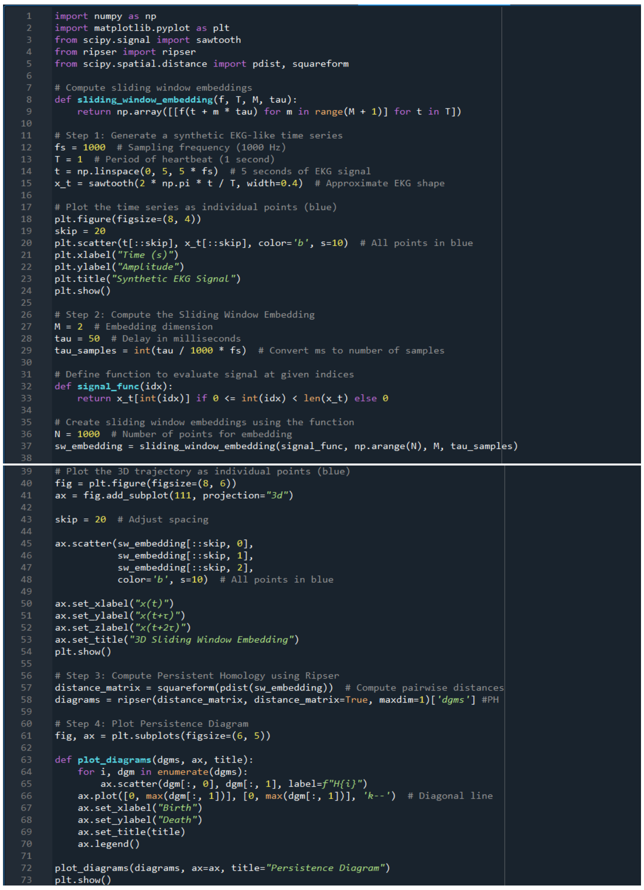
periodicity and that the values of the SW1PerS are pretty similar. In high noise levels, it even outperforms every other approach for damped cosine functions and cosine functions with increased peak steepness. For the cosine function and the trending cosine, it is only outperformed by the Lomb-Scargle periodogram.

One might wonder, whether the denoising part in our above steps the preprocess the data is really necessary, thus whether we can skip the application of moving average or mean-shift. In Figure 8.4, we see the ROC plots for our four periodic functions at different noise levels. It becomes clear, that our SW1PerS approach performs not near as good as with the denoised data.

Appendix A

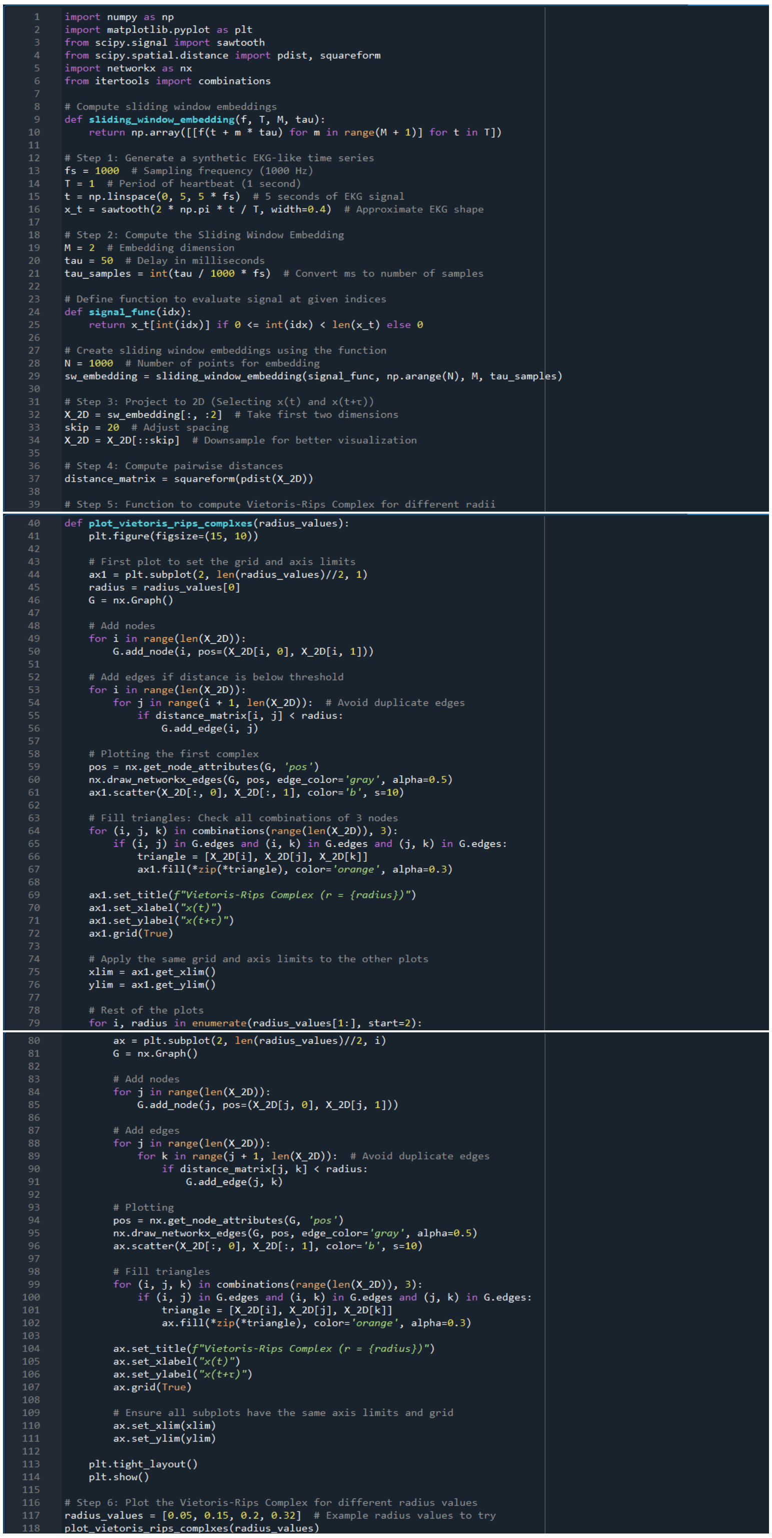
Appendix

To support with the coding in python, AI [22] was used for some parts of the codes.



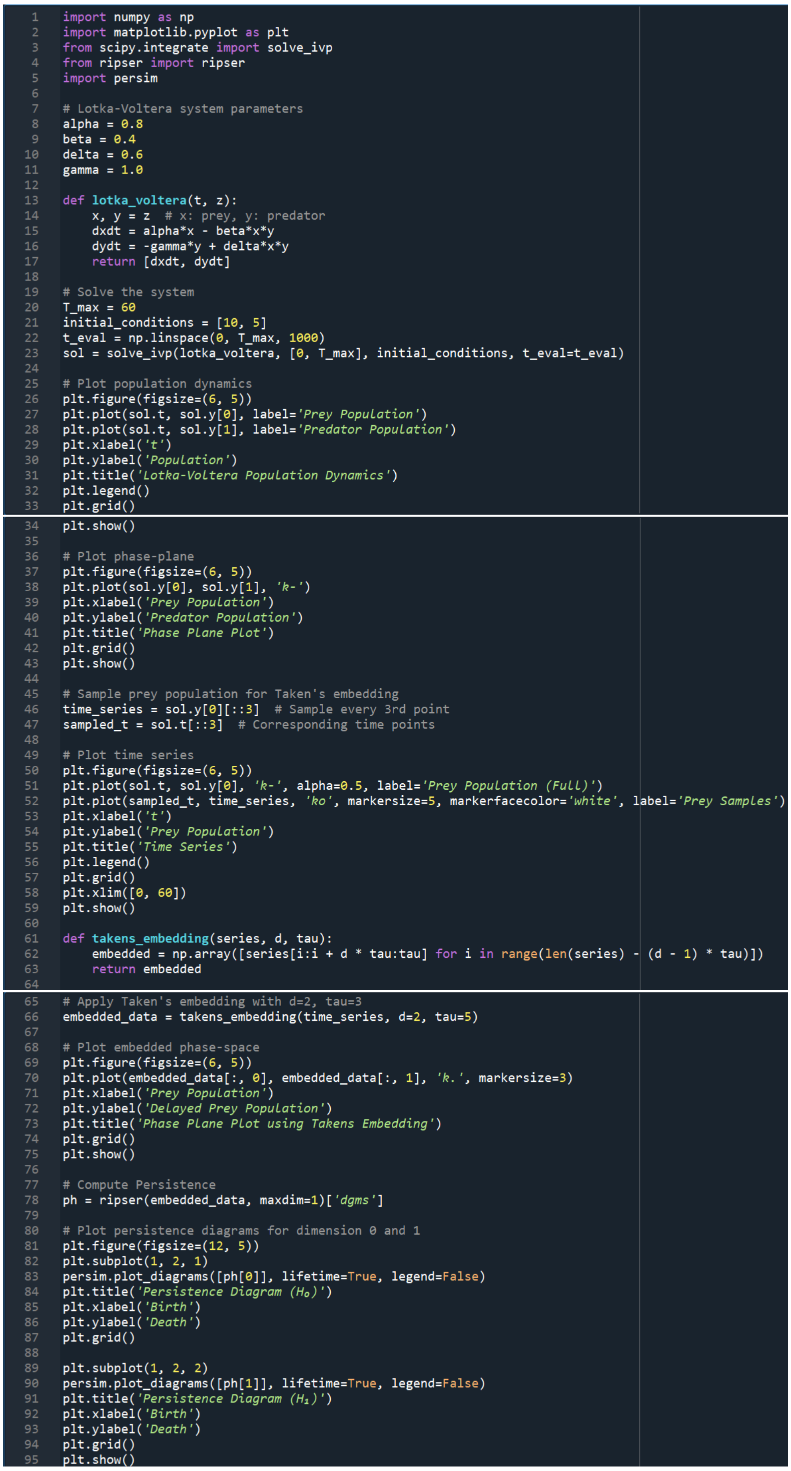
```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import sawtooth
from ripser import ripser
from scipy.spatial.distance import pdist, squareform

# Compute sliding window embeddings
def sliding_window_embedding(f, T, M, tau):
    return np.array([[f(t + m * tau) for m in range(M + 1)] for t in T])

# Step 1: Generate a synthetic EKG-like time series
fs = 1000  # Sampling frequency (1000 Hz)
T = 1  # Period of heartbeat (1 second)
t = np.linspace(0, 5, 5 * fs)  # 5 seconds of EKG signal
x_t = sawtooth(2 * np.pi * t / T, width=0.4)  # Approximate EKG shape

# Plot the time series as individual points (blue)
plt.figure(figsize=(8, 4))
skip = 20
plt.scatter(t[::skip], x_t[::skip], color='b', s=10)  # All points in blue
plt.xlabel("Time (s)")
plt.ylabel("Amplitude")
plt.title("Synthetic EKG Signal")
plt.show()

# Step 2: Compute the Sliding Window Embedding
M = 2  # Embedding dimension
tau = 50  # Delay in milliseconds
tau_samples = int(tau / 1000 * fs)  # Convert ms to number of samples

# Define function to evaluate signal at given indices
def signal_func(idx):
    return x_t[int(idx)] if 0 <= int(idx) < len(x_t) else 0

# Create sliding window embeddings using the function
N = 1000  # Number of points for embedding
sw_embedding = sliding_window_embedding(signal_func, np.arange(N), M, tau_samples)

# Plot the 3D trajectory as individual points (blue)
fig = plt.figure(figsize=(8, 6))
ax = fig.add_subplot(111, projection="3d")

skip = 20  # Adjust spacing

ax.scatter(sw_embedding[::skip, 0],
           sw_embedding[::skip, 1],
           sw_embedding[::skip, 2],
           color='b', s=10)  # All points in blue

ax.set_xlabel("x(t)")
ax.set_ylabel("x(t+τ)")
ax.set_zlabel("x(t+2τ)")
ax.set_title("3D Sliding Window Embedding")
plt.show()

# Step 3: Compute Persistent Homology using Ripser
distance_matrix = squareform(pdist(sw_embedding))  # Compute pairwise distances
diagrams = ripser(distance_matrix, distance_matrix=True, maxdim=1)['dgms'] #PH

# Step 4: Plot Persistence Diagram
fig, ax = plt.subplots(figsize=(6, 5))

def plot_diagrams(dgms, ax, title):
    for i, dgm in enumerate(dgms):
        ax.scatter(dgm[:, 0], dgm[:, 1], label=f"H{i}")
    ax.plot([0, max(dgm[:, 1])], [0, max(dgm[:, 1])], 'k--')  # Diagonal line
    ax.set_xlabel("Birth")
    ax.set_ylabel("Death")
    ax.set_title(title)
    ax.legend()

plot_diagrams(diagrams, ax=ax, title="Persistence Diagram")
plt.show()
```

Figure A.1: Code for Figures 1.1, 1.2 and 1.4.



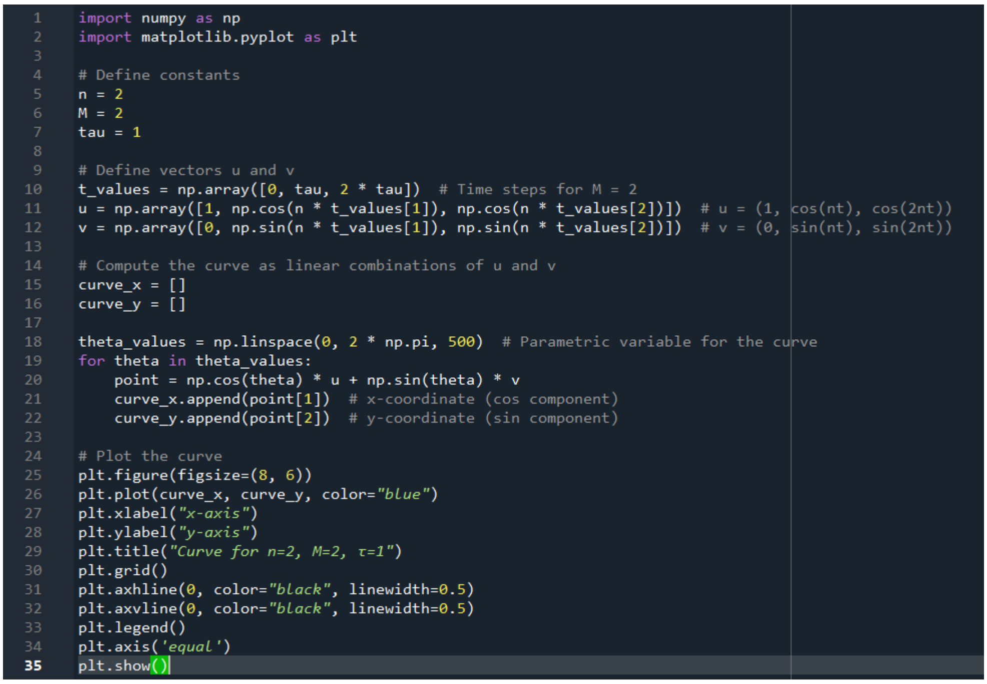
```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import sawtooth
from scipy.spatial.distance import pdist, squareform
import networkx as nx
from itertools import combinations

# Compute sliding window embeddings
def sliding_window_embedding(f, T, M, tau):
    return np.array([[f(t + m * tau) for m in range(M + 1)] for t in T])

# Step 1: Generate a synthetic EKG-like time series
fs = 1000  # Sampling frequency (1000 Hz)
T = 1  # Period of heartbeat (1 second)
t = np.linspace(0, 5, 5 * fs)  # 5 seconds of EKG signal
x_t = sawtooth(2 * np.pi * t / T, width=0.4)  # Approximate EKG shape

# Step 2: Compute the Sliding Window Embedding
M = 2  # Embedding dimension
tau = 50  # Delay in milliseconds
tau_samples = int(tau / 1000 * fs)  # Convert ms to number of samples

# Define function to evaluate signal at given indices
def signal_func(idx):
    return x_t[int(idx)] if 0 <= int(idx) < len(x_t) else 0

# Create sliding window embeddings using the function
N = 1000  # Number of points for embedding
sw_embedding = sliding_window_embedding(signal_func, np.arange(N), M, tau_samples)

# Step 3: Project to 2D (Selecting x(t) and x(t+τ))
X_2D = sw_embedding[:, :2]  # Take first two dimensions
skip = 20  # Adjust spacing
X_2D = X_2D[::skip]  # Downsample for better visualization

# Step 4: Compute pairwise distances
distance_matrix = squareform(pdist(X_2D))

# Step 5: Function to compute Vietoris-Rips Complex for different radii
def plot_vietoris_rips_complxes(radius_values):
    plt.figure(figsize=(15, 10))

    # First plot to set the grid and axis limits
    ax1 = plt.subplot(2, len(radius_values)//2, 1)
    radius = radius_values[0]
    G = nx.Graph()

    # Add nodes
    for i in range(len(X_2D)):
        G.add_node(i, pos=(X_2D[i, 0], X_2D[i, 1]))

    # Add edges if distance is below threshold
    for i in range(len(X_2D)):
        for j in range(i + 1, len(X_2D)):  # Avoid duplicate edges
            if distance_matrix[i, j] < radius:
                G.add_edge(i, j)

    # Plotting the first complex
    pos = nx.get_node_attributes(G, 'pos')
    nx.draw_networkx_edges(G, pos, edge_color='gray', alpha=0.5)
    ax1.scatter(X_2D[:, 0], X_2D[:, 1], color='b', s=10)

    # Fill triangles: Check all combinations of 3 nodes
    for (i, j, k) in combinations(range(len(X_2D)), 3):
        if (i, j) in G.edges and (i, k) in G.edges and (j, k) in G.edges:
            triangle = [X_2D[i], X_2D[j], X_2D[k]]
            ax1.fill(*zip(*triangle), color='orange', alpha=0.3)

    ax1.set_title(f"Vietoris-Rips Complex (r = {radius})")
    ax1.set_xlabel("x(t)")
    ax1.set_ylabel("x(t+τ)")
    ax1.grid(True)

    # Apply the same grid and axis limits to the other plots
    xlim = ax1.get_xlim()
    ylim = ax1.get_ylim()

    # Rest of the plots
    for i, radius in enumerate(radius_values[1:], start=2):
        ax = plt.subplot(2, len(radius_values)//2, i)
        G = nx.Graph()

        # Add nodes
        for j in range(len(X_2D)):
            G.add_node(j, pos=(X_2D[j, 0], X_2D[j, 1]))

        # Add edges
        for j in range(len(X_2D)):
            for k in range(j + 1, len(X_2D)):  # Avoid duplicate edges
                if distance_matrix[j, k] < radius:
                    G.add_edge(j, k)

        # Plotting
        pos = nx.get_node_attributes(G, 'pos')
        nx.draw_networkx_edges(G, pos, edge_color='gray', alpha=0.5)
        ax.scatter(X_2D[:, 0], X_2D[:, 1], color='b', s=10)

        # Fill triangles
        for (i, j, k) in combinations(range(len(X_2D)), 3):
            if (i, j) in G.edges and (i, k) in G.edges and (j, k) in G.edges:
                triangle = [X_2D[i], X_2D[j], X_2D[k]]
                ax.fill(*zip(*triangle), color='orange', alpha=0.3)

        ax.set_title(f"Vietoris-Rips Complex (r = {radius})")
        ax.set_xlabel("x(t)")
        ax.set_ylabel("x(t+τ)")
        ax.grid(True)

        # Ensure all subplots have the same axis limits and grid
        ax.set_xlim(xlim)
        ax.set_ylim(ylim)

    plt.tight_layout()
    plt.show()

# Step 6: Plot the Vietoris-Rips Complex for different radius values
radius_values = [0.05, 0.15, 0.2, 0.32]  # Example radius values to try
plot_vietoris_rips_complxes(radius_values)
```

Figure A.2: Code for Figure 1.3.



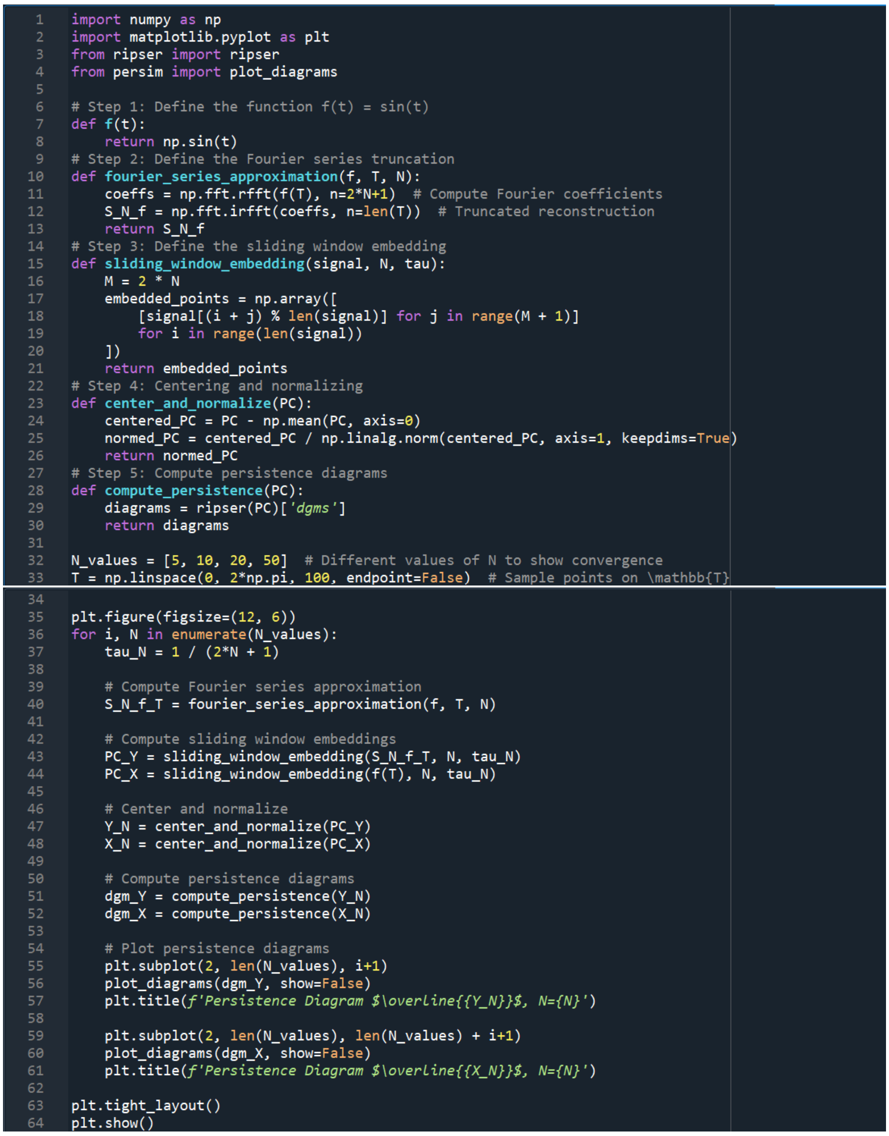
```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import solve_ivp
from ripser import ripser
import persim

# Lotka-Volterra system parameters
alpha = 0.8
beta = 0.4
delta = 0.6
gamma = 1.0

def lotka_volterra(t, z):
    x, y = z  # x: prey, y: predator
    dxdt = alpha*x - beta*x*y
    dydt = -gamma*y + delta*x*y
    return [dxdt, dydt]

# Solve the system
T_max = 60
initial_conditions = [10, 5]
t_eval = np.linspace(0, T_max, 1000)
sol = solve_ivp(lotka_voltera, [0, T_max], initial_conditions, t_eval=t_eval)

# Plot population dynamics
plt.figure(figsize=(6, 5))
plt.plot(sol.t, sol.y[0], label='Prey Population')
plt.plot(sol.t, sol.y[1], label='Predator Population')
plt.xlabel('t')
plt.ylabel('Population')
plt.title('Lotka-Volterra Population Dynamics')
plt.legend()
plt.grid()
plt.show()

# Plot phase-plane
plt.figure(figsize=(6, 5))
plt.plot(sol.y[0], sol.y[1], 'k-')
plt.xlabel('Prey Population')
plt.ylabel('Predator Population')
plt.title('Phase Plane Plot')
plt.grid()
plt.show()

# Sample prey population for Taken's embedding
time_series = sol.y[0][::3]  # Sample every 3rd point
sampled_t = sol.t[::3]  # Corresponding time points

# Plot time series
plt.figure(figsize=(6, 5))
plt.plot(sol.t, sol.y[0], 'k-', alpha=0.5, label='Prey Population (Full)')
plt.plot(sampled_t, time_series, 'ko', markersize=5, markerfacecolor='white', label='Prey Samples')
plt.xlabel('t')
plt.ylabel('Prey Population')
plt.title('Time Series')
plt.legend()
plt.grid()
plt.xlim([0, 60])
plt.show()

def takens_embedding(series, d, tau):
    embedded = np.array([series[i:i + d * tau:tau] for i in range(len(series) - (d - 1) * tau)])
    return embedded

# Apply Taken's embedding with d=2, tau=3
embedded_data = takens_embedding(time_series, d=2, tau=5)

# Plot embedded phase-space
plt.figure(figsize=(6, 5))
plt.plot(embedded_data[:, 0], embedded_data[:, 1], 'k.', markersize=3)
plt.xlabel('Prey Population')
plt.ylabel('Delayed Prey Population')
plt.title('Phase Plane Plot using Takens Embedding')
plt.grid()
plt.show()

# Compute Persistence
ph = ripser(embedded_data, maxdim=1)['dgms']

# Plot persistence diagrams for dimension 0 and 1
plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
persim.plot_diagrams([ph[0]], lifetime=True, legend=False)
plt.title('Persistence Diagram (H₀)')
plt.xlabel('Birth')
plt.ylabel('Death')
plt.grid()

plt.subplot(1, 2, 2)
persim.plot_diagrams([ph[1]], lifetime=True, legend=False)
plt.title('Persistence Diagram (H₁)')
plt.xlabel('Birth')
plt.ylabel('Death')
plt.grid()
plt.show()
```

Figure A.3: Code for Figures in Example 3.9.



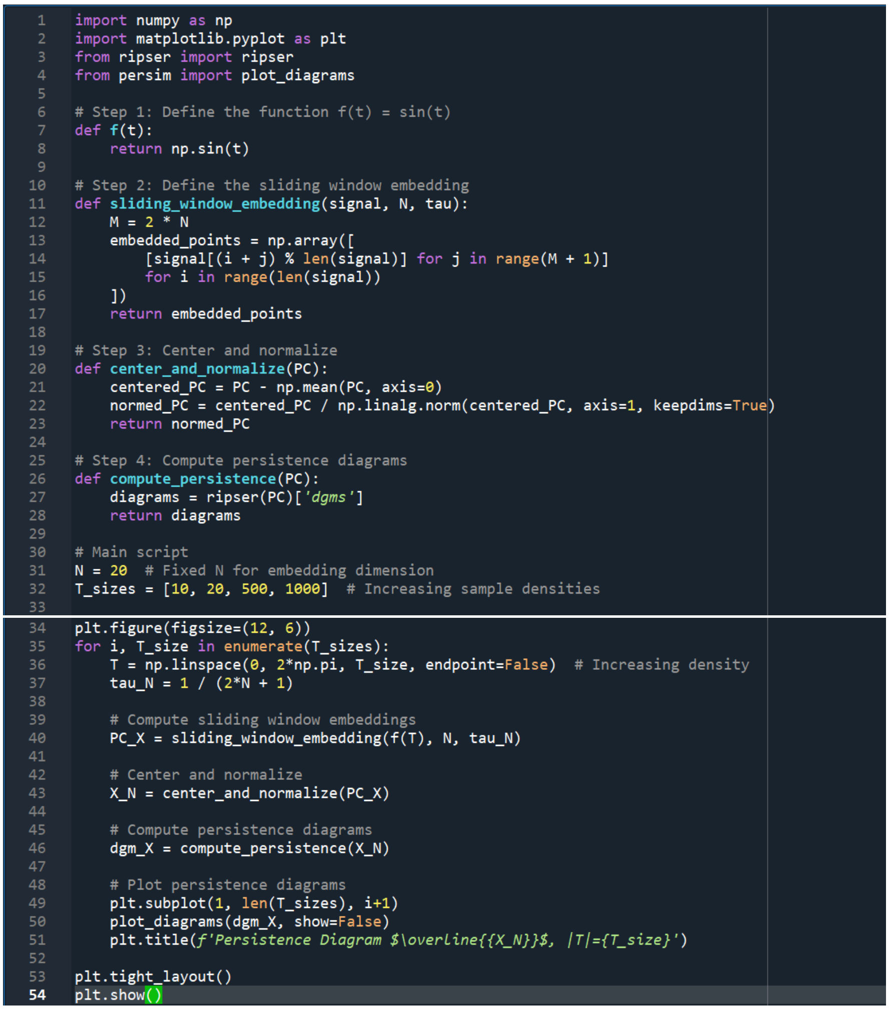
```python
import numpy as np
import matplotlib.pyplot as plt

# Define constants
n = 2
M = 2
tau = 1

# Define vectors u and v
t_values = np.array([0, tau, 2 * tau])  # Time steps for M = 2
u = np.array([1, np.cos(n * t_values[1]), np.cos(n * t_values[2])])  # u = (1, cos(nt), cos(2nt))
v = np.array([0, np.sin(n * t_values[1]), np.sin(n * t_values[2])])  # v = (0, sin(nt), sin(2nt))

# Compute the curve as linear combinations of u and v
curve_x = []
curve_y = []

theta_values = np.linspace(0, 2 * np.pi, 500)  # Parametric variable for the curve
for theta in theta_values:
    point = np.cos(theta) * u + np.sin(theta) * v
    curve_x.append(point[1])  # x-coordinate (cos component)
    curve_y.append(point[2])  # y-coordinate (sin component)

# Plot the curve
plt.figure(figsize=(8, 6))
plt.plot(curve_x, curve_y, color="blue")
plt.xlabel("x-axis")
plt.ylabel("y-axis")
plt.title("Curve for n=2, M=2, τ=1")
plt.grid()
plt.axhline(0, color="black", linewidth=0.5)
plt.axvline(0, color="black", linewidth=0.5)
plt.legend()
plt.axis('equal')
plt.show()
```

Figure A.4: Code for Figure 4.2.



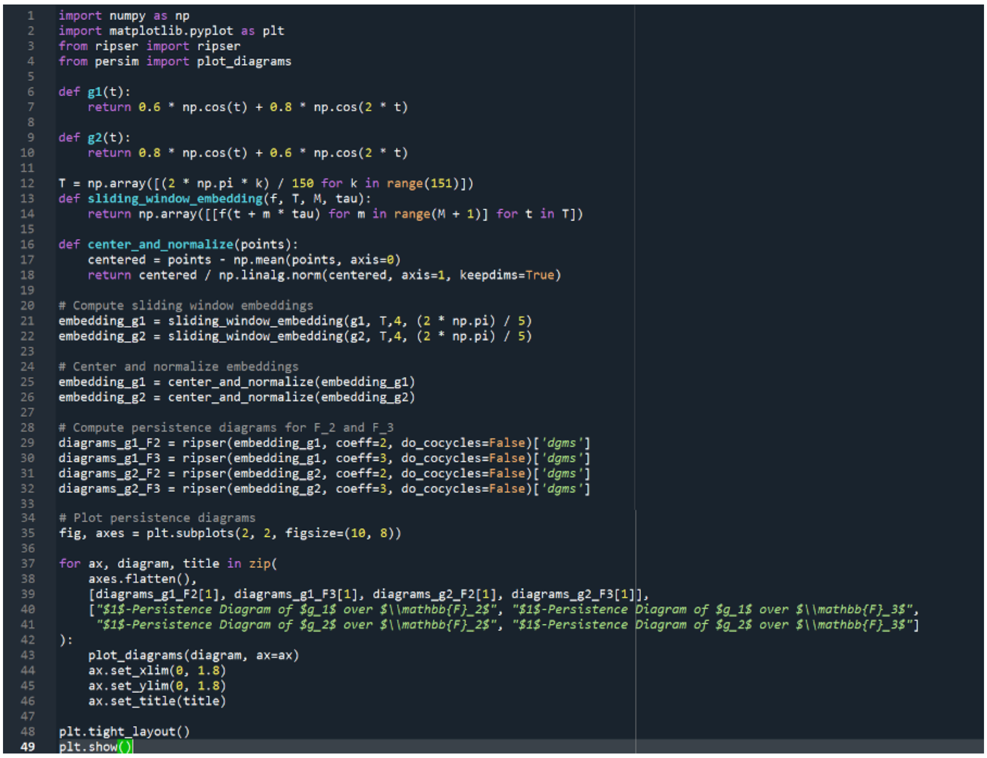
```python
import numpy as np
import matplotlib.pyplot as plt
from ripser import ripser
from persim import plot_diagrams

# Step 1: Define the function f(t) = sin(t)
def f(t):
    return np.sin(t)
# Step 2: Define the Fourier series truncation
def fourier_series_approximation(f, T, N):
    coeffs = np.fft.rfft(f(T), n=2*N+1)  # Compute Fourier coefficients
    S_N_f = np.fft.irfft(coeffs, n=len(T))  # Truncated reconstruction
    return S_N_f
# Step 3: Define the sliding window embedding
def sliding_window_embedding(signal, N, tau):
    M = 2 * N
    embedded_points = np.array([
        [signal[(i + j) % len(signal)] for j in range(M + 1)]
        for i in range(len(signal))
    ])
    return embedded_points
# Step 4: Centering and normalizing
def center_and_normalize(PC):
    centered_PC = PC - np.mean(PC, axis=0)
    normed_PC = centered_PC / np.linalg.norm(centered_PC, axis=1, keepdims=True)
    return normed_PC
# Step 5: Compute persistence diagrams
def compute_persistence(PC):
    diagrams = ripser(PC)['dgms']
    return diagrams

N_values = [5, 10, 20, 50]  # Different values of N to show convergence
T = np.linspace(0, 2*np.pi, 100, endpoint=False)  # Sample points on \mathbb{T}

plt.figure(figsize=(12, 6))
for i, N in enumerate(N_values):
    tau_N = 1 / (2*N + 1)

    # Compute Fourier series approximation
    S_N_f_T = fourier_series_approximation(f, T, N)

    # Compute sliding window embeddings
    PC_Y = sliding_window_embedding(S_N_f_T, N, tau_N)
    PC_X = sliding_window_embedding(f(T), N, tau_N)

    # Center and normalize
    Y_N = center_and_normalize(PC_Y)
    X_N = center_and_normalize(PC_X)

    # Compute persistence diagrams
    dgm_Y = compute_persistence(Y_N)
    dgm_X = compute_persistence(X_N)

    # Plot persistence diagrams
    plt.subplot(2, len(N_values), i+1)
    plot_diagrams(dgm_Y, show=False)
    plt.title(f'Persistence Diagram $\overline{{Y_N}}$, N={N}')

    plt.subplot(2, len(N_values), len(N_values) + i+1)
    plot_diagrams(dgm_X, show=False)
    plt.title(f'Persistence Diagram $\overline{{X_N}}$, N={N}')

plt.tight_layout()
plt.show()
```

Figure A.5: Code for Figure 7.1.



```python
import numpy as np
import matplotlib.pyplot as plt
from ripser import ripser
from persim import plot_diagrams

# Step 1: Define the function f(t) = sin(t)
def f(t):
    return np.sin(t)

# Step 2: Define the sliding window embedding
def sliding_window_embedding(signal, N, tau):
    M = 2 * N
    embedded_points = np.array([
        [signal[(i + j) % len(signal)] for j in range(M + 1)]
        for i in range(len(signal))
    ])
    return embedded_points

# Step 3: Center and normalize
def center_and_normalize(PC):
    centered_PC = PC - np.mean(PC, axis=0)
    normed_PC = centered_PC / np.linalg.norm(centered_PC, axis=1, keepdims=True)
    return normed_PC

# Step 4: Compute persistence diagrams
def compute_persistence(PC):
    diagrams = ripser(PC)['dgms']
    return diagrams

# Main script
N = 20  # Fixed N for embedding dimension
T_sizes = [10, 20, 500, 1000]  # Increasing sample densities

plt.figure(figsize=(12, 6))
for i, T_size in enumerate(T_sizes):
    T = np.linspace(0, 2*np.pi, T_size, endpoint=False)  # Increasing density
    tau_N = 1 / (2*N + 1)

    # Compute sliding window embeddings
    PC_X = sliding_window_embedding(f(T), N, tau_N)

    # Center and normalize
    X_N = center_and_normalize(PC_X)

    # Compute persistence diagrams
    dgm_X = compute_persistence(X_N)

    # Plot persistence diagrams
    plt.subplot(1, len(T_sizes), i+1)
    plot_diagrams(dgm_X, show=False)
    plt.title(f'Persistence Diagram $\overline{{X_N}}$, |T|={T_size}')

plt.tight_layout()
plt.show()
```

Figure A.6: Code for Figure 7.2.



```python
import numpy as np
import matplotlib.pyplot as plt
from ripser import ripser
from persim import plot_diagrams

def g1(t):
    return 0.6 * np.cos(t) + 0.8 * np.cos(2 * t)

def g2(t):
    return 0.8 * np.cos(t) + 0.6 * np.cos(2 * t)

T = np.array([(2 * np.pi * k) / 150 for k in range(151)])
def sliding_window_embedding(f, T, M, tau):
    return np.array([[f(t + m * tau) for m in range(M + 1)] for t in T])

def center_and_normalize(points):
    centered = points - np.mean(points, axis=0)
    return centered / np.linalg.norm(centered, axis=1, keepdims=True)

# Compute sliding window embeddings
embedding_g1 = sliding_window_embedding(g1, T,4, (2 * np.pi) / 5)
embedding_g2 = sliding_window_embedding(g2, T,4, (2 * np.pi) / 5)

# Center and normalize embeddings
embedding_g1 = center_and_normalize(embedding_g1)
embedding_g2 = center_and_normalize(embedding_g2)

# Compute persistence diagrams for F_2 and F_3
diagrams_g1_F2 = ripser(embedding_g1, coeff=2, do_cocycles=False)['dgms']
diagrams_g1_F3 = ripser(embedding_g1, coeff=3, do_cocycles=False)['dgms']
diagrams_g2_F2 = ripser(embedding_g2, coeff=2, do_cocycles=False)['dgms']
diagrams_g2_F3 = ripser(embedding_g2, coeff=3, do_cocycles=False)['dgms']

# Plot persistence diagrams
fig, axes = plt.subplots(2, 2, figsize=(10, 8))

for ax, diagram, title in zip(
    axes.flatten(),
    [diagrams_g1_F2[1], diagrams_g1_F3[1], diagrams_g2_F2[1], diagrams_g2_F3[1]],
    ["$1$-Persistence Diagram of $g_1$ over $\\mathbb{F}_2$", "$1$-Persistence Diagram of $g_1$ over $\\mathbb{F}_3$",
     "$1$-Persistence Diagram of $g_2$ over $\\mathbb{F}_2$", "$1$-Persistence Diagram of $g_2$ over $\\mathbb{F}_3$"]
):
    plot_diagrams(diagram, ax=ax)
    ax.set_xlim(0, 1.8)
    ax.set_ylim(0, 1.8)
    ax.set_title(title)

plt.tight_layout()
plt.show()
```

Figure A.7: Code for Figure 7.4.

Bibliography

[1] Andrew Accardi, Robert Miller, and James Holmes. Enhanced diagnosis of narrow complex tachycardias with increased electrocardiograph speed 1 1 original contributions is coordinated by john a. marx, md, of carolinas medical center, charlotte, north carolina. *Journal of Emergency Medicine - J EMERG MED*, 22:123–126, 01 2002.

[2] Daniel Becker. Fundamentals of electrocardiography interpretation. *Anesthesia progress*, 53:53–63; quiz 64, 02 2006.

[3] Andrew J. Blumberg, Itamar Gal, Michael A. Mandell, and Matthew Pancia. Robust statistics, hypothesis testing, and confidence intervals for persistent homology on metric measure spaces, 2014.

[4] Peter J. Brockwell and Richard A. Davis. *Introduction to Time Series and Forecasting*. Springer Cham, 2016.

[5] Gunnar Carlsson. Topology and data. *Bulletin of The American Mathematical Society - BULL AMER MATH SOC*, 46:255–308, 04 2009.

[6] Gunnar Carlsson. Topological pattern recognition for point cloud data. *Acta Numerica*, 23:289–368, 2014.

[7] Frederic Chazal, Vin de Silva, and Steve Oudot. Persistence stability for geometric complexes, 2013.

[8] Renjie Chen, Jingyue Zhang, Nalini Ravishanker, and Karthik Konduri. Clustering activity-travel behavior time series using topological data analysis, 2019.

[9] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete and Computational Geometry - DCG*, 37:263–271, 06 2005.

[10] David Cohen-Steiner, Herbert Edelsbrunner, John Harer, and Yuriy Mileyko. Lipschitz functions have lp-stable persistence. *Foundations of Computational Mathematics*, 10:127–139, 02 2010.

[11] Dorin Comaniciu and Peter Meer. Meer, p.: Mean shift: A robust approach toward feature space analysis. ieee transactions on pattern analysis and machine intelligence 24(5), 603-619. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24:603 – 619, 06 2002.

[12] Brittany Terese Fasy, Jisu Kim, Fabrizio Lecci, and Clément Maria. Introduction to the R package TDA. *CoRR*, abs/1411.1830, 2014.

[13] Renato Fiorenza. *Hölder and locally Hölder Continuous Functions, and Open Sets of Class $C^k, C^{k,\lambda}$*. Birkhäuser Cham, 01 2016.

[14] Earl Glynn, Jie Chen, and Arcady Mushegian. Detecting periodic patterns in unevenly spaced gene expression time series using lomb-scargle periodograms. *Bioinformatics (Oxford, England)*, 22:310–6, 03 2006.

[15] Allen Hatcher. *Algebraic Topology*. Cambridge Universisty Press, 2001.

[16] Michael Hughes, John Hogenesch, and Karl Kornacker. Jtk_cycle: An efficient nonparametric algorithm for detecting rhythmic components in genome-scale data sets. *Journal of Biological Rhythms*, 25:372–380, 10 2010.

[17] Holger Kantz and Thomas Schreiber. *Phase space methods*, page 30–47. Cambridge University Press, 2003.

[18] H.S. Kim, R. Eykholt, and J.D. Salas. Nonlinear dynamics, delay times, and embedding windows. *Physica D: Nonlinear Phenomena*, 127(1):48–60, 1999.

[19] Vijay Kotu and Bala Deshpande. Chapter 4 - classification. In Vijay Kotu and Bala Deshpande, editors, *Data Science (Second Edition)*, pages 65–163. Morgan Kaufmann, second edition edition, 2019.

[20] Yuriy Mileyko, Sayan Mukherjee, and John Harer. Probability measures on the space of persistence diagrams. *Inverse Problems*, 27(12):124007, nov 2011.

[21] Konstantin Mischaikow and Vidit Nanda. Morse theory for filtrations and efficient computation of persistent homology. *Discrete and Computational Geometry*, 50, 09 2013.

[22] OpenAI. Chatgpt: A large language model, 2025. 2025.

[23] Jose Perea, Anastasia Deckard, Steven Haase, and John Harer. Sw1pers: Sliding windows and 1-persistence scoring; discovering periodicity in gene expression time series data. *BMC bioinformatics*, 16:257, 08 2015.

[24] Jose Perea, Anastasia Deckard, Steven Haase, and John Harer. Sw1pers: Sliding windows and 1-persistence scoring; discovering periodicity in gene expression time series data. *BMC bioinformatics*, 16:257, 08 2015.

[25] Jose Perea and John Harer. Sliding windows and persistence: An application of topological methods to signal analysis, 2013.

[26] M.A. Pinsky. *Introduction to Fourier Analysis and Wavelets*. Graduate studies in mathematics. American Mathematical Society, 2008.

[27] Floris Takens. Detecting strange attractors in turbulence. In David Rand and Lai-Sang Young, editors, *Dynamical Systems and Turbulence, Warwick 1980*, pages 366–381, Berlin, Heidelberg, 1981. Springer Berlin Heidelberg.

[28] Jing Tang Xing. Chapter 12 - mixed finite element—smoothed particle methods for nonlinear fluid–solid interactions. In Jing Tang Xing, editor, *Fluid-Solid Interaction Dynamics*, pages 487–575. Academic Press, 2019.

[29] Ángel López-Oriona and José A. Vilar. Analyzing categorical time series with the r package ctsfeatures. *Journal of Computational Science*, 76:102233, 2024.

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every written paper or thesis authored during the course of studies. In consultation with the supervisor, one of the following three options must be selected:

○ I confirm that I authored the work in question independently and in my own words, i.e. that no one helped me to author it. Suggestions from the supervisor regarding language and content are excepted. I used no generative artificial intelligence technologies[1].

● I confirm that I authored the work in question independently and in my own words, i.e. that no one helped me to author it. Suggestions from the supervisor regarding language and content are excepted. I used and cited generative artificial intelligence technologies[2].

○ I confirm that I authored the work in question independently and in my own words, i.e. that no one helped me to author it. Suggestions from the supervisor regarding language and content are excepted. I used generative artificial intelligence technologies[3]. In consultation with the supervisor, I did not cite them.

Title of paper or thesis:

Sliding Windows and Persistence: Topological Analysis of Time Series Data

Authored by:
If the work was compiled in a group, the names of all authors are required.

Last name(s):	**First name(s):**
Meinecke	Luisa Sophie

With my signature I confirm the following:

- I have adhered to the rules set out in the Citation Guide.
- I have documented all methods, data and processes truthfully and fully.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for originality.

Place, date	**Signature(s)**
Aarau, 18.03.2025	L. Meinecke

If the work was compiled in a group, the names of all authors are required. Through their signatures they vouch jointly for the entire content of the written work.

[1] E.g. ChatGPT, DALL E 2, Google Bard
[2] E.g. ChatGPT, DALL E 2, Google Bard
[3] E.g. ChatGPT, DALL E 2, Google Bard