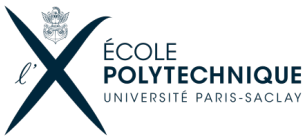


École Polytechnique
Université Paris-Saclay

Master Thesis

by

Daniel PEREZ

under the supervision of

Pierre PANSU

Title:

Persistence Theory and its Applications

2019

Contents

Introduction

Persistence theory is a generalization of persistent homology, a topological technique which arose from the needs of topological data analysis. In this thesis, we will study both of these subjects as well as give real calculation examples to see how the theoretical constructs are retrieved in practice. To do this, we first treat persistence modules abstractly and in full generality. This will allow us to give two key powerful results: on one hand Gabriel's Theorem (theorem 1.1.1), which gives some conditions under which some persistence modules are decomposable into direct sums of well-understood interval modules, and on the other hand the Isometry Theorem (theorem 1.4.1), which actually equates two notions of distance, namely the bottleneck and interleaving distances.

Having done this, we focus our attention to the particular case of persistent homology. After some definitions and recalling some facts about filtrations and homological algebra, we consider the typical scenario of topological data analysis : the calculation of the persistent homology of a point cloud in $\mathbb{R}^d$. By making some reasonable hypotheses likely to happen in real data sets on the point cloud, it is possible to infer multiple facts about the persistent homology of the latter. Namely, that if this point cloud lies Hausdorff close to a certain compact $K \subset \mathbb{R}^d$ which is "geometrically nice enough", we may read off the homology of this set K off of the persistent homology of the point cloud itself. This will be the main result of the second chapter, *i.e.* the Sweet Range Theorem (theorem 2.3.2) and its important corollary giving the explicit allure of these so-called sweet ranges as well as an explicit length of the intervals corresponding to "topological noise".

Realizing that there are many computational difficulties which are encountered in practice when computing the persistent homology of a point cloud, we give methods of approximation by providing alternative filtrations (Rips and Witness filtrations) of the point clouds. We are able to quantify the error introduced by these approximative filtrations due to the Isometry Theorem presented in the first chapter. While these methods present the inconvenience of introducing this error, they remain to date the only feasible way known to date of performing calculations once the ambient dimension of the point cloud is greater than 4.

Finally, we give concrete examples of computations which were performed using the Python topological data analysis package `gudhi`. With the help of these examples, we will see exactly how the theorems of the previous two chapters are retrieved in practice and how given a point cloud, the use of different filtration affects the performance of the computation of the persistent homology of the point cloud and how the results vary at the level of the persistence diagrams. We explore the possible effects of torsion by first giving ourselves a point cloud sparsed on the Klein bottle. More interestingly, we also give an example in which the point cloud is sparsed on a manifold which doesn't explicitly exhibit torsion itself, but whose geometrical embedding in $\mathbb{R}^3$ yields field-dependent results at the persistent homology level. These two examples reflect the importance of performing persistent homology calculations over different fields in order avoid making false topological inferences. The code used to calculate the persistent homology

of these examples is given in appendix A.

Chapter 1

Persistence Modules

The ideas behind persistence theory are for the most part rather simple and inspired by techniques of data analysis. While the latter are traditionally approached from a very pragmatic angle, we will introduce these notions from the top down, starting with very general and abstract statements, which we will apply in subsequent chapters to study persistent homology in greater detail.

In order to understand the main motivations behind the definitions and theorems to come, let us consider a toy data set sparsed on a helicoidal path on the surface of a torus in dimension 3 (*cf.* figure 1.1). By construction, the above data set presents some interesting topological features which give us precious information about its distribution in 3D space. Notice that if we look at the point cloud from afar, we see that the distribution lies on the torus. On the other hand, after a closer inspection, we actually are able to see that there is a finer structure and that the distribution is actually closer to a helicoidal path on this torus. Thus, a sense of *scale* at which we look at the data is inherently important when considering such data sets and their topological characterizations.

Choosing the scale at which we should look at a given data set is an ill-posed problem, since we have no *a priori* knowledge of the distribution of the data itself. Instead, we look at all possible scales and obtain a scale-dependent characterization of the topology of our space. This is how we arrive naturally at a construction which is essentially one in which topology is a one-parameter dependent object. One of the ways to see this concretely is to place small balls of radius ε around each point. Clearly, the topology of this collection of balls is dependent on ε, as previously discussed. In particular, one of the objects which helps us characterize the topology of the set of balls at any fixed ε is the homology of this set. The result is a family of vector spaces, which is parametrized by the radius of the balls around each point, ε. This is what motivates the definition and the study of objects defined as a "family of vector spaces" characterized by some partially ordered set, which we call *persistence modules*. In our toy model, the family of homologies parametrized by the increasing radius ε of the balls around each point in the data set is an example of such a persistence module we call the persistent homology of the point cloud.

1.1 Definitions, Properties and Representations

It turns out that it is in many ways simpler to use category theoretic language in order to translate this idea of what we mean by the "family of vector spaces" mentioned above.

Definition 1.1.1. Let a partially ordered set (or **poset**), $\mathcal{I}$, be equipped with its natural

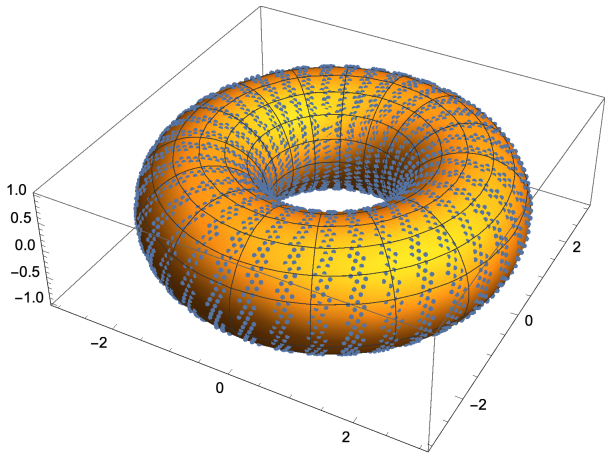


Figure 1.1: A data set sparsed on a torus. Notice that at different scales, the data seems to have different intrinsic geometry. At small scales, we notice that it is a closed curve following a helicoidal path and at a bigger scale, we are sensible to the fact it also lies on the torus.

small category structure induced by its partial order. A **persistence module** is a functor $\mathbb{V} : \mathcal{I} \to \mathbf{Mod}_A$, where $\mathbf{Mod}_A$ is the category of modules over a ring A.

Remark 1.1.1. This definition is perhaps too general, for the rest of this manuscript, we will usually consider functors $\mathbb{V} : \mathcal{I} \to \mathbf{Vect}_k$ where $\mathbf{Vect}_k$ is the category of vector spaces over a field k.

Example 1.1.1. *Let $\mathcal{I} = \mathbb{R}$ with its poset small category structure, an interval $[a, b] \subset \mathbb{R}$ and let k be a field. We may define a persistence module $k[a, b]$, prescribed by its action as a functor on $t \in \mathcal{I}$:*

$$k[a,b](t) := \begin{cases} k & \text{if } t \in [a,b] \\ 0 & \text{else} \end{cases} \tag{1.1.1}$$

Notice that we haven't yet fully defined the persistence module, since we haven't specified what the maps between these vector spaces should be. We define that for $s < t \in \mathcal{I}$, the map $\phi : V_s \to V_t$ is given by:

$$\phi := \begin{cases} \mathrm{id} & \text{if } s,t \in [a,b] \\ 0 & \text{else} \end{cases} \tag{1.1.2}$$

Remark 1.1.2. A natural extension of this kind of module is to consider the same construction but with intervals which may be clopen or open instead of just closed. In order to deal with all of these at once, it is useful to place ourselves in an enlarged version of the reals, the *hyperreal* numbers, which we will denote ${}^*\mathbb{R}$ and consider only closed intervals in this bigger set. In this way, we write: a^+ a point which is infinitesimally bigger than a *but not equal to* a, a^- a point which is infinitesimally smaller than a, but again, not equal to a. In this way, we cover all of the possibilities for clopen, open and closed intervals by writing $[a^\pm, b^\pm]$. When we wish to leave ambiguity in the clopeness of the interval, we can replace this $\pm$ with a $*$, where $*$ is a placeholder for a $^+$ or a $^-$. The fact that we can define such numbers without logical contradictions is a consequence of model theory, but this is not the topic of the present manuscript, so we will see the conventions above as simply being of notational convenience.

It turns out that the example of persistence module given above will be crucial for the rest of our exploration of the topic. In fact, it is so useful that we give it a name:

Definition 1.1.2. The persistence modules defined in example 1.1.1 are called **interval modules**.

Remark 1.1.3. Interval modules are to persistence theory what indicator functions are to the theory of integration; it is precisely this analogy which motivates the notation we will introduce for them later on as quivers. Indeed, in the same way that it is possible to decompose some kinds of functions into a sum of indicator functions, *some* persistence modules allow a decomposition into interval modules. This statement is in general not true for an arbitrary persistence module and is in particular heavily dependent on the fact that we must have a persistence module which takes values in $\mathbf{Vect}_k$ for some field k. However, much like in the theory of integration, measure theory can and will come to our rescue later on in order to allow at least partial decomposition of some more general persistence modules. In any case, from an intuitive point of view, just as indicator functions are the building blocks of the theory of integration, interval modules can be viewed as the building blocks of persistence module theory (at least for some reasonable persistence modules taking values in $\mathbf{Vect}_k$).

Definition 1.1.3. A persistence module $\mathbb{V} : \mathcal{I} \to \mathbf{Vect}_k$ is said to be **decomposable** if it can be written as a direct sum of interval modules, where an **interval** $[a, b] \subset \mathcal{I}$ is the set:

$$[a, b] := \{x \in \mathcal{I} \,|\, a \leq x \leq b\} \tag{1.1.3}$$

Using the theory of quivers it is possible to determine that the facts stated in remark 1.1.3 are true for at least some specific types of persistence modules.

Theorem 1.1.1. *(Gabriel, Auslander, Ringel-Tachikawa, Webb, Crawley-Boevey [1, §2.5]) Let $\mathbb{V} : \mathcal{I} \to \mathbf{Vect}_k$ be a persistence module over some poset $\mathcal{I} \subset \mathbb{R}$. Then, $\mathbb{V}$ can be decomposed as a direct sum of interval modules if either:*

1. *$\mathcal{I}$ is a finite set;*
2. *Each $\mathbb{V}(t)$ is a finite-dimensional vector space.*

Note that there are a number counterexamples [1, §2.5] to the decomposition of a general persistence module outside of these conditions. The fact that we can find such counterexamples is a rather obnoxious fact, which we will try to circumvent using various techniques which will allow us to get rid of the regions where the decomposition may not hold. In order to understand how to do this, we must first introduce different objects which will be fundamental to the understanding of persistence modules and that will allow us to generalize slightly the result of Gabriel, *et al.*

1.1.1 Persistence Diagrams for Decomposable Persistence Modules

The objective of this section is to introduce a way of representing persistence modules graphically. For now, we place ourselves in the situation where the persistence module is decomposable, *i.e.* we can write write $\mathbb{V}$ as

$$\mathbb{V} = \bigoplus_{a,b \in \mathcal{I}} k[a^*, b^*] \tag{1.1.4}$$

where $a^*, b^* \in {}^*\mathbb{R}$. Note that all the important information is contained in the intervals. We may represent each of these intervals $[a, b]$ as a point in $(a, b) \in \mathbb{R}^2$. Furthermore, since $a < b$ for every interval, we ensure that all this information will always be located above the diagonal

$x = y$ in $\mathbb{R}^2$. Finally, if we don't wish to lose the extra information carried by the potential (cl)openness of the intervals over $\mathbb{R}$, we may "decorate" each of the points with a tick, which will be pointing to the quadrant which has the right signs. For example: the interval $[a^+, b^-]$ will be decorated with a tick pointing towards the fourth quadrant, since in this region the x-axis is positive but the y-axis is negative. A little bit more precisely:

Definition 1.1.4. The **persistence diagram**, $\mathrm{Dgm}(\mathbb{V})$ of a persistence module $\mathbb{V} = \bigoplus_{\ell \in L} k[p_\ell^*, q_\ell^*]$ is the multiset given by:

$$\mathrm{Dgm}(\mathbb{V}) := \{(p_\ell^*, q_\ell^*) \in {}^*\mathbb{R}^2 \,|\, \ell \in L\} \tag{1.1.5}$$

Sometimes it is helpful to consider the **undecorated diagram**, $\mathrm{dgm}(\mathbb{V})$, which is the multiset of points of $\mathrm{Dgm}(\mathbb{V})$ in $\mathbb{R}^2$ without their decorations.

This notion is perhaps best illustrated by an example.

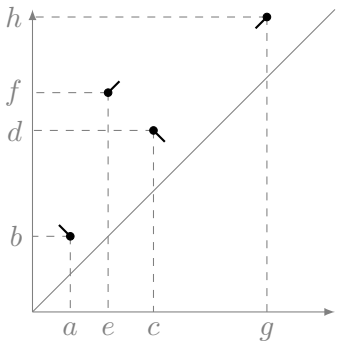
Example 1.1.2. *Consider a persistence module $\mathbb{V} : \mathbb{R} \to \boldsymbol{Vect}_k$ given by :*

$$\mathbb{V} := k[a^-, b^+] \oplus k[c^+, d^-] \oplus k[e^+, f^+] \oplus k[g^-, h^-] \tag{1.1.6}$$

Recall that in conventional clopen notation of intervals this simply means that the persistence module is:

$$\mathbb{V} = k[a, b] \oplus k\,]c, d[\oplus k\,]e, f] \oplus k[g, h[\tag{1.1.7}$$

The decorated persistence diagram of this module looks like the following:



By Gabriel's theorem (1.1.1), all the information contained in the persistence module is supported by the (decorated) persistence diagram. It is thus important to understand how to interpret this diagrams and to know how to read them.

Remark 1.1.4. Long intervals in the decomposition of $\mathbb{V}$ will lie further away from the diagonal than shorter ones. This means that if there are points close to the diagonal in the persistence diagram, these intervals are very short. We will later see that for most applications, these very short intervals represent noise and the longer intervals constitute the signal in the data.

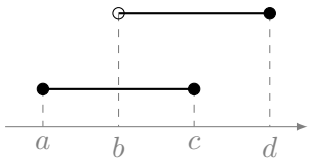
The persistence diagram is a neat graphical way of representing the features of a persistence module. In what will follow, we will aim to define what persistent diagrams are in the case where we don't have a decomposable module.

We note that we can also represent decomposable modules by so-called *barcodes*. The idea is to stack the intervals represented in the direct sum of the decomposition of a persistence module $\mathbb{V}$ along an interval. Additionally, we may decide to decorate the barcode just as we did for the persistence diagrams by including or excluding the endpoints of intervals. This concept is best illustrated with the help of an example:

Example 1.1.3. *Consider a persistence module* $\mathbb{V} : \mathbb{R} \to \boldsymbol{Vect}_k$ *given by :*

$$\mathbb{V} = k[a, c] \oplus k\,[b, d[\tag{1.1.8}$$

The (decorated) barcode for this persistence module is given by:



The filled and non-filled circles represent whether the end point of the interval is included or not. Typically, however, in all practical applications we deal with undecorated barcodes, which do not specify whether the endpoints are included or not.

Note that decorated barcodes are completely equivalent to persistence diagrams, since they contain exactly the same information. On the other hand, since we are dealing with multisets, giving both diagrams is useful in practice because from the diagram we quickly get a sense of how big the intervals are, and the barcode gives us a quick read on the multiplicity of the intervals in the decomposition of the persistence module.

1.2 Quivers and Persistence Measures

In order to define a notion of a persistence diagram for a non-decomposable persistence module, it is necessary to introduce the concept of a persistence measure, which is itself defined as being the multiplicity of a certain quiver representation within a persistence module. Because of this, it is helpful to first introduce well-suited notation for these concepts before going further.

1.2.1 Quivers and Useful Notation

We start by placing ourselves in the situation where $\mathbb{V} : \mathcal{T} := \{a_1, \cdots, a_n\} \to \mathbf{Vect}_k$. The finiteness of $\mathcal{T}$ implies that this persistence module is nothing other than a representation of an A_n-type quiver of the form:

$$\bullet \text{ — } \bullet \cdots \bullet \text{ — } \bullet \tag{1.2.9}$$

Concretely, this representation is simply:

$$V_{a_1} \to V_{a_2} \to \cdots \to V_{a_n} \tag{1.2.10}$$

where the V_{a_i}'s are k-vector spaces. Since $\mathcal{T}$ is finite, such a persistence module decomposes into a finite sum of interval modules as a consequence of Gabriel's theorem (1.1.1). As previously remarked, having a clear understanding of interval modules is of key importance and in particular, finding good and well-suited notation for these modules will be of great help for what is to come.

Notation 1.2.1. Let $\bullet$ denote the vector space k and let $\circ$ denote the zero vector space. We denote the interval module $k[a_i, a_j]$ by:

$$\circ_{a_1} \text{ — } \circ_{a_2} \cdots \circ_{a_{i-1}} \text{ — } \bullet_{a_i} \cdots \bullet_{a_j} \text{ — } \circ_{a_{j+1}} \cdots \circ_{a_{n-1}} \text{ — } \circ_{a_n} \tag{1.2.11}$$

As before, we choose the isomorphism between the $\bullet$'s to be the identity. Note also that we have no choice for the arrows stemming or ending in a $\circ$ so we need not be specific about which map is used here, since it must necessarily be the zero map.

More often than not, we will have persistence modules which are indexed over a poset which is not finite (typically $\mathbb{R}$), using the notation we just introduced, it is also practical to consider the following:

Notation 1.2.2. Let $\mathbb{V}$ be a persistence module over a poset $\mathcal{I}$ (which is not necessarily finite) and let $\mathcal{T} \subset \mathcal{I}$ be a finite subset $\mathcal{T} := \{a_1 < a_2 < \cdots < a_n \,|\, a_i \in \mathcal{I}\}$. Define the persistence module $\mathbb{V}_\mathcal{T}$ as being the restriction as a functor of the persistence module $\mathbb{V}$ to the poset $\mathcal{T}$. Given $k[a_i, a_j]$ an interval module, let $\langle [a_i, a_j] \,|\, \mathbb{V}_\mathcal{T} \rangle$ denote the multiplicity of the interval module $k[a_i, a_j]$ in the persistence module $\mathbb{V}_\mathcal{T}$. Sometimes, it is practical to tweak this notation by combining it with the notation we introduced for interval modules in 1.2.1 in the following way:

$$\langle [a_i, a_j] \,|\, \mathbb{V}_\mathcal{T} \rangle =: \langle \circ_{a_1} \cdots \circ_{a_{i-1}} \text{—} \bullet_{a_i} \cdots \bullet_{a_j} \text{—} \circ_{a_{j+1}} \cdots \circ_{a_n} \,|\, \mathbb{V} \rangle \tag{1.2.12}$$

Remark 1.2.1. The poset $\mathcal{T}$ is given explicitly in the quiver notation for interval modules, so both notations really contain the same information.

We can immediately extend this notation to include direct sums of interval modules in the obvious way: in fact, the second notation is particularly well-suited for this, as it is sufficient to replace $\circ$'s with $\bullet$'s where appropriate.

Remark 1.2.2. Note that the above notation is well-defined, because over any such finite poset $\mathcal{T}$, Gabriel's theorem (1.1.1) guarantees the existence of a decomposition and therefore the ability to count multiplicities.

Example 1.2.1. *The invariants of a linear map $\nu : \mathbb{V}(a) \to \mathbb{V}(b)$ stemming from $\mathbb{V}$ can be intuitively pictured with the notation above, indeed:*

$$rank(\nu) = \langle \bullet_a \text{—} \bullet_b \,|\, \mathbb{V} \rangle \tag{1.2.13}$$

$$\dim ker(\nu) = \langle \bullet_a \text{—} \circ_b \,|\, \mathbb{V} \rangle \tag{1.2.14}$$

$$\dim coker(\nu) = \langle \circ_a \text{—} \bullet_b \,|\, \mathbb{V} \rangle \tag{1.2.15}$$

Example 1.2.2. *Perhaps a bit more generally, let $\mathbb{V}$ be a persistence module over $\mathbb{R}$, and take $a < b < c$ three real numbers and $\mathbb{V}_{a,b,c}$ its restriction to the poset $\{a, b, c\} \subset \mathbb{R}$, i.e. :*

$$\mathbb{V}_{a,b,c} : V_a \to V_b \to V_c \tag{1.2.16}$$

We could then consider, for instance:

$$\langle [b, c] \,|\, \mathbb{V}_{a,b,c} \rangle \text{ or } \langle \circ_a \text{—} \bullet_b \text{—} \bullet_c \,|\, \mathbb{V} \rangle \tag{1.2.17}$$

the multiplicity of $\circ_a$ — $\bullet_b$ — $\bullet_c$ in the persistence submodule $\mathbb{V}_{a,b,c}$ described above. Note that the persistence module in which we take the multiplicity is important, since changing the persistence module can obviously change the multiplicity we count, this is why noting $\langle [b, c] \rangle$ is ambiguous, because $\langle [b, c] \,|\, \mathbb{V}_{b,c} \rangle$ and $\langle [b, c] \,|\, \mathbb{V}_{a,b,c} \rangle$ might in general not be the same. When the underlying persistence module $\mathbb{V}$ is clear from context, we can omit it in the notation and instead just note:

$$\langle \circ_a \text{—} \bullet_b \text{—} \bullet_c \rangle \tag{1.2.18}$$

It is possible to demonstrate some neat results, which in some sense justify why we introduced the notation above, as the results are easily interpreted graphically. The proofs will be omitted, but they clearly follow from considering interval module decompositions of the finitely indexed persistence modules and restrictions of intervals. For full details, the reader is encouraged to consult [1].

Proposition 1.2.1 (Direct sums)**.** Suppose that a persistence module $\mathbb{V}$ can be written as:

$$\mathbb{V} = \bigoplus_{\ell \in L} \mathbb{V}^{\ell} \tag{1.2.19}$$

for some index set L. Then, for any finite index set $\mathcal{T} := \{a_1, \cdots, a_n\}$ and interval $[a_i, a_j] \subset \mathcal{T}$, we have that:

$$\langle [a_i, a_j] \,|\, \mathbb{V}_{\mathcal{T}} \rangle = \sum_{\ell \in L} \langle [a_i, a_j] \,|\, \mathbb{V}^{\ell}_{\mathcal{T}} \rangle \tag{1.2.20}$$

Proposition 1.2.2 (Restriction principle)**.** Let $\mathcal{S}$ and $\mathcal{T}$ be finite index sets with $\mathcal{S} \subset \mathcal{T}$, then for any interval $\mathbb{I}$ of $\mathcal{S}$:

$$\langle \mathbb{I} \,|\, \mathbb{V}_{\mathcal{S}} \rangle = \sum_{\mathbb{J}} \langle \mathbb{J} \,|\, \mathbb{V}_{\mathcal{T}} \rangle \tag{1.2.21}$$

where the sum is carried out precisely over all intervals $\mathbb{J} \subset \mathcal{T}$ which restrict to $\mathbb{I}$ over $\mathcal{S}$.

With the help of examples, we illustrate why our notation is convenient in practical cases:

Example 1.2.3. *Suppose we have a persistence module $\mathbb{V}$ over a poset $\mathcal{T} = \{a < p < b < q < c\}$ and let $\mathcal{S} := \{a, b, c\}$. Suppose we wish to compute $\langle [b, c] \,|\, \mathbb{V}_{\mathcal{S}} \rangle$, by the restriction principle we have:*

$$\langle \circ_a \text{——} \bullet_b \text{——} \bullet_c \rangle = \langle \circ_a \text{—} \circ_p \text{—} \bullet_b \text{—} \bullet_q \text{—} \bullet_c \rangle + \langle \circ_a \text{—} \bullet_p \text{—} \bullet_b \text{—} \bullet_q \text{—} \bullet_c \rangle \tag{1.2.22}$$

since the intervals $[b, c]$ and $[p, c]$ are the only intervals of $\mathcal{T}$ who restrict to $[b, c]$ as an interval of $\mathcal{S}$. Note that an extra term occurs when a new index appears between a $\circ$ and a $\bullet$, because then there are two possible intervals which restrict to the original interval.

With the machinery and notation we have introduced, we are now ready to introduce a measure on the persistence diagram of a persistence module which will allow us to somewhat generalize the decomposition of persistence modules.

1.2.2 Persistence Measures

In what will follow, we will introduce the notion of a *persistence measure* to a degree of generality which is appropriate for our use of this concept for the rest of this manuscript. More general notions exist, such as *rectangle measures* which are treated in [1, §3.3] and the reader is encouraged to consult this more complete treatment of the topic. However, for the sake of brevity, we focus mainly on the aspects of persistence measures which we will use in following chapters.

Definition 1.2.1. Let $\mathbb{V}$ be a persistence module indexed over $\mathcal{I}$. The **persistence measure** of $\mathbb{V}$ is the function defined on rectangles $R = [a, b] \times [c, d] \subset \mathcal{I}^2$ with $a < b \leq c < d$:

$$\mu_{\mathbb{V}}(R) := \langle \circ_a \text{—} \bullet_b \text{—} \bullet_c \text{—} \circ_d \,|\, \mathbb{V} \rangle \tag{1.2.23}$$

Remark 1.2.3. Usually, we deal with posets which are subsets of $\mathbb{R}$ so that we may consider the measures above are actually defined on rectangles above the diagonal in $\mathbb{R}^2$. Also note that if $\mathbb{V}$ is decomposable, the persistence measure counts the number of points (with multiplicity) of the decorated persistence diagram of $\mathbb{V}$ inside the rectangle R. In a way, a persistence measure is akin to a Dirac measure over rectangles of the weighted points of the decorated persistence diagram. This comparison is important for the sake of intuition, as all the statements that will

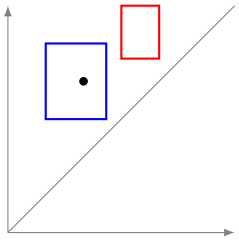
come become instantly more transparent. In fact, getting a bit ahead of ourselves, it is crucial to make this parallel, because we will later extend the notion of a (decorated) persistence diagram using persistence measures in order to extract decomposable parts out of more general persistence modules.

To justify the above remark, we have the following proposition:

Proposition 1.2.3. Let $\mathbb{V} = k\,[p^*, q^*]$ be an interval module and let $R = [a, b] \times [c, d]$ where $a < b \leq c < d$, then

$$\mu_{\mathbb{V}}(R) = \begin{cases} 1 & \text{if } [b, c] \subset [p^*, q^*] \subset]a, d[\\ 0 & \text{else} \end{cases} \tag{1.2.24}$$

This represents exactly what we stated above in the remark, *i.e.* graphically we are in the following situation:



Here, the blue rectangle has a measure of 1, since it contains the point (p^*, q^*) whereas the red one has a measure of 0, since it doesn't contain the point. Note that we could be in an ambiguous situation where the point (p, q) lies exactly on the edge of a rectangle. In this case, if the decoration of the point (p^*, q^*) lies inside the rectangle, it is contained inside it, otherwise, it isn't. Graphically, we have figure 1.2

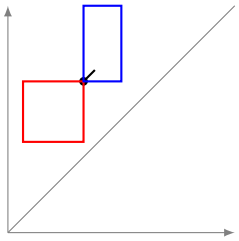


Figure 1.2: The red rectangle does *not* contain the point (p^*, q^*), but the blue one does.

Proof. By the definition we have of the persistence measure:

$$\mu_{\mathbb{V}}(R) = \langle \circ_a \text{ — } \bullet_b \text{ — } \bullet_c \text{ — } \circ_d \,|\, \mathbb{V} \rangle \tag{1.2.25}$$

It is clear that $k[p^*, q^*]$ restricted to the finite poset $\{a, b, c, d\}$ is either 0 or it is an interval module, since $\mathbb{V}$ is itself an interval module. Thus, $\mu_{\mathbb{V}}(R) \leq 1$. Furthermore, if the persitence measure is non-trivial, we must be in the following situation:

$$\circ_a \text{ — } \bullet_{p^*} \text{ — } \bullet_b \text{ — } \bullet_c \text{ — } \bullet_{q^*} \text{ — } \circ_d \tag{1.2.26}$$

Since this is the only diagram in $\mathbb{V}$ whose restriction to $\mathbb{V}_{a,b,c,d}$ will yield something non-trivial for $\circ_a$ — $\bullet_b$ — $\bullet_c$ — $\circ_d$. Thus, we must necessarily have that $[b, c] \subset [p^*, q^*] \subset]a, d[$. ∎

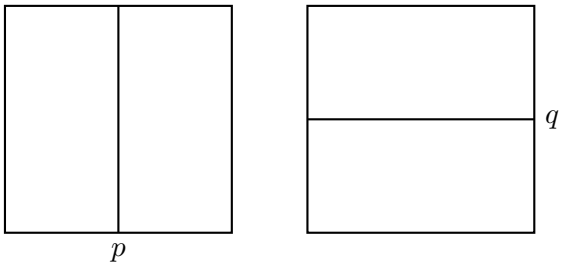


Figure 1.3: To the left, we have the vertical splitting of a rectangle at q. To the right, we have the horizontal splitting of a rectangle at p.

At this point, it is helpful to clarify a couple of points. We have yet to show that the persistence measure is indeed a measure over rectangles in the upper-half-plane defined by the diagonal in $\mathcal{I}^2$ (which we will henceforth take to be $\mathbb{R}^2$). Already in the case of decomposable persistence modules, this is actually a measure on the *decorated* persistence diagram. This is simply because we don't want double-counting of points along edges of rectangles. It must thus be clear from the measure's perspective whether a point is inside a rectangle R or not. The decorations in the diagram allow us to make this distinction.

We can prove some propositions proving subadditivity of the persistence measure, which is the only condition we have left to prove that the persistence measure is indeed a measure. This is once again an instance where the intuitive parallel between the persistence measures and Dirac measures of the decorated diagram's points comes in handy, as it gives us a feeling that subadditivity should indeed hold. Note that it is enough to show that we have subadditivity for horizontal and vertical splittings of rectangles (*cf.* figure 1.3), since all other partitions in rectangles of a rectangle can be obtained in this way.

Proposition 1.2.4. Let $\mathbb{V}$ be a persistence module. The persistence measure $\mu_{\mathbb{V}}$ is subadditive under vertical and horizontal splittings, *i.e.* for $a < p < b \leq c < q < d$:

$$\mu_{\mathbb{V}}([a,b] \times [c,d]) = \mu_{\mathbb{V}}([a,p] \times [c,d]) + \mu_{\mathbb{V}}([p,b] \times [c,d]) \tag{1.2.27}$$

$$\mu_{\mathbb{V}}([a,b] \times [c,d]) = \mu_{\mathbb{V}}([a,b] \times [c,q]) + \mu_{\mathbb{V}}([a,b] \times [q,d]) \tag{1.2.28}$$

Proof. The proof is a direct and straightforward consequence of the restriction principle. We only prove vertical subadditivity as the horizontal case can be done analogously [1, §3.2].

$$\begin{aligned}
\mu_{\mathbb{V}}([a,b] \times [c,d]) \quad &= \langle \circ_a \text{ — } \bullet_b \text{ — } \bullet_c \text{ — } \circ_d \,|\, \mathbb{V} \rangle \\
&= \langle \circ_a \text{ — } \bullet_p \text{ — } \bullet_b \text{ — } \bullet_c \text{ — } \circ_d \rangle + \langle \circ_a \text{ — } \circ_p \text{ — } \bullet_b \text{ — } \bullet_c \text{ — } \circ_d \rangle \\
&= \mu_{\mathbb{V}}([a,p] \times [c,d]) + \mu_{\mathbb{V}}([p,b] \times [c,d])
\end{aligned}$$

■

The next fact that we need in order to truly establish everything stated in remark 1.2.3 is the following theorem, which states the existence of a correspondence between finite persistence measures and multisets of points A. Again, these results are established in a somewhat higher degree of generality in [1], but we won't need the more general results in this manuscript.

Theorem 1.2.1 (The equivalence theorem)**.** *Consider a region $\mathcal{D} \subset \mathbb{R}^2$. Then, there is a bijective correspondence between:*

- *Finite persistence measures μ on the rectangles R of $\mathcal{D}$, where finite means $\mu(R) < \infty$ for all rectangles R inside the region $\mathcal{D}$;*
- *Locally finite multisets A contained in $\mathcal{D}$, where locally finite means that for every rectangle R contained in $\mathcal{D}$, we have $\#A|_R < \infty$.*

The idea is that from the persistence measure, we can extract information in regions $\mathcal{D}$ where the measure is locally finite. By the equivalence theorem, we may assimilate the measure to its corresponding multiset, which means that a notion of a persistence diagram defined precisely by the multiset of points extracted from the measure exists and makes sense. By analogy, we can say that we have in some sense extracted a decomposable part of the module over the regions where the measure is locally finite, since every point in the persistence diagram actually corresponds to an interval module.

Proof. We will only give a sketch of proof here. For the detailed version of the proof, the reader is encouraged to consult [1, §3.4]. One direction of the equivalence is easy. We can attribute to a given multiset the Dirac weighted measure over its points, which will yield the persistence measure we seek, explicitly:

$$\mu(R) = \#(A|_R). \tag{1.2.29}$$

Establishing the converse is a little more intricate, but the idea is that, given a persistence measure μ over which is locally finite over the region $\mathcal{D}$, we define a multiplicity function for each decorated point (p^*, q^*) defined by:

$$m(p^*, q^*) := \min\{\mu(R) \,|\, R \text{ is a rectangle, } R \subset \mathcal{D}, (p^*, q^*) \in R\} \tag{1.2.30}$$

This minimum is necessarily attained, since $\mu(R)$ takes values over the natural numbers. The idea is to establish the multiset we are looking for by taking a family of smaller and smaller rectangles which eventually converge to a single point, whose multiplicity will be given by the multiplicity function. This will necessarily work because the measure is locally finite, takes values over the integers and splits vertically and horizontally. ■

1.2.3 Tameness and the Redefinition of Persistence Diagrams

Given the equivalence theorem 1.2.1, it is possible to redefine what we mean by persistence diagrams to encompass a larger class of persistence modules other than decomposable ones.

Definition 1.2.2. Let μ be a finite persistence measure over a region $\mathcal{D} \subset \mathbb{R}^2$.

- The **decorated persistence diagram** of μ is the unique locally finite multiset $\mathrm{Dgm}(\mu)$ in $\mathcal{D}$ such that:
$$\mu(R) = \#(\mathrm{Dgm}(\mu)|_R) \tag{1.2.31}$$
- The **undecorated persistence diagram** of μ is the locally finite multiset $\mathrm{dgm}(\mu)$ in the interior of $\mathcal{D}$, such that:
$$\mathrm{dgm}(\mu) = \{(p, q) \in \mathring{\mathcal{D}} \,|\, (p^*, q^*) \in \mathrm{Dgm}(\mu)\} \tag{1.2.32}$$

Remark 1.2.4. It is possible to extend this notion to measures which may not be finite. To do this, we look at the regions in which the measure is finite and apply the previous definitions to this region, thereby excluding the problematic regions which have infinite measure. Note that

the persistence modules which exhibit persistence measures with such characteristics often correspond to non-decomposable persistence modules. We can thus use this in order to construct examples of modules which don't obey the hypotheses of Gabriel's theorem (1.1.1) which are non-decomposable, by, for example, giving accumulation points in the persistence diagram of a persistence module or defining persistence modules with regions of infinite measure.

In light of the previous remark, it is possible to define classes of persistence modules which are more or less decomposable in the sense that the regions which present infinite measure are bounded to certain parts of $\mathbb{R}^2$. Intuitively, these correspond to modules which are "more or less decomposable" in the sense explained in remark 1.2.4. In our case, we will only define one such notion, but others can be found in [1, §3.9].

Definition 1.2.3. Let $\mathbb{V}$ be a persistence module whose persistence measure is $\mu_{\mathbb{V}}$. We say that $\mathbb{V}$ is **q-tame** if $\mu_{\mathbb{V}}(Q) < \infty$ for all rectangles Q not touching the diagonal, *i.e.* :

$$\langle \bullet_b \text{ — } \bullet_c \,|\, \mathbb{V} \rangle < \infty \tag{1.2.33}$$

In other words, the rank of every map in the persistence module is finite.

This is actually quite useful in practice, as this means that we need only worry about the rank of each map being finite, which relaxes the conditions in which we are allowed to apply all the machinery of persistence diagrams in order to be able to say something about the persistence module. It turns out that this relaxation is just good enough to be used in some cases which arise in practice but which don't lie in the scope of Gabriel's theorem (1.1.1).

1.3 Interleavings Between Persistence Modules

In this section, we introduce *interleavings* in an effort to give a looser concept of an isomorphism between persistence modules. This is done because we want to establish a link between two persistence modules which are the same "up to some noise". Indeed, recall that an isomorphism between persistence modules is an isomorphism of functors, *i.e.* $\mathbb{U}$ and $\mathbb{V}$ are isomorphic if there are natural transformations $\Phi \in \mathrm{Hom}(\mathbb{U}, \mathbb{V})$ and $\Psi \in \mathrm{Hom}(\mathbb{V}, \mathbb{U})$ such that:

$$\Psi\Phi = 1_{\mathbb{U}} \text{ and } \Phi\Psi = 1_{\mathbb{V}} \tag{1.3.34}$$

Asking for such an isomorphism is, for most purposes, far too much to ask. To see why, it is perhaps useful to understand where the idea of interleavings originally germinated: topological data analysis. Going back to our toy model of of data sparsed across a torus (*cf.* figure 1.1), we can see that computing the persistent homology of the full point cloud or of a well-picked subset of this data should roughly give us the same persistence module since in the end, they both have the same topological structure. This makes us want to say that for all practical purposes, the two modules are "the same" or that at least we have "no real loss of information". On the other hand, we could never hope to obtain an isomorphism in the proper sense of the word between these two modules. Instead, we say that these two modules are *interleaved*. In what will follow we will make this concept precise, and state and prove some facts about interleaved modules.

1.3.1 Shifted Morphisms and Interleavings of Modules

Definition 1.3.1. Let $\mathbb{V}$ be a persistence module indexed over $\mathbb{R}$ and $\delta \in \mathbb{R}$. The **shifted module** $\mathbb{V}[\delta]$ is defined by:

$$\mathbb{V}[\delta](t) := V_{t+\delta} \text{ and } (\nu[\delta])_t^s := \nu_{t+\delta}^{s+\delta} \tag{1.3.35}$$

where $\nu_t^s : V_s \to V_t$ is the map induced by functoriality of $\mathbb{V}$ from the map in the small category structure induced by $s < t$. In other words, we shift all the information of $\mathbb{V}$ downwards by δ.

Definition 1.3.2. Let $\mathbb{U}$ and $\mathbb{V}$ be two persistence modules indexed over $\mathbb{R}$ and let $\delta \in \mathbb{R}$. A **homomorphism of degree** δ is a natural transformation:

$$\Phi : \mathbb{U} \to \mathbb{V}[\delta] \tag{1.3.36}$$

Definition 1.3.3. Amongst the homomorphisms of degree δ, one is particularly important namely the shifting map:

$$1_{\mathbb{V}}^{\delta} : \mathbb{V} \to \mathbb{V}[\delta] \tag{1.3.37}$$

which is the natural transformation which shifts the information of $\mathbb{V}$ downwards by δ.

With these definitions, we are ready to define what an interleaving is:

Definition 1.3.4. Let $\delta \geq 0$ and let $\mathbb{U}$ and $\mathbb{V}$ be two persistence modules indexed over $\mathbb{R}$. $\mathbb{U}$ and $\mathbb{V}$ are **interleaved** if there are maps:

$$\Phi \in \mathrm{Hom}(\mathbb{U}, \mathbb{V}[\delta]) \text{ and } \Psi \in \mathrm{Hom}(\mathbb{V}, \mathbb{U}[\delta]) \tag{1.3.38}$$

such that:

$$\Psi\Phi = 1_{\mathbb{U}}^{2\delta} \text{ and } \Phi\Psi = 1_{\mathbb{V}}^{2\delta} \tag{1.3.39}$$

At this point, it is helpful to give a nice graphical way of thinking about interleavings, which will actually also help us define interpolations between persistence modules later on. First we need a setting for considering both persistence modules at once under the same poset. To do this, we look at $\mathbb{R}^2$ with its standard order, *i.e.* :

$$(p_1, q_1) \leq (p_2, q_2) \iff p_1 \leq p_2 \text{ and } q_1 \leq q_2 \tag{1.3.40}$$

Furthermore, for any $x \in \mathbb{R}$ we denote the shifted diagonal by x:

$$\Delta_x := \{t \in \mathbb{R} \mid (t - x\,, t + x) \in \mathbb{R}^2 \text{ for } t \in \mathbb{R}\} \tag{1.3.41}$$

This shifted diagonal is naturally isomorphic to $\mathbb{R}$ equipped with its natural partial order. In particular, this means that we can regard any persistence module over $\mathbb{R}$ as a persistence module over Δ_x for some $x \in \mathbb{R}$ by using this identification. In this way, given any two persistence modules $\mathbb{U}$ and $\mathbb{V}$ over $\mathbb{R}$ we can consider for instance their homologues indexed over Δ_x and Δ_y for $x, y \in \mathbb{R}$. Note that this defines a new persistence module $\mathbb{W}$ indexed over $\Delta_x \cup \Delta_y$ (as a subposet of $\mathbb{R}^2$). For the sake of symmetry, we generally take $x = -1$ and $y = 1$.

Proposition 1.3.1. Let $\mathbb{U}$ and $\mathbb{V}$ be persistence modules. $\mathbb{U}$ and $\mathbb{V}$ are $|b-a|$-interleaved if and only if there exists a persistence module $\mathbb{W}$ defined over $\Delta_a \cup \Delta_b$ such that $\mathbb{W}|_{\Delta_a} = \mathbb{U}$ and $\mathbb{W}|_{\Delta_b} = \mathbb{V}$.

Proof. We give a sketch of proof which can be easily understood graphically. For the proof in full detail, we refer the reader to [1, §4.3]. Let $\delta = |b - a|$ and let x' and y' define translated axes such that

$$\begin{cases} x' = x - \frac{a+b}{2} \\ y' = y + \frac{a+b}{2} \end{cases} \tag{1.3.42}$$

To prove that there is such an interleaving given the persistence module $\mathbb{W}$, we need only show that the natural transformations Ψ and Φ exist along with their commutative diagrams (as

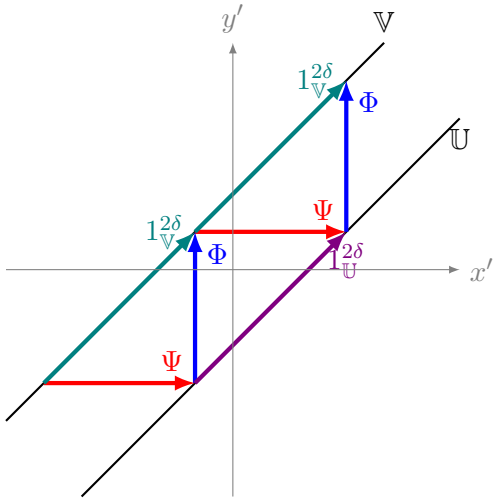


Figure 1.4: The commutation relations of naturality and of the interleaving can be explicitly represented graphically on the persistence module defined on $\Delta_a \cup \Delta_b$

well as showing the conditions laid out in equation 1.3.39). Graphically, we can see that the partial order we defined on $\mathbb{R}^2$ gives rise to the following maps in the persistence module $\mathbb{W}$.

In particular, all these maps must necessarily commute, since $\mathbb{W}$ is a persistence module. It follows that all the commutation relations of naturality (which correspond to parallelograms) and the ones in the definition of an interleaving (the diagrams determined by the triangles formed by the arrows in figure 1.4) are satisfied by considering the maps Φ and Ψ above. Conversely, it is easy to see that from the commutation relations of the interleaving imply that such a persistence module $\mathbb{W}$ exists, simply by taking composition of the morphisms Ψ and Φ with the translation maps.

■

1.4 Notions of Distance in Persistence Modules

The ideas explored in this section will be about introducing different notions of distances between persistence modules. Intuitively, it is easy to see how we could come up with different notions of what it means for two persistence modules to be close to each other. On one hand, we would like to say that two modules are close to each other if they are "close" to being isomorphic. We can give a more precise sense to this now that we have introduced the concept of interleavings. Concretely, this is translates the idea that two modules are close if they are the same up to little or no noise between them. On the other hand, another notion of distance between modules could be imagined by looking at the persistence diagrams of q-tame persistence modules. In that case, we would say that if the two diagrams are "similar", then the two persistence modules should be close. These ideas are exactly the notions we will detail in this section : in fact, we will also establish that looking at these notions of distance is actually exactly equivalent. This has different repercussions and interpretations that we will be able to appreciate once we have fully stated what exactly we mean by the above.

1.4.1 Interleaving Distance

We start by considering the so-called interleaving distance. Here, we will use the existence of interleavings between two persistence modules to give sense to the notion of two modules being close to each other. It is first helpful to make the following remark:

Remark 1.4.1. Two persistence modules $\mathbb{U}$ and $\mathbb{V}$ that are δ-interleaved are also $(\delta + \varepsilon)$-interleaved for every $\varepsilon > 0$. This is simply because we may take:

$$\Phi' = \Phi 1^{\varepsilon}_{\mathbb{U}} = 1^{\varepsilon}_{\mathbb{U}}\Phi \quad \text{and} \quad \Psi' = \Psi 1^{\varepsilon}_{\mathbb{V}} \tag{1.4.43}$$

In order to be able to define the notion of distance, we would like thus to take the minimum δ such that the two modules are interleaved. However, there is some difficulty in making this precise, because this minimum is not necessarily attained. Indeed, it is not because two interleaving modules are $(\delta + \varepsilon)$-interleaved that they are necessarily interleaved: we can see this easily by simply taking a decomposable module and its direct sum with an interval module $k[a, a]$. Clearly, these are not isomorphic, so they are not 0-interleaved, by they are ε-interleaved for every $\varepsilon > 0$. If we are in such a situation, *i.e.* two modules which are $(\delta + \varepsilon)$-interleaved, we say that they are δ^+-interleaved. With this clarification in mind we are ready to define the interleaving distance.

Definition 1.4.1. The **interleaving distance** between two persistence modules $\mathbb{U}$ and $\mathbb{V}$ is defined as:

$$d_i(\mathbb{U}, \mathbb{V}) := \inf\{\delta \,|\, \mathbb{U} \text{ and } \mathbb{V} \text{ are } \delta\text{-interleaved}\} \tag{1.4.44}$$

Remark 1.4.2. Notice that this distance may very well be infinite if there is no δ-interleaving between the two modules.

It turns out that this "distance" is actually only a pseudo-distance, as is is clearly not true that if $d_i(\mathbb{U}, \mathbb{V}) = 0$ then $\mathbb{U} = \mathbb{V}$ because of the caveat we explained above. We stress this fact in the following example:

Example 1.4.1. *The modules $k[a, b]$, $k[a, b[$, $k\,]a, b]$ and $k\,]a, b[$ are all 0^+-interleaved, but are not isomorphic to each other*

However, the interleaving distance still obeys the triangle inequality:

Proposition 1.4.1. The interleaving distance satisfies the triangle inequality, *i.e.* for any three persistence modules $\mathbb{U}, \mathbb{V}, \mathbb{W}$:

$$d_i(\mathbb{U}, \mathbb{W}) \leq d_i(\mathbb{U}, \mathbb{V}) + d_i(\mathbb{V}, \mathbb{W}) \tag{1.4.45}$$

1.4.2 Bottleneck Distance

Next is the notion of the bottleneck distance, which corresponds to the intuitive fact exposed earlier which consists in considering that two persistence modules are close if their persistence diagrams are closed. Much of what we will introduce is actually done so that things are consistent with the previous notion of distance we introduced in terms of interleavings. In the end, we hope to recover some kind of link between the two notions of distance if we do things in a consistence manner since we hope that if two modules are almost isomorphic to each other (in the sense of interleavings), their persistence diagrams should be similar. In order to make this precise, we actually need to make some things explicit. First, we shall consider the undecorated diagrams when measuring the bottleneck distance : this is in order to avoid

complications, but also corresponds to the fact that the bottleneck distance is actually not a distance, but a pseudo-distance, for the same reasons that the interleaving distance is not a distance itself. This is reflected here by ignoring what happens close to the diagonal of the persistence diagram, but also completely disregarding the decorations of the points so that the modules considered in example 1.4.1 are also at bottleneck distance zero from one another, yet are not isomorphic. The idea behind the bottleneck distance is to qualify two persistence diagrams as close if there is a bijection between them that doesn't move any point "too far". This "too far" will be quantified using the ℓ^∞-metric. For a pair of points (p,q), (r,s) we have:

$$d^\infty((p,q),(r,s)) = \max\{|p-r|, |q-s|\} \tag{1.4.46}$$

The reason for this definition is to have consistency with the interleaving distance and its behaviour with respect to interval modules. Indeed, we have the following proposition:

Proposition 1.4.2. Let $[p^*, q^*]$ and $[r^*, s^*]$ be two intervals (possibly infinite) and let:

$$\mathbb{U} = k[p^*, q^*] \text{ and } \mathbb{V} = k[r^*, s^*] \tag{1.4.47}$$

be the corresponding interval modules. Then:

$$d_i(\mathbb{U}, \mathbb{V}) \leq d^\infty((p,q),(r,s)) \tag{1.4.48}$$

Proof. Let us consider the case where $p, q, r, s \leq \infty$. We must show that if

$$\delta > \max\{|p-r|, |q-s|\}, \tag{1.4.49}$$

then $\mathbb{U}$ and $\mathbb{V}$ are δ-interleaved. We must thus simply show the existence of the maps $\Phi : \mathbb{U} \to \mathbb{V}[\delta]$ and $\Psi : \mathbb{V} \to \mathbb{U}[\delta]$ and ensure the commutation relations imposed by naturality and their definition of being an interleaving. We we define Φ and Ψ to be given by:

$$\Phi(t) := \begin{cases} id & \text{if } \mathbb{U}(t) = k \text{ and } \mathbb{V}(t+\delta) = k \\ 0 & \text{else} \end{cases} \quad \text{and} \quad \Psi(t) := \begin{cases} id & \text{if } \mathbb{V}(t) = k \text{ and } \mathbb{U}(t+\delta) = k \\ 0 & \text{else} \end{cases} \tag{1.4.50}$$

For Φ, the condition of naturality imposes that for every $\eta > 0$ and for all t the following diagram commutes:

$$\begin{array}{ccc} \mathbb{U}(t) & \longrightarrow & \mathbb{U}(t+\eta) \\ \downarrow{\scriptstyle \Phi(t)} & & \downarrow{\scriptstyle \Phi(t+\eta)} \\ \mathbb{V}(t+\delta) & \longrightarrow & \mathbb{V}(t+\delta+\eta) \end{array} \tag{1.4.51}$$

Because of the modules that we are considering and still adopting our quiver notation mentioned in a previous section, it is enough to check that the following two situations don't occur:

$$\begin{array}{ccc} \bullet & \longrightarrow & \circ \\ \downarrow & & \downarrow \\ \bullet & \longrightarrow & \bullet \end{array} \quad \text{or} \quad \begin{array}{ccc} \bullet & \longrightarrow & \bullet \\ \downarrow & & \downarrow \\ \circ & \longrightarrow & \bullet \end{array} \tag{1.4.52}$$

for these two cases to occur, the following inequalities should hold, respectively:

$$\begin{cases} q \leq t+\eta \\ t+\delta+\eta \leq s \end{cases} \quad \text{or} \quad \begin{cases} p \leq t \\ t+\delta \leq r \end{cases} \tag{1.4.53}$$

This in turn would entail that:

$$\delta \leq s-t-\eta \leq s-q \text{ ※} \quad \text{or} \quad \delta \leq r-t \leq r-p \text{ ※} \tag{1.4.54}$$

both lead to contradictions, since $\delta > \max\{|r-p|, |s-q|\}$ by assumption. ■

With this result proved, we see the motivation for choosing the ℓ^∞-norm as our norm for the distance between points in the persistence diagram. On the other hand, asking for a bijection between all points in the persistence diagrams of two modules might be too much to ask, indeed, as we understood in a previous section, what happens near the diagonal can most of the time be understood as being noise. We thus also want to say that even if we cannot pair all points between the diagrams of two persistence modules, as long as the points that remain unmatched are close enough to the diagonal, we consider that these persistence modules are still close to one another. Once again, we use a slightly tweaked version of the ℓ^∞-metric. For an unpaired point we have:

$$d^\infty((p,q),\Delta) = \frac{1}{2}|p-q| \tag{1.4.55}$$

This is motivated by the following proposition, which characterizes the behaviour of interval modules with respect to interleavings.

Proposition 1.4.3. Let $k[p^*,q^*]$ be an interval module and let 0 denote the zero persistence module. Then:

$$d_i(k[p^*,q^*],0) = \frac{1}{2}|p-q| \tag{1.4.56}$$

Proof. We can see this quite simply. Indeed, if for $\delta \geq 0$ we have a δ-interleaving between the two modules if all the interleaving maps are zero. The only condition to check is thus that:

$$\Phi\Psi = 1^{2\delta}_{k[p^*,q^*]} \tag{1.4.57}$$

which implies that $1^{2\delta}_{k[p^*,q^*]} = 0$ but this is only true if $\delta > \frac{1}{2}|p-q|$, where the inequality is sharp. ■

Next, we need a formal way of defining what we mean by a matching between two multisets of points.

Definition 1.4.2. A δ-matching between the multisets A and B is a collection of pairs $M \subset A \times B$ such that:

- For every $\alpha \in A$ (resp. $\beta \in B$) there is at most one $\beta \in B$ (resp. $\alpha \in A$) such that $(\alpha,\beta) \in M$;
- If $(\alpha,\beta) \in M$ then $d^\infty(\alpha,\beta) \leq \delta$;
- If $\alpha \in A$ (resp. $\beta \in B$) is unmatched, then $d^\infty(\alpha,\Delta) \leq \delta$ (resp. $d^\infty(\beta,\Delta) \leq \delta$)).

Definition 1.4.3. Given two q-tame persistence modules $\mathbb{U}$ and $\mathbb{V}$, the **bottleneck distance** between $\mathbb{U}$ and $\mathbb{V}$ is the distance on the multisets of points $A = \mathrm{dgm}(\mu_\mathbb{U})$ and $B = \mathrm{dgm}(\mu_\mathbb{V})$ such that:

$$d_b(\mathbb{U},\mathbb{V}) := \inf\{\delta \,|\, \exists \delta\text{-matching between } A \text{ and } B\} \tag{1.4.58}$$

Proposition 1.4.4. As for the interleaving distnce, the bottleneck distance satisfies the triangle inequality, *i.e.* :

$$d_b(\mathbb{U},\mathbb{W}) = d_b(\mathbb{U},\mathbb{V}) + d_b(\mathbb{V},\mathbb{W}) \tag{1.4.59}$$

1.4.3 The Isometry Theorem

The isometry theorem is the main result which we will use in the latter part of this manuscript. It states the link (and indeed the equivalence) of the bottleneck distance and the interleaving distance. With it, we are guaranteed to be able to find interleavings between modules given that their distance for the bottleneck distance is bounded provided that we impose a certain hypothesis of tameness. Since in most practical cases we deal with q-tame or decomposable modules, we state the theorem as follows:

Theorem 1.4.1 (Isometry Theorem)**.** *Let $\mathbb{U}$ and $\mathbb{V}$ be two q-tame persistence modules. Then:*

$$d_i(\mathbb{U}, \mathbb{V}) = d_b(\mathbb{U}, \mathbb{V}) \tag{1.4.60}$$

Proof. The proof is naturally split into two different parts. On one hand we can establish the inequality:

$$d_i(\mathbb{U}, \mathbb{V}) \geq d_b(\mathbb{U}, \mathbb{V}) \tag{1.4.61}$$

which we will call the **stability theorem** and, on the other hand, we can look at the converse inequality

$$d_i(\mathbb{U}, \mathbb{V}) \leq d_b(\mathbb{U}, \mathbb{V}) \tag{1.4.62}$$

which we will call the **converse stability theorem**. In this manuscript, we will only give a proof of the stability theorem in the case where we have decomposable modules. For the full details of the proof, the reader is welcome to consult [1, §5]. Since we will only consider the case where we have a decomposable module, it is perhaps noteworthy to clarify how exactly we perform the extension to q-tame modules. This is done through statements about persistence measures in regions where the measure remains finite. Indeed the main ingredients are the so-called box inequalities, which locally relate the persistence measures of $\mathbb{U}$ and $\mathbb{V}$, and the interpolation lemma, which is a way of extending the persistence module naturally given by the interleaving over the two diagonals in $\mathbb{R}^2$ using the procedure explored in proposition 1.3.1 to a persistence module over the whole fringe comprised between the two diagonals. It turns out that these two statements are enough to ensure the result in its full generality.

Going back to the decomposable case, suppose that $\mathbb{U}$ and $\mathbb{V}$ are both decomposable persistence modules. Then we can write:

$$\mathbb{U} = \bigoplus_{i \in I} k[p_i^*, q_i^*] \text{ and } \mathbb{V} = \bigoplus_{j \in J} k[p_j^*, q_j^*] \tag{1.4.63}$$

where the index sets I and J are both finite. We must thus show that whenever we have a δ-matching between $\mathrm{dgm}(\mathbb{U})$ and $\mathrm{dgm}(\mathbb{V})$ we have $d_i(\mathbb{U}, \mathbb{V}) \leq \delta$. The result then simply follows by taking the infimum over all such δ. Given such a δ-matching, it is possible to rewrite the above direct sums over a common index set L such that:

$$\mathbb{U} = \bigoplus_{\ell \in L} \mathbb{U}_\ell \text{ and } \mathbb{V} = \bigoplus_{\ell \in L} \mathbb{V}_\ell \tag{1.4.64}$$

where each pair $(\mathbb{U}_\ell, \mathbb{V}_\ell)$ is one of the following:

1. a pair of matched intervals in the δ-matching,
2. $\mathbb{U}_\ell$ is unmatched and $\mathbb{V}_\ell = 0$,
3. $\mathbb{V}_\ell$ is unmatched and $\mathbb{U}_\ell = 0$.

In all three cases, by propositions 1.4.2 and 1.4.3 we have that $d_i(\mathbb{U}_\ell, \mathbb{V}_\ell) \leq \delta$. Finally, the result follows from lemma 1.4.2 ■

Lemma 1.4.2 (Distance between direct sums of persistence modules)**.** Let $(\mathbb{U}_\ell)_{\ell \in L}$ and $(\mathbb{V}_\ell)_{\ell \in L}$ be two families of persistence modules indexed by L. Furthermore, let $\mathbb{U} := \bigoplus_{\ell \in L} \mathbb{U}_\ell$ and $\mathbb{V} := \bigoplus_{\ell \in L}$. Then:

$$d_i(\mathbb{U}, \mathbb{V}) \leq \sup_{\ell \in L} \{d_i(\mathbb{U}_\ell, \mathbb{V}_\ell)\} \tag{1.4.65}$$

Proof of the lemma. Given δ-interleavings (Φ_ℓ, Ψ_ℓ) for each pair $(\mathbb{U}_\ell, \mathbb{V}_\ell)$, the direct sum maps $\Phi = \bigoplus_{\ell \in L} \Phi_\ell$ and $\Psi = \bigoplus_{\ell \in L}$ constitute a δ-interleaving of $\mathbb{U}$ and $\mathbb{V}$. Thus any upper bound on the $d_i(\mathbb{U}_\ell, \mathbb{V}_\ell)$ is an upper bound for $d_i(\mathbb{U}, \mathbb{V})$. In particular, this is true for the least upper bound, or supremum. ■

Chapter 2

Persistent Homology

The theory of persistence modules is often applied to persistent homology, a concept to which we have thus far only vaguely referenced. In fact, it is in the context of persistent homology that the theory of persistence modules was first studied. Our aim in this chapter is to give a formal introduction to the topic and to place ourselves in the context in which persistence homology is usually studied, *i.e.* topological data analysis.

For starters, we will recall some facts from homological algebra as well as some facts about simplicial objects [3, 4]. After giving some examples of how we retrieve some elements of the general theory in practice, we will ease ourselves into the theory of persistent homology by introducing filtrations, which we will then apply to study point clouds and their geometry, under some simple assumptions.

We then give some theoretical guarantees concerning the persistent homology of point clouds and how it relates to their geometry as well give an idea of how interleavings enter the game in order to ease things from the computational point of view. The results from the previous chapter, and in particular the Isometry Theorem (1.4.1), will play a crucial role in this and we will see exactly why and how all of these concepts are intertwined in this particular context.

Finally, we will give some explicit examples of persistence diagrams of some canonical examples we have computed ourselves.

2.1 Simplicial Complexes and Homological Algebra

We start off by giving a couple of basic definitions:

Definition 2.1.1. The **simplex category** Δ, is the category whose objects are totally ordered sets $[n] = \{0, 1, \cdots, n\}$, where $n \in \mathbb{N}$ and in whose morphisms are the order-preserving functions.

The morphisms of the simplex category are actually generated by two families of morphisms, namely

$$\delta_i^n : [n-1] \longrightarrow [n] \tag{2.1.1}$$

$$\sigma_i^n : [n+1] \longrightarrow [n] \tag{2.1.2}$$

where δ_i^n is the order-preserving injection whose image does not contain i and σ_i^n is the order-preserving surjection which sends $i \mapsto i$ and $i+1 \mapsto i$. These maps satisfy the following relations:

$$\delta_j \delta_i = \delta_i \delta_{j-1} \quad \text{if } i < j \tag{2.1.3}$$

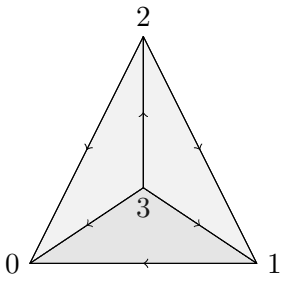


Figure 2.1: The object $[3] \in \Delta$ can be seen as a 3-simplex whose vertices are ordered as above.

$$\sigma_j \sigma_i = \sigma_i \sigma_{j-1} \quad \text{if } i \leq j \tag{2.1.4}$$

$$\sigma_j \delta_i = \begin{cases} \delta_i \sigma_{j-1} & \text{if } i < j \\ \text{id} & \text{if } i = j \text{ or } i = j+1 \\ \delta_{i-1} \sigma_j & \text{if } i > j+1 \end{cases} \tag{2.1.5}$$

These are the only relations, in the sense that any other relation can be expressed in terms of the relations explicited above. Intuitively, one can picture an object $[n]$ in this category as being the honest-to-goodness n-simplex which is directed according to the order relation on its vertices. The map σ_i above can be understood as the map which maps the simplex obtained by collapsing the ith vertex onto the face formed by its complementary points to the $(n-1)$-simplex. Similarly, the δ_i can be understood as being the map embedding the $(n-1)$-simplex onto the face of the n-simplex which doesn't contain i (while preserving the order of the vertices) as in figure 2.1.

Definition 2.1.2. A **simplicial object** in a category $\mathcal{C}$ is a functor $X : \Delta^{\text{op}} \to \mathcal{C}$. We denote by $s\mathcal{C}$ the category of simplicial objects in $\mathcal{C}$, whose morphisms are natural transformations.

2.1.1 Simplicial Homology

The main goal in this subsection is to define a framework in which we may apply all the heavy machinery developped in the context of homological algebra to categories which may not necessarily be abelian. In order to do this, we need first to recall some constructions and definitions in order to define what we will call *simplicial* or *singular* homology. Since we want to compute homology, our first goal will be to build a complex from which we may derive our homology from. We briefly recall what the notion of a chain complex means for abelian categories:

Definition 2.1.3. Let $\mathcal{A}$ be an abelian category, then a **complex in** $\mathcal{A}$, denoted by $x_\bullet$, is a sequence:

$$\cdots \xrightarrow{\partial_1} x_1 \xrightarrow{\partial_0} x_0 \xrightarrow{\partial_{-1}} x_{-1} \xrightarrow{\partial_{-2}} \cdots \tag{2.1.6}$$

such that $\partial_{n+1}\partial_n = 0$ for all n. A **morphism of complexes** $f_\bullet : x_\bullet \to y_\bullet$ is a family of morphisms of $\mathcal{A}$, $(f_n)_{n\in\mathbb{N}}$ such that the following diagram commutes:

$$\begin{array}{ccc} x_{n+1} & \xrightarrow{\partial_n} & x_n \\ \downarrow{\scriptstyle f_{n+1}} & & \downarrow{\scriptstyle f_n} \\ y_{n+1} & \xrightarrow{\partial_n} & y_n \end{array} \tag{2.1.7}$$

It is thus possible to define a **category complexes of** $\mathcal{A}$, $Ch(\mathcal{A})$, which is the category whose objects are complexes in $\mathcal{A}$ and whose morphisms are given by morphisms of complexes.

As previously remarked, we would like to extend such notions to the cases where our category of interest is not necessarily abelian. On the other hand, we could hope that we can build some kind of simplicial object of the category $\mathcal{C}$ from which we could construct a simplicial set. Using the $\mathbf{Free}_A : \mathbf{Set} \to \mathbf{Mod}_A$ functor, we can then funtorially assign a free module generated by the elements of the simplicial set we have at hand in order to then be able to consider chain complexes (which are now defined) and compute the homology of this complex. The first step is thus to construct this simplicial set.

Definition 2.1.4. Consider then a functor $Y : s\mathbf{Set} \to \mathcal{C}$ and an object $X \in \mathcal{C}$, we define a functor $S_\bullet(X) : \Delta^{\mathrm{op}} \to \mathbf{Set}$ as follows:

$$S_\bullet(X) : [n] \mapsto S_n(X) := \mathrm{Hom}_{\mathcal{C}}(Y[n], X) \tag{2.1.8}$$

We call $S_\bullet(X)$ a **singular set**.

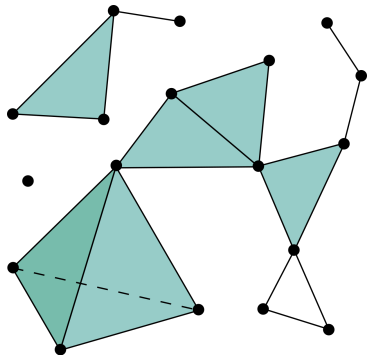


Figure 2.2: Intuitively, we can see a singular set as an ensemble of simplices arranged together in some shape or form.

We can now construct a free module by composing with the $\mathbf{Free}_A$ functor. Finally, we may define the maps ∂ giving rise to a chain complex. Indeed, we can achieve this by defining the following:

Definition 2.1.5. Given $M_\bullet \in s\mathbf{Mod}_A$, we define the **Moore complex** as the chain complex given by:

$$C(M)_n := M_n = M[n] \quad \text{and} \quad \partial_n := \sum_{i=0}^{n+1} (-1)^i M(\delta_i^{n+1}) \tag{2.1.9}$$

All in all, we have that the composition $M := \mathbf{Free}_A \circ S_\bullet(X) \in s\mathbf{Mod}_A$ yields a chain complex $C(M)$ of which we may then take the homology.

Definition 2.1.6. We call the **singular homology** or the **simplicial homology** the homology of the complex $C(M)$ obtained through the procedure above.

2.1.2 Some Familiar Examples

We now give some examples in which we will see how this construction naturally gives rise to some of the well-known and established types of homology.

Simplicial Homology in Topological Spaces

In our first description of the objects simplicial categories, we have already made allusion to some kind of possible geometrical realization of these objects. It turns out that we have a functor we will call the **geometric realization functor**, $\mathbf{Geo} : s\mathbf{Set} \to \mathbf{Top}$. To define it, we start by considering its definition on n-simplices:

$$\mathbf{Geo}([n]) := \left\{(x_0, \ldots, x_n) \in \mathbb{R}^{n+1} : 0 \leq x_i \leq 1, \sum x_i = 1\right\} \tag{2.1.10}$$

The definition extends naturally to any simplicial set X by taking:

$$\mathbf{Geo}(X) := \lim_{[n]\to X} \mathbf{Geo}([n]) \tag{2.1.11}$$

Morally speaking, this functor constructs a topological space in which we assign a standard simplex to every n-simplex of X and we glue these simplices in a way which is the same as the one in which the n-simplices lie in X. Doing this, we lose all notion of orientability of the simplices. With this geometric realization, we may now consider a singular set given by:

$$S_n(X) := \mathrm{Hom}_{\mathbf{Top}}(\mathbf{Geo}[n], X) \tag{2.1.12}$$

To define the corresponding singular homology, it then suffices to continue on with the steps above, while typically taking $\mathbb{Z}$ for the commutative ring A.

Posets or Graphs

Once again, we seek to give a canonical singular set given a poset. We start by remarking that each $[n] \in \Delta$ is actually also a poset, this allows us to give the following definition of a simplicial set:

Definition 2.1.7. Given a poset $(X, \leq)$, we define a simplicial set $N[X] : \Delta^{\mathrm{op}} \to \mathbf{Set}$ called the **nerve of** X by letting:

$$N[X] := \mathrm{Hom}_{\mathbf{Poset}}(-, X) \tag{2.1.13}$$

where $\mathrm{Hom}_{\mathbf{Poset}}(A, B)$ is the set of order preserving maps from A to B.

Note that as before with the general procedure outlined above, this prescribes what it means to take the singular homology of such a poset. As an interesting side remark, there is a natural extension of what we mean by the nerve of a category, which may not be **Poset**.

Definition 2.1.8. If $\mathcal{C}$ is a general category, we set the **nerve of** $\mathcal{C}$, $N[\mathcal{C}] : \Delta^{\mathrm{op}} \to \mathcal{C}$, to be:

$$N[\mathcal{C}] := \mathrm{Hom}_{\mathbf{Cat}}(-, \mathcal{C}) \tag{2.1.14}$$

where we see each $[n]$ as being equipped with a small category structure stemming from its structure as a totally ordered set.

Constructions à la Čech

We can use the notion of the nerve defined above to define a construction of Čech homology, which is recurrently used in topological data analysis. We start by remarking that given an open cover $\mathcal{U}$ of some topological space X, we may regard this cover as a poset where the order

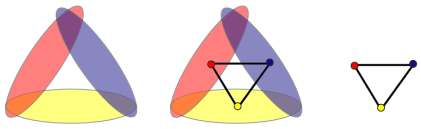


Figure 2.3: Given the open cover provided by the three shaded sets above in the plane, we can compute its nerve by linking two sets if they intersect each other.

relation is simply given by the inclusion. We can then regard the nerve of the covering as being defined in the same way as it was for posets:

$$N[\mathcal{U}] := \mathrm{Hom}_{\mathbf{Poset}}(-,\mathcal{U}) \tag{2.1.15}$$

Definining the homology as done previously defines a notion of homology which is similar to the one considered by Čech (although in the case of Čech, we consider cohomology instead).

Note that this is compatible with the usual way of understanding Čech homology, since the simplicial set constructed above is actually the same as the fibre product of the covering over X, in other words:

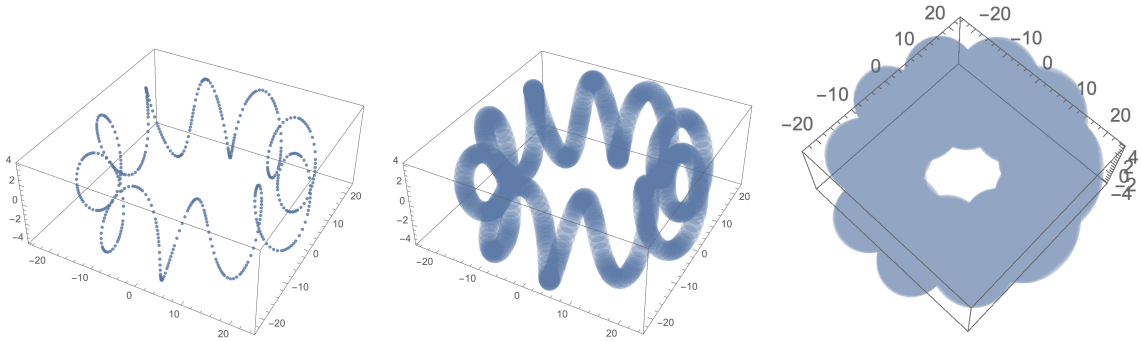
$$\mathrm{Hom}_{\mathbf{Poset}}([n],\mathcal{U}) = \underbrace{\mathcal{U} \times_X \cdots \times_X \mathcal{U}}_{n \text{ times}} \tag{2.1.16}$$

so it is indeed not really surprising that we recover the usual Čech homology in the end.

Point Clouds

At this point, we reach the main object of interest in the context of topological data analysis, *i.e.* point clouds. The reader might wonder exactly how all the machinery previously introduced in this chapter can be of use to study such objects. The main idea is to exploit some sort of "intrinsic geometry" that the point cloud might have. To illustrate this, we refer again to the toy model of the data set sparsed on the torus introduced in the previous chapter and illustrated in figure 2.4.

As we remarked then, the problem which we might encounter is the problem of geometrical features which may only appear at a certain scale. Concretely, it is necessary to introduce some kind of model for the data in order to really understand the motivations behind the use of homology in data analysis. For now and for the sake of illustrating the idea, let us suppose that our data lies on some manifold M embedded in some (potentially high dimensional space) $\mathbb{R}^d$. Our interest then is to see this data set as a random sample taken from this manifold M. Given the data set, we would like to infer something about its distribution and in particular, the topology of this manifold is of interest. On the other hand, as illustrated by our toy model, the data might exhibit different behaviour at different scales. Since the manifold M is unknown and inaccessible, there is no *a priori* way of choosing the scales at which we should study the data set. Instead, we study all possible scales in the hopes of extracting as much information as possible. Concretely, what we do is place a ball of radius t around each point of the point cloud. The notion of "scale" is thus captured by this radius t. By studying the topology of this inflated data set for a given t, we hope to recover some information about M and the topology of the distribution of the point cloud in $\mathbb{R}^d$. Finally, we let t grow and obtain a family of topological spaces dependent on this parameter, of which we want to study the topology and how it changes as t varies. We can appreciate this change explicitly in the example given in figure 2.4.



(a) $t = 0.01$, the scale of individual points

(b) $t = 0.06$, a helicoidal path appears

(c) $t = 0.25$, the structure of the torus appears

Figure 2.4: A data set sparsed along a helicoidal path on a torus when seen at different scales t.

2.2 Filtrations and Persistent Homology

As we have started to notice at the end of the previous section, persistence modules arise naturally as algebraic invariants of families of topological spaces. Such families called *filtrations* can be seen as sequences of topological spaces with continuous maps linking them (typically inclusions). The connection between **Top** and $\mathbf{Vect}_k$ is given by some functor, usually the homology functor, which turn filtrations into persistence modules. To start, it is helpful to give some vocabulary and set precise ideas in place so that we may explore this in more detail.

Definition 2.2.1. Let $(T, \leq)$ be a poset. A **filtration over** T, denoted $\mathcal{X}$, is a family of topological spaces indexed by T, $(X_t)_{t \in T}$, such that if $s \leq t \in T$ then, $X_s \subset X_t$.

Notice that we may see $\mathcal{X}$ as a particular representation of the poset $(T, \leq)$ in the category of topological spaces. If all these X_t are inside a common topological space X, then by applying the homology functor (taken over some field k) to the topological spaces X_t and to their maps of inclusion, we get one persistence module $H_p(\mathcal{X}) : \mathbb{R} \to \mathbf{Vect}_k$ for every degree of homology p.

Definition 2.2.2. Given X a topological space and a filtration $\mathcal{X}$ whose elements live in X, the collection of the $H_p(\mathcal{X})$ over all dimensions p is called the **persistent homology of** $\mathcal{X}$ and is denoted by $H_*(\mathcal{X})$.

This means that each $H_p(\mathcal{X})$ has a vector space $H_p(X_i)$ at every index $i \in T$. In particular, consider $i \leq j \in T$, then there are canonical inclusions $X_i \hookrightarrow X_j$ which induce a map at the homology level $H_p(X_i) \to H_p(X_j)$.

Definition 2.2.3. For $i \leq j \in T$, $\mathrm{Im}(H_p(X_i) \to H_p(X_j))$ is called a **p-th persistent homology group of** $\mathcal{X}$.

From chapter 1, we know that persistent homology is decomposable as a persistence module under certain conditions, namely:

1. when the index set T is finite;

2. when the modules $H_*(\mathcal{X})$ are pointwise finite-dimensional;

3. when $T = \mathbb{R}$ and the modules are q-tame.

These conditions tend to be satisfied when dealing with multiple practical situations such as a Morse function on a compact manifold, or the calculation of persistent homology of a point cloud. An important remark is that for this to be true, we must absolutely consider the homology to be taken over some field, as opposed to over $\mathbb{Z}$ as usual. This is because to guarantee the decomposition under the conditions above, we have used Gabriel's theorem (1.1.1), which is only valid for persistence modules taking values over vector spaces. This can be problematic, in the sense that it would at first glance appear that we lose the notion of being able to compute torsion appropriately, albeit there is a way of being sensible to the effects of torsion by varying the field of computation of the homology. In practice, however, calculations are usually done over $\mathbb{Z}/2\mathbb{Z}$.

With the above concepts introduced, it is helpful to introduce some terminology as well as some examples of filtrations. If we let X be a topological space, we can construct a filtration by considering a function $f : X \to \mathbb{R}$. Each value $i \in \mathbb{R}$ will give rise to a topological subspace of X given by $F_i := f^{-1}(]-\infty, i])$. We can now consider the family $\mathcal{F}$ of these sublevel sets F_i, which is clearly a nested family of topological spaces, *i.e.* $i \leq j \implies F_i \subset F_j$. The following proposition, proved by Chazal *et al.*, gives practical situations in which $H_*(\mathcal{F})$ is q-tame.

Proposition 2.2.1 (Chazal *et al.* [1])**.** The function $f : X \to \mathbb{R}$ gives rise to a q-tame module $H_*(\mathcal{F})$ in the following two scenarios:

- X is finitely triangulable, *i.e.* homeomorphic to the underlying space of a finite simplicial complex, and f is continuous;
- X is locally triangulable, f is continuous, bounded from below and proper (*i.e.* preimages of compact intervals are compact).

Note that while these conditions are certainly sufficient, they are not necessary as noted in [2, §4]. At this level of generality, the study of such functions can be quite interesting from a geometric standpoint, by for instance studying sublevel sets of Morse functions over a manifold [14]. However, for the rest of this chapter it is illuminating to continue our discussion under the umbrella of topological data analysis. We will consider a finite point cloud P in $\mathbb{R}^d$, which is supposed to live on (or is at least Hausdorff close to) some compact manifold M. Here, we will set our manifold $X = \mathbb{R}^d$, which is locally triangulable. We set the Morse function f to be $d_P : \mathbb{R}^d \to \mathbb{R}$, the Euclidean distance to P. In what will follow and in accordance to our previous notations, we note $\mathcal{P}$ the induced filtration by sublevel sets of d_P.

Čech and Delaunay Filtrations

The so-called nerve lemma helps to ease the homology computations to be performed, especially since we are provided with a canonical cover of $P_i := d_P^{-1}([0, i])$, given by the balls of radius i around each point $p \in P$.

Lemma 2.2.1 (Nerve Lemma, [12])**.** Let X be a paracompact space, and let $\mathcal{U}$ be an open cover of X such that the $(k+1)$-fold intersections of elements of $\mathcal{U}$ are either empty or contractible for all $k \in \mathbb{N}$. Then, there is a homotopy equivalence $N[\mathcal{U}] \to X$, where $N[\mathcal{U}]$ denotes the Čech nerve.

Notice that the canonical cover of the space P_i provided by the balls of radius i around each point is actually a closed cover, which impedes the direct application of the nerve lemma.

In fact, in general, the nerve lemma is false for closed covers and finding necessary conditions for it to be true turns out to be a complicated question in its own right. Luckily, the following result [6] allow us to guarantee the validity of the nerve lemma in this particular context.

Proposition 2.2.2. The nerve lemma holds for a closed cover $\mathcal{U}$ as soon as each $(k+1)$-fold intersection V admits a retraction $r : W \to V$ from some open neighbourhood W.

Notice that for the intersection of finitely many Euclidean balls, this is indeed the case. Let us thus denote $\check{C}_i(P)$ the nerve of the covering provided by the closed balls of radius i around points of P. By the nerve lemma and the above proposition, we have pointwise homotopy equivalence between the spaces $\check{C}_i(P)$ and P_i. It is a result by Chazal *et al.* that this pointwise homotopy equivalence of spaces actually extends to an equivalence between the filtrations $\mathcal{C}(P)$ and $\mathcal{P}$. To be more precise:

Lemma 2.2.2 (Persistent Nerve [8])**.** Let $X \subset X'$ be two paracompact sets, and let $\mathcal{U}$ and $\mathcal{U}'$ be open covers of X and X' respectively, based on finite parameter sets $A \subset A'$. Assume that $U_a \subset U'_a$ for all $a \in A$ and that all the $(k+1)$-fold intersections of elements of $\mathcal{U}$ (resp. $\mathcal{U}'$) are either all empty of contractible for all $k \in \mathbb{N}$. Then, for each dimension p, the following diagram, induced by the inclusion map $X \hookrightarrow X'$, commutes:

$$\begin{array}{ccc} H_p(X) & \longrightarrow & H_p(X') \\ \downarrow & & \downarrow \\ H_p(N[\mathcal{U}]) & \longrightarrow & H_p(N[\mathcal{U}']) \end{array} \tag{2.2.17}$$

In particular, the pointwise homotopy equivalences between $\check{C}_i(P)$ and P_i at all indices i induce an isomorphism of persistence modules $H_p(\mathcal{C}(P)) \to H_p(\mathcal{P})$ at each homology degree p. This means that the two filtrations have the same persistent homology.

In fact, there are other so-called *simplicial filtrations* which give rise to the same persistence modules at the homology level, or at least in some cases to interleaved persistence modules. This has as a consequence that we can ease calculations if we pick the right simplicial complex to analyse given a situation. We start by remarking that the Čech nerve actually provides a triangulation of the sublevel set P_i at any given i. On the other hand, we could imagine using other types of triangulations which may be easier to compute and have lighter geometric predicates. For instance, we could consider:

Definition 2.2.4. The **i-Delaunay complex** (also known as the **α-complex** in the literature), denoted by $D_i(P)$, is the simplicial complex which has one k-simplex per $(k+1)$-tuple of points of P circumscribed by a ball of radius at most i and containing no other point of P in its interior. The **Delaunay filtration**, $\mathcal{D}(P) := \{D_i(P)\}_{i\in\mathbb{R}}$

In small dimension ($d \leq 4$), the Delaunay complex eases computations and is well optimized in libraries such as `gudhi` [18], it is thus the most practical tool for performing calculations on point clouds in small dimensions. For the unfamiliar reader, the definition given in terms of circumscribed balls might seem a bit strange, so we make a couple of useful remarks, which will help the reader understand what is really going on.

Remark 2.2.1. For every $i \in \mathbb{R}$, the i-Delaunay complex of P is actually a subcomplex of the Delaunay triangulation of P. We can understand the Delaunay triangulation in the following way: if we $P \subset \mathbb{R}^d$, we define a paraboloid in $\mathbb{R}^{d+1}$ given by $x_{d+1} = \sum_{i=1}^{d} x_i^2$. Since every point of P lies in the subspace spanned by $x_1 \cdots x_d$, we can look at their image on the paraboloid. We

then take the convex hull of this set of points in $\mathbb{R}^{d+1}$. This will yield a convex polytope. By projecting down the edges of this polytope back on $\mathbb{R}^d$ we obtain the full Delaunay triangulation of P. As an interesting side remark, we note that this way of understanding the Delaunay triangulation provides us with some algorithmic insights on how we may actually compute the triangulation in an efficient manner.

Notice that since the i-Delaunay complex is a subcomplex of the Delaunay triangulation of P it embeds linearly in $\mathbb{R}^d$ under the assumption that the points are generic, *i.e.* that there are no $(d+2)$-cospherical points and no $(d+1)$ affinely dependent points in P. We denote the image of this embedding also by $D_i(P)$. Edelsbrunner [11] worked out that the sublevel set P_i actually deformation retracts onto $D_i(P)$, so that the inclusion map $D_i(P) \hookrightarrow P_i$ is actually a homotopy equivalence. It turns out that here, we are also in a situation where we can actually exhibit an isomorphism between the persistent homology of the filtration $\mathcal{P}$ and $\mathcal{D}(P)$. Thus far, we have that:

Theorem 2.2.3. *The filtrations $\mathcal{P}$, $\mathcal{C}(P)$ and $\mathcal{D}(P)$ all have the same persistent homology.*

Despite having this result which might lead us to think that we have drastically improved our performance, we still have to face the so-called curse of dimensionality. Indeed, while Delaunay filtrations perform quite well in dimensions 2, 3 and sometimes 4 (for a reasonable point cloud), beyond dimension 4, we are faced with serious computational problems. Already for a point cloud in dimension 4, the Čech complex can be impossible to deal with. Indeed, the number of simplices grows towards unmanageable numbers from the memory point of view, to the extent that performing calculations becomes completely untractable, typically having to deal with an order of dozens of billions of simplices. We also have the problem of running time. Indeed, as the dimension grows, the running time for our computations does too. In the next section, we shall explore these problems in more detail as well as introduce another complex and filtration, the *Rips-Vietoris complex*, which will help us deal with some of these computational difficulties.

2.3 Geometry of Point Clouds

So far, we've mostly stated facts about the point cloud itself, but most of the time, we are under the assumption that this data is a discrete sample coming from an underlying (compact) manifold. Note that in all realistic purposes, we will have some noise introduced on these points, so we must be careful about also imposing a condition on the fact that the points are not necessarily *on* the manifold itself, but rather lie a bounded Hausdorff distance away from it.

- **Data model**: The finite point cloud $P \subset \mathbb{R}^d$ is ε-close in the Hausdorff distance to an unknown compact set $K \subset \mathbb{R}^d$.

Ideally, we would like to be able to infer something about the topology of K itself by studying the persistent homology of the point cloud P. However, it is clear that just demanding that K be a compact set is too much to ask as this set could potentially be quite wild. Due to the finiteness of P, we cannot hope to recover most of the topology of K under these assumptions alone. Chazal *et al.* [8] have actually studied this in detail and have provided theoretical guarantees on the topological inference that is possible to make given a point cloud around the compact set K, provided that this compact set satisfies certain geometrical properties. In what follows, we introduce and study these geometrical properties in order to use Chazal's results to give theoretical guarantees on topological inferences on the compact set K.

2.3.1 Weak Feature Size and Doubling Dimension

In what will follow, $K \subset \mathbb{R}^d$ is a compact set and we denote by $d_K(x)$ the Euclidean distance from the point $x \in \mathbb{R}^d$ to K.

Definition 2.3.1. Let K be a compact set in $\mathbb{R}^d$. The **projection set** $\Pi_K(x)$ is the set defined by:

$$\Pi_K(x) := \arg\min_{y \in K} \|x - y\| \tag{2.3.18}$$

This set is never empty, but it can have more than one element. The points where this is the case are said to lie on the *medial axis of* K. Concretely:

Definition 2.3.2. The **medial axis of** K, denoted by $\mathfrak{M}(K)$, is the set of points $x \in \mathbb{R}^d$ for which $\#\Pi_K(x) \geq 2$. This set is in general neither open nor closed.

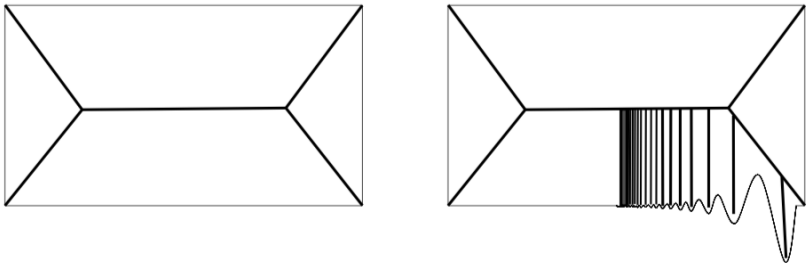


Figure 2.5: The medial axes for two different compact sets illustrating the fact that it can happen that the medial axis is not open (left rectangle) or closed (right rectangle). The figure is taken from [13]

.

Remark that, the function d_K^2 is differentiable everywhere outside the closure of the medial axis. It is thus possible to find its gradient everywhere outside of this region. We extend this notion to the medial axis itself in the following way:

Definition 2.3.3. The **generalized gradient** function $\nabla_K : \mathbb{R}^d \to \mathbb{R}^d$ given by:

$$\nabla_K(x) := \begin{cases} \frac{x - c(\Pi_K(x))}{d_K(x)} & \text{if } x \in \mathbb{R}^d \setminus K \\ 0 & \text{if } x \in K \end{cases} \tag{2.3.19}$$

where $c(\Pi_K(x))$ denotes the centre of the ball enclosing $\Pi_K(x)$ with minimal radius.

In the following proposition, we give multiple important properties of this generalized gradient which will allow us to integrate the generalized gradient vector field using Euler schemes.

Proposition 2.3.1. The generalized gradient vector field is semi-Lipschitz and the map $x \mapsto \|\nabla_K(x)\|$ is lower-semicontinuous, *i.e.* :

$$(\nabla_K(x) - \nabla_K(y)) \cdot (x - y) \leq \frac{1}{i}(x - y)^2 \qquad \forall i > 0 \quad \forall x, y \notin K_i \tag{2.3.20}$$

$$\liminf_{y \to x} \|\nabla_K(y)\| \geq \|\nabla_K(x)\| \qquad \forall x \in \mathbb{R}^d \tag{2.3.21}$$

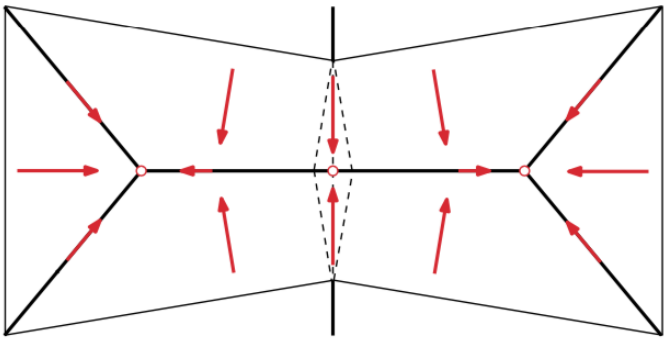


Figure 2.6: The generalized gradient ∇_K of d_K and its associated flow in red. In bold is the medial axis of K. The circles indicate critical points of d_K outside of K. The figure is taken from [13]

.

The above implies that as the integration step decreases, the Euler scheme converges uniformly to a continuous flow $\mathbb{R}^+ \times \mathbb{R}^d \to \mathbb{R}^d$ which is right differentiable and whose right derivative is ∇_K [13]. It is also possible to give a notion of critical points for this generalized gradient.

Definition 2.3.4. A point $x \in \mathbb{R}^d$ is said to be **critical** if its generalized gradient $\nabla_K(x) = 0$. A value $v = d_K(x)$ for a critical x is called a **critical value of** d_K. In particular, all points of K are critical and 0 is a critical value.

The gradient flow obtained from the integration of the generalized gradient vector field yields a deformation retraction of sublevel sets under certain conditions. More precisely:

Lemma 2.3.1. If $0 < i < j$ are such that there is no critical value of d_K in the interval $[i, j]$, then K_j deformation retracts onto K_i via the gradient flow of d_K, so that the inclusion map $K_i \hookrightarrow K_j$ is a homotopy equivalence.

It's under the context of this result that we are able to define the following:

Definition 2.3.5. The **weak feature size of** K, denoted by $\text{wfs}(K)$, is the quantity defined by:

$$\text{wfs}(K) := \inf\{i > 0 \mid i \text{ is a critical value of } d_K\} \tag{2.3.22}$$

Finally, we introduce the concept of doubling dimension:

Definition 2.3.6. The doubling dimension of a metric space (X, d) at scale r is the smallest positive integer m such that any d-ball of radius r in X can be covered by $2m$ balls of radius $r/2$. The doubling dimension of (X, d) is the supremum of the doubling dimensions over all scales $r > 0$.

Since we have this geometrical condition on the compact set K turns out to be a quite useful one in order to show a few results concerning the stability of the persistent homology of point clouds, *i.e.* whether it is possible or not to read off the homology of the compact set K based off of a barcode of a point cloud close enough to K. From now on, we revise the data model we previously gave ourselves and include the condition that K have at least positive weak feature size.

- **Data model:** The input is an n-points set P in Euclidean space $\mathbb{R}^d$, located ε-close in the Hausdorff distance to some unknown compact set K with positive weak feature size and small (constant) doubling dimension m.

The last condition is set in order to bound the number of possible simplices which might arise in the computation of the simplicial complexes we shall consider. Indeed, our best hope is that the running time of the computation scales like $2^{O(m^2)}n$ if n is the number of points in the point cloud, so having a bound on m is crucial if we want the computations to actually be feasible.

2.3.2 Sweet Ranges

Since our data model bases itself on the fact that the data intrinsically lives on (or Hausdorff close to) some compact set K, we would like to give a result linking the homology of K with the point cloud itself. There are many results in this direction, most of which are due to Chazal *et al.* Most of these rely on the weak feature size of this compact set K being non-zero. For instance, Chazal and Lieutier [7] proved the following result:

Theorem 2.3.2. *Let K be a compact set in $\mathbb{R}^d$ with $wfs(K) > 0$. Let $P \subset \mathbb{R}^d$ be a point cloud such that P is ε-close to K in the Hausdorff distance. If*

$$d_H(K, P) \leq \varepsilon < \frac{1}{4} wfs(K), \tag{2.3.23}$$

then, for any levels i, j such that $\varepsilon < i < i + 2\varepsilon \leq j < wfs(K) - \varepsilon$, the persistent homology group $Im(H_*(P_i) \to H_*(P_j))$ induced by the inclusion map $P_i \hookrightarrow P_j$ is isomorphic to $H_*(K_r)$ for any $r \in]0, wfs(K)[$.

Proof. We give a sketch of proof of this result. The first thing we note is that the inequalities in the hypotheses actually imply that we have the following set of inclusions:

$$K_{i-\varepsilon} \hookrightarrow P_i \hookrightarrow K_{i+\varepsilon} \hookrightarrow P_j \hookrightarrow K_{j+\varepsilon} \tag{2.3.24}$$

Note that due to lemma 2.3.1, we actually have homotopy equivalences between $K_{i-\varepsilon}, K_{i+\varepsilon}$ and $K_{j+\varepsilon}$, so that at the homology level the composed maps above actually become isomorphisms. By factorizing the maps above through the image of the map $H_*(P_i) \to H_*(P_j)$ induced by the inclusion $P_i \hookrightarrow P_j$ and using the fact that we know that the composed maps $H_*(K_{i-\varepsilon}) \to H_*(K_{i+\varepsilon}) \to H_*(K_{j+\varepsilon})$ are isomorphisms, we get the desired result. ∎

Remark 2.3.1. Note that considering the persistent homology groups as opposed to the naïve approach of considering only the persistent homology of the offsets of P is necessary in order to capture the homology of the small offsets K_r or K. In fact, in figure 2.7 we exhibit a counter-example which shows that not a single P_i has the same homological type as K_r.

Another way of understanding the previous theorem is to say that we must look at the homological type of a given offset of the point cloud P_i, where we have killed the parts of the module which die after a short time in the offset P_j.

This theoretical guarantee concerning the ability to read off the homological type of K given these hypotheses justifies the following definition:

Definition 2.3.7. A **sweet range** is an interval, T, over which it is possible to read off the homology of the compact set K off of a persistent homology diagram.

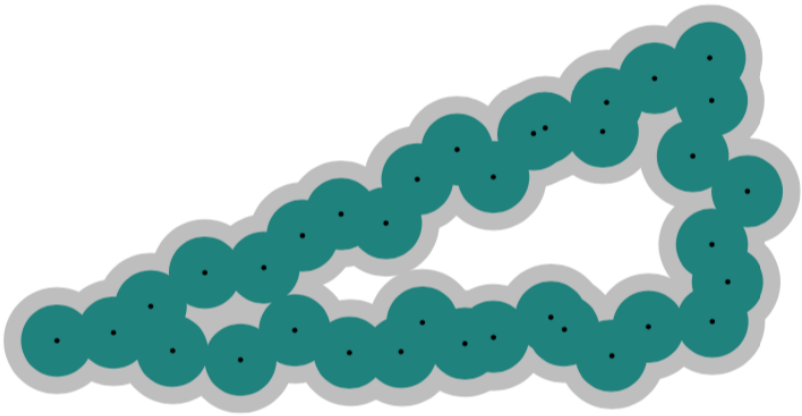


Figure 2.7: Sampling P of a triangle K in the plane. Because of the small angle of the triangle and the noise in the data, there is no single P_i which has the same homological type as K. This illustrates the need to consider the image of the homology of the offset P_i onto that of the offset P_j.

Theorem 2.3.2 gives us such a result for a possible sweet range. Indeed, one of the consequences of theorem 2.3.2 is:

Corollary 1. *Let K be a compact set in $\mathbb{R}^d$ with positive weak feature size. Let $P \subset \mathbb{R}^d$ be a point cloud such that $d_H(P, K) = \varepsilon < \frac{1}{4} wfs(K)$, then there is a sweet range $T =]\varepsilon, wfs(K) - \varepsilon[$ whose intersection with the barcode of the offset filtration $\mathcal{P}$ has the following properties:*

- *The intervals that span T encode the homology of K, i.e. their number for each homology degree p is the same as the dimension of $H_p(K_r)$, for any $r \in]0, wfs(K)[$;*
- *The remaining intervals have length at most 2ε.*

In particular, the corollary tells us that it is possible to separate the intervals of the decomposition of the persistent homology into two parts. On the one hand, the *topological signal*, which comprises all intervals which span the sweet range described above, and the *noise*, which are the left over short intervals lying close to the diagonal. This theoretical guarantee is reassuring, as it allows us to infer something on data, provided that we make some hypotheses on the geometry of the compact set from which it stems. There are actually further relaxations to the geometrical conditions which the compact set K must obey and the reader is strongly recommended to consult [2, §4, 5] for further details on this, but for our purposes, the result above suffices to show that it is possible to make topological inferences based on the result of the persistent homology calculation performed on the point cloud P.

2.4 Towards Computational Sustainability

At this point, we have a sensible data model with some fairly constraining assumptions on the geometry of the point cloud P and its underlying compact set K. These are enough to let us simplify the calculations involved when computing the homology of different complexes. We will introduce multiple tools in this section in order to achieve this. We start by introducing the Rips-Vietoris complex, whose main purpose is to simplify the geometric predicates which we encounter in the construction of simplicial complexes. This is because while Delaunay

triangulation remains a very good computational option for low dimensions, this is no longer true once we are past dimension 4. The Čech filtration we previously introduced is also not an option, due to the number of simplices it involves and due also to the complexity of establishing the (non)-intersection of intersections of balls in high dimensional spaces. On the other hand, a simple distance critereon would be ideal, hence the introduction of Rips-Vietoris.

Definition 2.4.1. The **Rips-Vietoris complex** of a point cloud $P \subset \mathbb{R}^d$ of parameter i, denoted by $R_i(P)$, is the simplicial complex which has one k-simplex per $(k+1)$-tuple of points of P whose Euclidean diameter is at most i. The **Rips-Vietoris filtration** is the indexed family $\mathcal{R}(P) := \{R_i(P)\}_{i\in\mathbb{R}}$

It turns out that this complex is actually quite handy because it turns out that the Čech complex and the Rips-Vietoris complex are actually *multiplicatively interleaved*, *i.e.* :

$$\forall i > 0,\ \check{C}_i(P) \subset R_{2i}(P) \ \text{ and } \ R_i(P) \subset \check{C}_i(P) \tag{2.4.25}$$

This turns into the typical additive interleaving we explored for persistence modules by simply considering the log-scale. Notice that doing this, we have actually introduced a new persistence modules both for the Rips filtration and the Čech filtrations which we may define as follows:

Definition 2.4.2. Let $\mathcal{X}$ be a filtration over $\mathbb{R}$. The **log-scale persistent homology of a filtration** $\mathcal{X}$, denoted $H_*(\mathcal{X}^{\log})$ is the persistence module obtained by taking the homology functor of the following filtration:

$$\mathcal{X}^{\log} := \{X_i^{\log} := X_{2^i}\}_{i\in\mathbb{R}} \tag{2.4.26}$$

And so, we may reformulate our previous remark of the multiplicative interleaving as follows:

Proposition 2.4.1 (de Silva *et al.* [10])**.** Let $P \subset \mathbb{R}^d$ be a point cloud and let $\mathcal{R}(P)$ and $\mathcal{C}(P)$ denote its Rips-Vietoris and Čech filtrations respectively. Let:

$$\theta_d := \sqrt{\frac{d}{2(d+1)}} \in \left[\frac{1}{2}, \frac{1}{\sqrt{2}}\right] \tag{2.4.27}$$

and let us define a rescaled version of the Rips-Vietoris complex, $\tilde{\mathcal{R}}(P)$ as follows:

$$\tilde{R}_i(P) := R_{i\sqrt{\frac{2}{\theta_d}}}(P) \tag{2.4.28}$$

Then, the log-scale persistent homologies of the filtrations $\tilde{\mathcal{R}}(P)$ and $\mathcal{C}(P)$ are $\sqrt{2\theta_d}$-interleaved as persistence modules.

Recall that there is an isomorphism of persistence modules between $\mathcal{D}(P)$ and $\mathcal{C}(P)$, so a similar statement can be made between Delaunay filtrations and Rips-Vietoris filtrations.

Remark 2.4.1. Recall that this interleaving implies, through the isometry theorem for persistence modules (theorem 1.4.1), that the log-scale persistence diagrams for both of these filtrations remain $\sqrt{2\theta_d}$-close in the bottleneck distance. This allows us to translate the results into what they mean as far as the persistence diagrams are concerned.

What this means is that up to some error in the persistence diagram, we can compute the persistent homology of the Rips-Vietoris filtration as opposed to considering Čech or Delaunay filtrations. Actually, this notions of computing diagrams up to some potential error is what is going to give us some room to manoeuvre away from the computational difficulties we're

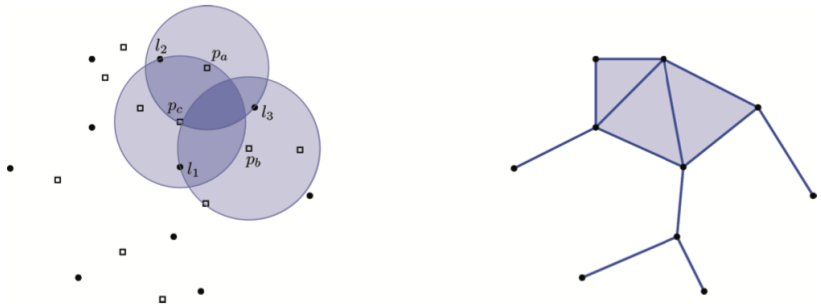


Figure 2.8: On the left we see a set of landmarks L (dots) and witnesses P (squares). The point p_a witnesses vertex l_2 and edge $\{l2, l3\}$. Point p_b witnesses $\{l_3\}$ and $\{l_3, l_1\}$. Point p_c witnesses $\{l_1\}$, $\{l_1, l_2\}$ and $\{l_1, l_2, l_3\}$. Thus, triangle $\{l_1, l_2, l_3\}$ belongs to the witness complex $W_0(L, P)$ shown to the right.

exhibiting. Indeed, the next problem we face beyond the complexity of geometric predictes is simply the number of simplices which we must consider. On the other hand, if we consider a random subset of the full point cloud, we could hope that it "looks" just like the original point cloud. If this was the case, we wouldn't need to compute the homology of the filtration over the full point cloud, but rather simply consider a small subset of this point cloud and compute the homology of the filtration over this more computationally reasonable set. Since this smaller cloud still looks like the original set, we can hope to recover the persistent homology of the full filtration, up to some error. This is a concept called *witness filtration*. The main idea behind this filtration is to use a weaker version of the Delaunay predicate, based only on distance comparisons. In order to make all of these concepts precise, we must introduce some further verbiage.

Definition 2.4.3. Let $P \subset \mathbb{R}^d$ be a point cloud and let $L \subset P$ be a subset of P, which we will call a **landmark set of** P. A point $p \in P$ is an **i-witness of a simplex** $\sigma \subset L$ if:

$$\|p - l\| \leq \|p - l'\| + i \text{ for all } l \in \sigma \text{ and all } l' \in L \setminus \sigma \tag{2.4.29}$$

An illustration of what this definition encapsulates is shown in figure 2.8.

Remark 2.4.2. Note that the Delaunay predicate corresponds to taking $i = 0$ and letting l' range over the entire set L as opposed to $L \setminus \sigma$. If p and σ satisfy the Delaunay predicate, we call p a **strong witness** of σ.

With this definition, we can now introduce the concept of a witness filtration as follows:

Definition 2.4.4 (de Silva, [9])**.** Given $i \in \mathbb{R}$, the **i-witness complex** of the pair (L, P) denoted by $W_i(L, P)$, is the maximal simplicial complex of vertex set L whose simplices are i-witnessed by points of P. The 0-witness complex is simply called the **witness complex of** (L, P). The **witness filtration**, denoted $\mathcal{W}(L, P)$, is the indexed family $\{W_i(L, P)\}_{i \in \mathbb{R}}$.

In fact the witness complex is known to be a subcomplex of the Delaunay triangulation of P [5]. Under certain conditions, it is possible to establish an interleaving between the Čech filtration and the witness filtration of a given landmark set $L \subset P$.

Theorem 2.4.1 (Chazal and Oudot [8])**.** *Let K be a connected compact set in $\mathbb{R}^d$ and let $L \subset P \subset \mathbb{R}^d$ be finite sets such that*

$$d_H(K, P) \leq d_H(P, L) = \varepsilon < \frac{1}{8} diam(K), \tag{2.4.30}$$

where $diam(K)$ is the Euclidean diameter of the compact connected set K. Then,

$$\forall i \geq 2\varepsilon,\ \check{C}_{\frac{i}{4}}(L) \subset W_i(L,P) \subset \check{C}_{8i}(L) \tag{2.4.31}$$

It remains to show that we may still infer a sweet range from the approximations we made above. This is indeed the case as was shown by Chazal *et al.*, provided that a relevant choice of landmarks is made, the result is very similar to what we saw previously in theorem 2.3.2.

Theorem 2.4.2 (Sweet ranges, [8])**.** *Let K be a compact set in $\mathbb{R}^d$ with positive weak feature size and let P be a point cloud in $\mathbb{R}^d$ and $L \subset P$ a subset of landmarks such that:*

$$d_H(K,P) \leq d_H(P,L) = \varepsilon < \min\left\{\frac{1}{8}diam(K),\ \frac{1}{2^{11}+1}wfs(K)\right\}, \tag{2.4.32}$$

then, there is a sweet range

$$T = \left]\log_2(4\varepsilon), \log_2\left(\frac{wfs(K)-\varepsilon}{8}\right)\right[\tag{2.4.33}$$

whose intersection with the log-scale barcode of the witness filtration $\mathcal{W}(L,P)$ has the following properties:

- *The intervals that span T encode the homology of K;*
- *The remaning intervals have length at most 6.*

Chapter 3

Applications

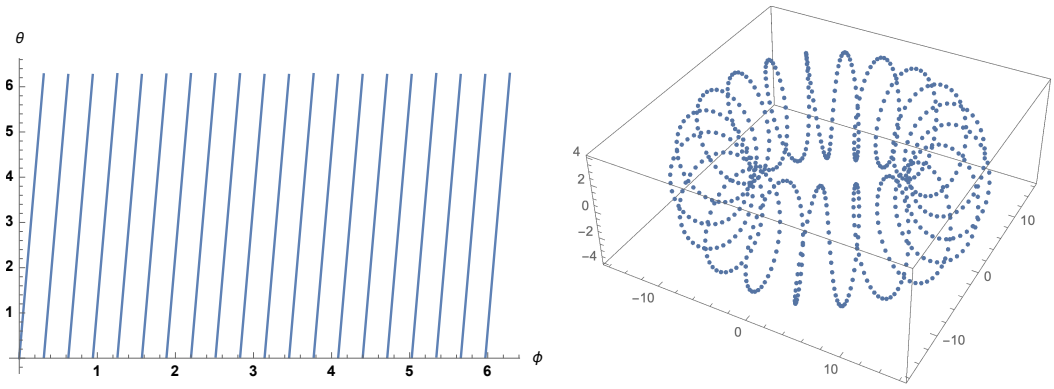
In this chapter, we will illustrate the concepts introduced in chapter 2 with some examples and explore how torsion can be uncovered by performing the homology computations over different finite finite fields.

3.1 The Torus Toy Model

We start by giving some explicit results for the example we have been talking about time and time again throughout this manuscript: the torus toy model. Here, we choose an embedding of the torus in $\mathbb{R}^3$ parametrically defined by:

$$x = (R + r\cos\theta)\cos\varphi \tag{3.1.1}$$
$$y = (R + r\cos\theta)\sin\varphi \tag{3.1.2}$$
$$z = r\sin\theta \tag{3.1.3}$$

Furthermore, we choose a closed curve on the torus defined by $\theta = \ell\varphi \mod 2\pi$. This defines a curve which curves anticlockwise on the torus p times. For the rest of our discussion, we shall fix $\ell = 20$ for all computations. The set-up is depicted in figure 3.1. In order to choose points



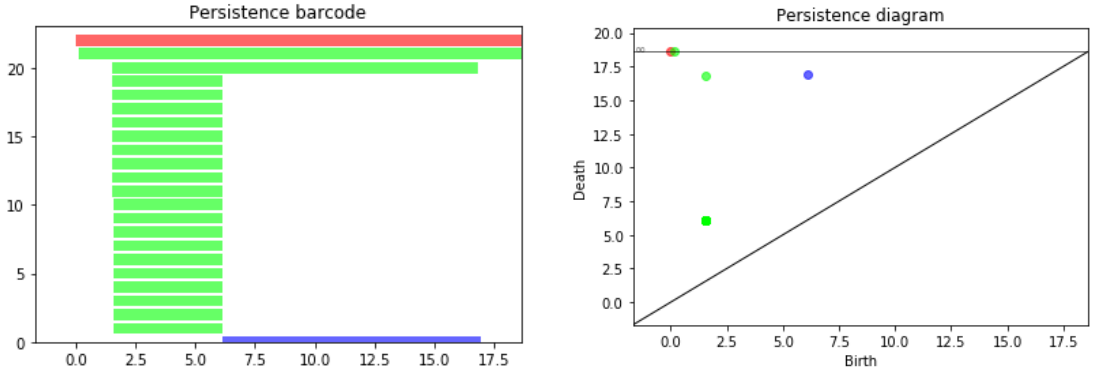
(a) $\theta = 20\varphi$ on the flat torus

(b) $\theta = 20\varphi$ when embedded in $\mathbb{R}^3$

Figure 3.1: Our choice of embedding and curve. In our case, we consider $R = 12$ and $r = 4$

randomly sparsed throughout this curve, we let φ range over the integers from 0 to 2000. This yields a distribution which will not be a uniform distribution, due to our embedding in $\mathbb{R}^3$, but we make up for this lack of uniformity with density, choosing a point cloud of 2000 points, so

that even at tiny scales, we can regard the data set as having a single connected component. Since we are in low enough dimension, the Delaunay filtration is the optimal computational tool to calculate the persistent homology of this point cloud using the `gudhi` package [18] and use $\mathbb{Z}/2\mathbb{Z}$ as our base field in order to accelerate calculations.



(a) Barcode of the data set.

(b) Persistence diagram of the data set.

Figure 3.2: Persistence diagram and barcode of the torus data set. In red we can read off the H_0, in green the H_1 and in blue the H_2 components of the persistent homology as calculated over $\mathbb{Z}/2\mathbb{Z}$. We have excluded the intervals of length less than 0.3 in order for the diagram not to be cluttered with entries close to the diagonal (*i.e.* tiny bars in the barcode).

Note that the barcode and the persistence diagram both hold exactly the same information. Nevertheless, it is practical to have both displayed in parallel, as it is sometimes difficult to read the multiplicity of a given point off of the persistence diagram, and this information is readily available on the barcode. In this case, for example, the multiplicity of the H_1 component of the persistent homology tells us about the way we have embedded the closed curve in $\mathbb{R}^3$.

By recalling that what is plotted by `gudhi` in the persistence diagrams and barcode of the Delaunay filtrations is actually the square of the scale parameter, we may understand qualitatively what is going on in the diagram and the barcode as follows. Since the points lie close enough to each other, after a certain time, the balls grow large enough so that we have one single connected component. This is the H_0 bar we observe being born at very small radius and perdure until infinity. At the same time, as soon as the set of balls has only a single connected component, there is the creation of an interval module of H_1, which represents the chain going around the helicoidal path. As the balls continue to grow larger, the loops end up meeting first along the inner radius of the torus. This is where we see exactly 20 bars of H_1 being born at the same time: they are the representatives of the cycles which go around each of the 20 circles. Meanwhile, the original cycle that went along the helicoidal path gets sent to the cycle of the inner radius, since it is homologically equivalent to the latter. The 20 cycles born during this merger persist until eventually the balls are large enough to touch at the outer radius of the torus, too. At that point, 19 of these bars die, leaving only the cycle which goes around the circle of radius r, as well as of course the cycle generated by the circle of radius R. We also have a 2D cavity created at this point, which is why we see a bar of H_2 being born: it represents the cavity which is left inside of the torus after the balls have merged on the outer part of the torus, too. Finally, after the balls grow larger than $r/2$, the 2D cavity is completely shut off, which makes one H_1 disappear as well as the H_2 disappear, too.

While here we have given an entire interpretation of what happens in terms of the radius of the growing balls, it is imperative to point out that the scales are off as far as the diagrams

are concerned. This is because we have used the Delaunay filtration to compute the persistent homology of the data set. On the other hand, we know that the persistent homology is *isomorphic* to the persistent homology of the Čech filtration of the set of balls, so this interpretation remains nonetheless completely valid, since there is absolutely no loss of information from passing from one of these filtrations to the other as far as the persistent homology is concerned.

This example illustrates the fact that the persistent homology is very clearly dependent on the embedding we choose for the topological space we are looking at. Indeed, notice that we were seamlessly able to recover the number of turns around the torus the embedding perform and that this information was available by simple inspection of the diagram. Notice also that we have information concerning R and r also provided by the lengths of the bars. This goes to show that barcodes can indeed contain fine information about the geometry of the data set in its ambient space. This is why, intuitively, embedding-dependent geometrical concepts such as the weak feature size should naturally arise in theorems such as the Sweet Range Theorem (2.3.2).

3.1.1 Comparison of Different Filtrations

Contrarily to the isomorphism which exists between the Delaunay and the Čech filtrations as persistence modules, we expect to obtain substantially different results when we look at the two other filtrations we introduced in the previous chapter. Indeed, in those cases, we get only an interleaving between them and the Čech filtration persistence module (at least when we look at them in the log-scale).

We have performed computations for the torus toy model example by taking a sample of the helicoidal curve of 500, 1000 and 2000 points. In the case of Witness filtration, we sampled 100 points in the 500 point case and 300 for the other two. This is so that we could get a sense of how the number of simplices grows for each corresponding filtration. Indeed, it turns out that not all these filtrations are created equal. As stated before, the Delaunay filtration exhibits very good behaviour with respect to the number of simplices and running time in low-dimensional situations such as this one. On the other hand, this behaviour quickly turns sour as we increase the ambient dimension in which the point cloud lives. In turn, the Rips filtration exhibits awry behaviour already at this stage, since it behaves mostly like Čech filtration, with the exception that the geometric predicate to be computed gets greatly simplified as we scale in dimensionality. In fact, the Rips filtration behaves already so badly in our case that we had to cap off the dimensionality of the simplicial complex to be calculated to 2 for the 1000 points and 2000 points cases, since the number of simplices was way too high and the computation too slow to be performed within a reasonable time frame on a standard PC. The utility of the Rips filtration comes when it is coupled to witness filtration, or by means of algebraic tricks coupled with so-called iterative subsampling, a method introduced by Chazal *et al.* for the purpose of performing higher dimensional calculations [2]. Whilst we did not introduce the latter, these methods behave considerably better than the Čech and Rips filtrations, in that they scale up as $2^{O(m^2)}n$, where m is the doubling dimension of the point cloud and n is the number of points. The theoretical and experimental performances of each filtration are shown in tables 3.1 and 3.2 respectively. We immediately notice that the Delaunay filtration greatly outperforms the other filtrations by a longshot. On the other hand, as we can see in table 3.1, this performance becomes exponentially worse as we increase the ambient dimension d of the point cloud. This is why when dealing with low-dimensional situations one should always use Delaunay first, as it is the most reliable filtration, contains no noise due to a potential interleaving and is easily computable.

Filtration	Scale	Noise	Memory usage	Computation cost
Čech	linear	$O(\varepsilon)$	$O(2^n)$	$O(2^n)$
Delaunay	linear	$O(\varepsilon)$	$O(n^{\lceil d/2 \rceil})$	$O(n^{\lceil d/2 \rceil})$
Rips	log	2	$O(2^n)$	$O(2^n)$
Rips Witness	log	6	$O(2^w)$	$O(2^w)$

Table 3.1: Theoretical comparison [2] in performance between the different filtrations we introduced. d is the ambient dimension, n is the number of points in the point cloud w is the cardinality of the subsample of the point cloud chosen for witness filtration and ε is the Hausdorff distance between the point cloud and the compact set K which models P (*cf.* the Sweet Range Theorem, 2.3.2).

Number of points	Delaunay filtration		Rips filtration		Witness filtration	
	Dimension	# simplices	Dimension	# simplices	Dimension	# simplices
$n = 500$ $w = 100$	3	42 328	3	36 095 804	3	487 303
$n = 1000$ $w = 300$	3	165 040	2	11 686 085	3	32 025 345
$n = 2000$ $w = 300$	3	654 092	2	93 460 271	3	29 217 681

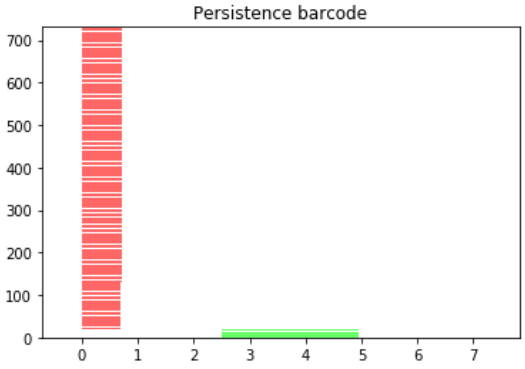
Table 3.2: Details of the memory cost of the computations performed for the different filtrations.

The results of the calculations for the 2000 points data set are given in figure 3.3. We refer the reader to the figures A.1 and A.2 of the appendix to see the results of the calculations for 500 points and 1000 points respectively, as well as for the 100 point Witness filtration.

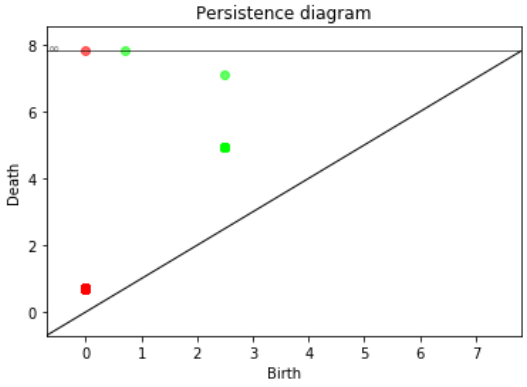
A couple of remarks are in order about these results. We have already fully explored the details of why the Delaunay filtration looks the way it does at every scale. By proposition 2.4.1, we know that there exists an interleaving between the persistence modules of the Čech and the Rips filtrations and we also know that the Delaunay and Čech filtrations are isomorphic as persistence modules. This means that the explanations given for the allure of the persistence diagrams of the Delaunay filtration easily carry over to the Rips case, up to some supplementary noise introduced by the fact that we are dealing with an interleaving (once again recalling that `gudhi` plots the square of the scaling parameter for the Delaunay filtration).

Indeed, the persistence diagrams 3.2b and 3.3b show that, excluding certain points close to the diagonal, there seems to be a matching between points of the diagrams (at each fixed degree in homology). We can also easily conceive that there is a multiplicative factor by which the points seem to diverge from one diagram to another. This confirms exactly what we expected from the existence of an interleaving between the two modules in the log-scale.

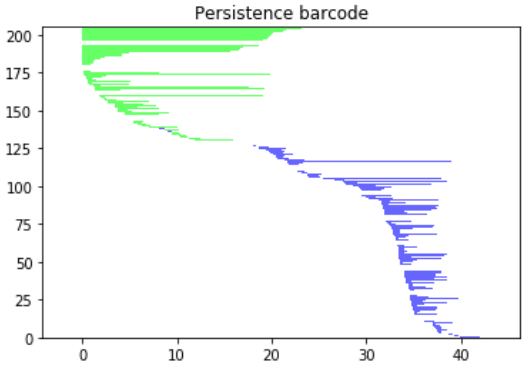
This logic should carry word-for-word to the witness filtration, since by theorem 2.4.1, we also have an interleaving between the Čech and Witness filtration's persistence modules in the log-scale. However, we note that the constant of the interleaving is larger, which has as a consequence the introduction of more noise. Additionally, the bounds obtained in the Sweet Range theorem for the Witness complex (theorem 2.4.2), while in any case not optimal, actually imply that in order to see the same Sweet Ranges in the Delaunay and Witness filtrations, we need to have a quite dense sampling of landmarks amongst the point cloud. This is seemingly not the case for the number of points we have taken as landmarks. Notice that we are also sensible in this analysis to the fact that the bounds of the Sweet Range theorem for the Witness



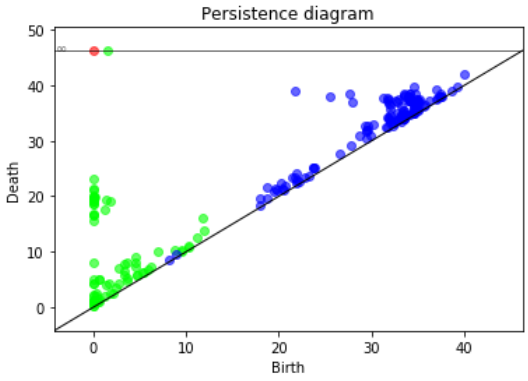
(a) Barcode, Rips filtration.



(b) Persistence diagram, Rips filtration.



(c) Barcode, Witness filtration.



(d) Persistence diagram, Witness filtration.

Figure 3.3: Persistence diagrams and barcodes of the torus data set of 2000 points (and 300 points in the case of witness filtration). In red we can read off the H_0, in green the H_1 and in blue the H_2 components of the persistent homology as calculated over $\mathbb{Z}/2\mathbb{Z}$. We have excluded the intervals of length less than 0.3 in order for the diagram not to be cluttered with entries close to the diagonal (*i.e.* tiny bars in the barcode).

complex (theorem 2.4.2) depend explicitly on the weak feature size of the compact set K. We can see this because while it is difficult to see the smaller features of the toy model, the circle is perfectly clearly resolved by the Witness complex, as its weak feature size is considerably larger than the one of other compact sets we could model our data on, such as the torus or the helicoidal path. It is noteworthy that there are techniques which exist in order to improve the results of witness filtration based on multi-scale reconstructions [2], but the discussion of these methods is beyond the scope of this manuscript. In essence however, we can see that while it might appear at first that

3.1.2 Introduction of Noise

In order to illustrate the robustness with respect to statistical error guaranteed provided by corollary 1 of theorem 2.3.2, we introduce a gaussian isotropic (in the sense that the distribution has mean zero and that the covariance matrix of the multivariate normal distribution is the identity, scaled by some factor σ) error for each data point and calculate the persistence diagram in the same way we did previously. A depiction of this data set is given in figure 3.4. Since the torus and the helicoidal path itself have positive weak feature size, we expect that for σ

small enough, we have guaranteed sweet ranges off of which we can read the homology of the torus and the helicoidal path itself. The results of these computations can be found in figure 3.5. It is no surprise that for small σ (in figure 3.5, $\sigma = 0.2$), we recover a qualitatively

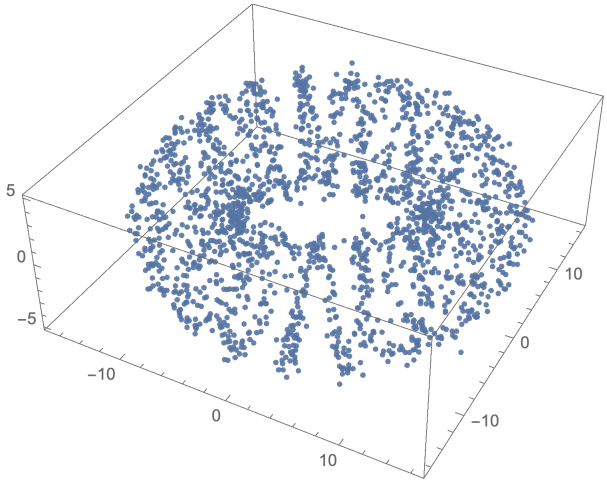
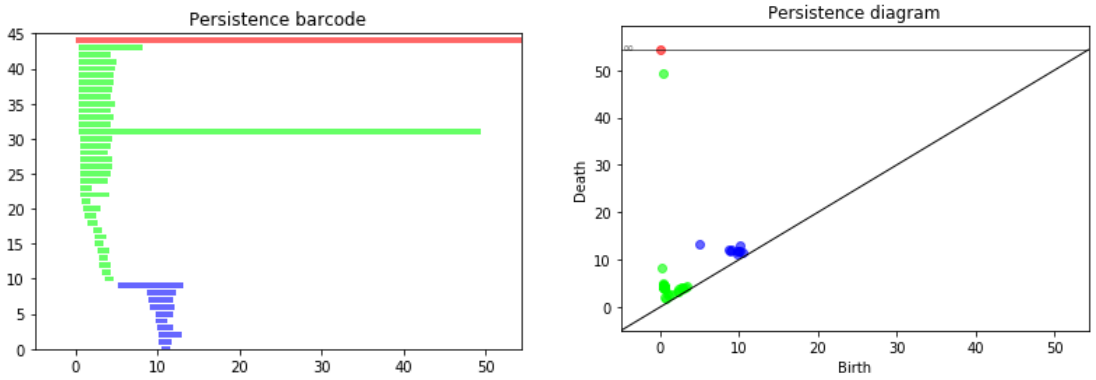


Figure 3.4: The same data set as before, with an introduction of Gaussian statistical error of $\mu = 0$ and covariance matrix $\Sigma = \sigma\,\mathrm{id}$ on each point of the point cloud



(a) Barcode of the data with noise.

(b) Persistence diagram of the data with noise.

Figure 3.5: Persistence diagram and barcode of the torus data set. In red we can read off the H_0, in green the H_1 and in blue the H_2 components of the persistent homology calculated in $\mathbb{Z}/2\mathbb{Z}$. We have excluded the intervals of length less than 0.5. The parameter σ was set to a value of 0.2.

similar result as before, *i.e.* we clearly see the sweet ranges prescribed by the Sweet Range Theorem (2.3.2) in the barcode and the persistence diagram. They are given by the longest bars in the barcode. Just as before and from figure 3.4, it is possible to appreciate that the picture does not change too much from a qualitative point of view despite of the topological noise introduced. Thus, we still see the corresponding 20 bars appear (although sooner) for the H_1. The intervals corresponding to this noise seem to be bound in length was also to be expected from the Sweet Range Theorem (2.3.2).

As we make σ larger, eventually it will be so large that the conditions that guarantee the existence of the sweet range no longer hold. Experimentally, we have confirmed this as seen in the following results of the computation for $\sigma = 0.8$. In fact, we have chosen a value of the

variance close to $\frac{1}{4}\text{wfs}(K)$, where K is the helicoidal path on the torus, *cf.* figure 3.7. Note

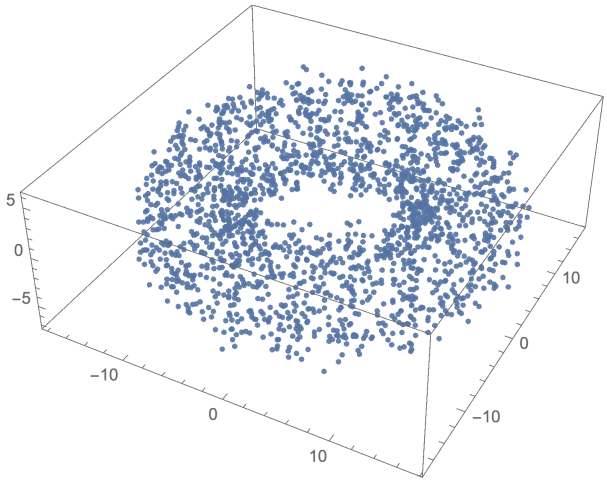
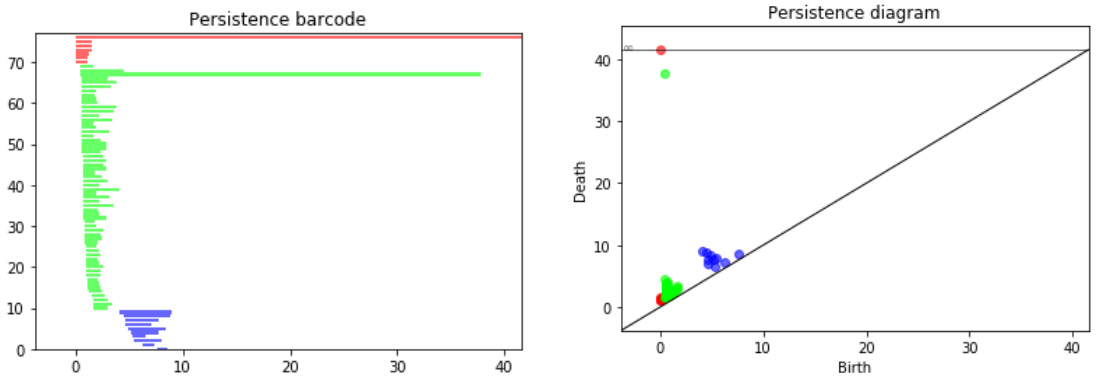


Figure 3.6: The same data set as before, with an introduction of Gaussian statistical error of $\mu = 0$ and $\sigma = 0.8$ on each point of the point cloud



(a) Barcode with $\sigma = 0.8$.

(b) Persistence diagram with $\sigma = 0.8$.

Figure 3.7: Persistence diagram and barcode of the torus data set. In red we can read off the H_0, in green the H_1 and in blue the H_2 components of the persistent homology calculated in $\mathbb{Z}/2\mathbb{Z}$. We have excluded the intervals of length less than 0.5. The parameter σ was set to a value of 0.8.

that in figure 3.7, it is no longer possible to distinguish the intervals of the H_1 we highlighted previously, since the theorem no longer guarantees our ability to perceive a sweet range. It also becomes inherently difficult to read off the actual homology of the torus from this picture, despite clearly being able to read the homology of the circle with radius R, which is normal since the weak feature size of this circle is still much larger than 4σ.

3.2 Torsion

Despite the power of the theory of persistence modules in its general form as presented in chapter 1, this general theory has one big flaw, at least when applied to persistent homology. Indeed, in usual computations of singular homology, we take the homology singular set to take

values in $\mathbf{Mod}_{\mathbb{Z}}$. The singular homology of a topological space when studied over the ring $\mathbb{Z}$ has information about the so-called torsion of the topological space. Unfortunately, most of the useful results and indeed most of the power of the theory of persistence modules stems from Gabriel's theorem on the decomposition of persistence modules, or extensions thereof provided by Chazal *et al.*'s work on q-tame modules [1, §3.9]. An important and upsetting fact is that Gabriel's theorem is simply not true for modules over a commutative ring which is not a field. However, we still would like to recover some kind of information about the torsion of the compact set K which our data set P is said to lie close to. It turns out that it is at least possible to recover part of this lost information. By computing the persistent homology of data sets in different fields, we can detect the presence of torsion in the degrees for which we have calculated. This suggests that a calculation over all rings $\mathbb{Z}/p^k\mathbb{Z}$ for all primes p and $k \in \mathbb{Z}$ would yield similar information to the one contained in singular homology computed over $\mathbb{Z}$.

There is a myriad of questions we could ask about torsion. Amongst others, one of the most important is whether we can actually detect it by performing calculations over finite fields. In this section, we will try to address this issue from an experimental point of view. First, we will do a case study of the Klein bottle and look at a data set randomly sparsed through it. We will then introduce some Gaussian noise into the mix in order to see how this affects our ability to perceive torsion, while keeping in mind the results of the Sweet Range Theorem (2.3.2).

3.2.1 Case Study of the Klein Bottle

As previously discussed, we will take the particular case of the Klein bottle to study how exactly the notion of torsion can be observed in practice. For this, we will need to give ourselves an embedding of the Klein bottle in $\mathbb{R}^4$. We then proceed to give a sufficiently sparse set on the bottle. We will also need to compute the homology of the Klein bottle in different fields, so that we know what and over which fields we should look for to see differences at the level of the persistent homology.

Embedding in $\mathbb{R}^4$

As we saw in the example of the torus, the embedding we choose matters to the results we obtain for the persistent homology. We here give the explicit embedding of the Klein bottle we will use for the computations that will follow:

$$x = (R + r\cos\theta)\cos\varphi \tag{3.2.4}$$
$$y = (R + r\cos\theta)\sin\varphi \tag{3.2.5}$$
$$z = r\sin\theta\cos(\varphi/2) \tag{3.2.6}$$
$$t = r\sin\theta\sin(\varphi/2) \tag{3.2.7}$$

We set the parameters $R = 4$ and $r = 1$ for what will follow. In order to give ourselves a sufficiently sparse set, we take the following $\theta = \varphi/\pi$ and we let φ range over the integers from 0 to 2000. This gives us a set of 2000 points which is depicted in figure 3.8 on the (θ, φ)-plane.

Homology Calculation

To start, we consider the Klein bottle. It can be illuminating, albeit easy, to compute explicitly the homology of the Klein bottle. To do this, we give an explicit CW-complex of the Klein bottle. The chain complex associated to the CW-complex in this case is:

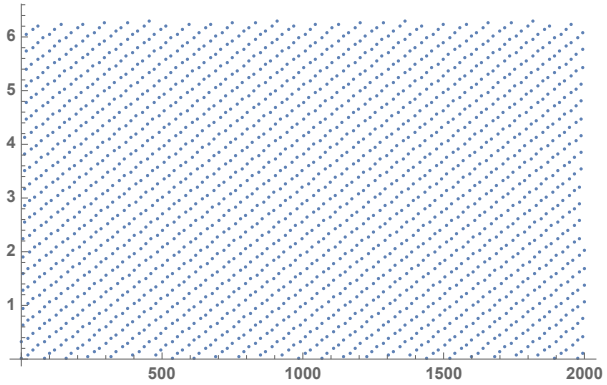


Figure 3.8: Sparse set of 2000 points prescribed by $\theta = \varphi/\pi$ in the (θ, φ)-plane.

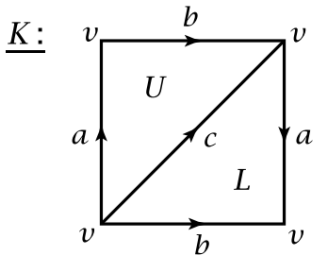


Figure 3.9: CW-complex of the Klein bottle

$$0 \longrightarrow \mathbb{Z}U \oplus \mathbb{Z}L \xrightarrow{\partial_2} \mathbb{Z}a \oplus \mathbb{Z}b \oplus \mathbb{Z}c \xrightarrow{\partial_1} \mathbb{Z}v \xrightarrow{\partial_0} 0 \tag{3.2.8}$$

We start by noticing that the map $\partial_1 = 0$, since all elements of C_1 begin and end at a common vertex. Furthermore, we can express the map ∂_2 in the form of a matrix as follows:

$$\partial_2 = \begin{pmatrix} 1 & -1 \\ -1 & -1 \\ 1 & 1 \end{pmatrix} \sim \begin{pmatrix} 1 & 2 \\ -1 & 0 \\ 1 & 0 \end{pmatrix} \tag{3.2.9}$$

where in the last step we have simply column-reduced ∂_2. So that we have:

$$H_1(K, \mathbb{Z}) = \frac{\ker(\partial_1)}{\mathrm{Im}(\partial_2)} = \frac{\mathbb{Z}a \oplus \mathbb{Z}c \oplus \cancel{\mathbb{Z}(a - b + c)}}{2\mathbb{Z}a \oplus \cancel{\mathbb{Z}(a - b + c)}} \cong \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z} \tag{3.2.10}$$

All that is left to do is to find $\ker \partial_2$. By row-reducing ∂_2 we obtain:

$$\partial_2 \sim \begin{pmatrix} 1 & -1 \\ 0 & 2 \\ 0 & 0 \end{pmatrix} \tag{3.2.11}$$

Notice that this implies that $\ker \partial_2 = 0$ over $\mathbb{Z}$, so that :

$$H_0(K, \mathbb{Z}) = \mathbb{Z}\ ,\ H_1(K, \mathbb{Z}) = \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z} \ \text{ and } \ H_2(K, \mathbb{Z}) = 0. \tag{3.2.12}$$

It is easy to see that this is heavily field dependent, indeed, over $\mathbb{Z}/2\mathbb{Z}$, the column- and row-reduction of ∂_2 are drastically different. Over this finite field we get that:

$$H_1(K, \mathbb{Z}/2\mathbb{Z}) = (\mathbb{Z}/2\mathbb{Z})^2 \ \text{ and } \ H_2(K, \mathbb{Z}/2\mathbb{Z}) = \mathbb{Z}/2\mathbb{Z} \tag{3.2.13}$$

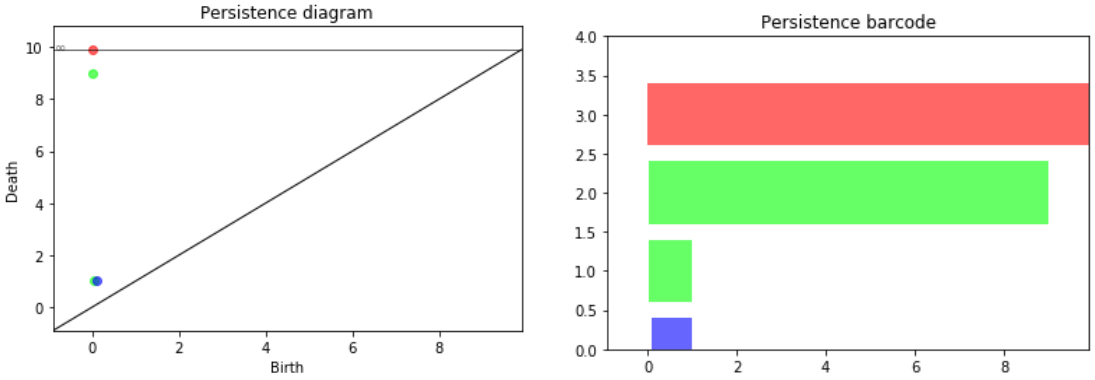
Finally, for any prime $p \neq 2$ we have that:

$$H_1(K, \mathbb{Z}/p\mathbb{Z}) = \mathbb{Z}/p\mathbb{Z} \text{ and } H_2(K, \mathbb{Z}/p\mathbb{Z}) = 0 \tag{3.2.14}$$

We conclude that performing the calculation over $\mathbb{Z}/2\mathbb{Z}$ and $\mathbb{Z}/3\mathbb{Z}$ suffices in order to see the desired results.

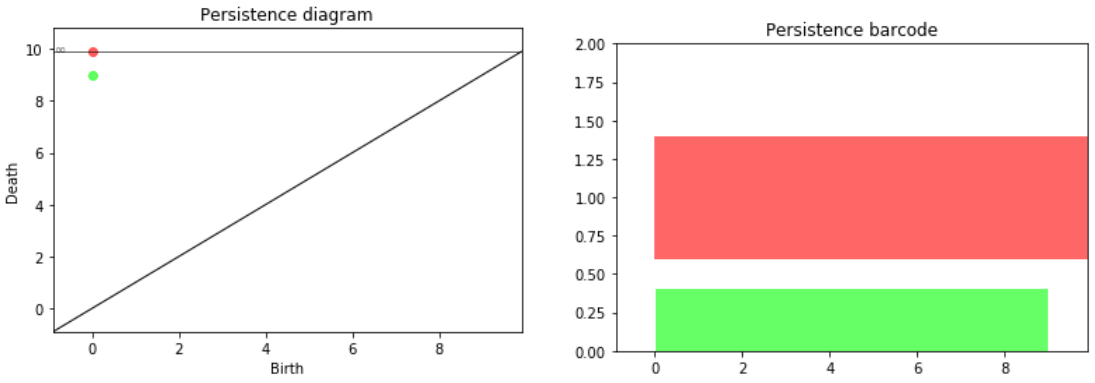
Experimental Results

Given the data set previously described in $\mathbb{R}^4$ along the Klein bottle, we computed the persistence diagram once again using the Delaunay filtration, which is still manageable in terms of running time for this particular data set in dimension 4. The results are depicted in figures 3.10 and 3.11.



(a) Persistence diagram over $\mathbb{Z}/2\mathbb{Z}$ (b) Barcode over $\mathbb{Z}/2\mathbb{Z}$

Figure 3.10: Persistence diagrams and barcodes of the data set on the Klein bottle over $\mathbb{Z}/2\mathbb{Z}$. We have excluded bars of length less than 0.1 in order to avoid clutter in the diagrams.



(a) Persistence diagram over $\mathbb{Z}/3\mathbb{Z}$ (b) Barcode over $\mathbb{Z}/3\mathbb{Z}$

Figure 3.11: Persistence diagrams and barcodes of the data set on the Klein bottle over $\mathbb{Z}/3\mathbb{Z}$. We have excluded bars of length less than 0.1 in order to avoid clutter in the diagrams.

As expected, we retrieve the elements guaranteed by the Sweet Range Theorem 2.3.2. Of course, due to the field of computation, we also retrieve the results we had previously obtained for the homology of the Klein bottle over $\mathbb{Z}/2\mathbb{Z}$ and $\mathbb{Z}/3\mathbb{Z}$ respectively. Since this is only

dependent on the clauses of the Sweet Range Theorem, we should expect that for a reasonable amount of noise, *i.e.* noise such that the Hausdorff distance between the point cloud and the Klein bottle K is less than $\frac{1}{4}\text{wfs}(K)$. We should see similar results for the perturbed data set under these conditions and indeed we do.

Remark 3.2.1. We also tried to do this calculation with $\theta = 20\varphi$. In that case, we expect to get a path through the Klein bottle. In fact, we observe a similar phenomena in this case where we have enough loops so that at a large scale, the topology of the Klein bottle emerges from the data set. Since it is a large scale effect, we are still sensible to the sweet range results for the Klein bottle itself, since we are at an interval in which the scale is still inferior to $\frac{1}{4}\text{wfs}(K)$, yet is large enough so as not to be confounded with noise (recall we have a bound of 2ε on the length of the noise intervals according to the Sweet Range Theorem, 2.3.2).

This last remark leads us to notice that we must be careful about the inferences we make when dealing with higher dimensional data sets. Indeed, as we saw, a closed curve such as $\theta = 20\varphi$ on the Klein bottle can lead to torsion-dependent homology. Since calculations are routinely only ever done in $\mathbb{Z}/2\mathbb{Z}$, we must take these kind of results with care for higher dimensional data, keeping in mind this kind of example. This effect might be compounded by a couple of other factors.

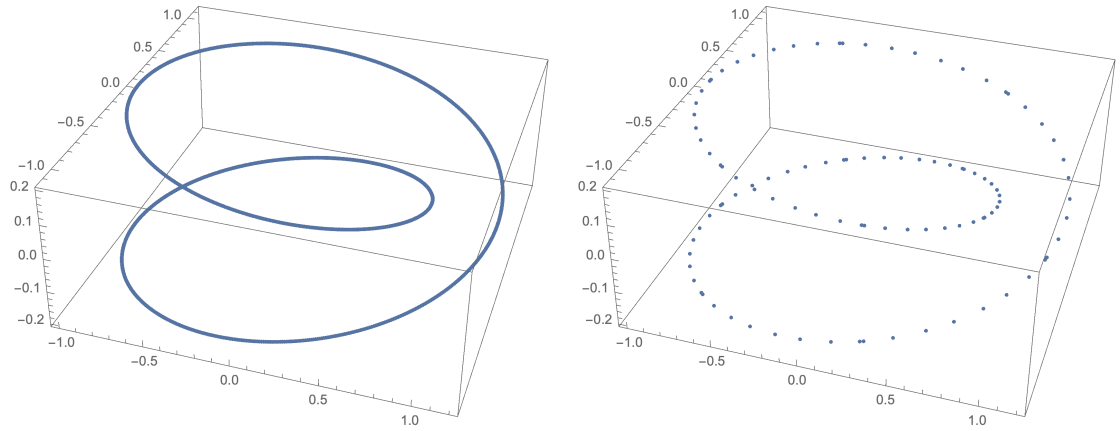
- For higher-dimensional data, we typically only bother computing H_0 and H_1. This is due to the explosion in the memory required to store all the simplices in higher dimensions. This means that we usually won't have all the information on the full homology even over $\mathbb{Z}/2\mathbb{Z}$ as in this example.

- In this example, we used Delaunay filtration, which provides perhaps one of the most reliable calculations for the persistent homology of a point cloud. This is because of the isomorphism that exists between the Delaunay filtration and the Čech filtration of the set balls. However, in higher dimensions, it is computationally unfeasible to apply a Delaunay filtration. To deal with this, we consider Witness and/or Rips-Vietoris filtrations, which are only interleaved with the Čech filtration. This might obscure whether there are effects of torsion even further.

3.2.2 Specific Geometric Configurations

Under certain geometric configurations of the point cloud, it is actually also possible to see field-dependent differences in the persistent diagram of the point cloud. The reason why this occurs has to do with the way some homology classes are imaged by the linear transformations provided by the persistent homology functor, which might sometimes involve a multiplicative factor which may be null over certain finite fields. To illustrate this, we have the following example: consider the edge of a Möbius strip, depicted in figure 3.12a and take a point cloud sparsed densely enough on this curve (*cf.* figure 3.12b), it is clear that for a relatively small radius, the complex of balls will have the same homology type as the curve of figure 3.12a. The reason for this is graphically clear, but is also inherently linked to the Sweet Range Theorem (2.3.2) we have thus far evoked multiple times.

At a larger value for the radius of the balls in the complex, the balls of the inner ring will intersect the balls of the larger ring. This will result in a simplicial complex which will basically have the same homology type as a circle. On the other hand, the homology class corresponding to the single generator of the H_1 of figure 3.13 will be sent to *twice* the generator of the H_1 of the circle. As a result, the computation of the persistent homology of this point cloud over $\mathbb{Z}/2\mathbb{Z}$ should yield a different result than the one computed in another finite field. The



(a) The edge of the Möbius strip.

(b) The point cloud considered.

Figure 3.12: The point cloud considered consists of a set of 100 points distributed as above and whose embedding is given by equations 3.2.15.

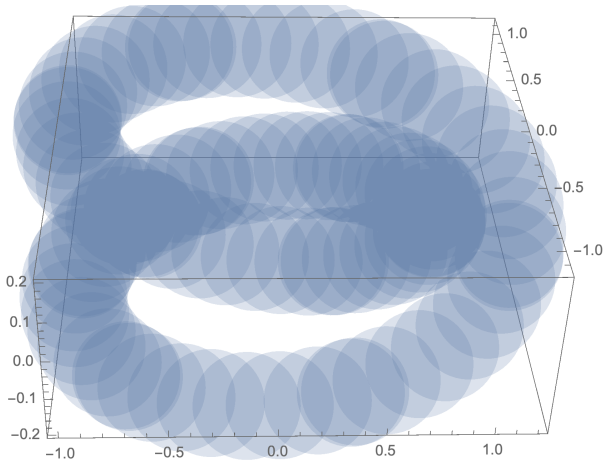


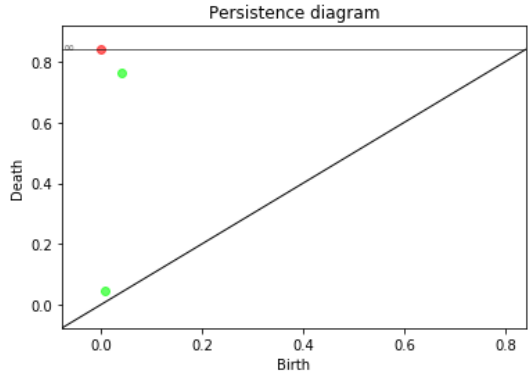
Figure 3.13: As the balls around each point grow larger, we have a merger of the inner and outer ring of the curve when the radius of the balls is approximately 0.2.

parametrization we have used for the edge of the Möbius strip is as follows:

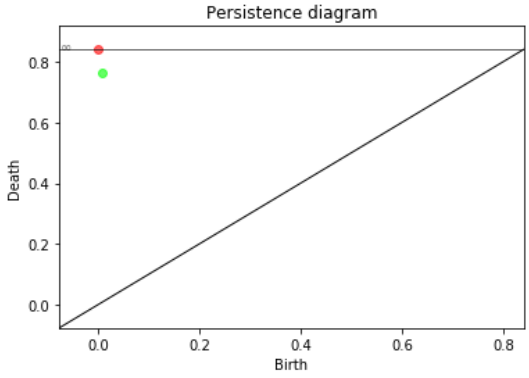
$$\begin{aligned} x &= \left(1 \pm \frac{1}{5}\cos\frac{u}{2}\right)\cos u \\ y &= \left(1 \pm \frac{1}{5}\cos\frac{u}{2}\right)\sin u \\ z &= \pm\frac{1}{5}\sin\frac{u}{2} \end{aligned} \tag{3.2.15}$$

The experimental results to confirm this observations were obtained by using a Delaunay filtration, whose simplicial complex was of dimension 3 and carried 9691 simplices over the 100 vertices of the point cloud. The reason for this choice of filtration is due purely to computational considerations, indeed as we have previously shown in section 3.1, the low-dimensionality of the problem render the Delaunay filtration optimal and avoid introducing noise into the diagrams.

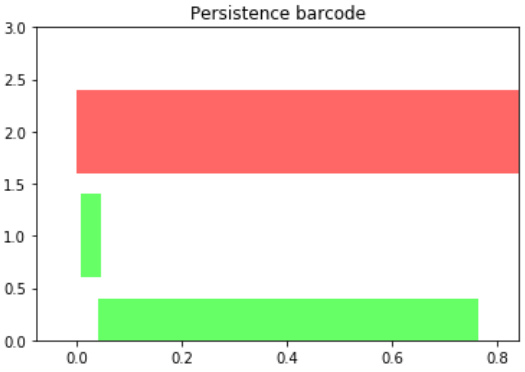
The persistence diagrams and barcodes of the point cloud performed over $\mathbb{Z}/2\mathbb{Z}$ and $\mathbb{Z}/3\mathbb{Z}$



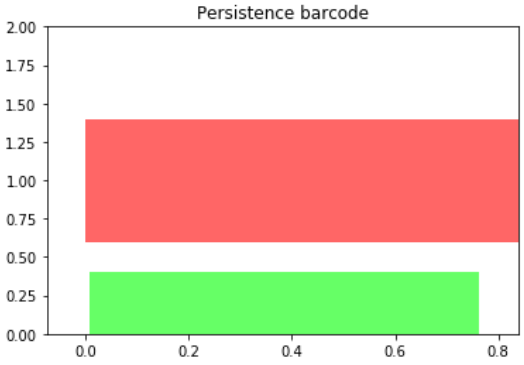
(a) Persistence diagram over $\mathbb{Z}/2\mathbb{Z}$.



(b) Persistence diagram over $\mathbb{Z}/3\mathbb{Z}$.



(c) Barcode over $\mathbb{Z}/2\mathbb{Z}$.



(d) Barcode over $\mathbb{Z}/3\mathbb{Z}$.

Figure 3.14: Results of the persistence calculation for the Möbius edge point cloud over $\mathbb{Z}/2\mathbb{Z}$ and $\mathbb{Z}/3\mathbb{Z}$.

are depicted in figure 3.14. Recalling that in the persistence diagrams stemming from a Delaunay filtration`gudhi` always plots the square of the scale parameter of the filtration, we see clearly that the phenomenon discussed above is portrayed in the results we see. Indeed, the scale at which we see the first bar of the H_1 disappear corresponds exactly to the scale of roughly $r^2 \approx 0.04$, which corresponds to a radius $r \approx 0.2$, which is depicted in figure 3.13, which corresponds to the scale at which the inner and outer ring of the curve merge. By contrast, the $\mathbb{Z}/3\mathbb{Z}$ calculation does not exhibit this behaviour, as $2 \neq 0$ over $\mathbb{Z}/3\mathbb{Z}$.

Asides from this effect due to the geometry of the cloud, we notice that the Sweet Range Theorem still applies in full force, as we are still guaranteed to obtain some sweet ranges over which we may read the homology of the cloud over $\mathbb{Z}/2\mathbb{Z}$ and $\mathbb{Z}/3\mathbb{Z}$ respectively, which corresponds to the homology of a circle.

This example illustrates the fact that the specific parametrization and the geometrical layout of a point cloud also might affect the results obtained for the persistent homology over a single field and is a warning against any hasty geometric interpretations we might want to give to persistent homology results based only on a single field.

Conclusion

Throughout this thesis, we have explored the theory of persistence modules and of persistent homology under the umbrella of topological data analysis. We adopted a first principles approach in our treatment of the general theory of persistence modules, which gave us the appropriate tools to understand in detail some results specific to topological data analysis and persistent homology. In particular, we recall two such main results of this manuscript : the Isometry Theorem (theorem 1.4.1), which equates the interleaving distance to the bottleneck distance and the Sweet Range Theorem (theorem 2.3.2), which guarantees the existence of an interval in which we may read off the homology of a compact set K which is ε-Hausdorff close to a given point cloud P, provided that K satisfies certain geometrical conditions.

In chapter 3, we experimentally confirmed the theoretical statements made in chapter 2 about the persistent homology of point clouds. In particular, we explored the results and computational performance of the different filtrations we introduced in this chapter with the help of the torus toy model (*cf.* section 3.1.1) and explored how Gaussian noise perturbation could affect the persistence diagrams and the potential topological inferences we could make about the point cloud. We also explored the effects of torsion, first by giving an examples in which we clearly expected to see torsion related effects (the Klein bottle), which we incidentally also later perturbed with Gaussian noise (*cf.* section 3.2.1). However, with the example of the Möbius strip edge, we showed that the effects of torsion were not only limited to cases where the data is sparsed close to a manifold exhibiting torsion, but that the embedding of the point cloud in its ambient space could also yield field-dependent results at the persistent homology level (*cf.* section 3.2.2). These examples all illustrate how the theoretical statements given in this manuscript are used and retrieved in practical cases.

Still on the experimental front, we considered the case where we have a point cloud on the Klein bottle, a manifold which exhibits torsion. This allowed us to reiterate what we already knew, but also gave us some hints at the elusiveness of torsion in practical applications, especially in high dimensional data sets. For the interested reader, other questions and some answers on the computation of torsion can be found in [15, 16].

Finally, persistence theory has many other applications outside of topological data analysis which may yield interesting results in the future, for instance, it has already given interesting results in geometry [14, 17]. As we have shown multiple times throughout this manuscript, persistence theory is, loosely speaking, a theory of scale, a concept which enters quite naturally in geometry and other fields of mathematics and physics, it is thus not unreasonable to hope that other such significant results are still left to find.

Appendix A

Code for Calculations

We give here the lines of code necessary to compute the examples given in chapter 3. We have also included an example which was not featured in the main body of this manuscript: the computation of the perstent homology of a random distrubution on the Möbius strip. It is important to note that the installation of the python package `gudhi` is necessary for the code to run. We refer the reader to the extensive documentation that can be found online for the `gudhi` package [18]. As a sidenote, we recommend that the reader use some python installer such as `conda` or `pip` in order to install `gudhi` as smoothly as possible.

Note that the codes all look alike, but what varies is mostly the embedding we choose. Also, it is possible to change the type of filtration considered in the calculation. For more information on how to implement other filtrations other than the Delaunay one, we invite once again the reader to consult the `gudhi` documentation.

A.1 Torus Toy Model

We have already given explicitly the embedding of our toy model back in section 3.1. Here, we note that the code also includes the possibility of adding noise to the sample of the toy model. Of course, by setting the parameter $s = 0$, we include the case of no noise in the same piece of code.

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


pi = np.pi
torus_sampling= []
projtorus2d= []
"""Embedding parameters, R is big radius, r is small radius, p is number of
    turns and s is a factor of noise"""
R=12
r=4
p=20
s=0.
mean = [0,0,0]
cov = [[s,0,0],[0,s,0],[0,0,s]]
"""Number of witnesses:"""
w=300

for n in range(0,2000):
```

```
    pt=[(R+r*np.cos(p*n))*np.cos(n),(R+r*np.cos(p*n))*np.sin(n),r*np.sin(p*n)]
    veps= np.random.multivariate_normal(mean,cov)
    for i in range(0,3):
        pt[i]+= veps[i]
    torus_sampling.append(pt)
    projtorus2d.append([pt[0],pt[1]])

"""Define landmarks and w witnesses"""
landmarks = gudhi.pick_n_random_points(points=torus_sampling,nb_points=w)

plt.scatter(*zip(*projtorus2d))
plt.show()


"""Generate different complexes associated to the sampling"""
print("#######################################################################")
print("Delaunay Complex creation from points")
alpha_complex= gudhi.AlphaComplex(points=torus_sampling)
print("#######################################################################")
print("Rips Complex creation from points")
rips_complex= gudhi.RipsComplex(points=torus_sampling,max_edge_length=12.0)
print("#######################################################################")
print("Witness Complex creation from points")
witness_complex= gudhi.EuclideanWitnessComplex(witnesses=torus_sampling,
    landmarks=landmarks)

"""Generate the associated simplex tree structure to each of the complexes"""
simplex_tree_alpha = alpha_complex.create_simplex_tree(max_alpha_square=60.0)
simplex_tree_rips = rips_complex.create_simplex_tree(max_dimension=2)
simplex_tree_witness = witness_complex.create_simplex_tree(max_alpha_square
    =100.0,limit_dimension=3)


"""Print out information about the complexes"""
result_str_alpha = 'Delaunay complex is of dimension ' + repr(
    simplex_tree_alpha.dimension()) + ' - ' + \
    repr(simplex_tree_alpha.num_simplices()) + ' simplices - ' + \
    repr(simplex_tree_alpha.num_vertices()) + ' vertices.'
result_str_rips = 'Rips complex is of dimension ' + repr(simplex_tree_rips.
    dimension()) + ' - ' + \
    repr(simplex_tree_rips.num_simplices()) + ' simplices - ' + \
    repr(simplex_tree_rips.num_vertices()) + ' vertices.'
result_str_witness = 'Witness complex is of dimension ' + repr(
    simplex_tree_witness.dimension()) + ' - ' + \
    repr(simplex_tree_witness.num_simplices()) + ' simplices - ' + \
    repr(simplex_tree_witness.num_vertices()) + ' vertices.'

print(result_str_alpha)
print(result_str_rips)
print(result_str_witness)


print("#######################################################################")
print("Persistence diagrams of the torus with different filtrations")
"""Delaunay Filtration"""
diag_alpha = simplex_tree_alpha.persistence(homology_coeff_field=2,
    min_persistence=0.3)
pplot_alpha = gudhi.plot_persistence_diagram(diag_alpha)
print("Diagram Delaunay Complex")
pplot_alpha.show()
```

```
74 print("Barcode Delaunay")
75 plt_alpha = gudhi.plot_persistence_barcode(diag_alpha)
76 plt_alpha.show()
77
78 """Rips"""
79 diag_rips = simplex_tree_rips.persistence(homology_coeff_field=2,
       min_persistence=0.3)
80 pplot_rips = gudhi.plot_persistence_diagram(diag_rips)
81 print("Diagram Rips Complex")
82 pplot_rips.show()
83 print("Barcode Rips")
84 plt_rips = gudhi.plot_persistence_barcode(diag_rips)
85 plt_rips.show()
86
87 """Witness"""
88 diag_witness = simplex_tree_witness.persistence(homology_coeff_field=2,
       min_persistence=0.3)
89 pplot_witness = gudhi.plot_persistence_diagram(diag_witness)
90 print("Diagram Witness Complex")
91 pplot_witness.show()
92 print("Barcode Witness")
93 plt_witness = gudhi.plot_persistence_barcode(diag_witness)
94 plt_witness.show()
```

Listing A.1: Python code to calculate the persistent homology of the torus toy model

A.1.1 Results

The persistence diagrams and barcodes for the 500- and 1000-point point clouds alluded to in chapter 3 are illustrated in figures A.1 and A.2 respectively.

A.2 Möbius Strip

For the case of the Möbius strip, we took a random sample on the square $[0, 2\pi] \times [-1, 1]$ which we then proceed to embed in 3-dimensional space on the Möbius strip in the following way:

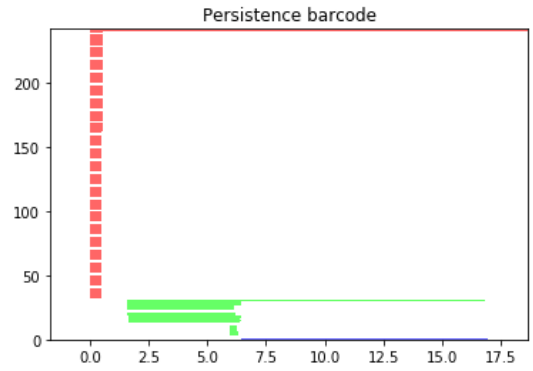
$$x = \left(1 + \frac{v}{2}\cos\frac{u}{2}\right)\cos u \tag{A.2.1}$$

$$y = \left(1 + \frac{v}{2}\cos\frac{u}{2}\right)\sin u \tag{A.2.2}$$

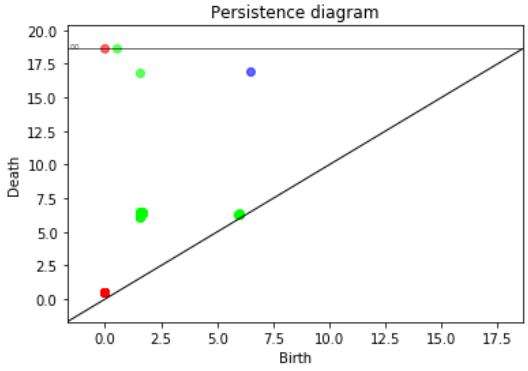
$$z = \frac{v}{2}\sin\frac{u}{2} \tag{A.2.3}$$

Once this choice of embedding is made, we calculate the persistent homology using the code shown below.

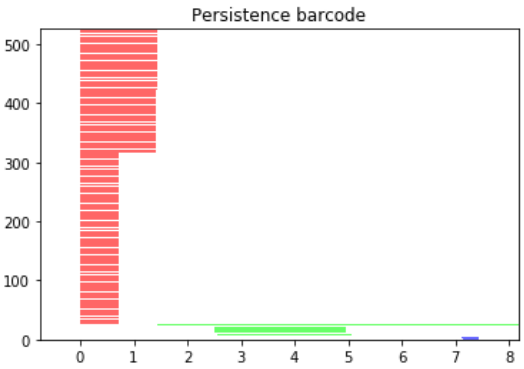
```
1 import gudhi
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 mobius_sampling =[]
6
7 """Generate a random point cloud on the Mobius strip"""
8 for n in range(0,2000):
9   u=random.uniform(0,2*pi)
10    v=random.uniform(-1,1)
```



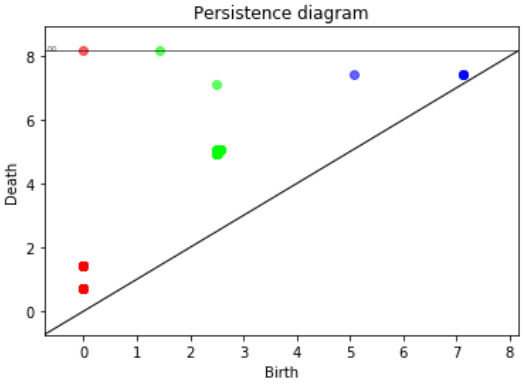
(a) Barcode, Delaunay filtration.



(b) Persistence diagram, Delaunay filtration.



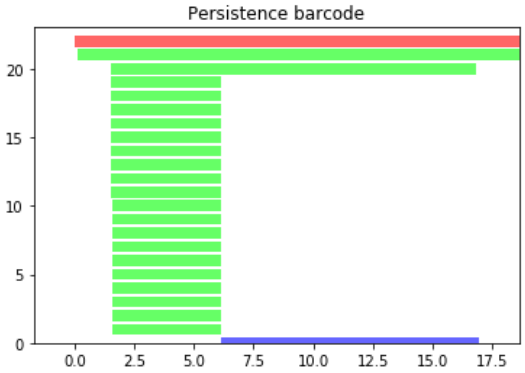
(c) Barcode, Rips filtration.



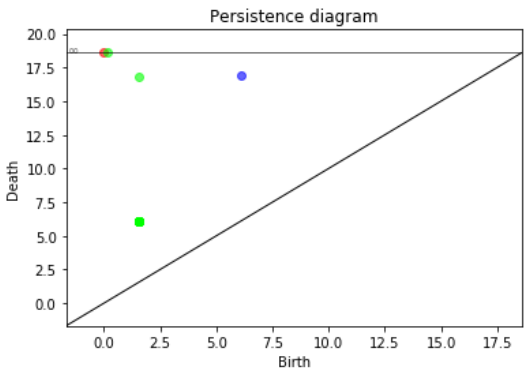
(d) Persistence diagram, Rips filtration.

Figure A.1: Persistence diagrams and barcodes of the torus data set of 500 points. In red we can read off the H_0, in green the H_1 and in blue the H_2 components of the persistent homology as calculated over $\mathbb{Z}/2\mathbb{Z}$. We have excluded the intervals of length less than 0.3 in order for the diagram not to be cluttered with entries close to the diagonal (*i.e.* tiny bars in the barcode).

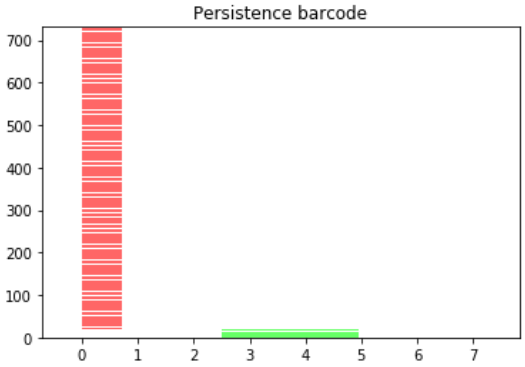
```
    mobius_sampling.append([(1+(v/2)*np.cos(n/2))*np.cos(n),
                            (1+(v/2)*np.cos(n/2))*np.sin(n),
                            (v/2)*np.sin(u)])


"""Construct the simplicial complex associated to the Delaunay filtration of
    the cloud"""
print("#########################################################################")
print("Alpha Complex creation from points")
alpha_complex= gudhi.AlphaComplex(points=mobius_sampling)
simplex_tree = alpha_complex.create_simplex_tree(max_alpha_square=120.0)
result_str = 'Alpha complex is of dimension ' + repr(simplex_tree.dimension())
    + ' - ' + \
    repr(simplex_tree.num_simplices()) + ' simplices - ' + \
    repr(simplex_tree.num_vertices()) + ' vertices.'
print(result_str)


"""Calculate and plot the results as calculated in Z2 and Z3"""
print("#########################################################################")
print("Persistence diagram of the Embedded loops")
diag2 = simplex_tree.persistence(homology_coeff_field=2, min_persistence=0.05)
```



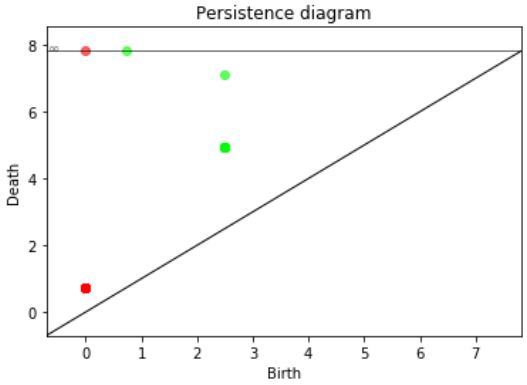
(a) Barcode, Delaunay filtration.



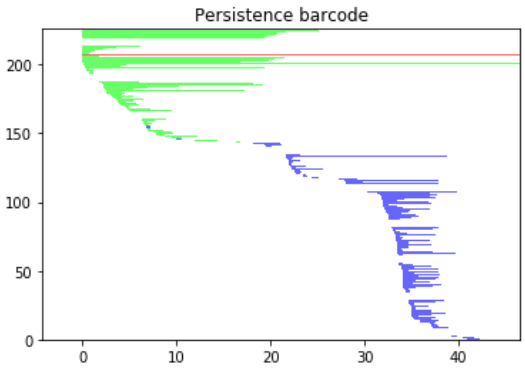
(b) Persistence diagram, Delaunay filtration.



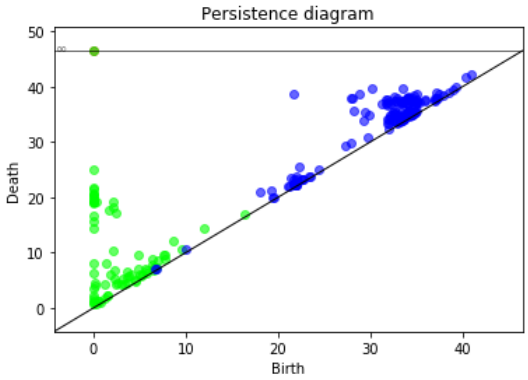
(c) Barcode, Rips filtration.



(d) Persistence diagram, Rips filtration.



(e) Barcode, Witness filtration.



(f) Barcode, Witness filtration.

Figure A.2: Persistence diagrams and barcodes of the torus data set of 1000 points (and 100 points in the case of witness filtration). In red we can read off the H_0, in green the H_1 and in blue the H_2 components of the persistent homology as calculated over $\mathbb{Z}/2\mathbb{Z}$. We have excluded the intervals of length less than 0.3 in order for the diagram not to be cluttered with entries close to the diagonal (*i.e.* tiny bars in the barcode).

```
pplot2 = gudhi.plot_persistence_diagram(diag2)
print("Diagram Z2")
pplot2.show()
print("Barcode Z2")
plt2 = gudhi.plot_persistence_barcode(diag2)
plt2.show()
```

```
diag3 = simplex_tree.persistence(homology_coeff_field=3, min_persistence=0.05)
pplot3 = gudhi.plot_persistence_diagram(diag3)
print("Diagram Z3")
pplot3.show()
print("Barcode Z3")
plt3 = gudhi.plot_persistence_barcode(diag3)
plt3.show()
```

Listing A.2: Python code for the calculation of the persistent homology of a random distribution on the Möbius strip

A.3 Möbius Strip Edge

The code relevant to the experimental results obtained in section 3.2.2 is shown below. Recall that it consistent of points chosen on the curve given by the embedding of the Möbius strip edge whose specific equation is given in equation 3.2.15. The sampling was obtained by simply replacing the parameter u with integer values ranging from 0 to 50.

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


pi = np.pi
mobius_sampling =[]


for n in range(0,50):
    mobius_sampling.append([(1+0.2*np.cos(n/2))*np.cos(n),
                            (1+0.2*np.cos(n/2))*np.sin(n),
                            0.2*np.sin(n/2)])
    mobius_sampling.append([(1-0.2*np.cos(n/2))*np.cos(n),
                            (1-0.2*np.cos(n/2))*np.sin(n),
                            -0.2*np.sin(n/2)])


print("######################################################################")
print("Alpha Complex creation from points")
alpha_complex= gudhi.AlphaComplex(points=mobius_sampling)
simplex_tree = alpha_complex.create_simplex_tree(max_alpha_square=10)
result_str = 'Alpha complex is of dimension ' + repr(simplex_tree.dimension())
    + ' - ' + \
    repr(simplex_tree.num_simplices()) + ' simplices - ' + \
    repr(simplex_tree.num_vertices()) + ' vertices.'
print(result_str)



print("######################################################################")
print("Persistence diagram of the Embedded loops")
diag2 = simplex_tree.persistence(homology_coeff_field=2, min_persistence=0.01)
pplot2 = gudhi.plot_persistence_diagram(diag2)
print("Diagram Z2")
pplot2.show()
print("Barcode Z2")
plt2 = gudhi.plot_persistence_barcode(diag2)
plt2.show()
diag3 = simplex_tree.persistence(homology_coeff_field=3, min_persistence=0.01)
```

```
pplot3 = gudhi.plot_persistence_diagram(diag3)
print("Diagram Z3")
pplot3.show()
print("Barcode Z3")
plt3 = gudhi.plot_persistence_barcode(diag3)
plt3.show()
```

Listing A.3: Python code for the persistent homology calculation of a point cloud on the Klein bottle.

A.4 Klein Bottle

Finally, we fall into the case of the Klein bottle, whose explicit embedding we used was already given back in section 3.2.1. Just as for the example of the torus toy model, we also include the possibility of including noise by tweaking the parameter s in the code. We cover two cases in this code, one of them is a winding along the Klein bottle, and the other one is simply a random sample on the Klein bottle, in the code that we give the case of the random sample is commented out in its corresponding section in the code.

```
import gudhi
import numpy as np
import matplotlib.pyplot as plt
import random


pi = np.pi
klein_sampling =[]
p=40
R=4
r=1
p=10
s=0.1

mean= [0,0,0,0]
cov = [[s,0,0,0],[0,s,0,0],[0,0,s,0],[0,0,0,s]]


"""Add points to the data set, in this case we have the winding and commented
    the possibility of adding a random sample"""
for n in range(0,2000):
    """u=random.uniform(0,2*pi)"""
    """v=random.uniform(0,2*pi)"""
    u=n
    v=6*n
    pt=[(R+r*np.cos(v))*np.cos(u),
                            (R+r*np.cos(v))*np.sin(u),
                            r*np.sin(v)*np.cos(u/2),
                            r*np.sin(v)*np.sin(u/2)]
    veps= np.random.multivariate_normal(mean,cov)
    for i in range(0,4):
        pt[i]+= veps[i]
    klein_sampling.append(pt)


"""Construct the simplicial complex associated to the Delaunay filtration of
    the cloud"""
print("#########################################################################")
```

```
print("Alpha Complex creation from points")
alpha_complex= gudhi.AlphaComplex(points=klein_sampling)
simplex_tree = alpha_complex.create_simplex_tree(max_alpha_square=60.0)
result_str = 'Alpha complex is of dimension ' + repr(simplex_tree.dimension())
    + ' - ' + \
    repr(simplex_tree.num_simplices()) + ' simplices - ' + \
    repr(simplex_tree.num_vertices()) + ' vertices.'
print(result_str)


"""Calculate and plot the results as calculated in Z2 and Z3"""
print("######################################################################")
print("Persistence diagram of Klein Bottle")
diag2 = simplex_tree.persistence(homology_coeff_field=2, min_persistence=0.1)
pplot2 = gudhi.plot_persistence_diagram(diag2)
print("Diagram Z2")
pplot2.show()
print("Barcode Z2")
plt2 = gudhi.plot_persistence_barcode(diag2)
plt2.show()
diag3 = simplex_tree.persistence(homology_coeff_field=3, min_persistence=0.1)
pplot3 = gudhi.plot_persistence_diagram(diag3)
print("Diagram Z3")
pplot3.show()
print("Barcode Z3")
plt3 = gudhi.plot_persistence_barcode(diag3)
plt3.show()
```

Listing A.4: Python code for the persistent homology calculation of a point cloud on the Klein bottle.

Bibliography

[1] F. Chazal, V. de Silva, M. Glisse and S. Oudot, "The Structure and Stability of Persistence Modules," Springer, 2016.

[2] S. Oudot, "Persistence Theory: From Quiver Representations to Data Analysis," American Mathematical Society, 2015.

[3] C. A. Weibel, "An Introduction to Homological Algebra," Cambridge University Press, 1995.

[4] S. Mac Lane, "Categories for the Working Mathematician," Springer, 1971.

[5] D. Attali, H. Edelsbrunner, and Y. Mileyko, "Weak Witnesses for Delaunay Triangulations of Submanifolds," Proceedings of the 2007 ACM Symposium on Solid and Physical Modeling. SPM 2007. Beijing, China: ACM, 2007, pp. 143–150. doi: 10.1145/1236246.1236267.

[6] K. Borsuk, "On the imbedding of systems of compacta in simplicial complexes," Fundamenta Mathematicae 35.1 (1948), pp. 217–234.

[7] F. Chazal and A. Lieutier, 'Stability and Computation of Topological Invari- ants of Solids in $\mathbb{R}^n$," Discrete & Computational Geometry 37.4 (2007), pp. 601–617.

[8] F. Chazal and S. Y. Oudot, "Towards Persistence-Based Reconstruction in Euclidean Spaces," Proceedings of the 24th ACM Symposium on Computational Geometry. 2008, pp. 232–241. doi: 10.1145/1377676.1377719

[9] V. de Silva, "A weak characterisation of the Delaunay triangulation," Geometriae Dedicata 135.1 (2008), pp. 39–64.

[10] V. de Silva and R. Ghrist, "Coverage in Sensor Networks via Persistent Homology," Algebraic and Geometric Topology 7 (2007), pp. 339–358.

[11] H. Edelsbrunner, "The union of balls and its dual shape," Discrete & Computational Geometry 13.1 (1995), pp. 414–440

[12] A. Hatcher. "Algebraic Topology," Cambridge University Press, 2001

[13] A. Lieutier, "Any open bounded subset of Rn has the same homotopy type as its medial axis," Computer-Aided Design 36.11 (2004), pp. 1029–1046

[14] C. Viterbo, F. LePeutrec and F. Nier, "Precise Arrhenius law for p-forms: The Witten Laplacian and Morse-Barannikov complex," Annales Henri Poincaré, vol. 14 (3) 2013, pp. 567–610.

[15] J.-D. Boissonnat, C. Maria, "Computing Persistent Homology With Various Coefficient Fields in a Single Pass," RR-8436, 2013, pp.16. hal-00922572v2f

[16] L. Polanco, J.A. Perea, "Coordinatizing data with lens spaces and persistent cohomology," https://arxiv.org/pdf/1905.00350.pdf

[17] L. Polterovich, D. Rosen, K. Samvelyan, J. Zhang, "Topological Persistence in Geometry and Analysis," https://arxiv.org/abs/1904.04044

[18] Gudhi Library, http://gudhi.gforge.inria.fr/