Thesis

walch

February 2023

CONTENTS

CHAPTER 1

TOPOLOGY FROM DATA

Abstract

The basic TDA pipeline is presented. Starting with its input step, it will be discussed how structure can be introduced into point cloud data. Simplicial complexes form the basis for the homological computations on the sample data, which are continued in Chapter 2. With respect to Chapter 4, Mayer-Vietoris blowup complexes are also presented. The nerve theorem forms the central theoretical guarantee for the construction of certain simplicial complexes (and approximations thereof) on point clouds.

Contents

1.1 The TDA Pipeline

Topological data analysis (TDA) emerged as a synergetic field between applied (algebraic) topology and computational geometry driven by pioneering works of Edelsbrunner [ELZ00], Zomorodian and Carlsson [ZC05; Car09]. The aim of TDA is to provide mathematical, statistical and algorithmic methods in order to qualitatively and quantitatively analyze the structure of data. Following [CM17], the basic TDA pipeline consists of four steps visualized in figure 1.1. The TDA pipeline starts with a data input. Data can come from a variety of sources and appear in different forms: for example as images, sound and picture

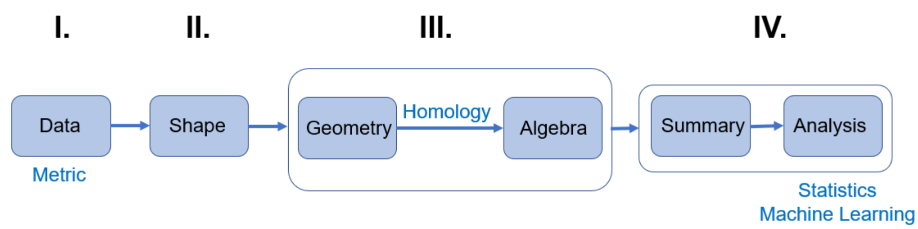


Figure 1.1: TDA Pipeline as described in [CM17].

recordings, time series measurements or text modules. In general, data is given as a set of points, formally a *space*.

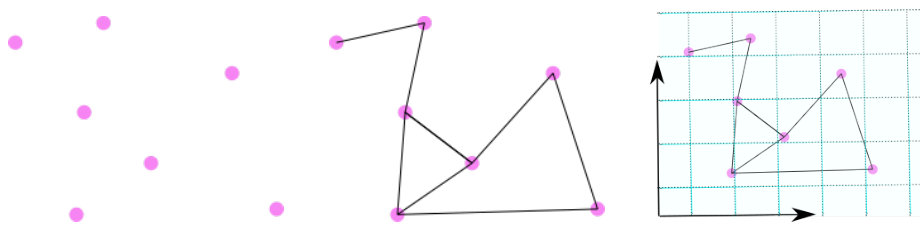


Figure 1.2: From left to right: Space, topological space and metric space.

A space has no structure and thus is not interesting. The addition of topology adds some sort of connectivity to the space meaning that each point in the space is associated to its *neighbouring* points. If this notion of neighborhood is derived from a metric, then we call the space a metric space. With regard to step four and further analysis with statistical and machine learning methods, data sets representing finite metric spaces (M, d_M) are especially interesting. These are called **point clouds**. For many applications (M, d_M) is just a subspace of the n-dimensional Euclidian space $\mathbb{R}^n$ with the metric given by restriction of the Euclidian metric. If this is not the case, it will be denoted explicitly.
In the next step called "shape" in Figure 1.1, $\mathcal{D}$ is partitioned (combinatorically) into subsets and maps between them in order to induce the structure of a non-discrete topological space on $\mathcal{D}$. Subsequently, the geometric and topological properties of this space are extracted. The methods to do so have to be stable with respect to perturbations and noise in the data. Lastly, the extracted topological and geometric features must be summarized in a way that they can be proceeded with statistical or machine learning methods.

1.1.1 Software

There is a bunch of open source software available for topological data analysis and (persistent) (co)homology related calculations. Extensive lists (with discussions) can be found in [Ott+17], [Ada23] and [Bub23]. In the following, we will describe **ripser**, **gudhi** and **giotto-tda** more thoroughly, as these will be the implementations used for the illustrative examples in the following. Furthermore, with regard to the machine and deep learning methods presented in Chapter 3, we will focus on the python implementations.

RIPSER

Ripser [Bau21] was developed by Ulrich Bauer and started as a C++ code for the computation of Vietoris-Rips barcodes. With regard to efficiency, ripser has been state of the art until adaptations of the algorithm for parallel computing evolved with ripser++, [ZXW20]. With ripser.py [TSB18] there exists also a python binding and with ripserer.jl [Čuf20] a julia binding for ripser. These also include some plotting routines. Ripser does not offer the wealth of additional topological features that the other libraries do. To cite Ulrich Bauer: "It can do just this one thing, but does it extremely well."

Language	**PH Algorithm**
C++/Python	dual [SMV11] twist [CK11]
Coefficient Field	**Homology**
$\mathbb{Z}/p\mathbb{Z}$	simplicial
Input	**Complexes**
point cloud distance matrix	Vietoris-Rips
Visualizations	**Additional Features**
persistence diagrams lifetime of generators representative (co)cycles	-

Table 1.1: Overview of RIPSER.

GUDHI

Gudhi uses the algorithms of [DFW12] and [SV09] to compute persistent cohomology, in combination with the compressed annotation matrix implementation of [BDM13]. The latter guarantees improved time and memory performance via separation of the representation of the simplicial complex and the representation of the cohomology groups. Gudhi can calculate simplicial and cubical cohomology and thus can also possibly be used for greyscale image analysis. It offers the caclulation of large variety of complexes from either point clouds, distance matrices or correlation matrices. Unfortunately, not all of these functionalities are available in python (for example the Čech complex computation). Persistence intervals can be visualized via persistence diagrams or barcodes. More elaborate visualizations such as e.g. the entropy summary function [AGS20] or persistence landscapes [Bub20] are listed in Table 1.2. Gudhi also provides the calculation of metrics (bottleneck and Wasserstein distance) between persistence diagrams, diagram kernels (persistence Fisher kernel [LY18], heat kernel [Rei+14], persistent weighted Gaussian kernel [KFH16] and sliced Wasserstein kernel [CCO17]). Additional diagram features and vectorizations include persistence entropy [AGR17], topological vector [COO15] and complex polynomial [DF15]. For machine learning and deep learning, two libraries are related to gudhi: ATOL (Automatic Topologically-Oriented Learning) [Roy+21] and PersLay [Car+19] (a simple and versatile neural network layer for persistence diagrams).

Language	**PH Algorithm**
C++/Python	dual via simplicial maps [DFW12] multifield dual [SV09]
Coefficient Field	**Homology**
$\mathbb{Z}/p\mathbb{Z}$	simplicial/cubical
Input	**Complexes** (C++)
point cloud distance matrix correlation matrix	Vietoris-Rips Čech Witness α Cover Cubical Tangential Delaunay
Visualizations	**Additional Features**
nerves (e.g. via KeplerMapper) persistence diagram persistence density [CD19] barcode entropy summary function Betti curve persistence landscapes silhouette [Cha+13] persistence image [Ada+15]	topological vector persistence kernels complex polynomial metrics persistence entropy ATOL PersLay

Table 1.2: Overview of GUDHI.

GIOTTO-TDA

Giotto-tda [Tau+20] is based on a python backend of ripser and gudhi. It also incorporates directed simplicial cohomology via a python binding for flagser [Lue+19], developed as a sibling project. Giotto-tda combines gudhi's edge collapse algorithm [Pri] with ripser and thus improved on the state-of-the-art ([BP20]) runtimes for the computation of Vietoris-Rips barcodes. Giotto-tda was developed to integrate computational topological data analysis into machine learning and deep learning and as such includes parallelizable computations across batches of data. One of the main advantages is the inbuilt preprocessing, which enables the direct integration not only of point cloud data or distance matrices, but also of graphs, image data and time series as input data. Another benefit is the possibility to visualize data, persistence diagrams and persistence features such as Betti curves, persistence landscapes and persistence images. Compared to gudhi, giotto-tda offers less persistence kernels, but as an additional feature, the L^p distance between representations. Also, complexes can be calculated only implicitly on the data as a step involved in calculating persistence (co)homology.

Language	**PH Algorithm**
Python	Ripser and GUDHI backend
	Flagser backend via pyflagser
Coefficient Field	**Homology**
$\mathbb{Z}/p\mathbb{Z}$	undirected simplicial
	directed simplicial [Lue+19]
	cubical
Input	**Complexes**
point cloud	Vietoris-Rips
distance matrix	Čech
image data	α (weak) [Gio]
time series	cubical
Visualizations	**Additional Features**
persistence diagrams	metrics
Betti curves	persistence kernels
silhouettes	vector representations
heat representation [Rei+14]	curve features
persistence image	complex polynomial
persistence landscapes	amplitude [Tau+20]
	persistence entropy

Table 1.3: Overview of GIOTTO-TDA.

1.1.2 Data

Of the data sources already mentioned above, we will pick out three that are most relevant in the overlap of TDA and deep learning. In the following, we will consider point cloud data, image data and time series data. From each of the above classes, we will extract an example and use it to illustrate the concepts discussed in Chapter 1, 2.1 and 3.

Point Cloud Data

As an example for point cloud data, we take the 4-, 8- and 16-dimensional random point cloud data used as a benchmark data set for the TDA methods presented in [Ott+17]. Figure 1.4 shows the first three dimensions of the 8- and the 16-dimensional random point cloud. The first one consisting of 1000 data points, whereas the latter is much sparser with only 50 data points in total.

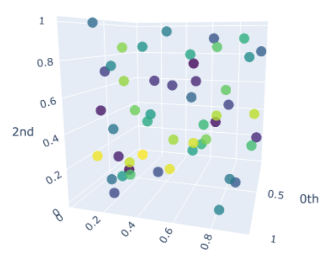


Figure 1.3: 16D point cloud

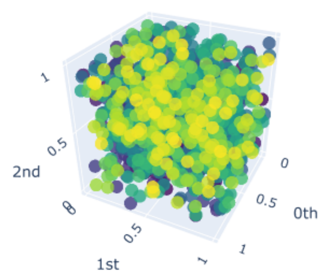


Figure 1.4: 8D point cloud

Image Data

For image data input (and cubical persistence) we use MNIST ([LCB]), a database of handwritten digits commonly used to benchmark image processing systems and algorithms. In addition to that, we use Fashion-MNIST ([XRV17]), a dataset of Zalando's article images. Both datasets are integrated by default in many machine learning libraries, e.g. pytorch, keras or tensorflow. Figure 1.5 and figure 1.6 show two examples thereof.

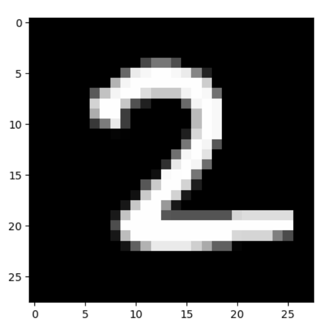


Figure 1.5: Example of MNIST

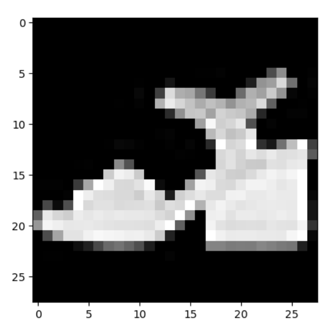
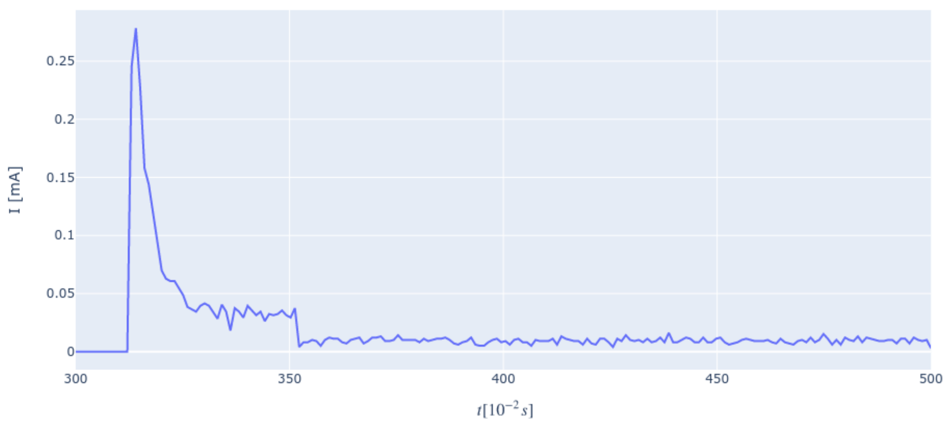


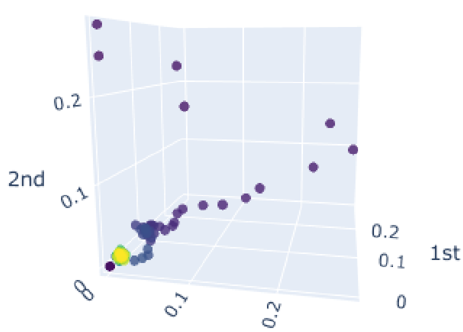
Figure 1.6: Example of FashionMNIST

Time Series Data

As an example for time series data, we take the DeepVALVE data set. It is an industrial data set containing a sequence of random closing and opening events of an industrial valve. In total, there are five different labels: 'start', 'lose', 'linear', 'stuck' and 'end'. These labels were set manually via markers on the current-time measurement of the engine driving the valve. A part of the DeepVALVE data set is shown in Figure 1.7a



(a) Part of the DeepValve time series data set. The blue line represents the electrical current per time.



(b) First 3D of Takens embedding of the DeepVALVE data segment shown above.

To input (uni- or multivariate) time series into the TDA pipeline, they have to be represented as a point cloud. This is done based on the Takens (time-delay) embedding technique [Tak81; PH13]. This technique involves two parameters:

the **embedding dimension** $d \in \mathbb{N}$ and the **time delay** $\tau \in \mathbb{R}$. Given a discrete time series $(X_0, X_1, ..., X_N)$, where $X_i \in \mathbb{R}$ for the univariate case and $X_i \in \mathbb{R}^n$ for the multivariate case, and a sequence of evenly sampled times $t_0, t_1, ..., t_N, \quad t_i \in \mathbb{R}$, one extracts a set of d-dimensional vectors of the form $(X_{t_i}, X_{t_i+\tau}, ..., X_{t_i+(d-1)\tau})$ for $i \in \{1, N\}$. The difference between t_{i+1} and t_i is called the **stride**. Giotto-tda involves dynamic parameter selection for d and τ, and time-delay embeddings can also be computed for several time series simultaneously. Applying dynamic Takens parameter search for the data segment shown in Figure 1.7a returns an optimal embedding dimension of 5 and time delay of 2. The first is not surprising, as we already know that there are five features present in the data. Figure 1.7b shows the first three dimensions for the Takens embedding of the DeepVALVE segment shown in Figure 1.7a.

1.2 Homotopy and Homology

The topological study of shape can be characterised by three properties [Car14]:

- **Independence of coordinate representation:** Only the pairwise distances between the points making up a shape matter.
- **Deformation Invariance**: Continuous transformations such as stretching or compression of a shape do not change its topological properties.
- **Compressive Representations:** Topology ignores the fine details of shape.

Intuitively, algebraic topology distinguishes spaces by counting the occurrences of geometric patterns. Doing this counting directly is not feasible, hence *equivalence classes* of occurrences of patterns under an equivalence relation are counted.

1.2.1 Homotopy

Homotopy is a way to determine whether two given spaces have 'the same shape' in a much coarser sense than homeomorphism [Hat02]. In the following, let X and Y be two topological spaces.

Definition 1.2.1 (Homotopic Maps) *Given two maps $f, g : X \to Y$, f and g are called homotopic $f \simeq g$, if there is a continuous map $H : X \times [0, 1] \to Y$ so that $H(x, 0) = f(x)$ and $H(x, 1) = g(x)$ for all $x \in X$.*

Homotopy is an equivalence relation on the set of continuous functions between X and Y. The set of equivalence classes (**homotopy classes**) with regard to this relation is a discrete invariant giving a high level description of the set of maps from X to Y. When there are fixed basepoints x_0 and y_0 for X and Y, based (or base-preserving) maps $f : X \to Y$ are maps for which $f(x_0) = y_0$. Likewise, based homotopies fullfill $H(x_0, t) = y_0$ for all t. Analogeously, based homotopy is an equivalence relation on the set of based maps from X to Y.

Homotopy Group

Let X be the based n-sphere S^n. Then the homotopy classes of base-preserving maps of S^n to Y exhibit a group structure with the group operation given by gluing of two spheres at their basepoint. The group $\pi_n(Y, y_0)$ is called the n-th **homotopy group** of Y. For $n \geq 2$ this group is abelian. Intuitively, the homotopy groups count n-dimensional holes in Y. A space Y is said to have an n-dimensional hole if and only if it contains an n-dimensional sphere that cannot be shrunk to a single point. This holds if and only if the map $S^n \to Y$ is not homotopic to the constant function and this is the case if and only if $\pi_n(Y, y_0) \ncong 0$.

Homotopy Equivalence

Definition 1.2.2 (Homotopy Equivalence) *Two topological spaces X and Y are said to be homotopy equivalent if there are continuous maps $f : X \to Y$ and $g : Y \to X$ such that gf is homotopic to the identity map in Y and fg is homotopic to the identity map in X.*

If two topological spaces X and Y are based homotopy equivalent then all their homotopy groups are isomorphic (Proposition 2.4 in [Car14]). Although the definition of homotopy groups appears quite easy, homotopy groups are very difficult to compute. Thus they are not used practically as an invariant for algebraic topology. Rather, one uses **homology groups**.

1.2.2 Homology

Let X be a topological space and A an abelian group. Homology associates a group $H_i(X, A)$ to X for each natural number $i \in \{0, 1, 2, ...\}$. For any field $\mathbb{F}$, $H_i(X, \mathbb{F})$ ($H_i(X)$ for short) is a vector space over $\mathbb{F}$. The dimension of $H_0(X)$ counts the number of path components in X, in general, the dimension of $H_n(X)$, called the n-th **Betti number**, counts the number of n-dimensional holes in X. As such, the homology groups $H_n(X)$ are closely related to the homotopy groups $\pi_n(X, x_0)$, but in general, homology is a coarser invariant than homotopy. If two topological spaces X and Y are homotopy equivalent, then the homology groups are isomorphic, but the converse is not true.

Example

The torus $T^2 = S^1 \times S^1$ and the wedge sum $S^1 \vee S^1 \vee S^2$ have isomorphic homology groups but differing homotopy groups, already for π_1 and π_2.

Homology was initially defined for combinatorial structures called **simplicial complexes** (Section 1.3.3), but Eilenberg showed that the definition of homology groups can be extended to all spaces [ES45]. Simplicial complexes are practicable because homology is calculable on them: It boils down to linear algebra. Many topological spaces of interest are either explicitly simplicial complexes or at least homotopy equivalent to such [Car14].

1.3 Constructing Complexes on Data

A complex structure on a data set $\mathcal{D}$ is the power set of certain subsets of $\mathcal{D}$ chosen in a way that allows defining "gluing" maps between them. This defines a non-trivial topological structure on $\mathcal{D}$ in a combinatorial way. This structure should approximate the structure of the underlying geometric space that the point cloud data is supposed to be drawn from. In the following, CW complexes, Δ-complexes and simplicial complexes will be presented. CW complexes herein denote the most general structure, where the "gluing" between the subsets called *cells* is defined inductively. Δ-complexes are special CW complexes, where the relevant subsets are called *simplices* and the maps between them are restricted. Simplicial complexes are Δ-complexes where simplices must intersect in a specific way.

1.3.1 CW Complexes

needed for Chapter 4, [TP21]
A nonempty CW complex X is an unbased Hausdorff space together with a sequence of unbased subspaces called *skeleta*

$$X^0 \subseteq X^1 \subseteq ... \subseteq X^n \subseteq X^{n+1}..., \tag{1.1}$$

whose union is X.

Definition 1.3.1 (Adjunction Space) *Let X and Y be unbased topological spaces, with $W \subseteq X$ and a free map $\phi : W \to Y$. In the disjoint union $Y \amalg X$ introduce the equivalence relation $w \sim \phi(w)$ for all $w \in W$. The **adjunction space** $Y \cup_\phi X$ is given by the resulting identification space.*

CW complexes represent a way to inductively construct a topological space on $\mathcal{D}$ via the following procedure [Hat02; Ark10]:

- Start with $\mathcal{D}$ whose elements are called 0-cells or *vertices*.
- Assume that X^{n-1} is defined for $n \geq 1$ and obtain X^n from X^{n-1} by attaching n-cells via *attaching maps* $\phi_\beta : S^{n-1}_\beta \to X^{n-1}$, where for some index set B and each $\beta \in B$, S^{n-1}_β denotes the $(n-1)$-sphere S^{n-1} centered at β. The ϕ_β determine a free (i.e. non-basepoint preserving) map $\phi : \amalg_{\beta \in B} S^{n-1}_\beta \to X^{n-1}$. Then the n-skeleton X^n is defined as

$$X^n = X^{n-1} \cup_\phi \amalg_{\beta \in B} E^n_\beta, \tag{1.2}$$

where $E^n_\beta = E^n$ is the unit ball in Euclidian n-space $\mathbb{R}^n$ centered around β, with

$$E^n = \{x \in \mathbb{R}^n | \; |x| \leq 1\}. \tag{1.3}$$

- The composition of the inclusion $E^n_\beta \to X^{n-1} \sqcup \amalg_{\beta\in B} E^n_\beta$ and the quotient map $X^{n-1} \sqcup \amalg_{\beta\in B} E^n_\beta \to X^n$ gives a continuous function called the *characteristic function* of pairs $\Phi_\beta = \Phi^n_\beta : (E^n_\beta, S^{n-1}_\beta) \to (X^n, X^{n-1})$. Then the subset $\Phi_\beta(E^n_\beta) \subseteq X^n \subseteq X$ is called an *n-cell* and denoted as $\bar{e}^n_\beta$.

- This inductive process can be either stopped at finite stage, setting $X = X^n$ for some $n < \infty$, or can be continued indefinitly, setting $X = \cup_n X^n$. In the latter case X is given the weak topology meaning that a set $A \subset X$ is open (or closed) iff $A \cap X^n$ is open (or closed) in X^n for each n.

A CW complex is called finite if it has finitely many cells and finite-dimensional, if, for some N, the N-skeleton $X^N = X$. The smallest N for which this is the case is called the dimension of X, i.e. the maximum dimension of cells of X. Furthermore, a CW complex is said to be **regular** if every attaching map can be chosen to be a homeomorphism on the entire cell being attached.

Examples

- A 1-dimensional cell complex $X = X^1$ is called a **graph** and consists of vertices (0-cells) and edges (1-cells) attached to the vertices.

- The sphere S^n has the structure of a two-cell complex $\bar{e}^0, \bar{e}^n$. The n-cell is attached by the constant map $S^{n-1} \to \bar{e}^0$.

- The n-dimensional real projective space admits a CW structure with one cell in each dimension.

- Polyhedra are naturally CW complexes.

A subcomplex of a cell complex X is defined as a closed subspace $A \subset X$ that is a union of cells of X. A subcomplex of a cell complex is a cell complex in its own right: Since A is closed, the characteristic map and in particular the attaching map of each cell in A has an image contained in A. Up to weak equivalence (i.e. isomorphism of the homotopy groups) every topological space is a CW complex by the following theorem:

Theorem 1.3.1 (CW Approximation Theorem) *If X is any space, then there is a CW complex Y and a map $f : X \to Y$ inducing isomorphisms on all homotopy, homology and cohomology groups.*

1.3.2 Δ-Complexes

Definition 1.3.2 (Standard n-Simplex) *The standard n-simplex is the convex hull of the unit vectors in the $n+1$-dimensional Euclidian space along the coordinate axes:*

$$\Delta^n = \{(t_0, ..., t_n) \in \mathbb{R}^{n+1} \mid \sum_i t_i = 1 \text{ and } t_i \geq 0 \ \forall i\}, \tag{1.4}$$

where the vertices $(t_0, ..., t_n)$ *are ordered.*

The ordering of the vertices is crucial for the definition of an arbitrary n-simplex.

Definition 1.3.3 (n-Simplex) *An arbitrary n-simplex on some ordered points $[v_0, ..., v_n]$ in the $n+1$ dimensional Euclidian space is defined as a map from the standard n-simplex to $[v_0, ..., v_n]$ preserving the order of the vertices: $(t_0, ..., t_n) \to \sum_i t_i v_i$. The coefficients t_i are called the **barycentric coordinates** of the point $\sum_i t_i v_i$ in $[v_0, ..., v_n]$.*

Deleting one of the $n+1$ vertices of an n-simplex $[v_0, ..., v_n]$ results in an $n-1$-simplex, called the **face** of $[v_0, ..., v_n]$. This is a subsimplex of $[v_0, ..., v_n]$, with the remaining vertices being ordered accordingly. The union of all the faces of Δ^n is called the **boundary** $\delta\Delta^n$ of Δ^n, and $\Delta^n - \delta\Delta^n$ is called the interior $\mathring{\Delta}^n$ of Δ^n.

Definition 1.3.4 (Δ-complex) *A Δ-complex is another way to define structure on the data $\mathcal{D}$ via a collection of maps $\sigma_\alpha : \Delta^n \to \mathcal{D}$, with n depending on the index α, such that [Hat02]:*

- *The restriction of σ_α to the interior of Δ^n is injective and each point in $\mathcal{D}$ is the image of exactly one such restriction map.*
- *Each restriction of σ_α to a face of Δ^n consists one of the maps $\sigma_\beta : \Delta^{n-1} \to \mathcal{D}$.*
- *A set $A \subset \mathcal{D}$ is open iff σ_α^{-1} is open in Δ^n for each σ_α.*

Using the third property, a Δ complex on $\mathcal{D}$ can be built as a quotient space of a collection of disjoint simplices and thus must be a Hausdorff space.

1.3.3 Simplicial Complexes

Simplicial complexes are examples of regular CW complexes. Given a set V, an **abstract simplicial complex** with vertex set V is a set $\tilde{K}$ of finite subsets of V such that the elements of V belong to $\tilde{K}$ and for any $\sigma \in \tilde{K}$ and $\tau \subset \sigma$, also $\tau \in \tilde{K}$. The dimension of an abstract simplex σ is given by $\mathrm{card}(\sigma) - 1$ and the dimension of $\tilde{K}$ is given by the largest dimension of its simplices. With any abstract simplicial complex, one can associate a **geometric simplicial complex**.

Definition 1.3.5 (Geometric Simplicial Complex) *A simplicial complex K in $\mathbb{R}^n$ (geometric simplicial complex) is a collection of simplices in $\mathbb{R}^n$ such that:*

1. *Every face of a simplex of K is in K.*
2. *The intersection of any two simplices of K is a face of each of them.*

A subcollection L of K that contains all faces of its elements is a simplicial complex in its own right, called a subcomplex. The collection of all simplices of K of dimension at most p is called the p-skeleton of K and denoted as K^p.

The points of the collection K^0 are called the vertices of K. A **filtration** of K is defined as a partial ordering on the simplices of K such that every prefix of the ordering is a subcomplex. It is denoted as $\leq_K$.

Definition 1.3.6 (Subspace Topology) *Let $|K|$ be the subset of $\mathbb{R}^n$ that is the union of the simplices of K. One way to topologize $|K|$ is by giving each simplex its natural topology as a subspace of $\mathbb{R}^n$.*

For infinite K one needs a finer topology than the subspace topology $K \subset \mathbb{R}^n$.

Definition 1.3.7 (Final Topology) *Let $f_i : F_i \to K$ be a family of maps whose domains carry a topology. The final topology on K is the topology for which a subset $U \subset K$ is open whenever $f_i^{-1}(U) \subset F_i$ is open for all i.*

For infinite K, we give $|K|$ the final topology with respect to all maps $|\sigma| \xhookrightarrow{|j|} |K|$, where $\sigma \xhookrightarrow{j} K$ is the inclusion of a finite subcomplex. Thus $U \subset |K|$ is open iff $U \cap |\sigma|$ is open for all finite $\sigma \subset K$. The space $|K|$ is called the **underlying space** of K. $|K|$ is Hausdorff and if K is finite, $|K|$ is compact. The combinatorial description of any geometric simplicial complex K gives rise to an abstract simplicial complex $\tilde{K}$. Conversely, for a given abstract simplicial complex $\tilde{K}$, if K is a geometric simplicial complex whose combinatorial structure is the same as $\tilde{K}$, K is called the **geometric realization** of $\tilde{K}$. Thus simplicial complexes can either be described as purely combinatorial objects or topological spaces. Common examples include [Ott+17]:

- Čech Complexes,
- Vietoris-Rips-Complexes,
- Delaunay Complexes,
- Alpha Complexes,
- Witness Complexes,
- Graph-Induced Complexes,
- Cubical Complexes.

Cubical complexes are based on standard n-cubes rather than n-simplices. They form the basis of cubical (co)homology for image analysis, as grey scale images are naturally arranged in a grid.

1.3.4 Simplicial Homology

Let $\mathbb{F}$ be a field, K a simplicial complex and p a dimension. Homology for simplicial complexes is defined via formal sums of p-simplices in K called p-chains. Together with the addition operation the p-chains form a group (for field coefficients in fact a vector space) denoted as $C_p = C_p(K)$. For a p-chain

$c = \sum a_i \sigma_i$, $a_i \in \mathbb{F}$, $\sigma_i \in K$, the boundary is given by the sum of the boundaries of its simplices. Hence taking the boundary maps a p-chain to a $(p-1)$-chain: $\delta_p : C_p \to C_{p-1}$.

Definition 1.3.8 (Simplicial Homology) *For any $p \in \mathbb{N}_0$, the p-th homology of a simplicial complex K is the quotient vector space*

$$H_p(K) := \mathit{kernel}(\delta_p)/\mathit{image}(\delta_{p+1}). \tag{1.5}$$

Elements in the image of δ_{p+1} are called p-boundaries, and elements in the kernel of δ_p are called p-cycles. The fundamental Lemma of Homology states that the boundary of a boundary is zero, i.e. $\delta_p \delta_{p+1} c = 0$ for every integer p and every $(p+1)$-chain c, [EH10]. Hence the group of p-boundaries B_p is a subgroup of the cycle group Z_p. Thus simplicial homology counts the cycles that are not boundaries. Intuitively, p-cycles that are not boundaries represent p-dimensional holes. If K is a simplicial complex of dimension n, then $H_p(K) = 0$ for any $p > n$ and thus one obtains the following exact sequence of vector spaces and linear maps:

$$0 \xrightarrow{d_{n+1}} C_n(K) \xrightarrow{d_n} \dots \xrightarrow{d_2} C_1(K) \xrightarrow{d_1} C_0(K) \xrightarrow{d_0} 0. \tag{1.6}$$

1.3.5 Vietoris-Rips and Čech Complexes

Of the simplicial complexes listed in Section 1.3.3, two simplicial complexes central in use are the (Vietoris-)Rips (due to its computational simplicity) and the Čech complex (due to its relation to the Nerve Theorem 1.4.1), as described in Section 1.4 below. The theoretical guarantees and the worst-case sizes of complexes as functions of the cardinality N of their vertex set are summarized in [Ott+17]. For the Čech- and Rips-complex, the complexes for N vertices have size of order $2^{\mathcal{O}(N)}$. Approaches to limit the dimension of the simplices to the dimension of the ambient space include the Delaunay and the alpha complex (the latter constitutes a subcomplex of the Delaunay complex). Unfortunately these rely on geometric constructions (Voronoi calculation), which also makes them uncomputable for large point clouds in high dimensions.

Example

For the deepVALVE data segment (figure 1.7a), the resulting alpha complex computed with gudhi is of dimension 5 with 136731 simplices.

Definition 1.3.9 (Vietoris-Rips Complex) *Let (M, d_M) be a metric space, $X \subseteq M$. For $\epsilon \in \mathbb{R}$ a simplicial complex $\mathit{Rips}(X, \epsilon)$ is defined on the vertex set X as follows:*

$$[x_0, x_1, \dots, x_k] \in \mathit{Rips}(X, \epsilon) \leftrightarrow d_M(x_i, x_j) \leq \epsilon \tag{1.7}$$

for all $i, j \in \{0, k\}$. For $\epsilon \leq 0$ $\mathit{Rips}(X, \epsilon)$ consists of the vertex set X alone.

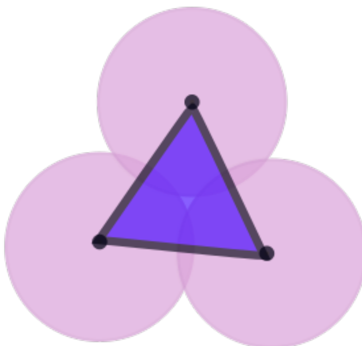


Figure 1.8: Rips-Complex: Graph with a 2-Simplex.

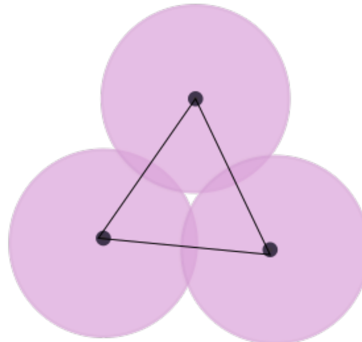


Figure 1.9: Čech Complex: Graph.

On the contrary, the Čech Complex for a set of points $X \subseteq M$ is defined via a certain cover of this set.

Definition 1.3.10 (Cover) *Let X be set and $C = \{U_\alpha : \alpha \in A\}$ an indexed family of subsets $U_\alpha \subset X$ indexed by a set A. Then C is a cover of X if*

$$X = \cup_{\alpha \in A} U_\alpha. \tag{1.8}$$

*If the cover consists of open sets, it is called an **open cover**.*

Definition 1.3.11 (Čech Complex) *Let (M, d_M) be a metric space, $X \subseteq M$ and $\epsilon \in \mathbb{R}$. The Čech complex $\check{C}ech(X, \epsilon)$ is defined as follows:*

$$[x_0, x_1, ..., x_k] \in \check{C}ech(X, \epsilon) \leftrightarrow \cap_{i=0}^{k} B(x_i, \epsilon) \neq \emptyset, \tag{1.9}$$

where $B(x, \epsilon) = \{x' \in X : d_M(x, x') \leq \epsilon\}$ denotes the closed ball with centre $x \in X$ and radius ϵ.

Theorem 1.3.2 *For all $\epsilon > 0$, the following inclusions hold:*

$$\check{C}ech(X, \epsilon) \subset Rips(X, \epsilon) \subset \check{C}ech(X, 2\epsilon). \tag{1.10}$$

Thus the Vietoris-Rips complex approximates the Čech complex. In [SG07] it has been proven that the above inclusion can be tightened at most to

$$\mathrm{Rips}(X, \epsilon') \subset \check{\mathrm{C}}\mathrm{ech}(X, \epsilon) \subset \mathrm{Rips}(X, \epsilon) \quad \text{for} \quad \frac{\epsilon}{\epsilon'} \geq \sqrt{\frac{2d}{d+1}} \tag{1.11}$$

for a set of points $X \in \mathbb{R}^d$ and $\check{\mathrm{C}}\mathrm{ech}(X, \epsilon)$ the Čech complex of the cover of X by balls of radius $\epsilon/2$.

Figure 1.8 and Figure 1.9 illustrate the difference between the Rips- and the Čech complex: For the Rips-complex the pairwise intersections between the three disks lead to a two-simplex, whereas for the Čech construction we obtain only a one-simplex as the balls do not have a threefold intersection. Čech complexes are computationally unfavourable whereas Rips-complexes are determined completely by their 1-skeleta - the proximity graph of vertex points.

1.3.6 Theoretical and Computative Difficulties

One disadvantage of Rips- and Čech-based complex constructions is their rigid dependence on the (not necessarily metric [CO07]) distance between points. Thus arbitrarily small perturbations in the locations of the points can have dramatic effects on the topology of the associated simplicial complexes. Considering homology over a filtration of distance parameters rather than a static distance parameter is the idea underlying persistence homology in order to obtain robust topological invariants for the given data set.
Another problem with simplicial complexes is of computative nature: Also for reasonable vertex cardinalities, the number of simplices can become uncomputably high already for small Betti numbers counting holes of low dimension, as the example below illustrates for samples of the data presented in Section 1.1.2. Calculations were performed on a 64-Bit GNU/Linux machine with 12 cores 3.60GHz Intel(R) Xeon(R) CPU E5-1650 v4.

Example

For the 16-dimensional random point cloud (50 vertices) , one obtains the following Rips complexes with gudhi (for $\epsilon \leq 12$):

- **dimension 1**: 1275 simplices.
- **dimension 2**: 20875 simplices.

For the 8-dimensional random point cloud (1000 vertices) , one obtains the following Rips complexes under the same configurations:

- **dimension 1**: 500500 simplices.
- **dimension 2**: 166667500 simplices.

For the deepVALVE data segment shown in figure 1.7a (192 vertices for 5 dimensions), one obtains:

- **dimension 1**: 18528 simplices.
- **dimension 2**: 1179808 simplices.
- **dimension 3**: 56050288 simplices.

For these reasons, sparsified versions of the Čech and the Rips-complex have been introduced [She13; KS13; DSW16], however, up to now these are not implemented in the TDA libraries presented above.
A sparsified version of the Rips-complex based on [Buc+14] is available in gudhi (and giotto-tda).
Witness complexes (implemented in gudhi) also constitute an approximation of Čech or Rips complexes. They are constructed as follows: Given the point cloud data $\mathcal{D}$, a subset L of vertices is chosen randomly, or using a weak optimisation

strategy for good distribution. The cardinality of L constrains the complexity of the detectable topology. Some $x_0, x_1, ..., x_k \in L$ span a k-simplex if and only if:

- There is a point $w \in \mathcal{D}$ whose $k+1$ nearest neighbours in L are $x_0, x_1, ..., x_k$ (weak witness); and
- w is equidistant from $x_0, x_1, ..., x_k$ (strong witness).

As the existence of a strong witness point w in a finite point set is almost a probability zero event, one either needs to introduce a tolerance parameter instead of equidistance or use weak witness complexes. To overcome the curses of dimensionality, also weak witness complexes have to be used with some tolerance parameter. Yet, heuristically, weak witness complexes give good results at least for low dimensions($2D$, $3D$). For high dimensions they fail to capture the topology unless imposed with extra structures. Graph-induced complexes were proposed in [DFW13] to join the advantages of the Rips- with those of the witness complex with the aim to limit the number of simplices but approach the theoretical guarantees of the Rips-complex.

Definition 1.3.12 (Graph Induced Complex [DFW15]) *Let $G(V)$ be a graph with the vertex set V and let $\mu : V \to V'$ be a vertex map where $\mu(V) = V' \subseteq V$. The graph induced complex $\mathcal{G}(G(V), V', \mu)$ is defined as the simplicial complex where a k-simplex $\sigma = \{v'_1, v'_2, ..., v'_{k+1}\}$ is in $\mathcal{G}(V, V', \mu)$ if and only if there exists a $(k+1)$-clique $\{v_1, v_2, ..., v_{k+1}\} \subseteq V$ such that $\mu(v_i) = v'_i$ for each $i \in \{1, 2, ..., k+1\}$.*

For a finite metric space (P, d_P) and a subset $Q \subset P$ let $\mu_d : P \to Q$ denote the nearest point map where $\mu_d(p)$ is apoint in $\text{argmin}_{q \in Q} d_P(p, q)$. For a graph $\mathcal{G}(P)$ whose vertex set is P, a graph induced complex is devined as $\mathcal{G}(G(P), Q, d) := \mathcal{G}(G(P), Q, \mu_d)$.

An implementation of graph induced complexes is available in gudhi.

1.3.7 Simplicial Complex Selection

Starting with a point cloud $\mathcal{D} \in \mathbb{R}^n$ the question is which simplicial complex one should use to describe the topological features of its underlying structure. For high dimensions, the smallest-enclosing-ball-problem and the difficulty of high-dimensional hyperplane intersection computation represent restrictions which let complexes based on the overlap of ϵ-balls around points such as the Čech complex and the related alpha complex appear unsuitable. However, in general the answer to the question above can be rather found in the art of data presentation and preparation than in theory.

At least, from a theoretical perspective, one would like to compute either the Čech complex or the alpha complex, since both fall within the scope of the Nerve Theorem 1.4.1, however, computably, this is not possible for most realistic data sets. As already mentioned above, some sparsified approximations exist, but this comes with the risk of loosing important parts of the data. In general, one

does not know in advance at which 'scale' of the data topological features of interest appear and in which regions of the point cloud they are located.
Due to its computative simplicity, for point cloud data, the Rips-complex is most widely used, although the cardinality of simplices is not bounded by the ambient dimension (contrary to the alpha complex) and thus explodes as was illustrated in the example above. For large data sets, witness complexes have proven quite useful but also endure problems in high dimensions.
The question always remains whether the persistent homology calculated based on these complexes reflects the topological features that are of interest within the point cloud $\mathcal{D}$.

1.4 Nerve Theorem(s)

The **Nerve Theorem**, in its different versions given by Borsuk [Bor48], Leray [Ler45], Weil [Wei52], McCord [McC67], Björner [Bjö03] and Barmak [BM06], states that the intersection pattern of a cover of a topological space X fully determines X with respect to the algebraic topological invariants under suitable hypotheses. Among these hypotheses are contractibility or acyclicity of all non-empty and finite intersections of cover elements.

1.4.1 Nerve Theorem for Good Covers

Definition 1.4.1 (Good Cover) *A good cover is an open cover in which all sets and all nonempty intersections of finitely-many sets are contractible.*

Nerves ([Ale28]) provide a way to construct a computable topological space from a point cloud via a suitable cover. Each set of the cover being a local cluster or a grouping of data points sharing some common properties, the nerve is a way to compactly and globally describe the relationship between these sets through their intersection patterns.

Definition 1.4.2 (Nerve) *Given a simplicial complex K and a cover $\mathcal{U}$, the nerve $\mathcal{N}(\mathcal{U})$ is defined as the simplicial complex whose k-simplices represent the non-trivial intersection of subsets of $\mathcal{U}$ of size $k+1$.*

The nerve is a subcomplex of the standard n-simplex and for an indexing set J, its simplices are denoted by Δ^J, where Δ^J is the $(J-1)$ dimensional face of Δ^n defined on J. One can encode a cover $\mathcal{U}$ as a map from K to $\mathcal{N}(\mathcal{U})$, where each simplex $\sigma \in K$ is mapped to $\mathcal{N}(\sigma)$, the simplex in $\mathcal{N}(\mathcal{U})$ which lists the cover sets containing σ. A prominent example for a nerve is the Čech complex which is by definition the nerve of a good cover. In contrast to this, the Vietoris-Rips complex is usually not defined as the nerve of a cover but is isomorphic to a nerve (Lemma 4 in [GM05]), which however need not be the nerve of a good cover. Therefore, Vietoris-Rips complexes do not fall within the scope of the **Nerve Theorem**.

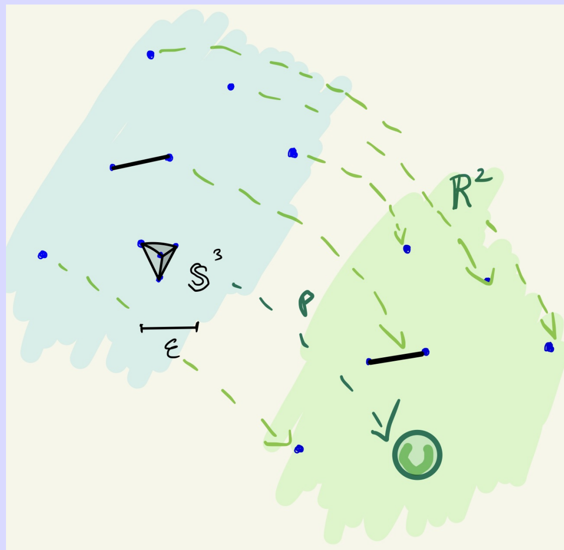
Theorem 1.4.1 (Nerve Theorem ([Hat02])) *Let $\mathcal{U}$ be an open cover of a paracompact space X such that every nonempty intersection of finitely many sets in $\mathcal{U}$ is contractible. Then X is homotopy equivalent to the nerve $N(\mathcal{U})$.*

1.4.2 Nerve Theorem and Rips/Čech Complexes

From the Nerve Theorem follows that the Čech complex of a point cloud Q is always homotopic to the union of diameter-ϵ balls about Q. Hence the Čech complex can be seen as an appropriate simplicial model for the topology of the point cloud.

This is not the case for the Vietoris-Rips complex. Unlike the Čech complex, even when Q is a finite subset of $\mathbb{R}^n$, the Vietoris-Rips complex $\mathrm{Rips}(Q,\epsilon)$ does not admit a geometric realization in $\mathbb{R}^n$. It can have nontrivial topological invariants (homotopy or homology groups) of dimension n and above. To illustrate this, consider a point cloud $Q \in \mathbb{R}^n$ and the projection map $p : \mathrm{Rips}(Q,\epsilon) \to \mathbb{R}^n$ taking the vertices of the Rips-complex to Q and the k-simplices of $\mathrm{Rips}(Q,\epsilon)$ to the convex hull of the associated vertices in Q. The image of p in $\mathbb{R}^n$ is called the **shadow** of the Vietoris-Rips complex. In general one wants that the images of complexes in $\mathbb{R}^n$ lie close to the data, however there is no chance of $\mathrm{Rips}(Q,\epsilon)$ being homeomorphic to its shadow and also homotopy equivalence is too much to ask, [Cha+10]. In fact, the domain of the projection p is likely to have higher dimension than the codomain. Herein, the choice of the parameter ϵ is crucial as is demonstrated in the following example.

Example: Vietoris-Rips Bubbles [Ghr14]



Projection of some points Q in $\mathbb{R}^2$ to their shadow in $\mathbb{R}^2$. The points on the right bottom of the simplicial complex form a simplicial 3-sphere $\mathbb{S}^3$ and are projected to the convex set denoted as a green bubble. Such bubbles are artifacts of the Vietoris-Rips complex and not representative for the point cloud topology.

In the above example, the data points were distributed rather sparsely except for one region at the right bottom. This region causes a 3-dimensional Vietoris-Rips bubble to appear. This is a topologically non-significant artifact that can appear when constructing Vietoris-Rips complexes on data. However, empirically, it is known that Vietoris-Rips constructions capture the topology of large scale structures correctly [GM05].

1.4.3 Nerve Theorem for Covers by Subcomplexes

Definition 1.4.3 (Nerve for Subcomplexes) *Given a simplicial complex K and a finite collection of subcomplexes U, the nerve of U, denoted $N(U)$ is the abstract simplicial complex whose vertices are the subcomplexes in U and whose simplices are the subcollections of U with a nonempty intersection.*

Definition 1.4.4 (k-Connectedness, [Bjö03]) *A space X is called k-connected ($k \geq 0$) if for every $0 \leq r \leq k$, every map of the r-sphere into X is homotopic to a constant map.*

Given a simplicial complex K and a cover of K by subcomplexes $\mathcal{U} = \{L_i\}_{i \in I}$, it has been proved by [Bjö03] that if every k-wise intersection of these subcomplexes is $(k - t + 1)$-connected, then K and the nerve of the covering $\mathcal{N}(\mathcal{U})$ are homotopy equivalent. Yet, for a simplicial complex K on a point cloud Q and a cover of K by subcomplexes $\mathcal{U} = \{L_i\}_{i \in I}$ the intersections of the cover elements are not necessarily connected but the connected components of the non-empty intersections are contractible [Car09]. The Nerve Theorem 1.4.1 does not apply to this case, such that $N(\mathcal{U})$ is not in general homotopy equivalent to K.

1.5 The Mayer-Vietoris Blowup Complex

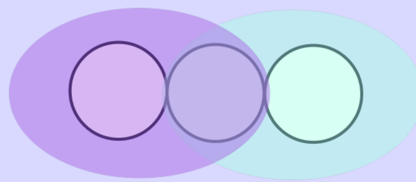
needed for Chapter 4, [TP21]
As the worst case time and space complexities for the computation of persistent homology are $\mathcal{O}(n^3)$ and $\mathcal{O}(n^2)$, with n the total number of simplices in the filtration, this shows that sparsified versions of complexes as well as data partition and parallellism are highly needed to remain computable. Especially in view of the fact that the size of the datasets processed in deep learning is much larger than what is usual in TDA application. Unfortunately, the literature and even more the implementation of parallel computing in topology is comparably young. One approach developed by [LZ14] relies on the **Mayer-Vietoris blowup complex**, a spatial version of the **Mayer-Vietoris spectral sequence** [ZC05] (the latter is further explained in Chapter 2.1). Parallelized homology firstly requires a decomposition of the space into pieces and secondly a recipe to put these parts together to a whole again. When a cover is used to decompose a space into pieces - corresponding to subspaces and their various intersections - this recipe is given by the Mayer-Vietoris spectral sequence which expresses the relationship between the homology of these subspaces to the homology of the space itself. The Mayer-Vietoris blowup complex is a topological space which encodes the data given as input to the spectral sequence and whose homology groups are isomorphic to those of the original space.

Definition 1.5.1 (Blowup Complex [LZ14]) *Given a simplicial complex K and a cover $U = \{U_i\}_{i \in [n-1]}$ of n subcomplexes, let $K^J = \cap_{k \in J} U_j$ for $J \subseteq [n-1]$. The Mayer-Vietoris blowup complex $K^U \subseteq K \times \Delta^{n-1}$ of K and U is defined as:*

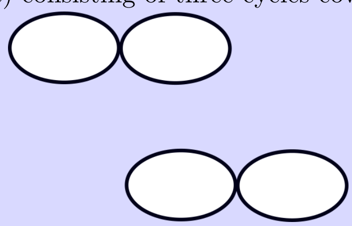
$$K^U = \cup_{\emptyset \neq J} K^J \times \Delta^J, \tag{1.12}$$

where $\times$ denotes the Cartesian product and Δ^J is a face of $\mathcal{N}(\mathcal{U})$.

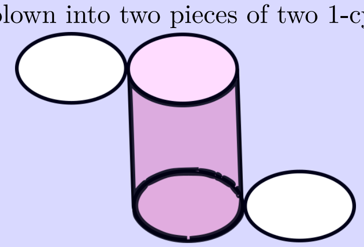
Example taken from [ZC08]



(a) Graph (space) consisting of three cycles covered by two sets.



(b) The graph is blown into two pieces of two 1-cycles each at $t = 0$.



(c) Blowup at $t = 1$.

The middle cycle of the original space in (a) is contained in the intersection of the cover sets, which makes it appear in both local pieces in step (b). To recover the global topology, the two copies of the middle cycle are equated by gluing a cylinder to them which results in construction (c). This is called the *Mayer-Vietoris blowup complex*, which has the same number of cycles as the original space but also incorporates the geometric cover information within its structure.

The Mayer-Vietoris blowup complex is defined as a subspace of a product space, which is triangulated with a simplicial complex as described in [Ito87], page 262. The homology of the product space is obtained from the homology of its factors through the Alexander-Whitney map. As was illustrated in the above example, to construct the blowup complex, one does not tear or glue, but only stretch certain pieces. The blowup complex has the same topology as the original space for open coverings of spaces homeomorphic to finite CW-complexes and also for coverings of finite simplicial complexes by subcomplexes (Lemma 1 in [ZC08]).

1.6 Reconstruction

Investigating the structure of a given point cloud intrinsically includes the assumption that the points are sampled from a shape. The aim of topological data analysis is to reconstruct this shape. One of the simplest methods to do so is by constructing an α-offset of the sample points.

Definition 1.6.1 (α-Offset) *An α-offset of a point set $Q \subseteq \mathbb{R}^n$ is a union of n-balls centered at the sampled points with radius α.*

If the underlying shape is a smooth manifold M, from which the points Q were sampled in a sufficiently dense way, [NSW08] proved that the α-offset is homotopic to M. Taking up this idea, the question arises under which conditions a given complex construction on the points will capture the topological and geometric properties of a suitable α-offset (for a specific α or a set of different α). The difficulty in creating geometric or topological criteria for this task is that these depend on properties of the sampling as well as on properties of the sampled shape. Problems include data sparsity, redundancy, noise or non-smoothness of the shape itself. Nevertheless, for surfaces embedded into $\mathbb{R}^3$, the *Delaunay triangulation* (a cell complex subdividing the convex hull of the sampling) is expected to match the surface with respect to topological and geometric properties [CG06]. This expectation is driven by the fact, that the reconstruction properties of the Delaunay triangulation for Lipschitz surfaces are well understood and theoretically guaranteed [BO06]. Unfortunately, for higher dimensions, the computational complexity of the Delaunay triangulation grows exponentially with the dimension [AAD07].

Generally, the first step towards surface reconstruction includes imposing sampling conditions on the points. Examples include the Hausdorff distance, indicating that the output is topologically correct if the Hausdorff distance is smaller than some *topological feature size* of the shape. Others are given by the *reach*, i.e. the infimum of distances between points in the shape and points in its medial axis and more elaborate measures such as the *convexity defects* proposed in [ALS13]. Subsequently these measures are used to derive sampling conditions for the given (simplicial) complex construction on the data. For the Čech complex, its homotopy equivalence to a given α-offset is intrinsic to its definition as nerve of a good cover. However, the problem with Čech complexes is that $Q \subseteq \mathbb{R}^n$ $n > 6$ the computation becomes unamenable. Thus there is an interest in using clique complexes such as the Vietoris-Rips complex, for which one only needs to store the metric pairwise distances between points. Ideas to use the Vietoris-Rips complex in a way guaranteeing topologically correct approximations of shapes include using either algebraic ([CO07] or geometric ([ALS13]) relations of the Vietoris-Rips complex to Čech complex constructions on a given point cloud Q.

One aspect of the reconstruction paradigm is topological estimation, whose aim is not to find a faithful approximation of the underlying shape X of a point cloud Q, but to infer the topological invariants of X from Q. This approach falls into the regime of **algebraic topology**. Algebraic topology comprises the investigation of spaces by mapping them to algebraic objects such as groups, but also more elaborate structures (e.g. rings, modules and algebras) arise. These objects, called *lanterns* by [Hat02], are created in a *functorial* way, such that continuous maps between spaces are projected onto homomorphisms between their algebraic images. The hope is to find algebraic images with enough detail to reconstruct accurately the shapes of *all* spaces. As pointed out above, homology is computable on a given simplicial complex constructed on a point cloud Q . Thus homology offers itself as an invariant to topologically infer shapes from point clouds. One classical approach to do so is to build a nested sequence of spaces $\mathcal{K}^0 \subseteq \mathcal{K}^1 \subseteq ... \subseteq \mathcal{K}^m$ (which might be subcomplexes of a simplicial complex) and studying the **persistence** of homology classes throughout this sequence. It has been proved by [CL07] and [CEH05] that, under comparably mild sampling conditions the **persistent homology** of the sequence defined by the α-offsets of a point cloud Q coincides with the homology of the underlying shape X. Even more, if the Hausdorff distance between Q and X is less than ϵ, for small enough ϵ, then for all $\alpha \geq \epsilon$ the canonical inclusion map of α-offsets $Q^\alpha \hookrightarrow Q^{\alpha+2\epsilon}$ induces homomorphisms between homology groups, whose images are isomorphic to the homology groups of X. Combined with the **Structure Theorem** (Theorem 2.1 in [ZC05]) which states that the persistent homology of the sequence $\{Q^\alpha\}_{\alpha\geq 0}$ is fully described by a finite set of intervals (the **persistent barcode** or **persistence diagram**) this implies that the homology of X can be deduced from this set of intervals by simply removing intervals as small that they are considered as topological noise.

CHAPTER 2

PERSISTENT HOMOLOGY

Abstract

Persistent homology is presented from an algebraic and algorithmic perspective with a view towards localized homology. Concepts are illustrated with the implementations presented in Section 1.1.1 on the samples of the data from Section 1.1.2.

Contents

2.1 Introduction

There are many excellent resources for an introduction to persistent homology: [Ghr14] and [Oud15] present persistent homology from a (quiver) representation-theoretic aspect, a theoretical introduction from a comprehensive algorithmic perspective can be found in [AJ05]

2.2 Persistence

Persistence ranks topological attributes by their life time in a filtration. Persistence means persistence in being a feature in the face of growth of the filtration parameter. From an algorithmic perspective, there are three main tasks connected to persistence [AJ05]:

- **Persistence**: Find efficient algorithms for computing persistence over arbitrary coefficients.

- **Topological Simplification**: Find algorithms for simplifying topology based on persistence. This means attributes should be sorted into noise and features within the filtration via increasing persistence.

- **Cycles and Manifolds**: This problem statement includes finding algorithms for computing representations. The persistence algorithm is designed to track the subspcaes that express nontrivial topological attributes and can be modified to identify these subspaces (cycles) as well as subspaces that eliminate them (manifolds).

Definition 2.2.1 (Persistent Homology) *Let $X = \{X_r | r \in \mathbb{R}\}$ be a nested sequence of spaces parametrized by the real numbers. The k-th persistent homology is defined as:*

$$PH_k(X) = \Pi H_k(X_r). \tag{2.1}$$

The product on the right of Eq. (2.1) formally is an uncountable dimensional vector space.

CHAPTER 3

MACHINE LEARNING AND AUTOENCODERS

BIBLIOGRAPHY

[AAD07] Nina Amenta, Dominique Attali, and Olivier Devillers. “Complexity of Delaunay Triangulation for Points on Lower-Dimensional Polyhedra”. In: SODA ’07. New Orleans, Louisiana: Society for Industrial and Applied Mathematics, 2007, pp. 1106–1113. ISBN: 9780898716245.

[Ada+15] Henry Adams et al. “Persistence Images: A Stable Vector Representation of Persistent Homology”. In: (2015). DOI: 10.48550/ARXIV.1507.06217. URL: https://arxiv.org/abs/1507.06217.

[Ada23] Adams, Henry. *What applied software options do I have?* https://www.math.colostate.edu/~adams/advising/appliedTopologySoftware/, Last accessed on 2023-03-10. 2023.

[AGR17] Nieves Atienza, Rocio Gonzalez-Diaz, and Matteo Rucco. *Persistent Entropy for Separating Topological Features from Noise in Vietoris-Rips Complexes*. 2017. arXiv: 1701.07857 [cs.OH].

[AGS20] Nieves Atienza, Rocio Gonzalez-Dıaz, and Manuel Soriano-Trigueros. “On the stability of persistent entropy and new summary functions for topological data analysis”. In: *Pattern Recognition* 107 (Nov. 2020), p. 107509. DOI: 10.1016/j.patcog.2020.107509. URL: https://doi.org/10.1016%2Fj.patcog.2020.107509.

[AJ05] Zomorodian A.J. *Topology for computing.* CUP, 2005. ISBN: 0511082207. URL: http://gen.lib.rus.ec/book/index.php?md5=bab116c86862a5ed75867e974798da98.

[Ale28] P. Alexandroff. “Über den allgemeinen Dimensionsbegriff und seine Beziehungen zur elementaren geometrischen Anschauung”. In: *Mathematische Annalen* 98 (1928), pp. 617–635. URL: http://eudml.org/doc/159232.

[ALS13] Dominique Attali, André Lieutier, and David Salinas. "Vietoris–Rips complexes also provide topologically correct reconstructions of sampled shapes". In: *Computational Geometry* 46.4 (2013). 27th Annual Symposium on Computational Geometry (SoCG 2011), pp. 448–465. ISSN: 0925-7721. DOI: https://doi.org/10.1016/j.comgeo.2012.02.009. URL: https://www.sciencedirect.com/science/article/pii/S0925772112001423.

[Ark10] Martin Arkowitz. *Introduction to Homotopy Theory*. Springer, 2010.

[Bau21] Ulrich Bauer. "Ripser: efficient computation of Vietoris-Rips persistence barcodes". In: *J. Appl. Comput. Topol.* 5.3 (2021), pp. 391–423. ISSN: 2367-1726. DOI: 10.1007/s41468-021-00071-5. URL: https://doi.org/10.1007/s41468-021-00071-5.

[BDM13] Jean-Daniel Boissonnat, Tamal K. Dey, and Clément Maria. "The Compressed Annotation Matrix: an Efficient Data Structure for Computing Persistent Cohomology". In: (2013). DOI: 10.48550/ARXIV.1304.6813. URL: https://arxiv.org/abs/1304.6813.

[Bjö03] Anders Björner. "Nerves, fibers and homotopy groups". In: *Journal of Combinatorial Theory, Series A* 102.1 (2003), pp. 88–93. ISSN: 0097-3165. DOI: https://doi.org/10.1016/S0097-3165(03)00015-3. URL: https://www.sciencedirect.com/science/article/pii/S0097316503000153.

[BM06] Jonathan Ariel Barmak and Elias Gabriel Minian. *Simple Homotopy Types and Finite Spaces*. 2006. DOI: 10.48550/ARXIV.MATH/0611158. URL: https://arxiv.org/abs/math/0611158.

[BO06] Jean-Daniel Boissonnat and Steve Oudot. "Provably Good Sampling and Meshing of Lipschitz Surfaces". In: *Proceedings of the Twenty-Second Annual Symposium on Computational Geometry*. SCG '06. Sedona, Arizona, USA: Association for Computing Machinery, 2006, pp. 337–346. ISBN: 1595933409. DOI: 10.1145/1137856.1137906. URL: https://doi.org/10.1145/1137856.1137906.

[Bor48] Karol Borsuk. "On the imbedding of systems of compacta in simplicial complexes". In: *Fundamenta Mathematicae* 35 (1948), pp. 217–234.

[BP20] Jean-Daniel Boissonnat and Siddharth Pritam. "Edge Collapse and Persistence of Flag Complexes". In: *SoCG 2020 - 36th International Symposium on Computational Geometry*. Zurich, Switzerland, June 2020. DOI: 10.4230/LIPIcs.SoCG.2020.19. URL: https://hal.inria.fr/hal-02873740.

[Bub20] Peter Bubenik. "The Persistence Landscape and Some of Its Properties". In: *Topological Data Analysis*. Springer International Publishing, 2020, pp. 97–117. DOI: 10.1007/978-3-030-43408-3_4. URL: https://doi.org/10.1007%2F978-3-030-43408-3_4.

[Bub23] Bubenik, Peter. *Software*. `https://people.clas.ufl.edu/peterbubenik/software/`, Last accessed on 2023-03-10. 2023.

[Buc+14] Mickael Buchet et al. *Efficient and Robust Persistent Homology for Measures*. 2014. arXiv: `1306.0039 [cs.CG]`.

[Car+19] Mathieu Carrière et al. *PersLay: A Neural Network Layer for Persistence Diagrams and New Graph Topological Signatures*. 2019. DOI: `10.48550/ARXIV.1904.09378`. URL: `https://arxiv.org/abs/1904.09378`.

[Car09] Gunnar Carlsson. "Topology and Data". In: *Bulletin of The American Mathematical Society - BULL AMER MATH SOC* 46 (Apr. 2009), pp. 255–308. DOI: `10.1090/S0273-0979-09-01249-X`.

[Car14] Gunnar Carlsson. "Topological pattern recognition for point cloud data". In: *Acta Numerica* 23 (May 2014), pp. 289–368. DOI: `10.1017/S0962492914000051`.

[CCO17] Mathieu Carrière, Marco Cuturi, and Steve Oudot. *Sliced Wasserstein Kernel for Persistence Diagrams*. 2017. arXiv: `1706.03358 [cs.CG]`.

[CD19] Frédéric Chazal and Vincent Divol. *The density of expected persistence diagrams and its kernel based estimation*. 2019. arXiv: `1802.10457 [cs.CG]`.

[CEH05] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. "Stability of Persistence Diagrams". In: vol. 37. June 2005, pp. 263–271. DOI: `10.1007/s00454-006-1276-5`.

[CG06] Frédéric Cazals and Joachim Giesen. "Delaunay Triangulation Based Surface Reconstruction". In: *Effective Computational Geometry for Curves and Surfaces*. Ed. by Jean-Daniel Boissonnat and Monique Teillaud. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 231–276. ISBN: 978-3-540-33259-6. DOI: `10.1007/978-3-540-33259-6_6`. URL: `https://doi.org/10.1007/978-3-540-33259-6_6`.

[Cha+10] Erin Chambers et al. "Vietoris–Rips Complexes of Planar Point Sets". In: *Discrete and Computational Geometry* 44 (July 2010), pp. 75–90. DOI: `10.1007/s00454-009-9209-8`.

[Cha+13] Frédéric Chazal et al. *Stochastic Convergence of Persistence Landscapes and Silhouettes*. 2013. DOI: `10.48550/ARXIV.1312.0308`. URL: `https://arxiv.org/abs/1312.0308`.

[CK11] Chao Chen and Michael Kerber. "Persistent Homology Computation with a Twist". In: 2011.

[CL07] Frédéric Chazal and André Lieutier. "Stability and Computation of Topological Invariants of Solids in R^n". In: *Discrete Computational Geometry* 37 (May 2007), pp. 601–617. DOI: `10.1007/s00454-007-1309-8`.

[CM17] Frédéric Chazal and Bertrand Michel. "An Introduction to Topological Data Analysis: Fundamental and Practical Aspects for Data Scientists". In: *Frontiers in Artificial Intelligence* 4 (Oct. 2017). DOI: 10.3389/frai.2021.667963.

[CO07] Frédéric Chazal and Steve Oudot. *Towards Persistence-Based Reconstruction in Euclidean Spaces*. 2007. DOI: 10.48550/ARXIV.0712.2638. URL: https://arxiv.org/abs/0712.2638.

[COO15] Mathieu Carrière, Steve Y. Oudot, and Maks Ovsjanikov. "Stable Topological Signatures for Points on 3D Shapes". In: *Computer Graphics Forum* (2015). DOI: 10.1111/cgf.12692.

[Čuf20] Matija Čufar. "Ripserer.jl: flexible and efficient persistent homology computation in Julia". In: *Journal of Open Source Software* 5 (Oct. 2020), p. 2614. DOI: 10.21105/joss.02614.

[DF15] Barbara Di Fabio and Massimo Ferri. *Comparing persistence diagrams through complex vectors*. 2015. DOI: 10.48550/ARXIV.1505.01335. URL: https://arxiv.org/abs/1505.01335.

[DFW12] Tamal K. Dey, Fengtao Fan, and Yusu Wang. *Computing Topological Persistence for Simplicial Maps*. 2012. DOI: 10.48550/ARXIV.1208.5018. URL: https://arxiv.org/abs/1208.5018.

[DFW13] Tamal K. Dey, Fengtao Fan, and Yusu Wang. *Graph Induced Complex on Point Data*. 2013. arXiv: 1304.0662 [cs.CG].

[DFW15] Tamal K. Dey, Fengtao Fan, and Yusu Wang. "Graph induced complex on point data". In: *Computational Geometry* 48.8 (2015), pp. 575–588. ISSN: 0925-7721. DOI: https://doi.org/10.1016/j.comgeo.2015.04.003. URL: https://www.sciencedirect.com/science/article/pii/S0925772115000292.

[DSW16] Tamal K. Dey, Dayu Shi, and Yusu Wang. "SimBa: An Efficient Tool for Approximating Rips-Filtration Persistence via Simplicial Batch-Collapse". en. In: (2016). DOI: 10.4230/LIPICS.ESA.2016.35. URL: http://drops.dagstuhl.de/opus/volltexte/2016/6386/.

[EH10] Herbert; Edelsbrunner and John L. Harer. *Computational Topology. An Introduction.* AMS, 2010.

[ELZ00] H. Edelsbrunner, D. Letscher, and A. Zomorodian. "Topological persistence and simplification". In: *Proceedings 41st Annual Symposium on Foundations of Computer Science*. 2000, pp. 454–463. DOI: 10.1109/SFCS.2000.892133.

[ES45] Samuel Eilenberg and Norman E. Steenrod. "Axiomatic Approach to Homology Theory". In: *Proceedings of the National Academy of Sciences of the United States of America* 31.4 (1945), pp. 117–120. ISSN: 00278424. URL: http://www.jstor.org/stable/87896 (visited on 03/06/2023).

[Ghr14] Robert Ghrist. *Elementary Applied Topology*. 2014. URL: `https://www2.math.upenn.edu/~ghrist/notes.html` (visited on 03/01/2023).

[Gio] Giotto-tda. *Weak Alpha Persistence*. `https://giotto-ai.github.io/gtda-docs/0.3.0/modules/generated/homology/gtda.homology.WeakAlphaPersistence.html`, Last accessed on 2023-04-03.

[GM05] Robert Ghrist and Abubakr Muhammad. "Coverage and hole-detection in sensor networks via homology". In: May 2005, pp. 254–260. ISBN: 0-7803-9201-9. DOI: `10.1109/IPSN.2005.1440933`.

[Hat02] Allen Hatcher. *Algebraic Topology*. Cambridge University Press, 2002.

[Ito87] Kiyosi Ito. *Encyclopedic Dictionary of Mathematics*. 2nd ed. MIT Press, 1987.

[KFH16] Genki Kusano, Kenji Fukumizu, and Yasuaki Hiraoka. *Persistence weighted Gaussian kernel for topological data analysis*. 2016. arXiv: `1601.01741 [math.AT]`.

[KS13] Michael Kerber and R. Sharathkumar. *Approximate Cech Complexes in Low and High Dimensions*. 2013. arXiv: `1307.3272 [cs.CG]`.

[LCB] Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. *THE MNIST DATABASE of handwritten digits*. `http://yann.lecun.com/exdb/mnist/`, Last accessed on 2023-03-16.

[Ler45] J. Leray. *Topologie algebrique.sur la forme des espaces topologiques et sur le points fixes des representations etc ...* Journal de mathematiques pures et appliquees. Gauthıer-Villars, 1945. URL: `https://books.google.de/books?id=E3uMGwAACAAJ`.

[Lue+19] Daniel Luetgehetmann et al. *Computing persistent homology of directed flag complexes*. 2019. DOI: `10.48550/ARXIV.1906.10458`. URL: `https://arxiv.org/abs/1906.10458`.

[LY18] Tam Le and Makoto Yamada. "Persistence Fisher Kernel: A Riemannian Manifold Kernel for Persistence Diagrams". In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. NIPS'18. Montréal, Canada: Curran Associates Inc., 2018, pp. 10028–10039.

[LZ14] Ryan H. Lewis and Afra Zomorodian. *Multicore Homology via Mayer Vietoris*. 2014. arXiv: `1407.2275 [cs.CG]`.

[McC67] Michael C. McCord. "Homotopy Type Comparison of a Space with Complexes Associated with its Open Covers". In: *Proceedings of the American Mathematical Society* 18.4 (1967), pp. 705–708. ISSN: 00029939, 10886826. URL: `http://www.jstor.org/stable/2035443` (visited on 03/03/2023).

[NSW08] Partha Niyogi, Stephen Smale, and Shmuel Weinberger. "Finding the Homology of Submanifolds with High Confidence from Random Samples". In: *Discrete and Computational Geometry* 39 (Mar. 2008), pp. 419–441. DOI: 10.1007/s00454-008-9053-2.

[Ott+17] Nina Otter et al. "A roadmap for the computation of persistent homology". In: *EPJ Data Science* 6.1 (Aug. 2017). DOI: 10.1140/epjds/s13688-017-0109-5. URL: https://doi.org/10.1140%2Fepjds%2Fs13688-017-0109-5.

[Oud15] Steve Oudot. *Persistence Theory: From Quiver Representations to Data Analysis*. AMS, 2015.

[PH13] Jose Perea and John Harer. *Sliding Windows and Persistence: An Application of Topological Methods to Signal Analysis*. 2013. DOI: 10.48550/ARXIV.1307.6188. URL: https://arxiv.org/abs/1307.6188.

[Pri] Siddharth Pritam. *Edge collapse*. https://gudhi.inria.fr/doc/3.4.1/group__edge__collapse.html, Last accessed on 2023-04-03.

[Rei+14] Jan Reininghaus et al. *A Stable Multi-Scale Kernel for Topological Machine Learning*. 2014. DOI: 10.48550/ARXIV.1412.6821. URL: https://arxiv.org/abs/1412.6821.

[Roy+21] Martin Royer et al. "ATOL: Measure Vectorization for Automatic Topologically-Oriented Learning". In: *AISTATS 2021 - 24th International Conference on Artificial Intelligence and Statistics*. The 24th International Conference on Artificial Intelligence and Statistics. Virtual conference, France, Apr. 2021. URL: https://hal.science/hal-02296513.

[SG07] Vin Silva and Robert Ghrist. "Coverage in sensor networks via persistent homology". In: *Algebraic and Geometric Topology* 7 (Apr. 2007). DOI: 10.2140/agt.2007.7.339.

[She13] Donald R. Sheehy. *Linear-Size Approximations to the Vietoris-Rips Filtration*. 2013. arXiv: 1203.6786 [cs.CG].

[SMV11] Vin de Silva, Dmitriy Morozov, and Mikael Vejdemo-Johansson. "Dualities in persistent (co)homology". In: *Inverse Problems* 27.12 (Nov. 2011), p. 124003. DOI: 10.1088/0266-5611/27/12/124003. URL: https://doi.org/10.1088%2F0266-5611%2F27%2F12%2F124003.

[SV09] Vin de Silva and Mikael Vejdemo-Johansson. "Persistent cohomology and circular coordinates". In: *Proceedings of the twenty-fifth annual symposium on Computational geometry*. ACM, June 2009. DOI: 10.1145/1542362.1542406. URL: https://doi.org/10.1145%2F1542362.1542406.

[Tak81] Floris Takens. "Detecting strange attractors in turbulence". In: *Dynamical Systems and Turbulence, Warwick 1980*. Ed. by David Rand and Lai-Sang Young. Berlin, Heidelberg: Springer Berlin Heidelberg, 1981, pp. 366–381. ISBN: 978-3-540-38945-3.

[Tau+20] Guillaume Tauzin et al. "giotto-tda: A Topological Data Analysis Toolkit for Machine Learning and Data Exploration". In: (2020). DOI: `10.48550/ARXIV.2004.02551`. URL: `https://arxiv.org/abs/2004.02551`.

[TP21] Álvaro Torras and Ulrich Pennig. *Interleaving Mayer-Vietoris spectral sequences*. 2021. arXiv: `2105.03307 [math.AT]`.

[TSB18] Christopher Tralie, Nathaniel Saul, and Rann Bar-On. "Ripser.py: A Lean Persistent Homology Library for Python". In: *Journal of Open Source Software* 3.29 (2018), p. 925. DOI: `10.21105/joss.00925`. URL: `https://doi.org/10.21105/joss.00925`.

[Wei52] André Weil. "Sur les théorèmes de de Rham." In: *Commentarii mathematici Helvetici* 26 (1952), pp. 119–145. URL: `http://eudml.org/doc/139040`.

[XRV17] Han Xiao, Kashif Rasul, and Roland Vollgraf. *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*. Aug. 28, 2017. arXiv: `cs.LG/1708.07747 [cs.LG]`.

[ZC05] Afra Zomorodian and Gunnar Carlsson. "Computing Persistent Homology". In: *Discrete and Computational Geometry* 33 (Feb. 2005), pp. 249–274. DOI: `10.1007/s00454-004-1146-y`.

[ZC08] Afra Zomorodian and Gunnar Carlsson. "Localized homology". In: *Computational Geometry* 41.3 (2008), pp. 126–148. ISSN: 0925-7721. DOI: `https://doi.org/10.1016/j.comgeo.2008.02.003`. URL: `https://www.sciencedirect.com/science/article/pii/S0925772108000187`.

[ZXW20] Simon Zhang, Mengbai Xiao, and Hao Wang. *GPU-Accelerated Computation of Vietoris-Rips Persistence Barcodes*. 2020. DOI: `10.48550/ARXIV.2003.07989`. URL: `https://arxiv.org/abs/2003.07989`.