Introduction to Topological Data Analysis
Lecture Notes FS 2025

Patrick Schnider <patrick.schnider@inf.ethz.ch>
Lucas Slot <lucas.slot@inf.ethz.ch>
Simon Weber <simon.weber@inf.ethz.ch>

May 23, 2025

Preface

These lecture notes are designed to accompany a course "Introduction to topological data analysis" taught at the Department of Computer Science, ETH Zürich, since 2023. The course is intended for students with a background in computer science or data science. It requires knowledge of linear algebra, but does not assume any previous experience with topology.

The course can be roughly divided into four parts. In the first part, we go over the necessary mathematical foundations, in particular concepts from algebraic topology such as *homology*. In the second part, we study the persistent homology pipeline. In the third part, we discuss Reeb graphs and the Mapper algorithm. Finally, the fourth part contains other applications of topology in computer and data science, as well as applications of topological data analysis to other fields. At the end of each chapter there is a list of questions that students are expected to be able to answer in the oral exam.

In the current setting, the course runs over 14 weeks, with three hours of lectures and two hours of exercises each week. In addition, there are two sets of graded homeworks which students have to hand in spread over the course.

These notes are an extended version of the scribe notes written by Simon Weber from the first iteration of the course. We have tried our best to avoid mistakes, but experience tells that there will be many that escape our detection. So in case you notice some problem, please let us know, regardless of whether it is a minor typo or punctuation error, a glitch in formulation, or a hole in an argument. This way the issue can be fixed for the next edition and future readers profit from your findings.

We thank Anton Künzi and Johann Wenckstern for their helpful contributions.

Patrick Schnider, Lucas Slot and Simon Weber
Institute of Theoretical Computer Science
Department of Computer Science, ETH Zürich
Andreasstrasse 5, CH-8050 Zürich, Switzerland
E-mail address: {patrick.schnider, lucas.slot, simon.weber}@inf.ethz.ch

Contents

Chapter 1

Introduction

In many applications in data science, the data is given to us as a point cloud in some (potentially high-dimensional) space. Interesting data about which we can actually make statements using tools from data analysis usually has some underlying shape, and this shape conveys information: think about points sampled from a sphere or a torus. In both cases the point cloud will "look like" a 2-dimensional object, but the objects look different. In order to describe and compare different shapes, in particular in higher dimensions, we need some mathematical language. Luckily this language has already been developed under the name of *topology*. In topological data analysis we use the classical tools and language from topology to detect and describe the notion of "shape" in data sets. The main tool we use for this is *homology*, which can be regarded as "counting holes". Let us illustrate this with some toy examples.
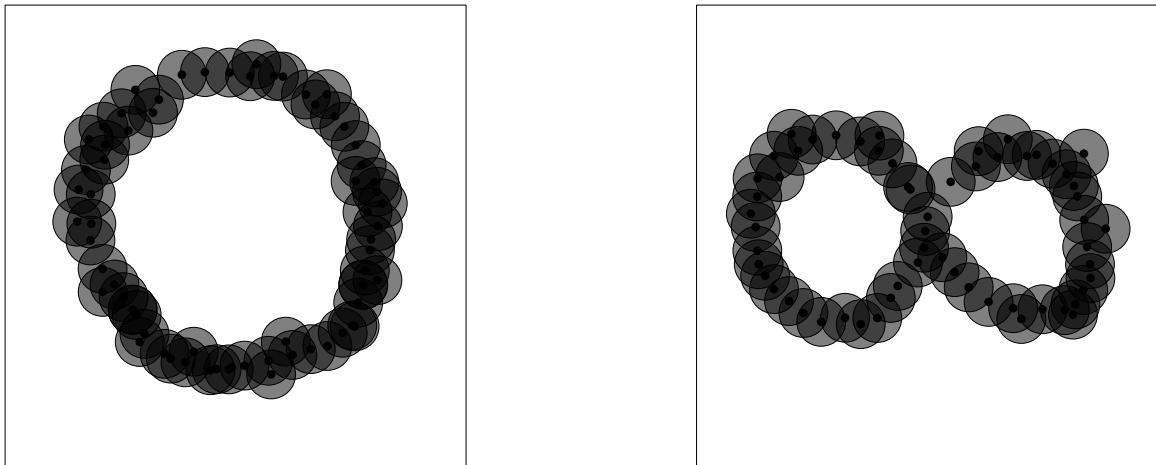
Figure 1.1: *A circle (left) and a Figure-8 (right)*

Consider the two shapes in Figure 1.1. On the left, you see a circle, on the right a Figure-8, both in $\mathbb{R}^2$. If you had to count how many "holes" these shapes have, you would probably argue along the following lines: the circle has one hole, as removing it from $\mathbb{R}^2$ we are left with one bounded connected component. Similarly, the Figure-8 has two holes. In particular, you would say that the two shapes are "different", as they have a different number of holes. This type of intuition is indeed correct, and topology provides us with the language to make this mathematically precise.

Figure 1.2: *Points sampled from a circle (left) and a Figure-8 (right).*
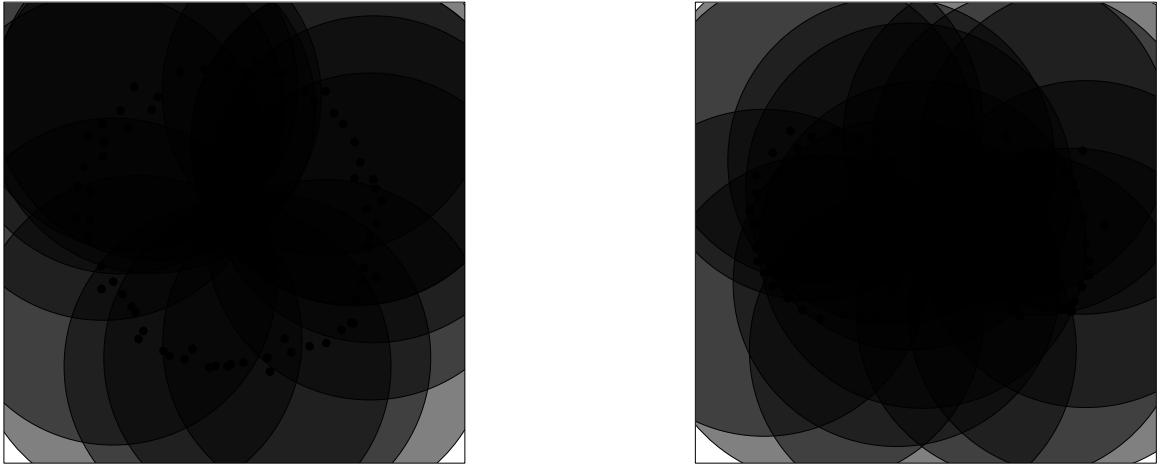
Let us now look at the 2-dimensional data sets in Figure 1.2. For a human, it is immediately visible that the data sets are (noisy) samples of the shapes of Figure 1.1. How do we get a computer to "see" this? Note that we cannot count the number of holes as we did above: as we are just given finitely many points, there are no holes in both data sets. However, by squinting our eyes a bit, we clearly see the different shapes from which we sampled. A mathematical analogue of squinting one's eyes could be to enlarge the points to disks with some small radius. This is depicted in Figure 1.3.

Figure 1.3: *Enlarging the points to disks shows the shape.*

Indeed, choosing a large enough radius, by considering the union of the disks we again get shapes that look like the two shapes we sampled from. In particular, on the left we again have one hole whereas on the right we have two holes. Unfortunately, it is not at all clear what radius we should choose. Indeed, by choosing a radius that is too small, we might not create all the holes of the original shape. On the other hand, by

choosing a radius that is too large, we might fill in some of the holes[1]. This is depicted in Figure 1.4.

Figure 1.4: *Enlarging the disks too much, we lose the shapes again.*
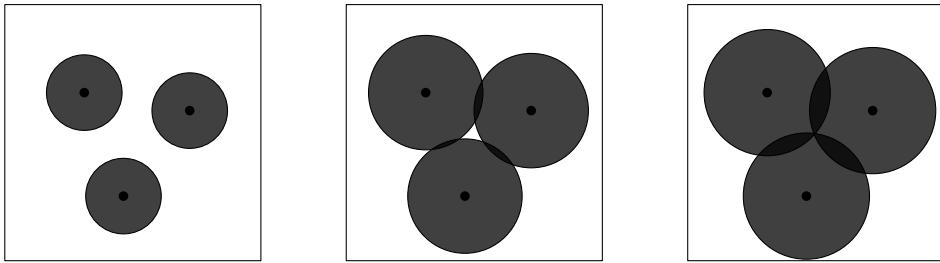
Of course we could now try to somehow estimate a good radius. However, this is quite a difficult task in general. The key idea of *persistent homology* is to continuously grow the disks and keep track of the number of holes and how long they live. During this process of growing the disks, many holes will be created. However, many of them are filled in shortly after they are created, see Figure 1.5 for an example. In persistent homology, we keep track of all holes created in the process, together with timestamps of when they are created and when they are filled in. This gives a *lifetime* for each hole. The intuition behind this is that holes with a short lifetime are just a result of the process, whereas holes with a long lifetime convey information about the shape of the underlying data.

Let us consider yet another point cloud, this time sampled from two nested circles, see Figure 1.6. Two circles give us two holes, so in the process of growing disks we expect to see two holes with a long lifetime.
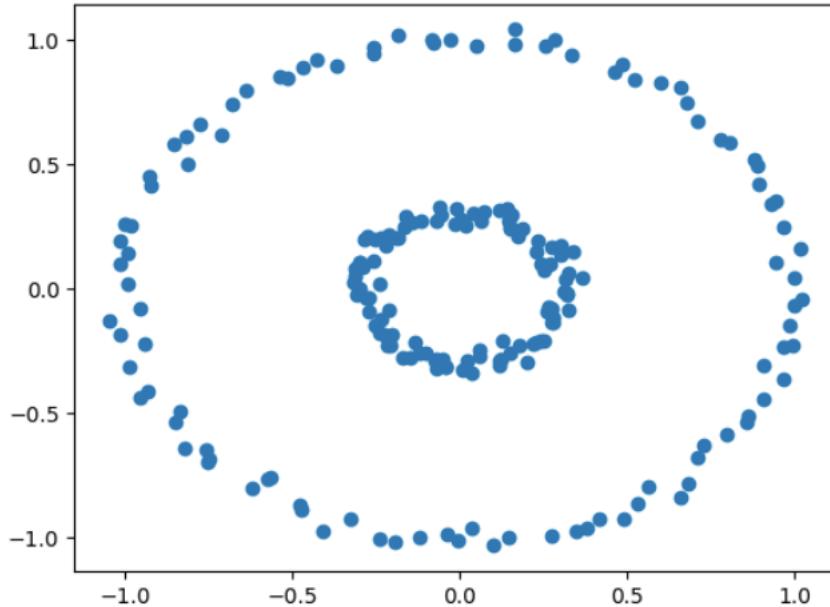
One way to visualize the lifetimes of holes is through *barcodes*: for each hole, we draw an interval whose startpoint and endpoint correspond to the time of creation of the hole, and when it got filled in. Doing this for the point cloud in Figure 1.6, we get the barcodes depicted in Figure 1.7. Indeed, while there are many intervals, only two of them are long, implying that there are two holes inherent to the underlying data, which agrees with the fact that the points were sampled from two circles.

---

[1] The same is true for squinting your eyes: if you squint them too much, you close them and do not see anything...
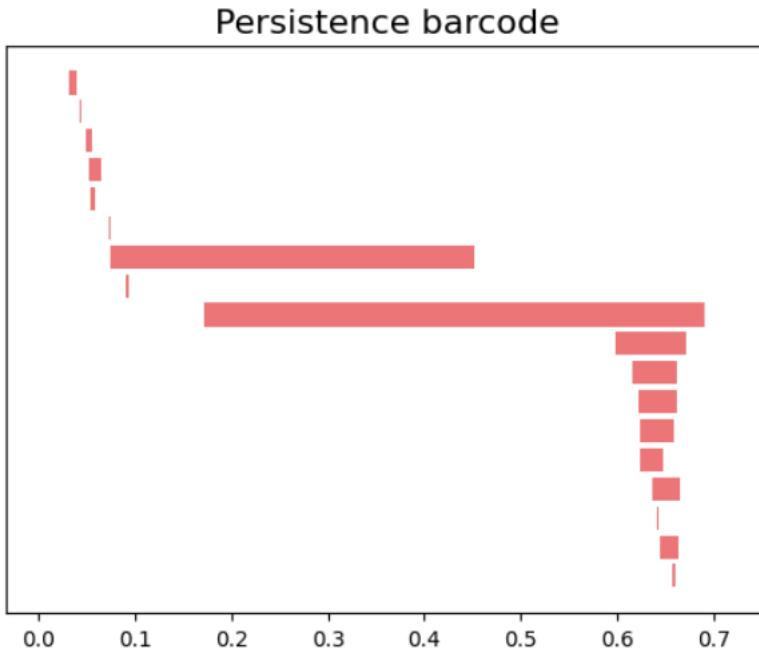
Figure 1.5: *In the growing disks process, many holes get filled in shortly after they are created.*

Figure 1.6: *Points sampled from two nested circles.*

The main work we will do in these lecture notes is to formalize this process of growing disks and keeping track of holes, as we sketched above. For this we first introduce some essential background of homology theory in Chapter 3. This requires some mathematical background that goes above linear algebra. This background will be introduced in Chapter 2. The growing disks process is modeled via nested *simplicial complexes*, and there are several different such complexes that can be defined. Some of these are discussed in Chapter 5. Keeping track of holes created and filled in is done via the theory of *persistent homology*, which we introduce in Chapter 4. In this chapter we also discuss algorithms to compute persistent homology. Once we have computed persistent

Persistence barcode

Figure 1.7: *The barcodes from two nested circles.*

homology and have the output, for example in the form of barcodes, we might want to compare different such outputs with each other. For this, there are several distance measures, which we discuss in Chapter 6. There we also mathematically prove stability results stating that if the data is perturbed only a little, then also the output cannot change too much.

Persistent homology is not the only application of topology to the analysis of data. Another widely used tool in topological data analysis is *Mapper*, which we discuss in Chapter 7. Finally, we discuss the computational problem of finding nice representatives of holes in Chapter 8 as well as persistence over multiple parameters in Chapter 9.

These lecture notes focus a lot on the mathematical theory behind topological data analysis. Topological data analysis has however seen many successful applications in recent years. We highlight some of them in Chapter 10. There are also powerful programming libraries available that implement many of the concepts discussed in these lecture notes. For the coding aspects of the topics in these lecture notes there are interactive notebooks on the course webpage.

Chapter 2

Mathematical Foundations

2.1 Topological Spaces

Topology, sometimes also called "rubber-sheet geometry", stems from the Greek words *tópos*, which means place or locality, and *lógos*, which means study. So, it can be roughly translated as the study of places and shapes. Indeed, as the name rubber-sheet geometry suggests, topology studies similar objects as geometry, but in a setting where properties are preserved under continuous deformations like stretching and twisting. In particular, these properties should be independent of metrics, but we would still like to have ways to describe proximity between points. We do this by looking at open neighborhoods of points. The core objects in topology are *topological spaces*, whose definition captures the system of open neighborhoods of the points in the space.

Definition 2.1. *A topological space* (X, T) *is a set of points* X, *with a system* T *of subsets of* X *(called the topology on* X*), such that*

1. $\emptyset \in T$, $X \in T$.

2. For every $S \subseteq T$, $\bigcup S \in T$.

3. For every **finite** $S \subseteq T$, $\bigcap S \in T$.

The sets in T *are called the* open *sets of* X.

    For example, setting $X = \mathbb{R}^2$ and T to be the collection of open subsets (in the geometric/calculus sense) of $\mathbb{R}^2$, we can check that (X, T) is a topological space. A further example of a topological space is $(X, 2^X)$, where 2^X denotes the family of all subsets of X. This is called a *discrete topology*.

    Another example is the Euclidean space $X = \mathbb{R}^d$, where the open sets T are defined as we know from calculus. This example also shows why we restrict the third condition of the definition above only to *finite* intersections of open sets: If we allowed infinite intersections in Condition 3, a set $\{p\}$ consisting of a single point $p \in \mathbb{R}^d$ (which by the

calculus definition is not an open set) would have to be considered to be open; it is the intersection of the infinite series of open balls of radius $1/n$ centered at p, for $n \in \mathbb{N}$.

In most applications in these lecture notes, we work with subspaces of this Euclidean space $\mathbb{R}^d$. In that context we not only know the notion of open sets from calculus, but also notions such as *closed sets, closure, interior* and *boundary*. These terms can also be defined for abstract topological spaces:

Definition 2.2. *A set $Q \subseteq X$ is called* closed, *if its complement $X \setminus Q$ is open. The* closure $\mathrm{cl}\, Q$ *is the smallest closed set containing Q. The* interior $\mathrm{int}\, Q$ *is the union of all open subsets of Q. The* boundary $\mathrm{bnd}\, Q$ *is the set minus its interior:* $\mathrm{bnd}\, Q = Q \setminus \mathrm{int}\, Q$.

Note that sets can be open and closed simultaneously: in every topological space (X, T), $\emptyset$ and X are such examples. In a discrete topology, every subset $S \subseteq X$ is both open and closed.

Exercise 2.3. *Show that a finite union of closed sets is closed.*

So far we have only seen two topological spaces: Euclidean space, and the (rather boring) discrete topology on any set X. In order to see the value in the abstractions we are doing, we would like to have more examples of topological spaces. In particular, it would be great if we had a way to get new topological spaces from known ones. In the following we discuss some ways to do this, starting with taking intersections.

Lemma 2.4. *Let (X, T) be some topological space, and $Y \subseteq X$. Then, $U := \{A \cap Y \mid A \in T\}$ is a topology on Y. We call this a* subspace topology.

Proof. We check the three conditions of a topology:

1. $\emptyset = \emptyset \cap Y$, therefore $\emptyset \in U$. Similarly, $Y = X \cap Y$, and thus $Y \in U$.

2. $\bigcup_{i \in I}(A_i \cap Y) = (\bigcup_{i \in I} A_i) \cap Y$, and thus $\bigcup_{i \in I}(A_i \cap Y) \in U$.

3. $\bigcap_{i=1}^{n}(A_i \cap Y) = (\bigcap_{i=1}^{n} A_i) \cap Y$, and thus $\bigcap_{i=1}^{n}(A_i \cap Y) \in U$. $\square$

Since we have seen a natural topology on $\mathbb{R}^d$, this already gives us a natural topology for all subsets of $\mathbb{R}^d$.

Another way to get topological spaces is as a product of spaces. We will not discuss the details of this here, and refer the interested reader to any textbook on topology, such as the excellent book by Munkres [2].

Fact 2.5. *Let $(X, T_X), (Y, T_Y)$ be two topological spaces. Then there exists a topology on $X \times Y$, called the* product topology.

Finally, we can also get a topological space by taking the union of two disjoint topological spaces. If a space can be obtained as such a union, we call it disconnected:

Definition 2.6. *A topological space* (X, T) *is* disconnected, *if there are two disjoint non-empty open sets* $U, V \in T$, *such that* $X = U \cup V$. *A topological space is* connected, *if it is not disconnected.*

Exercise 2.7. *In this exercise, we will use topology to prove that the set of primes is infinite.*

  We define the sets $S(a, b)$ *as follows:*

$$S(a, b) := \{an + b \mid n \in \mathbb{Z}\}, \ \forall a \in \mathbb{Z} \setminus \{0\}, b \in \mathbb{Z}$$

We then say that a set $U \subseteq \mathbb{Z}$ *is open, if and only if for all* $x \in U$, *there exists* $a \in \mathbb{Z}$ *such that* $S(a, x) \subseteq U$. *This is equivalent to saying that every open set* U *is a union of zero or more (including infinitely many) sets* $S(a, b)$.

 (a) Show that this defines a topology on $\mathbb{Z}$.

 (b) Let $A \subset \mathbb{Z}$ *be finite and non-empty. Show that* $\mathbb{Z} \setminus A$ *cannot be closed.*

 (c) Show that $S(a, b)$ *is both open and closed.*

 (d) Show that

$$\bigcup_{p \ prime} S(p, 0) = \mathbb{Z} \setminus \{-1, 1\}$$

 (e) Conclude that there are infinitely many primes.

2.2   Metric Spaces

Recall that topological spaces should capture neighborhoods of points without requiring the notion of a distance. However, if we do have distances, we should still be able to use the framework of topological spaces. In other words, topological spaces should be a generalization of spaces with distances.

Definition 2.8. *A* metric space (X, d) *is a set* X *of points and a distance function* $d : X \times X \to \mathbb{R}$ *satisfying*

 1. $d(p, q) = 0$ *if and only if* $p = q$.

 2. $d(p, q) = d(q, p)$, $\forall p, q \in X$.   *(Symmetry)*

 3. $d(p, q) \leqslant d(p, s) + d(s, q)$, $\forall p, q, s \in X$.   *(Triangle inequality)*

  Note that these three conditions imply that $d(p, q) \geqslant 0$ for all $p, q \in X$: If some distance $d(p, q)$ would be negative, we would have $0 = d(p, p) \leqslant d(p, q) + d(q, p) = 2 \cdot d(p, q) < 0$, a contradiction.

Fact 2.9. *Every metric space has a topology (the* metric space topology*) given by the open metric balls* $B(c, r) = \{p \in X \mid d(p, c) < r\}$ *and their unions.*

13

2.3 Maps Between Topological Spaces

In most areas of mathematics, there are two things that are at the core of every theory: objects, and mappings between them. For example, in linear algebra we study vector spaces and the linear maps between then. Now that we have defined the objects of study — topological spaces — we want to look at the mappings between them.

Definition 2.10. *A function* $f : X \to Y$ *is* continuous *if for every open set* $U \subseteq Y$, *its pre-image* $f^{-1}(U) \subseteq X$ *(the set of all elements* $x \in X$ *such that* $f(x) \in U$*) is open. Continuous functions are also called* maps*. If* f *is an injective map, it is called an* embedding*.*

Let us give some examples:

- For $X \subseteq Y$, we write $X \hookrightarrow Y$ for the function $f(x) = x, \forall x \in X$. This function, which is also called the *inclusion map*, is continuous: $f^{-1}(U) = U \cap X$, which is open in the subspace topology on X.

- For a function $f : \mathbb{R} \to \mathbb{R}$, continuity agrees with the "ϵ-δ" definition of continuity from calculus.

Exercise 2.11. *A topological space* (X, T) *is called* path-connected *if any two points* $x, y \in X$ *can be joined by a path, i.e., there exists a map* $f : [0, 1] \to X$ *of the segment* $[0, 1] \subset \mathbb{R}$ *onto* X *such that* $f(0) = x$ *and* $f(1) = y$. *Prove that a path-connected space is connected.*

An important question we have to answer is when we want to consider two topological spaces to be "the same". In the rest of this section we develop some notions of equivalence of topological spaces, each based on the existence of some continuous function(s).

Definition 2.12. *A* homeomorphism *is a bijective map* $f : X \to Y$ *whose inverse is also continuous. Two topological spaces are* homeomorphic, *if there is a homeomorphism between them. We also write* $X \simeq Y$ *to say that* X, Y *are homeomorphic.*

To make sure that homeomorphism is a reasonable notion of equivalence, we should check that it is indeed an equivalence relation.

Exercise 2.13. *Show that the relation of being homeomorphic is an equivalence relation, that is, show that every space is homeomorphic to itself, show that the relation is symmetric (*$X \simeq Y$ *iff* $Y \simeq X$*), and show that* $\simeq$ *is transitive (if* $X \simeq Y$ *and* $Y \simeq Z$, *then* $X \simeq Z$*).*

Let us apply our definition to some examples, to see whether it captures our intuition:

- The boundary of a tetrahedron is homeomorphic to the sphere S^2 (with both spaces considered as a subspace of $\mathbb{R}^3$). A homeomorphism can be found by taking a point c in the interior of the tetrahedron, and sending each point p of the boundary to the point $f(p)$ on the ray from c through p such that $d(c, f(p)) = 1$.

- The open interval $I := (-1, 1)$ is homeomorphic to $\mathbb{R}$. The following map f is a homeomorphism: $f : I \to \mathbb{R}$, $x \mapsto \frac{x}{1-|x|}$. Its inverse is $f^{-1} : \mathbb{R} \to I$, $y \mapsto \frac{y}{1+|y|}$.

- All knots (a knot is the image of an embedding of the circle into $\mathbb{R}^3$) are homeomorphic. Thus, we cannot distinguish between knots using only homeomorphism.
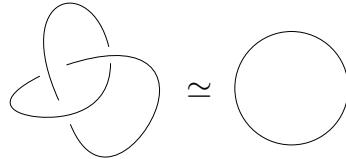
Figure 2.1: *Two knots.*

Exercise 2.14. *Give an example of a map* $f : X \to Y$ *that is bijective but not a homeomorphism.*

Exercise 2.15. *Consider a grid of 2 vertical line segments and* $k + 2$ *horizontal segments, for some* $k \geqslant 0$. *For* $k = 1$, *this looks as follows:*

Now, we consider the problem of placing a point on each of the $k + 2$ *horizontal line segments, such that each of the* $k + 4$ *total line segments contains at least one point.*

 (a) How could one define a topology on the set of all such point placements?

 (b) Convince yourself that this space is homeomorphic to S^k.
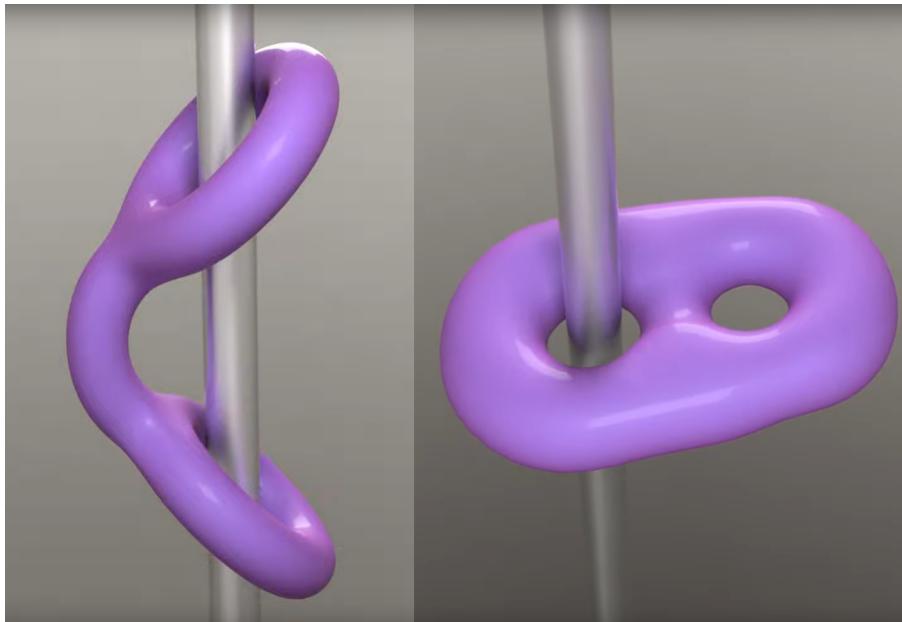
The example of the knots shows that in certain cases, homeomorphism does not capture all the information we can use to distinguish two spaces. In this example, this distinguishing information is not really stored in the topological spaces themselves, but in the way they are embedded in the "ambient" space (in this example $\mathbb{R}^3$). In such a case, where the two spaces we consider are both embedded into the same ambient space, we can not only look at maps between the two spaces themselves, but we can also consider whether one of them can be continuously deformed into the other:

Definition 2.16 ([1, Def. 1.18]). *An* isotopy *connecting* $X \subseteq A$ *and* $Y \subseteq A$ *is a continuous map* $\phi : X \times [0, 1] \to A$, *such that* $\phi(X, 0) = X$, $\phi(X, 1) = Y$, *and* $\forall t \in [0, 1]$, $\phi(\cdot, t)$ *is a homeomorphism between* X *and its image. Two spaces are called* isotopic, *if there is an isotopy connecting them.*

Exercise 2.17. *Show that the relation of being isotopic is an equivalence relation.*

Let us check isotopy on a few examples, starting with the knots from above:

- The two knots from Figure 2.1 above (embedded in $A = \mathbb{R}^3$) are homeomorphic but not isotopic. Isotopy thus captures our intuition more accurately than homeomorphism in this case.

- Let $X \subset \mathbb{R}$ be the union of $\{0\}$, and $[1, 2]$, and let $Y \subset \mathbb{R}$ be the union of $[0, 1]$ and $\{2\}$. These spaces are homeomorphic ($X \simeq Y$), but not isotopic. Just as with the knots, the difference between these spaces does not lie in their topology, but in the way they are embedded into the ambient space $\mathbb{R}$.

- Consider the two spaces in Figure 2.2, which are considered to be embedded in the ambient space A consisting of $\mathbb{R}^3$ minus the grey infinitely long pole in the middle. Do you think the spaces are isotopic? Most people would probably argue that they are not, as in the left space both loops of the handcuff are locked around pole while in the right space one loop is free. However, it turns out that the spaces are in fact isotopic. An isotopy is illustrated by the following video: https://www.youtube.com/watch?v=wDZx9B4TAXo

Figure 2.2: *Left: Both loops of the handcuffs are wrapped around an infinite pole. Right: Only one loop of the handcuffs is wrapped around the infinite pole. These spaces are isotopic.*

Using isotopy we have now managed to distinguish between two spaces (embeddings) that homeomorphism could not distinguish. On the other hand, homeomorphism is also very restrictive: For example, any two-dimensional space X (such as the mantle of a cylinder) cannot be homeomorphic to any one-dimensional space Y (such as a circle), simply due to the difference in dimension of X and Y. We thus also want to develop a weaker notion of equivalence than homeomorphism.

To do this, we take the idea of continuous deformations from isotopy, but instead of applying it to deform spaces into each other, we deform maps into each other:

Definition 2.18. *Let* g, h *be maps* $X \to Y$. *A* homotopy *connecting* g *and* h *is a map* $H : X \times [0, 1] \to Y$ *such that* $H(\cdot, 0) = g$ *and* $H(\cdot, 1) = h$. *In this case* g *and* h *are called* homotopic.

Before we use homotopies to define an equivalence on topological spaces, let us again consider some examples:

- The inclusion map $g : \mathbb{B}^3 \hookrightarrow \mathbb{R}^3$ (where $\mathbb{B}^3$ is the unit ball in $\mathbb{R}^3$), and the constant map $h : \mathbb{B}^3 \to \mathbb{R}^3$ which sends every point to the origin, are homotopic, as shown by the homotopy
$$H(x, t) = (1 - t)g(x).$$

- The identity map $g : S^1 \to S^1$, and the constant map $h : S^1 \to S^1$ which sends everything to a single point $p \in S^1$, are *not* homotopic.

The notion of homotopy now allows us to define our desired equivalence relation on topological spaces that is weaker than homeomorphism. Intuitively, this relation says that two spaces are the same if they can be continuously transformed into each other not only by bending, twisting and stretching, but also by shrinking or blowing up parts of different dimensions. However, note that unlike with isotopy, we do not need to consider the two spaces to be embedded in any ambient space.

Definition 2.19. *Two spaces* X, Y *are* homotopy equivalent *if there exist maps* $g : X \to Y$ *and* $h : Y \to X$ *such that:*

- $h \circ g$ *is homotopic to* id_X *(the identity map* $x \mapsto x$*), and*

- $g \circ h$ *is homotopic to* id_Y.

Exercise 2.20. *Show that the relation of being homotopy equivalent is an equivalence relation.*

Let us consider some examples:

- The circle S^1 and $\mathbb{R}^2 \setminus \{0\}$ are homotopy equivalent: We pick g as the inclusion map $S^1 \hookrightarrow \mathbb{R}^2 \setminus \{0\}$, and $h(x) := \frac{x}{|x|}$. We see that $h \circ g(x) = x$, i.e., $h \circ g = \mathrm{id}_{S^1}$. Furthermore, $g \circ h(x) = h(x)$. Finally, $g \circ h$ and $\mathrm{id}_{\mathbb{R}^2 \setminus \{0\}}$ are homotopic as certified by the homotopy $H(x, t) := tx + (1 - t)h(x)$.

- The cylinder mantle and the circle are homotopy equivalent, but not homeomorphic.

- Any ball $\mathbb{B}^d$ is homotopy equivalent to the single point. We call such spaces *contractible*.

The next lemma shows that homotopy equivalence is a strictly weaker notion than homeomorphism:

Lemma 2.21. *If X and Y are homeomorphic, they are also homotopy equivalent.*

Proof. Let $g : X \to Y$ be the homeomorphism, and $h := g^{-1}$ its inverse. Then $g \circ h = \mathrm{id}_Y$ and $h \circ g = \mathrm{id}_X$, and id is homotopic to itself. $\qquad\square$

With the need for two maps and a proof that they are homotopic, proving homotopy equivalence directly can be quite tedious. The following notion of *deformation retracts* gives an easy way of proving homotopy equivalence in some cases.

Definition 2.22. *Let $A \subseteq X$. A* deformation retract *of X onto A is a map $R : X \times [0, 1] \to X$, such that*

- $R(\cdot, 0) = \mathrm{id}_X$

- $R(x, 1) \in A, \forall x \in X$

- $R(a, t) = a, \forall a \in A, t \in [0, 1]$

If such a deformation retract of X onto A exists, we also say that A is a deformation retract *of X.*

The intuition behind a deformation retract is that the map R continuously shrinks X to A, while leaving A fixed. Note that unlike homeomorphism, isotopy and homotopy equivalence, deformation retracts are inherently asymmetric.

Fact 2.23. *If A is a deformation retract of X, then A and X are homotopy equivalent.*
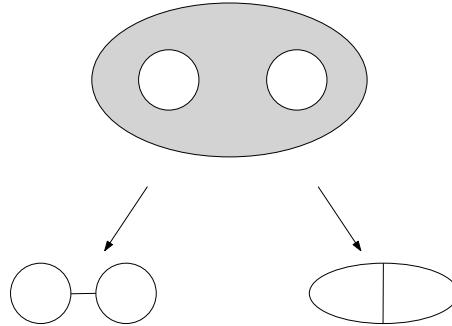
Let us use this to prove homotopy equivalence of some examples:

- The circle S^1 is a deformation retract of $\mathbb{R}^2 \setminus \{0\}$: $R(x, t) = (1 - t)x + t \cdot \frac{x}{|x|}$. Note how much easier this is to prove without needing to use the two maps h and g as above.

- A punctured torus can be deformation retracted onto the symbol 8 where one of the two circles is rotated by $90°$, as seen by the following video: https://www.youtube.com/watch?v=tz3QWrfPQj4

One may think that deformation retracts are only useful for proving homotopy equivalence when one space is a subspace of the other. However, the following fact shows that deformation retracts can prove homotopy equivalence of any pair of spaces:

Fact 2.24. *X, Y are homotopy equivalent if and only if there exists a space Z such that X and Y are deformation retracts of Z.*

An example of this fact can be found in Figure 2.3.
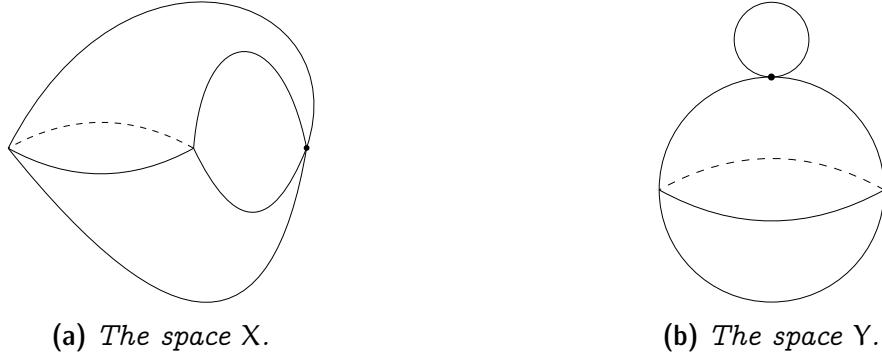
Figure 2.3: *The top space deformation retracts to both spaces below, showing that they are homotopy equivalent.*

Exercise 2.25. *Sort the letters of the alphabet into equivalence classes under homotopy equivalence.*

Exercise 2.26. *Show that both a cylinder and a Möbius strip are homotopy equivalent to a circle.*

Exercise 2.27. *Let X be S^2 where the north pole and the south pole have been glued together, see Figure 2.4a. Let Y be S^2 with an S^1 attached at the north pole, see Figure 2.4b.*

(a) *The space X.*

(b) *The space Y.*

Figure 2.4: *The spaces from Exercise 2.27.*

Give an informal argument that X and Y are homotopy equivalent. Bonus question: Are they also homeomorphic?

We note that in general showing existence of a map with certain properties (e.g., a homeomorphism, isotopy, homotopy) is easy: just give a map and show that it satisfies the required properties. On the other hand, showing that such a map cannot exist is hard, as there are usually infinitely many candidate maps. The idea of algebraic topology is to identify invariant properties preserved by these maps. Then, we know that no map can exist between spaces on which these invariants differ. An example of such an invariant is the number of "holes" a space has, which we will formalize when we introduce the notion of *homology*.

2.4   Algebra

In this section we recap the necessary background in algebra that is needed for the basics of homology theory. Just as in the previous sections, we first introduce the objects of study, followed by the maps between them.

Definition 2.28. *A group* $(G, +)$ *is a set* G *together with a binary operation "+" such that*

    1. $\forall a, b \in G$: $a + b \in G$

    2. $\forall a, b, c \in G$: $(a + b) + c = a + (b + c)$ *(Associativity)*

    3. $\exists 0 \in G$: $a + 0 = 0 + a = a \; \forall a \in G$

    4. $\forall a \in G \exists -a \in G$: $a + (-a) = 0$

$(G, +)$ *is* abelian[1] *if we also have*

    5. $\forall a, b \in G$: $a + b = b + a$ *(Commutativity)*

Let us point out some examples:

- $(\mathbb{Z}, +)$ is a group (even an abelian one), but $(\mathbb{N}, +)$ is not, since any non-zero number does not have an inverse element.

- Consider the (very large) set of all sequences of moves of a Rubik's cube that do not contain a subsequence equivalent to doing nothing. This set forms a group (with the "+" operation being concatenation), but not an abelian one: let L denote moving the left face clockwise, and let U denote moving the upper face clockwise. Replacing "clockwise" by counter-clockwise we get $-L$ and $-U$, respectively. Now, if the group was abelian, then $L + U - L - U$ should give the same configuration again, but if you do these moves on a Rubik's cube, you will see that the configuration has changed.

As groups can be very large, even infinitely large, it can be useful to have a concise way of writing them:

Definition 2.29. *Let* $(G, +)$ *be a group.*
A subset $A \subseteq G$ *is a* generator *if every element of* G *can be written as a finite sum of elements of* A *and their inverses.*
A subset $B \subseteq G$ *is a* basis *if every element of* G *can be* uniquely *written as a finite sum of elements of* B *and their inverses (ignoring trivial cancellations, i.e.,* $a + c + (-c) + (-b) = a + (-b)$*).*
An abelian group that has a basis is called free*.*

---

[1] Note that unlike other mathematical concepts named after a person, *abelian* is usually not capitalized.

Examples:

- The six standard moves of the Rubik's cube (rotating the top, bottom, front, back, left, or right layer clockwise by $90°$) are a generator for the Rubik's cube move sequences.

- $\{1\}$ is a basis of $(\mathbb{Z}, +)$.

Exercise 2.30. *A cyclic group is a group* G *that contains an element* $g \in G$ *such that* $\{g\}$ *is a generator of* G. *Show that every cyclic group is abelian (commutative).*

Exercise 2.31. *Consider a Rubik's cube. Prove that no sequence* X *of elementary moves exists such that every Rubik's cube can be solved by repeatedly applying* X.

Definition 2.32. *For some group* $(G, +)$, $H \subseteq G$ *is a* subgroup, *if* $(H, +)$ *is also a group.*

For example, the even integers (including 0) are a subgroup of $(\mathbb{Z}, +)$. Subgroups are important in group theory, as they can be used to partition a group into several parts:

Definition 2.33. *Let* $H \subseteq G$ *be a subgroup of* $(G, +)$, *and* $a \in G$.
The left coset $a + H$ *is the set* $a + H := \{a + b \mid b \in H\}$, *and the* right coset $H + a := \{b + a \mid b \in H\}$. *If* G *is abelian,* $a + H = H + a$, *and they are simply called the* coset. *For* G *abelian, the* quotient group *of* G *by* H, *denoted by* G/H, *is the group on the set of cosets* $\{a + H, a \in G\}$ *with the operation* $\oplus$ *defined as* $(a+H) \oplus (b+H) = (a+b)+H$, $\forall a, b \in G$.

Examples:

- Let $G = (\mathbb{Z}, +)$ and $H = n\mathbb{Z} = \{n \cdot a \mid a \in \mathbb{Z}\}$. Then, $G/H = \{0 + n\mathbb{Z}, 1 + n\mathbb{Z}, \ldots, (n-1) + n\mathbb{Z}\}$ is the group usually referred to as $\mathbb{Z}_n$, the group of modular arithmetic modulo n.

- $\mathbb{R}/\mathbb{Z}$ is the circle group (the multiplicative group of all complex numbers of absolute value 1). Try and convince yourself of this!

In order to compare groups with each other, we again want a notion of maps between groups, that behave well with the group structures:

Definition 2.34. *A map* $h : G \to H$ *between abelian groups* $(G, +)$ *and* $(H, \star)$ *is a* homomorphism *if* $h(a + b) = h(a) \star h(b)$, $\forall a, b \in G$.
A bijective homomorphism is called an isomorphism, *and then we write* $G \cong H$ *and say that* G *and* H *are* isomorphic.

kernel $\ker h := \{a \in G \mid h(a) = 0\}$

image $\operatorname{im} h := \{b \in H \mid \exists a \in G \text{ with } h(a) = b\}$

cokernel $\operatorname{coker} h := H/\operatorname{im} h$

Note that we are assuming something in our definition of the cokernel: for the definition of a quotient group to apply, we need the divisor group to be a subgroup of the dividend group. Luckily, the following lemma says that $\operatorname{im} h$ is always a subgroup of H.

Lemma 2.35. $\ker h$ *and* $\operatorname{im} h$ *are subgroups of* $(G, +)$ *and* $(H, \star)$, *respectively.*

Proof. We first prove this for $\ker h$.

1. $a, b \in \ker h \Rightarrow h(a) = h(b) = 0$. By definition of homomorphism, $h(a + b) = h(a) \star h(b) = 0 \star 0 = 0$, and thus by definition of $\ker h$, $a + b \in \ker h$. We conclude that $\ker h$ is closed under $+$.

2. Associativity follows from associativity of $+$ in G, since $\ker h \subseteq G$.

3. $\forall a \in G : h(0) \star h(a) = h(0 + a) = h(a)$, and thus $h(0) = 0$, from which $0 \in \ker h$ follows.

4. Let $a \in \ker h$. Then, $0 = h(0) = h(a - a) = h(a) \star h(-a) = 0 \star h(-a) = h(-a)$, and thus $-a \in \ker h$.

The proof for $\operatorname{im} h$ is left as an exercise. $\square$

Exercise 2.36. *Show that* $\operatorname{im} h$ *is a subgroup of* H.

Exercise 2.37. *For two abelian groups* $(G, \star)$ *and* $(H, +)$, *let the set of all homomorphisms* $f : G \to H$ *be denoted by* $\operatorname{Hom}(G, H)$.

(a) Show that $(\operatorname{Hom}(G, H), \oplus)$, *where the operation* $\oplus$ *is defined as*

$$(f \oplus g)(x) = f(x) + g(x), \forall x \in G,$$

is also a group.

(b) Show that $\operatorname{Hom}(\mathbb{Z}_2^2, \mathbb{Z}_2) \cong \mathbb{Z}_2^2$.

As the example of the integers shows, a big motivation for the study of groups comes from number theory. However, in number theory we do not only have addition but also multiplication. This motivates the following definition:

Definition 2.38. $(R, +, \cdot)$ *is a* ring, *if*

1. $(R, +)$ is an abelian group.

2. $\forall a, b, c \in R$:
$(a \cdot b) \cdot c = a \cdot (b \cdot c)$ *and* *(Associativity of $\cdot$)*
$a \cdot (b + c) = a \cdot b + a \cdot c,$
$(b + c) \cdot a = b \cdot a + c \cdot a$ *(Distributivity)*

3. $\exists 1 \in R$, such that $a \cdot 1 = 1 \cdot a = a \ \forall a \in R$.   *(Multiplicative identity)*

If $\cdot$ is commutative, we say that R is commutative.

Definition 2.39. *A commutative ring in which every non-zero element has a multiplicative inverse (i.e., $\forall a \in R \setminus \{0\}, \exists b \in R : a \cdot b = 1$) is called a* field.

Another important area of algebra, which you already know, is linear algebra. Here, vectors can be added and subtracted. Further the field of real numbers are called *scalars* and they can be multiplied with vectors. So, we have very similar operations at hand. This motivates the following generalization of the concept of vector spaces.

Definition 2.40. *Given a ring $(R, +, \cdot)$ with multiplicative identity 1, an R-module M is an abelian group $(M, \oplus)$ with an operation $\otimes : R \times M \to M$ such that for all $r, r' \in R$ and $x, y \in M$, we have*

1. $r \otimes (x \oplus y) = (r \otimes x) \oplus (r \otimes y)$

2. $(r + r') \otimes x = (r \otimes x) \oplus (r' \otimes x)$

3. $1 \otimes x = x$

4. $(r \cdot r') \otimes x = r \otimes (r' \otimes x)$

If R is a field, the R-module is called a vector space.

In the literature, often the same symbol $(\cdot)$ is used for both operations $\cdot$ and $\otimes$, and $+$ for both $+$ in R and $\oplus$ in M. For a vector space, this should feel quite normal, since for the vector space $\mathbb{R}^n$ (which is an $\mathbb{R}$-module), we also write $\cdot$ for multiplying scalars to both scalars and vectors, and $+$ for addition of both scalars and vectors.

Modules appear all over the place in homology theory. In some cases, in particular in all the cases we discuss in these lecture notes, the modules happen to be vector spaces. Thus, most of what we discuss in the following chapters could be phrased using only language from linear algebra. However, to be consistent with most of the existing literature, we will phrase most results in a slightly more general language.

Questions

1. *What is a topological space?* Give the formal definition and some examples.

2. *What is a continuous map between topological spaces? What is a homeomorphism?* State the definitions and give examples.

3. *What is a homotopy? What is a homotopy equivalence?* Give the formal definitions. Further, define deformation retracts and use them to give an alternative definition of homotopy equivalence.

4. *What are groups and the maps between them?* State the definitions and prove that the image and kernel are subgroups.

References

[1] Tamal Krishna Dey and Yusu Wang, *Computational topology for data analysis*, Cambridge University Press, 2022.

[2] J.R. Munkres, *Topology*, Prentice Hall, Incorporated, 2000.

Chapter 3

Homology

In this chapter, we introduce *homology*, a fundamental concept in algebraic topology and, as the name suggests, a crucial element of the *persistent homology pipeline* in topological data analysis. Very informally, homology can be used to count the number of "holes" of a topological space, where holes can have any dimension. While you might have an intuition of what a 2-dimensional hole in a subspace of $\mathbb{R}^2$ might be, it is not at all clear what a 4-dimensional hole in some 7-dimensional space should be. The main idea of homology is to use algebra to talk about holes in an abstract setting.

As we have already hinted at in the previous chapter, homology is an invariant of topological spaces preserved under homeomorphism and homotopy equivalence. We will manage to make this formal in Section 3.2.8.

3.1 Simplicial Complexes

In order to define homology, we restrict our attention (for now) to special types of topological spaces, namely *simplicial complexes*. We will see that this covers most natural spaces. Furthermore, homology for simplicial complexes is sufficient for all classical applications in topological data analysis. We will briefly outline a more general definition later in the chapter.
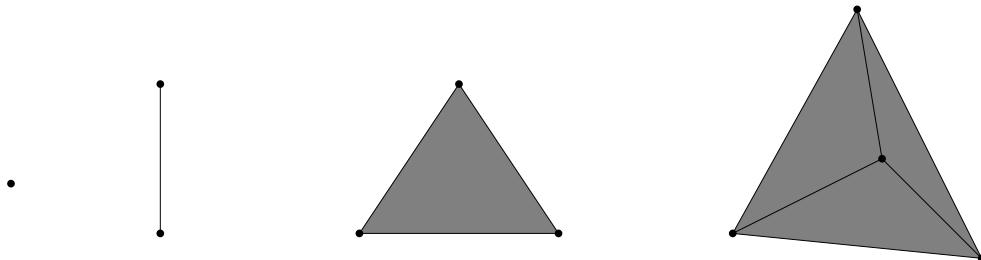
While simplicial complexes can be regarded as completely abstract objects, it is more intuitive to think of them in a geometric setting. The basic objects in a geometric simplicial complex are *simplices*:

Definition 3.1. *A* k-simplex *in* $\mathbb{R}^d$ *is the convex hull of* k+1 *affinely independent points in* $\mathbb{R}^d$.
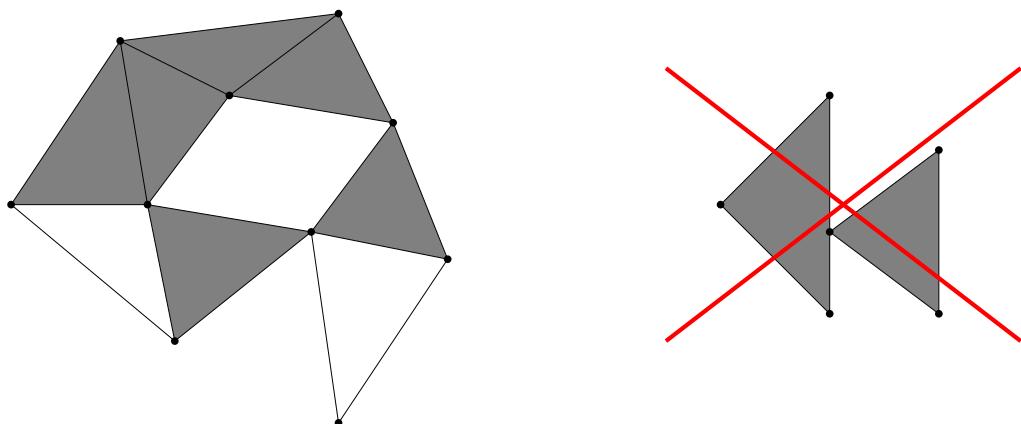
A *face* of a simplex is the convex hull of a subset of its vertices. In particular, every face of a simplex is also a simplex. The empty set $\emptyset$ is also a face. The $(k-1)$-faces of a k-simplex are called *facets*. We say the *dimension* of a k-simplex is k.

Definition 3.2. *A* geometric simplicial complex *is a family* K *of simplices such that*

- *if $\tau \in K$ and σ is a face of τ, then $\sigma \in K$, and*

- *for $\sigma, \tau \in K$, their intersection $\sigma \cap \tau$ is a face of both.*

Figure 3.1: *Some examples of simplices: a point (0-dimensional), a line segment (1-dimensional), a triangle (2-dimensional) and a (filled) tetrahedron (3-dimensional).*

Figure 3.2: *The left is a simplicial complex. The right is not, as the intersection of the two triangles is not a face of both of them.*

We say the *dimension* of a simplicial complex is the maximum dimension of any simplex. In these lecture notes, and for applications in topological data analysis in general, we may assume that all simplicial complexes are *finite*, that is, consisting of finitely many simplices.

The way we defined them, simplicial complexes are geometric objects. To arrive at a purely combinatorial description, we can simply forget about the points in space spanning our simplices.

Definition 3.3. *An* abstract simplicial complex K *is a family of subsets of a vertex set* $V(K)$ *such that if $\tau \in K$ and $\sigma \subseteq \tau$, then $\sigma \in K$.*

A k-simplex here is a subset of $k + 1$ elements, and thus again called k-dimensional. Note that 1-dimensional abstract simplicial complexes are exactly graphs: they are defined by a vertex set V and a system of two-element subsets of V, called *edges*.

From every geometric simplicial complex we get an abstract simplicial complex by simply taking the set of points as the vertex set and adding the correct subset for every simplex. For the inverse direction, we have to talk about geometric realizations:

Definition 3.4. *A geometric simplicial complex* K *is a* geometric realization *of some abstract simplicial complex* K'*, if there is an embedding* $e : V(K') \to \mathbb{R}^d$ *that takes every (abstract)* k*-simplex* $\{v_0, \ldots, v_k\}$ *in* K' *to the (geometric)* k*-simplex that is the convex hull of* $e(v_0), \ldots, e(v_k)$.

Does every abstract simplicial complex have a geometric realization? Let us only consider 1-dimensional complexes (graphs) for now. We know that not all graphs admit a straight-line embedding in the plane, as only planar graphs admit any embedding, i.e., crossing-free drawing, in the plane. However, by placing the vertices in $\mathbb{R}^3$ in such a way that no four vertices lie on a common plane, we see that we can always find a geometric realization of a graph in $\mathbb{R}^3$. This generalizes to the following realization theorem:

Theorem 3.5. *Every* k*-dimensional simplicial complex has a geometric realization in* $\mathbb{R}^{2k+1}$.

Proof. Place the vertices as distinct points on the *moment curve* in $\mathbb{R}^{2k+1}$, which is the curve given by $f(t) = (t, t^2, \ldots, t^{2k+1})$. This way, any $2k+2$ of the placed points are affinely independent. Thus, any two faces with disjoint vertex sets will not intersect in the realization, showing that the realization is indeed an embedding. $\square$

Since we now know that abstract and geometric simplicial complexes can be translated into one another, we will not make the distinction between them again and just use the word *simplicial complex* for both objects in the following. As a subset of Euclidean space, a simplicial complex thus also inherits the subspace topology from $\mathbb{R}^d$, which allows us to view simplicial complexes as topological spaces. We usually write K for the simplicial complex as a family of sets, and $|K|$ for the underlying topological space.

On the other hand, most topological spaces are not simplicial complexes. For example, the 2-sphere S^2 is not a simplicial complex, as it is not defined by a vertex set and faces. However, the boundary of a tetrahedron is a simplicial complex, and it is homeomorphic to S^2. Considering that we want to consider properties invariant under homeomorphism, we thus might as well work with the boundary of a tetrahedron instead of with S^2. This motivates the following definition.

Definition 3.6. *A simplicial complex* K *is a* triangulation *of a topological space* X*, if* $|K|$ *is homeomorphic to* X*. We say that a topological space* X *is* triangulable *if it has a triangulation.*

Triangulable spaces are nice for us, as we can replace them by simplicial complexes without any loss of topological information. Unfortunately, not all topological spaces are triangulable, but in this course we will not deal with such spaces.

While triangulations give us simplicial complexes from (triangulable) topological spaces, we would like to mention that one can also go the other way: many combinatorial structures naturally give rise to (abstract) simplicial complexes, which can in turn be interpreted as topological spaces. Thus, we can use the machinery of topological methods for gaining insights into many combinatorial problems. This gives rise to a subfield of combinatorics called *topological combinatorics*, where the topology of simplicial complexes associated to combinatorial objects is studied. Let us give some examples of such simplicial complexes.

- As we have already discussed, graphs are equivalent to 1-dimensional simplicial complexes.

- Given a graph $G = (V, E)$, we can define a simplicial complex on V by including a face $\{v_1, \ldots, v_k\}$ whenever these vertices form a clique in G. This is called the *clique complex* of G.

- For a poset $(P, \leqslant)$, the set of all chains of P forms a simplicial complex, giving rise to the *order topology*.

In topological data analysis, a highly relevant example is the *nerve*, which records the intersection pattern of a collection of sets:

Definition 3.7. *For a finite collection U of sets, its* nerve $N(U)$ *is a simplicial complex on the vertex set U that contains $u_0, \ldots, u_k$ as a k-simplex iff $u_0 \cap \ldots \cap u_k \neq \emptyset$.*

While the nerve can be seen as a purely combinatorial object describing the intersection pattern of U, it is also interesting to study its topology. If the considered sets in U are subsets of some topological space X, there is a very strong characterization of the topology of $N(U)$, if the intersections of sets in U are "well-behaved".

Definition 3.8. *Let X be a topological space, and U a finite family of closed subsets of X. We call U a* good cover*, if every non-empty intersection of sets in U is contractible.*

Under these conditions on the sets we get the following, very powerful theorem, which allows us to relate complex spaces (unions of sets) with a much simpler simplicial complex, namely the nerve of these sets. For a proof of this we refer to any textbook on algebraic topology, for example the one by Hatcher [2].

Theorem 3.9 (Nerve theorem). *If U is a good cover, then $|N(U)|$ is homotopy equivalent to $\bigcup U$.*

The nerve theorem also holds if all the sets in U are open with contractible intersections, but it may fail if some sets in U are closed, and some open: We can have an open and a closed set which do not intersect, but whose union is connected.

Now that we have defined simplicial complexes and considered some examples, we once again want to study maps between them. The study of simplicial complexes and the maps between them, as we will define them, is called *combinatorial topology*.

Definition 3.10. *A* vertex map $f : V(K_1) \to V(K_2)$ *maps vertices in* K_1 *to vertices in* K_2.

Definition 3.11. *A map* $f : K_1 \to K_2$ *is called* simplicial *if it can be described by a vertex map* $g : V(K_1) \to V(K_2)$ *such that for every simplex* $\{v_0, \ldots, v_k\}$ *we have* $f(\{v_0, \ldots, v_k\}) = \{g(v_0), \ldots, g(v_k)\}$. *Since* f *maps to* K_2 *we must have that* $f(\{v_0, \ldots, v_k\})$ *is a simplex in* K_2. *A simplicial map can also be seen as a map on the underlying spaces* $f : |K_1| \to |K_2|$.

Note that for a map to be simplicial, we do not require that $\{f(v_0), \ldots, f(v_k)\}$ is also a k-dimensional simplex, we merely require that it is *a* simplex of K_2. It is thus possible that distinct vertices of K_1 are mapped to the same vertex of K_2.

Recall that simplicial complexes are topological spaces, so there is also the notion of continuous maps between them. It can be shown that every simplicial map is continuous.

Exercise 3.12. *Let* $f : |K_1| \to |K_2|$ *be a simplicial map. Show that* f *is continuous.*

On the other hand, continuous maps in general do not need to map vertices to vertices, and are thus not simplicial. Simplicial maps are therefore more restrictive than continuous maps. However, the difference of the two concepts is smaller than one might think at first glance.

Fact 3.13. *Every continuous map* $f : |K_1| \to |K_2|$ *can be approximated arbitrarily closely by simplicial maps on appropriate subdivisions of* K_1 *and* K_2.

This shows that we can consider simplicial maps to be the analogue of continuous maps in the world of simplicial complexes. This begs the question whether other definitions from topology, such as homotopies or deformation retracts, have simplicial analogues. As we will see in the next few definitions, they do.

Definition 3.14. *Two simplicial maps* $f_1, f_2 : K_1 \to K_2$ *are* contiguous *if for every simplex* $\sigma \in K_1$ *we have that* $f_1(\sigma) \cup f_2(\sigma)$ *is a simplex in* K_2.

This is the simplicial analogue of two continuous maps being homotopic. We can thus show two simplicial complexes to be homotopy equivalent by providing two simplicial maps $f : K_1 \to K_2$ and $g : K_2 \to K_1$ such that $g \circ f$ is contiguous with the identity map on K_1 and $f \circ g$ is contiguous with the identity map on K_2.
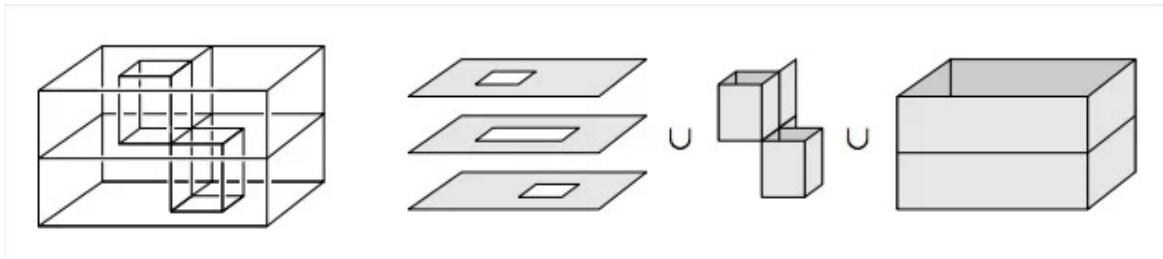
Definition 3.15. *A face of a simplicial complex is called* free, *if it is non-maximal (not inclusion-maximal) and contained in a unique maximal face.*

Note that every face that is a superset of a free face is either a maximal face or also free.

Definition 3.16. *A* collapse *is the operation of removing all faces* γ *that are a superset of some fixed free face* τ *(including* τ *itself). A simplicial complex is* collapsible *if there is a sequence of collapses leading to a single point.*

A collapse can be written as a deformation retract. Thus, a simplicial complex that is collapsible is contractible, and we consider collapses to be the simplicial analogue of deformation retracts.

You might wonder whether every contractible simplicial complex is also collapsible. We will see that this not hold: A good counterexample for this is Bing's house with two rooms, see Figure 3.3. In any triangulation of it, there are no free faces: As a 2-dimensional space, there are only vertices, edges and triangles. We only have to check edges, since triangles are maximal, and vertices are part of edges which are never maximal. Every edge is incident to at least two triangles (there are no edges on the "boundary"), and thus they are not free. Since we have no free faces, it is not collapsible.
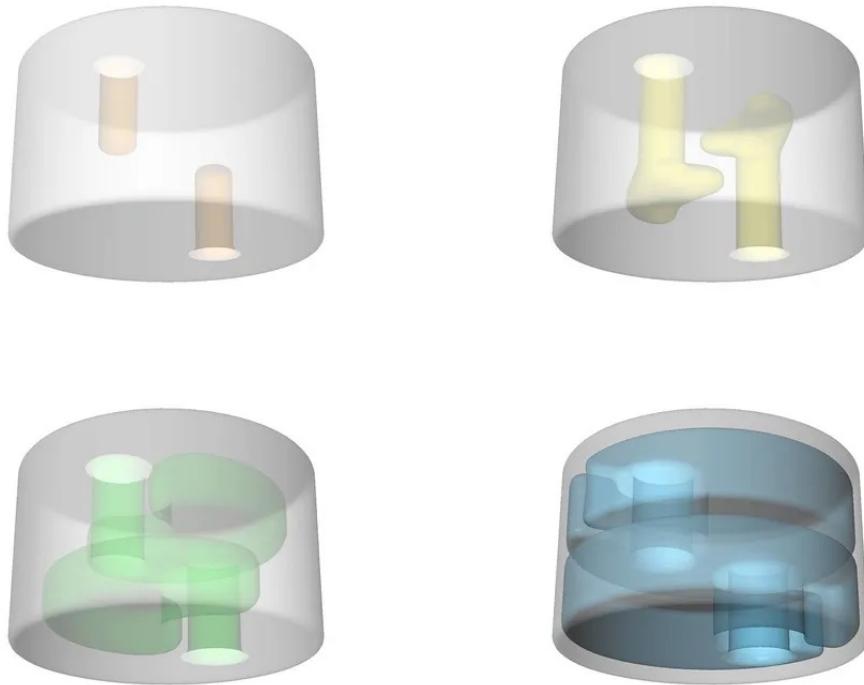
Figure 3.3: *Bing's house with two rooms. Image taken from [2].*

On the other hand, Bing's house is contractible: both Bing's house and a point are deformation retracts of a 3-dimensional ball, and thus by Fact 2.24 they are homotopy equivalent. For a visual sketch of the deformation retract from a 3-dimensional ball to Bing's house, see Figure 3.4.

To summarize, the connection between simplicial complexes and topological spaces is that every simplicial complex defines a topological space, since we can consider a geometric embedding, and the underlying space of the embedding inherits the subspace topology from $\mathbb{R}^d$. On the other hand, some topological spaces (the triangulable ones) can be expressed by simplicial complexes. As for maps, every simplicial map is continuous. On the other hand, continuous maps between simplicial complexes can be approximated by simplicial maps between subdivisions of the simplicial complexes. A similar property holds between homotopic maps and contiguous maps, as well as between deformation retracts and collapses. In general, we can say that the terms in combinatorial topology are special cases of their "continuous" counterparts, and if we consider triangulable spaces, the continuous terms can be approximated in some way by their combinatorial counterparts. The terms can thus be considered to be equivalent.

Table 3.1 summarizes the equivalent words in "continuous topology" on triangulable spaces and in combinatorial topology on simplicial complexes.

Figure 3.4: *A visual representation of the deformation retract from a 3-dimensional ball to Bing's house. Images taken from the blog Sketches of topology [1].*
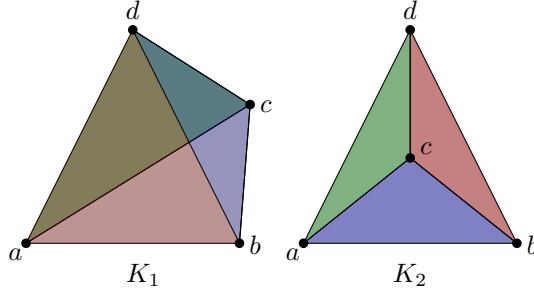
"continuous" topology	combinatorial topology
topological spaces	simplicial complexes
continuous maps	simplicial maps
homotopic maps	contiguous maps
deformation retracts	collapses

Table 3.1: *Equivalent notions in "continuous" and combinatorial topology*

3.2 Homology

Recall that homology is intended as a tool to count holes in objects, and recall that this hole count is intended as an invariant of topological spaces under homotopy equivalence. We have introduced simplicial complexes, which allow us to consider concrete combinatorial descriptions instead of abstract topological spaces.

Let us begin with some basic intuition for holes in simplicial complexes, before diving into the more technical definitions. Consider the two simplicial complexes shown in Figure 3.5. How many (and what kind of) holes should these complexes intuitively have?

Figure 3.5: *Two simplicial complexes.* K_1 *contains all four triangles* $\{a, b, c\}, \{a, b, d\}, \{a, c, d\}, \{b, c, d\}$ *as well as their subsets, while* K_2 *only contains the three triangles* $\{a, b, c\}, \{a, c, d\}, \{b, c, d\}$ *and their subsets.*

As can be seen, K_1 is the boundary of a tetrahedron. It is a triangulation of the 2-dimensional (hollow) sphere, so we would like to say that it has a hole, or *cavity*. In particular, because this cavity is of the same dimension as the cavity in the 2-dimensional sphere, we want to call this cavity a 2-dimensional hole.

On the other hand, K_2 can be viewed as a triangulation of four points in the plane, where the point a lies inside the convex hull of the other three points. It is homeomorphic to a 2-dimensional disk. Intuitively we would like to say that the complex K_2 does not have any holes.

As a 2-dimensional disk, K_2 has a *boundary*, consisting of the edges $\{a, b\}$, $\{b, d\}$ and $\{a, d\}$. On the other hand, K_1 has no boundary, just as a sphere has no boundary. We will later define a notion of boundary capturing this intuition, at least for *pure* simplicial complexes, that is, simplicial complexes whose maximal faces all have the same dimension. For example, a 1-dimensional pure simplicial complex is just a graph with no isolated vertices. In such a graph, the boundary will contain all the leaves (vertices of degree 1). Some complexes, like K_1, will have an empty boundary, and in analogy to graphs without leaves we call such complexes *cycles*[1]. Under this viewpoint, our d-dimensional holes of a simplicial complex K should be d-dimensional pure subcomplexes that are cycles. On the other hand, clearly not all cycles should be holes, as can be seen with the boundary of K_2. This boundary (the three edges $\{a, b\}$, $\{b, d\}$ and $\{a, d\}$) itself does not have a boundary, and is thus a 1-dimensional cycle. However we do not want to consider this cycle as a 1-dimensional hole of K_2 since it is "filled up", it is the boundary of the three filled in triangles.

Summed up, our intuition is that holes are subcomplexes that have no boundary (cycles) and that are not themselves boundaries of another subcomplex which would be filling in the hole. In the following we will make this intuition precise by defining the types of subcomplexes we consider, the notions of boundaries and cycles, and how to mathematically describe the cycles that are not boundaries.

---

[1]Note that technically graphs without leaves are not necessarily just cycles, but can also consist of multiple cycles glued together at vertices and edges.

3.2.1   Chains

In the following we let K be a simplicial complex, and we use m_p to denote the number of p-simplices in K. We first want to define p-*chains*, which are simply an algebraic way of formalizing and generalizing subsets of p-simplices.

Definition 3.17. *A p-chain c (in K) is a formal sum of p-simplices added with some coefficients from some ring R. A p-chain c can thus be written as*

$$c = \sum_{i=1}^{m_p} \alpha_i \sigma_i,$$

where $\alpha_i \in R$ and $\sigma_i \in K$ are p-simplices.

All we are doing in this formal sum is giving a coefficient from R to each p-simplex of K. A formal sum is only a sum in a syntactic sense (i.e., we use the symbols + and $\sum$), but there is no semantic meaning behind this operation; there is no other way to represent a chain other than the sum it is defined by.

Using the addition operation of the ring R however, we can now also add two p-chains $c = \sum \alpha_i \sigma_i$ and $c' = \sum \alpha_i' \sigma_i$ (both in K). Since the chains are both just formal sums, we can simply do this addition "component-wise", using addition in R on the coefficients:

$$c + c' := \sum_{i=1}^{m_p} (\alpha_i + \alpha_i') \sigma_i$$

We therefore have an addition operation on the set $C_p(K)$ of all p-chains in K. We show next that $C_p(K)$ endowed with this operation forms a group, and we call it the p-*th chain group* (of K).

Observation 3.18. $(C_p(K), +)$ *is an abelian group, it is free, and the p-simplices form a basis.*

Proof. To show that it is a group we observe that:

1. $C_p(K)$ is closed under addition, since $\forall c_1, c_2 \in C_p(K)$ we have $c_1 + c_2 \in C_p(K)$.

2. The operation + is associative: $\forall c_1, c_2, c_3 \in C_p(K)$,
   $(c_1 + c_2) + c_3 = \sum (\alpha_i^{(1)} + \alpha_i^{(2)}) \sigma_i + \sum \alpha_i^{(3)} \sigma_i = \sum (\alpha_i^{(1)} + \alpha_i^{(2)} + \alpha_i^{(3)}) \sigma_i = \sum \alpha_i^{(1)} \sigma_i + \sum (\alpha_i^{(2)} + \alpha_i^{(3)}) \sigma_i = c_1 + (c_2 + c_3)$.

3. We have a neutral element $0 = \sum 0 \sigma_i \in C_p(K)$.

4. Every element has an inverse: $\forall c \in C_p(K)$ we have $-c = \sum (-\alpha_i \sigma_i) \in C_p(K)$ and $c + (-c) = \sum (\alpha_i - \alpha_i) \sigma_i = 0$.

Commutativity follows from $+$ in R being commutative (recall that for any ring $(R, +, \cdot)$, $(R, +)$ is an abelian group), thus the group is abelian. Finally, the p-simplices clearly form a basis since the set of chains is defined as the set of formal sums of these p-simplices. $\square$

We can further turn $C_p(K)$ into an R-module:

Observation 3.19. *Equipped with the appropriate function $\cdot : R \times C_p(K) \to C_p(K)$, $C_p(K)$ is an R-module.*

Proof (sketch). We can define $r \cdot c$ by simply using the multiplication $\cdot$ of R component-wise on each coefficient of c, i.e., $r \cdot \sum_{i=1}^{m_p} \alpha_i \sigma_i = \sum_{i=1}^{m_p} (r \cdot \alpha_i) \sigma_i$. We leave the proof of the necessary properties as an exercise. $\square$

From now on we will always work with one of the simplest possible rings, the ring $R = \mathbb{Z}_2$. In particular this allows us to simply view chains as sets of p-simplices, the sum of chains being their symmetric differences, and we get the nice identity $c + c = 0$. With $R = \mathbb{Z}_2$, we will define *homology over* $\mathbb{Z}_2$, often also just called $\mathbb{Z}_2$-homology. Using some slightly more abstract definitions, all of the following can be extended to define homology over any ring R. For more on this, we refer to any textbook on algebraic topology, e.g., the one by Hatcher [2].

3.2.2   Boundary Maps

Now that we can talk algebraically about sets of p-simplices, we can now formalize the notion of the boundary. It should be intuitively clear what the boundary of a single p-simplex should be: just take the $(p-1)$-chain formed by its facets.

More formally, let $\sigma = \{v_0, \dots, v_p\}$ be a p-simplex. Then, $\delta_p(\sigma)$ is defined by

$$\{v_1, \dots, v_p\} + \{v_0, v_2, \dots, v_p\} + \dots + \{v_0, \dots, v_{p-1}\} = \sum_{i=0}^{p} \{v_0, \dots, \hat{v_i}, \dots, v_p\}$$

In the above notation, $\hat{v_i}$ denotes that the element v_i is omitted from the set. Note that $\delta_p(\sigma)$ is indeed a $(p-1)$-chain. For some examples, see Figure 3.6.

$$\delta_0(\, \cdot \,) = 0$$
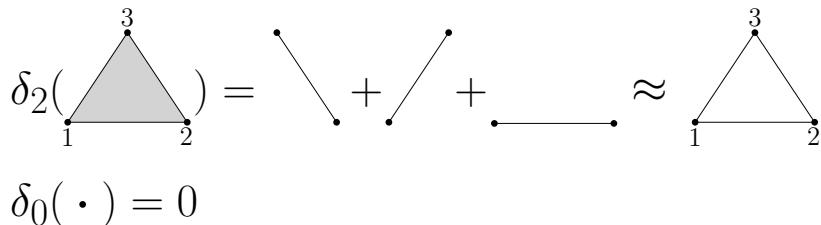
Figure 3.6: *The boundary chains of two different simplices.*
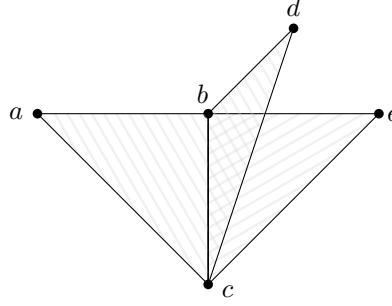
We have seen that δ_p is a map that sends a p-simplex to a $(p-1)$-chain. Thanks to the group structure of the chain group, we can now immediately extend this to any

chain. After this extension, δ_p defines a map from $C_p(K)$ to $C_{p-1}(K)$:

$$\delta_p : C_p(K) \to C_{p-1}(K)$$
$$c = \sum \alpha_i \sigma_i \mapsto \delta_p(c) = \sum \alpha_i(\delta_p(\sigma_i))$$

It is easy to prove that δ is a group homomorphism, and we call it the *boundary operator* homomorphism. Let us apply this definition to the following example. In a slight abuse of notation in favor of legibility, we denote faces $\{a, b, c\}$ by abc.

$$\delta_2(abc + bcd) = \delta_2(abc) + \delta_2(bcd)$$
$$= (ab + bc + ac) + (bc + cd + bd)$$
$$= ab + ac + cd + bd$$

$$\delta_2(abc + bcd + bce) = (ab + bc + ac) + (bc + cd + bd) + (bc + ce + be)$$
$$= ab + bc + ac + cd + bd + ce + be$$

We can see that an edge is in the boundary of a chain of triangles exactly if it is contained in an odd number of triangles of the chain, thanks to the fact that we use addition in $\mathbb{Z}_2$.

We have already seen that cycles can be boundaries. On the flipside we have also seen that the boundary of a simplex should have no boundary (i.e., it should be a cycle), where the interior of the simplex fills up the cavity given by its boundary. The following lemma generalizes this to boundaries of any chain: It states that the boundary of any boundary is empty.
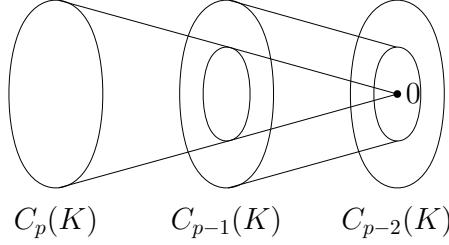
Lemma 3.20. *For* $p > 0$*,* $\delta_{p-1} \circ \delta_p(c) = 0$*, for any* p*-chain* c*.*

In the example above, $\delta_1(\delta_2(abc + bcd)) = (a + b) + (a + c) + (c + d) + (b + d) = 0$.

Proof. It is enough to show this for simplices, as $\delta_{p-1} \circ \delta_p(c) = \delta_{p-1}(\sum \alpha_i(\delta_p(\sigma_i))) = \sum \alpha_i(\delta_{p-1} \circ \delta_p(\sigma_i))$. For a p-simplex σ, every $(p-2)$-face of σ is contained in exactly 2 $(p-1)$-faces of σ, and does thus not appear in $\delta_{p-1} \circ \delta_p(\sigma)$. $\square$

The notions of homology we will introduce below actually generalize to any sequence of group homomorphisms δ_i that fulfill Lemma 3.20 above. Each such sequence of homomorphisms defines a so-called *chain complex*:

$$0 = C_{k+1}(K) \xrightarrow{\delta_{k+1}} C_k(K) \xrightarrow{\delta_k} C_{k-1}(K) \cdots C_2(K) \xrightarrow{\delta_2} C_1(K) \xrightarrow{\delta_1} C_0(K) \xrightarrow{\delta_0} C_{-1} = 0$$

$$C_p(K) \qquad C_{p-1}(K) \qquad C_{p-2}(K)$$

Figure 3.7: *A schematic illustration of a part of a chain complex.*

3.2.3 Cycle and Boundary Groups

As we already established intuitively, chains without boundaries are called cycles. These are the objects potentially giving rise to holes or cavities.

Definition 3.21. *A p-chain c is a p-cycle if $\delta(c) = 0$. $Z_p(K)$ is the p-th cycle group, consisting of all p-cycles of K.*

Lemma 3.22. *$Z_p(K)$ is a group.*

Proof. $Z_p(K) = \ker \delta_p$. (Recall that the kernel of a homomorphism is a subgroup of its domain.) □

So far we have only formally defined a boundary operator, but have not specified which chains we call boundaries. Of course, as already used implicitly before, the boundaries are the chains that are the result of applying the boundary operator.

Definition 3.23. *A p-chain c is a p-boundary if $\exists c' \in C_{p+1}(K)$ such that $\delta(c') = c$. $B_p(K)$ is the p-th boundary group, consisting of all p-boundaries of K.*

Lemma 3.24. *$B_p(K)$ is a group.*

Proof. $B_p(K) = \operatorname{im} \delta_{p+1}$. □

In the following, we will often drop the "(K)" of $C_p(K), Z_p(K)$, and $B_p(K)$ when it is clear which simplicial complex we are speaking about.

Fact 3.25. *$B_p \subseteq Z_p \subseteq C_p$, and all of them are abelian and free.*

We will not prove this statement here, but to see that $B_p \subseteq Z_p$, recall that by Lemma 3.20 the boundary of a boundary is empty.

3.2.4 Homology Groups

We are now ready to formalize the notion of holes or cavities. Recall that intuitively, a hole is a cycle that is not a boundary, that is, not filled by something higher-dimensional. Using that all objects defined so far form abelian groups, we can phrase this in algebraic terms using quotient groups.

Definition 3.26. *The p-th homology group $H_p(K)$ is the quotient group $Z_p(K)/B_p(K)$.*

Often in the literature we write $H_p(K; R)$ for homology over some ring R. Since we only work with homology over $\mathbb{Z}_2$ in these lecture notes, we just write $H_p(K)$ to mean $H_p(K; \mathbb{Z}_2)$.

Remember that the elements of a quotient group are *cosets*. In essence, each element of the homology group is a coset called a *homology class* which contains cycles that differ only by boundaries. The coset $[c] = c + B_p$ is the homology class of c. We say that c and c' are *homologous*, if $[c] = [c']$, which is equivalent to the statements $c \in c' + B_p$ and $c + c' \in B_p$. See Figure 3.8 for an example of homologous cycles, and Figure 3.9 for an example of the first homology group of a small complex.
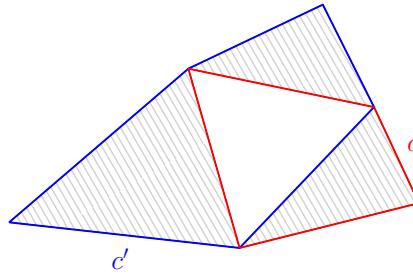
Figure 3.8: c' *and* c *are homologous cycles.*
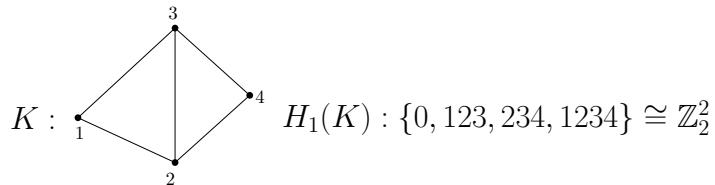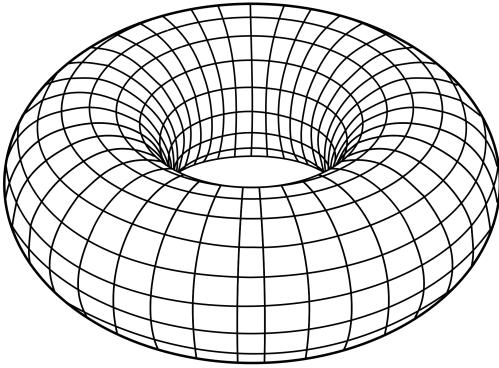
Figure 3.9: *The first homology group of a small complex.*

Exercise 3.27. *Visualize the following simplicial complex K: 0-faces $\{a, b, c, d, e\}$, 1-faces $\{ab, ac, ad, bc, bd, cd, ce, de\}$ and 2-faces $\{abc, abd, acd, bcd\}$. For the dimensions 1 & 2, what are the cycle, boundary, and homology groups of K? Note: You can express the groups by their generators. You do not need to write out all the elements.*

Figure 3.10: *A torus.*

Exercise 3.28. *Give an informal derivation for the homology groups of a torus (see Figure 3.10). Can you find a space with isomorphic homology that is not homeomorphic to the torus?*

Exercise 3.29. *For a simplicial complex* K, *its cone* CK *is the complex with the same set of vertices plus one additional vertex z, and such that for all simplices in* K *we have*

$$\{a, b, c, \ldots\} \in K \implies \{a, b, c, \ldots, z\} \in CK$$

(a) *Visualize a cone operation. What does it intuitively do to a complex?*

(b) *Show that the homology of the cone* CK *is 0 in all dimensions* $d > 0$, *for any* K.

(c) *Bonus: What would happen (intuitively and to the homology) if we extended* K *in the same way as before, but with two points? (this is called the* suspension *of* K*)*

Here are some nice properties of homology groups, that will be beneficial for us, but that we will not prove here.

Fact 3.30.

- H_p *is abelian and free.*

- H_p *is a* $\mathbb{Z}_2$*-vector space.*

Remark 3.31. *If we consider homology defined over other rings, e.g. over* $\mathbb{Z}$ *instead of* $\mathbb{Z}_2$, *the homology groups might not be free.*

Recall that our original motivation for introducing homology was to count the number of holes. With homology as we defined it, we have the algebraic structure of a vector space where we can add holes together. The number of distinct holes is now just the dimension of this vector space.

Definition 3.32. $\beta_p := \dim H_p = \dim Z_p - \dim B_p$ *is the* p-th Betti number.

In the definition above, dim denotes the dimension of a vector space as you know it from linear algebra, i.e., $\dim H_p$ is the number of elements in a basis of H_p.

Exercise 3.33. *The* **Euler characteristic** *of a simplicial complex* K *is defined as*

$$\chi = k_0 - k_1 + k_2 - \dots$$

with k_i *denoting the number of i-dimensional simplices in* K. *Convince yourself that this is an invariant property for all triangulations of the same topological space* X.
   Hint: *Show instead that* $\chi = \beta_0(K) - \beta_1(K) + \dots$. *The statement then follows by the fact that homeomorphic spaces have the same homology.*

Exercise 3.34. *Take any vector* $v = (a_0, \dots, a_d) \in \mathbb{N}^{d+1}$ *with* $a_0 > 0$. *Show that there exists a simplicial complex* K_v *with that vector as its Betti numbers.*

3.2.5   Singular Homology

With our definition of homology for simplicial complexes, we get for free a notion of homology for many topological spaces, namely the triangulable ones: we can simply triangulate them and take the homology of the triangulation. But, a topological space may have many triangulations, and it seems like the structure of the homology groups might differ depending on the choice of triangulation. The aim of this section is to sketch the tools that show that the homology of a triangulable space is independent of the chosen triangulation. The idea of singular homology is to remove the need for a fixed triangulation by looking at *all possible simplices* at once.
   Let X be a topological space, and let Δ^p be the standard p-simplex in $\mathbb{R}^{p+1}$. We want to consider all possible occurrences of this simplex in X.

Definition 3.35. *A* singular p-simplex *is a map* $\sigma : \Delta^p \to X$.

Note that in this definition we do not require σ to be injective, thus it would even be possible to map the simplex to a single point.
   We now define $C_p(X)$ the same way as before, but now on the family of all singular p-simplices, which in general makes the group uncountably infinite. We also define δ_p as before, defining $Z_p(X)$ and $B_p(X)$, which are now also uncountably infinite. Finally, we again define $H_p(X) = Z_p(X)/B_p(X)$. Surprisingly, this definition agrees with the simplicial definition of homology on any triangulation of X.

Theorem 3.36. *Let X be a topological space, K a triangulation of X. Then we have $H_p(X) \cong H_p(K)$ for all $p \geqslant 0$.*

As isomorphisms for vector spaces are an equivalence relation, we also get the desired independence of the triangulation.

Corollary 3.37. *Let K_1, K_2 be two distinct triangulations of X. Then, $H_p(K_1) \cong H_p(K_2)$ for all $p \geqslant 0$, that is, homology is independent of the chosen triangulation.*

For the remainder of these lecture notes we will only work with simplicial homology, but we often talk about the homology of a triangulable space without specifying a triangulation. The above corollary gives us the right to do this.

3.2.6 The 0-th homology group

The homology group that is easiest to understand is the 0-th one. Recall that the 0-simplices of a simplicial complex K are simply its vertices. Since vertices do not have any boundaries, every vertex is a 0-cycle. The boundary of a 1-simplex simply consists of the two vertices which are connected by the edge. We can thus see that two vertices v_1 and v_2 are homologous if there is a path from v_1 to v_2, and the homology class $[v_1]$ is simply the connected component containing v_1.

Observation 3.38. $\beta_0(K)$ *is the number of connected components of* K.

As a consequence, the 0-homology classes are all the formal sums of connected components.

3.2.7 Homology of Spheres

One of the main intuitions for us when we introduced homology was that a d-sphere should have a single d-hole and no other holes. We will now check whether our definition captured this intuition correctly. Since we have seen in Section 3.2.5 that homology is independent from the chosen triangulation, let us fix some triangulation of the sphere S^d. A good candidate (due to its simplicity) is the boundary of a simplex, that is, $S^d \simeq \delta(\Delta^{d+1})$, with the vertex set $V = \{v_0, \dots, v_{d+1}\}$.

$H_0(S^d)$: Let us first investigate $H_0(S^d)$. Since all vertices are connected, all vertices are homologous, and $H_0(S^d) = \langle [v] \rangle \cong \mathbb{Z}_2$.

$H_d(S^d)$: Next, let us check $H_d(S^d)$. We first compute Z_d: The d-simplices are exactly the sets $\sigma_i = \{v_0, \dots, \hat{v_i}, \dots, v_{d+1}\}$. Note that every $(d-1)$-simplex occurs as the boundary of exactly two such d-simplices. Thus, both the zero element (empty chain) as well as the chain c consisting of all d-simplices are part of Z_d. On the other hand, no chain $c' \notin \{0, c\}$ can be a cycle, since for such a chain there must be some d-simplex $\sigma \in c'$

neighboring some d-simplex $\sigma' \notin c'$. The $(d-1)$-simplex that is a boundary of both σ and σ' would then be part of $\delta_p(c')$. We conclude that $Z_d(S^d) = \langle c \rangle$.

Since $\delta(\Delta^{d+1})$ is a d-dimensional simplicial complex, and thus does not contain any $(d+1)$-simplices, no non-empty d-chain can be a boundary. We thus get that $B_d(S^d)$ is the group containing only 0.

We finally get $H_d(S^d) = Z_d/B_d = Z_d \cong \mathbb{Z}_2$.

$H_p(S^d)$:   Finally, let us go to $H_p(S^d)$, for $0 < p < d$: Let $c = \sum \alpha_i \sigma_i$ be a p-cycle. We aim to show that c is homologous to the 0-chain, i.e., that $[c] = [0]$. Equivalently, we show that c must be a boundary.

Let $\sigma = (v_{m_0}, \ldots, v_{m_p})$ be any p-simplex in c which does not include v_0. We will keep replacing such simplices by simplices which do contain v_0, until we have no more simplices not containing v_0.

Let b be the $(p+1)$-simplex $(v_0, v_{m_0}, \ldots, v_{m_p})$. Note that $b \in \delta(\Delta^{d+1})$ and thus $\delta(b)$ is a p-boundary. Also note that σ is in $\delta(b)$. Furthermore, σ is the only p-simplex in $\delta(b)$ which does not contain v_0. We now add $\delta(b)$ to c, to get $c' := c + \delta(b)$. Since we added a boundary, $[c] = [c']$ (i.e., c and c' are homologous). Furthermore, c' contains one fewer p-simplex not containing v_0, when compared to c.

We repeat this process until we reach a cycle c^* in which every p-simplex contains v_0. We now claim that c^* must be the trivial cycle: Assume c^* contains some p-simplex $a = (v_0, v_{a_1}, \ldots, v_{a_p})$. Then, the $(p-1)$-simplex $a' = (v_{a_1}, \ldots, v_{a_p})$ is part of $\delta(a)$. But, a' cannot be part of the boundary of any other p-simplex in c^*, since the only p-simplex containing a' as a face while also containing v_0 is a. Thus, to have an empty boundary, we have $c^* = 0$. By construction, $[c] = [c^*]$, therefore $[c] = 0$ as we aimed to prove.

We have proven that every cycle is homologous to 0, and we can conclude that for all $0 < p < d$, $H_p(S^d) = 0$.

Since S^d is d-dimensional, we do not have any simplices of dimensions $p > d$, and thus $H_p(S^d) = 0$ for $p > d$. Combining all these arguments we conclude the following theorem:

Theorem 3.39. *For any* $d > 0$, *we have*

$$H_p(S^d) = \begin{cases} \mathbb{Z}_2 & p \in \{0, d\} \\ 0 & else. \end{cases}$$

$$\beta_p(S^d) = \begin{cases} 1 & p \in \{0, d\} \\ 0 & else. \end{cases}$$

3.2.8 Induced Homology

As usual, now that we have defined some mathematical objects (homology groups) we are also interested in the maps between them. For simplicial complexes we have defined

simplicial maps, and we now want to study the effect that simplicial maps have on the homology of a space.

We first extend simplicial maps to the chain groups.

Definition 3.40. *Let* $f : K_1 \to K_2$ *be a simplicial map. This induces a* chain map

$$f_\# : C_p(K_1) \to C_p(K_2)$$

$$c = \sum \alpha_i \sigma_i \mapsto f_\#(c) := \sum \alpha_i \tau_i, \ \ where \ \tau_i = \begin{cases} f(\sigma_i) & if \ f(\sigma_i) \ is \ p\text{-}simplex \ in \ K_2 \\ 0 & otherwise \end{cases}$$

Note that $f(\sigma_i)$ is always a simplex in K_2 since f is a simplicial map, but it could be a simplex of smaller dimension. This is why we need the condition in the above definition of τ_i.

The following can be shown with a bit of work:

- $f_\# \circ \delta = \delta \circ f_\#$

- $f_\#(B_p(K_1)) \subseteq f_\#(Z_p(K_1))$

- $f_\#(Z_p(K_1)) \subseteq Z_p(K_2)$, $f_\#(B_p(K_1)) \subseteq B_p(K_2)$

From this chain map $f_\#$, we now get a well-defined *induced homomorphism* between the homology groups of K_1 and K_2.

Definition 3.41. *Let* f *be a simplicial map and* $f_\#$ *its induced chain map. This induces a homomorphism*
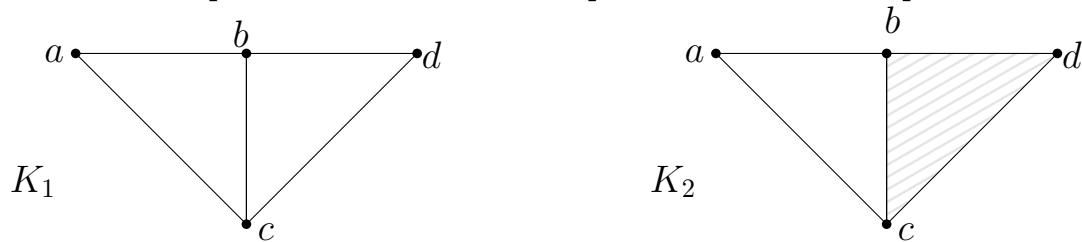
$$f_* : H_p(K_1) \to H_p(K_2)$$
$$[c] = c + B_p \mapsto f_\#(c) + B_p(K_2) = [f_\#(c)].$$

Fact 3.42. *If* $H_p(K_1)$ *and* $H_p(K_2)$ *are vector spaces (as they are in e.g.* $\mathbb{Z}_2$*-homology, which is what we are using), then* f_* *is a linear map.*

We also get the following *functorial property*, which we will not prove.

Fact 3.43. *For two simplicial maps* $f : X \to Y$, $g : Y \to Z$, *we have* $(g \circ f)_* = g_* \circ f_*$.

Let us compute the induced homomorphism of a small example:

We consider the inclusion map $f : K_1 \hookrightarrow K_2$.

$$H_1(K_1) = \{0, [abc], [bcd], [abdc]\} \cong \mathbb{Z}_2^2$$

$$f_*(0) = 0, \ f_*([abc]) = [abc]$$

$$f_*([bcd]) = 0, \ f_*([abdc]) = [abc]$$

Exercise 3.44. *Let*

$$K_1 = \{\emptyset, a, b, c, d, e, ab, ac, bc, bd, cd, ce, de, abc\}$$

and

$$K_2 = \{\emptyset, w, x, y, z, wx, wy, xy, xz, yz\}.$$

Consider the map $f : K_1 \to K_2$ *induced by the vertex map*

$$a \mapsto y, b \mapsto x, c \mapsto y, d \mapsto z, e \mapsto z.$$

You can verify easily that f *is simplicial. Compute* $f_* : H_p(K_1) \to H_p(K_2)$ *for* $0 \leqslant p \leqslant 2$.

Exercise 3.45. *Which of the following four statements is true for every simplicial map* f?
"If f *is {injective, surjective}, then* f_* *is {injective, surjective}."*

The following fact has some very powerful consequences, as we will see.

Fact 3.46. *If* $f, g : K_1 \to K_2$ *are contiguous,* $f_* = g_*$.

Note that the definition of induced homology extends from simplicial maps to maps between any topological spaces. We will not state the exact definitions, but the following fact is the continuous analogue (remember that two simplicial maps being contiguous is analogous to two maps being homotopic) of Fact 3.46.

Fact 3.47. *If* $f, g : X \to Y$ *are homotopic,* $f_* = g_*$.

Thanks to this fact we get the following corollary, which shows that homology is indeed an invariant under homeomorphisms, and even under homotopy equivalence. This also gives us the option to compute the homology of a space by computing the homology of a potentially simpler homotopy equivalent space.

Corollary 3.48. *If* $f : X \to Y$ *is a homotopy equivalence (i.e., there exists* $g : Y \to X$ *such that* $f \circ g$ *is homotopic to* id_Y *and* $g \circ f$ *is homotopic to* id_X), *then* f_* *is an isomorphism.*

Proof. Thanks to Fact 3.43 we have $(g \circ f)_* = g_* \circ f_*$. By Fact 3.47, $(g \circ f)_* = (\mathrm{id}_X)_*$, which is an isomorphism. Since we thus know that $g_* \circ f_*$ is an isomorphism, we know that f_* must be injective and g_* must be surjective. By a symmetric argument considering $f \circ g$ we also get that f_* is surjective and g_* is injective, and thus both f_* and g_* are isomorphisms. $\qquad\square$

Exercise 3.49.

Consider the space you get when you glue together two points of a torus. What is the homology of this space?

Consider the space you get when you simultaneously pierce a balloon at n distinct locations. What is the homology of this space?

Exercise 3.50. *Let $f, g : S^1 \to S^1$ be continuous maps such that $f(-x) = f(x)$ and $g(-x) = -g(x)$ for all $x \in S^1$.*

  a) *Convince yourself that $f_* : H_1(S^1) \to H_1(S^1)$ is trivial (maps everything to 0) and that g_* is an isomorphism.*

  b) *Show that f and g are not homotopic.*

  c) *Show that there is no map $h : S^2 \to S^1$ such that $h(-x) = -h(x)$.*

  d) *Conclude that every map $\phi : S^2 \to \mathbb{R}^2$ with $\phi(-x) = -\phi(x)$ has a zero.*

The statement you have proven in d) is equivalent to the 2-dimensional case of the famous Borsuk-Ulam theorem, which implies statements such as "at any time, there are two antipodal points on the earth with both the same temperature and atmospheric pressure".

3.2.9 Application: Brouwer Fixed Point Theorem

In this section we finally collect the fruits of our hard work by using homology to give a relatively short proof of the famous fixed point theorem by Brouwer. Here, $\mathbb{B}^d$ denotes the unit ball of dimension d.

Theorem 3.51 (Brouwer fixed point theorem). *Let $f : \mathbb{B}^d \to \mathbb{B}^d$ be continuous. Then, f has a fixed point, that is, $\exists x \in \mathbb{B}^d$ such that $f(x) = x$.*

  This theorem has many fascinating implications:

  • Take two sheets of paper lying on top of each other. Crumple the top sheet and set it back onto the other sheet. No matter how you crumpled the sheet, at least one point of the crumpled sheet lies exactly above its corresponding point in the bottom sheet.

  • If you open a map of Switzerland in Switzerland, there is at least one point on the map which is at its exact position.[2]

  • If you take a cup of liquid and stir or slosh it, at least one atom ends up at its original position (but if you shake you might break continuity).

  • The theorem also has many applications in mathematics and computer science, such as in fair divisions or for proving existence of Nash equilibria.

---

[2]The theorem only applies when ignoring the Italian and German exclaves of Campione d'Italia and Büsingen am Hochrhein.

To prove Theorem 3.51, we first introduce the following definition and a helper lemma, which we only prove after proving Theorem 3.51 itself.

Definition 3.52. *A map* $r : X \to A \subseteq X$ *is a* retraction *if* $r(a) = a$, $\forall a \in A$.

Lemma 3.53. *There is no retraction* $r : \mathbb{B}^d \to S^{d-1}$.

Proof of Theorem 3.51. We prove the theorem by contradiction. For an illustration of the argument see Figure 3.11. Assume $f : \mathbb{B}^d \to \mathbb{B}^d$ has no fixed point. For each x, consider the ray $\overrightarrow{f(x)x}$ and let $r(x)$ be the intersection of this ray with S^{d-1}. Then, $r : \mathbb{B}^d \to S^{d-1}$ is continuous (which we do not prove here) and $r(s) = s \; \forall s \in S^{d-1}$, since no matter where $f(s)$ lies, $\overrightarrow{f(s)s}$ first intersects S^{d-1} in s. Thus, r is a retraction, which does not exist by Lemma 3.53. $\qquad\qquad\square$
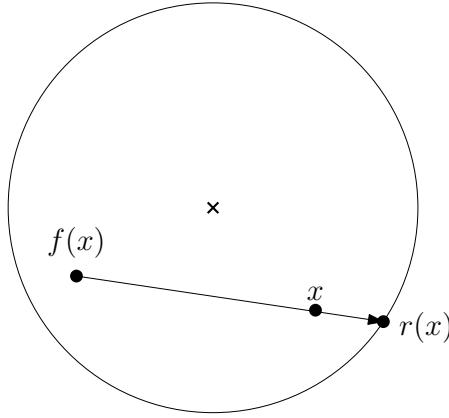
Figure 3.11: *If* f *has no fixed point, we get a retraction to the boundary.*

It remains to prove the helper lemma.

Proof of Lemma 3.53. Consider the inclusion map $i : S^{d-1} \hookrightarrow \mathbb{B}^d$, and a retraction $r : \mathbb{B}^d \to S^{d-1}$. By definition, we have $r \circ i = \mathrm{id}$. Let us look at the induced maps of r and i in the $(d-1)$-th homology of S^{d-1} and $\mathbb{B}^d$. Recall that $H_{d-1}(S^{d-1}) \cong \mathbb{Z}_2$ and $H_{d-1}(\mathbb{B}^d) \cong 0$. We thus view i_* as a homomorphism from $\mathbb{Z}_2$ to 0, and r_* as a homomorphism from 0 to $\mathbb{Z}_2$. But since $r \circ i = \mathrm{id}$, we also have $r_* \circ i_* = \mathrm{id}$. We can combine this to reach a contradiction:

$$1 = \mathrm{id}(1) = (r_* \circ i_*)(1) = r_*(i_*(1)) = r_*(0) = 0$$

Thus, either i or r cannot exist, but since i exists, r cannot. $\qquad\qquad\square$

Questions

5. *What is a simplicial complex?* Define geometric and abstract simplicial complexes and state and prove the realization theorem (Theorem 3.5).

6. *What are simplicial and contiguous maps?* State the definitions and discuss the connection to their counterparts in continuous topology.

7. *Is every contractible simplicial complex collapsible?* Define the notion of collapsibility and describe Bing's house with two rooms.

8. *What is simplicial homology?* Explain the intuition and give the formal definitions of chains, boundaries and cycles.

9. *Why is the homology of a triangulable space independent of the chosen triangulation?* Explain the idea of singular homology.

10. *What are the homology groups of a sphere?* State and prove the corresponding theorem (Theorem 3.39).

11. *How does a simplicial map between two simplicial complexes induce maps between their homology groups?* Define induced homomorphisms.

12. *What is the Brouwer fixed point theorem?* State, illustrate and prove the Brouwer fixed point theorem (Theorem 3.51).

References

[1] Kenneth Baker, Sketches of topology - Bing's house. https://sketchesoftopology.wordpress.com/2010/03/25/bings-house/, accessed: 2023-04-27.

[2] Allen Hatcher, *Algebraic topology*, Cambridge Univ. Press, Cambridge, 2000.

Chapter 4

Persistence

In the previous chapter, we have studied the homology of fixed simplicial complexes. In this chapter, we will look at simplicial complexes that grow over time. Let us start with a small example. Consider the following process of building up a triangle abc. At time t_1, we add the vertices a and b together with the edge ab. This gives birth to a single connected component. At time t_2 we add the vertex c, giving birth to a second connected component. At time t_3 we add the edge ac, connecting the two components. We can interpret this as the younger of the components being absorbed by the older component. In more crude language, we say that the younger component dies. At time t_4 we add the final edge bc, which gives birth to a hole, that is, an element of the homology group H_1. Finally, at time t_5 we add the interior of the triangle, killing the hole born at t_4. We can summarize this process as follows: we have a connected component that was born at t_1 and survived the entire process, and a connected component that was born at t_2 that died again at t_3. Finally, we have a hole born at t_4 dying at t_5. Capturing this information of holes with their birth and death is the goal of persistent homology.

Persistent homology can be applied to data analysis by defining (in a way that we will see soon) a process to build up a simplicial complex from point cloud data and computing the birth and death times of holes. Subtracting the birth time from the death time we get the lifespan of a hole; the underlying idea is that holes with a short lifetime are a byproduct of the process (noise), whereas holes with a long lifespan convey information about the shape of the underlying data.

4.1 Filtrations

We start by a mathematical formulation of the process of growing a simplicial complex or, more general, a topological space. A *filtration* is a nested sequence of subspaces

$$\mathcal{F} : X_0 \subseteq X_1 \subseteq X_2 \subseteq \ldots \subseteq X_n = X.$$

For each $i \leqslant j$, we have the inclusion map $\iota_{i,j} : X_i \hookrightarrow X_j$. Given these functions ι, we get induced maps in homology: $h_p^{i,j} = \iota_* : H_p(X_i) \to H_p(X_j)$. Filtrations are a very

general object that appear naturally in many settings. Let us look at some important examples of filtrations.

- Given a function $f : X \to \mathbb{R}$, we can define the (uncountably infinite) *sublevel set filtration* $X_a = f^{-1}(-\infty, a]$.

- A *simplicial filtration* is a nested sequence of subcomplexes

$$\mathcal{F} : K_0 \subseteq K_1 \subseteq \ldots \subseteq K_n = K.$$

  We call a simplicial filtration *simplex-wise*, if $K_i \setminus K_{i-1}$ is a single simplex (or empty).

- We call a function $f : K \to \mathbb{R}$ *simplex-wise monotone* if for every $\sigma \subseteq \tau$ we have $f(\sigma) \leqslant f(\tau)$. A simplex-wise monotone function guarantees us that the sublevel set filtration by f gives a proper simplicial filtration. Note that it does not necessarily guarantee us that the sublevel set filtration is simplex-wise (e.g., consider a function f that is not injective).

- We can also define a simplicial filtration by ordering our vertices $v_0, v_1, \ldots, v_n$. Then, let K_i be the simplicial complex induced by the vertices $v_0, \ldots, v_i$. We call the simplices $K_i \setminus K_{i-1}$ added when adding v_i the *lower star* of v_i. Thus, this type of filtration is also called the *lower star filtration*.

- Given some data points in $\mathbb{R}^d$, we can define a filtration based on our intuition of *growing balls*: We consider the nerve of all balls $B(p, r)$; with growing r we get more and more faces in this nerve. We will later formalize this into the so-called *Čech complex*.

4.2 Persistent Homology

As we have seen, from a filtration $X_0 \subseteq X_1 \subseteq \ldots \subseteq X_n$ we get a sequence of homology groups with homomorphisms between them:
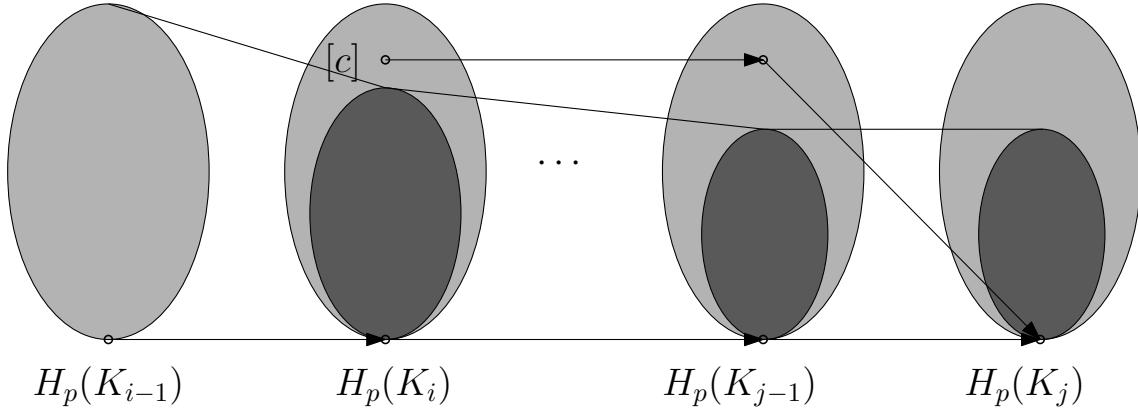
$$H_p(\mathcal{F}) : H_p(X_0) \to H_p(X_1) \to H_p(X_2) \to \ldots \to H_p(X_n).$$

Such an object is called a *persistence module*. Given a persistence module, we can now define groups that capture all the holes that are alive during a certain period.

Definition 4.1. *The* p-*th persistent homology group* $H_p^{i,j}$ *is defined by*

$$H_p^{i,j} := \operatorname{im} h_p^{i,j} = Z_p(K_i)/(B_p(K_j) \cap Z_p(K_i)).$$

This definition characterizes the cycles that that are present already in K_i and that are not boundaries even in K_j.

Figure 4.1: *An illustration of a class* [c] *being born at* K_i *and dying entering* K_j.

Definition 4.2. *The* p-*th persistent Betti numbers* $\beta_p^{i,j}$ *are the dimensions of the* p-*th persistent homology groups:* $\beta_p^{i,j} = \dim H_p^{i,j}$.

Exercise 4.3. *Let* $p \geqslant 1$. *For every* $n \geqslant 1$, *construct a filtration* $X_1 \subseteq X_2 \subseteq \ldots \subseteq X_n$ *such that*

- $H_p(X_k) \neq 0$ *for all* $k \in \{1, \ldots, n\}$ *and*

- $H_p^{i,j} = 0$ *for all* $i < j$.

We say that a p-homology class $[c]$ (a p-hole) is *born at* K_i if $[c] \in H_p(K_i)$ but $[c] \notin H_p^{i-1,i}$. Similarly, $[c]$ *dies entering* K_j, if $[c] \neq 0$ in $H_p(K_{j-1})$ but $h_p^{j-1,j}([c]) = 0$.

It is not always obvious which homology class dies. Consider the following filtration: X_1 consists of two points a and b, and in X_2 the two points are connected by an edge. Let us look at H_0, that is, the connected components. We have that $H_0(X_1) \simeq \mathbb{Z}_2^2$, with the natural basis $\{[a], [b]\}$. On the other hand, in X_2 there is only a single connected component, and $[a] = [b]$. So a homology class is dying, but both our basis elements $[a]$ and $[b]$ survive. What is happening?

It turns out that we were not careful with our choice of basis: $H_0(X_1)$ can also be viewed as being generated by $[a]$ and $[a + b]$, and the class $[a + b]$ indeed dies going into X_2. In general, if two homology classes merge, they both do not die, but their sum does. There is a consistent choice of basis which allows us to only look at persistent homology in terms of basis elements, but we do not go into this at this point.

If we have a simplex-wise filtration, we can circumvent the above issue by sorting homology classes by the time where they were born (recall the solution to Exercise 3.33 to see why this gives a total order). When two classes merge, we just say the "younger one" dies. This can be seen as adapting the considered basis along the way.

Persistence pairings are another way around this issue. We add some final complex K_{n+1} which has trivial homology (i.e., by adding all simplices that are not yet present). Then, we aim to figure out how many holes get born at K_i and die entering K_j. For this,
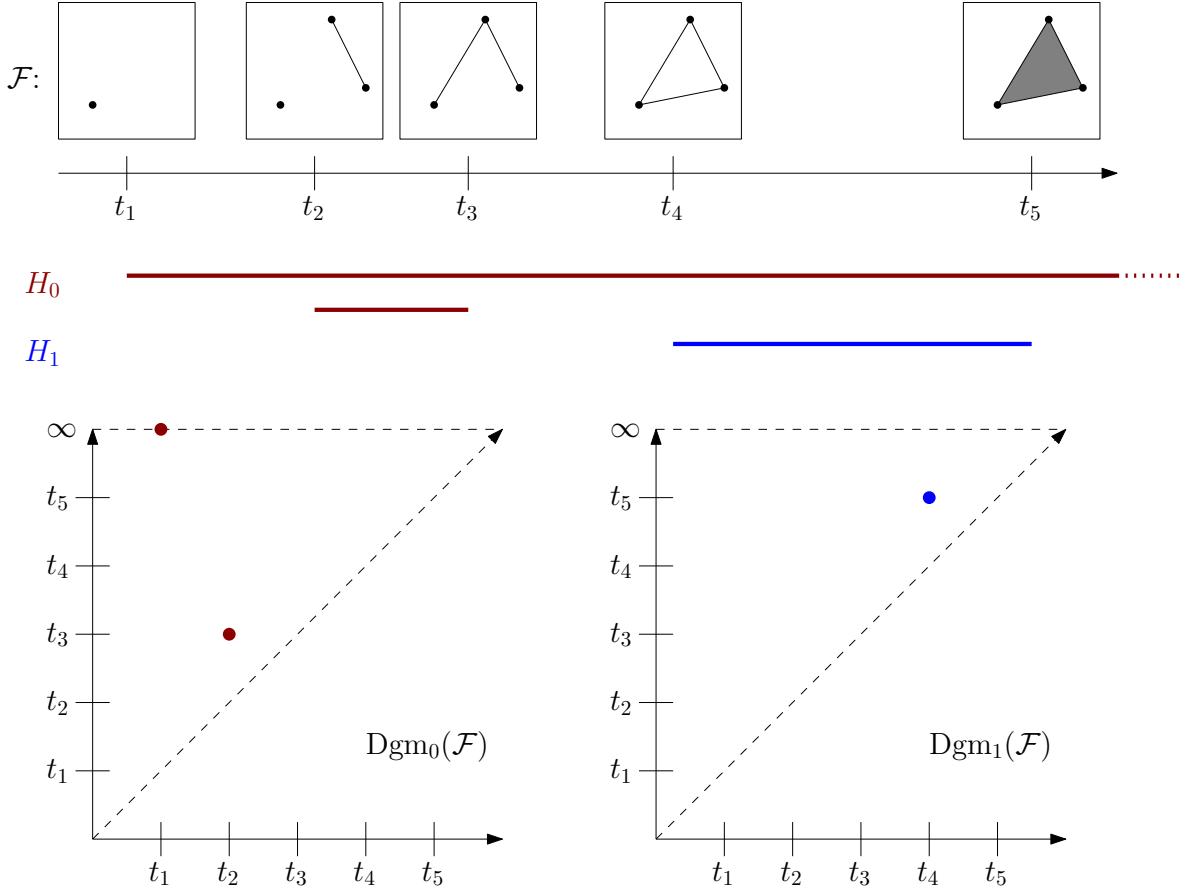
Figure 4.2: *An example of a filtration with the corresponding barcodes and persistence diagrams.*

we define

$$\mu_p^{i,j} := (\beta_p^{i,j-1} - \beta_p^{i,j}) - (\beta_p^{i-1,j-1} - \beta_p^{i-1,j}), \text{ for } i < j \leqslant n+1.$$

Here, the content of the left parenthesis denotes the number of holes born at or before K_i, which die entering K_j. Conversely, the right parenthesis denotes the number of holes born strictly before K_i, and die entering K_j. Thus, subtracting the two, gives the number of holes born exactly at K_i and die entering K_j. Note that this conveys the information that we are interested in, but does not require choosing any basis.

The *persistence diagram* $\mathrm{Dgm}_p(\mathcal{F})$ is a birth-death diagram which contains a point for every pair i, j for which $\mu_p^{i,j} > 0$. If we give each K_i a timestamp a_i, the point is drawn at the coordinates (a_i, a_j). We give each point multiplicity $\mu_p^{i,j}$. On the diagonal we add points with infinite multiplicity, for some technical reasons that will become apparent later. We can also represent the same information by *barcodes*: For every i, j, we draw $\mu_p^{i,j}$ intervals $[a_i, a_j]$. This is then called the p-th persistence barcode.

Exercise 4.4. *Consider the simplex-wise filtration induced by the order $\sigma_1, \ldots, \sigma_N$ on the simplices of a complex K. When does the order*

$$\sigma_1, \ldots, \sigma_{k-1}, \sigma_{k+1}, \sigma_k, \sigma_{k+2}, \ldots, \sigma_N$$

induce a simplex-wise filtration too? When it does, describe the relation between the corresponding persistence diagrams.

Exercise 4.5. *Give two filtrations $X_1 \subseteq \ldots \subseteq X_n$ and $Y_1 \subseteq \ldots \subseteq Y_n$ that have the same persistence diagrams but for which for any $i \in \{1, \ldots, n\}$, X_i is not homotopy-equivalent to Y_i.*

4.3   Algorithms for Persistent Homology

So far we have considered homology and persistent homology only on a mathematical level. However, for practical applications we are interested in actually computing homological information. In this section we discuss how we can compute persistence pairings given simplicial filtrations. This will of course also allow us to compute persistence diagrams and persistence barcodes.

4.3.1   Persistence Pairing Algorithm

The first algorithm we consider is the so-called persistence pairing algorithm. It only works on simplex-wise filtrations, we thus restrict our attention to such filtrations. In any time step j, we add a single simplex $\sigma_j := K_j \setminus K_{j-1}$. Let p be its dimension. There are only two things that can happen to the homology when adding σ_j: Either, a new non-boundary p-cycle c (i.e., a hole) is born, or a $(p-1)$-cycle becomes a boundary (i.e., a hole dies). In the first case we say that σ_j is a *creator*. Otherwise, we say that σ_j is a *destructor*. The fact that in every step exactly one of the two events happens is a consequence of the Euler characteristic, as discussed in Exercise 3.33.

When a new simplex σ_j destroys a hole, this corresponds to an interval of the persistence barcode ending. The beginning of that interval is at the time step when this hole was born, which corresponds to a unique simplex (recall, we are considering simplex-wise filtrations only). This unique simplex must be a creator, since when it was inserted a hole was born. The idea of the persistence pairing algorithm is to form pairings between destructors and creators. To do this, the algorithm assumes the newly added simplex σ_j to be a destructor, and tries to find the corresponding unpaired creator using a simple heuristic. If no such creator can be found by the procedure, we know that σ_j must actually be a creator itself.

The heuristic is quite simple to describe. We have to look for an unpaired creator only within a cycle c that becomes a boundary due to the insertion of σ_j. Among this cycle c, we wish to pair σ_j with the youngest unpaired creator. Any such cycle c must be homologous to $\delta\sigma_j$, which is the simplest candidate for such a cycle c. This is thus

where the search begins. We first try to pair σ_j with the youngest $(p-1)$-simplex ρ of its boundary. If ρ is unpaired, we pair it to σ_j and we are done. Otherwise, ρ is already paired with some (p-simplex) τ. In this case we replace c by $c + \delta\tau$. This is now a new candidate cycle, in which we can try pairing σ_j to the youngest simplex. We repeat this process until we found an unpaired creator we can pair σ_j to, or until we cannot continue because $c = 0$. In this case we label σ_j as a new creator. At the end of the algorithm (after processing all steps of the filtration), all remaining unpaired creators correspond to holes present at the last step of the filtration, and we pair them with the element ∞.

We refrain from giving a complete proof of this algorithm's correctness. Such a proof can be found in [1], however the algorithm presented there is slightly more complex and more efficient. We would only like to note that when we label a simplex a creator that this is correct to do so: If we reach $c = 0$ we know that the boundary of σ_j is homologous to 0 (we obtained 0 by adding boundaries to $\delta\sigma_j$). Thus, σ_j cannot be a destructor. We can thus safely label σ_j as a new creator.

We summarize this algorithm in the following pseudocode:

Algorithm 1: The persistence pairing algorithm.

    Input: A simplex-wise filtration of K given by an order of simplices $\sigma_1, \ldots, \sigma_N$

    for $j = 1$ *to* n **do**

        $c := \delta\sigma_j$;

        while $c \neq 0$ **do**

            $i :=$ largest integer such that $\sigma_i \in c$ and σ_i is creator;

            $\rho := \sigma_i$;

            if ρ *is unpaired* **then**

                Label σ_j as destructor and pair ρ and σ_j;

                $c := 0$

            else

                $\tau :=$ simplex ρ is paired to;

                $c := c + \delta\tau$;

            end

        end

        if σ_j *has not been labelled a destructor* **then**

            Label σ_j a constructor;

        end

    end

    Pair all unpaired constructors with ∞;

Exercise 4.6. *Let* G *be a weighted connected graph, where all edge weights are pairwise distinct. Consider a filtration that first inserts all vertices (in some arbitrary order) and then inserts the edges one by one, ordered by increasing weight. What is the set of destructors?*

4.3.2 Matrix Reduction Algorithm

In practice, a different algorithm is used, the *Matrix Reduction Algorithm*. This algorithm implements the same intuition as the persistence pairing algorithm. It has a few advantages: First off, it is more efficient (it avoids the need to add the same boundaries multiple times, similarly to the version of the persistence pairing algorithm provided in [1]). Second, it is phrased in the language of matrices, which allows us to implement it more efficiently using matrix-multiplication techniques. Lastly, the way we describe it in the following it also works with non-simplex-wise filtrations.

In the matrix reduction algorithm, we first find a total order on our simplices. If the input filtration is simplex-wise, this is just the insertion order. Otherwise, we order the simplices primarily by insertion order, and within each set of simultaneously added simplices, we order the simplices by increasing dimension, and then lexicographically. Then, we construct an $N \times N$ matrix, which is the so-called *boundary matrix*. Each row and column is labelled by a simplex, ordered by the order we defined above. We then insert a 1 at row σ and column τ, if σ is part of the boundary of τ.

We now modify this boundary matrix to obtain the *reduced boundary matrix*, from which the persistence pairings can then be read off. We process the columns from left to right. For each column c, we look at the lowest 1 in the column. We call this 1 the *pivot element* of the column. If there is a column $c' < c$ to the left that also has a pivot element in the same row, we add c' to c (in $\mathbb{Z}_2$). This is repeated until no such column $c' < c$ exists.

After processing all the columns, the matrix is in a reduced form: For every row, there is at most one column whose lowest 1 (its pivot element) lies in that row. From this we can now read the persistence pairings: Empty columns correspond to creators (births). To find the death of a creator, look at its corresponding row, and search for a column that has a pivot element in that row. This column is the destructor corresponding to the creator. If there is no such column, this creator never dies, i.e., is unpaired or paired with ∞.

We again summarize this algorithm in the pseudocode below. Let us now analyze at the runtime of this algorithm. For each column ($O(N)$), we might have to add $O(N)$ times a column, and each addition takes $O(N)$. So, by this very rough analysis we have a runtime of $O(N^3)$. But, since the reduction process is very similar to Gaussian elimination, we can actually perform the reduction using techniques that yield a runtime of $O(N^\omega)$, where ω is the matrix-multiplication exponent. However, in practice this is not very useful since efficient matrix-multiplication algorithms are very complex and have large constants, while the naive implementation runs in essentially $O(N)$ time anyways since the involved matrices are so sparse.

Exercise 4.7. *Consider the following simplicial complex, and the simplex-wise filtration which first inserts the vertices in the order* a, b, c, d, e*, and the rest of the simplices as specified by the numbering in Figure 4.3.*

Execute both the persistence pairing algorithm and matrix reduction algorithm on this filtration. What are the similarities and differences in the algorithms? To

---

Algorithm 2: The matrix reduction algorithm.

Input: A filtration of K.

Find an ordering $\sigma_1, \ldots, \sigma_N$ corresponding to a simplex-wise filtration of K consistent with the given filtration;

$M := 0^{N \times N}$;

for $1 \leqslant i, j \leqslant N$ **do**

    if $\sigma_i \in \delta\sigma_j$ **then**

        $M_{ij} := 1$;

    end

end

for $j = 1$ *to* n **do**

    $\ell := \max(\{-1\} \cup \{i \mid M_{ij} = 1\})$;

    while $\ell \neq -1$ *and* $\exists j' < j$ *such that* $\ell = \max(\{-1\} \cup \{i \mid M_{ij'} = 1\})$ **do**

        $M_{\cdot j} := M_{\cdot j} + M_{\cdot j'}$;

        $\ell := \max(\{-1\} \cup \{i \mid M_{ij} = 1\})$;

    end

end

for $j = 1$ *to* n **do**

    if $M_{\cdot j} = 0^N$ **then**

        Label σ_j a constructor;

        for $j' = 1$ *to* n **do**

            if $j = \max(\{-1\} \cup \{i \mid M_{ij'} = 1\})$ **then**

                Pair σ_j to $\sigma_{j'}$;

                Label $\sigma_{j'}$ a destructor;

            end

        end

    end

end

Pair all unpaired constructors with ∞;

---

better see what happens, label the columns in the matrix by the sum of columns they currently represent.

Represent the results you obtained by a persistence diagram, and also by the persistence barcodes.
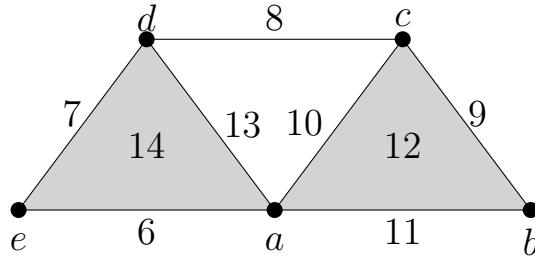
Figure 4.3: *The filtration for Exercise 4.7.*

Exercise 4.8. *A Union-Find data structure is a data structure that maintains disjoint sets dynamically. Given a ground set X, such a data structure maintains a family S of disjoint subsets of X, where each subset is represented by the* smallest *element contained in it. It supports three operations: MakeSet(x) creates a new set {x}. FindSet(x) returns the representative (minimum) of the set in S which contains x (or "no" if x is not contained in any set). Union(x,y) merges the sets containing x and y into a single one. All of these operations can be implemented in amortized $\Theta(\alpha(n))$ time, where α is the extremely slowly growing inverse Ackermann function and can be considered a constant for any real world application.*

Consider a simplicial complex K with its vertices ordered $v_0,\dots,v_n$, and consider its lower star filtration. Find an algorithm to compute the 0-dimensional persistence diagram (i.e., the persistence pairings) of K which makes use of a Union-Find data structure. How many Union-Find operations do you need to perform?

Questions

13. *What is a filtration?* State the definition and describe different ways how filtrations appear in topology and data analysis.

14. *What is persistent homology?* State the formal definitions and give examples.

15. *How can persistent homology be computed?* Discuss the two algorithms described in Section 4.3.

References

[1] Tamal Krishna Dey and Yusu Wang, *Computational topology for data analysis*, Cambridge University Press, 2022.

Chapter 5

Simplicial Complexes on Point Clouds

In general, the data we wish to analyze will not come in the form of a simplicial filtration, so in order to use persistent homology we need to transform our data into one. Ideally, the way we do this should retain the underlying shape of the data we want to analyze. In this section we discuss several ways of constructing simplicial complexes from point cloud data, and more generally, from finite metric spaces (i.e., a finite set of data points with given pairwise distances).

5.1 Čech and Vietoris-Rips complexes

Definition 5.1. *Given a metric space* (M, d)*, a finite point set* $P \subseteq M$*, and a real number radius* $r > 0$*, the* Čech complex $\mathbb{C}^r(P)$ *is defined as the nerve of the set of balls* $B(p, r) = \{x \in M \mid d(p, x) \leqslant r\}$ *for all* $p \in P$*.*

The Čech complex has the nice property that (at least for some metric spaces M including Euclidean space $\mathbb{R}^d$) by the Nerve theorem, it is homotopy equivalent to the union of the balls $B(p, r)$. In particular, for nice radii, it will capture the underlying shape. Sadly, checking whether a large number of balls have a common intersection can be computationally expensive. Further, the definition requires that the data points are embedded in a metric space. These two issues motivate the next definition.

Definition 5.2. *Given a finite metric space* (P, d) *and a real number radius* $r > 0$*, the* Vietoris-Rips complex $\mathbb{VR}^r(P)$ *is defined as the simplicial complex containing a simplex* σ *if and only if* $d(p, q) \leqslant 2r$ *for every pair* $p, q \in \sigma$*.*

Clearly, for finite subsets of metric spaces, by definition, the Čech complex and the Vietoris-Rips complex for the same radius and the same point set have the same set of 1-simplices (the same 1-skeleton). While the Čech complex then contains additional information about the common intersections of balls, the Vietoris-Rips complex is simply the clique complex of this 1-skeleton. This makes the Vietoris-Rips complex easier to compute. Furthermore, we make the following simple observation, showing that the Vietoris-Rips complex still approximately captures shapes in the data:

Observation 5.3. $\mathbb{C}^r(P) \subseteq \mathbb{VR}^r(P) \subseteq \mathbb{C}^{2r}(P)$.

Exercise 5.4. *Prove Observation 5.3.*

Exercise 5.5. *Find a point set $P \subset \mathbb{R}^2$ and a radius r such that its Vietoris-Rips complex has non-trivial 2-homology, i.e., such that $H_2(\mathbb{VR}^r(P)) \not\cong 0$.*
Furthermore, is there a dimension k such that $H_{k'}(\mathbb{VR}^r(Q)) = 0$ for all $k' \geqslant k$, all $r > 0$, and all point sets $Q \subset \mathbb{R}^2$?

5.2 Delaunay and Alpha complexes

Recall that computing persistent homology takes $O(N^3)$ time, where N is the size of the simplicial complex in the filtration. For large enough radii, both the Čech and the Vietoris-Rips complex become complete, and thus contain 2^n simplices. Computing persistent homology using those complexes is therefore computationally very expensive, which is why in many applications we would like to have sparser complexes. For data in $\mathbb{R}^d$ we can look at the so-called Delaunay triangulation, which only has complexity $O(n^{\lceil d/2 \rceil})$.

Definition 5.6. *Given a finite point set $P \subset \mathbb{R}^d$, a* Delaunay simplex *is a geometric simplex whose vertices are in P and lie on the boundary of a ball whose interior contains no points of P.*

A Delaunay triangulation $\mathrm{Del}(P)$ *of P is a geometric simplicial complex with the vertex set P where every simplex is a Delaunay simplex and whose underlying space covers the convex hull of P.*

Given a finite point set $P \subset \mathbb{R}^d$, the extended Delaunay complex *is the simplicial complex where for every face σ, for $d' \leqslant d$, every d'-face of σ is a Delaunay simplex.*

It is a well-known fact that for a point set in general position (no $d + 2$ points lie on a common sphere), there is a unique Delaunay triangulation. Furthermore, in this case the extended Delaunay complex and this unique Delaunay triangulation coincide.

Definition 5.7. *Given a finite point set $P \subset \mathbb{R}^d$, the* Voronoi diagram *is the tessellation of $\mathbb{R}^d$ into the* Voronoi cells

$$V_p = \{x \in \mathbb{R}^d \mid d(x,p) \leqslant d(x,q) \forall q \in P\}$$

for all $p \in P$.

Fact 5.8. *The nerve of the Voronoi cells of P is the extended Delaunay complex of P.*

Exercise 5.9. *Convince yourself that for a point set in $\mathbb{R}^2$, the nerve of the Voronoi diagram is the extended Delaunay complex. Furthermore, convince yourself that if the points are in general position (there are no three points that are collinear, and no four points that are cocircular), then there is a unique Delaunay triangulation.*

Based on the Delaunay triangulation, we define the *Alpha complex* by parameterizing using a radius as follows:

Definition 5.10. *Given a finite point set* $P \subset \mathbb{R}^d$ *in general position as well as a real number radius* $r > 0$, *the* Alpha complex $Del^r(P)$ *consists of all simplices* $\sigma \in Del(P)$ *for which the circumscribing ball of* σ *has radius at most* r.

The following fact provides us with an alternative definition of the Alpha complex:

Fact 5.11. *The Alpha complex* $Del^r(P)$ *is the nerve of the sets* $B(p, r) \cap V_p$ *for all* $p \in P$.

Since the Alpha complex is a subset of the Delaunay triangulation (and for large enough radius is equal to the Delaunay triangulation), it also has complexity $O(n^{\lceil d/2 \rceil})$. Further, the above fact together with the Nerve theorem implies that the Alpha complex $Del^r(P)$ is homotopy equivalent to the Čech complex $\mathbb{C}^r(P)$.

Exercise 5.12. *Is the following true or false? Consider a point set* $P \subset \mathbb{R}^2$ *in general position and a radius* $r > 0$. *Then the Alpha complex (with radius* r) *is the intersection of the Čech complex (with radius* r) *with the Delaunay triangulation.*

5.3 Subsample Complexes

For many applications, the Alpha complex is still too large. It is further expensive to compute, as computing a Delaunay triangulation in $\mathbb{R}^d$ takes $O(n^{\lceil d/2 \rceil})$ time. Sparser complexes can be constructed by looking at subsamples of the data, and relating the rest of the data to these subsamples. In the following, we will discuss two examples of complexes based on this idea.

Definition 5.13. *Given a finite point set* Q *and a point set* $P \supset Q$ *in some metric space, we say that a simplex* $\sigma \subseteq Q$ *is* weakly witnessed *by* $x \in P \setminus Q$, *if* $d(q, x) \leqslant d(p, x)$ *for every* $q \in \sigma$ *and* $p \in Q \setminus \sigma$.

Note that the set of weakly witnessed simplices is not downwards closed. We thus define a simplicial complex by requiring that all faces are weakly witnessed:

Definition 5.14. *The* Witness complex $\mathbb{W}(Q, P)$ *is the collection of simplices on* Q *for which every face is weakly witnessed by some point in* $P \setminus Q$.

Note that if we take the metric space $\mathbb{R}^d$ and we let P be the whole $\mathbb{R}^d$, then $\mathbb{W}(Q, P) = Del(Q)$, and by definition we thus get in general that $\mathbb{W}(Q, P) \subseteq Del(Q)$.

To arrive at a filtration, we again have to introduce a parameter $r > 0$:

Definition 5.15. *Given a finite point set* Q *and a point set* $P \supset Q$ *in some metric space as well as a real number radius* $r > 0$, *the* parameterized *Witness complex* $\mathbb{W}^r(Q, P)$ *is a simplicial complex on* Q *defined as follows:*
Every point $p \in Q$ *defines a vertex in* $\mathbb{W}^r(Q, P)$. *Further, an edge* pq *is in* $\mathbb{W}^r(Q, P)$ *if it is weakly witnessed by* $x \in P \setminus Q$ *and* $d(p, x) \leqslant r$ *and* $d(q, x) \leqslant r$. *A simplex* $\sigma \subseteq Q$ *is in* $\mathbb{W}^r(Q, P)$ *if all its edges are.*

Note that from this definition it is not guaranteed that the parameterized Witness complex is a subcomplex of the Witness complex.

The idea of the parameterized Witness complex is that it should approximate the Vietoris-Rips complex on P. There are theoretical guarantees about this approximation for manifolds of dimension at most 2, but the parameterized witness complex may fail to capture the topology of manifolds in dimension 3 and above.

Exercise 5.16. *Show that* $\mathbb{W}(Q, P) \subseteq \mathbb{W}(Q, P')$ *for* $P \subseteq P'$. *On the other hand, give an example of point sets* $Q \subseteq Q'$ *and* P *for which* $\mathbb{W}(Q, P) \not\subseteq \mathbb{W}(Q', P)$.

Let us now consider a second subsample complex, the graph induced complex.

Definition 5.17. *Given two finite point sets* Q, P *in* $\mathbb{R}^d$, *as well as a graph* $G(P)$ *with vertices in* P, *we define* $v : P \to Q$ *by sending each point in* P *to its closest point in* Q. *The* graph induced complex $\mathbb{G}(Q, G(P))$ *contains a simplex* $\sigma = \{q_0, \ldots, q_k\} \subset Q$ *if and only if there is a clique* $\{p_0, \ldots, p_k\}$ *in* $G(P)$ *for which* $v(p_i) = q_i$.

We again parameterize this:

Definition 5.18. *Let* $G^r(P)$ *be the graph on* P *where* pq *is an edge if and only if* $d(p, q) \leqslant 2r$. *The* parameterized graph induced complex $\mathbb{G}^r(Q, P)$ *is defined as* $\mathbb{G}(Q, G^r(P))$.

This complex again has theoretical guarantees of approximating the Vietoris-Rips complex on $P \cup Q$.

Exercise 5.19. *Let* P, Q *be point sets and* $G(P)$ *a graph with* P *as its vertex set. Let* $v : P \to Q$ *be the map sending each point of* P *to its closest point of* Q *(assume that this closest point is always unique). Let* C *be the clique complex of* $G(P)$ *(the complex which includes a simplex iff its corresponding vertices in* $G(P)$ *form a clique).*

Show that v *extends to a simplicial map* $\bar{v} : C \to \mathbb{G}(Q, G(P))$. *Also show that any simplicial complex* K *with* $V(K) = Q$ *for which* v *has a simplicial extension must contain* $\mathbb{G}(Q, G(P))$.

Questions

16. *What are the Čech and Vietoris-Rips complexes?* Give the definitions, discuss their size and theoretical guarantees, and how they are related.

17. *What are the Delaunay and Alpha complexes?* Give the definitions, discuss their size and theoretical guarantees, and how they are related.

18. *What is the Witness complex?* State the Definition and describe how it relates to the non-sparse complexes.

19. *What is the Graph induced complex?* State the Definition and describe how it relates to the non-sparse complexes.

Chapter 6

Distances and Stability

6.1 Distance Metrics on Persistence Diagrams

As we have seen in the previous sections, persistent homology of simplicial filtrations, for example in the form of persistence diagrams or persistence barcodes, can give us a lot of insight into a given point cloud. However, so far we have always been analyzing this information manually. In this section we will show how this can be done on a more mathematical level, by defining some distance metrics that can be used to compare different persistence diagrams, and thus to assess the similarity of point clouds.

6.1.1 Bottleneck Distance

Let $\mathcal{F}, \mathcal{G}$ be two filtrations giving rise to persistence modules $H_p(\mathcal{F}), H_p(\mathcal{G})$, and let $\mathrm{Dgm}_p(\mathcal{F})$ and $\mathrm{Dgm}_p(\mathcal{G})$ be their corresponding persistence diagrams. How can we now compare these two filtrations $\mathcal{F}$ and $\mathcal{G}$ by using only the information stored in these diagrams?

The general idea of the bottleneck distance is to find a matching between the points of the two persistence diagrams, i.e., we consider bijections between the points of $\mathrm{Dgm}_p(\mathcal{F})$ and those of $\mathrm{Dgm}_p(\mathcal{G})$. Since we can only find bijections between sets of the same cardinality, we need the two diagrams to have the same number of points. Recall that the way we defined it, a persistence diagram includes every possible point on the diagonal with infinite multiplicity. This peculiar definition now finally pays off; since both sets of points have the same (infinite) cardinality, and bijections between these sets are thus well-defined.

We do not want to consider any arbitrary bijection, but only the *best possible*. To measure the "quality" or "distance" of such a bijection, we use the L_∞-norm:

Definition 6.1. *Let* $x = (x_1, x_2), y = (y_1, y_2)$ *be two points in* $\mathbb{R}^2$. *Then,*

$$\|x - y\|_\infty := \max(|x_1 - y_1|, |x_2 - y_2|),$$

where we say that $\infty - \infty = 0$ *for points with coordinates that are* ∞ *(i.e., points in persistence diagrams that correspond to holes that did not die).*

Definition 6.2. *Let* $\Pi = \{\pi : \mathrm{Dgm}_p(\mathcal{F}) \to \mathrm{Dgm}_p(\mathcal{G}) \mid \pi \text{ is bijective}\}$ *be the set of all bijections between* $\mathrm{Dgm}_p(\mathcal{F})$ *and* $\mathrm{Dgm}_p(\mathcal{G})$. *Then, the* Bottleneck distance *is defined as*

$$d_b(\mathrm{Dgm}_p(\mathcal{F}), \mathrm{Dgm}_p(\mathcal{G})) := \inf_{\pi \in \Pi} \sup_{x \in \mathrm{Dgm}_p(\mathcal{F})} \|x - \pi(x)\|_\infty.$$

The Bottleneck distance thus minimizes the maximum L_∞-norm of any pairing, over all pairings of points.
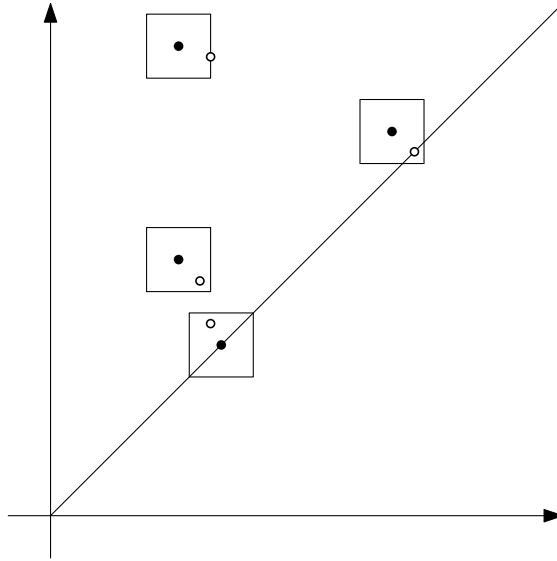

Figure 6.1: *An illustration of the idea of bottleneck distance.*

Observation 6.3. *The Bottleneck distance is a metric on the space of persistence diagrams with finitely many off-diagonal points.*

Proof. We check the three properties of metrics:

1. $d_b(X, Y) = 0$ if and only if $X = Y$: This is simple to see, since if $X = Y$, every point can be matched to its copy, and if $X \neq Y$, there exists some point $p \in X \setminus Y \cup Y \setminus X$ which must be matched to some point with positive L_∞-distance to p.

2. $d_b(X, Y) = d_b(Y, X)$: This is clear by definition.

3. $d_b(X, Y) \leqslant d_b(X, Z) + d_b(Z, Y)$: Take a bijection π_1 witnessing[1] $d_b(X, Z)$ and a bijection π_2 witnessing $d_b(Z, Y)$, and concatenate the two: $\pi := \pi_2 \circ \pi_1$ is a bijection $X \to Y$ where for every $x \in X$ we can use the triangle equality of $\|\cdot\|_\infty$ to bound $\|x - \pi(x)\|_\infty \leqslant \|x - \pi_1(x)\|_\infty + \|\pi_1(x) - \pi_2(\pi_1(x))\|_\infty$. $\qquad\square$

---

[1]Note that since d_b is an infimum and not a minimum, there may not be π_1 and π_2 witnessing d_b. In this case, the same argument can be applied to the converging sequences of bijections witnessing d_b.

Exercise 6.4. *Give an algorithm to compute the Bottleneck distance between two persistence diagrams. Your algorithm should be polynomial in n, where n is the total number of off-diagonal points in the two persistence diagrams.*

Exercise 6.5. *Let $\mathcal{D}_1, \ldots, \mathcal{D}_k$ be k persistence diagrams. Assume that for any two of them we have $d_b(\mathcal{D}_i, \mathcal{D}_j) \leqslant 2$. Prove or disprove: there exists a persistence diagram $\mathcal{D}_0$ such that $d_b(\mathcal{D}_0, \mathcal{D}_i) \leqslant 1$ for all $i \in \{1, \ldots, k\}$.*

The Bottleneck distance can be used to compare any two filtrations, of possibly wildly different spaces. What if we consider two filtrations of the same simplicial complex? Recall that simplex-wise monotone functions $f, g : K \to \mathbb{R}$ give rise to simplicial sublevel set filtrations $\mathcal{F}_f, \mathcal{F}_g$. While these two filtrations can be compared using the Bottleneck distance, we can also define a metric that directly compares the two functions f, g:

Definition 6.6 (infinity norm). *Let $f, g : X \to \mathbb{R}$. Then, the infinity norm of $f - g$ is defined as*

$$\|f - g\|_\infty := \sup_{x \in X} |f(x) - g(x)|.$$

The following theorem tells us that this infinity norm and the Bottleneck distance are closely related:

Theorem 6.7 (Stability for simplicial filtrations). *Let $f, g : K \to \mathbb{R}$ be simplex-wise monotone functions. Then, $\forall p \geqslant 0$ we have $d_b(\mathrm{Dgm}_p(\mathcal{F}_f), \mathrm{Dgm}_p(\mathcal{F}_g)) \leqslant \|f - g\|_\infty$.*

Proof. We consider the linear interpolation $f_t := (1-t)f + tg$ for $t \in [0, 1]$ between f and g. Note that $f_0 = f, f_1 = g$.

We first show that each f_t is a simplex-wise monotone function: Let $\sigma \subseteq \tau$. Since f and g are monotone, we have $f(\sigma) \leqslant f(\tau)$ and $g(\sigma) \leqslant g(\tau)$. Thus,

$$f_t(\sigma) = (1-t)f(\sigma) + tg(\sigma) \leqslant (1-t)f(\tau) + tg(\tau) = f_t(\tau).$$

Let $p \geqslant 0$ be fixed. We now draw the family of persistence diagrams $\mathrm{Dgm}_p(\mathcal{F}_{f_t})$ as a multiset in $\mathbb{R}^2 \times [0, 1]$. Each off-diagonal point of $X_t := \mathrm{Dgm}_p(\mathcal{F}_{f_t})$ is of the form $x(t) = (f_t(\sigma), f_t(\tau), t)$ for σ being the creator and τ being the destructor. Note that the persistence pairings (σ, τ) may only change when the order of simplex insertion changes, which only happens finitely many times when going from $t = 0$ to $t = 1$. Let us call these values $0 = t_0 < t_1 < t_2 < \ldots < t_n < t_{n+1} = 1$. For simplicity, we assume that at each of these values t_i exactly two simplices have the same value f_{t_i}.
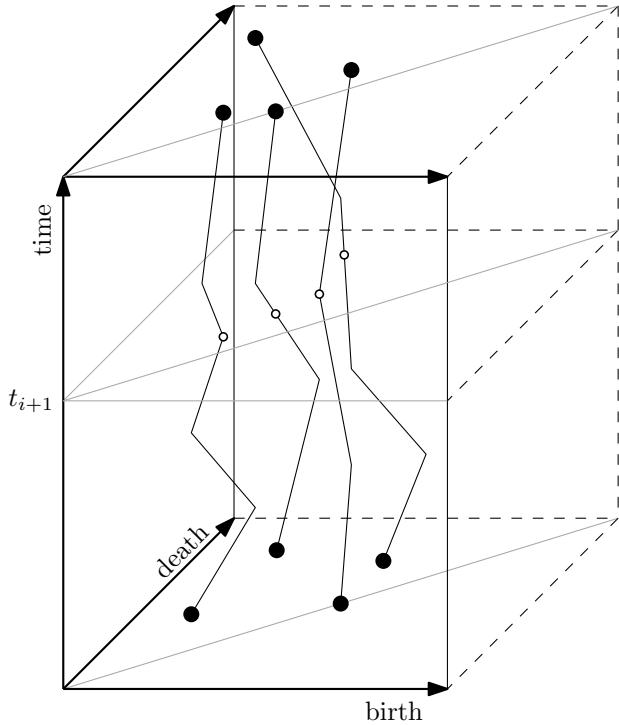
Within each open interval (t_i, t_{i+1}) the pairings stay constant. Furthermore, every off-diagonal point $x(t)$ is a linear function of t in all three coordinates, meaning that it defines a line segment.

Let us first assume that at some value t_{i+1}, $x(t_{i+1})$ is an off-diagonal point whose creator and destructor are still paired after t_{i+1}. In this case, $x(t)$ continues in the same direction after t_{i+1}.

If on the other hand $x(t_{i+1})$ is an off-diagonal point whose creator and destructor get paired differently, recalling Exercise 4.4, there are exactly two pairs that swap their creators or destructors, and these creators or destructors that are swapped must have the same value in $f_{t_{i+1}}$. In the persistence diagram, this means that two points vertically *or* horizontally of each other swap creators/destructors. However, this just means that for both of these line segments going into t_{i+1}, there is a unique continuing line segment.

Note that for $t = 0$ or $t = 1$ we can also have that $x(t)$ lies on the diagonal. This means that its past/future creator and destructor have the same value in f_t.

Every point thus moves along a polygonal path monotone in t. Every such path is called a *vine*, and the multiset of all vines is called a *vineyard*, see Figure 6.2 for an illustration. Based on this vineyard, we now wish to find a good matching giving an upper bound on the Bottleneck distance. We simply take the matching where we match the start point of every vine with its endpoint. To get a bound on the Bottleneck distance, we simply need to get a bound for the distance of each matched pair.

Figure 6.2: *The vineyards in the proof of Theorem 6.7.*

Between any t_i and t_{i+1}, a point $x(t)$ moves at the rate $\frac{\delta x(t)}{\delta t}$, which we can compute to be

$$\frac{\delta}{\delta t} \left((1 - t) \cdot (f(\sigma), f(\tau), t) \right) + t \cdot (g(\sigma), g(\tau), t)) = (g(\sigma) - f(\sigma), g(\tau) - f(\tau), 1).$$

Projecting $x(t_{i+1})$ and $x(t_i)$ to $\mathbb{R}^2$ we thus get two points y_{i+1}, y_i for which we can see

that

$$\|y_{i+1} - y_i\|_\infty = (t_{i+1} - t_i) \cdot \max\left(|g(\sigma) - f(\sigma)|, |g(\tau) - f(\tau)|\right) \leqslant (t_{i+1} - t_i) \cdot \|f - g\|_\infty.$$

Thus, since $\|\cdot\|_\infty$ is a norm and fulfills the triangle inequality, we also have that from $t = 0$ to $t = 1$, the point can move at most $\|f - g\|_\infty$. We thus have the desired bound on the Bottleneck distance. $\qquad\qquad\qquad\square$

Exercise 6.8. *Show that Theorem 6.7 above can be tight for all $p \geqslant 0$ and all values of $\|f - g\|_\infty$.*

Exercise 6.9. *Let P and Q be subsets of a metric space. We say that $P \cup Q$ is in δ-separated position if for any $p \in P$ and $q_1, q_2 \in Q$ we have that $|d(p, q_1) - d(p, q_2)| > \delta$. Assume that $P \cup Q$ is in δ-separated position and let Q' be an ε-perturbation of Q, that is, there is a bijection between Q and Q' such that for every original point $q \in Q$ and its image $q' \in Q'$ we have $d(q, q') \leqslant \varepsilon$. Let $\mathcal{D}_p$ and $\mathcal{D}'_p$ be the persistence diagrams for the p-dimensional persistent homology of the filtration induced by the parameterized witness complexes $\mathbb{W}^r(Q, P)$ and $\mathbb{W}^r(Q', P)$, respectively. Show that if $\varepsilon < \delta/2$ then*

$$d_b(\mathcal{D}_p, \mathcal{D}'_p) \leqslant \varepsilon.$$

Further, show an example where this fails for $\varepsilon > \delta/2$.

We wish to generalize the stability result above from simplicial filtrations to filtrations of general topological spaces. To this end we consider some topological space X and a function $f : X \to \mathbb{R}$, which induces a sublevel set filtration for every $r \in \mathbb{R}$. We only want to consider *tame* functions: A function f is *tame* if all homology groups of sublevel sets have finite dimension, and the homology groups only change at finitely many values, called *critical values*.

Theorem 6.10. *Let X be a triangulable topological space, and $f, g : X \to \mathbb{R}$ be two tame functions. Then $\forall p \geqslant 0$, we have*
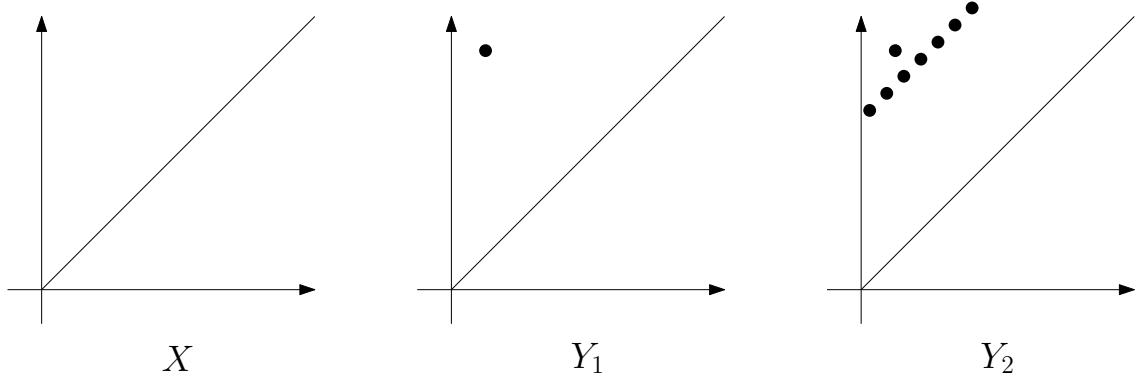
$$d_b(\mathrm{Dgm}_p(\mathcal{F}_f), \mathrm{Dgm}_p(\mathcal{F}_g)) \leqslant \|f - g\|_\infty.$$

We do not prove this theorem at this point, but with additional tools that we will develop in Section 6.2, the proof of this (and of Theorem 6.7) will follow quite easily.

6.1.2 Wasserstein Distance

Consider the following three diagrams:

Which of Y_1 and Y_2 is X closer to? Intuitively, one clearly says Y_1: There are simply fewer features in Y_1 that are not present in X. In terms of Bottleneck distance, there is only one reasonable matching between X and Y_1, and also only one between X and Y_2: We simply match each off-diagonal point with its closest point on the diagonal. Since

$$X \qquad\qquad\qquad Y_1 \qquad\qquad\qquad Y_2$$

we only look at the longest edge in this matching, the Bottleneck distance is actually the same for both pairs of diagrams, i.e., $d_b(X, Y_1) = d_b(X, Y_2)$.

We can get rid of this counter-intuitive behavior of the Bottleneck distance by using the Wasserstein distance.
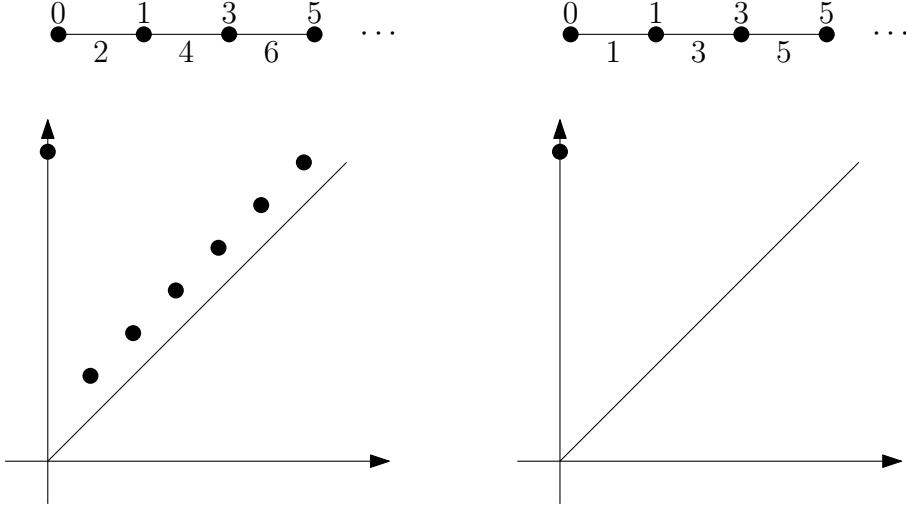
Definition 6.11 (Wasserstein distance). *For $p \geqslant 0$, and $q \geqslant 1$, the q-Wasserstein distance is defined as*

$$d_{W,q}(\mathrm{Dgm}_p(\mathcal{F}), \mathrm{Dgm}_p(\mathcal{G})) := \left[\inf_{\pi \in \Pi} \left(\sum_{x \in \mathrm{Dgm}_p(\mathcal{F})} (\|x - \pi(x)\|_\infty)^q \right) \right]^{1/q}$$

Intuitively, we now consider the length of all edges in the matching induced by the bijection, as opposed to just the longest one, but the longer ones get more weight. Note that for $q = \infty$, we retrieve the bottleneck distance, that is, $d_{W,\infty} = d_b$.

We can see that the stability theorem we proved for Bottleneck distance does not hold for Wasserstein distance: consider two simplex-wise monotone functions f and g on a path, as illustrated in Figure 6.3. In both f and g the first vertex on the path is mapped to 1 and the edges along the path are mapped to increasing odd numbers. In f the remaining vertices along the path get mapped to increasing even numbers, and in g to increasing odd numbers. In particular, $\|f - g\|_\infty = 1$. In the filtration defined by f, at every even step we add a vertex, creating a new connected component, which gets connected to the rest of the path at the next step. Thus, each vertex of the path will give an off-diagonal point in the 0-persistence diagram, where all of them except the first one have a lifespan of 1. On the other hand, in the filtration defined by g, we always add the new vertices and their connecting edge in the same step, thus the 0-persistence diagram only has a single off-diagonal point with infinite lifespan. In particular, we have that for arbitrarily long paths we get arbitrarily large Wasserstein distances between the diagrams for all $q < \infty$.

A similar counterexample can also be found for topological spaces. Consider the topological space $[0, 1]$ and the two functions depicted by the curves in Figure 6.4. Here we again have that $\|f - g\|_\infty \leqslant \epsilon$, but the Wasserstein distance between the two diagrams can be made arbitrarily big.

Figure 6.3: *Two simplex-wise monotone functions with bounded infinity norm whose persistence diagrams have unbounded Wasserstein distance.*

To avoid these types of counterexamples, we want to consider even nicer functions:

Definition 6.12 (Lipschitz). *Let* (X, d) *be a metric space. A function* $f : X \to \mathbb{R}$ *is Lipschitz if there exists a constant* C *such that* $|f(x) - f(y)| \leqslant c \cdot d(x, y)$ *for all* $x, y \in X$.

Exercise 6.13. *For each* $\ell \in \mathbb{R}$*, give two continuous functions* $f, g : [0, 1] \to \mathbb{R}$ *such that* $\|f - g\|_\infty \leqslant 1$*, but the Wasserstein distance between the diagrams of the sublevel set filtrations of* f, g *is at least* ℓ*. What happens to the Lipschitz constant of your functions as* $\ell \to \infty$*?*
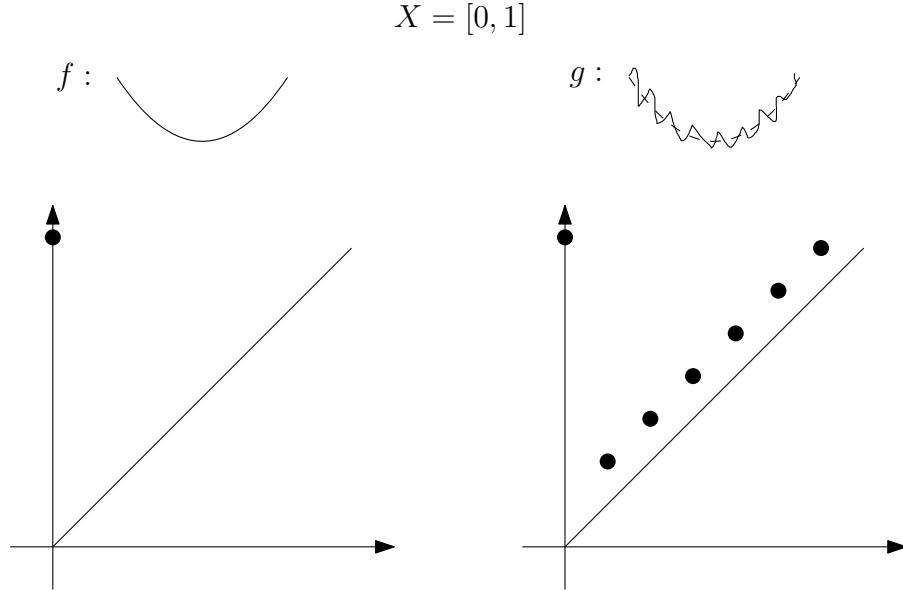
For Lipschitz-functions we again get stability theorems, however we will not prove these here.

Theorem 6.14. *Let* X *be a triangulable, compact metric space. Let* $f, g : X \to \mathbb{R}$ *be tame Lipschitz functions. Then there exist constants* C *and* k *(that may only depend on* X *and on the Lipschitz constants of* f, g*) such that for every* $p \geqslant 0$ *and every* $q \geqslant k$,

$$d_{W,q}(\mathrm{Dgm}_p(\mathcal{F}_f), \mathrm{Dgm}_p(\mathcal{F}_g)) \leqslant C \cdot \|f - g\|_\infty^{1 - k/q}.$$

Theorem 6.15. *Let* $f, g : K \to \mathbb{R}$ *be simplex-wise monotone functions. Then for all* $p \geqslant 0$ *and all* $q \geqslant 1$,

$$d_{W,q}(\mathrm{Dgm}_p(\mathcal{F}_f), \mathrm{Dgm}_p(\mathcal{F}_g)) \leqslant \|f - g\|_q = \left(\sum_{\sigma \in K} |f(\sigma) - g(\sigma)|^q \right)^{1/q}.$$

$$X = [0,1]$$

$f:$

$g:$

Figure 6.4: *Two functions $[0,1] \to \mathbb{R}$ with bounded infinity norm whose persistence diagrams have unbounded Wasserstein distance.*

6.2 Interleaving of Persistence Modules

6.2.1 Interleaving Distance

Until now, we only compared persistence diagrams. We will now introduce the interleaving distance, which instead compares persistence modules directly. Let us begin with a formal definition of persistence modules.

Definition 6.16. *A* persistence module $\mathbb{V}$ *over $\mathbb{R}$ is a collection $\mathbb{V} = \{V_a\}_{a \in \mathbb{R}}$ of vector spaces V_a together with linear maps $v_{a,a'} : V_a \to V_{a'}$ for $a \leqslant a'$, such that $v_{a,a} = \mathrm{id}$ and $v_{b,c} \circ v_{a,b} = v_{a,c}$ for all $a \leqslant b \leqslant c$.*

You already know a few examples of persistence modules, e.g., the persistent homology of sublevel set filtrations or of Čech or Vietoris-Rips complexes (here one simply defines $V_a = 0$ for $a < 0$).

To be able to define distances between persistence modules, we first need to figure out when we want to call two persistence modules "the same", or more formally speaking, we want to define a notion of isomorphism.

Definition 6.17. *We say that two persistence modules $\mathbb{U}$ and $\mathbb{V}$ are* isomorphic *if there are isomorphisms $f_a : U_a \to V_a$ for all $a \in \mathbb{R}$ such that*

$$
\begin{array}{ccc}
U_a & \xrightarrow{u_{a,a'}} & U_{a'} \\
\big\uparrow{\scriptstyle f_a} & & \big\uparrow{\scriptstyle f_{a'}} \\
V_a & \xrightarrow{v_{a,a'}} & V_{a'}
\end{array}
$$

commutes both ways, i.e., $f_{a'} \circ u_{a,a'} = v_{a,a'} \circ f_a$, and $u_{a,a'} \circ f_a^{-1} = f_{a'}^{-1} \circ v_{a,a'}$.

The basic idea of interleaving distance is to measure how close two persistence modules are to being isomorphic. For this, we allow ourselves some slack, in the sense that U_a does not need to map to V_a, but it can instead map to $V_{a+\epsilon}$, as long as all the relevant maps still behave like they would for an isomorphism. We make this formal in the next definition.

Definition 6.18 (ϵ-interleaving persistence modules). *Let $\mathbb{U}$ and $\mathbb{V}$ be persistence modules over $\mathbb{R}$ and let $\epsilon \geqslant 0$. We say that $\mathbb{U}$ and $\mathbb{V}$ are ϵ-interleaved if there exist two families of linear maps, $\varphi_a : U_a \to V_{a+\epsilon}$ and $\psi_a : V_a \to U_{a+\epsilon}$ such that the following four diagrams are commutative:*

$$
\begin{array}{ccc}
U_a & \xrightarrow{\;u_{a,a'}\;} & U_{a'} \\
\downarrow{\scriptstyle\varphi_a} & & \downarrow{\scriptstyle\varphi_{a'}} \\
V_{a+\epsilon} & \xrightarrow{\;v_{a+\epsilon,a'+\epsilon}\;} & V_{a'+\epsilon}
\end{array}
\qquad and \qquad
\begin{array}{ccc}
U_{a+\epsilon} & \xrightarrow{\;u_{a+\epsilon,a'+\epsilon}\;} & U_{a'+\epsilon} \\
{\scriptstyle\psi_a}\nearrow & & {\scriptstyle\psi_{a'}}\nearrow \\
V_a & \xrightarrow{\;v_{a,a'}\;} & V_{a'}
\end{array}
$$

$$
\begin{array}{ccc}
U_a & \xrightarrow{\;u_{a,a+2\epsilon}\;} & U_{a+2\epsilon} \\
{\scriptstyle\varphi_a}\searrow & & {\scriptstyle\psi_{a+\epsilon}}\nearrow \\
& V_{a+\epsilon} &
\end{array}
\qquad and \qquad
\begin{array}{ccc}
& U_{a+\epsilon} & \\
{\scriptstyle\psi_a}\nearrow & & {\scriptstyle\varphi_{a+\epsilon}}\searrow \\
V_a & \xrightarrow{\;v_{a,a+2\epsilon}\;} & V_{a+2\epsilon}
\end{array}
$$

Note that if $\mathbb{U}$ and $\mathbb{V}$ are isomorphic, then they are 0-interleaved: the first type of diagrams (the *square diagrams*) are the commutative diagrams in the definition of isomorphic persistence modules and the the second type of diagrams (the *triangular diagrams*) collapse to two arrows that say that the maps φ_a are isomorphisms with inverses ψ_a.

Theorem 6.19. *Assume $\mathbb{U}$ and $\mathbb{V}$ are ϵ-interleaved. Let $\delta > \epsilon$. Then $\mathbb{U}$ and $\mathbb{V}$ are also δ-interleaved.*

Proof. Given $\varphi_a' : U_a \to V_{a+\epsilon}$ we define $\varphi_a : U_a \to V_{a+\delta}$ simply as $\varphi_a := v_{a+\epsilon,a+\delta} \circ \varphi_a'$. Symmetrically, we define $\psi_a := u_{a+\epsilon,a+\delta} \circ \psi_a'$. To check that the correct diagrams commute, we only check the right diagram of every pair of symmetric diagrams shown above. We have to distinguish two cases for the first diagram, $a + \delta < a' + \epsilon$ and $a + \delta > a' + \epsilon$.

For the first case, we get the following diagram:

$$
\begin{array}{ccccc}
U_a & \xrightarrow{\hspace{4cm}} & U_{a'} & & \\
\downarrow & \searrow & & \searrow & \\
& V_{a+\epsilon} \rightarrow V_{a+\delta} & \xrightarrow{\hspace{2cm}} & V_{a'+\epsilon} \rightarrow & V_{a'+\delta}
\end{array}
$$

For the second case we get the diagram:

$$
\begin{array}{ccccccc}
U_a & \longrightarrow & U_{a'} & & & & \\
& \searrow & & \searrow & & & \\
& V_{a+\epsilon} & \longrightarrow & V_{a'+\epsilon} & \dashrightarrow & V_{a+\delta} & \longrightarrow & V_{a'+\delta}
\end{array}
$$

And finally, for the triangular diagram we get:

$$
\begin{array}{ccccccc}
U_a & \longrightarrow & U_{a+2\epsilon} & \longrightarrow & U_{a+\delta+\epsilon} & \longrightarrow & U_{a+2\delta} \\
& \searrow & & \nearrow & & \nearrow & \\
& V_{a+\epsilon} & \longrightarrow & V_{a+\delta} & & &
\end{array}
$$

One can now verify that in all of these diagrams the correct paths commute. $\square$

Thus, the following definition makes sense:

Definition 6.20 (Interleaving distance). $d_I(\mathbb{U}, \mathbb{V}) := \inf\{\epsilon \mid \mathbb{U} \text{ and } \mathbb{V} \text{ are } \epsilon\text{-interleaved }\}$.

Exercise 6.21. *Show that interleaving distance is a pseudo-metric for persistence modules (up to isomorphism), i.e., prove that (i) the interleaving distance between isomorphic persistence modules is 0, (ii) the interleaving distance is non-negative, and (iii) the interleaving distance fulfills the triangle inequality.*

Also show that it is not *a metric by showing that there exist non-isomorphic persistence modules with interleaving distance 0.*

Exercise 6.22. *Let W_1 and W_2 be two arbitrary vector spaces. Let $\mathbb{U}$ be the persistence module such that $U_a = W_1$ for $a \in [w, x)$, and $U_a = 0$, otherwise. For $a, a' \in [w, x)$ we have $u_{a,a'}$ being the identity map. For $a < w$ or $a' \geqslant x$ (or both), we have $u_{a,a'}$ being the zero map. Similarly, we define the persistence module $\mathbb{V}$ which is W_2 in $a \in [y, z)$ and 0 otherwise.*
Show that $d_I(\mathbb{U}, \mathbb{V}) \leqslant \max(\frac{w-x}{2}, \frac{z-y}{2})$.

The underlying ideas that allowed us to define the interleaving distance of persistence modules can also be applied to filtrations.

Definition 6.23 (Interleaving for Filtrations). *Let $\mathcal{F}, \mathcal{G}$ be filtrations over $\mathbb{R}$. $\mathcal{F}$ and $\mathcal{G}$ are ϵ-interleaved if there exist maps $\varphi_a : F_a \to G_{a+\epsilon}$ and $\psi_a : G_a \to F_{a+\epsilon}$ such that the same type of diagrams commute up to homotopy, that is, for example $\varphi_{a'} \circ \iota^F_{a,a'} \simeq \iota^G_{a+\epsilon, a'+\epsilon} \circ \varphi_a$ are homotopic (contiguous).*
We again define the interleaving distance (now between filtrations):

$$d_I(\mathcal{F}, \mathcal{G}) = \inf\{\epsilon \mid \mathcal{F} \text{ and } \mathcal{G} \text{ are } \epsilon\text{-interleaved }\}.$$

From induced homology, we immediately get the following observation:

Observation 6.24. *For all* $p \geqslant 0$, $d_I(H_p\mathcal{F}, H_p\mathcal{G}) \leqslant d_I(\mathcal{F}, \mathcal{G})$.

As a first application of interleaving distance, we can quantify how different the Čech and Vietoris-Rips filtrations are. Recall that for a point cloud P and a radius r, we have the relationship between the Čech and Vietoris-Rips complexes as follows: $\mathbb{C}^r(P) \subseteq \mathbb{VR}^r(P) \subseteq \mathbb{C}^{2r}(P)$. Since this factor 2 is multiplicative, and we need an additive ϵ for interleaving, let us just take the logarithmic scale (base 2) for the radius, i.e., we define $\mathbb{C}^r_{\log} = \mathbb{C}^{2^r}$ and similarly $\mathbb{VR}^r_{\log} = \mathbb{VR}^{2^r}$. Since $2^{(r+1)} = 2 \cdot 2^r$, we have $\mathbb{C}^r_{\log}(P) \subseteq \mathbb{VR}^r_{\log}(P) \subseteq \mathbb{C}^{r+1}_{\log}(P)$.

We thus have the following inclusions:

$$
\begin{array}{ccccc}
\mathbb{C}^r_{\log} & \longrightarrow & \mathbb{C}^{r+1}_{\log} & \longrightarrow & \mathbb{C}^{r+2}_{\log} \\
& \times & & \times & \\
\mathbb{VR}^r_{\log} & \longrightarrow & \mathbb{VR}^{r+1}_{\log} & \longrightarrow & \mathbb{VR}^{r+2}_{\log}
\end{array}
$$

Since these are all inclusions, all relevant diagrams must commute, and thus we get that $d_I(\mathbb{C}_{\log}, \mathbb{VR}_{\log}) \leqslant 1$.

6.2.2 Stability with Respect to Interleaving Distance

The main motivation for interleaving distance is that it can be used to prove stability results, at least under some tameness conditions.

Definition 6.25. *A persistence module* $\mathbb{V}$ *is* q-tame *if the linear maps have finite dimension.*

Note that in this definition, the q is not a parameter, just a name. All persistence modules that show up in the context of persistent homology on point clouds are q-tame, so this condition is not restrictive.

Theorem 6.26. *If* $\mathbb{U}, \mathbb{V}$ *are q-tame persistence modules over* $\mathbb{R}$*, then*

$$d_b(\mathrm{Dgm}\,\mathbb{U}, \mathrm{Dgm}\,\mathbb{V}) = d_I(\mathbb{U}, \mathbb{V}).$$

Thus, for every interleaving one can find between two persistence modules or between filtrations, one immediately gets an upper bound on the Bottleneck distance. This is a very powerful result, and the proof of it is out of scope for these lecture notes. One direction of the proof however follows from a decomposition result of persistence modules that we will discuss in Section 6.3. But first, we will consider some examples of how Theorem 6.26 can be used to prove stability theorems.

Exercise 6.27. *Prove Theorem 6.10.*

6.2.3 Stability for Čech Complexes

So far, we have only seen stability results comparing filtrations induced by different functions on a fixed space. However, in applications in data analysis, we consider complexes on point clouds, and two different point clouds might not have the same size. Thus, the simplicial complexes on which we get filtrations are generally different. Using interleaving distance, we can however still give stability results. In this section, we will do this for Čech complexes.

Consider two point clouds P, Q in the same metric space X. Let us first consider the really simple case, where $P = \{p\}$, and $Q = \{q\}$ with $d(p, q) = d$. Then, $B(p, r) \subseteq B(q, r + d)$. Now, how does this generalize to larger point sets? To get the same kind of behavior, we need that for every point in P, there exists some point in Q with distance at most d. This motivates the following distance measure:

Definition 6.28 (Hausdorff distance). *Let $A, B \subseteq X$ be compact sets. Then the Hausdorff distance between A and B is defined as*

$$d_H(A, B) := \max\{\max_{a \in A} d(a, B), \max_{b \in B} d(b, A)\}.$$

Exercise 6.29. *Show that Hausdorff distance is a metric.*

We can now see that if $d_H(P, Q) = d$, then $\bigcup_{p \in P} B(p, r) \subseteq \bigcup_{q \in Q} B(q, r + d)$. From this, we get the following lemma:

Lemma 6.30. *For P, Q with $d_H(P, Q) = d$, the (filtrations given by) the Čech complexes of P and Q are d-interleaved.*

Proof. Consider the following diagram:

The relevant sub-diagrams commute up to homotopy, since we only chain together homotopies and inclusion maps. Note that for the trapezoidal diagrams we would need to consider the horizontal inclusions between the Čech complexes in the diagram above for arbitrary distances and not just d. □

We can conclude the following

Theorem 6.31. $d_b(\mathrm{Dgm}_p(\mathbb{C}(P)), \mathrm{Dgm}_p(\mathbb{C}(Q))) \leqslant d_H(P, Q)$ *for all* $p \geqslant 0$.

Proof. By Theorem 6.26, Observation 6.24, and finally Lemma 6.30, we have

$$d_b(\ldots) = d_I(H_p\mathbb{C}(P), H_p\mathbb{C}(Q)) \leqslant d_I(\mathbb{C}(P), \mathbb{C}(Q)) \leqslant d_H(P, Q). \qquad \square$$

6.3   Interval Decomposition of Persistence Modules

In this section, we again look at persistence modules, this time as algebraic structures. We consider persistence modules over $\mathbb{R}$ of vector spaces over some field $\mathbb{F}$. We start by looking at some special persistence modules, called *interval modules*.

Definition 6.32. *An interval module* $\mathbb{I}[b, d]$ *is a persistence module*

$$V_a = \begin{cases} \mathbb{F} & \text{if } a \in [b, d], \\ 0 & \text{otherwise.} \end{cases} \qquad \text{and} \qquad v_{a,a'} = \begin{cases} \mathrm{id} & b \leqslant a \leqslant a' \leqslant d, \\ 0 & \text{otherwise.} \end{cases}$$

Similarly, we can define interval modules on open and clopen intervals, denoted by $\mathbb{I}(b, d)$, $\mathbb{I}(b, d]$, and $\mathbb{I}[b, d)$. We write $\mathbb{I}\langle b, d\rangle$ to include all four of these types.

For an interval module we can easily talk about birth and death as we did in persistent homology. If we have a persistent homology module that is (isomorphic to) an interval module, the birth and death correspond to the boundaries b, d of the interval.

Definition 6.33. *A persistence module* $\mathbb{U}$ *is called* pointwise finite dimensional (p.f.d.) *if for all* $a \in \mathbb{R}$, U_a *has finite dimension.*

Note that all p.f.d. persistence modules are also q-tame.

Definition 6.34. *Given two persistence modules* $\mathbb{U}, \mathbb{V}$, *we define their direct sum* $\mathbb{U} \oplus \mathbb{V}$ *by* $(\mathbb{U} \oplus \mathbb{V})_a = U_a \oplus V_a$ *and* $(u \oplus v)_{a,a'} = u_{a,a'} \oplus v_{a,a'}$.

Here, the direct sum of maps just means applying the respective maps component-wise.

Proposition 6.35. *If* $\mathbb{U}_1, \mathbb{U}_2$ *are* ϵ-*interleaved, and* $\mathbb{V}_1, \mathbb{V}_2$ *are* δ-*interleaved, then* $\mathbb{U}_1 \oplus \mathbb{V}_1$ *and* $\mathbb{U}_2 \oplus \mathbb{V}_2$ *are* $\max\{\epsilon, \delta\}$-*interleaved.*

Proof. Without loss of generality, let $\epsilon \geqslant \delta$, so we need to show that they are ϵ-interleaved. Recall that if two persistence modules are δ-interleaved, they are also ϵ-interleaved. Let φ^u, ψ^u be (series of) functions showing that $\mathbb{U}_1, \mathbb{U}_2$ are ϵ-interleaved. Similarly, let φ^v, ψ^v be (series of) functions showing that $\mathbb{V}_1, \mathbb{V}_2$ are ϵ-interleaved. Then, $\varphi^u \oplus \varphi^v, \psi^u \oplus \psi^v$ show that $\mathbb{U}_1 \oplus \mathbb{V}_1$ and $\mathbb{U}_2 \oplus \mathbb{V}_2$ are ϵ-interleaved. $\qquad \square$

If we now have a direct sum of interval modules, we can still nicely talk about births and deaths: we just consider each interval module in isolation. The following theorem shows that surprisingly most persistence modules can be expressed as direct sums of interval modules.

Theorem 6.36 (Structure theorem). *Any p.f.d. persistence module decomposes uniquely into interval modules, i.e., we have*

$$\mathbb{U} \cong \bigoplus_{i \in I} \mathbb{I}\langle b_i, d_i \rangle.$$

The intervals $\langle b_i, d_i \rangle$ are exactly the barcodes if $\mathbb{U}$ is a persistent homology module.

Note that unless we have some additional tame-ness condition on $\mathbb{U}$, I is not guaranteed to be finite.

Recall that when we talked about persistent homology, we said that there is some consistent global choice of basis for persistent homology groups. That is a consequence of the structure theorem. The structure theorem also allows us to prove one direction of Theorem 6.26, which we will do in the following.

Proposition 6.37. *Consider two interval modules $\mathbb{I}_1 = \mathbb{I}\langle b_1, d_1 \rangle$ and $\mathbb{I}_2 = \mathbb{I}\langle b_2, d_2 \rangle$. Then, $d_I(\mathbb{I}_1, \mathbb{I}_2) = d_b(\mathrm{Dgm}\mathbb{I}_1, \mathrm{Dgm}\mathbb{I}_2)$.*

Proof. To prove that $d_I(\mathbb{I}_1, \mathbb{I}_2) \geqslant d_b(\mathrm{Dgm}\mathbb{I}_1, \mathrm{Dgm}\mathbb{I}_2)$, we show that every upper bound on d_I is also an upper bound on d_b: assume that we have maps φ, ψ showing that the two modules are ϵ-interleaved. Then, consider $\psi_{a+\epsilon} \circ \varphi_a = v^1_{a, a+2\epsilon}$, equality holding because φ, ψ certify ϵ-interleaving and the triangular diagram commutes. Consider $a \in \langle b_1, d_1 \rangle$.

Case 1: $v^1_{a, a+2\epsilon} = 0$ for all $a \in \langle b_1, d_1 \rangle$. Then, $d_1 - b_1 < 2\epsilon$, and the (infinity-norm) distance of (b_1, d_1) to the diagonal is less than ϵ. If this case would also hold for v^2 we would be done, since for both $\mathrm{Dgm}\mathbb{I}_1$ and $\mathrm{Dgm}\mathbb{I}_2$ we could match the point to the diagonal.

Case 2: $v^1_{a, a+2\epsilon} = \mathrm{id}$ for some $a \in \langle b_1, d_1 \rangle$. Then, $d_1 - b_1 \geqslant 2\epsilon$. Furthermore, we have $\varphi_a(\mathbb{F}) = \mathbb{F}$ for all $a \in \langle b_1, d_1 - 2\epsilon \rangle$. So, for these a, we must also have $a + \epsilon \in \langle b_2, d_2 \rangle$. This tells us that $\langle b_2, d_2 \rangle$ must "cover" a large part of $\langle b_1, d_1 \rangle$, namely we get $b_2 \leqslant b_1 + \epsilon$, and $d_2 \geqslant d_1 - \epsilon$. We can now see that $|b_2 - b_1| \leqslant \epsilon$ and $|d_2 - d_1| \leqslant \epsilon$: to violate this, $\langle b_2, d_2 \rangle$ would have to be a larger interval than $\langle b_1, d_1 \rangle$ (in particular, it would be longer than 2ϵ), and we could thus exchange their roles and get that $b_1 \leqslant b_2 + \epsilon$ and $d_1 \geqslant d_2 - \epsilon$. We thus know that we can just match the off-diagonal points to each other, and since we must have $d_\infty((b_1, d_1), (b_2, d_2)) \leqslant \epsilon$ we get the bound on d_b.

We now prove the other direction, $d_I(\mathbb{I}_1, \mathbb{I}_2) \leqslant d_b(\mathrm{Dgm}\mathbb{I}_1, \mathrm{Dgm}\mathbb{I}_2)$. To see this, we show that from every matching whose longest edge is ϵ, we get an ϵ-interleaving.

Case 1: The two off-diagonal points are matched to the diagonal. Then, we get that $d_i - b_i \leqslant 2\epsilon$ for both of them, and thus for all $\epsilon' > \epsilon$, $\mathbb{I}_1$ and $\mathbb{I}_2$ are ϵ'-interleaved with $\varphi, \psi = 0$ (recall Exercise 6.22). Thus, $d_I \leqslant \epsilon$.

Case 2: The points are matched with each other. Then, $|b_2 - b_1| \leqslant \epsilon$ and $|d_2 - d_1| \leqslant \epsilon$. Taking $\varphi, \psi = \text{id}$ we can see that $\mathbb{I}_1$ and $\mathbb{I}_2$ are ϵ-interleaved. Thus, $d_I \leqslant \epsilon$. $\qquad\square$

We can now use this to show one direction of Theorem 6.26.

Corollary 6.38. *Let $\mathbb{U}, \mathbb{V}$ be p.f.d. persistence modules. Then we have that $d_I(\mathbb{U}, \mathbb{V}) \leqslant d_b(\text{Dgm}\,\mathbb{U}, \text{Dgm}\,\mathbb{V})$.*

Proof. We apply the structure theorem to decompose $\mathbb{U}, \mathbb{V}$ into direct sums of interval modules, i.e., $\mathbb{U} = \bigoplus_{i \in I} \mathbb{I}\langle b_i, d_i \rangle \oplus \bigoplus_{j \in J} 0$ and $\mathbb{V} = \bigoplus_{i \in I} 0 \oplus \bigoplus_{j \in J} \mathbb{I}\langle b_j, d_j \rangle$. From the Bottleneck matching we get a matching between parts making up $\mathbb{U}$ and $\mathbb{V}$. Since the Bottleneck distance is the maximum length of any edge, we have $d_b(\text{Dgm}\,\mathbb{U}, \text{Dgm}\,\mathbb{V}) \geqslant d_b(\text{Dgm}\,\mathbb{I}_1, \text{Dgm}\,\mathbb{I}_2) = d_I(\mathbb{I}_1, \mathbb{I}_2)$ for every two interval modules that were matched together, where we used Proposition 6.37. Finally, we use Proposition 6.35 to get the desired statement. $\qquad\square$

Questions

20. *How can we measure distances between persistence diagrams?* Discuss Bottleneck and Wasserstein distance.

21. *How stable are filtrations derived from simplex-wise monotone functions with respect to Bottleneck distance?* State, illustrate and prove the stability theorem (Theorem 6.7).

22. *How can we measure distances between persistence modules?* Define interleaving distance and discuss its relation to Bottleneck distance.

23. *How stable are Čech complexes to perturbations of the underlying point set?* Define Hausdorff distance, state and prove the stability theorem for Čech complexes (Theorem 6.31).

Chapter 7

Reeb Graphs and Mapper

In this chapter we look at another tool in topological data analysis, called *Mapper*. The underlying idea of Mapper has its roots in Morse theory, where Georges Reeb defined a graph to summarize a Morse function on a manifold. We first discuss these graphs, called *Reeb graphs*, and then how to mimic the ideas for the case where instead of a manifold we have point cloud data.

Before we dive into the mathematical details, a short remark about the pronunciation of the word "Reeb graph": Georges Reeb, after whom these graphs are named, was a French mathematician born in the German speaking region Alsace. Thus, he likely pronounced his name the German way, that is, with the "ee" spoken similar to the "ea" in "bear" (as opposed to "beer").

7.1   Reeb Graphs
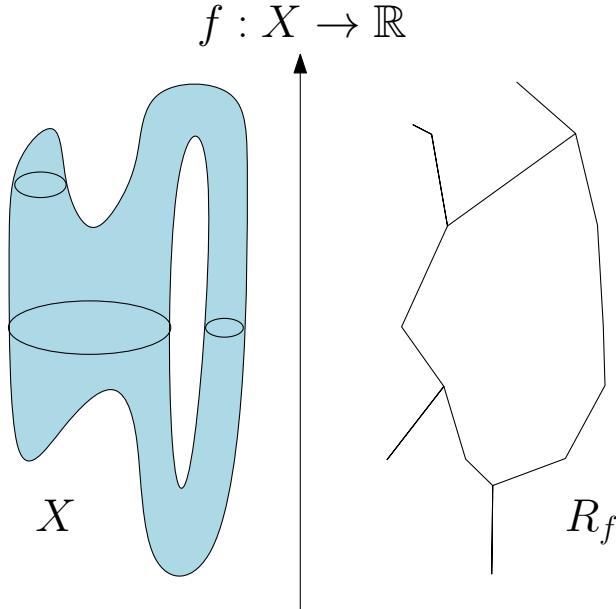
The idea of Reeb graphs is that given some topological space X, and some function $f : X \to \mathbb{R}$, we consider the preimage of f for some fixed value $a \in \mathbb{R}$. We place one point per connected path-component of the preimage. We do this for all values in $\mathbb{R}$, and connect the points corresponding to neighboring connected components in adjacent preimages. More formally,

Definition 7.1. *Let X be some topological space, and f a function $f : X \to \mathbb{R}$. Two points x, y are called equivalent ($x \sim y$), iff $f(x) = f(y) = \alpha$ and x and y are in the same path-connected component of $f^{-1}(\alpha)$. The* Reeb graph R_f *is the quotient space $X/\sim$.*

We assume all of our functions to be *levelset tame* for the space X:

Definition 7.2. *A function $f : X \to \mathbb{R}$ is* levelset tame *if*

- *each levelset $f^{-1}(\alpha)$ has finitely many connected components, all of which are path-connected, and*

$$f : X \to \mathbb{R}$$

Figure 7.1: *An example of a Reeb graph*

- *the homology groups of the levelsets only change at finitely many critical values.*

As we have defined it, the Reeb graph is just a (continuous) topological space. We call it a graph since it is 1-dimensional, but to arrive at a graph as we know it in combinatorics, we will need to discretize it. For this we need to define vertices and edges. There are many different possibilities of defining vertices and edges to discretize the Reeb graph, but we want to define some type of minimal one.

Let us look at the neighborhood of some point p in the Reeb graph (as a topological space). We look at how many ways there exist to go from p towards the direction of higher f-value (we call this number the up-degree u), and how many ways to go towards the direction of lower f-value (we call this the down-degree l). Depending on u and l, we classify p as in Table 7.1.

Table 7.1: *Classifications of points in the Reeb graph.*

u	l	Classification
1	1	regular
0	> 0	maximum
> 0	0	minimum
$\geqslant 2$	l	up-fork
u	$\geqslant 2$	down-fork

Note that a point can fall into multiple of these classes, for example it can be a maximum and a down-fork simultaneously, or an up-fork and a down-fork simultaneously. We

call the minima, maxima, up-forks, and down-forks *critical points*. Our discretization places vertices at the critical points, and all the connections between critical points (that are made up of only regular points) become the edges of our graph. Note that the graph we get through this process is not necessarily simple, we may have multi-edges.

Exercise 7.3. *Consider a double torus embedded in $\mathbb{R}^3$. You can imagine it as the result of taking the figure depicted in Figure 7.2 embedded in the plane $x_3 = 0$, replacing every point by a 3-dimensional ball with radius $r < \min\{d/2, R/2\}$, and taking the boundary of the union of these balls.*
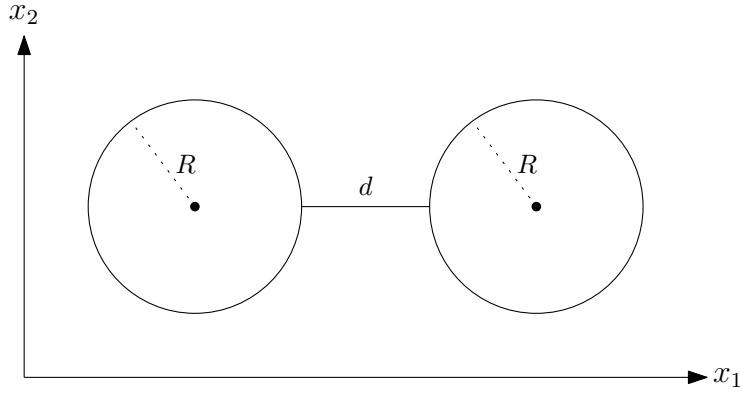

Figure 7.2: *The space blown up to a double torus in Exercise 7.3.*

Draw the Reeb graphs for the functions $f_1(x) = x_1$, $f_2(x) = x_2$, and $f_3(x) = x_3$.

We next consider merge trees and split trees, which are variants of the Reeb graph, where instead of levelsets, we look at sub-level sets or super-level sets.

Definition 7.4. *Let X be some topological space, and f a function $f : X \to \mathbb{R}$. We have $x \sim_M y$ for two points x, y, if and only if $f(x) = f(y) = \alpha$ and x and y are in the same connected component of $f^{-1}((-\infty, \alpha])$. The merge tree T_M is the quotient space $X/\sim_M$.*

Note that in the merge tree, since we only increase the space under consideration, we never have a connected component that splits. We can only have new connected components appearing, and connected components merging. This also tells us that the Merge tree (or its discretization) is always a tree.

Definition 7.5. *Let X be some topological space, and f a function $f : X \to \mathbb{R}$. We have $x \sim_S y$ for two points x, y, if and only if $f(x) = f(y) = \alpha$ and x and y are in the same connected component of $f^{-1}([\alpha, \infty))$. The split tree T_S is the quotient space $X/\sim_S$.*

To be able to compute Reeb graphs in actual TDA applications, we now look at Reeb graphs in the context of simplicial complexes. We consider a simplicial complex K and a function $f : |K| \to \mathbb{R}$, which is piece-wise linear (linear on each simplex). We observe that

the Reeb graph then only depends on the 2-skeleton of K. This is the case since looking at a levelset is the same as cutting through the simplicial complex. When we cut through a simplex, we generally get a simplex of one dimension lower. In a simplicial complex, connectivity is completely determined by the 1-skeleton. Thus, before cutting, the 2-skeleton suffices. We can also see that the critical points are images of the vertices of K. This happens since a connected component can only appear, disappear, split, or merge at some local maximum or minimum of the connected component. Since the function is linear, the maximum or minimum of every simplex is always attained at some vertex. We define the *augmented Reeb graph* of a simplicial complex with a PL-function, as the discretization of the Reeb graph by placing a vertex for every connected component in all the levelsets for the values α for which $f(v) = \alpha$ for some vertex of the simplicial complex.

Theorem 7.6. *Given a simplicial complex* K *with* m *faces and a piece-wise linear function* $f : |K| \to \mathbb{R}$ *on it, we can compute the augmented Reeb graph* R_f *of* K *with respect to* f *in time* $O(m \log m)$.

Proof. We only have to consider the 2-skeleton of K. To compute the augmented Reeb graph, we perform a discrete sweep (or scan) through K in the order given by f, only stopping at values α such that $f(v) = \alpha$ for some vertex v. In this sweep, we want to keep track of the connected components.

For any $\alpha \in \mathbb{R}$, the levelset $f^{-1}(\alpha)$ of the 2-skeleton of K is just a graph G_α: vertices and edges of K induce vertices of G_α, triangles induce edges. In the open interval between any two values α with $f(v) = \alpha$, G_α stays the same. In our sweep, we track the graph G_α. Whenever we enter a vertex v, faces may end: The vertices in G_α that correspond to edges vw in K with $f(v) > f(w)$ all get merged into a single vertex. Whenever we then exit v, some faces may start. The vertex corresponding to v is split into one vertex per edge vw in K such that $f(v) < f(w)$. Some of these vertices are connected: the vertices corresponding to the edges vw and vx are connected by an edge if vwx is a triangle of K.

To build the augmented Reeb graph out of these G_α graphs, we use a dynamic spanning forest data structure. Such a data structure can update connected components under both insertion and deletions of edges (vertices we only delete when also deleting all their incident edges). There exist such data structures that can do each update in amortized time $O(\log m)$, where m is the size of the graph. The size of the graph is bounded by the sum m of vertices, edges, and triangles in K. Each such feature appears at one point, and disappears at one point, and we thus have at most $2m$ insertions and deletions in total, giving an $O(m \log m)$ algorithm. $\qquad\square$

Exercise 7.7. *Consider a simplicial complex* K *and a PL (piece-wise linear) function* $f : |K| \to \mathbb{R}$. *What happens to the Reeb graph when you add one additional face to* K *and extend* f *accordingly?*

7.1.1   Homology of Reeb Graphs

The Reeb graph of a topological space X with respect to a function f can be viewed as a summary of X through the lens of f. The natural question is: how good of a summary is it? It is clear that in general we lose information, for example on the dimension of X, but we can still hope that some topological information is retained. In this section, we thus compare the homology of the Reeb graph to the homology of X. Since the Reeb graph R_f is a graph (a 1-dimensional object), we have $H_p(R_f) = 0$ for $p \geqslant 2$, so any higher-dimensional homology gets lost. However, a graph can still have non-trivial homology in dimensions 0 and 1.

Observation 7.8. *For a levelset tame* $f : X \to \mathbb{R}$*, we have* $\beta_0(X) = \beta_0(R_f)$*.*

In other words, the Reeb graph captures the 0-homology of the input space X perfectly, no matter which levelset tame function f we use.

Sadly, the same does not hold for the 1-homology. Let us consider a torus, as in Figure 7.3. In general, it can be that the choice of function f determines whether we capture a hole or not, consider for example a cylinder. However note that for the torus, it is actually the case that no matter which function f we choose, we cannot capture its 1-homology (this is non-trivial to show).
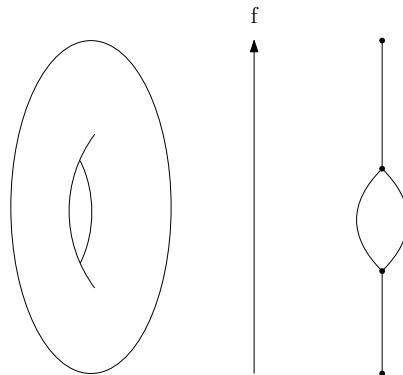

Figure 7.3: *The torus and its Reeb graph.*

On the other hand, we can see that every cycle in the Reeb graph is indeed also a cycle in the topological space X, and it cannot be filled in, so it is indeed a hole. Thus we also get the following observation:

Observation 7.9. *For a levelset tame* $f : X \to \mathbb{R}$*, we have* $\beta_1(X) \geqslant \beta_1(R_f)$*.*

Can we somehow formalize which holes we lose? To do this, we split up homology into "horizontal" and a "vertical" parts. The intuition behind these terms is that we consider f to be a function pointing upwards, and thus roughly, horizontal means "perpendicular to f" and vertical means "along f".

Definition 7.10. *A p-th homology class* $h \in H_p(X)$ *is called* horizontal *if there is a finite set of values* $A = \{a_1, \ldots, a_k\}$ *such that* $h \in \mathrm{im}(\iota_*)$ *where* ι_* *is the map* $H_p(\bigcup_{a \in A} X_a) \to H_p(X)$ *induced by inclusion, where* $X_a = f^{-1}(a)$.

This definition means that we need to be able to find a finite set of levelsets, such that we can find cycles contained in these levelsets, which are in the homology class h in $H_p(X)$.

One now wonders whether the set of horizontal homology classes forms a group. Let this set be $\overline{H_p}(X)$. It turns out that it is indeed a group.

Lemma 7.11. $\overline{H_p}(X)$ *is a subgroup of* $H_p(X)$.

Proof. First, we see that the identity element 0 is in $\overline{H_p}(X)$. We can take an arbitrary set A, and we can always map the 0 element of $H_p(\bigcup_{a \in A} X_a)$ to 0.

Next, we show that the set is closed under addition. Let $p, q \in \overline{H_p}(X)$, and we show that $p + q \in \overline{H_p}(X)$. p has a pre-image in some levelset A_p, and q has a pre-image in some levelset A_q. $p + q$ must have a pre-image in $A_p \cup A_q$.

Finally, we show that the inverse of every element is contained in the group, but since every element is self-inverse in $\mathbb{Z}_2$-homology, we get that for every element its inverse is also contained in $\overline{H_p}(X)$. $\qquad\square$

Since the horizontal homology is a sub-group, we can now easily define *vertical homology* by taking quotient groups.

Definition 7.12. *The* vertical homology group *of* X *with respect to* f *is the group* $\overset{\vee}{H_p}(X) := H_p(X)/\overline{H_p}(X)$.

Observation 7.13. $\dim(H_p(X)) = \dim(\overline{H_p}(X)) + \dim(\overset{\vee}{H_p}(X))$.

The following fact that we do not prove here shows that when we go from a space X to its Reeb graph, we keep the vertical homology classes, and lose the horizontal ones.

Fact 7.14. *The surjection* $\phi : X \to R_f$ *induces an isomorphism* $\overset{\vee}{\Phi} : \overset{\vee}{H_1}(X) \to \overset{\vee}{H_1}(R_f)$.

Corollary 7.15. *Given* X *an orientable connected compact 2-manifold, and a Morse function* $f : X \to \mathbb{R}$, *then* $\beta_1(R_f) = \beta_1(X)/2$.

Here, a 2-manifold is a space that locally at every point looks like $\mathbb{R}^2$. Orientable means that there is an inside and an outside side. A Morse function is a "nice enough" function defined in terms of some derivatives, which we do not need to specify here.

Exercise 7.16. *(a) Consider a 2-dimensional geometric simplicial complex* K *embedded in* $\mathbb{R}^2$. *Consider the function* $f(x) = x_1$. *Show that* $\beta_1(K) = \beta_1(R_f)$.

(b) Find a geometric simplicial complex K *embedded in* $\mathbb{R}^2$ *and a map* $f : K \to \mathbb{R}$ *such that* $\beta_1(K) > \beta_1(R_f)$.

7.2   Distances for Reeb Graphs

In order to compare Reeb graphs to each other, we again want to define distance measures between them. We discuss two such measures here. The first one, called *interleaving distance*, is, not surprisingly, similar to the interleaving distance of persistence modules. The second one, called *functional distortion distance* is similar to the Gromov-Hausdorff distance for metric spaces.

7.2.1   Interleaving Distance

When do we want two Reeb graphs to be considered the same, and thus have distance 0? We definitely need that the graphs are isomorphic in the sense of graph isomorphism. But further than that, we also want that this graph isomorphism is "function preserving". In other words, the critical points should lie on the same function levels. The idea of the interleaving distance is to measure how far away from this we are. Thus, given two Reeb graphs R_f, R_g, "how much" is missing towards a "function preserving isomorphism"? Towards formalizing this idea, we need a few definitions.

Note that when we compare two Reeb graphs R_f, R_g, those can be Reeb graphs of *different spaces* with regards to *different functions*.

Definition 7.17. *An ϵ-thickening X_ϵ of some topological space X is given by $X_\epsilon := X \times [-\epsilon, +\epsilon]$.*

Definition 7.18. *For a Reeb graph R_f consider a function $f_\epsilon : (R_f)_\epsilon \to \mathbb{R}$ such that*
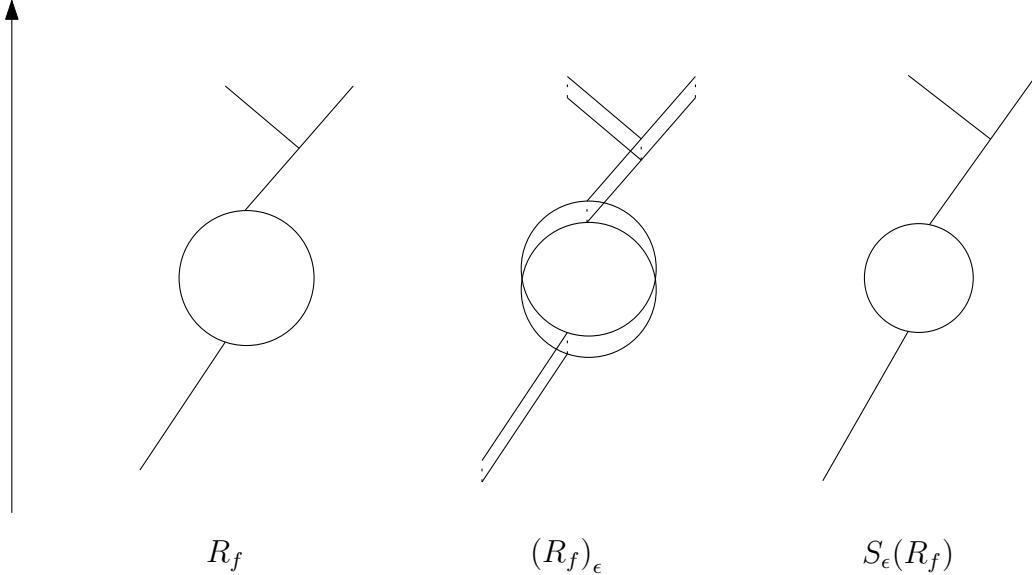
$$(x, t) \mapsto f(x) + t.$$

The ϵ-smoothing of R_f, denoted by $S_\epsilon(R_f)$ is the Reeb graph of $(R_f)_\epsilon$ with regards to f_ϵ.

An example of these definitions can be seen in Figure 7.4. Note that when we say $(R_f)_\epsilon$, we mean an ϵ-thickening of R_f, *not* a Reeb graph with regards to some function f_ϵ. The ϵ-smoothing $S_\epsilon(R_f)$ is then a Reeb graph with regards to the function f_ϵ, but *of* $(R_f)_\epsilon$, and not of the original space that R_f is the Reeb graph of. Furthermore, when we write $f(x)$ for some $x \in R_f$, we mean that we extend f to some function $f^* : R_f \to \mathbb{R}$, simply by mapping x (an element of $X/\sim$, i.e., an equivalence class of $\sim$) to $f(p)$ for some point $p \in X$ in the equivalence class x. We will just call this function f as well for simplicity.

Definition 7.19. *The function $\iota : R_f \to S_\epsilon(R_f)$ with $x \mapsto [(x, 0)]$ is the* quotiented inclusion map*. Here, $[(x, 0)]$ denotes the equivalence class, or the connected component that contains $(x, 0)$ in $f_\epsilon^{-1}(f_\epsilon(x, 0))$.*

Consider some function $\mu : R_f \to R_g$ which is function preserving, i.e., $f(x) = g(\mu(x))$ for all $x \in R_f$. A function-preserving map $\mu : R_f \to S_\epsilon(R_g)$ induces a function preserving map $\mu_\epsilon : S_\epsilon(R_f) \to S_{2\epsilon}(R_g)$ with $[(x, t)] \mapsto [(\mu(x), t)]$.

$$R_f \qquad\qquad (R_f)_\epsilon \qquad\qquad S_\epsilon(R_f)$$

Figure 7.4: *A Reeb graph, its ϵ-thickening, and its ϵ-smoothing.*

Definition 7.20 (Reeb graph interleaving). *Two Reeb graphs R_f, R_g are ϵ-interleaved, if there exists a pair of function preserving maps $\varphi : R_f \to S_\epsilon(R_g)$, $\psi : R_g \to S_\epsilon(R_f)$ such that the following diagram commutes:*

$$
\begin{array}{ccccc}
R_f & \xrightarrow{\ \iota\ } & S_\epsilon(R_f) & \xrightarrow{\ \iota_\epsilon\ } & S_{2\epsilon}(R_f) \\
& {\scriptstyle \psi}\ \searrow\ \nearrow\ {\scriptstyle \varphi} & & {\scriptstyle \psi_\epsilon}\ \searrow\ \nearrow\ {\scriptstyle \varphi_\epsilon} & \\
R_g & \xrightarrow{\ \iota\ } & S_\epsilon(R_g) & \xrightarrow{\ \iota_\epsilon\ } & S_{2\epsilon}(R_g)
\end{array}
$$

Here, to understand why ι_ϵ makes sense, we need the following fact, the proof of which is left as an exercise.

Observation 7.21. $S_\delta(S_\epsilon(R_f)) \cong S_{\delta+\epsilon}(R_f)$.

Note that by construction of ι, ι_ϵ and φ_ϵ (or ψ_ϵ, respectively), the trapezoidal parts of this diagram commute trivially: $\varphi_\epsilon \circ \iota(x) = \varphi_\epsilon([(x,0)]) = [(\varphi(x),0)] = \iota_\epsilon \circ \varphi(x)$. Furthermore, any two compact and connected Reeb graphs are ϵ-interleaved for some ϵ. Lastly, if R_f and R_g are ϵ-interleaved, then they are also δ-interleaved for all $\delta \geqslant \epsilon$.

Definition 7.22. $d_I(R_f, R_g) = \inf\{\epsilon \mid R_f, R_g \text{ are } \epsilon\text{-interleaved}\}$.

We once again have a stability theorem, which we will not prove here.

Theorem 7.23. *For tame functions $f, g : X \to \mathbb{R}$ we have $d_I(R_f, R_g) \leqslant \|f - g\|_\infty$.*

82

7.2.2 Functional Distortion Distance

As mentioned above, the functional distortion distance is motivated by the Gromov-Hausdorff distance for metric spaces. Thus, the first step is to define a metric on a single Reeb graph.

Definition 7.24. *Let R_f be a Reeb graph of a space X, and $u, v \in R_f$ (in the same connected component of R_f), and let π be a path from u to v. We define the* height *of π as* $\mathrm{height}(\pi) = \max_{x \in \pi} f(x) - \min_{x \in \pi} f(x)$. *To turn this into a distance metric, we consider $\Pi(u, v)$, the set of all paths between u and v. Then, the* function induced metric *on R_f is defined as*

$$d_f(u, v) = \min_{\pi \in \Pi(u,v)} \mathrm{height}(\pi).$$

In a sense, $d_f(u, v)$ is the "thickness" of the thinnest "slice" of the space X in which u and v are connected.

Definition 7.25 (Functional distortion distance). *Let R_f and R_g be two Reeb graphs. Let $\Phi : R_f \to R_g$, $\Psi : R_g \to R_f$ be continuous functions, but not necessarily function-preserving. Then, we define correspondence and distortion:*

$$C(\Phi, \Psi) = \{(x, y) \in R_f \times R_g \mid \Phi(x) = y \ or \ x = \Psi(y)\}$$

$$D(\Phi, \Psi) = \sup_{(x,y),(x',y') \in C(\Phi,\Psi)} \frac{1}{2} |d_f(x, x') - d_g(y, y')|.$$

And finally, we define the functional distortion distance,

$$d_{FD}(R_f, R_g) = \inf_{\Phi, \Psi} \max\{D(\Phi, \Psi), \|f - (g \circ \Phi)\|_\infty, \|g - (f \circ \Psi)\|_\infty\}.$$

Also for this distance measure there is a stability theorem.

Theorem 7.26. *Let $f, g : X \to \mathbb{R}$ be tame functions. Then, $d_{FD}(R_f, R_g) \leqslant \|f - g\|_\infty$.*

We can also quantify the relation between the two discussed distances.

Theorem 7.27. $d_I(R_f, R_g) \leqslant d_{FD}(R_f, R_g) \leqslant 3 d_I(R_f, R_g)$.

Exercise 7.28. *Consider a merge tree T with regards to a function f. We define the a-shift x^a for any $x \in T$ to be the unique "ancestor" of x with function value $f(x^a) = f(x) + a$.*

We now consider two merge trees; T_1 with regards to f, and T_2 with regards to g. We call T_1 and T_2 ϵ-compatible if there exist maps $\alpha : T_1 \to T_2$ and $\beta : T_2 \to T_1$ such that we get the following commutativities:

- $g(\alpha(x)) = f(x) + \epsilon$ *for all $x \in T_1$*

- $f(\beta(y)) = g(y) + \epsilon$ *for all* $y \in T_2$

- $\beta \circ \alpha(x) = x^{2\epsilon}$ *for all* $x \in T_1$

- $\alpha \circ \beta(y) = y^{2\epsilon}$ *for all* $y \in T_2$.

The interleaving distance between merge trees $d_I(T_1, T_2)$ *can now be defined as the infimum of all* ϵ *such that* T_1 *and* T_2 *are* ϵ-*compatible. Show that* $d_I(T_1, T_2) = d_{FD}(T_1, T_2)$.
Note: we technically only defined d_{FD} *for Reeb graphs. You can simply consider a merge tree to be the Reeb graph of itself (with regards to the same filter function).*

7.3 Mapper

Reeb graphs lose a lot of information, since they at most retain some 1-dimensional holes, but no larger holes. To generalize Reeb graphs further, we start looking at neighborhoods instead of levelsets, which will then lead to the Mapper algorithm.

7.3.1 An Approximation of the Reeb Graph

To begin, we consider the 1-dimensional case, and try to find an approximation of the Reeb graph. Instead of looking at pre-images of points, we will now look at pre-images of intervals. Let $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ be an open cover of $\mathbb{R}$ (i.e., a collection of open sets whose union is $\mathbb{R}$). As always, we consider a function $f : X \to \mathbb{R}$. For each $f^{-1}(U_\alpha)$, we consider a partition into path-connected components, i.e., $f^{-1}(U_\alpha) = \bigcup_{\beta \in B_\alpha} V_\beta$. We then look at $f^*(\mathcal{U}) := \{V_\beta\}$, the set of all V_β we get over all α. Our object of interest is the nerve of this family, i.e., $N(f^*(\mathcal{U}))$.

If we take sufficiently nice functions, and sufficiently fine covers, then $N(f^*(\mathcal{U}))$ is isomorphic to R_f.
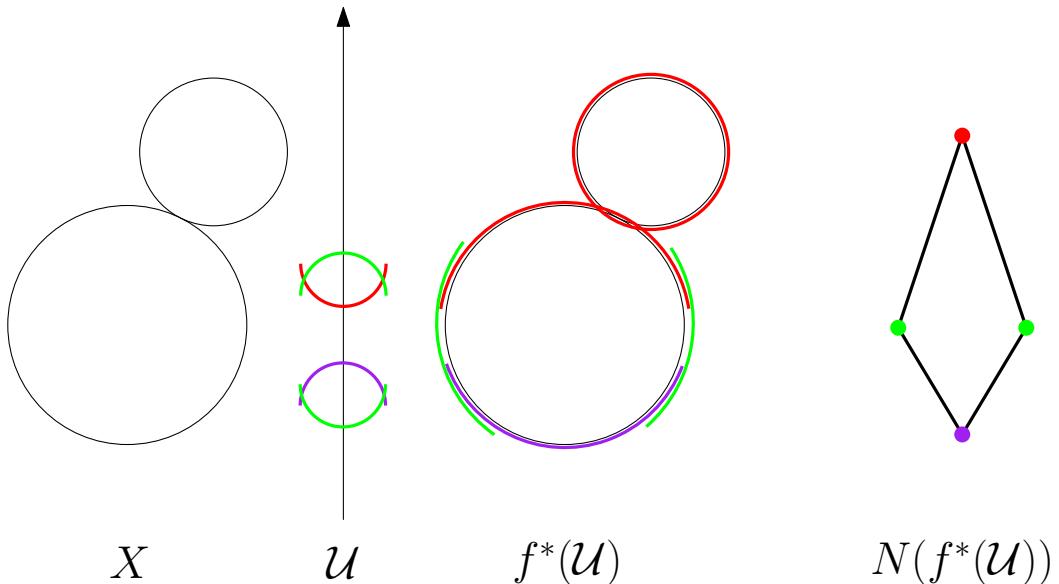
7.3.2 Topological Mapper

We can generalize this idea to maps to arbitrary spaces.

Definition 7.29. *Let* X, Z *be topological spaces. Then we call* $f : X \to Z$ *well-behaved if for all path-connected open sets* $U \subseteq Z$, $f^{-1}(U)$ *has finitely many path-connected components.*

Definition 7.30 (Mapper). *Let* $f : X \to Z$ *be well-behaved, and* $\mathcal{U}$ *be a (finite) open cover of* Z. *Then the* Mapper *is defined as* $M(\mathcal{U}, f) := N(f^*(\mathcal{U}))$.

As an example, we look at X being the boundary of the 3-cube $[0, 1]^3$. We then also look at $Z_1 = \mathbb{R}^2$ spanned by the x- and y-axis, with $f_1 : X \to Z_1$ being the projection onto this plane. Furthermore, we look at $Z_2 = \mathbb{R}$, spanned by just the x-axis, and $f_2 : X \to Z_2$ being again the projection.

$$X \qquad \mathcal{U} \qquad f^*(\mathcal{U}) \qquad N(f^*(\mathcal{U}))$$

Figure 7.5: *A space X, an open cover $\mathcal{U}$ of $\mathbb{R}$, the family $f^*(\mathcal{F})$, and its nerve.*

We consider the open cover $\mathcal{U}_2$ of Z_2: $\{(-\infty, \frac{1}{3}), (0, 1), (\frac{2}{3}, +\infty)\}$. For Z_1, we consider the cover $\mathcal{U}_1 := \mathcal{U}_2 \times \mathcal{U}_2$.
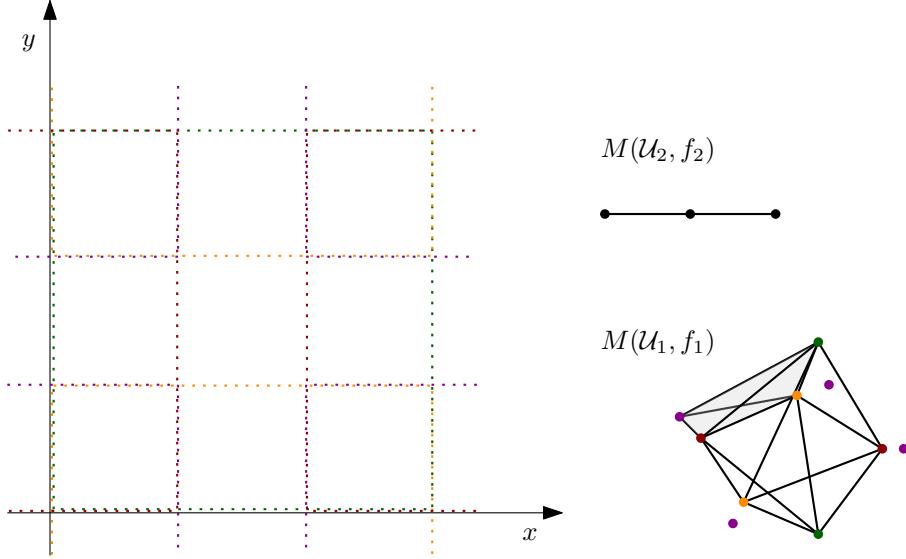
Exercise 7.31. *(a) Consider spaces X, Z, a filter function $f : X \to Z$, and an open cover $\mathcal{U}$ of Z. Show that if the* pullback cover $f^*(\mathcal{U})$ *is a* good cover *of X, then $M(\mathcal{U}, f)$ is homotopy equivalent to X.*

(b) Give an example of spaces X, Z, a filter function $f : X \to Z$, and a good cover $\mathcal{U}$ of Z, such that $M(\mathcal{U}, f)$ is not *homotopy equivalent to X.*

(c) Give an example of spaces X, Z, a filter function $f : X \to Z$, and an open cover $\mathcal{U}$ of Z such that the pullback cover $f^*(\mathcal{U})$ is not a good cover, but $M(\mathcal{U}, f)$ is still homotopy equivalent to X.

7.3.3   Mapper for Point Clouds

We would like to apply the ideas of the topological Mapper to analyze the shape of data. However, once again we have the issue that data usually does not come in the form of a topological space, but as a set of data points with a notion of distance between them. The *Mapper algorithm* for point clouds adapts the ideas of the topological Mapper to this setting.

Input: In the most general setting, data comes as a finite metric space (P, d_P), for example as points in $\mathbb{R}^d$ or as vertices of a graph. We also require a cover $\mathcal{U}$ of a space Z, usually $Z = \mathbb{R}$, as input. Finally, we also need a filter function $f : P \to Z$ and a clustering algorithm (which might also require some input parameters).

Figure 7.6: *The cover $\mathcal{U}_\infty$, and the two Mappers. The Mapper $M(\mathcal{U}_1, f_1)$ consists of an empty octahedron, with additional filled tetrahedra attached at the purple vertices. The whole space thus collapses to an octahedron.*

Algorithm: Since at the moment we only have a discrete metric space, we do not really have the notion of connected components yet. For every $U \in \mathcal{U}$, we thus cluster the pre-image $f^{-1}(U)$ using some clustering algorithm, which we can also consider as an input. Now, we can just consider each cluster C_i as a vertex of some simplicial complex K, and add a face $\{C_1, \ldots, C_k\}$ to K if these clusters (which are just point sets) have a common point.

Output: We output K, or even just its 1-skeleton.

As you can see, this algorithm requires a lot of input parameters. While this allows to encode previous knowledge of the data set (e.g. by choosing as filter function the distance to a known center of the data), it also makes the space of possible outputs very large. Picking the correct parameters is currently still an art form on its own, and there is significant research being done towards understanding the interplay between the parameters and statistical guarantees for certain good choices of parameters.

7.4 Multiscale Mapper

Motivated by the many tuneable parameters, we discuss here one idea to look at many values at once. The multiscale Mapper is a combination of the ideas of persistence and of Mapper. We here want to look at different covers.

Definition 7.32. *Let $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ and $\mathcal{F} = \{V_\beta\}_{\beta \in B}$ be two covers of the same space X. A* map of covers *is a map $\varphi : A \to B$ such that for every $\alpha \in A$, we have $U_\alpha \subseteq V_{\varphi(\alpha)}$.*

86

Proposition 7.33. *If $\varphi : \mathcal{U} \to \mathcal{V}$ is a map of covers (with a slight abuse of notation), then the map $N(\varphi) : N(\mathcal{U}) \to N(\mathcal{V})$ given on the vertices by φ is simplicial.*

Proof. Let $\sigma \in N(\mathcal{U})$. We need to show that the intersection $\bigcap_{\beta \in N(\varphi)(\sigma)} V_\beta$ is non-empty.

$$\bigcap_{\beta \in N(\varphi)(\sigma)} V_\beta = \bigcap_{\alpha \in \sigma} V_{\varphi(\alpha)} \supseteq \bigcap_{\alpha \in \sigma} U_\alpha \neq \emptyset$$

Thus, $N(\varphi)(\sigma) \in N(\mathcal{V})$. $\qquad\square$

An example of this map between nerves can be seen in Figure 7.7.

Proposition 7.34. *Let $f : X \to Z$ be some map, and $\mathcal{U}, \mathcal{V}$ be covers of Z, with $\varphi : \mathcal{U} \to \mathcal{V}$ some map of covers. Then, there exists a map of covers $f^*(\varphi) : f^*(\mathcal{U}) \to f^*(\mathcal{V})$.*

Recall that $f^*(\mathcal{U})$ is the cover of X consisting of the connected components of the pre-images of the sets of $\mathcal{U}$ under f.

Proof. For every α, we have $U_\alpha \subseteq V_{\varphi(\alpha)} \implies f^{-1}(U_\alpha) \subseteq f^{-1}(V_{\varphi(\alpha)})$. We now need to go from these pre-images to their connected components. Since every connected component of $f^{-1}(U_\alpha)$ must lie in a unique connected component of $f^{-1}(V_{\varphi(\alpha)})$, our desired map of covers is given by exactly mapping to this connected component. $\qquad\square$

If we have multiple maps of covers, $\mathcal{U} \xrightarrow{\varphi} \mathcal{V} \xrightarrow{\psi} \mathcal{W}$, we can concatenate the maps, and the f^* function distributes: $f^*(\psi \circ \varphi) = f^*(\psi) \circ f^*(\varphi)$.

Let $\mathfrak{U} = \mathcal{U}_1 \xrightarrow{\varphi_1} \mathcal{U}_2 \xrightarrow{\varphi_2} \ldots \xrightarrow{\varphi_{n-1}} \mathcal{U}_n$ be a sequence of covers of Z with maps between them, which we call a *cover tower*. By applying f^* we get a cover tower $f^*(\mathfrak{U})$ of X.

Definition 7.35 (Multiscale Mapper). *Let $f : X \to Z$, $\mathfrak{U}$ a cover tower of Z. Then, the Multiscale Mapper $MM(\mathfrak{U}, f)$ is*

$$MM(\mathfrak{U}, f) := N(f^*(\mathfrak{U})) = \{N(f^*(\mathcal{U}_i)) \mid \mathcal{U}_i \in \mathfrak{U}\}$$

together with the induced simplicial maps
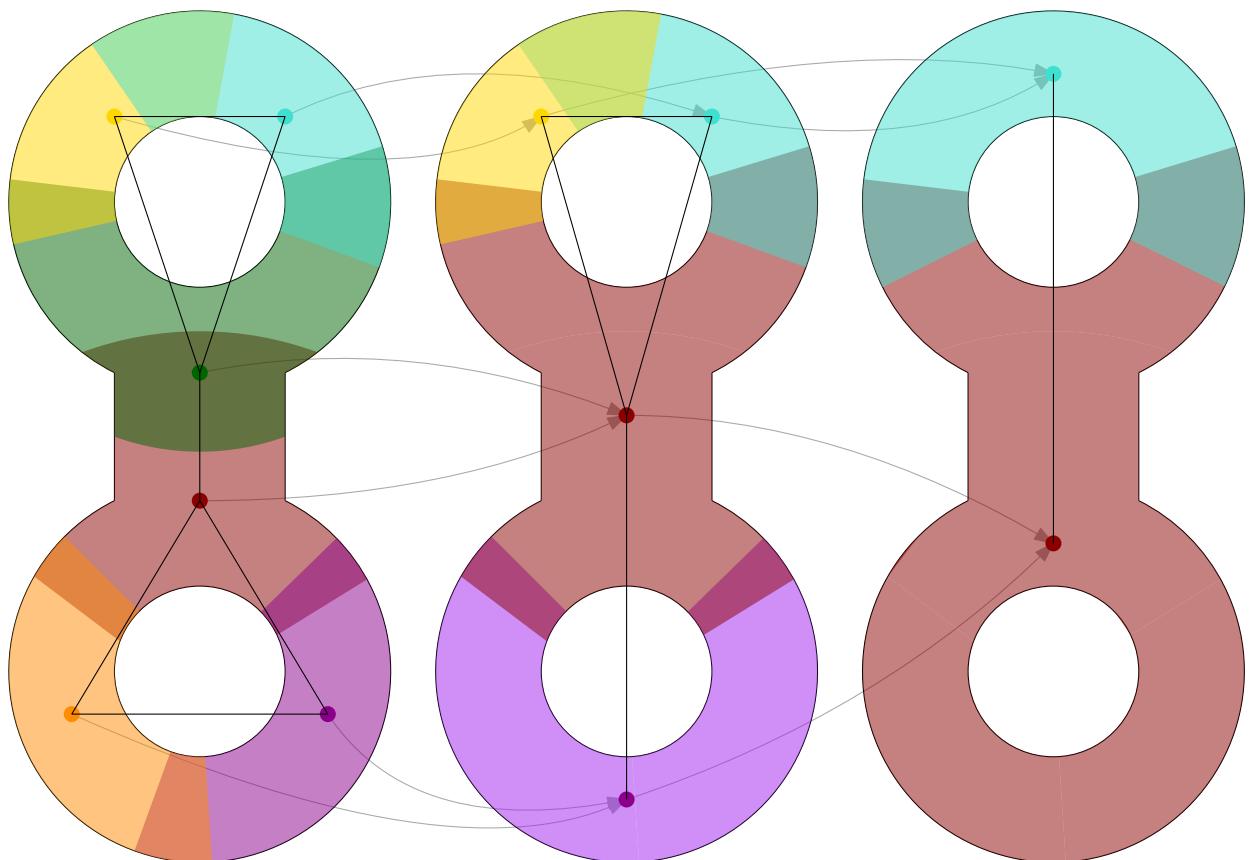
$$N(f^*(\varphi_i)) : N(f^*(\mathcal{U}_i)) \to N(f^*(\mathcal{U}_{i+1})).$$

Applying homology, we get the sequence homology groups with induced homomorphisms between them, i.e., a persistence module:

$$H_p(N(f^*(\mathcal{U}_1))) \xrightarrow{N(f^*(\varphi_1))_*} \ldots \xrightarrow{N(f^*(\varphi_{n-1}))_*} H_p(N(f^*(\mathcal{U}_n))).$$

We can now view $\mathrm{Dgm}_p MM(\mathfrak{U}, f)$ as a topological summary of f through the lens of $\mathfrak{U}$.

As opposed to the normal Mapper, at first glance the Multiscale Mapper adds even more parameters. But a cover tower can be seen as a way of looking at a whole interval of covers. For example, we can get a cover tower by increasing the size of all intervals in an interval cover. The features of the data should show up as a robust feature that persists for a longer time over this process, while spurious features obtained from choosing "wrong" Mapper parameters should disappear quickly.

Figure 7.7: *Three pullback covers of a space X (thickened figure "8"), with their nerves and the induced simplicial maps between them.*

Questions

24. *What is a Reeb graph?* State the definition and describe how we get the graph structure.

25. *How can we compute the augmented Reeb graph of a piece-wise linear function?* Define the augmented Reeb graph and explain the algorithm to compute it.

26. *How much of the homology of the underlying topological space is captured by the Reeb graph?* Explain vertical and horizontal homology.

27. *What is the interleaving distance for Reeb graphs?* Give the definitions and state the relevant stability theorems.

28. **(This topic was not covered in this year's course in FS25 and therefore the following question will not be asked in the exam.)** *What is the functional distortion distance for Reeb graphs?* Give the definitions and state the relevant stability theorems.

29. *What is the topological Mapper?* State the Definition and give an example.

30. *How can we use Mapper on point cloud data?* Explain the Mapper algorithm and describe the input parameters.

31. **(This topic was not covered in this year's course in FS25 and therefore the following question will not be asked in the exam.)** *How can we use Mapper on several covers at once?* Explain the Multiscale Mapper.

Chapter 8

Optimal Generators

In some applications, we are not only interested in the number of holes in our data, but we also want to get representations of these holes in the data. In other words, we are interested in finding a "representative" basis of the homology group. There are two main challenges to picking such a basis: First off, there are many different choices of homology classes which form a basis of the homology group. Second off, even within a single homology class, there are many homologous cycles. Thus, there are many different choices for cycles as bases of the homology group. How do we find good bases?

To specify our optimization target, we define a weight function $w : K_p \to \mathbb{R}_{\geq 0}$ on the p-simplices, and the weight of a chain is simply the sum, i.e., $w(c) = \sum \alpha_i w(\sigma_i)$ for $c = \sum \alpha_i \sigma_i$. The weight of a set of cycles $\mathcal{C}$ is then the sum of weights of each cycle. We are now interested in cycles that have minimal weight in their homology class, or at bases with minimum total weight.

We look at this problem in two settings: first we look at the case where we are given a fixed simplicial complex and we want to find an optimal basis for the homology of this complex. This can be applied for example if the persistence diagram of a filtration gives us a range of values in which we expect the complex to nicely capture the shape of the data. We can then compute an optimal basis for the homology of the fixed complex for some value in this range.

In some applications, we might also want to take a closer look at single intervals in the persistence barcode, that is, understand a hole that is born at time b and dies at time d (for example, to decide whether it corresponds to a feature in the data or is just a consequence of the process). This brings us to the second setting we look at in this chapter, where we want to find an optimal representative of a persistent homology class.

8.1 Optimal Basis of a Fixed Complex

Definition 8.1. *A set $\mathcal{C}$ of cycles is an* optimal basis *for $H_p(K)$ if it is a basis and there is no other basis $\mathcal{C}'$ with $w(\mathcal{C}') < w(\mathcal{C})$.*

How can we compute such an optimal basis?

In a first step, we are going to compute a set of cycles $\mathcal{C}$ which *contains* an optimal basis. Then, we perform a greedy algorithm to find the optimal basis in a similar way to Kruskal's algorithm for Minimum Spanning Tree: we sort the cycles by increasing weight, and start our basis B with the trivial cycle 0. Then, we simply iterate through our cycles and add a cycle c_i to our basis if it cannot be written as a linear combination of our current basis. Finally, we remove 0 again from B.

The correctness of this greedy approach, as well as the correctness of many other greedy algorithms including the above-mentioned Kruskal, follows from a more general framework in *matroid theory*.

Exercise 8.2. *A* matroid *is a collection $\mathcal{I}$ of subsets of some universe U, such that*

1. *$\emptyset \in \mathcal{I}$, and if some set L is in $\mathcal{I}$, all $L' \subseteq L$ are also in $\mathcal{I}$.*

2. *If some L, L' are in $\mathcal{I}$, and $|L'| = |L| + 1$, then there exists an element $f \in L' \setminus L$, such that $L \cup \{f\} \in \mathcal{I}$.*

The sets in $\mathcal{I}$ are also called the independent sets *of the matroid. The inclusion-maximal sets in $\mathcal{I}$ are called* bases.

(a) *Show that for U being any finite set of vectors in some vector space, the family $\mathcal{I}$ of subsets of U corresponding to linearly independent vectors forms the family of independent sets of a matroid.*

(b) *Show that for any graph $G = (V, E)$, the family $\mathcal{I}$ of subsets of E corresponding to forests in G forms the family of independent sets of a matroid.*

(c) *Consider a matroid on a universe U with a weight function $w : U \to \mathbb{R}$. Consider the following greedy algorithm: begin with $L = \emptyset$, and consecutively add the lowest-weight element $e \notin L$ such that $L \cup \{e\}$ remains an independent set, until reaching a basis. Show that this greedy algorithm finds a minimum-weight basis.*

We now need to show that we can implement all of the pieces of the algorithm we outlined above. For the first step, we need to be able to compute our set $\mathcal{C}$. For the second greedy step, we need to be able to check linear independence of a set of cycles.

Since the problem of computing an optimal basis is NP-hard for $p > 1$, we will focus on computing a basis for $H_1(K)$. Without loss of generality, we say that K is 2-dimensional, with n triangles and $O(n)$ edges and vertices.

To compute $\mathcal{C}$, we begin with $\mathcal{C} = \emptyset$. For all vertices v, we compute the shortest path tree T_v rooted at v. We can do this for example with Dijkstra's algorithm. For every edge e that is not in T_v, we add the unique simple cycle in $T_v \cup \{e\}$ to $\mathcal{C}$. This can be implemented in $O(n^2 \log n)$, and yields a set of cycles with $|\mathcal{C}| \in O(n^2)$. But, we need to prove that it is indeed a set which contains an optimal basis.

Lemma 8.3. $\mathcal{C}$ *as computed by the method above contains an optimal basis.*

Proof. Let $\mathcal{C}^*$ be an optimal basis. Towards a contradiction, let c be a cycle contained in $\mathcal{C}^* \setminus \mathcal{C}$. Since we consider $\mathbb{Z}_2$-homology, c is a simple cycle, i.e., no edge is used multiple times.[1]

Let v be a vertex in c, and let T_v be the corresponding shortest path tree. There must be an edge $e = \{u, w\}$ in $c \setminus T_v$, since T_v is a tree. Let $\Pi_{v,u}$ and $\Pi_{v,w}$ be the shortest paths from v to u and w, respectively. These paths must be contained in T_v. Let us similarly consider $\Pi'_{v,u}$ and $\Pi'_{v,w}$ to be the paths from v to u, w in c which do not use the edge e. If we now had $\Pi'_{v,u} = \Pi_{v,u}$ and $\Pi'_{v,w} = \Pi_{v,w}$, c would be the unique simple cycle in $T_v \cup \{e\}$ and thus c would be in $\mathcal{C}$. However, since we assumed this not to be the case, w.l.o.g. we can assume that $\Pi'_{v,u} \neq \Pi_{v,u}$.

We now define the cycles $c_1 = \Pi'_{v,w} + \{e\} + \Pi_{v,u}$ and $c_2 = \Pi_{v,u} + \Pi'_{v,u}$. We can now see that as we work in $\mathbb{Z}_2$, $c_1 + c_2 = \Pi'_{v,u} + \{e\} + \Pi'_{v,w} = c$. Furthermore, we have $w(c_1) \leqslant w(c)$, since $\Pi_{v,u}$ is a shortest path (in K), while $\Pi'_{v,u}$ is not necessarily shortest. The same also holds for c_2: $w(c_2) \leqslant w(c)$ since $\{\Pi'_{v,w}, e\}$ can not be shorter than $\Pi_{v,u}$.

Let us now consider the homology classes of c_1 and c_2. If both $[c_1]$ and $[c_2]$ were dependent on $\mathcal{C}^* \setminus \{c\}$, then so would $[c]$, since $c = c_1 + c_2$. Then, $\mathcal{C}^*$ would not be a basis. Thus, at least one of $[c_1]$ and $[c_2]$ has to be independent of $\mathcal{C}^* \setminus \{c\}$. Let us consider first that c_1 is independent. Then, we could replace c by c_1 in $\mathcal{C}^*$ and get a basis which is at least as good as $\mathcal{C}^*$. If c_2 is independent, we replace c by c_2 in $\mathcal{C}^*$. After replacing, we use the same argument again on the new cycle. If we replaced c by c_1, we repeat the argument with v', the common ancestor of $\Pi_{v,u}$ and $\Pi'_{v,w}$ and the same edge e. If we replaced c by c_2, we repeat the argument with v' the common ancestor of $\Pi_{v,u}, \Pi'_{v,u}$ and e an edge incident to u. By doing this we have made some "progress": In the second iteration of the argument, one of the two paths Π' must have become equal to Π. After doing the argument twice, we must have replaced c by a cycle c' that has $w(c') \leqslant w(c)$ and c' is actually in our set $\mathcal{C}$.

At the end, we thus get a basis $\mathcal{C}'$ with $w(\mathcal{C}') \leqslant w(\mathcal{C}^*)$ with $\mathcal{C}' \subseteq \mathcal{C}$. Therefore $\mathcal{C}$ contains an optimal basis. $\qquad\square$

Exercise 8.4. *Given a 2-dimensional simplicial complex* K *in which every pair of vertices* u, v *has a unique shortest path between them, show that the set* $\mathcal{C}$ *computed by the algorithm will contain all optimal bases of* $H_1(K)$.

So, we have finished the first step of our algorithm. It remains to figure out how to check independence. For this, we introduce *annotations*.

Definition 8.5. *An* annotation of p-simplices *is a function* $a : K^p \to \mathbb{Z}_2^g$ *giving each p-simplex a binary vector of size* g. *This extends to chains by sums. An annotation must fulfill:*

- $g = \beta_p(K)$

- $a(z_1) = a(z_2)$ *iff* $[z_1] = [z_2]$.

---

[1]In general this would also follow from the weights being non-negative and $\mathcal{C}^*$ being optimal.

Given an annotation, we can now clearly check linear independence of cycles by simply checking linear independence of a set of vectors, for which we have existing tools such as Gaussian elimination.

Proposition 8.6. *In every simplicial complex* K *and for every* $p \geqslant 0$*, there exists an annotation of* p*-simplices, and such an annotation can also be computed.*

Proof. (Sketch for $p = 1$) We can compute a spanning forest T, and let m be the number of remaining edges. We initialize annotations of length m, and set $a(e) = 0$ for every edge in the spanning forest T. For every remaining edge e_i, we set $a_j(e_i) = 1$ if and only if $j = i$, and 0 otherwise.

For every triangle $t \in K$, if the annotation of its boundary δt is not 0, we find a non-zero entry b_u in $a(\delta t)$ and add $a(\delta t)$ to every edge with $a_u(e) = 1$, and we delete the u-th entry from all annotations. One can show that this yields a valid annotation, and it can be implemented in $O(n^3)$, and more clever implementations work in $O(n^\omega)$.   □

To check independence of cycles in our set $\mathcal{C}$ more efficiently, we add auxiliary annotations also to vertices in a shortest path tree T_v rooted at v. We give v the annotation 0, and for a vertex x that is the child of y, we set $a(x) := a(y) + a(e_{xy})$. For every cycle defined by the non-tree edge $e = uw$, we now have $a(c_e) = a(u) + a(w) + a(e)$. So, we never actually have to compute an explicit representation of a cycle by its edges, we only need to store its weight, the shortest path trees with the auxiliary annotations, and the non-tree edge e. Note that the auxiliary annotations can be computed in $O(gn)$ for the whole tree, thus in $O(gn^2)$ for all trees.

Finally, we have to check independence. Given an $(n \times m)$ matrix M, we can find the lexicographic leftmost set of independent columns in time $O(\max(n, m)^\omega)$. Instead of naively doing this n^2 times (once for every cycle), we group our cycles of $\mathcal{C}$ into groups A_i of size g, and compute the leftmost set for $[B|A_i]$, and thus we get $O(n^2 g^{\omega-1})$ runtime for this step.

To summarize, computing $\mathcal{C}$ takes $O(n^2 \log n)$, sorting the $O(n^2)$ cycles also takes $O(n^2 \log n)$, and for checking linear independence we need $O(n^\omega)$ for the annotations of the edges, $O(gn^2)$ for the auxiliary annotations, and $O(n^2 g^{\omega-1})$ for the block-wise linear independence checking. Overall, we thus get a runtime of $O(n^\omega + n^2 g^{\omega-1})$.

Theorem 8.7. *Given a 2-dimensional simplicial complex* K *with* n *triangles and a weight function* w *on its edges, we can compute an optimal basis of* $H_1(K)$ *in time* $O(n^\omega + n^2 g^{\omega-1})$.

8.2 Persistent Cycles

In the persistent setting, given a simplex-wise filtration $\mathcal{F}$ and an interval $[b, d]$, can we find an optimal persistent p-cycle c that is born at b and dies at d?

Sadly, this problem is already known to be NP-hard for $d < \infty$ and $p \geqslant 1$. However, if we assume that K is a weak $(p + 1)$-pseudomanifold, i.e., a simplicial complex in which

each p-simplex is a face of at most 2 (p+1)-simplices, then there exists a polynomial-time algorithm, which we will describe in this section.

If we consider cycles that live until ∞, we can solve the problem in polynomial time for $p = 1$, but it is NP-hard for $p \geqslant 2$. Here, the assumption of K being a weak $(p + 1)$-pseudomanifold does not save us. However, if we further assume that the complex can be embedded in $\mathbb{R}^{p+1}$, then it is again polynomial.

To solve the problem for $d < \infty$ in a weak $(p + 1)$-pseudomanifold, we consider undirected flow networks: We have a graph, where every edge has a capacity in $[0, \infty]$, some sources, and some sinks, and we want to find the maximum flow we can send from the sources to the sinks without sending too much flow through any edge. Recall that if we consider a cut which separates the sources from the sinks, the capacity of this cut is an upper bound on the value of the maximum flow. Furthermore, if we consider the minimum such cut, its capacity is equal to the value of the maximum flow. This can be solved in polynomial time.

We can build a dual graph G, by placing a vertex into every $(p + 1)$-simplex and adding an edge whenever they share a p-simplex. We furthermore add a dummy vertex which gets connected to all vertices which only have one neighbor. We are going to make the vertex belonging to the $(p + 1)$-simplex which is the destructor of our desired cycle the source. Furthermore, we make the dummy vertex as well as all vertices belonging to $(p + 1)$-simplices added after the destructor into sinks. Edges added at or before the birth are getting the capacity equal to their weight, while all other edges get capacity ∞. Then, it turns out that the p-simplices belonging to the edges in a minimum cut separating the sources from the sinks are an optimal persistent cycle.

Exercise 8.8. *Consider a simplex-wise filtration on a simplicial complex that is a weak $(p + 1)$-pseudomanifold, and consider some interval $[b, d]$ (for $d < \infty$) such that there exists a p-cycle born at b and dying at d. We look at the dual graph G with source and sinks defined as in the lecture. Consider a cut with finite capacity that separates the source from the sinks. Let c be the chain corresponding to the p-simplices dual to the edges going over this cut. Show that c is a p-cycle born at b and dying at d, and show that its weight is equal to the capacity of the cut.*

This exercise proves one direction of the correctness of the algorithm described above. The other direction is similar. We get the following result.

Theorem 8.9. *Given a a simplex-wise filtration on a simplicial complex that is a weak $(p + 1)$-pseudomanifold and an interval $[b, d]$ (for $d < \infty$), we can compute an optimal p-cycle born at b and dying at d in polynomial time.*

For details, we refer to Chapter 5 in the book of Dey and Wang [1].

Questions

32. **(This topic was not covered in this year's course in FS25 and therefore the following question will not be asked in the exam.)** *How can we compute an optimal basis given a set of*

cycles that contain one? Explain the algorithm described in Section 8.1. Further, explain annotations and how they can be used to check linear independence.

33. **(This topic was not covered in this year's course in FS25 and therefore the following question will not be asked in the exam.)** *How can we compute a set of 1-cycles that contain an optimal basis of* H_1 *?* Describe the algorithm to do this and prove its correctness.

34. **(This topic was not covered in this year's course in FS25 and therefore the following question will not be asked in the exam.)** *How can we compute an optimal persistent cycle?* Explain the algorithm described in Section 8.2.

References

[1] Tamal Krishna Dey and Yusu Wang, *Computational topology for data analysis*, Cambridge University Press, 2022.

Chapter 9

Multiparameter persistence

This chapter closely follows the introductory paper [2] on multiparameter persistence. The interested reader is referred there for a (much) more comprehensive overview of the topic.

As we have seen, the persistence modules arising from the Čech or Vietoris-Rips complexes are *stable* under small perturbations of the underlying data, but not *robust*. That is, even a small number of outliers can drastically change the persistence diagram or barcode. We could try to remedy this problem by making these complexes *density-aware* in some way. For the Vietoris-Rips complex, a typical way to do this is as follows. For a vertex v in a simplicial complex, its degree $\deg(v)$ is the number of edges (1-simplices) in the complex that contain v. Then. for $d \in \mathbb{N}$, the degree-d Vietoris-Rips complex is

$$\mathbb{VR}_d^r(X) := \{\sigma \in \mathbb{VR}^r(X) : \text{ each vertex of } \sigma \text{ has degree at least } d\}.$$

Note that vertices corresponding to data points in high-density areas of X will have relatively higher degree, whereas outliers will have relatively lower degree. Thus, for d large enough, we should expect this modification to reduce the impact of outliers. On the other hand, if d is too large, we are 'throwing away the baby with the bathwater'. So, the question is: how to choose d? Here, we run into the same issue that originally motivated persistence: there might not be one choice of d that accurately reflects the entire data set. Even if this choice would exist, it might be hard to determine. Instead, we would like to consider all choices of d simultaneously. The solution is to look at persistence w.r.t. both the *scale parameter* r and the *density parameter* d at the same time. In this chapter, we formalize such *multiparameter* persistence, and take a look at the representability and robustness of the resulting *multiparameter persistence modules*.

9.1 Persistence modules indexed by a poset

A persistence module (indexed by $\mathbb{R}$) consists of a family of vectors spaces U_a, $a \in \mathbb{R}$, together with commuting maps $U_a \to U_b$ for $a \leqslant b$. If we want to define persistence

modules indexed by more exotic sets, we need to first extend the meaning of '$\leqslant$'.

Definition 9.1. *A* poset *(partially ordered set) is a tuple* $(P, \preceq)$ *consisting of a set* P *and a binary relation* $\preceq$ *on* P *satisfying*

- $a \preceq a$ *for all* $a \in P$;

- $a \preceq b$ *and* $b \preceq a$ *implies* $a = b$ *for all* $a, b \in P$;

- $a \preceq b$ *and* $b \preceq c$ *implies* $a \preceq c$ *for all* $a, b, c \in P$;

We will sometimes simply write P *if the (partial) ordering* $\preceq$ *is clear from context.*

Note that partial orderings allow for elements to be *incomparable*, i.e., it can happen for $a, b \in P$ that neither $a \preceq b$ nor $b \preceq a$ holds. In contrast, under a *total ordering* all elements in P must be comparable. All the posets we will see in this chapter are constructed from the following examples:

- The reals $\mathbb{R} = (\mathbb{R}, \leqslant)$ and natural numbers $\mathbb{N} = (\mathbb{N}, \leqslant)$ with their usual ordering are posets.

- If $(P, \preceq)$ is a poset, and $Q \subseteq P$, then $(Q, \preceq)$ is a poset;

- For any poset $(P, \preceq)$, we have the *opposite poset* $P^{\mathrm{op}} = (P, \succeq)$, where

$$a \succeq b \iff b \preceq a \quad \text{for all } a, b \in P.$$

- For two posets $(P, \preceq_P)$ and $(Q, \preceq_Q)$ we have the *product poset* $(P \times Q, \preceq)$, where

$$(p, q) \preceq (p', q') \iff p \preceq_P p' \text{ and } q \preceq_Q q' \quad \text{for all } p, p' \in P \text{ and } q, q' \in Q.$$

We can now define filtrations and persistence modules indexed by an arbitrary poset, generalizing our definitions from earlier chapters.

Definition 9.2. *Let* $(P, \preceq)$ *be a poset. A family* $(X_p)_{p \in P}$ *of topological spaces (or simplicial complexes) is called a* P-*filtration if* $X_p \subseteq X_{p'}$ *for all* $p, p' \in P$ *with* $p \preceq p'$.

Definition 9.3. *Let* $(P, \preceq)$ *be a poset, and let* $\mathbb{F}$ *be a field. A* P-*persistence module (over* $\mathbb{F}$*) consists of*

- *An* $\mathbb{F}$-*vector space* U_p *for each* $p \in P$;

- *A linear map* $u_{p,p'} : U_p \to U_{p'}$ *for each* $p, p' \in P$ *with* $p \preceq p'$, *satisfying* $u_{p_2,p_3} \circ u_{p_1,p_2} = u_{p_1,p_3}$ *for all* $p_1, p_2, p_3 \in P$ *with* $p_1 \preceq p_2 \preceq p_3$.

Observation 9.4. *If* $(X_p)_{p \in P}$ *is a* P-*filtration for some poset* P, *then taking* k-*dimensional homology gives us a* P-*persistence module* $H_k(X_p)$ *(where, as before, the maps* $u_{p,p'} : H_k(X_p) \to H_k(X_{p'})$ *are induced by the inclusions* $X_p \hookrightarrow X_{p'}$*).*

97

We can now give two examples of filtrations indexed by $P = \mathbb{R} \times \mathbb{N}^{\mathrm{op}}$, each of which induces a P-persistence module via the observation above. The first one formalizes the discussion at the beginning of the chapter. The second one applies a similar idea to the Čech complex.

Example 9.5. *Let X be a finite metric space. Then the family $(\mathbb{VR}_d^r(X))_{r \in \mathbb{R}, d \in \mathbb{N}}$ defined above is a filtration indexed by $\mathbb{R} \times \mathbb{N}^{\mathrm{op}}$. It is called the* degree-Rips bifiltration*.*

Example 9.6. *Let $X \subseteq \mathbb{R}^n$ be a finite point cloud. For $r \in \mathbb{R}$ and $d \in \mathbb{N}$, define*

$$\mathcal{MC}_d^r = \{x \in \mathbb{R}^n : |B(x, r) \cap X| \geqslant d\} \subseteq \mathbb{R}^n.$$

Note that, for $d = 1$, this is just the union-of-balls used to define the Čech complex. The family $(\mathcal{MC}_d^r(X))_{r \in \mathbb{R}, d \in \mathbb{N}}$ is a filtration (of topological spaces) indexed by $\mathbb{R} \times \mathbb{N}^{\mathrm{op}}$. It is called the multicover bifiltration*.*

Both of the filtrations above can also be thought of as indexed over $\mathbb{R}^2$ (after a reparameterization). In general, we call an $\mathbb{R}^n$-filtration (resp. $\mathbb{R}^n$-persistence module) an n-parameter filtration (resp. n-parameter persistence module). For $n \geqslant 2$, we also say 'multiparameter'. For $n = 2$ the terms bifiltration and bipersistence module are common.

9.2 Representing persistence modules indexed by posets

9.2.1 Barcodes?

Recall that a p.f.d. persistence module $\mathbb{U}$ indexed by $\mathbb{R}$ can be uniquely represented by a barcode. In Chapter 6, we saw that this follows from the fact that any such $\mathbb{U}$ can be decomposed into interval modules in a unique way:

$$\mathbb{U} \cong \bigoplus_{i \in I} \mathbb{I}_{\langle a, b \rangle}.$$

Each interval in this decomposition corresponds to a bar in the barcode of $\mathbb{U}$. We could hope a similar statement holds for modules indexed by any poset P. The following definition and theorem should give us some hope:

Definition 9.7. *A persistence module $\mathbb{U}$ is called* indecomposable *if $\mathbb{U} \cong \mathbb{U}_1 \oplus \mathbb{U}_2$ implies that $\mathbb{U}_1 = 0$ or $\mathbb{U}_2 = 0$.*

Exercise 9.8. *Show that interval modules (indexed by $\mathbb{R}$) are indecomposable.*

Theorem 9.9. *Let $\mathbb{U}$ be a p.f.d. persistence module (indexed by a poset P). Then, there is a unique decomposition:*

$$\mathbb{U} \cong \bigoplus_{i \in I} \mathbb{U}_i,$$

where each persistence module $\mathbb{U}_i$ is indecomposable.

Understanding P-persistence modules thus boils down to understanding indecomposables (indexed by P). For $P = \mathbb{R}$, it turns out that indecomposables are precisely interval modules, leading to Theorem 6.36. However, for general P, the situation is much more complicated. Without going into details: It is not possible to parameterize indecomposables in terms of some nice family of subsets of P. The decomposition of Theorem 9.9 therefore does not lead to a workable notion of barcode in general. In fact, it is not possible to define any reasonable notion of a barcode for P-persistence modules, in the following sense.

Definition 9.10. *Let* $(P, \preceq)$ *be a poset and let* $\mathbb{U}$ *be a* P*-persistence module. We say a multiset* $\mathcal{B}$ *of subsets of* P *is a* reasonable barcode *for* $\mathbb{U}$ *if*

$$\text{rank}(u_{p,p'}) = |\{B \in \mathcal{B} : p, p' \in B\}| \quad (\forall p \preceq p').$$

That is, the rank of the map $u_{p,p'} : U_p \to U_{p'}$ *can be computed by counting the number of 'bars' that contain both* p *and* p'.

Exercise 9.11. *Show that the usual barcode for a p.f.d. persistence module indexed by* $\mathbb{R}$ *is reasonable in the above sense.*

Exercise 9.12. *For* $p \in P$, *show that* $\dim U_p$ *is greater than or equal to the number of bars that contain* p *in a reasonable barcode for* $\mathbb{U}$.

Example 9.13. *Let* $P = \{0, 1, 2\} \times \{0, 1, 2\}$ *and consider the following persistence module indexed by* P*:*

$$\mathbb{U} = \begin{array}{ccccc} \mathbb{F} & \xrightarrow{\text{id}} & \mathbb{F} & \longrightarrow & 0 \\ {\scriptstyle\text{id}}\uparrow & & {\scriptstyle g}\uparrow & & \uparrow \\ \mathbb{F} & \xrightarrow{f} & \mathbb{F}^2 & \xrightarrow{h} & \mathbb{F} \\ \uparrow & & {\scriptstyle j}\uparrow & & {\scriptstyle\text{id}}\uparrow \\ 0 & \longrightarrow & \mathbb{F} & \xrightarrow{\text{id}} & \mathbb{F} \end{array}, \quad \text{where} \quad \begin{array}{l} f : a \mapsto (a, 0) \\[4pt] g : (a, b) \mapsto a \\[4pt] h : (a, b) \mapsto a + b \\[4pt] j : a \mapsto (0, a) \end{array}$$

We claim $\mathbb{U}$ *cannot have a reasonable barcode. To see this, suppose* $\mathcal{B}$ *is a reasonable barcode for* $\mathbb{U}$*. Note that*

$$\text{rank}(h \circ f) = \text{rank}(g \circ f) = \text{rank}(h \circ j) = 1.$$

By the reasonability assumption, there thus must be subsets $I, J, K \in \mathcal{B}$ *with*

$$(0, 1), (2, 1) \in I, \quad (0, 1), (1, 2) \in J, \quad (1, 0), (2, 1) \in K.$$

Since $\dim U_{0,1} = \dim U_{2,1} = 1$, *we know by Exercise 9.12 that* $(0, 1)$ *and* $(2, 1)$ *occur in at most one element of* $\mathcal{B}$*. But that means that* $I = J$, *and* $I = K$, *and so in fact* $I = J = K \supseteq \{(0, 1), (2, 1), (1, 2)\}$*. Thus, using reasonability again, we find that*

$$\text{rank}(g \circ j) \geqslant 1,$$

contradicting the fact that $g \circ j = 0$.

For completeness, we note that it is possible to have reasonable barcodes for multiparameter persistence modules if we allow 'bars' to occur with *negative* multiplicity. This leads to so-called *signed barcodes*. Furthermore, there are several special classes of multiparameter persistence modules which do admit an interval decomposition (note that we did not formally define what an interval is in the poset-setting!). This is true in particular for so-called *interlevel persistence*. We will not discuss this further here.

9.2.2   Other representations and visualizations

Even though we cannot hope for a barcode-like representation of multiparameter persistence modules, there are still several useful (partial) representations available. We mention just three of them here:

- The *Hilbert-function* of a p.f.d. persistence module $\mathbb{U}$ indexed by P sends $p \in P$ to $\dim \mathbb{U}_p \in \mathbb{N}$. It can be thought of as a 'homological heatmap'.

- The *rank-invariant* of $\mathbb{U}$ sends a pair (p, p'), $p \preceq p'$ to the rank $\mathrm{rank}(u_{p,p'})$ of the map between $\mathbb{U}_p$ and $\mathbb{U}'_p$. Note that any two modules with $\mathbb{U} \cong \mathbb{V}$ have the same rank invariant. For modules indexed by $\mathbb{R}$, this implication also holds in the other direction. However, for general P, there exist nonisomorphic modules with the same rank-invariant;

- Let $L : t \mapsto ta + b$ be an affine line in $\mathbb{R}^2$ with non-negative slope. Then the restriction $\mathbb{U}_L$ of a persistence module $\mathbb{U}$ indexed by $\mathbb{R}^2$ to the line L can be viewed as a persistence module indexed by $\mathbb{R}$ (with its usual, total ordering). The *Fibered Barcode* of $\mathbb{U}$ sends each such line L to the barcode $\mathcal{B}(\mathbb{U}_L)$. This idea can be extended to lines in $\mathbb{R}^n$;

Exercise 9.14. *Show that the rank-invariant of an $\mathbb{R}^2$-persistence module can be recovered from its fibered barcode, and vice versa.*

The RIVET software package (rivet.readthedocs.io) allows you to compute and visualize each of the invariants above for (among others) the degree-Rips bifiltration.

9.3   Distances and robustness for P-persistence modules

As we generally do not have barcodes for P-persistence modules, we cannot rely on the Bottleneck distance to compare them. Instead, we work directly with interleaving distance, which is still well-defined. For simplicity, we only consider $P = \mathbb{R}^n$.

Definition 9.15. *Write $\mathbb{1} = (1, 1, \ldots, 1) \in \mathbb{R}^n$. Two $\mathbb{R}^n$-persistence modules $\mathbb{U}, \mathbb{V}$ are ϵ-interleaved if there exist two families of linear maps, $\varphi_a : \mathbb{U}_a \to \mathbb{V}_{a+\epsilon\mathbb{1}}$ and*

$\psi_a : V_a \to U_{a+\epsilon\mathbb{1}}$, $a \in \mathbb{R}^n$, *such that the following diagrams (and their symmetric counterparts obtained by exchanging the role of $\mathbb{U}$ and $\mathbb{V}$) are commutative:*

$$
\begin{array}{ccc}
U_a \xrightarrow{\quad u_{a,a'}\quad} U_{a'} & & U_a \xrightarrow{\quad u_{a,a+2\epsilon\mathbb{1}}\quad} U_{a+2\epsilon\mathbb{1}} \\
\downarrow{\scriptstyle \varphi_a} \quad\searrow\quad \downarrow{\scriptstyle \varphi_{a'}} & \textit{and} & {\scriptstyle \varphi_a}\searrow \quad \nearrow{\scriptstyle \psi_{a+\epsilon\mathbb{1}}} \\
V_{a+\epsilon\mathbb{1}} \xrightarrow{\ v_{a+\epsilon\mathbb{1},a'+\epsilon\mathbb{1}}\ } V_{a'+\epsilon\mathbb{1}} & & V_{a+\epsilon\mathbb{1}}
\end{array}
$$

The interleaving distance between $\mathbb{U}, \mathbb{V}$ is $d_I(\mathbb{U}, \mathbb{V}) := \inf_{\epsilon \geqslant 0}\{\mathbb{U}, \mathbb{V}$ are ϵ-interleaved$\}$.

The definition above agrees with our earlier definition of interleaving when $n = 1$. Contrary to the 1-dimensional case, computing the interleaving distance for $n \geqslant 2$ is an NP-hard problem (even if the persistence modules $\mathbb{U}, \mathbb{V}$ are given to us in a 'nice' form). For this reason, the following alternative based on the fibered barcode is sometimes used in practice. It is known to be efficiently computable when $n = 2$.

Definition 9.16. *The* matching distance *between p.f.d. $\mathbb{R}^n$-persistence modules $\mathbb{U}, \mathbb{V}$ is*

$$
d_{\mathrm{match}}(\mathbb{U}, \mathbb{V}) := \sup_L \ \left\{ d_B\big(\mathcal{B}(\mathbb{U}_L),\ \mathcal{B}(\mathbb{V}_L)\big) \right\},
$$

where the supremum is taken over all lines $L : t \mapsto ta + b$ with $a \succeq \mathbb{1}$, $b \in \mathbb{R}^n$.

Theorem 9.17 (see [3]). *For any two p.f.d. $\mathbb{R}^n$-persistence modules,*

$$
d_{\mathrm{match}}(\mathbb{U}, \mathbb{V}) \leqslant d_I(\mathbb{U}, \mathbb{V}).
$$

Exercise 9.18. *Prove that $d_B\big(\mathcal{B}(\mathbb{U}_L),\ \mathcal{B}(\mathbb{V}_L)\big) \leqslant d_I(\mathbb{U}, \mathbb{V})$ for all lines $L : t \mapsto t\mathbb{1} + b$.*

9.3.1   Robustness of some bipersistence modules

Finally, we come back to the primary reason for studying multiparameter persistence: achieving robustness w.r.t. outliers in the data. To state formal guarantees of this form, we need to think about point-cloud data in a more probablistic way. For a finite (multi)set $X \subseteq \mathbb{R}^n$, we write μ_X for the uniform probability measure on X, meaning the measure that assigns probability $1/|X|$ to each $x \in X$. The following can be thought of as a 'probabilistic Hausdorff distance' between point clouds.

Definition 9.19. *Let $X, Y \subseteq \mathbb{R}^n$ finite. The* Prohorov distance[1] *between μ_X, μ_Y is*

$$
d_{\mathrm{Pr}}(\mu_X, \mu_Y) := \sup_A \inf\{\delta \geqslant 0 : \mu_X(A) \leqslant \mu_Y(A^\delta) + \delta \ \text{ and } \ \mu_Y(A) \leqslant \mu_X(A^\delta) + \delta\},
$$

where A ranges over all closed subsets of $\mathbb{R}^n$ and $A^\delta := \{y \in \mathbb{R}^n : \mathrm{dist}(y, A) \leqslant \delta\}$.

---

[1]The Prohorov distance can be defined between any two measures μ, ν on (the same) metric space M.

While the definition of the Prohorov distance appears a bit complicated, the following observation suggests that it captures 'robustness to outliers' in a meaningful way.

Exercise 9.20. *Let $X \subsetneq Y \subseteq \mathbb{R}^n$, finite, non-empty sets. Show that*

$$d_{\mathrm{Pr}}(\mu_X, \mu_Y) \leqslant \frac{|Y \setminus X|}{|X|}.$$

As an example, the *normalized* multicover persistence module turns out to be stable w.r.t. Prohorov distance (i.e., it is robust to outliers).

Definition 9.21. *Let $X \subseteq \mathbb{R}^n$ be a finite point cloud. For $r, \rho \in \mathbb{R}$, the sets*

$$\mathcal{NMC}_\rho^r(X) := \{x \in \mathbb{R}^n : |B(x, r) \cap X| \geqslant \rho|X|\} = \{x \in \mathbb{R}^n : \mu_X\big(B(x, r) \cap X\big) \geqslant \rho\}.$$

form a bilfiltration over $\mathbb{R} \times \mathbb{R}^{\mathrm{op}}$ called the normalized multicover bifiltration.

Theorem 9.22 (see [1]). *Let $X, Y \subseteq \mathbb{R}^n$ finite. For all $k \geqslant 0$, we have*

$$d_I\big(H_k(\mathcal{NMC}(X)),\ H_k(\mathcal{NMC}(Y))\big) \leqslant d_{\mathrm{Pr}}(\mu_X, \mu_Y).$$

Proof. Let $\epsilon = d_{\mathrm{Pr}}(\mu_X, \mu_Y)$. We will show that

$$\mathcal{NMC}_\rho^r(X) \subseteq \mathcal{NMC}_{\rho-\epsilon}^{r+\epsilon}(Y) \subseteq \mathcal{NMC}_{\rho-2\epsilon}^{r+2\epsilon}(X) \quad \forall r, \rho \in \mathbb{R}.$$

These inclusions induce the maps

$$\varphi_{r,\rho} : H_k(\mathcal{NMC}_\rho^r(X)) \to H_k(\mathcal{NMC}_{\rho-\epsilon}^{r+\epsilon}(Y)),$$
$$\psi_{r+\epsilon,\rho-\epsilon} : H_k(\mathcal{NMC}_{\rho-\epsilon}^{r+\epsilon}(Y)) \to H_k(\mathcal{NMC}_{\rho-2\epsilon}^{r+2\epsilon}(X))$$

required to show that $H_k(\mathcal{NMC}(X))$ and $H_k(\mathcal{NMC}(Y))$ are ϵ-interleaved.

So, let $r, \rho \in \mathbb{R}$, and suppose that $x \in \mathcal{NMC}_\rho^r(X)$. By definition, this means that $\mu_X(B(x, r) \cap X) \geqslant \rho$. Using the definition of the Prohorov distance, and the triangle-inequality, this implies that

$$\rho \leqslant \mu_X\big(B(x, r) \cap X\big) \leqslant \mu_Y\big(B(x, r)^\epsilon \cap Y\big) + \epsilon \leqslant \mu_Y\big(B(x, r+\epsilon) \cap Y\big) + \epsilon.$$

That is to say, $\mu_Y(B(x, r+\epsilon) \cap Y) \geqslant \rho - \epsilon$, meaning $x \in \mathcal{NMC}_{\rho-\epsilon}^{r+\epsilon}(Y)$, giving us the first inclusion. The second inclusion follows from an analogous argument, switching the roles of X and Y. $\square$

There are many similar theorems for other bipersistence modules, including the degree-Rips bifiltration. They are often somewhat hard to state and prove, relying on variants of the Prohorov distance, as well as variants of the interleaving distance [1].

Questions

35. *How can we define persistence modules over multiple parameters?* Discuss the technical definitions that are needed.

36. *What are some common bipersistence modules? How do they relate to the (1-parameter) persistence modules of earlier chapters?*

37. *What is a major upside of multiparameter persistence? What is a major downside?* Discuss robustness and representability.

38. *How can we visualize multiparameter persistence modules?*

References

[1] Andrew J. Blumberg and Michael Lesnick, Stability of 2-parameter persistent homology. *Found. Comput. Math.*, 24/2, (2024), 385–427.

[2] Magnus Bakke Botnan and Michael Lesnick, An Introduction to Multiparameter Persistence. 2023.

[3] Claudia Landi, The rank invariant stability via interleavings. In *Research in computational topology*, vol. 13 of *Assoc. Women Math. Ser.*, pp. 1–10, Springer, Cham, 2018.

Chapter 10

Applications

In this chapter we highlight some classical and recent applications of topological data analysis. The fields of application are as diverse as image analysis, medicine, analysis of time series and politics. Many of these applications are also described in the book "Topological Data Analysis with Applications" by Gunnar Carlsson and Mikael Vejdemo-Johansson [2]. We also discuss some of the approaches to use persistence diagrams in machine learning.

10.1 The Space of Image Patches
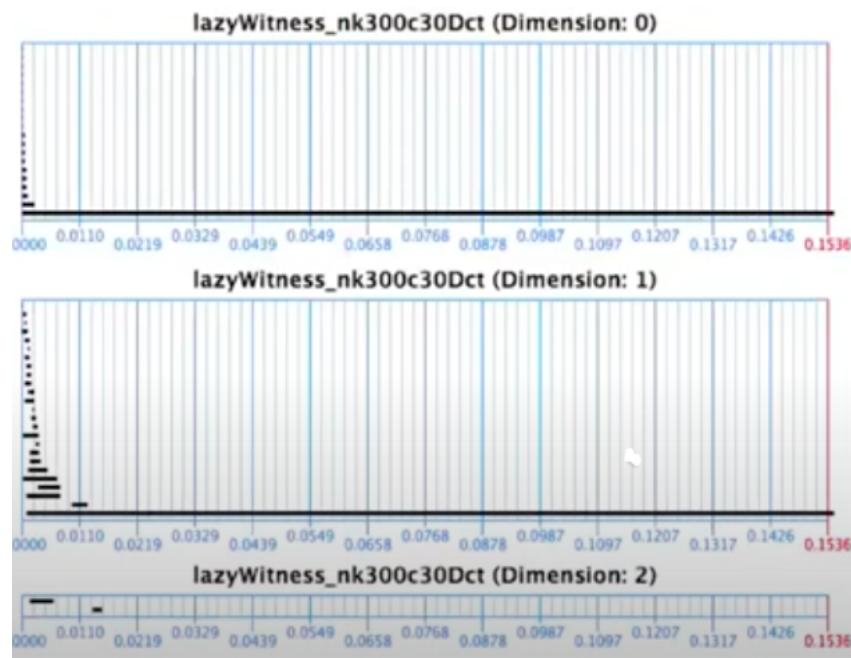
We consider a dataset of about 4000 greyscale images taken around Groningen (Netherlands) by van Hateren and van der Schaaf [8], see Figure 10.1 for some examples. Each such image is described by giving each pixel a number between 0 (white) and 1 (black). From every image, they sampled 5000 patches of 3-by-3 pixels. To get interesting patches, they only looked at the top 20% with the highest contrast. Thus, we end up with about 4 million three by three patches, each describable as a vector in $\mathbb{R}^9$. They further normalize the contrast (such that the darkest and brightest pixel are 1 and 0, respectively) and the overall norm of the vectors. Thus, we end up with points on S^7. How does this point cloud look? Is there some interesting structure in this point cloud?

Carlsson, Ishkhanov, de Silva and Zomorodian analyzed this point cloud using persistent homology [1]. Since there are a lot of points, they at first only sampled from the densest parts of the data. Computing the persistent homology of the Witness complex in dimensions 0, 1, and 2, one gets the barcodes depicted in Figure 10.2. It is thus a reasonable assumption that $\beta_0 = 1$, $\beta_1 = 1$, and $\beta_2 = 0$, suggesting that the data set looks like a circle. PCA also shows a circular arrangement of the data points, see Figure 10.3. This circle can be interpreted as the possible angles of a detected "edge" in the images, depicted in Figure 10.4.

What happens if we additionally also sample some points from parts of the data of intermediate density? Computing the persistent homology we end up with the barcodes in Figure 10.5, suggesting that instead of one 1-cycle, we end up with five. If we again

Figure 10.1: *Some examples of images in the data set.*

Figure 10.2: *The barcodes of the densest part of the data.*
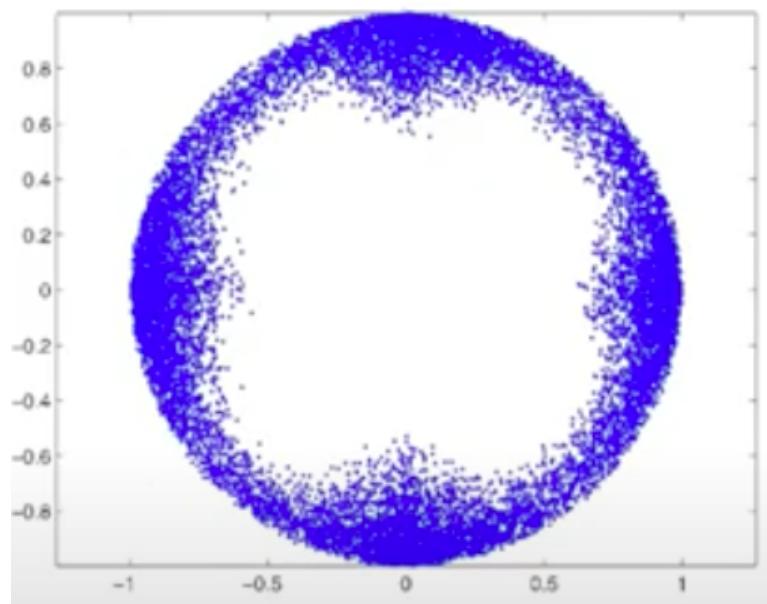
**Figure 10.3**: *Using PCA on the densest part of the data.*
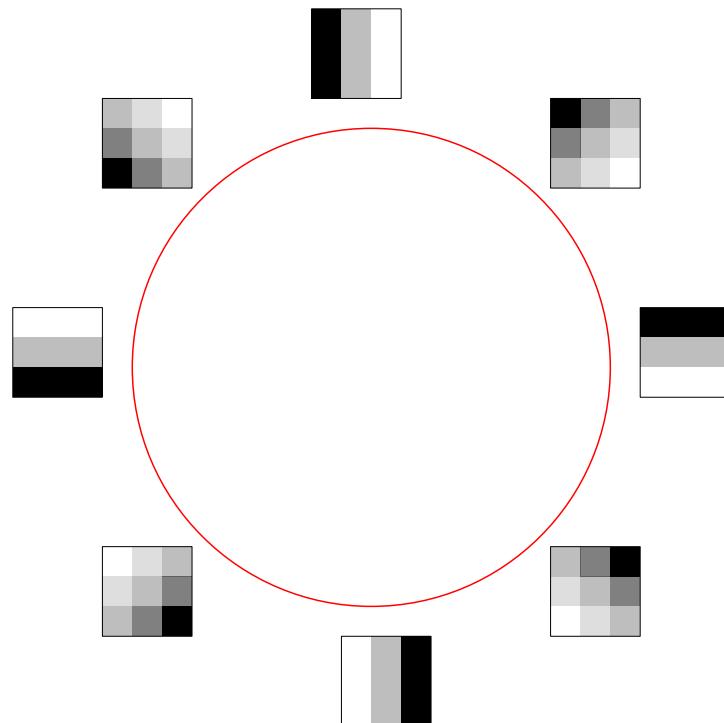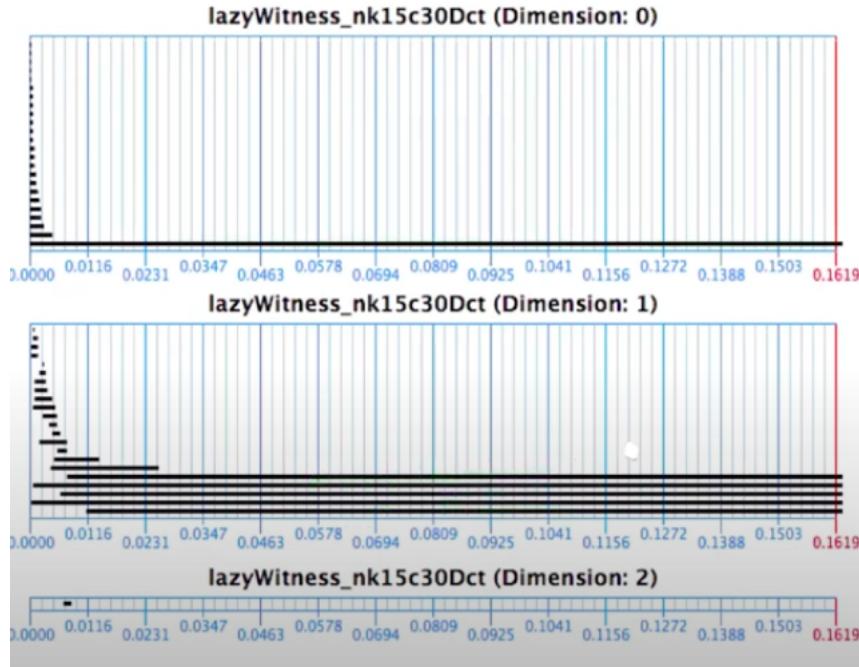
Figure 10.4: *The interpretation of the densest part of the data.*

Figure 10.5: *The barcodes including the medium density part of the data.*

consider the PCA, depicted in Figure 10.6, we see a cross in the middle of the circle seen previously. However, this only shows four circles! Where is the fifth one? Most likely some cycle is orthogonal to the projection chosen by the PCA. The interpretation picked from this situation is that the dataset consists of three great circles of some S^2 that are all orthogonal. We can interpret these three circles as the circle described above, a circle describing different translations of vertical edges, and one circle describing translations of horizontal edges. This is depicted in Figure 10.7.

However, if we finally also include points from the lowest density parts of the data, we get the barcodes depicted in Figure 10.8. We start seeing some feature in the persistent 2-homology, and reduce β_1 to most likely 2 again. How can this space look? It must somehow include the three circles found before. We can embed the circles as in the Figure 10.9, indicating that the space is a Klein bottle.

Clearly, this is not a proof. The persistent homology computed agrees with the Klein bottle, but it would also agree with a torus. To give more evidence, we could for example compute persistent homology over a different field such as $\mathbb{Z}_3$, with which we can distinguish between a Klein bottle and a torus. It turns out that the space is indeed a Klein bottle, as is shown in the work of Carlsson et al. [1].

10.2 Mapper for Medical Data

The following application comes from the original paper describing the Mapper approach [6]. The data comes from a diabetes study in 1979 on 145 participants, with six quantities
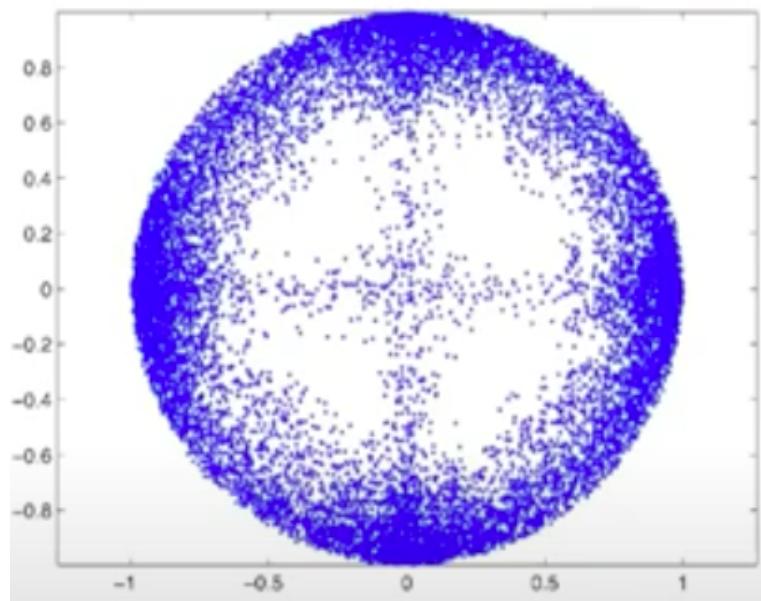
**Figure 10.6**: *Using PCA on the high and medium density part of the data.*
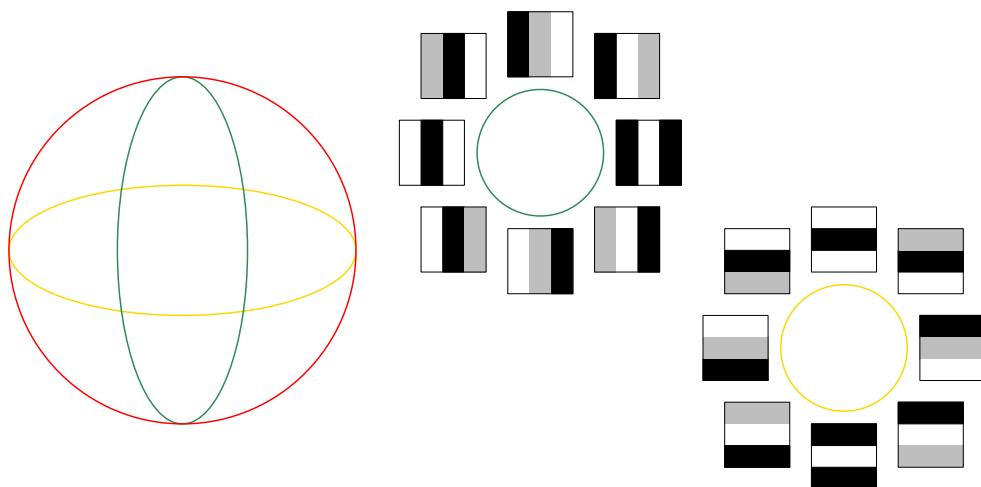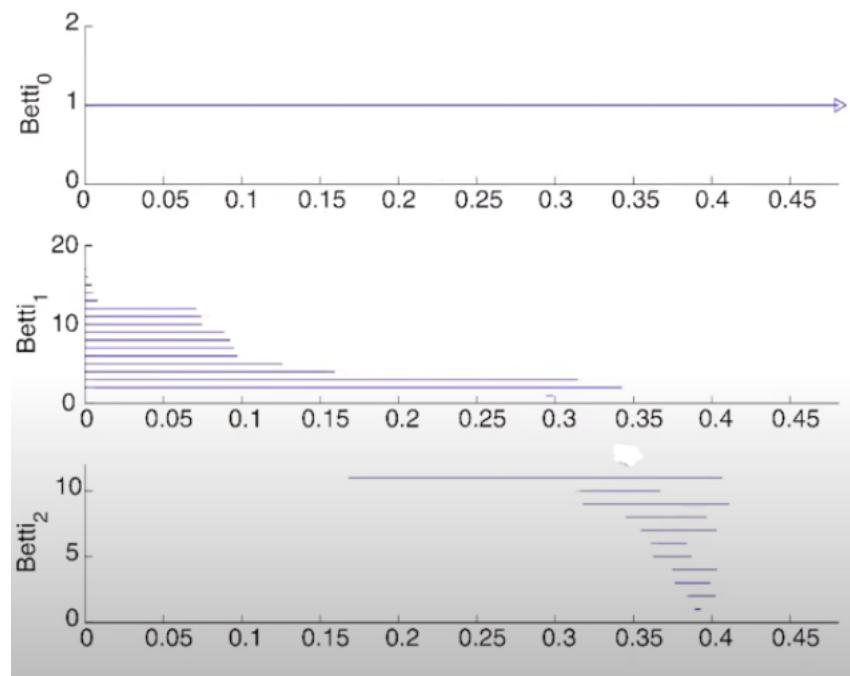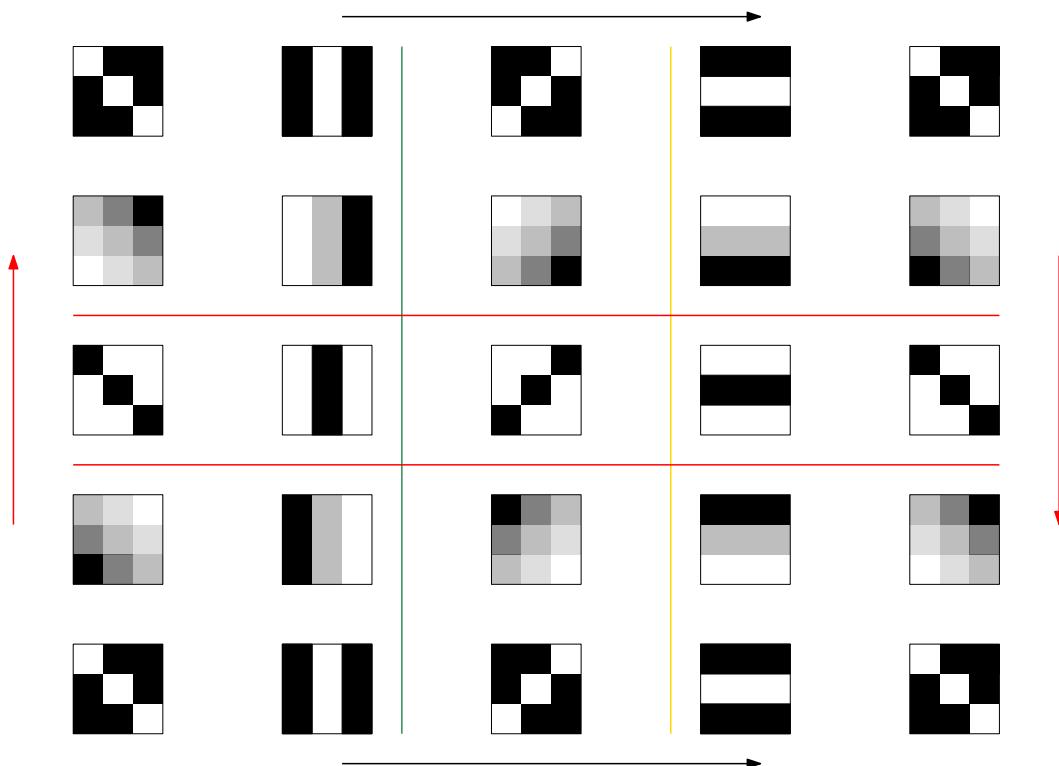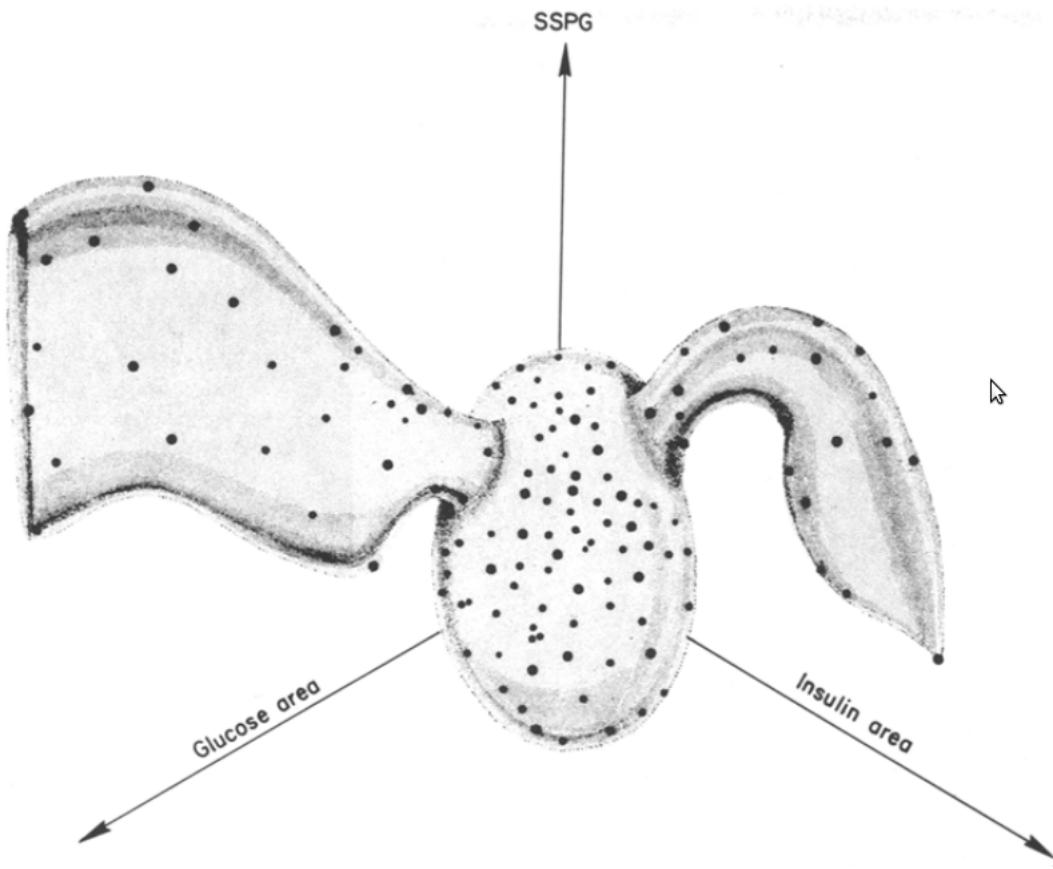
Figure 10.7: *The interpretation of the high and medium density part of the data.*

Figure 10.8: *The barcodes for the entire data.*

**Figure 10.9:** *The interpretation of the entire data.*

Figure 10.10: *The classical illustration of the diabetes data set.*

measured per participant: age, relative weight, fasting plasma glucose, area under the plasma glucose curve for the three hour glucose tolerance test (OGTT), area under the plasma insulin curve for the (OGTT), and steady state plasma glucose response. After applying various classical methods, the original study came up with a picture containing a blob of healthy people, with two strains coming out of this blob, called type 1 and type 2 diabetes [4]. See Figure 10.10 for their illustration. The same behaviour can be automatically detected using Mapper, as showcased in [6]. As the filter function they used a density estimator for each point. They created two different outputs, one for 3 and one for 4 intervals, always with a 50% overlap. In the output of Mapper depicted in Figure 10.11 you also see areas of large density, with 2 flares going out of it.

   The Mapper approach has later also been applied to genomic data from a breast cancer study [7] by Nicolau, Levine and Carlsson [5]. This data is very high dimensional, each patient defines a data point in $\mathbb{R}^{262}$. As a filter function they used the distance to some baseline healthy tissue. The output presented in their paper is depicted in Figure 10.12. The strain labelled *c-MYB$^+$ tumors* corresponds to a type of tumor which has previously not been classified, where the other strains confirm the strains of breast cancer known previously.
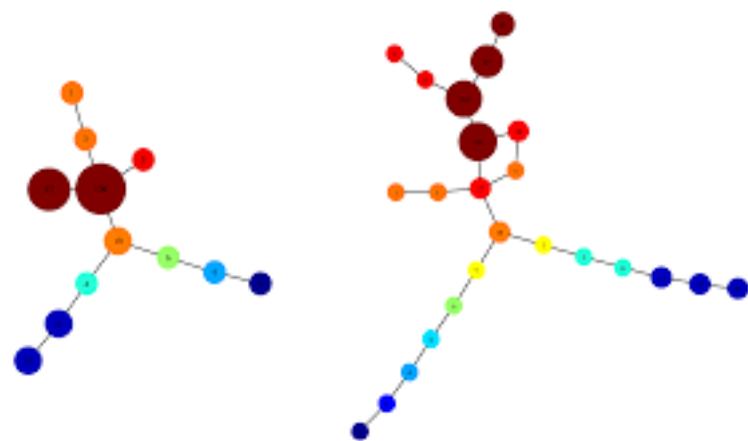
**Figure 10.11:** *The output of Mapper on the diabetes data set.*
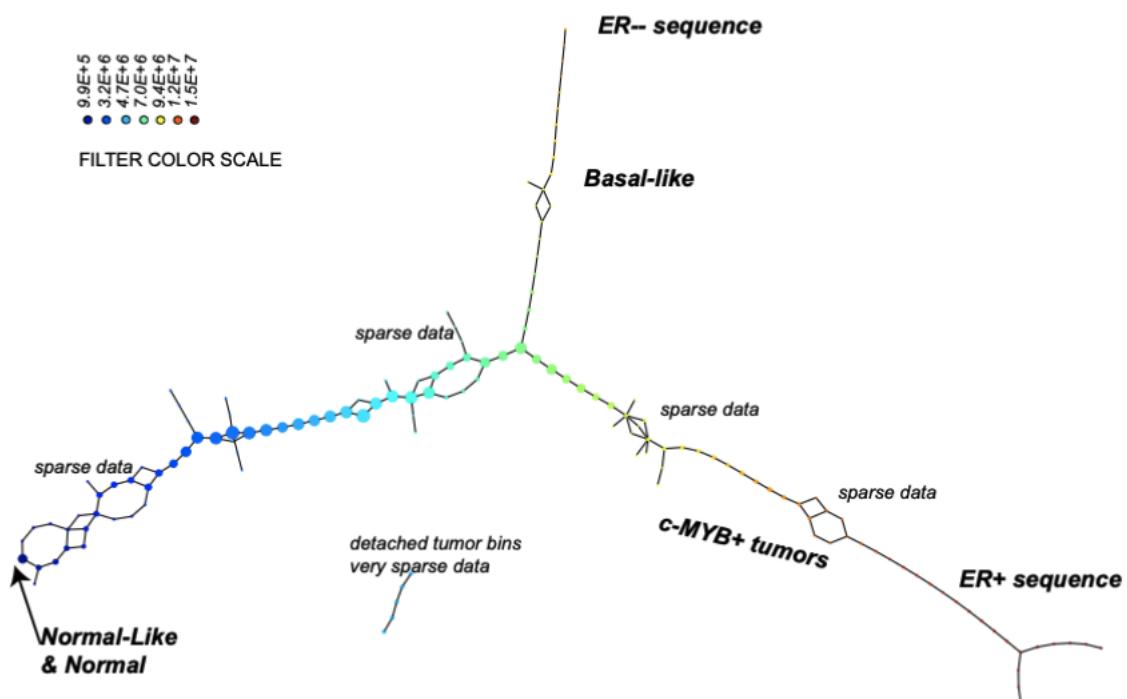
Figure 10.12: *The output of Mapper on the breast cancer data set.*

Topological data analysis has been used in many studies in medicine. TDA seems to excel in these high-dimensional datasets since many features seem to be the result of higher-order interactions of different coordinates. TDA can also deal with much smaller data sets than other approaches such as deep neural networks, that need tons of data points to train, validate, etc. In medicine, studies often have very few data points due to the large monetary cost, workload, and also ethical questions involved with data gathering.

10.3   Time Series

Time series is data given as a sequence of points x_t in some metric space X, where t is a (discrete) variable. The goal in analyzing time series is often analyzing and finding periodic behavior in the time series.

We can embed time series in some higher-dimensional space, by always considering a sequence of $l+1$ consecutive points, i.e., $\mathcal{I}_l := \{(x_{t_i}, x_{t_{i+1}}, \ldots, x_{t_{i+l}})\} \subset X^{l+1}$. The idea is that periodicity in the time series translates to loops (1-dimensional holes) in $\mathcal{I}_l$. As an example, consider $x_t = \sin(\frac{\pi}{4}t)$, and consider $l = 1$. Then, we get a loop in $\mathbb{R}^2$. If we however consider $x_t = t$, $\mathcal{I}_1$ is just a straight line.

This approach has been used by Vejdemo-Johansson, Pokorny, Skraba and Kragic [9] for analyzing motion capture data, where cameras track the location in 3-space of certain points on a human body marked by physical markers. The data considered are six seconds long recordings of movement such as boxing, see Figure 10.13 for some stills. Every data point in this set is roughly 70-dimensional. Using PCA to project onto 2 dimensions, some loops can be seen but we cannot really distinguish different loops, see Figure 10.14 (left). Computing persistent homology it can be seen that there are six different loops that persist a long time, see Figure 10.14 (right). Looking back at the input data, these six different loops can be mapped to six different boxing movements in the recording.

10.4   Politics

The data considered in this application, done in [2], is gathered from the sessions of the US house of representatives through the year 2010. For every vote, we set $x_i = 1$ if the member i says Yes to the vote, -1 for No, and 0 otherwise. Doing this for every of the 447 representatives in all 664 votes this gives 664 data points in $\mathbb{R}^{447}$. Using PCA to project to two dimensions and coloring the data by splitting the members into the two parties we get the picture depicted in Figure 10.15. We can see that there are four main "corners" in which issues lie: the ones that get bipartisan support, those that get bipartisan rejection, those that are supported by republicans and rejected by democrats, and those where it is the opposite way. We can also see that the corner with bipartisan rejection is very sparsely populated, since such issues rarely make it far enough to be voted on. Further there are few votes that lie in between two of these corners, and almost none that lie
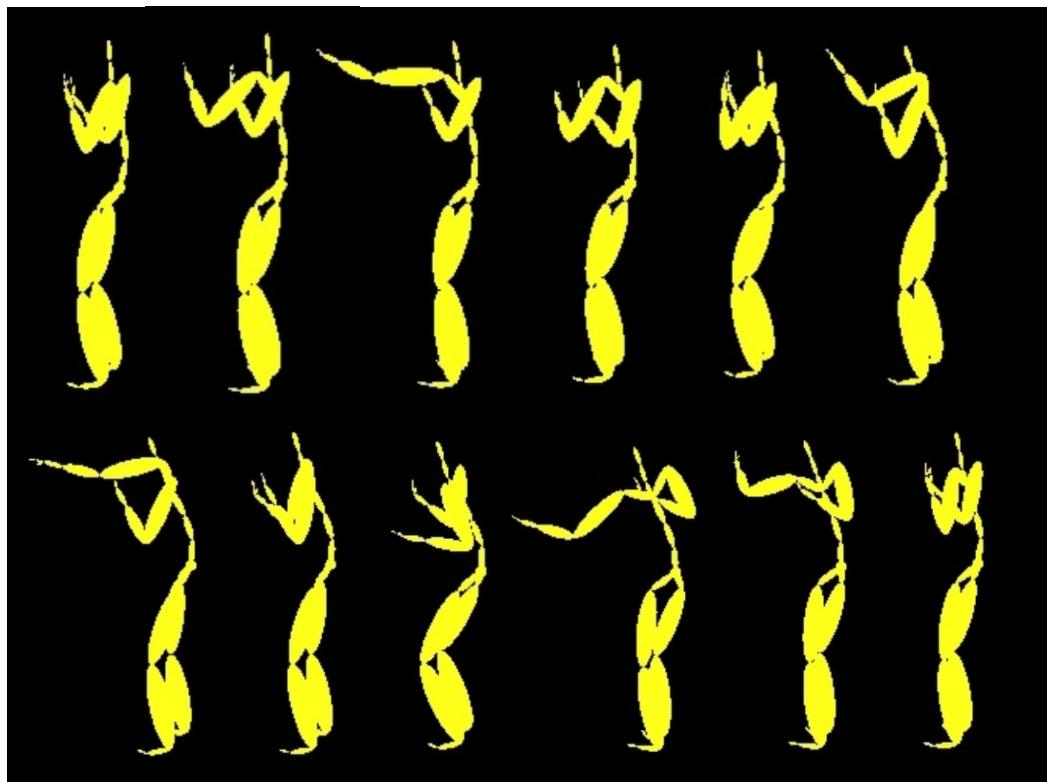
**Figure 10.13:** *Some stills from motion capture data of a boxer.*
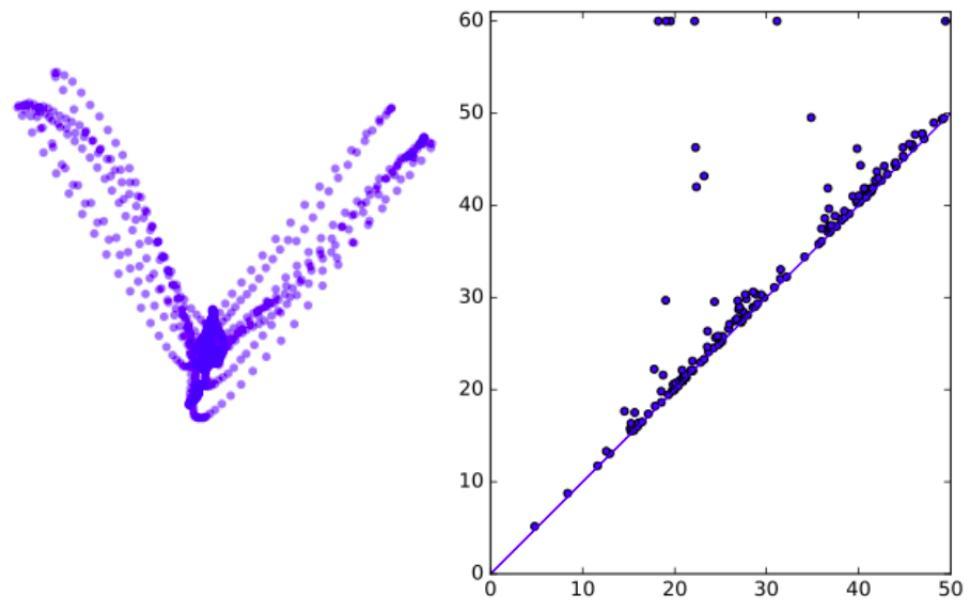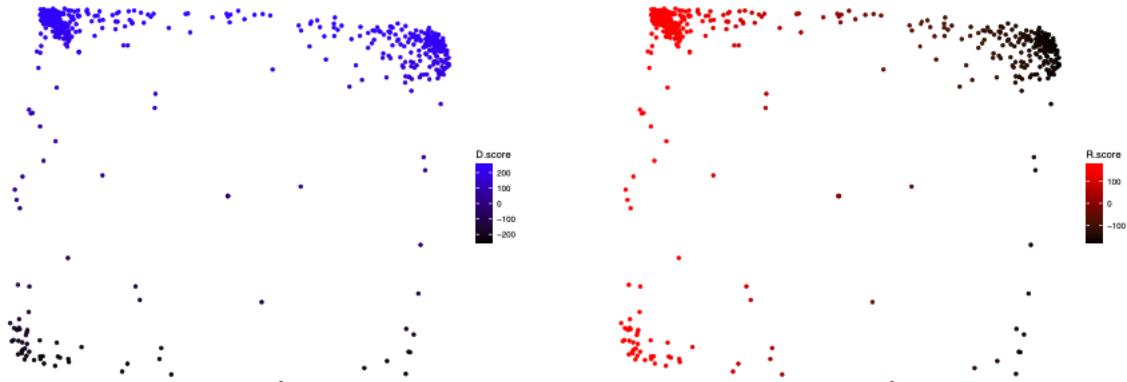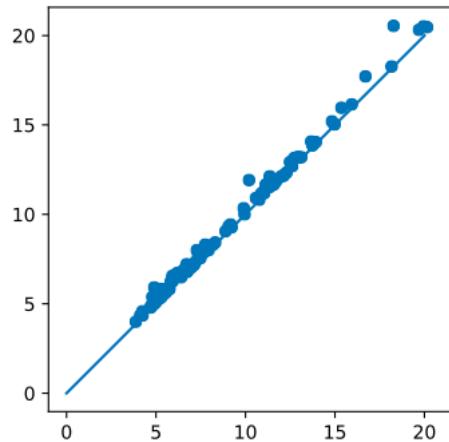
Figure 10.14: *Left: PCA of the data set obtained from the boxing recording. Right: the persistent homology of the same data.*
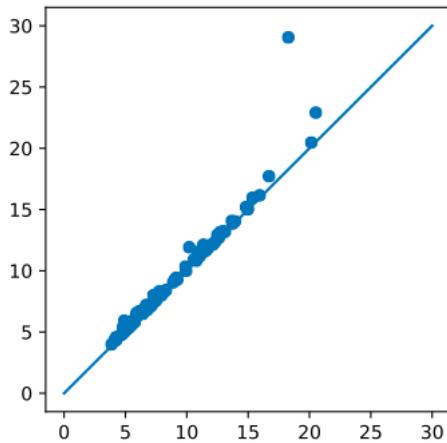
**Figure 10.15**: *PCA of the voting data from the US house of representatives, colored by parties.*

Figure 10.16: *Persistence diagram in dimension 1, computed on the entire voting data.*

in the middle of all four corners. There are however some: in particular, three of the recorded votes where Quorum calls, where both answers (Present/Not Present) are coded as 0, so there are at least 3 points exactly in the middle of the square.

Computing the 1-dimensional persistence diagram, we would hope to recognize the phenomenon of excluded middle by having a 1-dimensional homology class with long persistence, corresponding to the boundary of the square. However, as we can see in the diagram depicted in Figure 10.16, this is not the case. What is going on? If you think about the process of growing balls, it becomes apparent that already a single point in the center of the square is enough to "fill the hole" much faster than we would like. As you can see in Figure 10.15, there are several points in the middle of the square, and it is these points that make the persistence of any 1-dimensional homology class short. This is a general issue of persistent homology: it is very fragile to outliers in the data.

Figure 10.17: *Persistence diagram in dimension 1, computed on 99% of the voting data.*

One general approach to overcome this issue in many application is the following: compute the *density* for each data point (e.g., using Gaussian kernel density estimation or any other standard method) and only keep the points with highest density, e.g., remove the 1% with lowest density. Then, compute persistent homology only on the remaining, dense points. Doing this on the voting data, we get the diagram depicted in Figure 10.17. Here we can clearly see a single homology class with long persistence, just as we would expect.

10.5   Shape Segmentation

Consider a surface in $\mathbb{R}^3$ given as a triangular mesh, that is, by points, edges, and triangles. The goal of *shape segmentation* is to label different parts of the surface according to what part they are. For example, given a model of a human body, we want to segment it into categories such as "head", "torso" or "upper arm". While this is a well-studied problem in computer graphics, here we sketch a topological approach, described in [3].

We can pick some point on the body, and start growing a ball around it, using the geodesic distance (length of the shortest path along the surface). On the resulting filtration, we can perform persistent homology in dimension 1. If we do this for a point on the palm of a hand, for example, we get a 1-dimensional hole for every finger, see Figure 10.18. If we do this for a point on a finger, the persistence diagram looks very different, see Figure 10.19. We can then classify the persistence diagrams to segment the shape. But, how can we do this? We need to be able to insert persistence diagrams into classical ML pipelines. For this there are several approaches, and we discuss some of them in the next section.
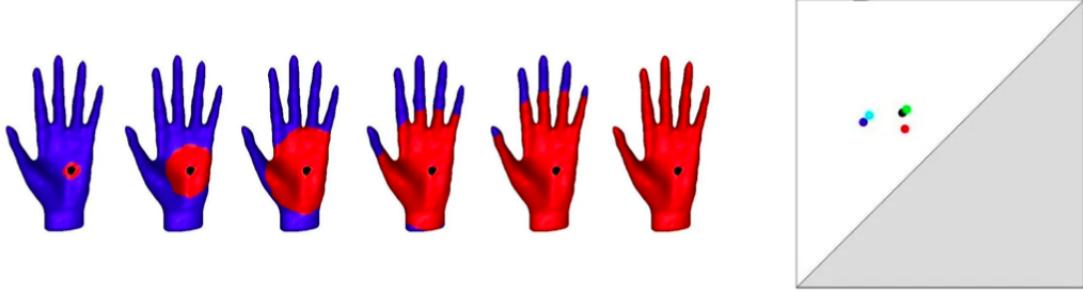
**Figure 10.18:** *Persistence diagram for a point on the palm.*
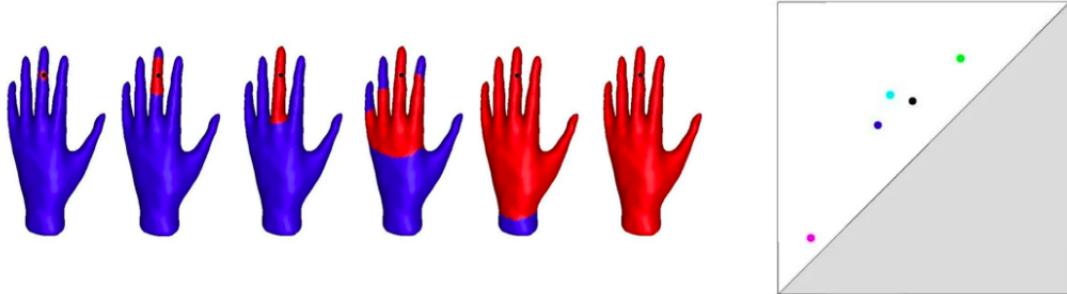
Figure 10.19: *Persistence diagram for a point on a finger.*
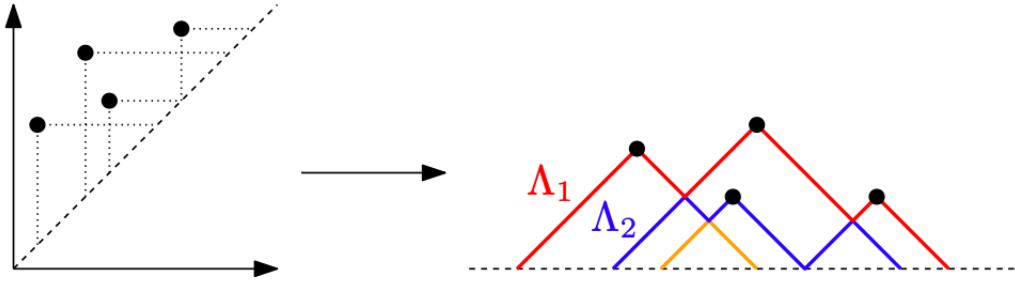
10.6   TDA in ML

Many machine learning pipelines require input points to be in Euclidean space (and not just any metric space, which the persistence diagrams would already fulfill), or in the case of kernel methods, at least in some space that has an inner product, also known as a *Hilbert space*.

There are many ways to turn persistence diagrams into elements of such metric spaces. These methods are also called *vectorizations*. In this section we introduce three such methods. On https://persistent-homology.streamlit.app you can see more examples and play around with various vectorization methods.

Persistence Statistics   For the persistence statistics, as the name suggests, we just summarize some statistical values of the persistence diagrams. In particular, for the birth times b, the death times d, the interval midpoints $\frac{b+d}{2}$ and the interval lengths $d - b$ we record

- the mean,

- the standard deviation,

- the median,

- the interquartile range,

Figure 10.20: *The persistence landscape of a persistence diagram.*

- the full range and

- four percentiles (10%, 25%, 75%, 90%).

Finally we also record the number of bars and the entropy. All in all, we record 38 numerical values, and we thus represent a persistence diagram as a point in $\mathbb{R}^{38}$.

Persistence Landscapes The main intuition behind persistence landscapes is the following: for each point in the persistence diagram draw a horizontal and a vertical line segment to the diagonal. Flipping this arrangement of line segments such that the diagonal is horizontal, we get something that looks like a mountain range, hence the name landscape. We can also interpret this as a set of piece-wise linear functions by looking at *envelopes*: a point on one of the segments is on the k'th envelope of the arrangement if there are $k-1$ segments strictly above it. Each such envelope is now a piece-wise linear function. See Figure 10.20 for an illustration.

More formally, let $D = \{(b_1, d_1), \ldots, (b_n, d_n)\}$ be a persistence diagram with finitely many off-diagonal points. Each off-diagonal point (b_i, d_i) gives rise to a triangle whose boundary is defined by the points

$$\{(t, \min[t - b_i, d_i - t]_+) \mid t \in \mathbb{R}\},$$
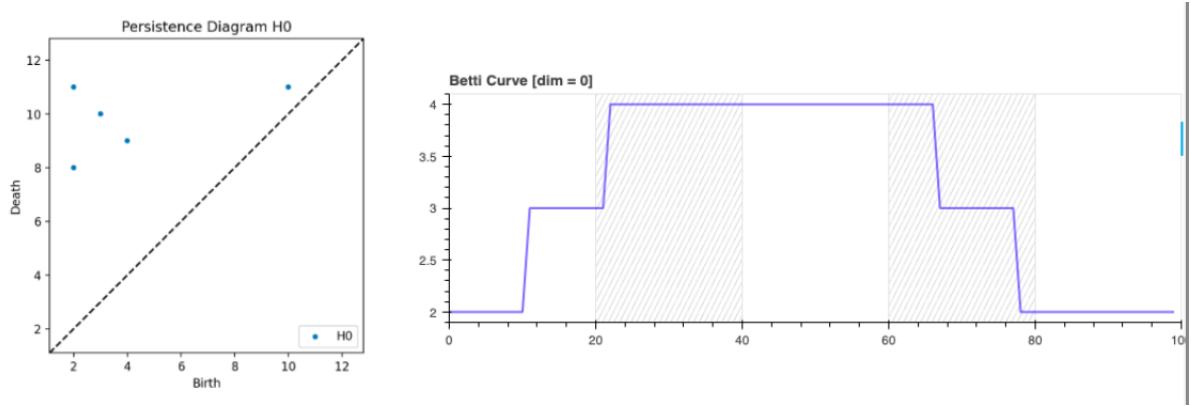
where $\min[a, b]_+ := \max(0, \min(a, b))$.

The persistence landscape of D is now a function $\lambda_D : \mathbb{N} \times \mathbb{R} \to \mathbb{R}$ defined by

$$\lambda_D(k, t) := \text{k'th largest value of } \min[t - b_i, d_i - t]_+ \text{ for } i \in \{1, \ldots, n\}.$$

For each k, we have that $\lambda_D(k, \cdot)$ is a piece-wise linear function. Recall that on functions, we have the following p-*norms*:

$$\|f\|_p := \left(\int_\mathbb{R} |f(x)|^p dx \right)^{1/p}.$$

Figure 10.21: *The Betti curve of a persistence diagram.*

We can extend this to a norm on our set of piece-wise linear functions:

$$\|\lambda_D\|_p := \left(\sum_{k=1}^{\infty} \|\lambda_D(k, \cdot)\|_p \right)^{1/p}.$$

In particular, for $p = 2$ this is a Hilbert space and thus usable for many machine learning pipelines. We can further define the following distance measure on persistence diagrams, called the p-*landscape distance* Λ_p:

$$\Lambda_p(D_1, D_2) := \|\lambda_{D_1} - \lambda_{D_2}\|_p.$$

Taking $p = \infty$ we get the following

Theorem 10.1. *Let D_1 and D_2 be persistence diagrams with finitely many off-diagonal points. Then $\Lambda_\infty(D_1, D_2) \leqslant d_b(D_1, D_2)$.*

Betti Curves   An easier way to get a piece-wise linear (even piece-wise constant) function from a persistence diagram is through Betti curves: the Betti curve of a persistence diagram is the function $\beta : \mathbb{R} \to \mathbb{N}_{\geqslant 0}$ which assigns each time t the current Betti number of the filtration. This is a piece-wise constant function and thus the space of all Betti curves is a Hilbert space with the standard 2-norm. A Betti curve still captures all the births and deaths, but throws away the pairing between them, see Figure 10.21 for an illustration.

Questions

39. *How can we use TDA to analyze the space of image patches?* Discuss the structure of the relevant data set and how persistent homology can be applied.

40. *Why does persistent homology sometimes fail to capture voids in the data?* Illustrate the problem with the voting data from the US house of representatives and explain how it can be solved in practice.

41. *How can persistence diagrams be used in machine learning pipelines?* Explain the following vectorizations: persistence statistics, persistence landscapes and Betti curves.

References

[1] Gunnar Carlsson, Tigran Ishkhanov, Vin De Silva, and Afra Zomorodian, On the local behavior of spaces of natural images. *International journal of computer vision*, 76, (2008), 1–12.

[2] Gunnar Carlsson and Mikael Vejdemo-Johansson, *Topological Data Analysis with Applications*, Cambridge University Press, 2021.

[3] Mathieu Carrière, Steve Y. Oudot, and Maks Ovsjanikov, Stable Topological Signatures for Points on 3D Shapes. *Computer Graphics Forum*, doi:10.1111/cgf.12692.

[4] Rupert G Miller, Discussion: Projection Pursuit. *The Annals of Statistics*, 13/2, (1985), 510–513.

[5] Monica Nicolau, Arnold J Levine, and Gunnar Carlsson, Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival. *Proceedings of the National Academy of Sciences*, 108/17, (2011), 7265–7270.

[6] Gurjeet Singh, Facundo Mémoli, and Gunnar Carlsson, Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object Recognition.

[7] Marc J Van De Vijver, Yudong D He, Laura J Van't Veer, Hongyue Dai, Augustinus AM Hart, Dorien W Voskuil, George J Schreiber, Johannes L Peterse, Chris Roberts, Matthew J Marton et al., A gene-expression signature as a predictor of survival in breast cancer. *New England Journal of Medicine*, 347/25, (2002), 1999–2009.

[8] J Hans Van Hateren and Arjen van der Schaaf, Independent component filters of natural images compared with simple cells in primary visual cortex. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 265/1394, (1998), 359–366.

[9] Mikael Vejdemo-Johansson, Florian T Pokorny, Primoz Skraba, and Danica Kragic, Cohomological learning of periodic motion. *Applicable algebra in engineering, communication and computing*, 26/1, (2015), 5–26.