NTNU
Norwegian University of
Science and Technology

Topology and Data

Birger Brekke

Master of Science in Physics and Mathematics
Submission date: June 2010
Supervisor: Nils A. Baas, MATH

Norwegian University of Science and Technology
Department of Mathematical Sciences

Problem Description

Learn to use the Plex and Mapper software, and describe the theory behind them. Plex is a software package for computing persistent homology of finite simplicial complexes. Mapper is a method for extracting simple descriptions, in the form of simplicial complexes, from high dimensional data sets. In addition, Plex and Mapper are to be applied on some point clouds generated from known spaces.

Assignment given: 25. January 2010
Supervisor: Nils A. Baas, MATH

Preface

This is my master's thesis, written the spring of 2010 at the Norwegian University of Science and Technology (NTNU). The purpose of this thesis has been to study how topology can be used to extract information from data sets. To be able to do this, one needs to acquire some knowledge from different topics in topology. Getting to know this topological machinery has been the main focus of this thesis. However, I have also spent some time on implementing both Mapper and persistent homology, which are the two main topics of this thesis, in Python.

Firstly, I would like to thank my supervisor professor Nils A. Baas. His advice and guidance throughout my study and the writing of this thesis was greatly appreciated. I would also like to thank Till Tantau for designing TikZ/PGF, Jörg Lehmann and Andre Wobst for designing PyX. TikZ is a graphical tool package for LaTeX, and PyX is a python package for creating PostScript figures. These tools has made it possible to draw all the diagrams and figures in this project.

Birger Brekke, June 15, 2010

Abstract

In the last years, there has been done research in using topology as a new tool for studying data sets, typically high dimensional data. These studies have brought new methods for qualitative analysis, simplification, and visualization of high dimensional data sets. One good example, where these methods are useful, is in the study of microarray data (DNA data). To be able to use these methods, one needs to acquire knowledge of different topics in topology. In this paper we introduce simplicial homology, persistent homology, Mapper, and some simplicial complex constructions.

Contents

1 Introduction

An important field of study in modern science is the process of extracting patterns from data. The reason for it is importance is partly because of the fact that the amount of data being produced by modern science and engineering is increasing at an unprecedented rate. For centuries the job of extracting patterns from data have been done manually, but the increasing volume of data calls for more automated approaches. One other important fact besides the vast amount of the data is the nature it comes in. The data is often given as very long vectors (high-dimensional), where only a few unknown coordinates turn out to be of importance to the question in mind. It is also more common that the data is a lot noisier and missing more data than in the past.

Unfortunately, our ability to analyse this data, both in terms of quantity and the nature of the data, is not keeping pace with the data being produced. In the last years, there has been done research in using topology and geometry as new tools to study data sets. Topology and geometry are old fields in mathematics, where geometry is the study of figures in a space of a given number of dimensions and of a given type. Topology is the mathematical study of the properties that are preserved through deformations, twisting, and stretching of objects. One can view geometry as the finest level of classification as it focuses on local properties of shapes. In this sense, geometry has a quantitative nature and can answer low level questions about a shape. But most of our questions have a qualitative feel and take a higher view of a shape. This prompts us to look at topological techniques that classify shapes according to the way they are connected globally - their connectivity. Topology in itself is often too coarse to be useful. For example, the topological invariant homology cannot distinguish between circles and ellipses, or even between circles and rectangles. On the other side, when we combine geometry and topology we get robust methods for classifying spaces. [1]

One good example, where topological methods work well, is in the study of DNA, where the data is collected from microarrays. A microarray is an instrument which can measure the expression level of thousands of different genes from a sample of cells. Although biomedical investigators have been quick to adopt this powerful new research tool, accurate analysis and interpretation of the data have provided unique challenges. Collected information from microarrays is noisy, high-dimensional, and may also be missing some data. When looking at the data, one may view it as a matrix, as in Table 1. In this matrix a row corresponds to a gene, a column corresponds to a sample, and the entries are gene expression levels in the different samples.

When we want to study the data using topology, the first thing we do is to represent the data as a finite set of points in Euclidean space with a distance function. This is what we call a point cloud. In the case of microarray data, it is both possible to either let each gene represent a point with the corresponding gene expressions from the samples as coordinates, or let each sample represent a point with the gene expressions from the sample as coordinates. In a typical case, we have 100 samples and 3000 different genes.

[1] For example: Instead of applying homology to a space, we apply it to a derived space with attached geometric content.

	Sample 1	$\cdots$	Sample m
Gene 1	a_{11}	$\cdots$	a_{1m}
$\vdots$	$\vdots$	$\ddots$	$\vdots$
Gene n	a_{n1}	$\cdots$	a_{nm}

Table 1: Microarray data represented as a matrix, where element a_{ij} is the expression of gene i in sample j.

Then the first point cloud will consists of 3000 points in $\mathbb{R}^{100}$ while the later case has 100 samples in $\mathbb{R}^{3000}$. When working with such data, one wishes to have lots of points in a low-dimensional space. Unfortunately, in our case it is the high-dimensional space, with 100 points in $\mathbb{R}^{3000}$, that turns out to be of importance. To make it even worse, there are no good distance functions available. A distance function, or a metric as it is commonly called, usually have an understandable physical meaning. In the case of DNA, there are many different ways of measuring distances, but there is no good choice of a metric that has a decent underlying understanding. In the world of biology, they mainly use BLAST[2] scores to measure similarity. It is calculated using some intuitively measures of similarity, but it is far from clear how much significance to attach to the actual distances, particularly at large scales. Fortunately for us, topology need not depend too much on the distance function. What is important is not the distance between two points, but if they are "close" or not.

After obtaining the point cloud, topology can be used to extract qualitative geometric information, which gives the data a signature (classification). One way of doing so is by assigning the point cloud to some topological space, and then calculating the homology of this space. There are many viable ways of connecting the point cloud to a space. One natural way of doing this is by letting the space be the union of the collection of balls, see Figure 1, where we have one ball with a fixed radius ϵ for each element in the point cloud. When we calculate the homology of this space, we get information about the connectivity of the space, which tends to be important information in the study of microarray data. In fact, the lowest level of the connectivity information is about the number of connected components, and is analogous to the information we get from clustering methods in statistics.

Clustering methods are very central and probably the most used tools in data analysis. They are good at finding clues of structures in the data, but have a lack of validity. There are many different clustering methods, but very little is known about the theoretical foundation of clustering methods. One other problem with these clustering methods is that they often demand some choice of parameters, such as the ϵ we used as the radius on our balls in Figure 1. Choices like these makes the methods less robust. It is possible to do the calculations multiple times by changing the parameters, but there will still be a lack of information of what has happened between the different calculations

[2]In bioinformatics, Basic Local Alignment Search Tool, or BLAST, is an algorithm for comparing primary biological sequence information.

("slices") and about the connection between them. In the world of topology, we have tools such as homology, which are well studied and understood. Functorial properties of homology makes it possible to get more complete pictures compared to just looking at each "slice". Moreover, as mentioned above, homology gives more information than clustering techniques. Informally, we can say that clustering measures 0-dimensional loops while homology measures all n-dimensional loopiness. In this sense homology looks deeper into the data, and gives an increased amount of usable information, even though not all of the information may be of importance.

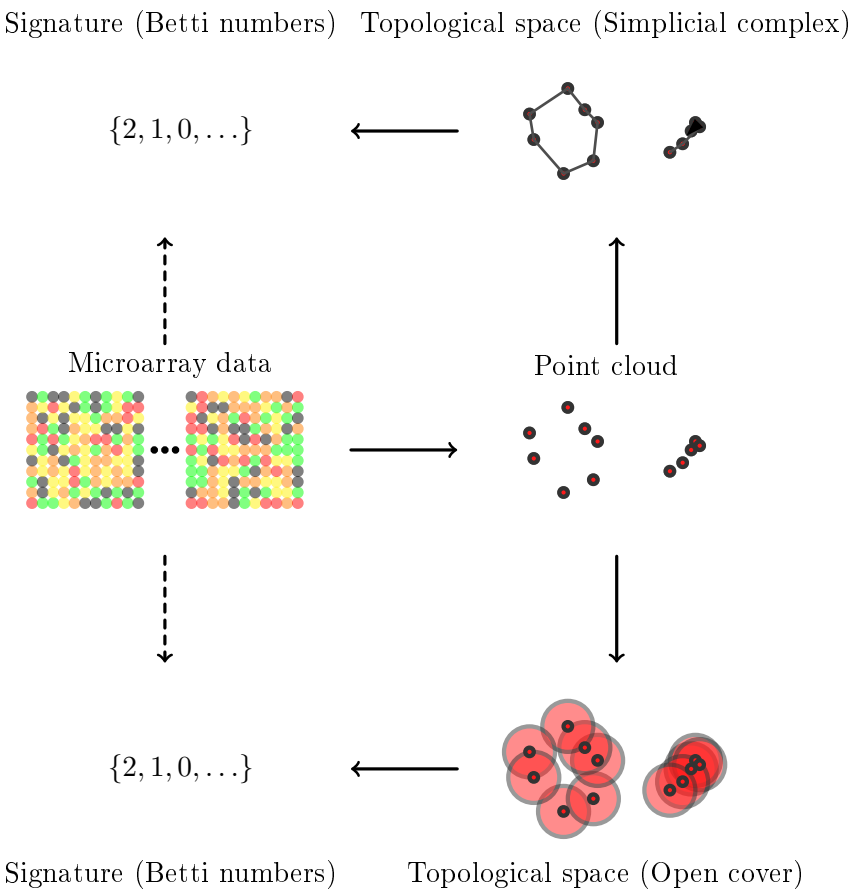


Figure 1: Illustrates how a data set, in this case microarray data, can be assigned a signature. There is mainly three steps in this process. Step 1: Data set → Point cloud. This first step consists of representing the data as some metric space, i.e. a space with a distance function. In the case of microarray data, it is both possible to either let each gene represent a point with the corresponding gene expressions from the samples as coordinates, or let each sample represent a point with the gene expressions from the sample as coordinates. There are also many possibilities of choosing a distance function. The straightforward way of doing this is to let the point cloud be a collection of points in Euclidean space with the usual metric. Step 2: Point cloud → Space. This is a difficult step. The data cloud is only a sample from some underlying space. It is important in this step to make a good guess of a space representing the underlying space. A good guess would be a space that is homotopic to the unknown underlying space. One natural way of doing this is by letting the space be the union of the collection of balls, where we have one ball with a fixed radius ϵ for each element in the point cloud. A more general and frequently used method is to build a simplicial complex as a model of the space. Step 3: Space → Signature. This last step consists of extracting topological information about key properties, such as the number of n-dimensional holes. The number of n-dimensional holes in the different dimensions, is called the Betti numbers and is a homotopy invariant. These Betti numbers can be computed with different types of homology theories. The most commonly used is simplicial homology, which can calculate the homology of simplicial complexes.

2 Simplicial homology

This introduction to simplicial homology follows definitions from planetmath[14] and the book "An Introduction to Intersection Homology Theory"[11] by Frances Kirwan and Jonathan Woolf.

2.1 Motivation

In this section we will establish simplicial homology. Simplicial homology is usually the choice when doing computerized computations, since it gives an easy way of computing homology. The idea behind simplicial homology is to use simple models, called simplicial complexes, of the given topological space to infer global information.

2.2 Homological algebra

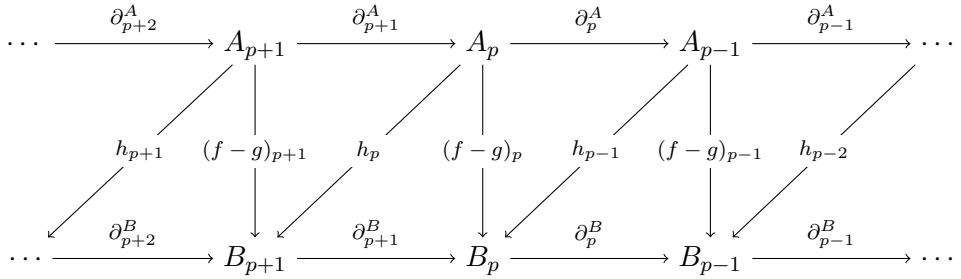
Before we can proceed with simplicial homology, we need to obtain some definitions from homological algebra.

Definition 1. Chain complex
Let $\{C_p\}_p$ be a sequence of abelian groups or modules, and let ∂, called the boundary operator, be a sequence of maps $\{\partial_p : C_p \longrightarrow C_{p-1}\}_p$ s.t. $\partial_{p-1} \circ \partial_p = 0 \; \forall p$. Then (C_*, ∂) is called a chain complex.

Definition 2. Homology of a chain complex
Let $\{C_*, \partial\}$ be a chain complex. Then the homology of this chain complex is given by

$$H_p = \frac{\ker \partial_p}{\operatorname{im} \partial_{p+1}} \; \forall p.$$

Definition 3. Chain homotopy
Let (A_*, ∂^A) and (B_*, ∂^B) be chain complexes, and let $f, g : A_* \longrightarrow B_*$ be chain maps. Then a chain homotopy h between f and g is a sequence of homomorphisms $\{h_p : A_p \longrightarrow B_{p+1}\}_p$ s.t. $(f-g)_p = h_{p-1}\partial_p^A + \partial_{p+1}^B h_p$.



Note. *If there exists a chain homotopy between two chains f and g, then f and g are said to be chain homotopic, which we will denote $f \simeq g$.*

Note. *If f and g are chain homotopic, then f and g induce the same map $[f] = [g]$ on homology groups.*

Definition 4. Chain equivalence
Let (A_*, ∂^A) and (B_*, ∂^B) be chain complexes, and let $f : A_* \longrightarrow B_*$ be a chain map. Furthermore, let 1_{A_*} be the identity map on A_*, and let 1_{B_*} be the identity map on B_*. Then f is called a chain equivalence if there $\exists$ a chain $g : B_* \longrightarrow A_*$ s.t. $f \circ g \simeq 1_{B_*}$ and $g \circ f \simeq 1_{A_*}$.

Note. *When there exists a chain equivalence between two chain complexes, they are said to be homotopic.*

2.3 Direct limit

We are also going to use the definition of the direct limit of a directed system.

Definition 5. Directed set
Let $(A, \leq)$ be a partially ordered set. If $\forall a, b \in A$ there $\exists x \in A$ s.t. $a \leq x$ and $b \leq x$, then $(A, \leq)$ is a directed set.

Definition 6. Direct system and direct family
Let $\mathcal{A} = \{A_i | i \in I\}$ be a family of algebraic systems of the same type. If there $\exists$ a family of homomorphisms $\{\phi_{ij} : A_i \longrightarrow A_j | i \leq j \in I\}$ s.t.

1. I is a directed set,
2. $\exists \phi_{ij} : A_i \longrightarrow A_j$,
3. $\phi_{ii} = 1_{A_i}$ on A_i, and
4. $\phi_{jk} \circ \phi_{ij} = \phi_{ik}$

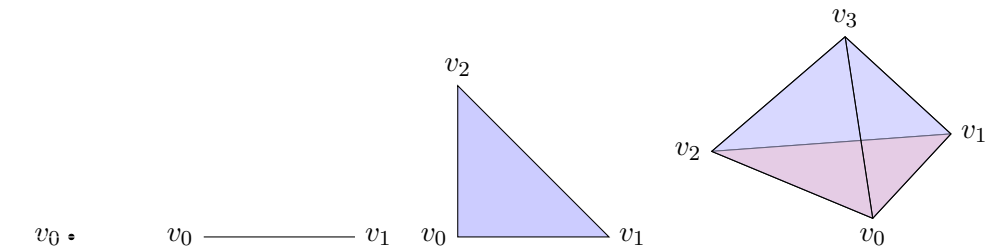
$\forall i \leq j \leq k$, then $\mathcal{A} = \{A_i\}_{i \in I}$ is a directed family, and (A_i, ϕ_{ij}) is a directed system.

Definition 7. Direct limit
Let (A_i, f_{ij}) be a direct system, let $\coprod_{i \in I} A_i$ be the disjoint union of $\{A_i\}_{i \in I}$, and let $x_i \in A_i$ and $x_j \in A_j$ be equivalent if there $\exists k \in I$ s.t. $f_{ik}(x_i) = f_{jk}(x_j)$. Then the direct limit of (A_i, f_{ij}), $\varinjlim_i A_i = \coprod_i A_i / \sim$.

2.4 Simplicial complex

Simplicial complexes use simplistic building blocks, called n-simplices, which are generalizations of triangles and tetrahedrons to an arbitrary dimension, see Figure 2.

Definition 8. n-simplex
Let $\{v_\alpha\}_{\alpha \in I} \subseteq \mathbb{R}^m$ be a set of n points, and let $\alpha_0 \in I$ be some point in I. Then the convex hull of $\{v_\alpha\}_{\alpha \in I}$ is an n-simplex if the elements of $\{v_\alpha - v_{\alpha_0}\}_{\alpha \neq \alpha_0 \in I}$ are linearly independent.



(a) 0-simplex (b) 1-simplex (c) 2-simplex (d) 3-simplex

Figure 2: Example of ordered simplices.

Notation. *Let* $\mathrm{vert}(\sigma)$ *be the set of vertices which construct* σ*.*

Notation. *When* σ *is an* n*-simplex, with vertices* $\mathrm{vert}(\sigma) = \{v_\alpha\}_{\alpha \in I}$*, we will denote* σ *as* $<v_\alpha>_{\alpha \in I}$*.*

Definition 9. Face of an n-simplex
Let σ be an n-simplex with vertices $\{v_\alpha\}_{\alpha \in I}$, and let σ' be an $(n-1)$-simplex with vertices $\{v_{\alpha'}\}_{\alpha' \in I'}$. Then σ' is a face of σ if $\{v_{\alpha'}\}_{\alpha' \in I'} \subseteq \{v_\alpha\}_{\alpha \in I}$.

With these terms we can define a simplicial complex as follows.

Definition 10. Simplicial complex in $\mathbb{R}^n$
Let N be a set of simplices. Then N is a simplicial complex in $\mathbb{R}^n$ if the following statements hold:

1. If $\sigma \in N$, then $\sigma' \in N \; \forall \sigma'$ face of σ;
2. if $\sigma, \sigma' \in N$ and $\sigma \cap \sigma' \neq \emptyset$, then $\mathrm{vert}(\sigma \cap \sigma') = \mathrm{vert}(\sigma) \cap \mathrm{vert}(\sigma')$;
3. if $\sigma \in N$ and $x \in \sigma$, then there exists a neighborhood U of x s.t. $U \cap \sigma' \neq \emptyset$ for only finitely many simplices $\sigma' \in N$.

2.5 Ordered simplicial complex

The boundary of a simplicial complex is important for constructing simplicial homology. Before we can establish our boundary map, we need to have an ordering on our simplicial complex. When given a simplicial complex, we can always make it into an ordered simplicial complex by numbering the vertices. Keeping track of the ordering under operations like boundary operations is a bit messy, and is therefore often hidden under the carpet since it is trivial. Even though it is not nice, we will keep track of the orientation until we have defined the boundary map on a simplicial complex.

An n-simplex with an ordering can be defined as follows.

Definition 11. Ordered n-simplex
Let σ be an n-simplex with $\{v_\alpha\}_{\alpha\in I}$ as vertices, and let $h : [0,\dots,n] \longrightarrow I$ be a bijection. Then the pair (σ, h) is an ordered n-simplex.

Notation. *Let σ be an n-simplex with $\{v_i\}_{i\in[0,\dots,n]}$ as vertices. Then we will denote the ordered n-simplex $\left(\sigma, 1_{[0,\dots,n]}\right) = [v_0,\dots,v_n]$.*

With this definition of an ordered n-simplex, an ordered simplicial complex becomes as follows.

Definition 12. Ordered simplicial complex
Let $N = \{\sigma_\alpha\}_{\alpha\in\mathcal{N}}$ be a simplicial complex, and let h_* be a sequence of functions s.t. $(\sigma_\alpha, h_\alpha)$ is an ordered simplex $\forall\sigma_\alpha \in N$. Then the pair (N, h_*) is an ordered simplicial complex if $\sigma_\alpha, \sigma_\beta \in N$, where $\mathrm{vert}(\sigma_\alpha) \cap \mathrm{vert}(\sigma_\beta) \neq \emptyset$, implies that the equation

$$h_\alpha^{-1}(x) < h_\alpha^{-1}(y) \Leftrightarrow h_\beta^{-1}(x) < h_\beta^{-1}(y)$$

holds $\forall x, y \in \mathrm{vert}(\sigma_\alpha) \cap \mathrm{vert}(\sigma_\beta)$.

Notation. *Let (N, h_*) be an ordered simplicial complex in $\mathbb{R}^n$. Then $(N, h_*)^i$ is the set of ordered i-simplices, i.e. $(N, h_*)^i = \{(\sigma_\alpha, h_\alpha) | \sigma_\alpha$ is an i-simplex$\}$.*

Now that we have an ordered simplicial complex, we can construct a boundary map on the n-simplices by using partial boundary maps.

Definition 13. j-th partial boundary of ordered n-simplices
Let (σ, h) be an ordered n-simplex. Then the j-th partial boundary of (σ, h) is given by $\partial_j : (\sigma, h) \longmapsto (\partial_j\sigma, \partial_j h)$, where

$$\partial_j : \sigma = <v_{h(0)},\dots,v_{h(n)}> \longmapsto <v_{h(0)},\dots,\hat{v}_{h(j)},\dots,v_{h(n)}>$$

and $\partial_j h$ is given by

$$(\partial_j h)(k) = \begin{cases} h(k) & \text{if } k < j \\ h(k+1) & \text{if } k \geq j \end{cases}.$$

These partial boundary maps give us a boundary map on the n-simplices.

Definition 14. Boundary of ordered n-simplices
Let (σ, h) be an ordered n-simplex. Then the boundary of (σ, h) is given by

$$\partial(\sigma, h) = \sum_{j=0}^{n} (-1)^j (\partial_j\sigma, \partial_j h).$$

Note. $\partial^2 = 0$.

2.6 Simplicial homology of a simplicial complex

We will now build our chain complex that gives us simplicial homology. To make this more notation friendly, we will use the following notations.

Notation. *Let (N, h_*) be an ordered simplicial complex, and let (σ, h) be an ordered simplex. Then the following notations will be used:*

- $N = (N, h_*)$;
- $N^i = (N, h_*)^i$;
- $\sigma = (\sigma, h)$;
- *"simplex" = "ordered simplex";*
- *"simplicial complex" = "ordered simplicial complex".*

The chains in our chain complex will be formal finite sums of i-simplices with coefficients in a field.

Definition 15. i-chain of a simplicial complex
Let F be a field, and let N be a simplicial complex. Then an i-chain of N is given by

$$\xi = \sum_{\sigma \in N^i}^{\text{finite}} \xi_\sigma \sigma,$$

where $\xi_\sigma \in F \; \forall \sigma \in N^i$.

Notation. *Let $C_i(N)$ be the set of all i-chains in N, i.e. $C_i(N) = \{\xi \; i\text{-chain in } N\}$.*

Now that we have the groups in our chain complex, we only need to construct a boundary operator to get a complete chain complex. We need our boundary operator ∂ to turn an i-chain into an $(i-1)$-chain, while satisfying the condition $\partial^2 = 0$. By using the boundary map we defined on our simplices, we can make an i-chain into a $(i-1)$-chain by using the boundary map on each of the i-simplices in the formal sum.

Definition 16. Boundary operator on i-chains
Let $\xi = \sum_{\sigma \in N^n}^{\text{finite}} \xi_\sigma \sigma$ be an i-chain, where $\xi_\sigma \in F \; \forall \sigma \in N^i$. Then the boundary operator on ξ is given by the linear map

$$\partial \xi = \sum_{\sigma \in N^i}^{\text{finite}} \xi_\sigma \partial \sigma.$$

This boundary operator satisfies $\partial^2 = 0$. Hence, we have a chain complex $(C_*(N), \partial)$. By taking the homology of this chain complex, we get the simplicial homology of a simplicial complex.

Definition 17. Homology of a simplicial complex
Let N be a simplicial complex. Then the simplicial homology of N is given by

$$H_i(N) = \frac{\ker \partial : C_i(N) \longrightarrow C_{i-1}(N)}{\operatorname{im} \partial : C_{i+1}(N) \longrightarrow C_i(N)}.$$

2.7 Simplicial homology of a topological space

We have now defined the simplicial homology of a simplicial complex. What we are interested in is to make a homology theory on (triangulable) topological spaces by using the simplicial complexes. We wish to describe the topological space with a collection of simplicial complexes, and then use these to calculate the homology. In fact, we are only going to do simplicial homology on triangulable topological spaces, and use the family of all triangulations as our collection of simplicial complexes.

By a triangulation we mean.

Definition 18. Support of a simplicial complex in $\mathbb{R}^n$
Let N be a simplicial complex in $\mathbb{R}^n$. Then the support of N is given by

$$|N| = \bigcup_{\sigma \in N} \sigma.$$

Definition 19. Triangulation of a topological space
Let X be a topological space, and let N be a simplicial complex. Then a triangulation of X is a homeomorphism $T : |N| \xrightarrow{\cong} X$.

Note. *We will call N a triangulation of X when there exists a triangulation from $|N|$ to X.*

We will also be using the following notation.

Notation. *Let $T : |N| \longrightarrow X$ be a triangulation of X. Then we will write $H_i^T(X) = H_i(N)$ and $C_i^T(X) = C_i(N)$.*

There may exist several triangulations of a topological space. To make the simplicial homology independent of a chosen triangulation, we will make a direct system by using refinement of triangulations.

Definition 20. Refinement of a triangulation
Let N and N' be two simplicial complexes, and let $T : |N| \longrightarrow X$, $T' : |N'| \longrightarrow X$ be two triangulations of X. Then T is a refinement of T' if $\forall \sigma \in N$ there $\exists \sigma' \in N'$ s.t. $T(\sigma) \subseteq T'(\sigma')$.

These triangulations will then induce chain maps on our chain complexes. Moreover, the collection of chain complexes together with the induced maps will form a direct system.

Proposition 2.1. *Let N and N' be two simplicial complexes, and let $T : |N| \longrightarrow X$, $T' : |N'| \longrightarrow X$ be two triangulations of X s.t. T is a refinement of T'. Then there $\exists$ a natural map $\phi_{T',T} : C_i^{T'}(X) \longrightarrow C_i^T(X)$ compatible with the boundary maps s.t. if $\sigma' \in N_i'$, then*

$$\phi_{T',T} : \sigma' \longmapsto \sum_{\sigma \in N_i, T(\sigma) \subseteq T'(\sigma')} \pm \sigma,$$

where the signs depend on the orientation of σ compared to σ'.

Note. *The index set $I = \{\alpha | T_\alpha$ triangulation of $X\}$ is a directed set by $\alpha \leq \beta$ if T_β is a refinement of T_α.*

Note. *Let $\{T_\alpha\}_\alpha$ be the family of all triangulations of X, and let*

$$\left\{\phi_{\alpha,\beta} : C_i^{T_\alpha}(X) \longrightarrow C_i^{T_\beta}(X)\right\}_{\alpha,\beta}$$

be the corresponding family of natural maps. Then the pair $(C_i^{T_\alpha}(X), \phi_{\alpha,\beta})$ form a direct system.

By taking the direct limit we get a chain complex independent of any choice of triangulation.

Definition 21. Space of piecewise linear i-chains
Let $\{T_\alpha\}$ be the family of all triangulations of X, and let $(T_\alpha, \phi_{\alpha,\beta})$ form a direct system. Then the space of all piecewise linear i-chains is given by $C_i(X) = \varinjlim_\alpha C_i^{T_\alpha}(X)$.

Note. *Note that the boundary maps $\partial_\alpha : C_i^{T_\alpha}(X) \longrightarrow C_{i-1}^{T_\alpha}(X)$ induce boundary maps $\partial : C_i(X) \longrightarrow C_{i-1}(X)$ s.t. $\partial^2 = 0$.*

The homology of this chain complex gives us the simplicial homology of a triangulable space.

Definition 22. Simplicial homology
Let X be a triangulable space. Then the simplicial homology of X is given by

$$H_i^{\text{simp}}(X) = \frac{\ker \partial : C_i(X) \longrightarrow C_{i-1}(X)}{\text{im}\, \partial : C_{i+1}(X) \longrightarrow C_i(X)}.$$

We will not show this, but simplicial homology is a homology theory. An important property of simplicial homology is that we need only use one simplicial complex, which gives a triangulation of our topological space, to calculate the simplicial homology. Moreover, it corresponds with the singular homology.

Theorem 2.2. *Let N be a simplicial complex, and let $T : |N| \longrightarrow X$ be a triangulation of a topological space X. Then $H_*^{sing}(X) \cong H_*^{simp}(X) = H_*^{simp}(N)$.*

Unfortunately, not every topological space have a triangulation, but some does. In fact, every compact manifold of dimension 3 or less has a triangulation.

2.8 Simplicial homology on data sets

When we are studying data sets, we do not know the underlying topological space X; we are usually only given some sampled data from X. What we want to do is to use the sampled data to infer information about the geometric and topological structure of X. Information, such as the simplicial homology of X, is then very useful, and it

would be of great help to be able to calculate it, or get an approximation. When we are given a data set, we need to construct some simplicial complex that approximates a triangulation of X. Since simplicial homology is a homology theory, it is sufficient that our simplicial complex is a triangulation of a topological space Y that is homotopic to X. This increases the possibility of our constructed simplicial complexes to be good approximations. A diagram, which illustrates the process of approximating the simplicial homology of a topological space X by using only some sampled data, is given in Figure 3. In the construction of simplicial homology, we used a family of triangulations to

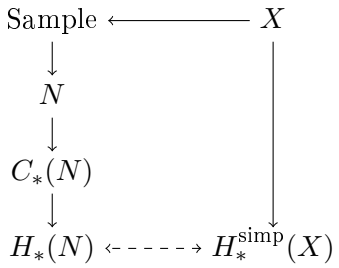


Figure 3: This diagram illustrates the process of approximating the simplicial homology of a topological space X by using some sampled data. The step Sample $\to N$ consists of constructing some simplicial complex from the sampled data.

determine the homology. When we are dealing with sampled data, a natural question is if we can construct and use multiple simplicial complexes to get more insight. This idea is illustrated in Figure 4. In Section 4 we will talk about persistent homology which

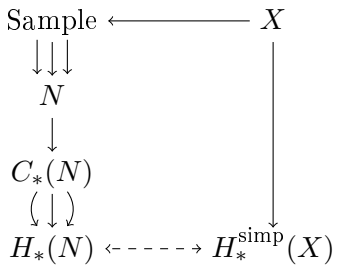


Figure 4: This diagram illustrates the process of approximating the simplicial homology of a topological space X by using some sampled data. The step from Sample to N consists of creating a collection of simplicial complexes from the sampled data, and the step from $C_*(N)$ to $H_*(N)$ consists of assembling the results by some manner.

follows this idea, but first we will talk about some different methods for constructing simplicial complexes.

3 Simplicial complexes

In this section we will introduce some methods for constructing simplicial complexes (abstract simplicial complexes). Resources have been gathered from [5], [3], [12], [7] and [8].

3.1 Motivation

When we are given some sampled data from a topological space X, we want to use the data to construct a simplicial complex, which is a triangulation or close to a triangulation of the underlying space X, or a space homotopic to X. If we can find such a simplicial complex, then it will give us the simplicial homology of X, which we are interested in. There has been done a lot of study on constructing simplicial complexes that are homeomorphic to a topological space X. One well known example is the Delaunay complex. The Delaunay complex method makes a relative small simplicial complex, and in many cases it becomes homeomorphic to the given space X when the resolution is fine enough. In our case we only need a simplicial complex homotopy equivalent to X. This gives us more freedom since a homeomorphic equivalence is more strict than a homotopy equivalence; moreover, homeomorphic equivalence implies homotopy equivalence. On the other hand, we are not given the topological space X that we want to approximate. We are only given a (discrete) finite sample. When working with a discrete sample, constructions like the Delaunay complex becomes useless. In this section we will introduce some methods for building simplicial complexes that makes sense when dealing with discrete samples from a topological space.

One of the most theoretically backed up constructions for creating simplicial complexes is the Čech complex. The Čech complex method constructs abstract simplicial complexes by taking the nerve of the covers. Before we can define the nerve of a cover, we need to define an abstract simplicial complex.

3.2 Abstract simplicial complex

Note. *Let S be a set, then the power set of S is given by $\mathcal{P}(S) = \{A | A \subseteq S\}$.*

Definition 23. Abstract simplicial complex
Let V be a set, and let Σ be a family of subsets of V, i.e. $\Sigma \subseteq \mathcal{P}(V)$. Then the pair (V, Σ) is an abstract simplicial complex iff

1. $\sigma \neq \emptyset$,
2. σ is a finite set, and
3. $\tau \in \Sigma \; \forall \tau \subseteq \sigma$

for all $\sigma \in \Sigma$. Moreover, (V, Σ) is called a finite abstract simplicial complex if $|V|$ is finite.

There is a close relation between abstract simplicial complexes, for which the nerve is an example, and simplicial complexes. An abstract simplicial complex can in many cases be interpreted as a simplicial complex; one such case is when it is a finite abstract simplicial complex. To make the relation more clear, we will just state some definitions and show a few results.

Definition 24. Vertex scheme
Let N be a simplicial complex, let $V = \cup_{\sigma \in N} \text{vert}(\sigma)$, and let $\Sigma = \{\text{vert}(\sigma)\}_{\sigma \in N}$. Then (V, Σ) is an abstract simplicial complex. Moreover, the set Σ is the vertex scheme of N.

Note. *The dimension of a simplicial complex N is given by* $\sup\{|\text{vert}(\sigma)| \mid \sigma \in N\}$*, i.e. the order of the largest simplex in N.*

Note. *The dimension of an abstract simplicial complex (V, Σ) is given by* $\sup\{|\sigma|\}_{\sigma \in \Sigma}$*, i.e. the order of the largest simplex in (V, Σ).*

Definition 25. Geometric realization
Let (V, Σ) be an abstract simplicial complex, and let K be a simplicial complex. Then K is a geometric realization of (V, Σ) if Σ is isomorphic to the vertex scheme of K. Moreover, the vertex scheme K is unique up to a linear isomorphism.

Proposition 3.1. *Let (V, Σ) be a finite abstract simplicial complex. Pick a bijection $\phi : V \longrightarrow [1, \ldots, N]$ to give V a total order, and let $c(\sigma)$ be the convex hull of $\{e_{\phi(v)} \mid v \in \sigma\}$. Then the space given by $|(V, \Sigma)| = \bigcup_{\sigma \in \Sigma} c(\sigma)$ is a geometric realization of (V, Σ).*

Note. *Let ϕ_1 and ϕ_2 be two bijections giving V an order, and let $|(V, \Sigma)|_{\phi_1}$ and $|(V, \Sigma)|_{\phi_2}$ be the corresponding geometric realizations. Then $|(V, \Sigma)|_{\phi_1}$ and $|(V, \Sigma)|_{\phi_2}$ are homotopy equivalent.*

Proposition 3.2. *Every finite abstract simplicial complex has a geometric realization.*

Theorem 3.3. *Let (V, Σ) be a finite abstract simplicial complex of dimension k. Then (V, Σ) has a geometric realization in $\mathbb{R}^{2k+1}$.*

Since there is no difference between simplicial complexes and abstract simplicial complexes when they are finite, we will feel free to mix their names and notation. We are also only going to consider the case where the data is embedded in a metric space.

3.3 The nerve of a cover

Now that we have defined an abstract simplicial complex, we can define the nerve of a cover.

Definition 26. Nerve of a cover $\mathcal{N}(\mathcal{U})$
Let X be a topological space, and let $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ be cover of X. Then the nerve of $\mathcal{U}$ is the abstract simplicial complex $\mathcal{N}(\mathcal{U}) = (A, \Sigma)$, where $\sigma = \{\alpha_0, \ldots, \alpha_n\} \in \Sigma$ iff $U_{\alpha_0} \cap \cdots \cap U_{\alpha_n} \neq \emptyset$.

Not all covers are fitted to create simplicial complexes by applying the nerve. However, a family of suitable covers is the family of good covers.

Definition 27. Good cover
Let $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ be an open cover of some topological space X. Then $\mathcal{U}$ is a good cover of X if $\mathcal{U}$ is locally finite and $\bigcap_{\beta \in B} U_\beta$ is empty or contractible $\forall\ B \subseteq A$.

Note. *A good cover is also called a Čech cover.*

These covers have the essential property that under certain conditions the geometric realization of the nerve is homotopy equivalent to the space itself.

Lemma 3.4. *The nerve lemma*
Let $\mathcal{U}$ be a good cover of some paracompact topological space X. Then the geometric realization of $\mathcal{N}(\mathcal{U})$ is homotopy equivalent to X.

The nerve lemma (Lemma 3.4) gives the following corollary.

Corollary. *Let $\mathcal{U}$ be a good cover of some paracompact topological space X. Then the abstract simplicial complex $\mathcal{N}(\mathcal{U})$ has a geometric realization.*

We will not define paracompactness, but state the fact that topological spaces in Euclidean space are paracompact; moreover, every metric space is paracompact. The nerve lemma (Lemma 3.4) and the following theorem (Theorem 3.5) makes the nerve a useful tool.

Theorem 3.5. *Let M be a compact Riemannian manifold. Then there $\exists$ a positive $e \in \mathbb{R}$ s.t. $\mathcal{N}(B_\epsilon(M))$ is homotopic to M $\forall$ positive $\epsilon \leq e$. Moreover, $\forall$ positive $\epsilon \leq e$ there $\exists$ a finite subset V of M s.t. $\mathcal{N}(B_\epsilon(V))$ is homotopic to M.*

3.4 Čech - and Vietoris-Rips complex

As we mentioned, the nerve is an important tool; especially when used on covers consisting of epsilon balls. We will therefor define the Čech complex as follows.

Definition 28. Čech complex - $\check{C}(X, \epsilon)$
Let (X, d) be a metric space, let $\epsilon \in \mathbb{R}$ be positive, and let $\mathcal{U} = \{B(x, \epsilon)\}_{x \in X}$. Then the Čech complex with parameter ϵ of X is the nerve of $\mathcal{U}$, and will be denoted by $\check{C}(X, \epsilon)$.

Since we are working in a metric space, the nerve lemma (Lemma 3.4) gives us that the Čech complex is homotopic to the union of the balls. Unfortunately, the Čech complex is inefficient when doing calculations. There are mainly three reasons for this.

1. It is cumbersome to check if intersections are empty.
2. A lot of storage space is needed.
3. It produces large simplicial complexes with high dimensional simplices.

A solution to the first and second problem is the Vietoris-Rips complex.

Definition 29. Vietoris-Rips complex - VR(X, ϵ)
Let (X, d) be a metric space, and let $\epsilon \in \mathbb{R}$ be positive. Then the Vietoris-Rips complex VR(X, ϵ) of X attached to the parameter ϵ is the abstract simplicial complex given by (X, Σ), where $\{x_0, \ldots, x_k\} \in \Sigma$ iff $d(x_i, x_j) \leq \epsilon \; \forall \; i, j$ s.t. $0 \leq i, j \leq k$.

A Vietoris-Rips complex is somewhat less detailed, but it does only need the distances between each pair of data points to characterize the complex. However, it is more difficult to get an understanding of the homotopy type of a Vietoris-Rips complex. On the other hand, the Vietoris-Rips complex will give the same results as Čech complex when using persistent homology, which we are going to use anyway. The key reason for why they give the same persistent homology is the following property.

Proposition 3.6. *Let (X, d) be a metric space, and let $\epsilon \in \mathbb{R}$ be positive. Then*

$$\check{C}(X, \epsilon) \leq \text{VR(X, } 2\epsilon) \leq \check{\text{C}}(\text{X}, 2\epsilon).$$

How the Vietoris-Rips complex differs from the Čech complex is illustrated in Figure 5. The Vietoris-Rips complex is computationally more friendly than the Čech complex,

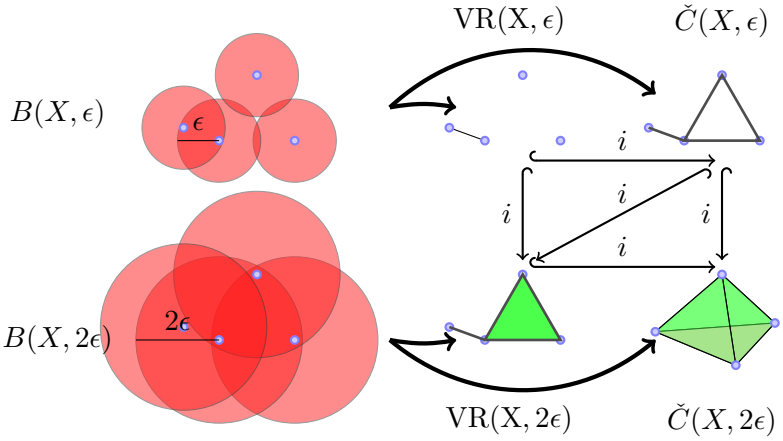


Figure 5: Illustrates the similarities between the Čech complex and the Vietoris-Rips complex.

but it is still inefficient. This is mostly because of the large vertex set we get with the Čech and Vietoris-Rips complexes.

3.5 Voronoi diagram

A solution to the large vertex set problem, which has been used to study subspaces of Euclidean space, is the Voronoi diagram (Voronoi decomposition). One way of reducing the vertex set is to only use a subset of the data as the vertex set. We call these points for landmark points.

Note. *A landmark point set of X is simply a subset of X.*

Notation. *The metric space X with $\mathcal{L}$ as the landmark point set will be denoted by $(X, \mathcal{L})$.*

One common known way of dividing a space into cells, with each cell corresponding to a landmark point, is the Voronoi diagram (Voronoi decomposition).

Definition 30. Voronoi cell
Let (X, d) be a metric space, and let $\mathcal{L}$ be a subset of X. Then the Voronoi cells $\{V_\lambda\}_{\lambda\in\mathcal{L}}$ of X with $\mathcal{L}$ as the landmark point set are given by $V_\lambda = \{x \in X | d(x, \lambda) \leq d(x, \lambda')\ \forall \lambda' \in \mathcal{L}\}$.

Definition 31. Voronoi diagram
Let (X, d) be a metric space, and let $\mathcal{L}$ be a subset of X. Then the Voronoi diagram of $(X, \mathcal{L})$ is the cover $\mathcal{U}$ of X given by $\mathcal{U} = \{V_\lambda\}_{\lambda\in\mathcal{L}}$, where $\{V_\lambda\}_{\lambda\in\mathcal{L}}$ are the Voronoi cells of $(X, \mathcal{L})$.

This gives a cover of the space. By taking the nerve of a Voronoi diagram, we get what is called the Delaunay complex.

Definition 32. Delaunay complex
Let (X, d) be a metric space, let $\mathcal{L}$ be a subset of X, and let $\mathcal{U}$ be the Voronoi diagram of $(X, \mathcal{L})$. Then the Delaunay complex Del(X, $\mathcal{L}$) of $(X, \mathcal{L})$ is the nerve of $\mathcal{U}$.

The complex Del(X, $\mathcal{L}$) carries a great deal of information about the topology of X and may even be homeomorphic to X if $\mathcal{L}$ is sampled sufficiently fine. Unfortunately, it is not very useful in the case of finite data sets unless we have an integer valued metric or something similar. In the case of finite sets data sets in Euclidean space with the usual metric, the Delaunay complex will most likely only consist of the 0-simplices corresponding to the landmark points of X. There will only be a 1-simplex if there exists a point s.t. the distances to the two closest landmark points are equally long. Using landmark points is a very efficient way of reducing the vertex set. However, the Delaunay complex is too strict for our use. An example of a Voronoi diagram of some sampled data, is given in Example 3.7.

Example 3.7. *Let $X = [0, 1]^2 \subseteq \mathbb{R}^2$, let $S \subseteq X$ be a sample consisting of 10000 randomly selected points, and let $\mathcal{L} \subseteq S$ be a set consisting of 20 landmark points. Then the Voronoi diagram of $(S, \mathcal{L})$ is as shown in Figure 6.*

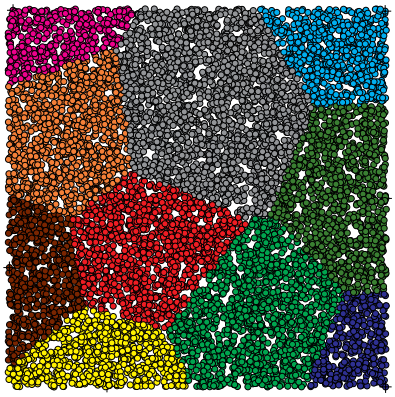


Figure 6: Refer to Example 3.7 for details. Voronoi diagram of some sampled data. Each landmark point is marked by a cross.

3.6 Witnesses

We want to add some slack to the conditions of the Delaunay complex such that we get a more interesting complex. There are multiple ways of doing this. To make the next complexes seem more similar, and to shorten the notation we introduce what we call a witness for a p-simplex.

We allow $\mathcal{L}$ to be infinite, even though $\mathcal{L}$ will be finite when doing computations. When $\mathcal{L}$ is finite, these abstract simplicial complexes will become simplicial complexes. The motivation for allowing $\mathcal{L}$ to be infinite is to be able to state the (strong) version of the weak witness theorem (Theorem 3.8), where $\mathcal{L}$ may be infinite. Moreover, allowing $\mathcal{L}$ to be infinite makes it possible to investigate and compare with cases where $\mathcal{L}$ is infinite.

Definition 33. Strong witness for an abstract p-simplex
Let (X, d) be a metric space, let $\mathcal{L}$ be a subset of X, and let $\sigma = \{l_0, \dots, l_p\} \subseteq \mathcal{L}$ be an abstract p-simplex. Then $x \in X$ is a strong witness for σ iff $d(l_i, x) \leq d(l', x)$ for $i = 0, \dots, p$ and $\forall\, l' \in \mathcal{L}$.

Definition 34. Weak witness for an abstract p-simplex
Let (X, d) be a metric space, let $\mathcal{L}$ be a subset of X, and let $\sigma = \{l_0, \dots, l_p\} \subseteq \mathcal{L}$ be an abstract p-simplex. Then $x \in X$ is a weak witness for σ iff $d(l_i, x) \leq d(l', x)$ for $i = 0, \dots, p$ and $\forall l' \in \mathcal{L} - \{l_0, \dots, l_p\}$.

Moreover, a strong witness can also be written in terms of weak witnesses.

Definition 35. Strong witness for an abstract p-simplex (alternative)
Let (X, d) be a metric space, let $\mathcal{L}$ be a subset of X, and let $\sigma = \{l_0, \dots, l_p\} \subseteq \mathcal{L}$ be an abstract p-simplex. Then $x \in X$ is a strong witness for σ iff x is a weak witness for σ and $d(x, l_0) = d(x, l_1) = \cdots = d(x, l_p)$.

Definition 36. ϵ strong witness for an abstract p-simplex
Let (X, d) be a metric space, let $\mathcal{L}$ be a subset of X, $\epsilon \in \mathbb{R}$ be positive, and let $\sigma = \{l_0, \ldots, l_p\} \subseteq \mathcal{L}$ be an abstract p-simplex. Then $x \in X$ is a ϵ strong witness for σ iff $d(l_i, x) \leq d(l', x) + \epsilon \; \forall \; i = 0, \ldots, p$ and $\forall \; l' \in \mathcal{L}$.

Definition 37. ϵ weak witness for an abstract p-simplex
Let (X, d) be a metric space, let $\mathcal{L}$ be a subset of X, $\epsilon \in \mathbb{R}$ be positive, and let $\sigma = \{l_0, \ldots, l_p\} \subseteq \mathcal{L}$ be an abstract p-simplex. Then $x \in X$ is a ϵ weak witness for σ iff $d(l_i, x) \leq d(l', x) + \epsilon \; \forall \; i = 0, \ldots, p$ and $\forall l' \in \mathcal{L} - \{l_0, \ldots, l_p\}$.

3.7 Witness complexes

We will now use these witness definitions to construct some complexes. As mentioned, the Delaunay complex has some very nice properties as a model of a topological space X, but it is insufficient when we are only given a discrete set of X. It is therefore desirable to design some new variations of the Delaunay complex, which approximates Del(X, $\mathcal{L}$) and makes sense when dealing with discrete data sets $Z \subseteq X$.

Let us first reface the Delaunay complex with the use of witnesses.

Definition 38. Strong witness complex (Delaunay complex) - $W^s(X, \mathcal{L})$
Let (X, d) be a metric space, and let $\mathcal{L}$ be a subset of X. Then the strong witness complex $W^s(X, \mathcal{L})$ is the abstract simplicial complex $(\mathcal{L}, \Sigma)$, where $\sigma = \{l_0, \ldots, l_p\} \in \Sigma$ iff there $\exists$ a strong witness for σ.

There are two well studied methods for slacking the conditions of the strong witness complex (Delaunay complex).

1. By using a tolerance parameter.
2. By using weak witnesses.

These two methods can also be combined, which results in what we call an ϵ weak witness complex. The first method by itself gives us what we call an ϵ strong witness complex.

Definition 39. ϵ strong witness complex - $W^s(X, \mathcal{L}, \epsilon)$
Let (X, d) be a metric space, $\mathcal{L}$ be a subset of X, and let $\epsilon \in \mathbb{R}$ be positive. Then the ϵ strong witness complex $W^s(X, \mathcal{L}, \epsilon)$ is the abstract simplicial complex $(\mathcal{L}, \Sigma)$, where $\sigma = \{l_0, \ldots, l_p\} \in \Sigma$ iff there $\exists$ an ϵ strong witness for σ.

Definition 40. Weak witness complex - $W^w(X, \mathcal{L})$
Let (X, d) be a metric space, and let $\mathcal{L}$ be a subset of X. Then the weak witness complex $W^w(X, \mathcal{L})$ is the abstract simplicial complex $(\mathcal{L}, \Sigma)$, where $\sigma = \{l_0, \ldots, l_p\} \in \Sigma$ iff there $\exists$ a weak witness for σ and a weak witness x_τ for each subset τ of σ.

The weak witness complex is also known as the strict witness complex and as the Martinetz-Schulten complex $MS_\infty(X, \mathcal{L})$. This second method, which gives the weak

witness complex, may seem more unmotivated than the first, which gave us the ϵ strong witness complex. Even though it may seem unmotivated, it has a close connection to the strong witness complex (Delaunay complex), see [7]. One of the results from Vin de Silva's work[7] is the following theorem.

Theorem 3.8. *Weak witness theorem*
Let $\mathcal{L}$ be a subset of $\mathbb{R}^n$, and let $\sigma = \{l_0, \ldots, l_p\}$ be a p-simplex with vertices in $\mathcal{L}$. Then σ has a strong witness iff there $\exists$ a weak witness $x_\tau \in \mathbb{R}^n$ for each subset τ of σ.

Let $X = \mathbb{R}^n$, and let $\mathcal{L}$ be a subset of X, then this theorem gives that $W^w(X, \mathcal{L}) = W^s(X, \mathcal{L})$. This does also hold for other spaces than $\mathbb{R}^n$, but it does not hold in general. Even though it does not hold for all spaces, it gives us a connection between the strong witness complex (Delaunay complex) and the weak witness complex. When we combine the methods of using a tolerance parameter and weak witnesses, we get the ϵ weak witness complex.

Definition 41. ϵ weak witness complex - $W^w(X, \mathcal{L}, \epsilon)$
Let (X, d) be a metric space, let $\mathcal{L}$ be a subset of X, and let $\epsilon \in \mathbb{R}$ be positive. Then the ϵ weak witness complex $W^w(X, \mathcal{L}, \epsilon)$ is the abstract simplicial complex $(\mathcal{L}, \Sigma)$, where $\sigma = \{l_0, \ldots, l_p\} \in \Sigma$ iff there $\exists$ an ϵ weak witness for σ and an ϵ weak witness x_τ for each subset τ of σ.

There is also a theorem connecting the ϵ weak witness complex with the ϵ strong witness complex, see [7]. Figure 9 illustrates the weak witness complex construction and the ϵ weak witness complex construction. Moreover, we can make "Vietoris-Rips" versions as we did with the Čech complex.

Definition 42. $W^s_{\mathrm{VR}}(X, \mathcal{L}, \epsilon)$
Let (X, d) be a metric space, let $\mathcal{L}$ be a subset of X, and let $\epsilon \in \mathbb{R}$ be positive. Then the complex $W^s_{\mathrm{VR}}(X, \mathcal{L}, \epsilon)$ is the abstract simplicial complex $(\mathcal{L}, \Sigma)$, where $\sigma = \{l_0, \ldots, l_p\} \in \Sigma$ iff there $\exists$ an ϵ strong witness $x_{i,j}$ for each 1-subsimplex $\{l_i, l_j\}$ of σ.

The Vietoris-Rips version of the weak witness complex $W_{\mathrm{VR^w}}(X, \mathcal{L})$ is also known as the Martinetz-Schulten complex $MS_1(X, \mathcal{L})$.

Definition 43. $W^w_{\mathrm{VR}}(X, \mathcal{L})$
Let (X, d) be a metric space, and let $\mathcal{L}$ be a subset of X. Then the complex $W^w_{\mathrm{VR}}(X, \mathcal{L})$ is the abstract simplicial complex $(\mathcal{L}, \Sigma)$, where $\sigma = \{l_0, \ldots, l_p\} \in \Sigma$ iff there $\exists$ a weak witness $x_{i,j}$ for each 1-subsimplex $\{l_i, l_j\}$ of σ.

Definition 44. $W^w_{\mathrm{VR}}(X, \mathcal{L}, \epsilon)$
Let (X, d) be a metric space, let $\mathcal{L}$ be a subset of X, and let $\epsilon \in \mathbb{R}$ be positive. Then the complex $W^w_{\mathrm{VR}}(X, \mathcal{L}, \epsilon)$ is the abstract simplicial complex $(\mathcal{L}, \Sigma)$, where $\sigma = \{l_0, \ldots, l_p\} \in \Sigma$ iff there $\exists$ an ϵ weak witness $x_{i,j}$ for each 1-subsimplex $\{l_i, l_j\}$ of σ.

A natural question about the Vietoris-Rips versions is: When do they give the same persistent homology as their non-VR versions? I do not have the answers to this, but it is an interesting question.

To get an overview of the validity of the strong witness complex (Delaunay complex), weak witness complex, ϵ strong witness complex, and the ϵ weak witness complex as methods on a discrete data set, see Figure 7 and Figure 8. An example showing simplicial complexes constructed by the different witness complex constructions is given in Example 3.9.

$$\begin{array}{ccccc} X & \xrightarrow[\text{lemma}]{\text{Nerve}} & \mathrm{Del(X,\mathcal{L})} = \mathrm{W^s(X,\mathcal{L})} & \xrightarrow[\text{theorem}]{\text{Weak witness}} & W^w(X,\mathcal{L}) \\ & & \vdots & & \downarrow \\ & & W^s(Z,\mathcal{L}) & & W^w(Z,\mathcal{L}) \end{array}$$

Figure 7: This gives an overview of the validity of the strong witness complex and the weak witness complex. In this diagram X is a topological space, Z is a discrete sample from X, and $\mathcal{L} \subseteq Z$ is the landmark point set. The solid arrows indicate plausible equality while the dashed arrows indicate most likely not equal.

$$\begin{array}{ccccc} X & \xrightarrow[\text{lemma}]{\text{Nerve}} & \mathrm{Del(X,\mathcal{L},\epsilon)} = \mathrm{W^s(X,\mathcal{L},\epsilon)} & \xrightarrow[\text{theorem}]{\epsilon\text{ weak witness}} & W^w(X,\mathcal{L},\epsilon) \\ & & \downarrow & & \downarrow \\ & & W^s(Z,\mathcal{L},\epsilon) & & W^w(Z,\mathcal{L},\epsilon) \end{array}$$

Figure 8: This gives an overview of the validity of the ϵ strong witness complex and the ϵ weak witness complex. In this diagram X is a topological space, Z is a discrete sample from X, and $\mathcal{L} \subseteq Z$ is the landmark point set. The solid arrows indicate plausible equality. For the ϵ weak witness theorem see [7].

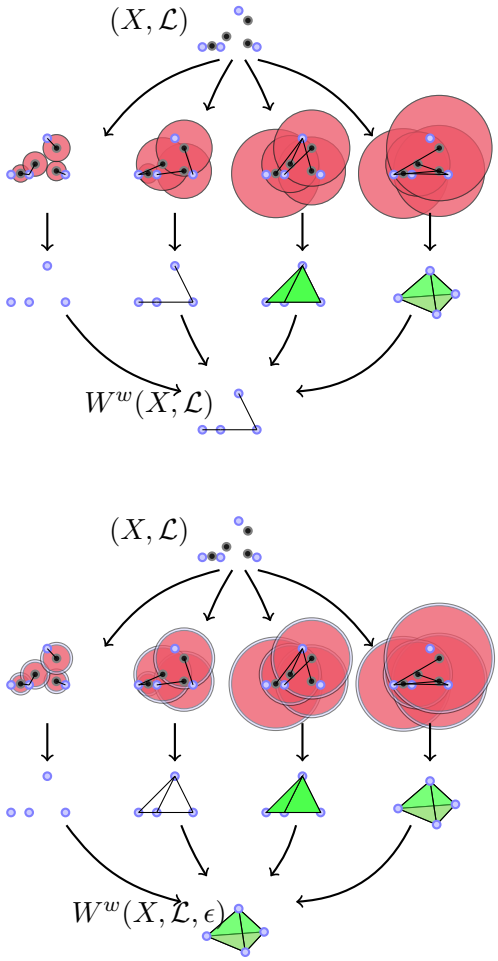
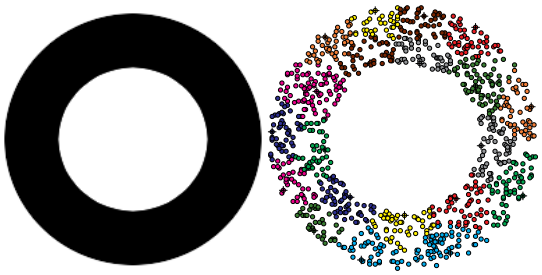


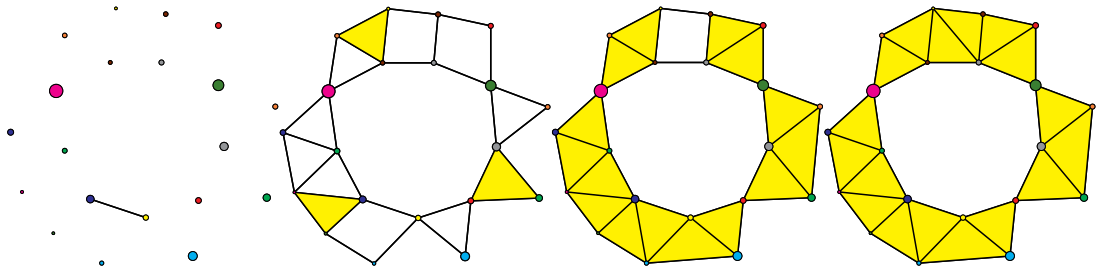
Figure 9: Illustrates the weak witness complex and the ϵ weak witness complex. The radius of the balls around the non-landmark points are dependent on the distance to the nearby landmark points. Each of the balls on the left side, i.e. the smallest ones, have its radius given by the distance to its closest landmark point. Note that each radius increases by ϵ in the illustration of the ϵ weak witness complex, i.e. in the lower figure. The radius of the balls second to the left depends on the distance to their second closet landmark point; the radius of the other balls are given in the same manner. When we build a witness complex, i.e. a complex that uses witnesses, the landmark points may also be witnesses. However, we omit drawing balls around the landmark points since those balls would have made the figures less readable and in this illustration we get the same results without using the landmark points as witnesses.

Example 3.9. *Let X be the annulus, let $S \subseteq X$ be a sample consisting of* 1000 *randomly selected points from X, and let $\mathcal{L}$ be the landmark point set as shown in Figure 10. Then the results from applying the different witness complex constructions, including the Delaunay construction on $(X, \mathcal{L})$, are shown in Figure 11.*



(a) X (b) Voronoi diagram

Figure 10: Refer to Example 3.9 for details. Figure 10a shows the space X while Figure 10b shows the Voronoi diagram, in which the landmark points are marked.



(a) DEL(X, $\mathcal{L}$) (b) $W^s(X, \mathcal{L}, \epsilon)$ (c) $W^w(X, \mathcal{L})$ (d) $W^w(X, \mathcal{L}, \epsilon)$

Figure 11: Refer to Example 3.9 for details. Results from applying various complex construction methods on the space shown in Figure 10a, with landmark points $\mathcal{L}$ as shown in Figure 10b.

3.8 CDT complexes

The combinatorial Delaunay triangulation (CDT) modifies our weak witness complex and the Vietoris-Rips variant $W^w_{\mathrm{VR}}(X, \mathcal{L})$ of the weak witness complex. Instead of just giving some slack, this method creates a new metric. The metric is defined by first creating a weighted graph; thereafter, the metric value on a pair of points is defined as the shortest path between the two points.

Definition 45. CDT^w and $\mathrm{CDT}^w_{\mathrm{VR}}$
Let (X, d) be a metric space, let $\mathcal{L}$ a subset of X, and let $k \in \mathbb{R}$ be positive. Furthermore, let (V, E, W) form a weighted graph, where $V = X$ is the set of vertices, E is the set of edges, and $W : (x, y) \mapsto d(x, y)$ is the function giving the weight between two vertices that share an edge. The set of edges is constructed by one of the two following methods:

Method 1: Let $k \in \mathbb{N}$. Then $(x, y) \in E$ if x is one of the k closest neighbours of y and y is one of the k closest neighbours of x.
Method 2: Let $k \in \mathbb{R}$. Then $(x, y) \in E$ if $d(x, y) < k$.

Let $d_g : X \times X \to \mathbb{R}$ give the shortest path between two connected vertices. Moreover, let $d_g(x, y) = \infty$ for two disconnected vertices x and y. Then the combinatorial Delaunay triangulations are given by

$$\mathrm{CDT}^w(X, \mathcal{L}, k) = W^w((X, d_g), \mathcal{L})$$

and

$$\mathrm{CDT}^w_{\mathrm{VR}}(X, \mathcal{L}, k) = W^w_{\mathrm{VR}}((X, d_g), \mathcal{L}).$$

Figure 12 and Figure 13 illustrates how the combinatorial Delaunay triangulation construction works. The combinatorial Delaunay triangulation is computationally efficient; moreover, the method has greater tolerance of nonlinearity and curvature than the other approaches that we have discussed, see [5] for results of comparison. An example

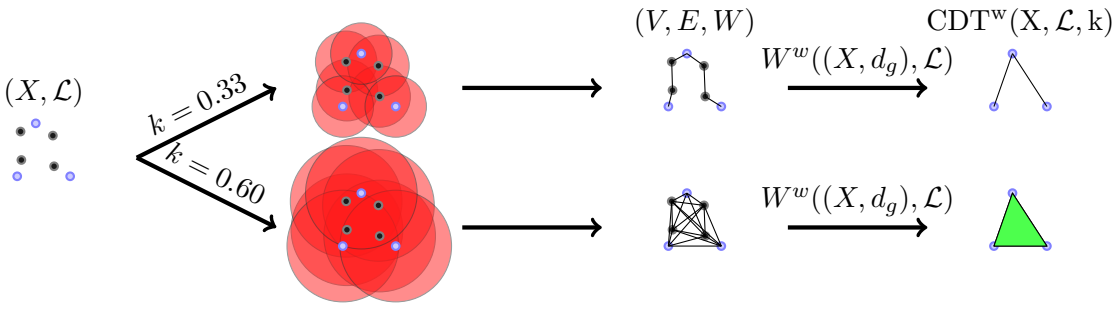


Figure 12: Illustration of how the combinatorial Delaunay triangulation construction works with method 1.

of when the CDT complex constructions does a good job is given in Example 3.10.

Example 3.10. *Let X be the space shown in Figure 14a, let $S \subseteq X$ be a sample consisting of 300 points, and let $\mathcal{L}$ be the landmark point set as shown in Figure 14b. Then the results from applying* $\mathrm{CDT^w}$ *on $(X, \mathcal{L})$ with $k = 0.7$ is given in Figure 15.*

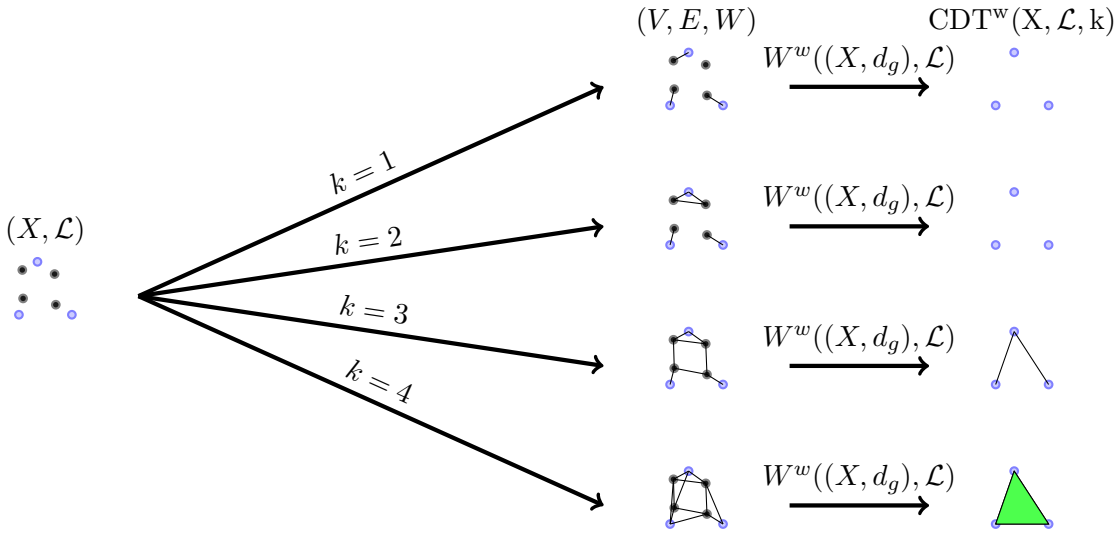
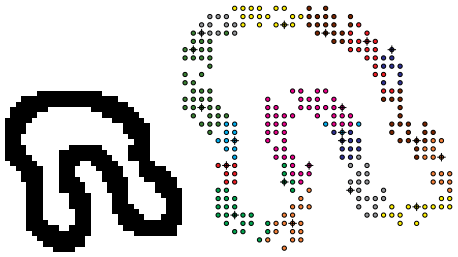


Figure 13: Illustration of how the combinatorial Delaunay triangulation construction works with method 2.



(a) X (b) Voronoi diagram

Figure 14: Refer to Example 3.10 for details. Figure 14a shows X while Figure 14b shows the Voronoi diagram of S, in which the landmark points are marked.

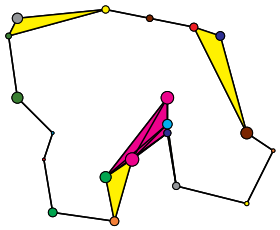


Figure 15: Refer to Example 3.10 for details. Results from applying $\mathrm{CDT^w}$ on the space shown in Figure 14.

3.9 Choosing landmark points

The methods we have introduced uses a set of landmark points to construct the simplices. The landmark point set is usually not given, and different landmark sets may give different results. Hence, we need some algorithm for selecting a landmark point set.

When we are selecting a landmark point set, we want the set to have the following properties.

1. Be well-separated.
2. Have more points where the data is dense or has a higher curvature.

There are two commonly used approaches for picking landmark points.

1. By random selection.
2. By the maxmin algorithm.

The maxmin algorithm is given in Algorithm 1. Both of them have pros and cons. The

Algorithm 1 Maxmin

Require: A metric space (X, d), and a number M of landmark points to pick.
Ensure: A set consisting of M landmark points.

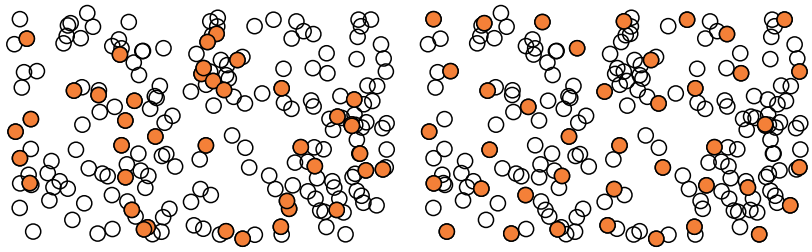
Randomly pick an element l_0 from X;
Let $\mathcal{L} = \{l_0\}$;
while $|\mathcal{L}| < M$ **do**
    Pick an element l' in X which maximizes the function $z \mapsto \min\{d(z,l)\}_{l\in\mathcal{L}}$;
    Let $\mathcal{L} = \mathcal{L} \cup \{l'\}$;
end while
return $\mathcal{L}$;

random selection may not give a well-separation; maxmin gives a well-separation, but it has a tendency to pick more extreme points, which are placed on the verge of the data set, and not so many points from where the data is dense. Moreover, the algorithmic process makes the theoretical analysis of the maxmin method more difficult.

An example of the random selection and the maxmin algorithm applied on a 2-dimensional random data set is given in Example 3.11.

Example 3.11. *Let X be some random data set in $\mathbb{R}^2$ consisting of 200 points. Then the results from applying random selection and the maxmin algorithm on X are shown in Figure 16.*

Another method for picking landmark points, which I suggest, consists of using Voronoi decompositions. The algorithm is given in Algorithm 2. This method tends to give landmark points more evenly spread than the random selection method, and it



(a) (b)

Figure 16: Refer to Example 3.11 for details. The circles represent data points while the disks represent landmark points. The result from random selection is shown in Figure 16a, and the result from the maxmin algorithm is shown in Figure 16b.

Algorithm 2 Voronoi landmarks(M^2n)

Require: A finite metric space (X, d) with n elements, and a number M of landmark points to pick.

Ensure: A set consisting of M landmark points.

1: Randomly pick an element l_0 from X;
2: Let $\mathcal{L} = \{l_0\}$;
3: **while** $|\mathcal{L}| < M$ **do**
4: Let $\{V_l\}_{l\in\mathcal{L}}$ be the Voronoi diagram of X with $\mathcal{L}$ as landmark points;
5: Pick a $l \in \mathcal{L}$ s.t. $|V_l| \geq |V_{l'}|$ for all $l' \in \mathcal{L}$;
6: Randomly pick an element $x \in V_l$ s.t. $x \neq l$;
7: Let $\mathcal{L} = \mathcal{L} \cup \{x\}$;
8: **end while**
9: **return** $\mathcal{L}$;

tends to select more landmark points in dense regions compared to the maxmin method. In Example 3.12 we give an example where the Voronoi landmark algorithm is more suited than the maxmin method; in Example 3.13 we give an example where it is more suited than the random selection method.

Example 3.12. *Let X be the space shown in Figure 17. If we use the maxmin method for selecting landmark points on X and do not use enough landmark points, then a typical resulting set of landmark points is as shown in Figure 18a. The landmark point set shown in Figure 18a consists of* 25 *points. With the same number of landmark points, the Voronoi landmark selection algorithm tends to give a result as shown in Figure 18b. If we increase the number of landmark points, then the maxmin method will eventually also select enough landmark points on the circle in the center. The tests done on our space X showed that to get enough points in the center, we needed about* 35 *landmark points with the maxmin method while we needed only* 25 *with the Voronoi landmark method.*

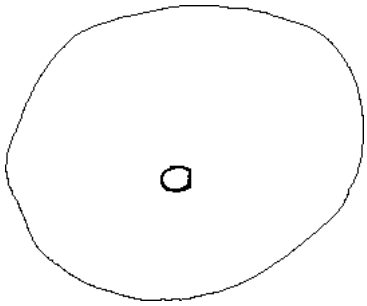
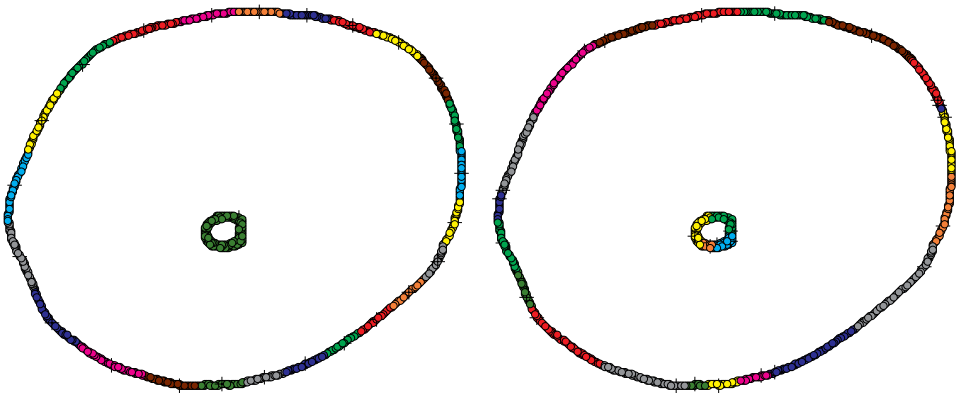


Figure 17: Refer to Example 3.12 for details. A space used as input in Example 3.12.



(a) (b)

Figure 18: Refer to Example 3.12 for details. Voronoi diagrams of the space shown in Figure 17. The Voronoi diagram in Figure 18a have landmark points selected by the maxmin method. Likewise, the Voronoi diagram in Figure 18a have landmark points selected by the Voronoi landmark selection method. Each Voronoi cell has a random assigned color, and each landmark point is marked by a cross.

Example 3.13. *Let X be the space shown in Figure 19. Let the complex construction method be the ϵ strong witness complex method with some fixed ϵ. Then a typical sequence of simplicial complexes created with $5, 10, \ldots, 50$ landmark points selected by random selection is shown in Figure 20. A typical sequence obtained with Voronoi landmark selection is shown in Figure 21.*

Figure 19: Refer to Example 3.13 for details. A space used as input in Example 3.13.

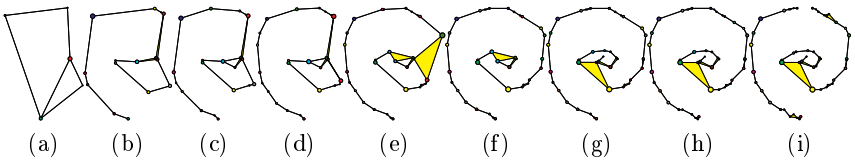


Figure 20: Refer to Example 3.13 for details. Simplicial complexes constructed with landmark point sets obtained by random selection. Note that each vertex has been assigned a random color, and that the size of each vertex is determined by the number of points in the corresponding Voronoi cell, i.e. large vertices have Voronoi cells which contains many points.

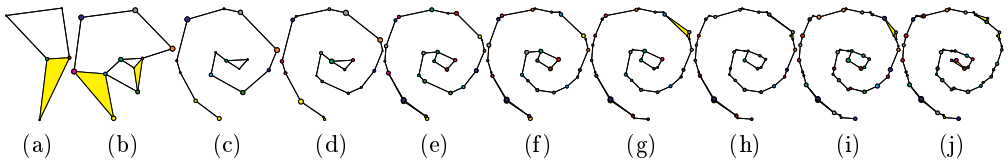


Figure 21: Refer to Example 3.13 for details. Simplicial complexes constructed with landmark point sets obtained by the Voronoi landmark algorithm. Note that each vertex has been assigned a random color, and that the size of each vertex is determined by the number of points in the corresponding Voronoi cell, i.e. large vertices have Voronoi cells which contains many points.

One downside with the Voronoi landmark algorithm compared to maxmin and random selection is that it is more computationally expensive. The Voronoi landmark algorithm consists of $\mathcal{O}(M^2 n)$ operations, where M is the number of landmark points and n is the number of points in the sample, while the maxmin method may be implemented with $\mathcal{O}(Mn)$ operations. The fastest of them is the random selection method, which has $\mathcal{O}(M)$ operations. However, in practice we often create a series of simplicial complexes by increasing some tolerance parameter while keeping the landmark point set fixed. This allows us to use the same landmark point set for constructing multiple simplicial complexes. A good algorithm for selecting landmark points makes it possible to get away with selecting fewer landmark points, which then greatly reduces the number of operations needed for calculating each simplicial complex.

4 Persistent homology

In this section we will introduce persistent homology. Resources have been gathered from [3], [18] and [26].

4.1 Motivation

Homology may be used to retrieve important quantitative information from a data set. One example, where we use simplicial homology on a complex derived from a data set, is given in Example 4.1.

Example 4.1. *Let $X \subseteq \mathbb{R}^2$ be a space consisting of n elements $\{x_i\}_{i=1}^n$, where each element corresponds to a sensor. Each sensor can detect objects in $\mathbb{R}^2$ that are closer than a distance ϵ. Furthermore, let $U_i = B(x_i, \epsilon)$. Then $\mathcal{U} = \{U_i\}_{i=1}^n$ gives a cover of the space observed by our sensors. By computing the homology of the simplicial complex given by the nerve of $\mathcal{U}$, we get the Betti numbers β_0 and β_1. Betti number β_0 tells us how many clusters there are in our sensor network while β_1 tells us how many "holes" there are in the space observed by our sensors. As an example, the sensor network in Figure 22 has five clusters and one hole.*

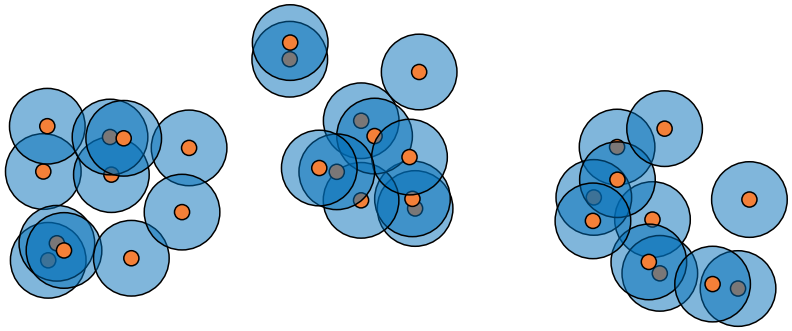


Figure 22: Refer to Example 4.1 for details. This figure illustrates a sensor network with five clusters and one hole.

Homology extracts pretty coarse information, and may not be a sufficient tool when comparing spaces. One example, where simplicial homology is insufficient, is given in Example 4.2.

Example 4.2. *The homology on the two simplicial complexes shown in Figure 23 does both give Betti numbers $\beta_0 = 4$, $\beta_1 = 3$, and $\beta_p = 0$ for $p > 1$.*

When we create simplicial complexes, there is often a natural way of making a nested sequence of subcomplexes $K_0 \subseteq \cdots \subseteq K_n = K$, which we will call a filtered complex of a simplicial complex K.

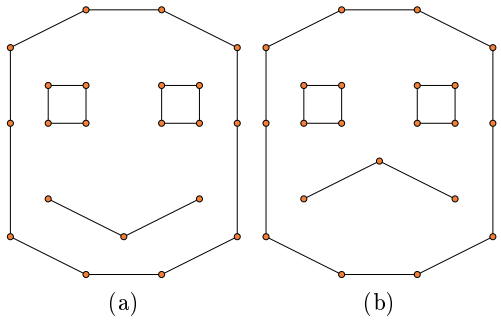


Figure 23: Two simplicial complexes with identical Betti numbers.

Definition 46. Filtered complex
Let $K^* = \{K^i\}_{i=1}^n$ be a sequence of simplicial complexes. Then a K^* is a filtered complex of K if $\emptyset = K^0 \subseteq K^1 \subseteq \cdots \subseteq K^n = K$.

One example, where we create a filtered complex to distinguish two objects, is given in Example 4.3.

Example 4.3. *Let $\check{K}$ and $\hat{K}$ be the simplicial complexes shown in Figure 23a and Figure 23b. Create the filtered complexes $\check{K}^*$ and $\hat{K}^*$ as shown in Figure 24 and Figure 25. By calculating the Betti numbers for each subcomplex, we can get the results shown in Figure 26. With these results we can see a significant difference between $\check{K}$ and $\hat{K}$.*

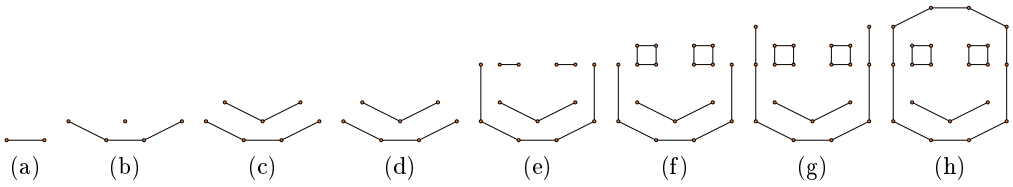


Figure 24: Refer to Example 4.4 for details. This figure illustrates a filtered complex constructed from the simplicial complex in Figure 23a.

The method of calculating the Betti numbers for each subcomplex in the filtered complex may also be too coarse. One example, where this is the case, is given in Example 4.4.

Example 4.4. *Let K and L be as shown in Figure 27a and Figure 27b. A natural way of constructing filtered complexes from these is shown in Figure 28 and Figure 29. With these filtered complexes, the method consisting of calculating the Betti numbers for each subcomplex will not give any difference.*

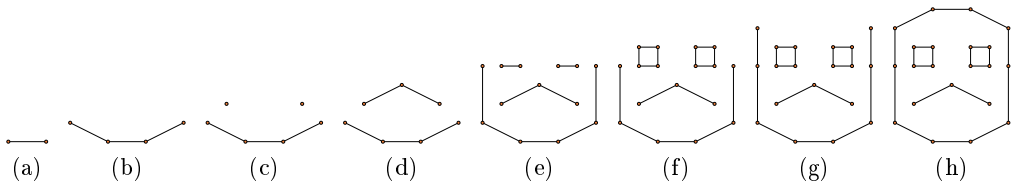


Figure 25: Refer to Example 4.4 for details. This figure illustrates a filtered complex constructed from the simplicial complex in Figure 23b.

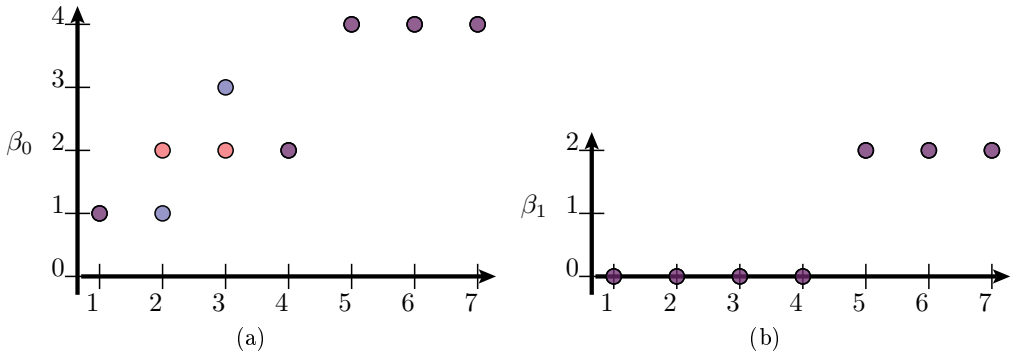


Figure 26: Refer to Example 4.3 for details. These figures show the Betti numbers of two different filtered complexes. The red dots correspond to the filtered complex in Figure 24 while the blue dots correspond to the filtered complex in Figure 25. Overlapping results give purple dots.

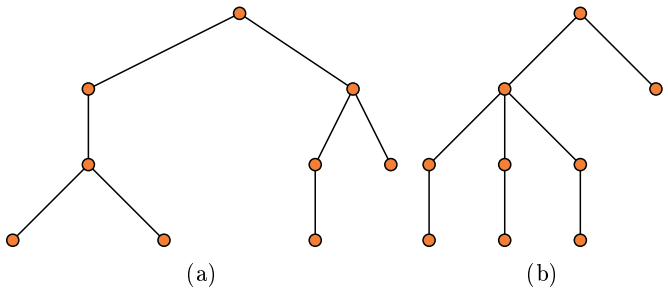


Figure 27: Refer to Example 4.4 for details.

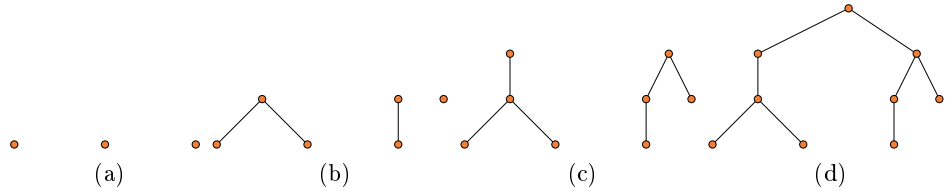


Figure 28: Refer to Example 4.4 for details. This figure illustrates a filtered complex constructed from the simplicial complex in Figure 27a.

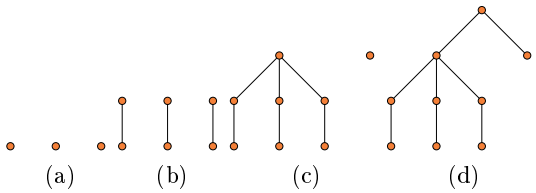


Figure 29: Refer to Example 4.4 for details. This figure illustrates a filtered complex constructed from the simplicial complex in Figure 27b.

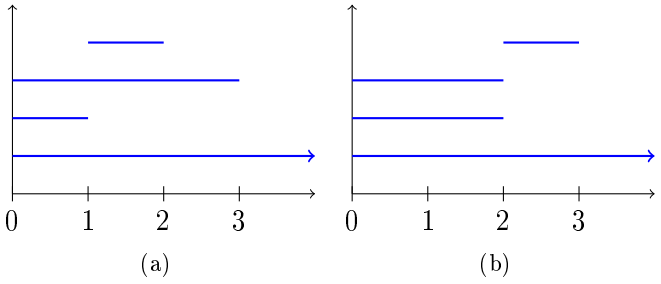


Figure 30: Barcodes for the filtered complexes shown in Figure 28 and Figure 29.

The information, we get by computing the Betti numbers for each of the complexes in a filtered complex, may be useful. However, this method does not take into account the fact that the simplicial complexes are nested. By using such information we may construct barcodes, as shown in Figure 30a and Figure 30b, from the filtered complexes in Example 4.4.

In this section we are going to discuss how we can make such barcodes. Speaking freely, each bar in the barcode represents a generator in the homology group of the filtered complex. Each bar starts at the time where the corresponding generator arises, and stops when the generator gets zeroed out. If a generator does not get zeroed out, then the corresponding barcode is endless. This kind of results have shown to be useful when dealing with sampled data from some space X. It is difficult to produce a simplicial complex, which is a good representation of X, from the sampled data. In Section 3 we introduced several methods for constructing simplicial complexes, which naturally give filtered complexes by increasing some epsilon parameter. By creating such filtered complexes, the artifacts tends to give short bars while the real properties tend to give longer bars. One example, where this is the case, is given in Example 4.5

Example 4.5. *Let X be an annulus, and let S be a sample consisting of* 200 *points from X. Let K^* be a filtered complex obtained by using the ϵ weak witness complex construction with* 35 *landmark points and varying epsilon. Then by applying persistent homology on K^*, we get the barcodes shown in Figure 31.*

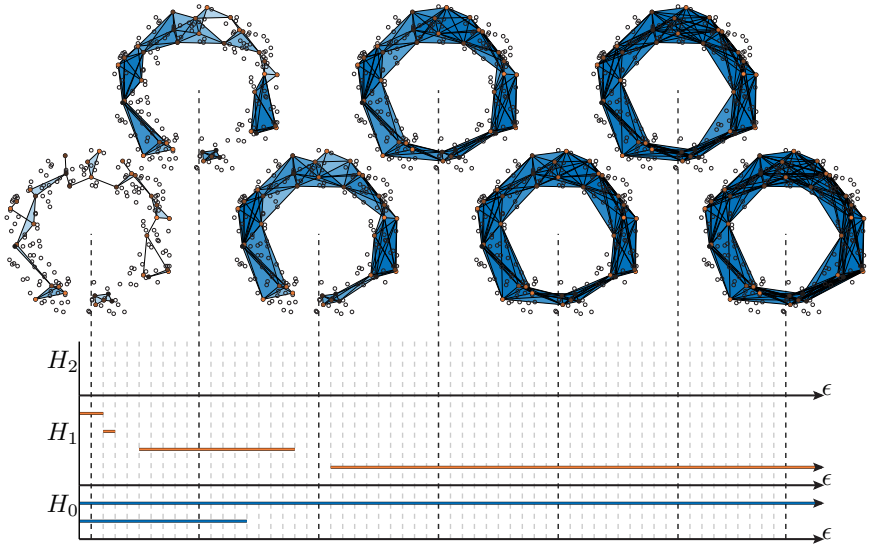


Figure 31: Refer to Example 4.5 for details. Barcodes from applying persistent homology on a filtered complex constructed from some sampled data from an annulus.

4.2 Algebra

Before we define persistent homology and do calculations, we need some definitions and propositions from algebra.

4.2.1 Graded rings and modules

Definition 47. Graded Ring
A ring R is called a graded ring, more precisely $\mathbb{Z}$-graded, if there exists a family of subgroups $\{R_n\}_{n\in\mathbb{Z}}$ of R s.t. $R = \oplus_n R_n$ as abelian groups, and $R_n R_m \subseteq R_{n+m}$. Moreover, a non-zero element $x \in R_n$ is called a homogeneous element of R of degree n.

Definition 48. Graded R-module
Let R be a graded ring, and let M be an R-module. Then M is a graded R-module if there $\exists$ a family of subgroups $\{M_n\}_{n\in\mathbb{Z}}$ of M s.t. $M = \oplus_n M_n$ (as abelian groups), and $R_n M_m \subseteq M_{n+m}$ for all n, m. Moreover, if $u \in M - \{0\}$ and $u = u_{i_1} + \cdots + u_{i_k}$, where $u_{i_j} \in R_{i_j} - \{0\}$ and $i_j \neq i_{j'}$ when $j \neq j'$, then $u_{i_1}, \ldots, u_{i_k}$ are called the homogeneous components of u.

Proposition 4.6. *Let R be a graded ring, let M be a graded R-module, and let N be a graded submodule of M. Then M/N is a graded R-module, where*

$$(M/N)_n = (M_n + N)/N = \{m + N | m \in M_n\}.$$

Definition 49. Graded R-homomorphism
Let R be a graded ring, let M and N be graded R-modules, and let $f : M \to N$ be an R-homomorphism. Then f is a graded R-homomorphism of degree d if $f(M_n) \subseteq N_{n+d}$ for all n.

Definition 50. Graded module isomorphism
Let R be a graded ring, let M and N be two graded R-modules, and let $f : M \to N$ be a graded R-homomorphism. Then f is a graded R-isomorphism if there $\exists$ an inverse of f, and if f is of degree zero.

Note. *Let M be a graded R-module, and let $n \in \mathbb{Z}$. Then we can define a new graded R-module $M(n)$ by twisting the grading on M by n, i.e. $M(n)_k = M_{n+k}$.*

4.2.2 Representing homomorphisms as matrices

Note that when we have an $\mathbb{F}[t]$-homomorphism $h : M \to N$ between two finitely generated graded $\mathbb{F}[t]$-modules M and N, with ordered bases $\mathcal{B} = [b_j]_{i=1}^m$ and $\mathcal{B}' = [b'_i]_{i=1}^n$, we can represent h as a matrix $A \in \mathbb{F}[t]^{n\times m}$. Moreover, the matrix $A = (a_{ij})$ is given by $a_{ij} = \pi_i \circ h(b_j)$ for $j = 1, \ldots, m$ and $i = 1, \ldots, n$, where π_i is the projection map $\pi_i : \sum_{k=1}^n c_k b'_k \mapsto c_i$, where $c_k \in \mathbb{F}[t]$ for $k = 1, \ldots, n$.

Proposition 4.7. *Let $h : M \to N$ be a graded $\mathbb{F}[t]$-homomorphism of degree zero, and let $A = (a_{ij}) \in \mathbb{F}[t]^{n\times m}$ be a matrix representing h with respect to to a homogeneous basis $([e_j]_{j=1}^m, [\tilde{e}_i]_{i=1}^n)$, i.e. $A : (M, [e_j]_{j=1}^m) \longrightarrow (N, [\tilde{e}_i]_{i=1}^n)$. Then the following equation holds.*

$$\deg \tilde{e}_i + \deg a_{ij} = \deg e_j$$

Notation. *Let M and N be graded $\mathbb{F}[t]$-modules, with bases $\mathcal{M}$ and $\mathcal{N}$. If $A : (M, \mathcal{M}) \to (N, \mathcal{N})$ is a matrix representation of some graded $\mathbb{F}[t]$-homomorphism with respect to the bases $\mathcal{M}$ and $\mathcal{N}$, then $(M, \mathcal{M})$ will be called the domain of A, and $(N, \mathcal{N})$ will be called the codomain of A. Furthermore, $\mathcal{M}$ will be called the domain basis of A, $\mathcal{N}$ will be called the codomain basis of A, and $(\mathcal{M}, \mathcal{N})$ will be called the basis of A.*

4.2.3 Column Operations

We will be using the following column operations. We will also talk about row operations, but we will not bother to define them since they are almost identical to the column operations.

Note that we will use the short notation e_{ij} for a matrix that has 1 as entity at position (i, j) and zero as entity otherwise.

Column operations
Let $A \in \mathbb{F}^{n\times m}$ be a matrix.
1. Switching two columns
Let $\tilde{A}$ be the matrix after switching column i and j, then $\tilde{A} = AE_{ij}$, where

$$E_{ij} = 1 - e_{ii} - e_{jj} + e_{ji} + e_{ij}.$$

Moreover, the inverse of E_{ij} is given by $E_{ij}^{-1} = E_{ij}$.
2. Scaling a column
Let $\tilde{A}$ be the matrix after multiplying column i by $\alpha \in \mathbb{F}$, then $\tilde{A} = AL_i(\alpha)$, where

$$L_i(\alpha) = 1 + (\alpha - 1)e_{ii}.$$

Moreover, the inverse of $L^i(\alpha)$ is given by $L_i^{-1}(\alpha) = L_i(\alpha^{-1})$.
3. Adding a column times an element to another column
Let $\tilde{A}$ be the matrix after adding column i times $\alpha \in \mathbb{F}$ to column j, then $\tilde{A} = AM_{ij}(\alpha)$, where

$$M_{ij}(\alpha) = 1 + \alpha e_{ij}.$$

Moreover, the inverse of $M_{ij}(\alpha)$ is given by $M_{ij}(\alpha)^{-1} = M_{ij}(-\alpha)$.

Note that the matrices E_{ij}, $L_i(\alpha)$, and M_{ij} are change of basis matrices, i.e. they change the basis of a matrix.

Definition 51. Change of basis matrix
Let M be a graded $\mathbb{F}[t]$-module, and let $[e_i]_{i=1}^m$ and $[\hat{e}_j]_{j=1}^m$ be two ordered bases for M. Then each $\hat{e}_j$ can be written as a unique linear combination of e_i's, say

$$\hat{e}_j = \sum_{i=1}^{m} p_{ij} e_i$$

for $j = 1, \ldots, m$, where $p_{ij} \in \mathbb{F}[t]$. The $m \times m$ matrix

$$P = (p_{ij}) : (M, [\hat{e}_j]_{j=1}^m) \to (M, [e_i]_{i=1}^m)$$

is called change of basis matrix, or more precisely a matrix of transformation from $[\hat{e}_j]_{j=1}^m$ to $[e_i]_{i=1}^m$.

Change of basis matrices may be used to change both the domain basis and the codomain basis of a matrix representation. Let M and N be two graded $\mathbb{F}[t]$-modules, let $\mathcal{M}$ and $\hat{\mathcal{M}}$ be two bases for M, let $\mathcal{N}$ and $\hat{\mathcal{N}}$ be two bases for N, and let $A : (M, \mathcal{M}) \to (N, \mathcal{N})$ be some matrix representation of a graded $\mathbb{F}[t]$-homomorphism. Furthermore, let $V : (M, \hat{\mathcal{M}}) \to (M, \mathcal{M})$ be a change of basis matrix on M, and let $U : (N, \mathcal{N}) \to (N, \hat{\mathcal{N}})$ be a change of basis matrix on N. Then we may use V to change the domain basis of A and U to change the codomain basis of A as follows.

$$AV : (M, \hat{\mathcal{M}}) \to (M, \mathcal{M})$$
$$UA : (M, \mathcal{M}) \to (N, \hat{\mathcal{N}})$$
$$UAV : (M, \hat{\mathcal{M}}) \to (N, \hat{\mathcal{N}})$$

Notation. *When doing column operations, we will use the following notations.*
1. *Switching column i and column j: $C_i \leftrightarrow C_j$.*
2. *Scaling a column by $\alpha \in \mathbb{F}$: αC_i.*
3. *Adding column i times α to column j: $C_j + \alpha C_i$.*

Example 4.8. *Let $A = \left(\begin{smallmatrix} 1 & 2 \\ 3 & 5 \end{smallmatrix}\right)$, and let $\tilde{A}$ be the result of doing the following column operations.*

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 5 \end{pmatrix} \underset{\sim}{C_1 \leftrightarrow C_2} \begin{pmatrix} 2 & 1 \\ 5 & 3 \end{pmatrix} \underset{\sim}{2C_2} \begin{pmatrix} 2 & 2 \\ 5 & 6 \end{pmatrix} \underset{\sim}{C_2 - C_1} \begin{pmatrix} 2 & 0 \\ 5 & 1 \end{pmatrix} = \tilde{A}$$

Then

$$\begin{aligned} \tilde{A} &= AE_{12}L_2(2)M_{12}(-1) \\ &= \left(\begin{smallmatrix} 1 & 2 \\ 3 & 5 \end{smallmatrix}\right)\left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)\left(\begin{smallmatrix} 1 & 0 \\ 0 & 2 \end{smallmatrix}\right)\left(\begin{smallmatrix} 1 & -1 \\ 0 & 1 \end{smallmatrix}\right) \\ &= \left(\begin{smallmatrix} 1 & 2 \\ 3 & 5 \end{smallmatrix}\right)\left(\begin{smallmatrix} 0 & 2 \\ 1 & 0 \end{smallmatrix}\right)\left(\begin{smallmatrix} 1 & -1 \\ 0 & 1 \end{smallmatrix}\right) \\ &= \left(\begin{smallmatrix} 1 & 2 \\ 3 & 5 \end{smallmatrix}\right)\left(\begin{smallmatrix} 0 & 2 \\ 1 & -1 \end{smallmatrix}\right) \\ &= \left(\begin{smallmatrix} 2 & 0 \\ 5 & 1 \end{smallmatrix}\right). \end{aligned}$$

In this example, the matrix $\left(\begin{smallmatrix} 0 & 2 \\ 1 & -1 \end{smallmatrix}\right)$ *is the change of basis matrix. Let this change of basis matrix be called* V*, then* $\tilde{A} = AV$*. Moreover, since* $V = E_{12}L_2(2)M_{12}(-1)$*, the inverse is given by* $M_{12}^{-1}(-1)L_2^{-1}(2)E_{12}^{-1} = M_{12}(1)L_2(0.5)E_{12} = \left(\begin{smallmatrix} 0.5 & 1 \\ 0.5 & 0 \end{smallmatrix}\right)$.

Note. *A change of basis matrix represents the identity map. Hence, it has degree zero.*

4.2.4 Algorithm for Column reduction

Algorithm 3 Column reduction($n \times m$)

Require: A matrix $M = (m_{ij})$ representing a graded $\mathbb{F}[t]$-homomorphism of degree zero.
Ensure: The column echelon form of M.

```
1:  p=1;
2:  for i = 1, ..., n do
3:      for j = p, ..., m do
4:          if m_ij ≠ 0 then
5:              C_p ↔ C_j;
6:              for j' = p + 1, ..., m do
7:                  C_j' = C_j' − (m_ij'/m_ip) C_p;
8:              end for
9:              p = p + 1;
10:             break;
11:         end if
12:     end for
13: end for
14: return M;
```

4.2.5 Smith normal form

In this section we introduce an algorithm for computing the Smith normal form of matrices representing graded $\mathbb{F}[t]$-homomorphisms.

Given a graded homomorphisms $h : M \longrightarrow N$ of degree zero, where M and N are finitely graded $F[t]$-modules with m and n generators respectively. Let $A : (M, \mathcal{M}) \rightarrow (N, \mathcal{N})$ be a matrix in $\mathbb{F}[t]^{n\times m}$ representing h with respect to some ordered bases $\mathcal{M}$ and $\mathcal{N}$. Furthermore, let $\mathcal{M} = [e_i]_{i=1}^m$ and $\mathcal{N} = [\tilde{e}_i]_{i=1}^n$ be homogeneous bases, and let $A = (a_{ij})$.Since h also has degree zero, property 4.7 holds. Hence,

$$\deg \tilde{e}_i + \deg a_{ij} = \deg e_j$$

for $0 \leq i \leq n$ and $0 \leq j \leq m$. There are 4 steps in the computation of the Smith normal form.

Step 1: Sort $[\tilde{e}_i]_{i=1}^n$ by decreasing degree, and let $\tilde{\mathcal{N}}$ denote the reordered basis. This

is done by interchanging rows on A. Let P^{-1} denote the corresponding change of basis matrix $P^{-1} : \mathcal{N} \longrightarrow \tilde{\mathcal{N}}$.

Step 2: Sort $[e_i]_{i=1}^m$ by increasing degree, and let $\tilde{\mathcal{M}}$ denote the reordered basis. This is done by interchanging columns on A. Let Q denote the corresponding change of basis matrix $Q : \tilde{\mathcal{M}} \longrightarrow \mathcal{M}$. Let $\tilde{A} : (M, \tilde{\mathcal{M}}) \to (N, \tilde{\mathcal{N}})$ be the matrix given by $\tilde{A} = P^{-1}AQ$, where P and Q are the change of basis matrices on N and M. Furthermore, let $(\tilde{a}_{ij}) = \tilde{A}$. Now that the domain and codomain bases are ordered, we have

$$\deg \tilde{a}_{ij} \leq \deg \tilde{a}_{ij'}, \text{ i.e. } \tilde{a}_{ij} | \tilde{a}_{ij'},$$

for $1 \leq j \leq j' \leq m$ and $0 \leq i \leq n$, and

$$\deg \tilde{a}_{ij} \leq \deg \tilde{a}_{i'j}, \text{ i.e. } \tilde{a}_{ij} | \tilde{a}_{i'j}, \tag{1}$$

for $1 \leq j \leq m$ and $0 \leq i \leq i' \leq n$. Hence, it is possible to use our column reduction algorithm on $\tilde{A}$.

Step 3: Perform Column reduction. This gives us a new matrix $\hat{A} = \tilde{A}\tilde{V}$, where $\tilde{V}$ is a change of basis matrix on M. Let $(M, \tilde{\mathcal{M}}^{\tilde{V}})$ denote the domain of $\tilde{V}$. Then $\tilde{V} : (M, \tilde{\mathcal{M}}^{\tilde{V}}) \longrightarrow (M, \tilde{\mathcal{M}})$, and $\hat{A} : (M, \tilde{\mathcal{M}}^{\tilde{V}}) \longrightarrow (N, \tilde{\mathcal{N}})$. This new matrix $\hat{A}$ is on the following form.

$$\begin{pmatrix} * & 0 & 0 & 0 & \cdots & 0 \\ * & * & 0 & 0 & \cdots & 0 \\ * & * & 0 & 0 & \cdots & 0 \\ * & * & * & 0 & \cdots & 0 \\ * & * & * & 0 & \cdots & 0 \end{pmatrix}$$

Moreover, since the column operations does not change the basis of the codomain N, Equation 1 still holds. Hence, it is possible to perform row reduction on $\hat{A}$.

Step 4: Perform Row reduction. This gives us a new matrix $\bar{A} = \tilde{U}\hat{A} = \tilde{U}\tilde{A}\tilde{V}$, where $\tilde{U}$ is the change of basis matrix on N. Let $(N, \tilde{\mathcal{N}}^{\tilde{U}})$ be the domain of $\tilde{U}$. Then $\tilde{U} : (N, \tilde{\mathcal{N}}) :\longrightarrow (N, \tilde{\mathcal{N}}^{\tilde{U}})$, and $\bar{A} : (M, \tilde{\mathcal{M}}^{\tilde{U}}) \longrightarrow (N, \tilde{\mathcal{N}}^{\tilde{V}})$. The new matrix $\bar{A}$ will be on the form

$$\begin{pmatrix} d_1 & & & & \\ & \ddots & & & \\ & & d_r & & \\ & & & 0 & \\ & & & & \ddots \\ & & & & & 0 \end{pmatrix},$$

where $d_1, \ldots, d_r \in \mathbb{F}[t]$ are the diagonal elements; moreover, they will have the following property

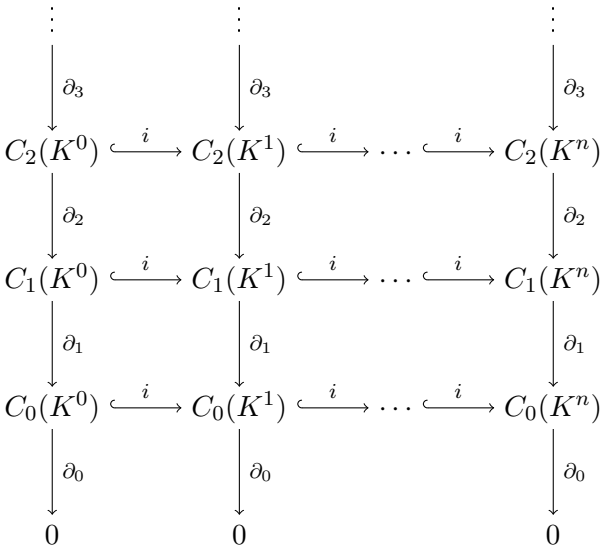
$$d_1 | \cdots | d_r.$$

By letting $U = \tilde{U}P^{-1}$ and $V = Q\tilde{V}$, we get that $\bar{A} = UAV$, which then gives us a Smith normal form representation of A.

4.3 Construction

We wish to define some homology on filtered complexes that takes into account that we have inclusion maps between the subcomplexes. The homology we will introduce is called persistent homology. The first step of this method this is to construct the chain complex.

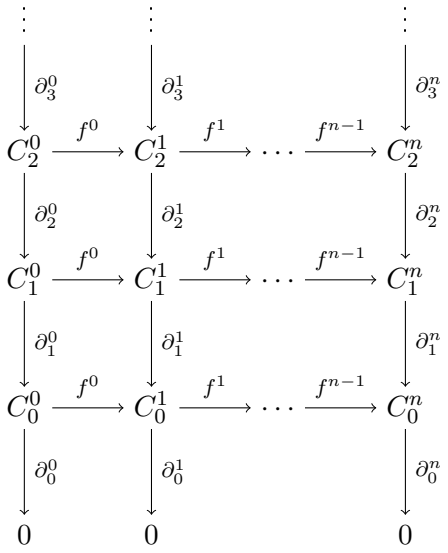
In simplicial homology we already have a method for creating chain complexes from simplicial complexes. Applying that method on the simplicial complexes in our filtered complex $K^* = \{K^p\}_{p=1}^n$, gives us a sequence of chain complexes, but also induced inclusion maps as follows.



We will call such a construction for a persistence complex.

Definition 52. Persistence complex
Let $\left\{C_*^i\right\}_{i\geq 0}$ be a sequence of chain complexes, and let $\left\{f^i\right\}_{i\geq 0}$ be a sequence of maps $f^i : C_*^i \to C_*^{i+1}$. Then the sequence $\left\{C_*^i, f^i\right\}_{i\geq 0}$ is a persistence complex.

The structure of a persistence complex may be represented as the following diagram.



A persistence complex $\left\{C_*^i, f^i\right\}_{i\geq 0}$ has a natural corresponding chain complex $\{\tilde{C}_p, \tilde{\partial}_p\}_{p\geq 0}$, where each element $\tilde{C}_p$ is a graded $\mathbb{F}[t]$-module. Given a persistence complex $\left\{C_*^k, i^k\right\}_{k\geq 0}$ [3], our chain complex $\{\tilde{C}_p, \tilde{\partial}_p\}_{p\geq 0}$ will be defined as follows. Let $\tilde{C}_p$ be the direct product of the $\mathbb{F}$-module components $\left\{C_p^k\right\}_{k\geq 0}$, and give it a grading by letting $(\tilde{C}_p)_k = C_p^k$. Since $C_p^k \subseteq C_p^{k'}$ for $k \leq k'$, we can define a map $\mathbb{F}[t] \times \tilde{C}_p \to \tilde{C}_p$, given by $(at, c) \mapsto at(c^0, c^1, \ldots) = (0, ac^0, ac^1, \ldots)$ for $c = (c^0, c^1, \ldots) \in \tilde{C}_p$ and $a \in \mathbb{F}$, which then gives $\tilde{C}_p$ a graded $\mathbb{F}[t]$-module structure. The boundary map $\tilde{\partial} = \left\{\tilde{\partial}_p\right\}_{p\geq 0}$ will be a graded $\mathbb{F}[t]$-homomorphism of degree zero, and is given by $\tilde{\partial}_p(c) = (\partial_p^0(c^0), \partial_p^1(c^1), \ldots)$, where $c = (c^0, c^1, \ldots) \in \tilde{C}_p$. Moreover,

$$\tilde{\partial}_p \circ \tilde{\partial}_{p+1}(c) = (\partial_p^0 \circ \partial_{p+1}^0(c), \partial_p^1 \circ \partial_{p+1}^1(c), \ldots) = (0, 0, \ldots) = 0$$

for $c \in \tilde{C}_{p+1}$ and $\forall p \geq 0$. Hence, it gives us a boundary map on our chain complex.

Now that we have defined our chain complex, we can define the persistent homology by taking the homology of our chain complex, i.e.

$$\tilde{H}_p = \frac{\ker \tilde{\partial}_p}{\operatorname{im} \tilde{\partial}_{p+1}} \ \forall p \geq 0.$$

Note that $\ker \tilde{\partial}_p$ is a graded $\mathbb{F}[t]$-module, and that $\operatorname{im} \tilde{\partial}_{p+1}$ is a graded $\mathbb{F}[t]$-submodule of $\ker \tilde{\partial}_p$. Hence, $\tilde{H}_p$ is a graded $\mathbb{F}[t]$-module $\forall p \geq 0$.

[3] Note that we do not need the sequence of maps $\left\{i^k\right\}_{k\geq 0}$ to be a sequence of inclusion maps, but restricting them to be inclusion maps allows us to shorten the notation.

When we are working with chain complexes, we will often refer to the standard basis of the chain complex. In simplicial homology the simplices in a simplicial complex K induce a standard basis for each module $C_p(K)$ in the chain complex $C_*(K)$. Let $\left\{C_p^k\right\}_{k\geq 0, p\geq 0}$ be the elements in our persistence complex, and for each $k \geq 0$ and $p \geq 0$ let $B_p^k = \left\{(b_p^k)_i\right\}_{i\in I_p^k}$ be the standard basis for C_p^k. Note that $B_p^k \subseteq B_p^{k+1}$ for $k \geq 0$. Let π_p^k be the inclusion map $\pi_p^k : C_p^k \hookrightarrow \tilde{C}_p$. Then the standard bases for our graded $\mathbb{F}[t]$-modules $\{\tilde{C}_p\}_{p\geq 0}$ in $\tilde{C}_*$ are given by

$$\tilde{B}_p = \bigcup_{k\geq 0} \pi_p^k(B_p^k - B_p^{k-1})$$

for $p \geq 0$, where $B_p^k = \emptyset$ for $k < 0$.

To make this more clear, we will give the standard basis for the chain complex constructed from the filtered simplicial complex shown in Figure 34.

Example 4.9. *Let $\tilde{C}_*$ be the chain complex constructed from the filtered complex in Figure 34. Then we will have the following collection of sets as basis:*

- $\tilde{C}_0 : \{(a, 0, \dots), (b, 0, \dots), (0, c, 0, \dots), (0, d, 0, \dots)\}$
- $\tilde{C}_1 : \{(0, ab, 0, \dots), (0, bc, 0, \dots), (0, 0, cd, 0, \dots), (0, 0, ad, 0, \dots), (0, 0, 0, ac, 0, \dots)\}$
- $\tilde{C}_2 : \{0, 0, 0, 0, abc, 0, \dots), (0, 0, 0, 0, 0, acd, 0, \dots\}$

Moreover,

$$\begin{aligned}
\tilde{C}_0 &= \mathbb{F}[t]a \oplus \mathbb{F}[t]b \oplus \mathbb{F}[t]tc \oplus \mathbb{F}[t]td && \cong \mathbb{F}[t] \oplus \mathbb{F}[t] \oplus \mathbb{F}[t]t \oplus \mathbb{F}[t]t \\
\tilde{C}_1 &= \mathbb{F}[t]tab \oplus \mathbb{F}[t]tbc \oplus \mathbb{F}[t]t^2cd \oplus \mathbb{F}[t]t^2ad \oplus \mathbb{F}[t]t^3ac && \cong \mathbb{F}[t]t \oplus \mathbb{F}[t]t \oplus \mathbb{F}[t]t^2 \oplus \mathbb{F}[t]t^2 \oplus \mathbb{F}[t]t^3 \\
\tilde{C}_2 &= \mathbb{F}[t]t^4abc \oplus \mathbb{F}[t]t^5acd && \cong \mathbb{F}[t]t^4 \oplus \mathbb{F}[t]t^5
\end{aligned}$$

Note that we will shorten the notation by dropping the zeros, i.e. we will write ab instead of $(0, ab, 0, \dots)$. Moreover, since we will only talk about boundary maps on and homology of chain complexes derived from filtered complexes, we will use the following notations:

- $\partial = \tilde{\partial}$,
- $\partial_p = \tilde{\partial}_p$,
- $H_p = \tilde{H}_p$.

4.4 Decomposition

The structure theorem, given for finitely generated graded modules over a PID, states that a finitely generated graded module M over a graded PID D decomposes uniquely into the form

$$M \cong \left(\bigoplus_{i=1}^{n} D(\alpha_i) \right) \oplus \left(\bigoplus_{j=1}^{m} \frac{D}{(d_j)}(\gamma_j) \right),$$

where $d_j \in D$ are homogeneous elements s.t. $d_j | d_{j+1}$ and $\alpha_i, \gamma_j \in \mathbb{Z}$. Recall that $D(n)$ is given by $(D(n))_k = D_{n+k}$. Since our graded modules are graded modules over the graded polynomial ring $\mathbb{F}[t]$, where $\mathbb{F}$ is a field, the only ideals of our graded modules are $\left\{(t^i)\right\}_{i=0}^{\infty}$. Hence, a graded $\mathbb{F}[t]$ module M decomposes uniquely into the form

$$M \cong \left(\bigoplus_{i=1}^{n} \mathbb{F}[t](\alpha_i) \right) \oplus \left(\bigoplus_{j=1}^{m} \frac{\mathbb{F}[t]}{(t^{\beta_j})}(\gamma_j) \right),$$

where $\alpha_i \beta_i, \gamma_i \in \mathbb{Z}$ for $i = 1, \ldots, n$ and $j = 1, \ldots, m$; moreover, $\beta_j | \beta_{j+1}$.

This structure theorem shows that a finitely generated graded $\mathbb{F}[t]$-module can be identified by two collections $\{\alpha_i\}_{i=1}^{n}$ and $\{(\beta_j, \gamma_j)\}_{j=1}^{m}$. We will use a similar class to represent our finitely generated graded $\mathbb{F}[t]$-modules, namely the class consisting of finite collections of $\mathcal{P}$-intervals.

Definition 53. $\mathcal{P}$-interval
A $\mathcal{P}$-interval is an ordered pair (i, j) with $0 \leq i < j \in \mathbb{Z}^{\infty} = \mathbb{Z} \cup \{+\infty\}$.

Note. *Note that a collection of $\mathcal{P}$-intervals may have multiple equal entities.*

This class can be associated to the set of finitely generated graded $\mathbb{F}[t]$-modules by a bijection $Q : \mathcal{S} \mapsto Q(\mathcal{S})$, where $\mathcal{S} = \{(i_1, j_1), \ldots, (i_n, j_n)\}$ is a collection of $\mathcal{P}$-intervals and $Q(\mathcal{S})$ is a finitely generated graded $\mathbb{F}[t]$-module. This is done by defining

$$Q(i, j) = \begin{cases} \mathbb{F}[t](-i) & \text{if } j = \infty, \\ \frac{\mathbb{F}[t]}{(t^{j-i})}(-i) & \text{otherwise,} \end{cases}$$

and

$$Q(\mathcal{S}) = \bigoplus_{l=1}^{n} Q(i_l, j_l).$$

Corollary. *The correspondence $\mathcal{S} \mapsto Q(\mathcal{S})$ defines a bijection between the finite collections of $\mathcal{P}$-intervals and the finitely generated graded $\mathbb{F}[t]$-modules.*

One of the positive things with this representation is that it has an intuitive illustration as a barcode. The following figure, Figure 32, illustrates how a graded module can be represented as a barcode.

Example 4.10. *The collection $\mathcal{S} = \{(0, \infty), (1, 2), (2, 4), (0, 3), (5, \infty), (1, 2)\}$ of $\mathcal{P}$-intervals gives the barcode in Figure 33.*

Note that the order of the lines in the barcode is unimportant.

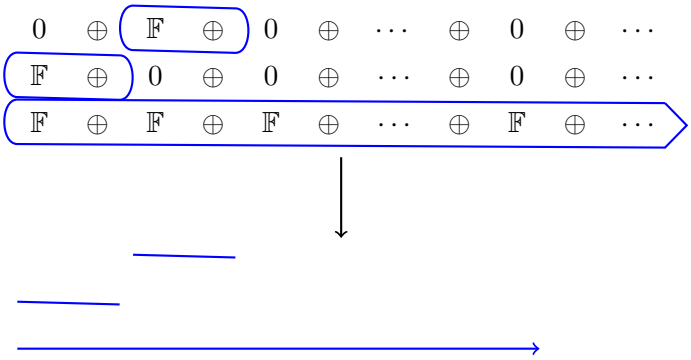


Figure 32: Illustration of how a graded module can be represented as a collection of $\mathcal{P}$-intervals.

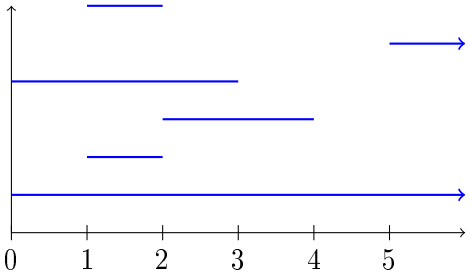


Figure 33: Refer to Example 4.10 for details. Barcode for the $\mathcal{P}$-interval collection $\mathcal{S} = \{(0,\infty), (1,2), (2,4), (0,3), (5,\infty), (1,2)\}$.

4.5 Calculation

In this section we will show how the persistent homology of a filtered complex may be calculated. We will start by showing a simple but computationally less efficient method. Thereafter we will show how the method may be improved.

Let us recall what we want to achieve and what we are given. What we have is a set of standard bases, one for each C_p, together with the boundary matrices M_p, which represent the boundary maps ∂_p. What we want is to calculate $H_p = \ker M_p / \operatorname{im} M_{p+1}$ as a graded $\mathbb{F}[t]$-module. We can determine H_k if we have a basis $\{z_i\}_{i=1}^{s}$ for $\ker M_p$ and a basis $\{b_i\}_{i=1}^{r}$ for $\operatorname{im} M_{p+1}$ s.t. $(b_j) \subseteq (z_{\phi(j)})$ for $j = 1, \dots, r$ and for some 1-1 function $\phi : [1, r] \to [1, s]$. Let us assume ϕ is the injection map, i.e. $\phi(i) = i$, s.t. $(b_i) \subseteq (z_i)$ for $i = 1 \dots r$. Then we get that

$$H_k \cong \frac{(z_1)}{(b_1)} \oplus \cdots \oplus \frac{(z_r)}{(b_r)} \oplus (z_{r+1}) \oplus \cdots \oplus (z_s).$$

Let $\alpha_i = \deg z_{r+i}$ for $i = 1, \ldots, s-r$, let $\beta_j = \deg b_j - \deg z_j$, and let $\gamma_j = \deg z_j$ for $j = 1, \ldots, r$. Then

$$H_k \cong \left(\bigoplus_{i=1}^{n} \mathbb{F}[t](\alpha_i) \right) \oplus \left(\bigoplus_{j=1}^{m} \frac{\mathbb{F}[t]}{(t^{\beta_j})}(\gamma_j) \right).$$

Moreover, represented as a set of $\mathcal{P}$-intervals we get

$$\mathcal{S} = \{(\alpha_i, +\infty)\}_{i=1}^{s-r} \cup \{(\gamma_j, \gamma_j + \beta_j)\}_{j=1}^{r}.$$

There are multiple paths for finding such bases. If we have had a matrix $\bar{M}_{p+1}$ also representing ∂_{p+1}, but with $(\mathcal{C}_{p+1}, \mathcal{Z}_p)$ instead of $(\mathcal{C}_{p+1}, \mathcal{C}_p)$ as basis, where $\mathcal{Z}_p$ is a basis for $\ker M_p = \ker \partial_p$, then we could find suitable bases $\{b_i\}_{i=1}^{r}$ and $\{z_i\}_{i=1}^{s}$, with the property $(b_i) \subseteq (z_i)$ for $i = 1, \ldots, r$, by computing the Smith normal form of $\bar{M}_{p+1}$. Let us call the Smith normal form of $\bar{M}_{p+1}$ for $\hat{M}_{p+1} = \bar{U}_{p+1}\bar{M}_{p+1}\bar{V}_{p+1}$. Then $\hat{M}_{p+1}$ can be written as

$$\begin{array}{c|cccccc} & \bar{v}_1 & \cdots & \bar{v}_r & \bar{v}_{r+1} & \cdots & \bar{v}_m \\ \hline \bar{z}_1 & d_1 & & & & & \\ \vdots & & \ddots & & & & \\ \bar{z}_r & & & d_r & & & \\ \bar{z}_{r+1} & & & & 0 & & \\ \vdots & & & & & \ddots & \\ \bar{z}_s & & & & & & 0 \end{array},$$

where $([\bar{v}_j]_{j=1}^{m}, [\bar{z}_i]_{i=1}^{s})$ is the basis for $\hat{M}_{p+1}$. The image of M_{p+1}, i.e. $\operatorname{im} \partial_{p+1}$, is given by the basis $\{d_i \bar{z}_i\}_{i=1}^{r}$, where $\{d_i\}_{i=1}^{r}$ are the diagonal elements of $\hat{M}_{p+1}$. Hence, if we let $b_i = d_i \bar{z}_i$, then $\{b_i\}_{i=1}^{r}$ and $\{\bar{z}_i\}_{i=1}^{s}$ will be a suitable set of bases.

To be able to do this, we first need a map such as $\bar{M}_{p+1}$ that is a matrix representation of ∂_{p+1} and has $\ker \partial_p$ as codomain. There are different approaches for constructing such a map. Since $\operatorname{im} M_{p+1} \subseteq \ker \partial_p$, we can change the codomain of M_{p+1} by doing row operations on M_{p+1}, and then remove some rows s.t. the new codomain basis is a basis for $\ker \partial_p$. Note that this change of codomain basis can be described by two maps Q and D, where Q is a change of basis matrix and D is a map which delete rows. Hence, we can construct $\bar{M}_{p+1}$ by

$$\bar{M}_{p+1} = DQM_{p+1}.$$

What we need is a method for finding some suitable maps Q and D s.t. $DQ[e_i]_{i=1}^{n}$, where $[e_i]_{i=1}^{n} = \mathcal{C}_p$, is an ordered basis for $\ker \partial_p$. One way of doing this is to use a change of basis matrix V that reduce M_p to a column echelon matrix M_pV. Let s be the number

of zero columns in M_pV, then the last s elements of the domain basis $\mathcal{C}_p^V = [v_i]_{i=1}^n$ for V give an ordered basis $\mathcal{Z}_p$ for $\ker \partial_p$. If we let D be the map deleting all the rows, except the last s rows, then $DV^{-1}[e_i]_{i=1}^n = \mathcal{Z}_p$. Moreover,

$$DV^{-1}M_{p+1} : (C_{p+1}, \mathcal{C}_{p+1}) \to (\ker \partial_p, \mathcal{Z}_p)$$

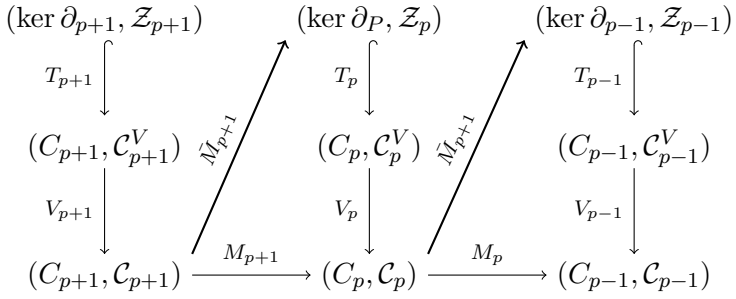
will be a matrix, with $\ker \partial_p$ as codomain, representing ∂_{p+1}. Note that $D|_{\ker \partial_p} = T_p^{-1}$, where T_p is the injection $T_p : (\ker \partial_p, \mathcal{Z}_p) \hookrightarrow \left(C_p, \mathcal{C}_p^V\right)$. Hence, $D|_{\operatorname{im} M_{p+1}} = T_p^{-1}|_{\operatorname{im} M_{p+1}}$, where

$$T_p = \left(\begin{array}{c|ccc} & z_1 & \cdots & z_s \\ \hline v_1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ v_{n-s} & 0 & \cdots & 0 \\ \hline v_{n-s+1} & 1 & & 0 \\ \vdots & & \ddots & \\ v_n & 0 & & 1 \end{array}\right).$$

We will therefore write T_p^{-1} instead of D. Hence,

$$T_p^{-1}V^{-1}M_{p+1} : (C_{p+1}, \mathcal{C}_{p+1}) \to (\ker \partial_p, \mathcal{Z}_p)$$

is a matrix, with $\ker \partial_p$ as codomain, representing ∂_{p+1}. The following commutative diagram gives an overview of how $\bar{M}_{p+1}$ is constructed.



4.5.1 Summary

Summing up our algorithm for calculating the persistent homology, we get the following steps.

Step 1: Calculate the column echelon form M_pV_p for each $p = 0, \ldots, N$, where N is some number s.t. $C_p = 0$ for $p > N$. This does also give us T_p for each p.

Step 2: Calculate $\bar{M}_{p+1} = T_p^{-1}V_p^{-1}M_{p+1}$ for each $p = 0, \ldots, N$, by calculating the inverse matrices and by doing the matrix multiplications.

Step 3: For each $p = 0, \ldots, N$, reduce $\bar{M}_{p+1}$ to Smith normal form, this gives a matrix with some nonzero diagonal elements $\{d_i^p\}_{i=1}^r$ and a codomain basis $[z_i^p]_{i=1}^s$, where $r \leq s$.
Then: For each $p = 0, \ldots, N$, H_p is given by

$$H_p \cong \bigoplus_{i=1}^{r} \frac{\mathbb{F}[t]}{(t^{\deg d_i^p})}(-\deg z_i^p) \oplus \bigoplus_{j=r+1}^{s} \mathbb{F}[t](-\deg z_j^p)$$

Moreover, the $\mathcal{P}$-interval collections are given by

$$\mathcal{S}_p = \bigcup_{i=1}^{r} \{(\deg z_i^p, \deg z_i^p + \deg d_i^p)\} \bigcup_{j=r+1}^{s} \left\{(\deg z_j^p, \infty)\right\}.$$

4.6 Improved algorithm

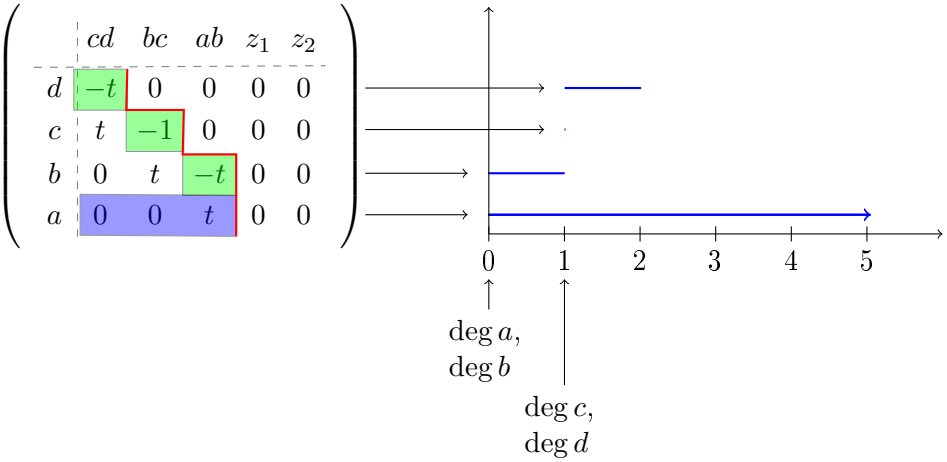
In this section we will introduce some shortcuts, and then use them in an example, where we calculate the persistent homology of the filtered complex in Figure 34.

Let us first look on how we can improve step 3. In step 3 we reduce $\bar{M}_{p+1}$ to Smith normal form; this is computationally expensive, and it would be nice if we do not need to do a full Smith normal form reduction. What we need from the Smith normal form is the degree of the codomain basis elements, the degree of the diagonal elements, and the knowledge of which pairs of diagonal elements and basis elements that share rows. The following theorem states that we can get this information by just reducing the matrix to column echelon form if we order the codomain basis decreasingly by degree before doing the reduction.

Theorem 4.11. *Let M be a matrix representing a graded $\mathbb{F}[t]$-homomorphism, with codomain basis $[\hat{e}_i]_{i=1}^n$, s.t. M is in column echelon form and $[\hat{e}_i]_{i=1}^n$ has a decreasing order. Let $\tilde{M}$ be the matrix after doing the row operations which makes M into a matrix in normal form, and let $[\tilde{e}_i]_{i=1}^n$ be the codomain basis of $\tilde{M}$. Let p_1,...,p_r be the pivot elements of M, where p_1, which has row number $\phi(1)$, is the first pivot element, p_2, which has row number $\phi(2)$, is the second pivot element, etc. Furthermore, let $d_1, \ldots, d_s$ be the non-zero diagonal elements of $\tilde{M} = (\tilde{m}_{ij})$, where $d_i = \tilde{m}_{ii}$ for $1 \leq i \leq s$. Then $s = r$ and $p_i = d_i$ for $1 \leq i \leq r$. Moreover, $\deg \tilde{e}_{\phi(i)} = \deg \hat{e}_i$ for $1 \leq i \leq r$.*

Proof. Let $\{e_j\}_{j=1}^m$ be the domain basis of M. Because $[\hat{e}_i]_{i=1}^n$ is sorted, the degree of the codomain elements $\hat{e}_i$ are monotonically decreasing from the top row down. For each fixed column j, the degree of e_j is a constant c. By Proposition 4.7, $\deg m(i,j) = c - \deg \hat{e}_i$. Therefore, the degree of the elements in each column is monotonically increasing with row. Hence, we may eliminate the non-zero elements below the pivot elements by using row operations, which do not change the pivot elements or the degree of the codomain basis elements. After doing this, we may place the matrix in diagonal form using only row and column swaps. □

Corollary. *Let M_k be a matrix in column echelon form that represents ∂_k with $(C_k, [e_j]_{j=1}^m)$ as domain and $(Z_{k-1}, [\hat{e}_i]_{i=1}^n)$ as codomain. If row i has a pivot element with degree d, then it contributes $(\mathbb{F}[t]/t^d)(-\deg \hat{e}_i)$ to the description of H_{k-1}; if the row does not have a pivot element, it contributes $\mathbb{F}[t](-\deg \hat{e}_i)$. Equivalently, we get $(\deg \hat{e}_i, \deg \hat{e}_i + d)$ and $(\deg \hat{e}_i, \infty)$ as $\mathcal{P}$-intervals for H_{k-1}.*



Let us now look at step 1. In this step we construct the matrices T_p and V s.t.

$$T_p^{-1}V^{-1} : (C_p, \mathcal{C}_p) \to (\ker \partial_p, \mathcal{Z}_p),$$

where $\mathcal{Z}_p$ is some ordered basis for $\ker \partial_p$, is the identity when restricted to $\ker \partial_p$. For this sake we might as well use the change of basis matrix that we get from step 3 instead; since the last s elements of the domain basis of this matrix will also give a basis for $\ker \partial_p$. Hence, if we compute the persistent homology for each p in ascending order, we may skip step 1. Note that step 1 is trivial for $p = 0$, i.e. $M_0 = 0$ and $T_0^{-1}V^{-1} = I$.

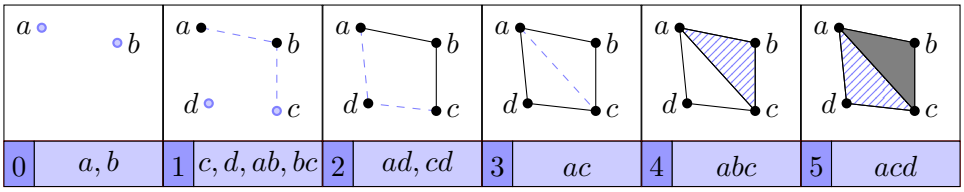


Figure 34: A filtered complex with newly added simplices highlighted.

Example 4.12. *In this example we calculate the persistent homology of the filtered complex shown in Figure 34. We will do this by first calculating H_0, then H_1 and so on.*

Calculating H_0: *The map ∂_0 is the zero map, hence $M_0 = 0$; moreover, M_0 is diagonal*

since it does not have any non-zero elements. The Smith normal form decomposition of M_0 *can then be written as*

$$\hat{M_0} = IM_0I,$$

where I *is the identity matrix. The injection map* T_0 *is also equal to* I *because* $\ker \partial_0 = C_0$. *Hence,*

$$\bar{M_1} = T_0^{-1}V_0^{-1}M_1 = IIM_1 = M_1.$$

The matrix $M_1 = (m_{ij}^1)$ *is known and given by* $m_{ij}^1 = \pi_i \circ \partial_0(e_j)$, *where* π_i *is the projection map* $\pi_i : a_1\tilde{e}_1 + \cdots + a_n\tilde{e}_n \mapsto a_i$ *for* $1 \leq i \leq 4$ *and* $1 \leq j \leq 5$. *The matrix* M_2 *is given by the same fashion, and the matrices becomes as follows.*

$$M_1 = \begin{array}{c|ccccc} & ab & bc & cd & ad & ac \\ \hline a & t & 0 & 0 & t^2 & t^3 \\ b & -t & t & 0 & 0 & 0 \\ c & 0 & -1 & t & 0 & -t^2 \\ d & 0 & 0 & -t & -t & 0 \end{array} \qquad M_2 = \begin{array}{c|cc} & abc & acd \\ \hline ab & t^3 & 0 \\ bc & t^3 & 0 \\ cd & 0 & t^3 \\ ad & 0 & -t^3 \\ ac & -t & t^2 \end{array}$$

Next, we want to reduce M_1 *into a matrix in column echelon form. This may be done as follows.*

$$M_1 = \begin{array}{c|ccccc} & ab & bc & cd & ad & ac \\ \hline a & t & 0 & 0 & t^2 & t^3 \\ b & -t & t & 0 & 0 & 0 \\ c & 0 & -1 & t & 0 & -t^2 \\ d & 0 & 0 & -t & -t & 0 \end{array} \qquad \begin{matrix} R_1 \leftrightarrow R_5 \\ R_2 \leftrightarrow R_3 \\ \sim \end{matrix} \begin{array}{c|ccccc} & ab & bc & cd & ad & ac \\ \hline d & 0 & 0 & -t & -t & 0 \\ c & 0 & -1 & t & 0 & -t^2 \\ b & -t & t & 0 & 0 & 0 \\ a & t & 0 & 0 & t^2 & t^3 \end{array}$$

$$\begin{matrix} C_1 \leftrightarrow C_3 \\ \sim \end{matrix} \begin{array}{c|ccccc} & cd & bc & ab & ad & ac \\ \hline d & -t & 0 & 0 & -t & 0 \\ c & t & -1 & 0 & 0 & -t^2 \\ b & 0 & t & -t & 0 & 0 \\ a & 0 & 0 & t & t^2 & t^3 \end{array} \qquad \begin{matrix} C_4 - 1C_1 \\ \sim \end{matrix} \begin{array}{c|ccccc} & cd & bc & ab & * & ac \\ \hline d & -t & 0 & 0 & 0 & 0 \\ c & t & -1 & 0 & -t & -t^2 \\ b & 0 & t & -t & 0 & 0 \\ a & 0 & 0 & t & t^2 & t^3 \end{array}$$

$$\begin{matrix} C_4 - tC_2 \\ C_5 - t^2C_2 \end{matrix} \sim \left(\begin{array}{c|ccccc} & cd & bc & ab & * & * \\ \hline d & -t & 0 & 0 & 0 & 0 \\ c & t & -1 & 0 & 0 & 0 \\ b & 0 & t & -t & -t^2 & -t^3 \\ a & 0 & 0 & t & t^2 & t^3 \end{array}\right) \begin{matrix} C_4 - tC_3 \\ C_5 - t^2C_3 \end{matrix} \sim \left(\begin{array}{c|ccccc} & cd & bc & ab & z_1^1 & z_1^2 \\ \hline d & -t & 0 & 0 & 0 & 0 \\ c & t & -1 & 0 & 0 & 0 \\ b & 0 & t & -t & 0 & 0 \\ a & 0 & 0 & t & 0 & 0 \end{array}\right)$$

Note that $z_1^1 = ad - cd - t \cdot bc - t \cdot ab$ *and* $z_1^2 = ac - t^2 \cdot bc - t^2 \cdot ab$. *Let* $\tilde{M}_1$ *be the matrix obtained after ordering the codomain basis, and let* V_1 *be the change of domain basis matrix. Then the matrices* $\tilde{M}_1$ *and* V_1 *are given as follows.*

$$\tilde{M}_1 = \left(\begin{array}{c|ccccc} & ab & bc & cd & ad & ac \\ \hline d & 0 & 0 & -t & -t & 0 \\ c & 0 & -1 & t & 0 & -t^2 \\ b & -t & t & 0 & 0 & 0 \\ a & t & 0 & 0 & t^2 & t^3 \end{array}\right) \quad V_1 = \left(\begin{array}{c|ccccc} & cd & bc & ab & z_1^1 & z_1^2 \\ \hline ab & 0 & 0 & 1 & -t & -t^2 \\ bc & 0 & 1 & 0 & -t & -t^2 \\ cd & 1 & 0 & 0 & -1 & 0 \\ ad & 0 & 0 & 0 & 1 & 0 \\ ac & 0 & 0 & 0 & 0 & 1 \end{array}\right)$$

Note. *We do not need to calculate* V_1. *What we use to calculate* H_0 *is the reduced column echelon form* $\tilde{M}_1 V_1$, *which we already have calculated. We use* V_1^{-1} *to calculate* $\bar{M}_2$, *but we do not need to calculate* V_1 *since we can calculate* V_1^{-1} *directly. Despite this, we will calculate* V_1 *anyway since it gives a good example of how a change of basis matrix is constructed.*

We may calculate V_1 *by taking the product of the column operation matrices.*

$$V_1 = E_{13}M_{14}(-1)\left(M_{24}(-t)M_{25}(-t^2)\right)\left(M_{34}(-t)M_{35}(t^2)\right)$$

$$V_1 = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -t & -t^2 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -t & -t^2 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Now that we have our matrix in column echelon form, we can read of the pivot elements.

$$\tilde{M}_1 V_1 = \left(\begin{array}{c|ccccc} & cd & bc & ab & z_1 & z_2 \\ \hline d & -t & 0 & 0 & 0 & 0 \\ c & t & -1 & 0 & 0 & 0 \\ b & 0 & t & -t & 0 & 0 \\ a & 0 & 0 & t & 0 & 0 \end{array}\right)$$

This gives the pivot elements $p_1 = -t, p_2 = -1$ *and* $p_3 = -t$*, and by Corollary* 4.6 *we get that*

$$\begin{aligned} H_0 &\cong \frac{\mathbb{F}[t](-\deg d)}{(t)} \oplus \frac{\mathbb{F}[t](-\deg b)}{(t)} \oplus \mathbb{F}[t](-\deg a) \\ &= \frac{\mathbb{F}[t](-1)}{(t)} \oplus \frac{\mathbb{F}[t]}{(t)} \oplus \mathbb{F}[t]. \end{aligned}$$

Moreover, the collection of $\mathcal{P}$*-intervals becomes* $\{(0,\infty),(0,1),(1,2)\}$*, and the barcode becomes as shown in Figure 35.*

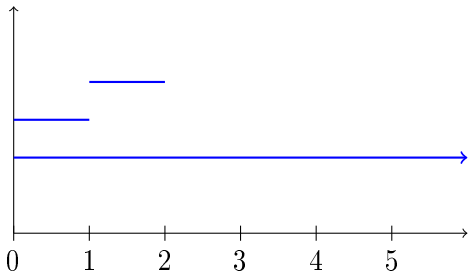


Figure 35: Barcode for dimension zero of the filtered complex in Figure 34.

Calculating H_1***:*** *In a similar fashion as we calculated* V_1*, we may calculate* V_1^{-1} *by constructing the inverse column operation matrices, and then take the product of these. Doing this gives us the following.*

$$\begin{aligned} V_1^{-1} &= \left(M_{35}(t^2)^{-1} M_{34}(-t)^{-1}\right)\left(M_{25}(-t^2)^{-1} M_{24}(-t)^{-1}\right) M_{14}(-1)^{-1} E_{13}^{-1} \\ V_1^{-1} &= \left(M_{35}(-t^2) M_{34}(t)\right)\left(M_{25}(t^2) M_{24}(t)\right) M_{14}(1) E_{13} \end{aligned}$$

$$V_1^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & t & t^2 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & t & t^2 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Taking the product of these matrices gives.

$$V_1^{-1} = \left(\begin{array}{c|ccccc} & ab & bc & cd & ad & ac \\ \hline cd & 0 & 0 & 1 & 1 & 0 \\ bc & 0 & 1 & 0 & t & t^2 \\ ab & 1 & 0 & 0 & t & t^2 \\ z_1^1 & 0 & 0 & 0 & 1 & 0 \\ z_1^2 & 0 & 0 & 0 & 0 & 1 \end{array}\right)$$

The matrix $T_1 : (\ker \partial_1, \mathcal{Z}_1) \hookrightarrow (C_1, \mathcal{C}_1^V)$ *is just the injection matrix, and the inverse* T_1^{-1} *is equal to the transpose of* T_1*. Moreover, the matrices* T_1 *and* T_1^{-1} *are given as follows.*

$$T_1 = \left(\begin{array}{c|cc} & z_1^1 & z_1^2 \\ \hline cd & 0 & 0 \\ bc & 0 & 0 \\ ab & 0 & 0 \\ \hline z_1^1 & 1 & 0 \\ z_1^2 & 0 & 1 \end{array}\right) \qquad T_1^{-1} = \left(\begin{array}{c|ccc|cc} & cd & bc & ab & z_1^1 & z_1^2 \\ \hline z_1^1 & 0 & 0 & 0 & 1 & 0 \\ z_1^2 & 0 & 0 & 0 & 0 & 1 \end{array}\right)$$

Now that we have both V_1^{-1} *and* T_1^{-1}*, we can calculate* $\bar{M}_2 = T_1^{-1} V_1^{-1} M_2$*. For pedagogical reasons we also calculate* $V_1^{-1} M_2$ *and* $T_1^{-1} V_1^{-1}$*.*

$$\bar{M}_2 = T_1^{-1} V_1^{-1} M_2 = \left(\begin{array}{c|cc} & abc & acd \\ \hline z_1^1 & 0 & -t^3 \\ z_1^2 & -t & t^2 \end{array}\right) \qquad T_1^{-1} V_1^{-1} = \left(\begin{array}{c|ccc|cc} & ab & bc & cd & ad & ac \\ \hline z_1^1 & 0 & 0 & 0 & 1 & 0 \\ z_1^2 & 0 & 0 & 0 & 0 & 1 \end{array}\right)$$

$$V_1^{-1} M_2 = \left(\begin{array}{c|cc} & abc & acd \\ \hline cd & 0 & 0 \\ bc & 0 & 0 \\ ad & 0 & 0 \\ z_1^1 & 0 & -t^3 \\ z_1^2 & -t & t^2 \end{array}\right)$$

Ordering the codomain of $\bar{M}_2$ *and performing column reduction give the following matrix.*

$$\bar{M}_2 = \begin{array}{c|cc} & abc & acd \\ \hline z_1^1 & 0 & -t^3 \\ z_1^2 & -t & t^2 \end{array} \quad \underset{\sim}{R_1 \leftrightarrow R_2} \quad \begin{array}{c|cc} & abc & acd \\ \hline z_1^2 & -t & t^2 \\ z_1^1 & 0 & -t^3 \end{array} \quad \underset{\sim}{C_2 + tC_1} \quad \begin{array}{c|cc} & acd & * \\ \hline z_1^2 & -t & 0 \\ z_1^1 & 0 & -t^3 \end{array}$$

This gives the pivot elements $p_1 = -t$ *and* $p_2 = -t^3$*, and by Corollary 4.6 we get that*

$$\begin{aligned} H_1 &\cong \frac{\mathbb{F}[t]}{(p_1)}(-\deg z_1^2) \oplus \frac{\mathbb{F}[t]}{(p_2)}(-\deg z_1^1) \\ &= \frac{\mathbb{F}[t]}{(t)}(-3) \oplus \frac{\mathbb{F}[t]}{(t^3)}(-2). \end{aligned}$$

Moreover, the collection of $\mathcal{P}$*-intervals becomes* $\{(3,4),(2,5)\}$*, and the barcode becomes as shown in Figure 36.*

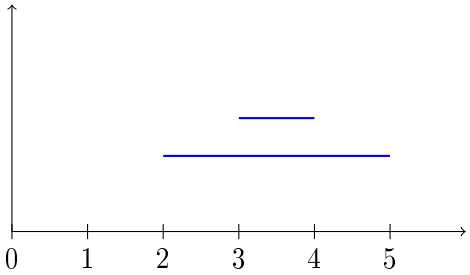


Figure 36: Barcode for dimension one of the filtered complex in Figure 34.

Calculating H_2***:*** *The next step is to calculate* H_2*. When we order the bases of* M_2 *and reduce it to column echelon form, we get that* $\ker \partial_2 = \ker M_2 = \emptyset$*.*

$$M_2 = \begin{array}{c|cc} & abc & acd \\ \hline ab & t^3 & 0 \\ bc & t^3 & 0 \\ cd & 0 & t^3 \\ ad & 0 & -t^3 \\ ac & -t & t^2 \end{array} \quad \tilde{M}_2 = \begin{array}{c|cc} & abc & acd \\ \hline ac & -t & t^2 \\ ad & 0 & -t^3 \\ cd & 0 & t^3 \\ bc & t^3 & 0 \\ ab & t^3 & 0 \end{array} \quad \tilde{M}_2 V_2 = \begin{array}{c|cc} & abc & * \\ \hline ac & -t & 0 \\ ad & 0 & -t^3 \\ cd & 0 & t^3 \\ bc & t^3 & t^4 \\ ab & t^3 & t^4 \end{array}$$

Hence,

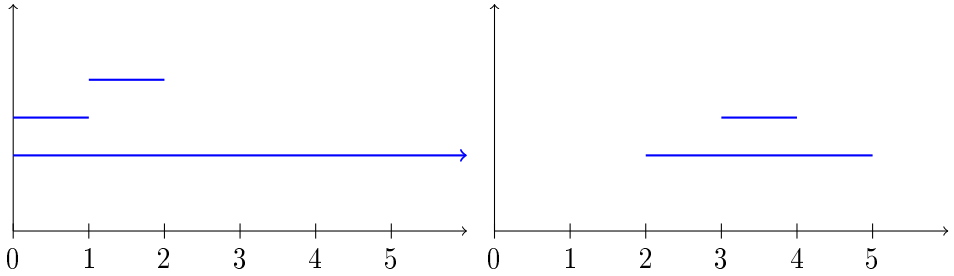
$$H_2 = 0.$$

Calculating $H_p, p > 2$*: Since* $C_p^* = 0$*, i.e. there are no* p*-simplices in our filtered complex, for* $p > 2$*, the persistent homology groups* H_p *will also be equal to zero for* $p > 2$*. Hence, we have*

$$H_p = 0$$

for $p > 2$*.*

Summing up: *We are now done with the calculations. Summing up we get that our filtered complex has three* 0*-cycles; one that persists forever and two that persists for one time interval each before they merge with the first cycle. Our filtered complex does also have two* 1*-cycles, which persist for one and three steps before they are filled in.*



(a) H_0 (b) H_1

Figure 37: Barcodes for the filtered complex in Figure 34.

4.7 Improved algorithm 2

Another shortcut, which is worth mentioning, is that we may construct the matrix $\bar{M}_{p+1}$ by simply deleting rows in M_{p+1}. Note that the new codomain basis of $\bar{M}_{p+1}$ will in general not have the same elements as the codomain basis of M_{p+1}; however, the degree of the basis elements in the codomain of $\bar{M}_{p+1}$ are equal to those of the corresponding basis elements in the codomain of M_{p+1}.

Theorem 4.13. *Let* $A : (N, [e_i^N]_{i=1}^n) \to (K, [e_i^K]_{i=1}^k)$ *and* $B : (M, [e_i^M]_{i=1}^m) \to (N, [e_i^N]_{i=1}^n)$ *be matrix representations of two graded* $\mathbb{F}[t]$*-homomorphisms* $g : N \to K$ *and* $f : M \to N$*, where* g *and* f *both have degree zero and* $g \circ f = 0$*. Let* σ *be the composition of the swapping operations executed while performing our column reduction algorithm on* A*. Let* r *be the rank of* A*, and let* $I_{\text{piv}} = \sigma^{-1}(\{1, \dots, r\})$*. Then there* $\exists$ *a basis* $[z_i]_{i=1}^{n-r}$ *of* $\ker A$ *s.t. the matrix* $\bar{B} : (M, [e_i^M]_{i=1}^m) \to (\ker A, [z_i]_{i=1}^{n-r})$*, obtained by deleting the rows of* B *that have indices in* I_{piv}*, is a matrix representation of* f*. Moreover, the degree of* z_i *is equal to the degree of the* i*'th element in the codomain basis of* B *that has not been removed.*

Proof. The permutation σ^{-1} induces a new order, given by $[e^N_{\sigma^{-1}(i)}]^n_{i=1}$ and denoted $[e^\sigma_i]^n_{i=1}$, from the ordered basis $[e^N_i]^n_{i=1}$. It does also induce a permutation matrix

$$P_{\sigma^{-1}} : (N, [e^\sigma_i]^n_{i=1}) \to (N, [e^N_i]^n_{i=1})$$

given by

$$P_{\sigma^{-1}} : e^\sigma_i \mapsto e^N_{\sigma^{-1}(i)}$$

for $i = 1, \dots, n$. Note that $P^{-1}_{\sigma^{-1}}$ is given by

$$P^{-1}_{\sigma^{-1}} : e^N_i \mapsto e^\sigma_{\sigma(i)}$$

for $i = 1, \dots, n$. Let $\tilde{A}$ be the matrix obtained by performing our column reduction algorithm on A, and let $[\tilde{e}_i]^n_{i=1}$ be the domain basis of $\tilde{A}$. Then the last $n - r$ elements of $[\tilde{e}_i]^n_{i=1}$ give a basis for $\ker A$. Moreover, $\tilde{A}$ can be represented as $\tilde{A} = AP_{\sigma^{-1}}Q$, where Q has the following form.

$$Q = \left(\begin{array}{c|ccc|ccc} & \tilde{e}_1 & \cdots & \tilde{e}_r & \tilde{e}_{r+1} & \cdots & \tilde{e}_n \\ \hline e^\sigma_1 & 1 & * & * & * & * & * \\ \vdots & & \ddots & * & \vdots & \vdots & \vdots \\ e^\sigma_r & & & 1 & * & * & * \\ \hline e^\sigma_{r+1} & & & & 1 & & \\ \vdots & & 0 & & & \ddots & \\ e^\sigma_n & & & & & & 1 \end{array}\right)$$

Let $\tilde{z}_i = \tilde{e}_{i+r}$ for $i = 1, \dots, n - r$. Then $[\tilde{z}_i]^{n-r}_{i=1}$ is an ordered basis for $\ker A$, and $T : \tilde{z}_i \mapsto \tilde{e}_{i+r}$ gives an injection $T : (\ker A, [\tilde{z}_i]^{n-r}_{i=1}) \to (N, [\tilde{e}_i]^n_{i=1})$. Since $\operatorname{im} B \subseteq \ker A$, the map $T^{-1}Q^{-1}P^{-1}_{\sigma^{-1}}B : (M, [e^M_i]^m_{i=1}) \to (\ker B, [\tilde{z}_i]^{n-r}_{i=1})$ is well defined.

We are now going to show that by reordering the codomain of this matrix, we will get the desired matrix $\bar{B}$. Note that Q^{-1} and T^{-1} have the following forms

$$Q^{-1} = \left(\begin{array}{c|ccc|ccc} & e^\sigma_1 & \cdots & e^\sigma_r & e^\sigma_{r+1} & \cdots & e^\sigma_n \\ \hline \tilde{e}_1 & 1 & * & * & * & * & * \\ \vdots & & \ddots & * & \vdots & \vdots & \vdots \\ \tilde{e}_r & & & 1 & * & * & * \\ \hline \tilde{e}_{r+1} & & & & 1 & & \\ \vdots & & 0 & & & \ddots & \\ \tilde{e}_n & & & & & & 1 \end{array}\right),$$

$$T^{-1} = \left(\begin{array}{c|ccc|ccc} & \tilde{e}_1 & \cdots & \tilde{e}_r & \tilde{e}_{r+1} & \cdots & \tilde{e}_n \\ \hline \tilde{z}_1 & & & & 1 & & \\ \vdots & & 0 & & & \ddots & \\ \tilde{z}_{n-r} & & & & & & 1 \end{array}\right).$$

Since the lower left block of Q is zero and the lower right block is the identity matrix, we get that $T^{-1}Q^{-1}$ is given by

$$T^{-1}Q^{-1} = \left(\begin{array}{c|ccc|ccc} & \bar{e}_1 & \cdots & \bar{e}_r & \bar{e}_{r+1} & \cdots & \bar{e}_n \\ \hline \tilde{z}_1 & & & & 1 & & \\ \vdots & & 0 & & & \ddots & \\ \tilde{z}_{n-r} & & & & & & 1 \end{array}\right),$$

i.e.

$$T^{-1}Q^{-1}(\sum_{i=1}^{n} c_i e_i^{\sigma}) = \sum_{i=1}^{n-r} c_{i+r}\tilde{z}_i,$$

where $c_i \in \mathbb{F}[t]$ for $i = 1, \ldots, n$. Moreover,

$$T^{-1}Q^{-1}P_{\sigma^{-1}}^{-1}(\sum_{i=1}^{n} c_i e_i^{N}) = T^{-1}Q^{-1}(\sum_{i=1}^{n} c_i e_{\sigma(i)}^{\sigma}) = T^{-1}Q^{-1}(\sum_{i=1}^{n} c_{\sigma^{-1}(i)} e_i^{\sigma})$$
$$= \sum_{i=1}^{n-r} c_{\sigma^{-1}(i+r)}\tilde{z}_i.$$

Let $q_i = \sigma^{-1}(i+r)$ for $i = 1, \ldots, n-r$. Then

$$\sum_{i=1}^{n-r} c_{\sigma^{-1}(i+r)}\tilde{z}_i = \sum_{i=1}^{n-r} c_{q_i}\tilde{z}_i.$$

Let $\omega \in S(n-r)$ be the permutation s.t. $q_{\omega(1)} < \cdots < q_{\omega(n-r)}$. The permutation ω induces a new order, given by $[\tilde{z}_{\omega(i)}]_{i=1}^{n-r}$ and denoted $[z_i]_{i=1}^{n-r}$, from the ordered basis $[\tilde{z}_i]_{i=1}^{n-r}$. It does also induce a permutation matrix $P_\omega : (\ker A, [z_i]_{i=1}^{n-r}) \to (\ker A, [\tilde{z}_i]_{i=1}^{n-r})$ given by

$$P_\omega : z_i \mapsto \tilde{z}_{\omega(i)}$$

for $i = 1, \ldots, n-r$. Note that P_ω^{-1} is given by

$$P_\omega^{-1} : \tilde{z}_i \mapsto z_{\omega^{-1}(i)}$$

for $i = 1, \dots, n-r$. The composition of P_{ω}^{-1} and $T^{-1}Q^{-1}P_{\sigma^{-1}}^{-1}$, denoted by D, gives

$$\begin{aligned} P_{\omega}^{-1}T^{-1}Q^{-1}P_{\sigma^{-1}}^{-1}(\sum_{i=1}^{n} c_i e_i^N) &= P_{\omega}^{-1}(\sum_{i=1}^{n-r} c_{q_i}\tilde{z}_i) \\ &= \sum_{i=1}^{n-r} c_{q_i} z_{\omega^{-1}(i)} = \sum_{i=1}^{n-r} c_{q_{\omega(i)}} z_i, \end{aligned}$$

where $q_{\omega(1)} < \cdots < q_{\omega(n-r)}$ and $c_i \in \mathbb{F}[t]$ for $i = 1, \cdots, n$. Hence, D is a matrix which deletes the rows of a vector that have its index in I_{piv}. Moreover, D represents the identity functions when restricted to the image of B. Hence, it gives a new matrix $\bar{B} = DB$, which also represents f and has $(\ker g, [z_i]_{i=1}^{n-r})$ as codomain. The following diagram gives an overview of how $\bar{B}$ is defined.

$$\begin{array}{ccccccc} (N, [e_i^N]_{i=1}^n) & \xrightarrow{P_{\sigma^{-1}}^{-1}} & (N, [e_i^\sigma]_{i=1}^n) & \xrightarrow{Q^{-1}} & (N, [\tilde{e}_i]_{i=1}^n) & \xrightarrow{T^{-1}} & (\ker g, [\tilde{z}_i]_{i=1}^{n-r}) \\ \uparrow B & & & \searrow^{D} & & & \downarrow P_{\omega}^{-1} \\ (M, [e_j^M]_{j=1}^m) & & & & \xrightarrow{\bar{B}} & & (\ker g, [z_i]_{i=1}^{n-r}) \end{array}$$

Furthermore, $\deg \tilde{e}_i = \deg e_i^\sigma$ for $i = 1, \dots, n$ since Q has degree zero and ones as entities along the diagonal. Hence,

$$\deg z_i = \deg \tilde{z}_{\omega(i)} = \deg \tilde{e}_{\omega(i)+r} = \deg e_{\omega(i)+r}^\sigma = \deg e_{\sigma^{-1}(\omega(i)+r)}^N = \deg e_{q_{\omega(i)}}^N$$

for $i = 1, \dots, n-r$, i.e. the degree of z_i is equal to the degree of the i'th element in $D[e_j^N]_{j=1}^n$. □

This gives us an easier and less computationally expensive way for computing $\bar{M}_{p+1}$. We do not need to remember or calculate V^{-1}, we just need to record which columns become zero columns and which becomes a column with a pivot element, i.e. we need to record the map σ described in Theorem 4.13.

Example 4.14. *In this example we will calculate the persistent homology of the filtered complex shown in Figure 34 by using Theorem 4.13. Recall that M_1 and M_2 are given as follows.*

$$M_1 = \left(\begin{array}{c|ccccc} & ab & bc & cd & ad & ac \\ \hline a & t & 0 & 0 & t^2 & t^3 \\ b & -t & t & 0 & 0 & 0 \\ c & 0 & -1 & t & 0 & -t^2 \\ d & 0 & 0 & -t & -t & 0 \end{array}\right) \quad M_2 = \left(\begin{array}{c|cc} & abc & acd \\ \hline ab & t^3 & 0 \\ bc & t^3 & 0 \\ cd & 0 & t^3 \\ ad & 0 & -t^3 \\ ac & -t & t^2 \end{array}\right)$$

Recall from Example 4.12 that M_1 already has $\ker \partial_0$ *as codomain and that after ordering the codomain and doing column reduction it becomes as follows.*

$$\tilde{M}_1 V_1 = \begin{array}{c|ccccc} & cd & bc & ab & z_1 & z_2 \\ \hline d & -t & 0 & 0 & 0 & 0 \\ c & t & -1 & 0 & 0 & 0 \\ b & 0 & t & -t & 0 & 0 \\ a & 0 & 0 & t & 0 & 0 \end{array}$$

Note that the only performed column swapping is the swapping of column 1 and 2, i.e. $\sigma = (1,2) \in S(5)$. *This σ will be used for constructing $\bar{M}_2$. Reading from our column echelon matrix, we get by Corollary 4.6, that H_0 is given by*

$$\begin{aligned} H_0 &\cong \frac{\mathbb{F}[t]}{(-t)}(-\deg d) \oplus \frac{\mathbb{F}[t]}{(-1)}(-\deg c) \oplus \frac{\mathbb{F}[t]}{(-t)}(-\deg b) \oplus \mathbb{F}[t](-\deg a) \\ &= \frac{\mathbb{F}[t]}{(-t)}(-1) \oplus 0 \oplus \frac{\mathbb{F}[t]}{(-t)}(0) \oplus \mathbb{F}[t](0) \\ &= \frac{\mathbb{F}[t]}{(t)}(-1) \oplus \frac{\mathbb{F}[t]}{(t)} \oplus \mathbb{F}[t]. \end{aligned}$$

Moreover, the collection of $\mathcal{P}$-intervals becomes $\{(0,\infty),(0,1),(1,2)\}$, and the barcode becomes as shown in Figure 35. We have now calculated the barcode for H_0, next up is to calculate the barcode for H_1. For calculating H_1, we first construct $\bar{M}_2$. Since $\tilde{M}_1$ had three pivot elements, we need to remove the corresponding rows from M_2. The three rows to be removed are given by $\sigma^{-1}(1)$, $\sigma^{-1}(2)$, and $\sigma^{-1}(3)$; this gives row 3,2 and 1. The new $\bar{M}_2$ matrix becomes as follows.

$$M_2 = \begin{array}{c|cc} & abc & acd \\ \hline \cancel{ab} & \cancel{t^3} & \cancel{0} \\ \cancel{bc} & \cancel{t^3} & \cancel{0} \\ \cancel{cd} & \cancel{0} & \cancel{t^3} \\ ad & 0 & -t^3 \\ ac & -t & t^2 \end{array} \longrightarrow \bar{M}_2 = \begin{array}{c|cc} & abc & acd \\ \hline z_1 & 0 & -t^3 \\ z_2 & -t & t^2 \end{array}$$

Note that $[z_1, z_2]$ is an ordered basis for $\ker \partial_1$. *Moreover,* $\deg z_1 = \deg ad = 2$ *and* $\deg z_2 = \deg ac = 3$. *By ordering the codomain descending with respect to the degree and then reducing it to column echelon form, we get that*

$$M_2 = \begin{array}{c|cc} & abc & * \\ \hline z_2 & -t & 0 \\ z_1 & 0 & -t^3 \end{array}.$$

Hence, we get that H_1 is given by

$$H_1 \cong \frac{\mathbb{F}[t]}{(-t)}(-\deg z_2) \oplus \frac{\mathbb{F}[t]}{(-t^3)}(-\deg z_1) = \frac{\mathbb{F}[t]}{(t)}(-3) \oplus \frac{\mathbb{F}[t]}{(t^3)}(-2).$$

Moreover, the collection of $\mathcal{P}$-intervals becomes $\{(3,4),(2,5)\}$, and the barcode becomes as shown in Figure 36.

4.8 Final algorithm

The algorithms we have discussed are based on doing Gaussian eliminations on matrices derived from the M_p matrices. In the process we found that a column in M_p either contributes a new cycle, or it stops a cycle from persisting. Following this idea, instead of working with each matrix M_p individually, we can evaluate one and one column (simplex) to see if the simplex contributes a new cycle, in which case we will call it a free column, or if it stops an existing cycle from persisting, in which case we will call it a pivot column.

Each simplex σ is given a unique index i_σ s.t. $\deg\sigma < \deg\sigma' \Rightarrow i_\sigma < i_{\sigma'}$, and $\dim\sigma < \dim\sigma' \Rightarrow i_\sigma < i_{\sigma'}$. The columns (simplices) are then evaluated by increasing order, and the evaluation of a column consists of first zeroing out or deleting the rows where the corresponding basis element corresponds to a pivot column. Thereafter, the algorithm tries to zero out the other elements of the column, starting with the row with the highest index, by adding already existing pivot columns times some coefficient to the column. If the entire column is zeroed out, then it becomes what we call a free column and contributes a new cycle; in addition, we mark the column (simplex) for being free. If there still are some nonzero elements left, let τ be the simplex corresponding to the nonzero row with the highest index. Then the cycle corresponding to τ will stop persisting when our simplex, let us call it σ, is added. This contributes the $\mathcal{P}$-interval $(\deg\tau, \deg\sigma)$ to $\mathcal{S}_{\dim\tau}$. After all the simplices are added, we can get the last $\mathcal{P}$-intervals by finding the simplices that have been marked, but not stopped by some simplex.

A more detailed description of this algorithm is given in Algorithm 4 and Algorithm 5. An illustration of the algorithm applied to the filtered complex in Figure 34 is given in Figure 38.

Algorithm 4 COMPUTEINTERVALS(m^3)

Require: A filtered complex K with m simplices $\{\sigma^j\}_{j=0}^{m-1}$.
Ensure: A collection $\mathcal{S}_k$ of $\mathcal{P}$-intervals for each dimension $k = 0, \ldots, \dim K$.

1: **for** $k = 0, \ldots, \dim K$ **do**
2: $\mathcal{S}_k = \emptyset$;
3: **end for**
4: **for** $j = 0, \ldots, m-1$ **do**
5: $d = \text{REDUCE}(\sigma^j)$;
6: **if** $d = \emptyset$ **then**
7: Mark σ^j;
8: **else**
9: $k = \dim \sigma^j$;
10: $i =$ maxindex d;
11: $\mathcal{S}_k = \mathcal{S}_k \cup \{(\deg \sigma^i, \deg \sigma^j)\}$;
12: Store d in $T[i]$;
13: **end if**
14: **end for**
15: **for** $j = 0, \ldots, m-1$ **do**
16: **if** σ^j is marked and $T[j]$ is empty **then**
17: $k = \dim \sigma^j$;
18: $\mathcal{S}_k = \mathcal{S}_k \cup \{(\deg \sigma^j, \infty)\}$;
19: **end if**
20: **end for**
21: **return** $\{\mathcal{S}_k\}_{k=0}^{\dim K}$;

Algorithm 5 REDUCE(m^2)

Require: A simplex σ.
Ensure: The reduced column corresponding to σ.

1: $k = \dim \sigma$;
2: $d = \partial_k \sigma$;
3: Remove unmarked terms in d;
4: **while** $d \neq \emptyset$ **do**
5: $i =$ maxindex d;
6: **if** $T[i]$ is empty **then**
7: **break**;
8: **end if**
9: Let q be the coefficient of σ^i in $T[i]$;
10: Let b be the coefficient of σ^i in d;
11: $d = d - bq^{-1}T[i]$;
12: **end while**
13: **return** d;

	0	1	2	3	4	5	6	7	8	9	10
	a	b	c	d	ab	bc	cd	ad	ac	abc	acd
a	0	0	0	0	t	0	0	t^2	t^3	0	0
b	0	0	0	0	$-t$	t	0	0	0	0	0
c	0	0	0	0	0	-1	t	0	$-t^2$	0	0
d	0	0	0	0	0	0	$-t$	$-t$	0	0	0
ab	0	0	0	0	0	0	0	0	0	t^3	0
bc	0	0	0	0	0	0	0	0	0	t^3	0
cd	0	0	0	0	0	0	0	0	0	0	t^3
ad	0	0	0	0	0	0	0	0	0	0	$-t^3$
ac	0	0	0	0	0	0	0	0	0	$-t$	t^2
abc	0	0	0	0	0	0	0	0	0	0	0
acd	0	0	0	0	0	0	0	0	0	0	0

Free columns

	0	1	2	3	4	5	6	7	8	9	10
	a	b	c	d	ab	bc	cd	(ad)	(ac)	abc	(acd)
a	0	0	0	0	t	0	0	0	0	0	0
b	0	0	0	0	$-t$	t	0	0	0	0	0
c	0	0	0	0	0	-1	t	0	0	0	0
d	0	0	0	0	0	0	$-t$	0	0	0	0
~~(ab)~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~t^3~~	~~0~~
~~(bc)~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~t^3~~	~~0~~
~~(cd)~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~t^3~~
(ad)	0	0	0	0	0	0	0	0	0	0	$-t^3$
(ac)	0	0	0	0	0	0	0	0	0	$-t$	0
~~(abc)~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~
~~(acd)~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~

Pivot columns

Figure 38: An illustration of Algorithm 4 applied to the filtered complex in Figure 34. Some of the simplices are enclosed in parentheses, that is because they are not valid basis elements when viewing the figure as a matrix. However, both the degree and dimension of the enclosed simplex is equal to that of the correct basis element.

4.9 Examples

In this section we will apply persistent homology on data from some known spaces.

Example 4.15. *Let X be the torus in $\mathbb{R}^3$ with outer radius 2 and inner radius 1, and let S be a sample of 500 randomly selected points from X. By using the ϵ weak witness complex construction with 50 landmark points and varying ϵ, and by applying persistent homology on the resulting filtered complex, we get the barcodes shown in Figure 39. From the barcodes in Figure 40, we clearly see that it fits with the fact that the torus consists of only one connected component. We can also see that there are two bars in dimension 1 that persist longer than the others; this fits with the fact that $\beta_1 = 2$. However, the barcode for dimension 2 is a bit more unclear and does not give a good reflection of the fact that $\beta_2 = 1$.*

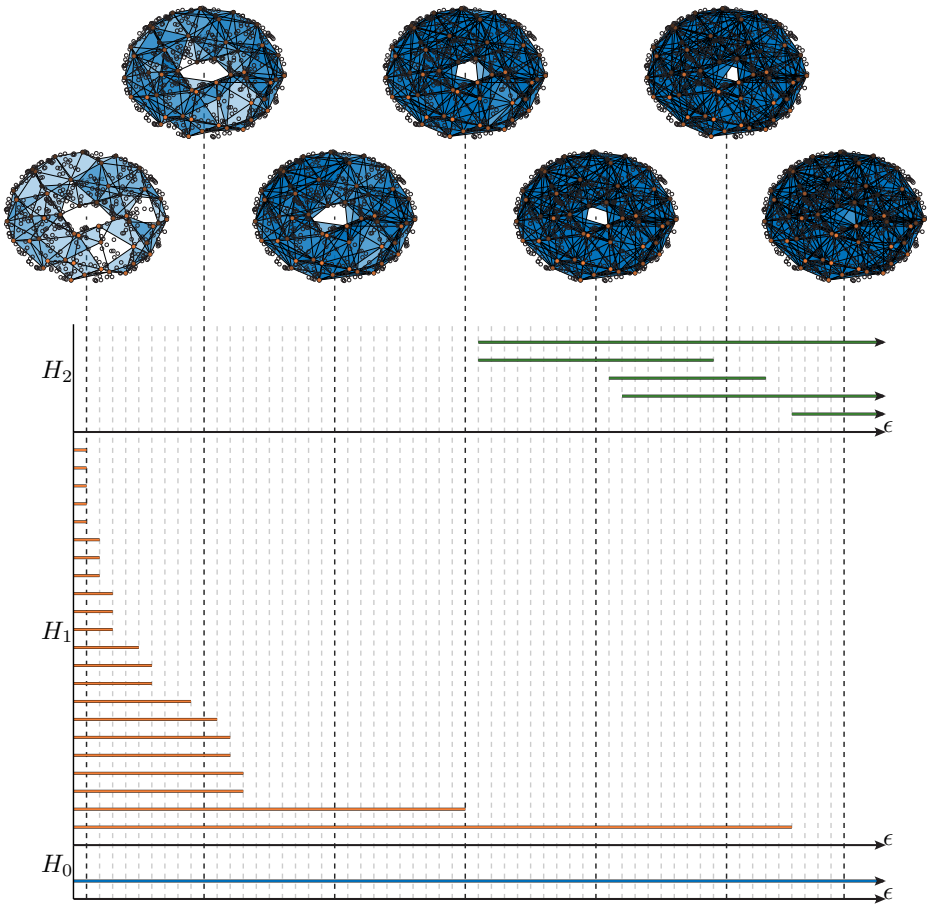


Figure 39: Refer to Example 4.15 for details. Barcodes from applying persistent homology on a filtered complex constructed from 500 points randomly selected from the torus.

Example 4.16. *Let X be the torus in $\mathbb{R}^3$ with outer radius 2 and inner radius 1, and let S be a sample of 4000 randomly selected points from X. By using the ϵ weak witness complex construction with 50 landmark points and varying ϵ, and by applying persistent homology on the resulting filtered complex, we get the barcodes shown in Figure 40. These barcodes fit quite good with the actual Betti numbers of the torus.*

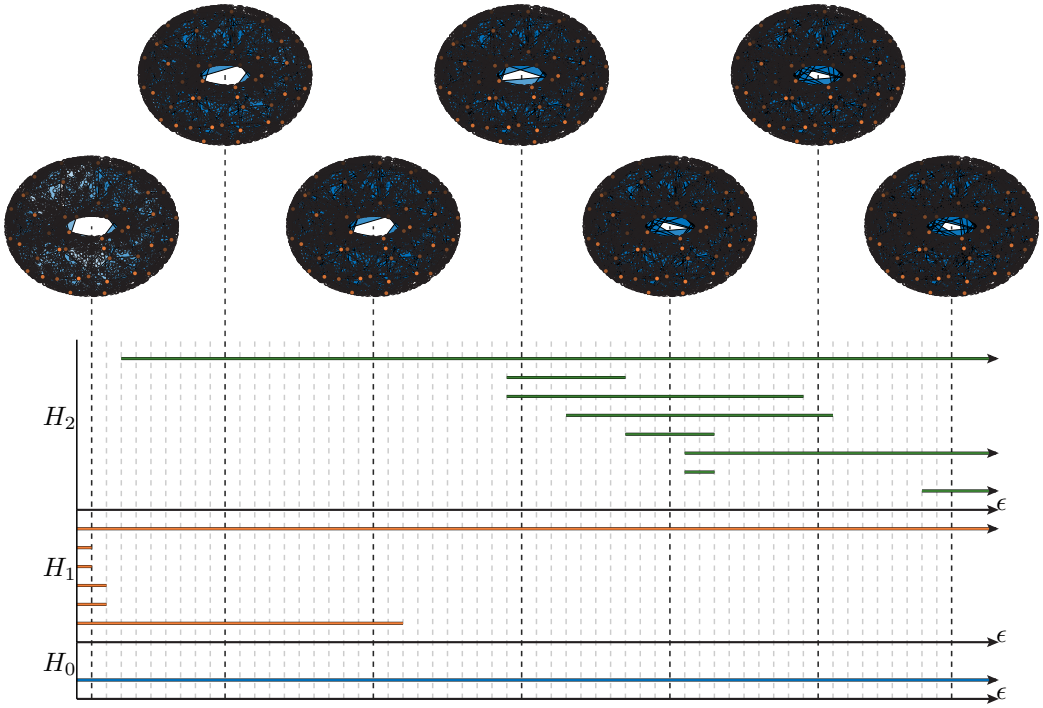


Figure 40: Refer to Example 4.16 for details. Barcodes from applying persistent homology on a filtered complex constructed from 4000 points randomly selected from the torus.

Example 4.17. *Let $X = \partial\bar{B}(0, 0.5) \cup \partial\bar{B}(0, 1.5) \cup \partial\bar{B}(0, 3) \subseteq \mathbb{R}^2$, and let S be 200 randomly selected points from X. Create a filtered complex K^* by using the ϵ weak witness complex construction with 80 landmark points and varying ϵ. By applying persistent homology on K^*, we get the barcodes shown in Figure 41. The barcodes in Figure 41 fits quite good with the fact that the space has three 1-dimensional loops, but it does not give a good representation at dimension zero.*

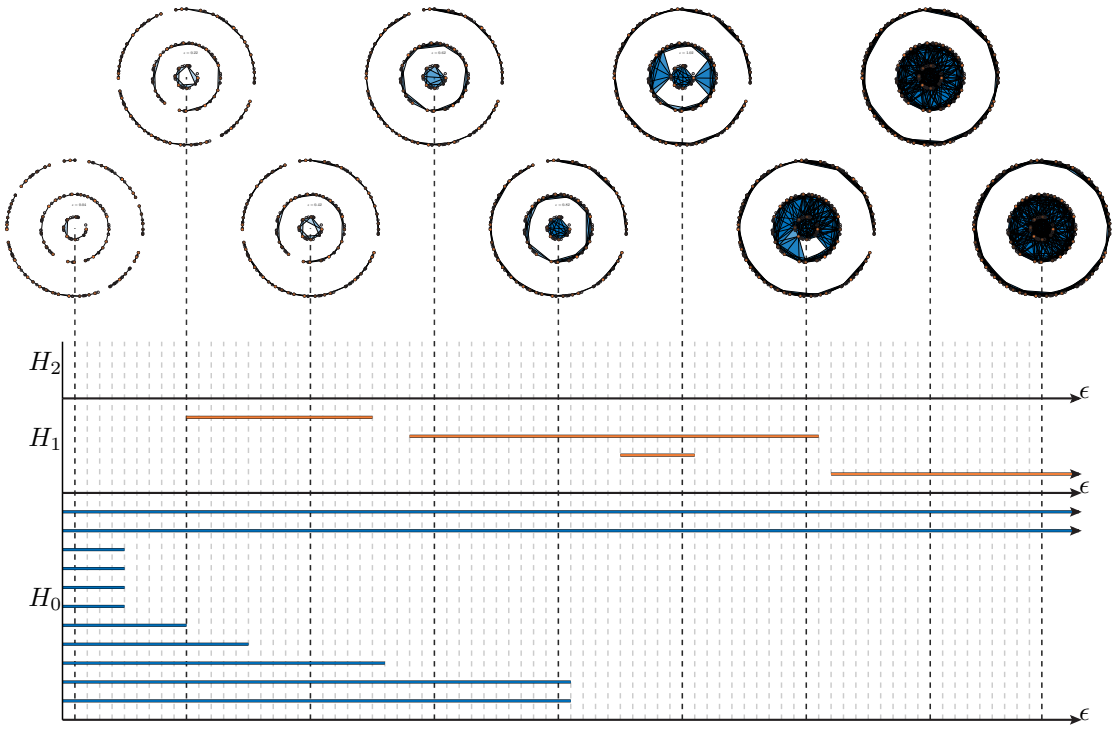


Figure 41: Refer to Example 4.17 for details. Barcodes from applying persistent homology on a filtered complex constructed from 200 points randomly selected from a space consisting of three circles.

5 Mapper

In this section we will introduce the Mapper method. Resources have been gathered from [23] and [3].

5.1 Motivation

In many cases the data coming from modern science and engineering is massive and it is not possible to visualize and recognise structures even in low dimensional projections. In this section we will talk about how high dimensional data sets may be reduced into simplicial complexes, which consists of far fewer points and may be used to capture topological and geometric information in the data.

5.2 Cover manipulation

An efficient method for creating such simplicial complexes is to create a cover of the space and then compute the nerve of the cover. This method is highly dependent on the choice of cover, and a bad choice may give misleading results. One such example is given in Example 5.1.

Example 5.1. *Let* $\mathbb{S}^1$ *be the* 1*-dimensional sphere and let* $\mathcal{U} = \{U_\alpha\}_{\alpha=1}^3$*, where* $U_1 = \{(x,y)|y < -0.3\}$*,* $U_2 = \{(x,y)| -0.6 < y < 0.6\}$*, and* $U_3 = \{(x,y)|0.3 < y\}$ *as shown in Figure 42. With this cover we get the simplicial complex* $S = \{\{1,2\},\{2,3\},\{1\},\{2\},\{3\}\}$ *illustrated in Figure 43, which is not of the same homotopy type as the* 1*-dimensional sphere.*

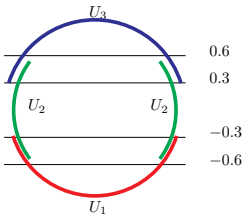


Figure 42: Refer to Example 5.1 for details. This figure illustrates a cover of the unit circle.

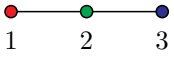


Figure 43: Refer to Example 5.1 for more details. The nerve of the cover displayed in Figure 42.

If we recall the nerve lemma (Lemma 3.4), it states that, for a cover to satisfy the criterions of the nerve lemma (Lemma 3.4), any intersection of elements in the cover should be empty or contractible. Let us say we are given a cover of some topological space X, where the contractability criterion does not hold. One can then ask if it is possible to construct a new cover from the given one such that the criterions hold. One example, of how the cover in Example 5.1 may be altered, is given in Example 5.2.

Example 5.2. *Let $\mathbb{S}^1$ and $\mathcal{U}$ be as in Example 5.1. Note that the open set U_2 consists of two path-connected components. This leads to the fact that U_2, $U_1 \cap U_2$ and $U_3 \cap U_2$ are neither contractible nor empty. If we instead of using U_2, replace it with $U_{2,1}$ and $U_{2,2}$, where $U_{2,1}$ and $U_{2,2}$ are the two path-connected components of U_2. The new cover $\{U_1, U_{2,1}, U_{2,2}, U_3\}$ will satisfy the criterions of the nerve lemma (Lemma 3.4) and gives the simplicial complex illustrated in Figure 44; moreover, it has the correct homotopy type.*

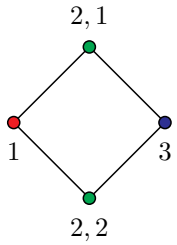


Figure 44: Refer to Example 5.2 for details. The nerve of the cover constructed in Example 5.2.

The method used in Example 5.2 may be generalized as the following algorithm (Algorithm 6). Note that this method only divides the covering elements into path-

Algorithm 6 F_0

Require: A cover $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ of a topological space.
Ensure: A new cover where each component is path-connected.

1: **for** $\alpha \in A$ **do**
2: Let $P_\alpha = \{U_{\alpha,\beta}\}_{\beta \in B_\alpha}$ be the set of path-connected components of U_α;
3: **end for**
4: **return** $\bigcup_{\alpha \in A} P_\alpha$;

connected components. There is no assurance that these components or the intersections of these are empty or contractible. However, for the elements and the intersections to be contractible, they need to be path-connected since contractability implies path-connectedness. Hence, when we are given a cover of the space in question, the method F_0 constructs a new cover that has a higher probability of satisfying the criterions of the

nerve lemma (Lemma 3.4). It is also important to note that even though the elements of the new cover are path-connected, intersections of them need not be. One example, where this is the case, is given in Example 5.3.

Example 5.3. *Let $X = \mathbb{S}^1$, and let $\mathcal{U} = \{U_1, U_2\}$, where $U_1 = \{(x, y)|y < 0.3\}$ and $U_2 = \{(x, y)|y > -0.3\}$, be a cover of X as shown in Figure 45. Then both U_1 and U_2 are path-connected (and even contractible), but $U_1 \cap U_2$ is not. Moreover, the nerve of $\mathcal{U}$ is given by the simplicial complex illustrated in Figure 46.*

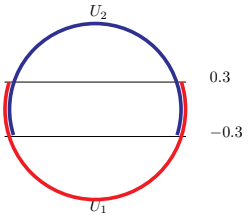


Figure 45: Refer to Example 5.3 for details. This figure illustrates a cover of the unit circle.

Figure 46: Refer to Example 5.3 for details. The nerve of the cover constructed in Example 5.3.

One might ask if there are some methods for modifying the cover such that the new cover also has the property that the intersection of two elements is path-connected. One example, of how this may be done on the cover in Example 5.3, is given in Example 5.4.

Example 5.4. *Let $\mathcal{U}$ and X be as in Example 5.3. Recall that $U_1 \cap U_2$ consists of two path-connected components. Let these two components be called $U_{1,2}^1$ and $U_{1,2}^2$. If we let $\hat{\mathcal{U}}$ be the cover given by $\hat{\mathcal{U}} = \left\{U_1 - U_{1,2}^1, U_1 - U_{1,2}^2, U_2 - U_{1,2}^1, U_2 - U_{1,2}^2\right\}$, as shown in Figure 47, then the intersection of any two elements in $\hat{\mathcal{U}}$ is either empty or path-connected (and even contractible in this example).*

The method used in Example 5.4 may also be generalized as the following algorithm (Algorithm 7). One should note that the cover given by F_1, see Algorithm 7, may have collections of three (or more) elements such that their intersection is neither empty nor path-connected.

When dealing with data sets, we need to remember that the given data S is only a sample from the underlying space X, which we are interested in. The sample space S is usually a finite metric space, and the given cover, which we will work with, is a

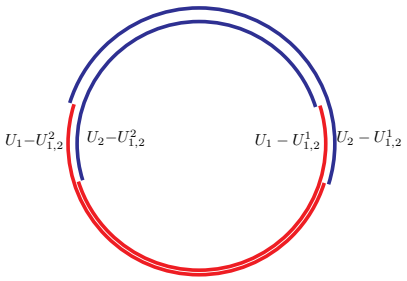


Figure 47: Refer to Example 5.4 for details. This figure illustrates a cover of the unit circle.

Algorithm 7 F_1

Require: A cover $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ of a topological space where each element is path-connected.

Ensure: A new cover where the intersection of any two elements is path-connected.

Search for a pair (U_α, U_β) s.t. $U_\alpha \cap U_\beta$ is non-empty and has more than one path-connected component;
if No such pair exists **then**
    return $\mathcal{U}$;
else
    Let $\left\{U^\gamma_{\alpha,\beta}\right\}_{\gamma \in B}$ be the path-connected components of $U_\alpha \cap U_\beta$;
    for $\gamma \in B$ **do**
        Let $\hat{U}_{\alpha,\gamma} = U_\alpha - U^\gamma_{\alpha,\beta}$ and let $\hat{U}_{\beta,\gamma} = U_\beta - U^\gamma_{\alpha,\beta}$;
    end for
    return $F_1\left((\mathcal{U} - \{U_\alpha\} - \{U_\beta\}) \cup F_0\left(\left\{\hat{U}_{\alpha,\gamma}, \hat{U}_{\beta,\gamma}\right\}_{\gamma \in \beta}\right)\right)$;
end if

cover of S not X. However, the idea is that a cover of S may act like a cover of X, i.e. have similar properties to an actual cover of X. Since the path-connected components of a finite metric space S only consists of the one point sets, using F_0 on a cover of S does not make much sense. To be able to decide if a subset U of a finite metric space S is to be considered as path-connected, we need some notion of connectedness, i.e. some clustering method that devides a given set into one or more disjoint subsets that represent the path-connected components. Hence, when we write $F_0(\mathcal{U})$, where $\mathcal{U}$ is a cover of some finite metric space S, then it is implied that a clustering method is used instead of the topological notion of path-connectedness.

In Example 5.2 and Example 5.4, we saw that the modification of cover algorithms F_0 and F_1 helped to create simplicial complexes that were homotopic to the given space. When dealing with sampled data, this may not always be the case. One example, where method F_0 actually prevents the correct result, is given in Example 5.5.

Example 5.5. *Let $X = \mathbb{S}^1$, and let S be the subset of points with $\mathcal{U} = \{U_1, U_2, U_3\}$ as cover as shown in Figure 48. To be able to consider if the finite subsets U_1, U_2 and U_3 are to be considered as path-connected, we need some notion of connectedness, i.e. some clustering method. In this example we will let two points x and y be contained in the same cluster (subset) if $d(x, y) <= \epsilon$ for some positive real value ϵ. Assume that an ϵ is chosen s.t. U_2 is divided into two clusters $U_{2,1}$ and $U_{2,2}$, as shown in Figure 49, while U_1 and U_3 consists of one cluster each. The nerve of the new cover $\mathcal{N} \circ F_0(\mathcal{U})$ is then given by the simplicial complex shown in Figure 51, while the nerve of $\mathcal{U}$ gives the simplicial complex in Figure 50, which is of the correct homotopy type.*

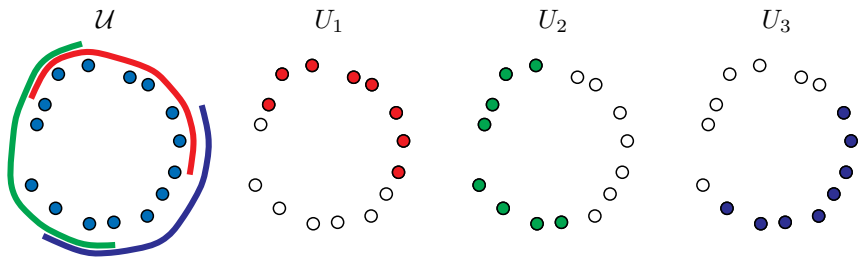


Figure 48: Refer to Example 5.5 for details. This figure illustrates a cover of a data set sampled from the unit circle.

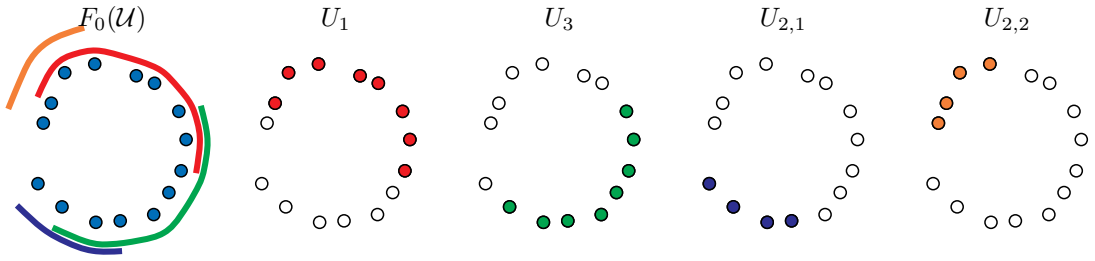


Figure 49: Refer to Example 5.5 for details. This figure illustrates a cover of a data set sampled from the unit circle.

The clustering method used in Example 5.5 is known as single-linkage clustering.

Definition 54. Single-linkage clustering $\hat{\pi}_{\text{SL}}(X, \epsilon)$
Let (X, d) be a metric space, and let $\epsilon \in \mathbb{R}$ be positive. Then the single-linkage clustering with ϵ as tolerance parameter on X is given by

$$\hat{\pi}_{\text{SL}}(X, \epsilon) = X/\sim_\epsilon,$$

where $x \sim_\epsilon y \Leftrightarrow d(x, y) \leq \epsilon$ for $x, y \in X$.

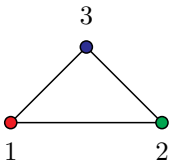


Figure 50: Refer to Example 5.5 for more details. The nerve of the cover displayed in Figure 48.

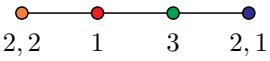


Figure 51: Refer to Example 5.5 for more details. The nerve of the cover displayed in Figure 49.

There are also examples where $\mathcal{N} \circ F_0(\mathcal{U})$ gives extra cycles compared to $\mathcal{N}(\mathcal{U})$ when it shouldn not. A similar example, where F_1 creates artifacts, is given in Example 5.6.

Example 5.6. *Let $X = [0, 2] \times [0, 1] \subseteq \mathbb{R}^2$, let S be some data set sampled from X, and let $\mathcal{U} = \{U_1, U_2\}$ be a cover of S as shown in Figure 52. By using single-linkage clustering with some suitable ϵ, the intersection $U_1 \cap U_2$ will consist of two clusters. Let $U^1_{1,2}$ be the cluster in the upper part of the intersection, and let $U^2_{1,2}$ be the lower one, see Figure 52. The new cover $\hat{\mathcal{U}} = F_1(\mathcal{U})$ is then given by $\hat{\mathcal{U}} = \{\hat{U}_{1,1}, \hat{U}_{1,2}, \hat{U}_{2,1}, \hat{U}_{2,2}\}$, where $\hat{U}_{1,1} = U_1 - U^1_{1,2}, \hat{U}_{1,2} = U_1 - U^2_{1,2}, \hat{U}_{2,1} = U_2 - U^1_{1,2}$, and $\hat{U}_{2,2} = U_2 - U^2_{1,2}$. Taking the nerve of the new cover $\hat{\mathcal{U}}$ gives the simplicial complex shown in Figure 55, while the nerve of $\mathcal{U}$ gives a simplicial complex that is homotopy equivalent to X, see Figure 54*

Note that a cover of S only is a tool for approximating the topology of X. If we only try to construct good covers of S, we might just end up with the cover where each open set consists of one point, which most likely is not a fitting approximation of a cover of X. To be more precise we will define what we mean with a fitting approximation of a cover.

Definition 55. Fitting cover approximation
Let $\mathcal{U}_S = \{U_\alpha\}_{\alpha \in A}$ be a cover of $S \subseteq X$, and let $\mathcal{V}_X = \{V_\beta\}_{\beta \in B}$ be a cover of X, where X is some topological space. Then $\mathcal{U}_S$ is a fitting cover approximation of $\mathcal{V}_X$ if there $\exists$ a bijection $\phi : A \to B$ s.t.

1. $U_\alpha \subseteq V_{\phi(\alpha)}$ $\forall \alpha \in A$ and
2. $\bigcap_{\alpha \in \tau} U_\alpha \neq \emptyset \Leftrightarrow \bigcap_{\alpha \in \tau} V_\phi \neq \emptyset$ $\forall \tau \subseteq A$.

Straight from the definition of a fitting approximation cover, we get Property 5.7.

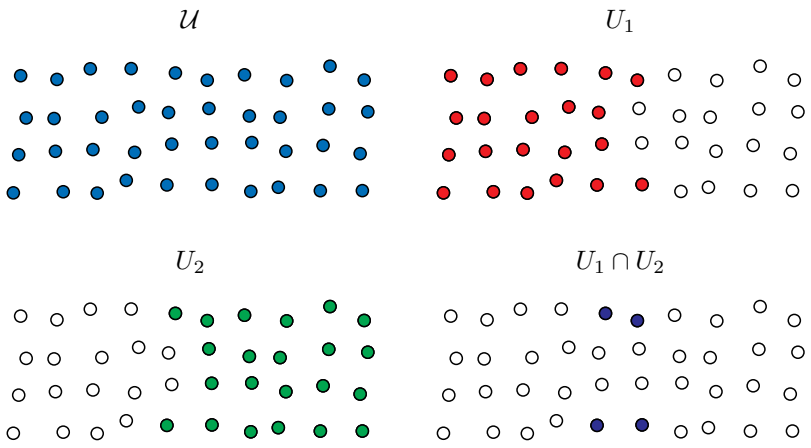


Figure 52: Refer to Example 5.6 for details. This figure illustrates a cover of a sampled data set.

Proposition 5.7. *Let X be some topological space, and let $\mathcal{U}$ be a cover of some sampled data $S \subseteq X$. If there $\exists$ a cover $\mathcal{V}$ of X s.t. $\mathcal{U}$ is a fitting approximation cover of $\mathcal{V}$, then*

$$\mathcal{N}(\mathcal{U}) \cong \mathcal{N}(\mathcal{V}),$$

i.e. they are isomorphic as abstract simplicial complexes.

Note that because of noise, the sampled data will often not be a subset of X. However, if we assume that X is a subspace of some bounded metric space (Z, d) s.t. $S \subseteq Z$, then

$$S \subseteq \bigcup_{x \in X} B(x, \epsilon) = B(X, \epsilon) \subseteq Z \tag{2}$$

for some positive value $\epsilon \in \mathbb{R}$, which depends on how noisy the sampled data is. With good measurements this ϵ will be small; moreover, $B(X, \epsilon)$ will in many cases be homotopic to X. One example, where there exists an ϵ s.t. $S \subseteq B(X, \epsilon)$ and $B(X, \epsilon)$ is homotopic to X, is given in Example 5.8.

Example 5.8. *Let $X = \mathbb{S}^1$, and let S be the sampled data as in Example 5.5. Then we can let $X \subseteq \mathbb{R}^2$, and pick a positive $\epsilon \in \mathbb{R}$ s.t. $B(X, \epsilon) \subseteq \mathbb{R}^2$ is homotopic to X and $S \subseteq B(X, \epsilon)$, as shown in Figure 56. Moreover, if we let $\mathcal{U}$ be the cover of S as in Example 5.5 and illustrated once more in Figure 57, then we can find a cover $\mathcal{V}$ of $B(X, \epsilon)$, as shown in Figure 58, s.t. S is a fitting cover approximation of $\mathcal{V}$. Note that our cover $\mathcal{V}$ is in fact a good cover of $B(X, \epsilon)$. Hence, by the nerve lemma (Lemma 3.4), Property 5.7, and the fact that $B(X, \epsilon)$ and X are homotopic gives*

$$\mathcal{N}(\mathcal{U}) \cong \mathcal{N}(\mathcal{V}) \simeq B(X, \epsilon) \simeq X,$$

i.e. $\mathcal{N}(\mathcal{U})$ and X are homotopic.

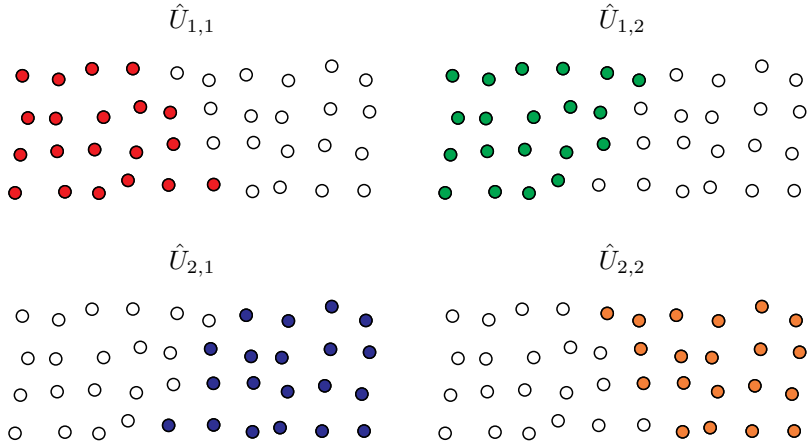


Figure 53: Refer to Example 5.6 for details. This figure illustrates a cover of a sampled data set.

Figure 54: Refer to Example 5.6 for more details. The nerve of the cover displayed in Figure 52.

It may be wise to use methods such as F_0 and F_1 together with a clustering method to try and make covers that are fitting cover approximations of some good cover of X or a blown up version of X. However, we can not rely too much on the clustering methods as they might lead to false judgements. In practice applying F_0 on a cover has shown to be a useful tool, and F_0 is used by our Mapper method. Moreover, most clustering algorithms use some positive real valued ϵ to determine the clusters, and by increasing ϵ the method F_0 becomes the identity function on covers, i.e. it does not alter the cover. I have not tested the F_1 method, but with this method it becomes even more difficult to determine path-connectedness since the intersection of two open sets will contain even fewer data points.

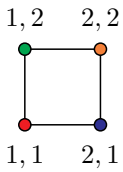


Figure 55: Refer to Example 5.6 for more details. The nerve of the cover displayed in Figure 53.

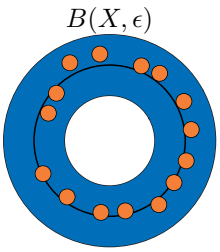


Figure 56: Refer to Example 5.8 for details. This figure illustrates the blown up space $B(X, \epsilon)$ of X, where the small dots are points in S and the black circle in the center is X.

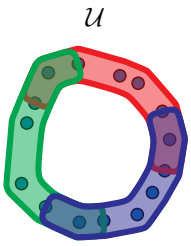


Figure 57: Refer to Example 5.8 for details. This figure illustrates a cover of a data set sampled from the unit circle.

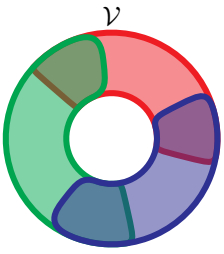


Figure 58: Refer to Example 5.8 for details. This figure illustrates a cover of $B(X, \epsilon)$.

5.3 Filter function

In this section we will talk about how we can construct covers of a given sampled data set S by using what we will call a filter function.

The idea comes from the fact that if X and Z are two topological spaces, $f : X \to Z$ is a continuous function, and $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ is an open cover of Z, then $f^{-1}(\mathcal{U}) = \left\{f^{-1}(U_\alpha)\right\}_{\alpha \in A}$ is an open cover of X. When dealing with data sets, the space X will be the sampled data while Z, which we will call the parameter space, will be some space determined by the image of f. The parameter space is usually some subspace of $\mathbb{R}^n$ or $\mathbb{S}^n$, where n is some positive integer. There are also different choices of covers for each of these parameter spaces; we will come back to those later. The filter function will just be some continuous like function that assigns a value to each data point. The freedom to choose such a filter function gives us many possibilities for constructing different covers of our data set. We will now take a look at three families of filter functions.

5.3.1 Density estimators

A density estimator is a possible filter function. Density estimators are well known in the area of statistics. One useful density estimator is the Gaussian kernel given by

$$f_\epsilon = C_\epsilon \sum_{y \in X} \exp\left(\frac{-d(x,y)^2}{\epsilon}\right),$$

where ϵ is a positive real number and C_ϵ is a constant s.t. $\int_X f_\epsilon(x) d_x = 1$.

5.3.2 Eccentricity

Another family of useful functions is the family of eccentricity functions given by

$$E_p(x) = \left(\frac{1}{N} \sum_{y \in X} d(x,y)^p\right)^{\frac{1}{p}},$$

where N is the number of elements in X, and p is some positive integer. This may also be extended to $p = +\infty$ by letting $E_\infty(x) = \max_{y \in X} d(x,y)$.

5.3.3 Projection maps

When dealing with objects in a low-dimensional space the projection maps such as $(x, y) \mapsto y$ are easy to use and will often do the job.

5.3.4 Filter functions applied to data sets

Example 5.9. *Let X be the space consisting of the pixels in Figure 59. Then E_2 evaluated on X gives the result shown in Figure 61 while the projection map $(x, y) \mapsto -y$ gives the result shown in Figure 60.*

Example 5.10. *Let X be the space consisting of the pixels in Figure 62. Then the density estimator, with some values for ϵ and C_ϵ, evaluated on X gives the result shown in Figure 63.*

Figure 59: Refer to Example 5.9 for details. This figure shows an image obtained by scanning a hand drawn figure.

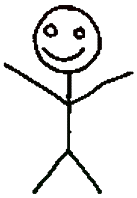


Figure 60: Refer to Example 5.9 for details. This figure illustrates the use of a height function as filter on the data shown in Figure 59. The color red indicates a low value while green indicates high values.

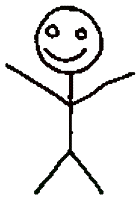


Figure 61: Refer to Example 5.9 for details. This figure illustrates the use of a eccentricity function as filter on the data shown in Figure 59. The color red indicates a low value while green indicates high values.

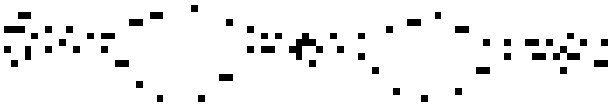


Figure 62: Refer to Example 5.10 for details. This figure illustrates some sampled data.

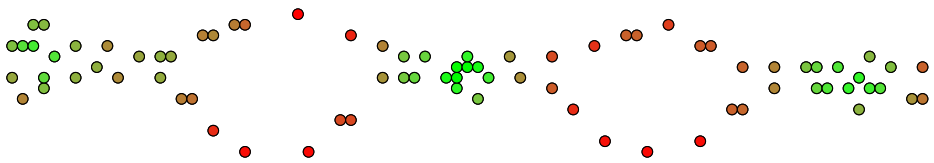


Figure 63: Refer to Example 5.10 for details. This figure illustrates the use of a density estimator as filter on the data shown in Figure 62. The color red indicates a low value while green indicates high values.

5.4 Cover of parameter space

The filter functions are used to pull back a cover of the parameter space into a cover of the sampled data X. Some typical examples of covers of $\mathbb{R}$ and $\mathbb{S}^1$ are given in Example 5.11 and Example 5.12.

Example 5.11. *Let $X = \mathbb{R}$, and let l and ϵ be two positive real numbers. Let $U_k = (kl-\epsilon, (k+1)l+\epsilon)$ for $k \in \mathbb{Z}$, then the family of these sets form an open cover $\mathcal{U} = \{U_k\}_{k\in\mathbb{Z}}$ of X. Note that when $\epsilon < l/2$, the intersection $U_k \cap U_n = \emptyset$ for $|k-n| > 1$; hence, the intersection of any three distinct sets is empty.*

Example 5.12. *Let $X = \mathbb{S}^1$, let $n \in N$ be greater then 2, and let ϵ be a positive real number. Let $U_k = \{(cos(\theta), sin(\theta))\}_{\theta\in(2\pi k/N-\epsilon, 2\pi k/N+\epsilon)}$ for $k = 1, \ldots, N$. Then $\mathcal{U} = \{U_k\}_{k=1}^N$ is an open covering of X when $\epsilon > \pi/N$. Note that the intersection of three distinct sets is empty when $\epsilon < 2\pi/N$.*

Some examples, where coverings of the parameter space are pulled back, are given in Example 5.13 and Example 5.14.

Example 5.13. *Let X be the sampled data consisting of the pixels in Figure 59, and let $\mathcal{U}$ be a cover of $\mathbb{R}$ as shown in Figure 64. Then the cover pulled back by using the eccentricity function E_2 becomes as shown in Figure 65.*

Example 5.14. *Let X be the sampled data consisting of the pixels in Figure 59, and let $\mathcal{U}$ be a cover of $\mathbb{R}$ as shown in Figure 66. Then the cover pulled back by using the projection map $(x, y) \mapsto -y$ becomes as shown in Figure 67.*

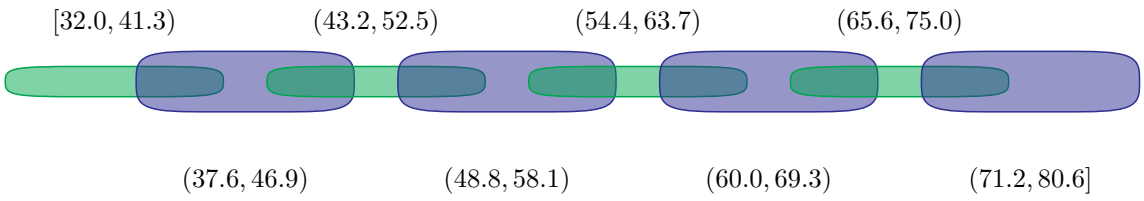


Figure 64: Refer to Example 5.13 for details. This figure illustrates a cover on the image of the eccentricity function E_2.

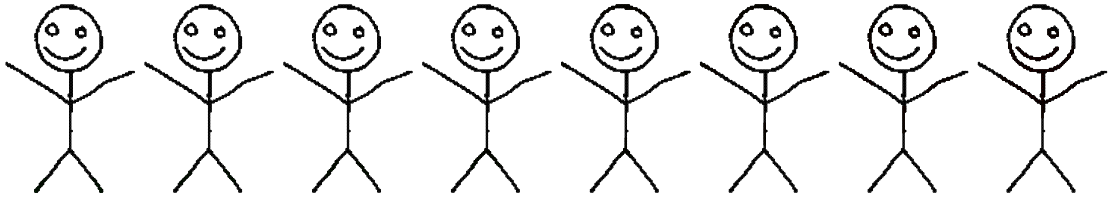


Figure 65: Refer to Example 5.13 for details. This figure illustrates a cover pulled back by the eccentricity function E_2.

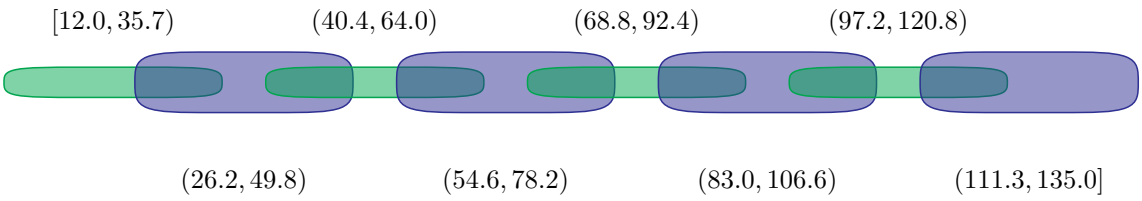


Figure 66: Refer to Example 5.14 for details. This figure illustrates a cover on the image of the projection map $(x, y) \mapsto -y$.

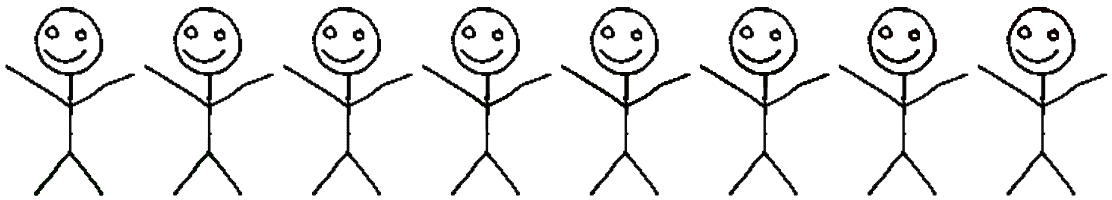


Figure 67: Refer to Example 5.14 for details. This figure illustrates a cover pulled back by the projection map $(x, y) \mapsto -y$.

A useful property is that when a covering $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ of a parameter space has the property that any intersection $U_{\alpha_1} \cap \cdots \cap U_{\alpha_n}$ of n distinct elements in $\mathcal{U}$ is empty $\forall n \geq N$ for a natural number N, then the same holds for the cover pulled back by the filter function.

Proposition 5.15. *Let X and Z be two topological spaces, let $f : X \to Z$ be a continuous function, let $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ be a cover of Z s.t. $U_{\alpha_1} \cap \cdots \cap U_{\alpha_n} = \emptyset$ for any intersection of $n \geq N$ distinct elements for some natural number N, and let $\mathcal{V} = f^{-1}(\mathcal{U}) = \{f^{-1}(U_\alpha)\}_{\alpha \in A}$ be the cover pulled back by f. Then $V_{\alpha_1} \cap \cdots \cap V_{\alpha_n} = \emptyset$ $\forall n \geq N$ and distinct $\alpha_1, \ldots, \alpha_n \in A$.*

Proof. Assume $V_{\alpha_1} \cap \cdots \cap V_{\alpha_n} \neq \emptyset$ for some distinct elements $V_{\alpha_1}, \ldots, V_{\alpha_n}$, where $n \geq N$. Then there $\exists x \in X$ s.t. $x \in V_{\alpha_1} \cap \cdots \cap V_{\alpha_n}$, i.e. $f(x) \in f(V_{\alpha_i}) \subseteq U_{\alpha_i}$ for $i = 1, \ldots, n$. Then $f(x) \in U_{\alpha_1} \cap \cdots \cap U_{\alpha_n}$, which gives a contradiction. Hence, $V_{\alpha_1} \cap \cdots \cap V_{\alpha_n}$ must be empty. □

5.5 Clustering

Clustering is a common technique in statistical data analysis. In Section 5.1 we showed how the single-linkage clustering method may be used, and we talked about what role the chosen clustering method plays.

There are many other clustering methods other than single-linkage clustering that may be used instead, but we will not talk about those. An important property for a clustering method is to be functorial.

Definition 56. Functorial clustering method
A clustering method is said to be a functorial clustering method if whenever one have an inclusion $X \hookrightarrow Y$ of points, i.e. a set map preserving distances, the image of each cluster in X may be included in one of the clusters in Y.

Note that this implies that each cluster of X may be included in exactly one of the clusters of Y. This will be an important property in Section 5.9, where we want to compare the results obtained by using two different covers on the parameter space. An example of single-linkage clustering applied on a data set is given in Example 5.16.

Example 5.16. *Figure 68 shows the results from applying single-linkage clustering on some sampled data in $\mathbb{R}^2$.*

5.6 Algorithm

Now that we have covered the essential material of the Mapper method, we may give the Mapper algorithm (Algorithm 8).

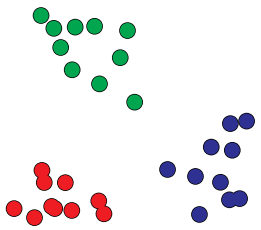


Figure 68: This figure shows the results from applying single-linkage clustering on a data set in $\mathbb{R}^2$. Each dot is a data point, and each color correspond to a cluster.

Algorithm 8 Mapper

Require: Some sampled data X.
Ensure: A simplicial complex.

1: Pick a filter function f defined on X;
2: Pick a cover $\mathcal{U}$ on $\operatorname{im} f$;
3: Pick a clustering method to be used by F_0;
4: If needed pick a metric to be used by the clustering method;
5: **return** $\mathcal{N} \circ F_0 \circ f^{-1}(\mathcal{U})$;

5.7 Examples

In this section we will look at some examples, where Mapper is applied to some data set. The following examples have been created by using an implementation of Mapper, which I wrote in python. Note that the output will have some extra information added to the simplicial complex. Each node will have a color, indicating the mean value of the filter function applied on the points in the corresponding cluster. The size of the node will correspond to the number of elements in the cluster. The positions of the nodes are calculated by neato, a Graphviz layout program, see [10]. Moreover, note that all the examples will be using the single-linkage clustering method.

Example 5.17. *In this example we will apply Mapper on the scanned hand drawn figures in Figure 69. The filter function will be the height function. For each figure, we will divide the image of the filter function into* 64 *equally long intervals with* 40% *overlap, and use those as a cover of the parameter space. The metric used by the single-linkage clustering will be the standard euclidean metric. With these settings together with a suitable clustering epsilon, we get the simplicial complexes shown in Figure 70.*

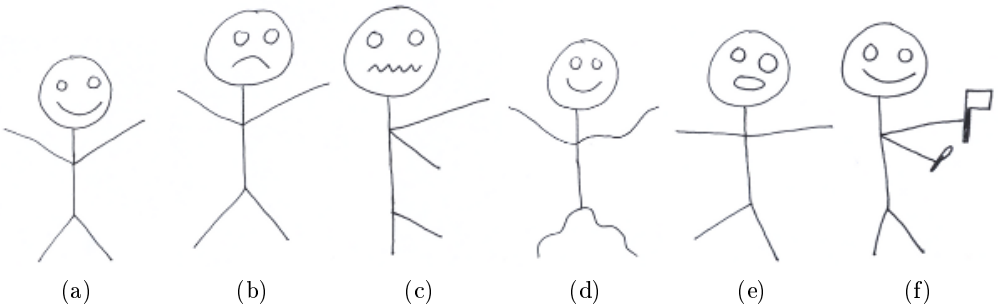


Figure 69: Input data for Example 5.17.

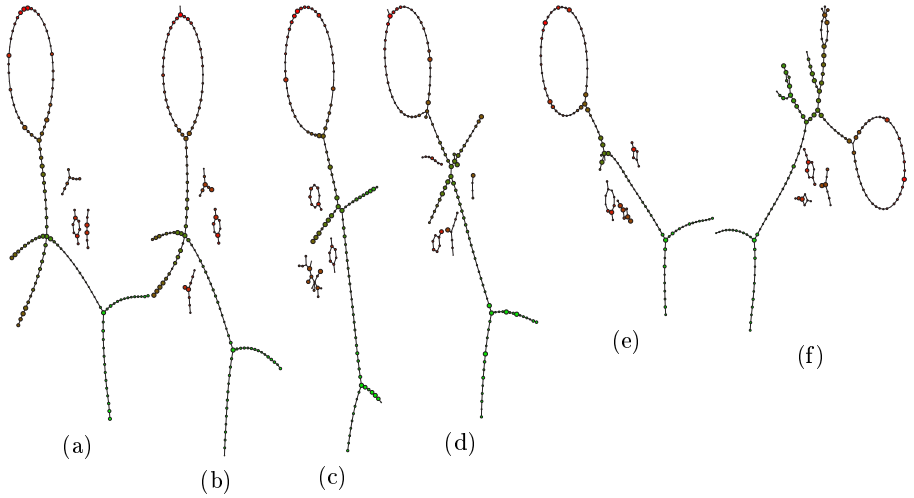


Figure 70: Refer to Example 5.17 for details. Results from applying Mapper on the images shown in Figure 69.

Example 5.18. *In this example we will apply Mapper on the image shown in Figure 71 with three different resolutions on the covers of the parameter space. The filter function will be the eccentricity function* E_2*. For each resolution we will divide the image of the filter function into equally long intervals with* 40% *overlap, but we will vary the number of intervals. The coarsest cover will consist of* 3 *intervals, the second will have* 6*, while the finest cover will consist of* 12 *intervals. The metric used by the single-linkage clustering will be the standard euclidean metric. With these settings together with a suitable clustering epsilon, we get the simplicial complexes shown in Figure 72, where the figure to the left is the coarsest and the one to the right is the finest.*

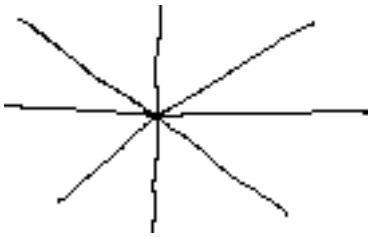


Figure 71: Input data for Example 5.18.

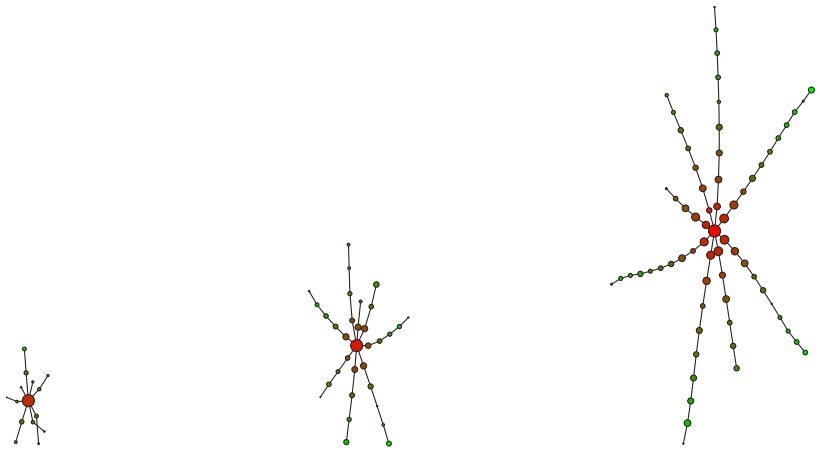


Figure 72: Refer to Example 5.18 for details. Results from applying Mapper, with 3 different resolutions on the cover of the parameter space, on the image shown in Figure 71.

Example 5.19. *In this example we will apply Mapper on the image shown in Figure 73 with two different resolutions. The filter function will be the projection map* $(x, y) \mapsto x$. *The values of our filter function applied to our data is shown in Figure 74. For each resolution, we will divide the image of the filter function into equally long intervals with* 40% *overlap, but we will vary the number of intervals. In addition to alter the number of intervals on the covers, we will use a different clustering epsilon for each resolution. The coarsest resolution will use* 8 *intervals and clustering epsilon* $\epsilon_1 = 10$, *while the finest resolutions will use* 30 *intervals and clustering epsilon* $\epsilon_2 = 4$. *With these settings we get the simplicial complexes shown in Figure 75, where Figure 75a has the coarsest resolution and Figure 75b has the finest.*

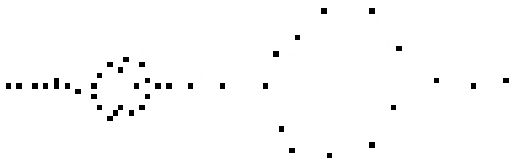


Figure 73: Input data for Example 5.19.

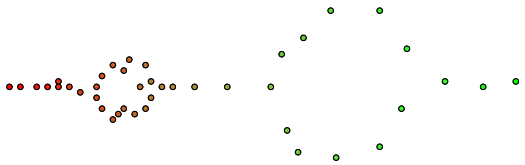


Figure 74: Refer to Example 5.19 for details. This figure shows the values of the filter function applied to the data shown in Figure 73.

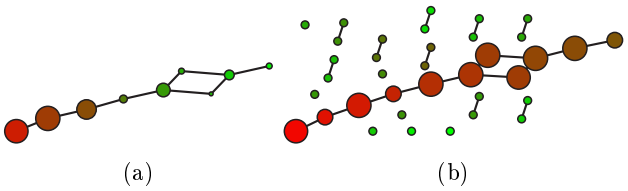


Figure 75: Refer to Example 5.19 for details. Results from applying Mapper with two different resolutions on the data set shown in Figure 73.

Example 5.20. *In this example we will apply Mapper on a data set consisting of squares of various size, color, and rotation. All three variables will be measured as numbers and a configuration will be a point in* $\mathbb{R}^3$*. The collection of squares is shown in Figure 76. Note that the three variables may not vary freely, i.e. there is some system in the configurations space. Let the filter function be the eccentricity function* E_2*, and divide the image of the filter function into* 10 *equally long intervals with* 40% *overlap. Let the metric be the* L_1 *metric, i.e.* $d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$*, where* $n = 3$*. With these settings together with a suitable clustering epsilon, we get the simplicial complex shown in Figure 77. In addition to the usual output, Figure 77 also display some of the squares in the data set together with lines pointing to the clusters that contain the square. Finding the pattern of the configuration space is left as an exercise; using the results, shown in Figure 77, from applying Mapper on the data set may be of help.*

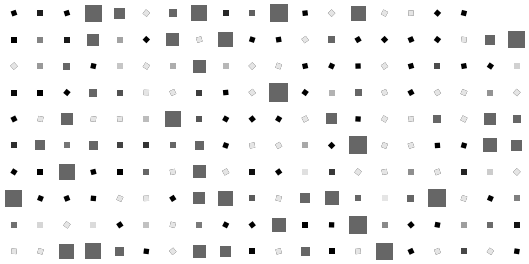


Figure 76: Refer to Example 5.20 for details. Collection of squares of various size, color, and rotation. Used as input data in Example 5.20.

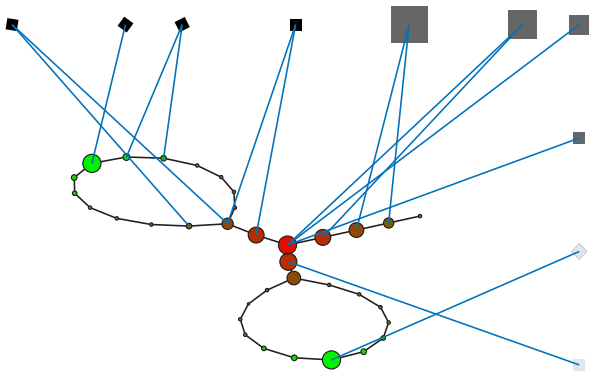


Figure 77: Refer to Example 5.20 for details. Results from applying Mapper on the data set shown in Figure 76. In addition to the usual output, this figure also display some of the squares from the input data set together with lines pointing to the clusters which contain the square.

Example 5.21. *In this example we will apply Mapper on the scanned hand drawn figure in Figure 69e. The filter function will be the identity map on* $\mathbb{R}^2$*. The parameter space will now be a subspace of a two dimensional (filled) rectangle. To make a cover of the rectangle, we first construct covers on the images of* $(x, y) \mapsto x$ *and* $(x, y) \mapsto -y$*, and then let the product of these covers be the cover of our rectangle. The cover on the image of* $(x, y) \mapsto -y$ *will have* 64 *elements while the cover on the image of* $(x, y) \mapsto x$ *will have* 20*. Both of them will have intervals overlapping* 40%*. This gives* 64×20 *elements in the cover of our parameter space. The metric used by the single-linkage clustering will be the standard euclidean metric. With these settings together with a suitable clustering epsilon, we get the simplicial complex shown in Figure 78. Note that the colors are given by the height function.*

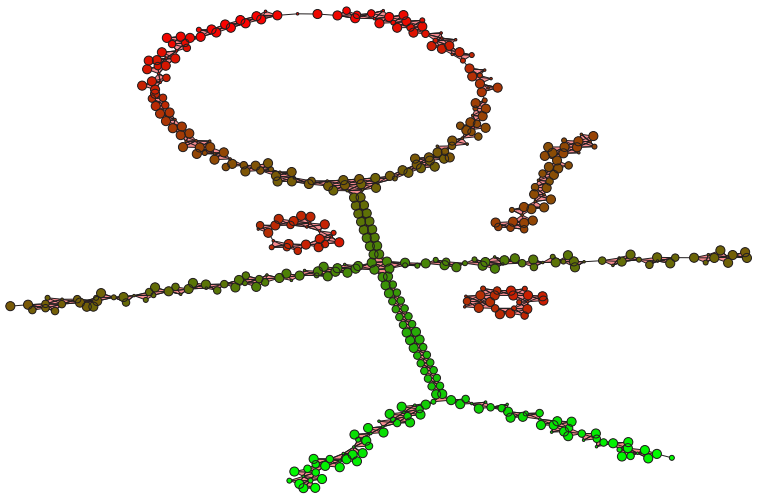


Figure 78: Refer to Example 5.21 for details. Results from applying Mapper on the data set shown in Figure 69e.

5.8 Parameters

The mapper method requires the user to give the following as input:

1. A finitely sampled data set X.
2. A filter function from X in to some parameter space Z.
3. A covering of the parameter space.
4. A clustering method to be used on subsets of X.

In addition, the choice of covering on the parameter space usually consists of choosing some values determining the length (volume) and the overlapping of the sets. The clustering method may also require a metric and a tolerance value to determine if points are to be interpreted as being close or not.

5.9 Map of coverings

In some of the examples in Section 5.7, we saw that the cover on the parameter space may be too coarse to recover significant details of the data set. It is also possible to make a cover of the parameter space that is too fine such that the resulting complex contains lots of artifacts. It is therefore useful to be able to look at different resolutions. When looking at the resulting complexes with different resolutions, it is useful to have some information connecting them, such as a simplicial map.

Definition 57. Simplicial map
Let K and L be two simplicial complexes. Then a simplicial map from K to L is a continuous map $f : |K| \to |L|$ s.t. if $\sigma = \{v_0, \dots, v_n\} \in K$, then $f(\sigma) = \{f(v_0), \dots, f(v_n)\} \in L$.

We will also use the definition of the vertex map together with Proposition 5.22 in this section.

Definition 58. Vertex map
Let K and L be two simplicial complexes. Then a vertex map from K to L is a map $\phi : \text{vert}(K) \to \text{vert}(L)$ s.t. if $\sigma = \{v_0, \dots, v_n\} \in K$, then $\phi(\sigma) = \{\phi(v_0), \dots, \phi(v_n)\} \in L$.

Proposition 5.22. *Let K and L be two simplicial complexes, let ϕ be a vertex map from K to L, and let $\text{vert}(K) = \{v_i\}_{i \in A}$. Let g be the extended map $g : |K| \to |L|$ of ϕ s.t. if $x = \sum_{i=0}^{n} t_i v_i$, where $\{v_0, \dots, v_n\} \in K$ and $t_i \in \mathbb{R}$ for $i = 0, \dots, n$, then $g(x) = \sum_{i=0}^{n} t_i \phi(v_i)$. Then g is a simplicial map from K to L.*

A map of coverings is defined as follows.

Definition 59. Map of coverings
Let $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ and $\mathcal{V} = \{V_\beta\}_{\beta \in B}$ be open covers of some topological space X. Then $\mathcal{U}$ is a refinement of $\mathcal{V}$ if $\forall \alpha \in A$ there $\exists \beta \in B$ s.t. $U_\alpha \subseteq V_\beta$, i.e. there $\exists$ a map $\phi : A \longrightarrow B$ called a map of coverings, also known as refinement map, s.t. $U_\alpha \subseteq V_{\phi(\alpha)}$.

Notation. *We will write $\mathcal{U} \leq \mathcal{V}$ when $\mathcal{U}$ is a refinement of $\mathcal{V}$.*

We will now look at how we can make simplicial complexes with different resolutions and have simplicial maps connecting them.

Assume we have two covers $\mathcal{U}$ and $\mathcal{U}'$, on our sampled space, and want to make a simplicial map from $\mathcal{N}(\mathcal{U})$ to $\mathcal{N}(\mathcal{U}')$. If we have a map of coverings from $\mathcal{U}$ to $\mathcal{U}'$, then Theorem 5.23 gives us a simplicial map from $\mathcal{N}(\mathcal{U})$ to $\mathcal{N}(\mathcal{U}')$.

Theorem 5.23. *Let X be a topological space, let $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ and $\mathcal{U}' = \{U'_\beta\}_{\beta \in B}$ be two covers of X, and let $\phi : A \to B$ be a map of coverings. Then ϕ induces a simplicial map from $\mathcal{N}(\mathcal{U})$ to $\mathcal{N}(\mathcal{U}')$.*

Proof. What we need to do is to use the map of coverings ϕ to induce a vertex map. This will then, by Proposition 5.22, give a simplicial map. Let $K = \mathcal{N}(\mathcal{U})$, and let $L = \mathcal{N}(\mathcal{U}')$. Note that the vertex sets of K and L are the sets A and B, i.e. $\text{vert}(K) = A$ and $\text{vert}(L) = B$. If $\sigma = \{\alpha_0, \ldots, \alpha_n\} \in K$, then $U_{\alpha_0} \cap \cdots \cap U_{\alpha_n} \neq \emptyset$, and since $U_\alpha \subseteq V_{\phi(\alpha)}$ for each $\alpha = \alpha_0, \ldots, \alpha_n$, we get that $V_{\phi(\alpha_0)} \cap \cdots \cap V_{\phi(\alpha_n)} \neq \emptyset$, i.e. $\phi(\sigma) = \{\phi(\alpha_0), \ldots, \phi(\alpha_n)\} \in L$. Hence, ϕ is an vertex map, and by Proposition 5.22 it induces a simplicial map. □

In the Mapper method the resulting complexes are $\mathcal{N} \circ F_0(\mathcal{U})$ and $\mathcal{N} \circ F_0(\mathcal{U}')$; hence, we need a map of coverings from $F_0(\mathcal{U}) \to F_0(\mathcal{U}')$ to be able to use Theorem 5.23. However, if we have a covering from $\mathcal{U}$ to $\mathcal{U}'$ and F_0 uses a functorial clustering method, then Theorem 5.24 gives a map of coverings $F_0(\mathcal{U}) \to F_0(\mathcal{U}')$.

Theorem 5.24. *Let $\hat{\pi}_0$ be the functorial clustering method used by F_0, let $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ and $\mathcal{U}' = \{U'_\beta\}_{\beta \in B}$ be two covers on a topological space, and let $\phi : A \to B$ be a map of coverings. Then ϕ induces a map of coverings from $F_0(\mathcal{U})$ to $F_0(\mathcal{U}')$.*

Proof. Let $\hat{\pi}_0(U_\alpha) = \{U_{\alpha,\gamma}\}_{\gamma \in G_\alpha}$, and let $\hat{\pi}_0(U'_\beta) = \{U'_{\beta,\eta}\}_{\eta \in G'_\beta}$ $\forall \alpha \in A$ and $\forall \beta \in B$. Then by the functionality property of $\hat{\pi}_0$, there exists a (unique) element $U'_{\phi(\alpha),\eta} \in \hat{\pi}_0(U'_{\phi(\alpha)})$ for each $U_{\alpha,\gamma} \in \hat{\pi}_0(U_\alpha)$ s.t. $U_{\alpha,\gamma} \subseteq U'_{\phi(\alpha),\eta}$ since $U_\alpha \subseteq U'_{\phi(\alpha)}$. This gives a map $(\alpha, \gamma) \mapsto (\phi(\alpha), \eta)$, which then gives a map of coverings from $F_0(\mathcal{U})$ to $F_0(\mathcal{U}')$. □

Hence, by Theorem 5.23 and Theorem 5.24, we can induce a simplicial map

$$\mathcal{N} \circ F_0(\mathcal{U}) \longrightarrow \mathcal{N} \circ F_0(\mathcal{U}')$$

from a map of coverings $\mathcal{U} \to \mathcal{U}'$. A map of coverings does not always exist between two arbitrary covers, but we may pick covers $\mathcal{U}$ and $\mathcal{U}'$ s.t. there exists one. Example 5.25 and Example 5.26 give examples where we have a map of coverings between two covers.

Example 5.25. *Let $\mathcal{U} = \{U_k\}_{k \in \mathbb{Z}}$ and $\mathcal{U}' = \{U'_k\}_{k \in \mathbb{Z}}$ be covers of $\mathbb{R}$, where $U_k = (kl - \epsilon, (k+1)l + \epsilon)$ and $U'_k = (klN - \epsilon, (k+1)lN + \epsilon)$, and let $N \in \mathbb{N}$. Then the map $\phi : k \mapsto \lfloor k/N \rfloor$ is a map of coverings from $\mathcal{U}$ to $\mathcal{U}'$.*

Example 5.26. *Let $\mathcal{U} = \{U_k\}_{k \in \mathbb{Z}}$ and $\mathcal{U}' = \{U'_k\}_{k \in \mathbb{Z}}$ be covers of $\mathbb{R}$, where $U_k = (kl - \epsilon, (k+1)l + \epsilon)$ and $U'_k = (kl - a\epsilon, (k+1)l + a\epsilon)$, and let a be a real value greater than 1. Then the identity map $1_\mathbb{Z} : \mathbb{Z} \to \mathbb{Z}$ is a map of coverings from $\mathcal{U}$ to $\mathcal{U}'$.*

In Example 5.27 we give an example of how a map of coverings may be illustrated.

Example 5.27. *In this example we will apply Mapper on the scanned hand drawn figure in Figure 59. The filter function will be the height function. We will use three different covers of the image of the filter function. The three covers will be the covers obtained by dividing the parameter space into 32, 16 and 8 equally long intervals with 40% overlap. The metric used by the single-linkage clustering will be the standard euclidean metric. With these settings together with a suitable clustering epsilon, we get the simplicial complexes together with the simplicial maps between them as shown in Figure 79. Note that the simplicial maps are the extensions of the vertex maps given by the arrowed lines.*

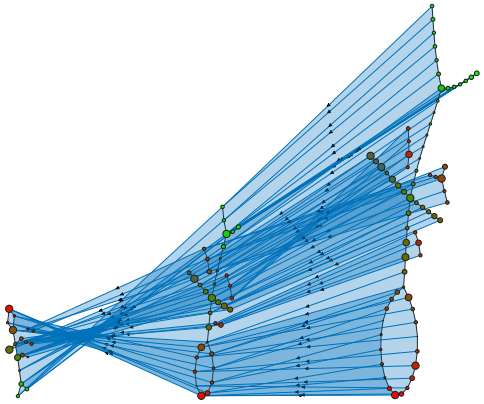


Figure 79: Refer to Example 5.27 for details. This figure illustrates two simplicial maps between three simplicial complexes obtained by Mapper.

5.10 Flexible clustering

In this section we will investigate the possibility of letting the clustering epsilon vary on X when doing single-linkage clustering. We will also see how we can construct simplicial maps between two resulting complexes, obtained by two different choices of clustering epsilon configurations.

Let us first take a look on how we may let the clustering epsilon vary on X and why this is useful. When studying data sets, there will in many cases not be possible to choose a single clustering epsilon that is suitable for all of X. One example, where this is the case, is given in Example 5.28.

Example 5.28. *Let the sampled data X be as shown in Figure 80, and assume that we are given the cover $\mathcal{U}$ of X as shown in Figure 81. If we choose a small clustering epsilon, we will end up with a simplex as shown in Figure 82; if we choose a large cluster epsilon, we will get a simplex as shown in Figure 83. If we instead of fixing one cluster epsilon, choose one value for each open set, then we may choose a small epsilon for the left part, a larger epsilon for the right part, and some intermediate clustering epsilons in the center. By doing this, we could get the simplicial complex in Figure 84, which is the desired result.*

Figure 80: Refer to Example 5.28 for details. This figure illustrates a sampled data set.

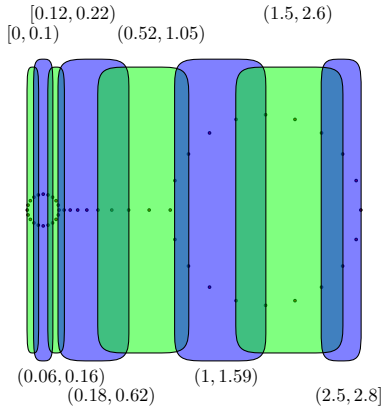


Figure 81: Refer to Example 5.28 for details. This figure illustrates a cover of the sampled data shown in Figure 80.

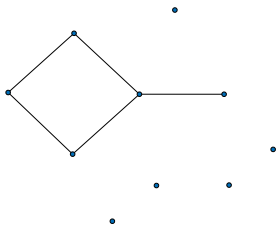


Figure 82: Refer to Example 5.28 for details. The result from applying Mapper, with a small clustering epsilon, on the cover shown in Figure 81.

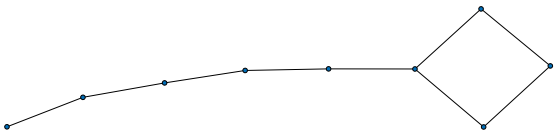


Figure 83: Refer to Example 5.28 for details. The result from applying Mapper, with a large clustering epsilon, on the cover shown in Figure 81.

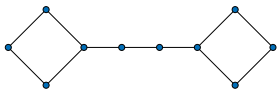


Figure 84: Refer to Example 5.28 for details. The result from applying Mapper, with varying cluster epsilon, on the cover shown in Figure 81.

The freedom to choose a fixed clustering epsilon leaves many possible choices, and by allowing it to variate gives even more. Luckily, small changes of the clustering epsilon does often not alter the result. This fact may be used to create equivalence classes of clustering epsilons and reduce the number of choices. Let us take a closer look on how we can construct these equivalence classes. Assume that we are given a finite metric space (X, d) and are interested in how $\hat{\pi}_{\mathrm{SL}}(X, \epsilon)$ varies with ϵ. Note that for $\epsilon = 0$, each point in X has its own cluster. When ϵ increases, the clusters merge and the number of clusters decrease. Let $\epsilon_1, \dots, \epsilon_n$ be the values for which the clusters merge. Furthermore, let $\epsilon_0 = 0$ and $\epsilon_{n+1} = \infty$. Then $\hat{\pi}_{\mathrm{SL}}(X, \epsilon)$ does not alter for ϵ-values between these values, i.e. $\hat{\pi}_{\mathrm{SL}}(X, \epsilon) = \hat{\pi}_{\mathrm{SL}}(X, \epsilon_k)\ \forall \epsilon \in [\epsilon_k, \epsilon_{k+1})$ and for $k = 0, \dots, n$. Note that we will let $I_k = [\epsilon_k, \epsilon_{k+1})$ for $k = 0, \dots, n$, and that we will call them stability intervals. The equivalence relation we will use is given by

$$\epsilon \sim \epsilon' \Leftrightarrow \exists k \in \{0, \dots, n\} \text{ s.t. } \epsilon, \epsilon' \in I_k.$$

With this knowledge, we know that there actually only are $n+1$ different choices of fixed ϵ-values on X; moreover, the length of each interval gives a notion of how stable each of these choices are.

In the Mapper method, we apply $\hat{\pi}_{\mathrm{SL}}$ on each set U_α of some cover $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ of X. Let us assume that we choose one ϵ_α for each $U_\alpha \in \mathcal{U}$. That would then be equivalent to choosing a function s defined on A and given by $s : \alpha \mapsto I_\alpha$, where I_α is one of the stability intervals of $\hat{\pi}_{\mathrm{SL}}$ on U_α. When choosing the ϵ_α-values, there should be some kind of continuity. One possibility for adding some continuity is to add the criterion

$$\bigcap_{\alpha \in \tau} U_\alpha \neq \emptyset \Rightarrow \bigcap_{\alpha \in \tau} s(\alpha) \neq \emptyset$$

for all $\tau \subset A$ on the choice of s. This gives us a method for varying the cluster epsilon in the Mapper method; moreover, we could have used it to get a proper result from the data set in Example 5.28.

Each configuration, i.e. choice of $s : \alpha \mapsto I_\alpha$, will give its own simplicial complex when used by the Mapper method, i.e. each configuration will give information about X in its own resolution. It would then be useful to find some simplicial map connecting them. In Section 5.9 we discovered that a map of coverings induce a simplicial map on the nerve of the covers. Hence, if we could produce some map of coverings on the covers produced by $\hat{\pi}_{\mathrm{SL}}$ with two configurations s and s', then we can produce a simplicial map between the resulting simplicial complexes. Note that if $\epsilon \leq \epsilon'$, then $\hat{\pi}_{\mathrm{SL}}(X, \epsilon) \leq \hat{\pi}_{\mathrm{SL}}(X, \epsilon')$, i.e. for each cluster $U \in \hat{\pi}_{\mathrm{SL}}(X, \epsilon)$ there $\exists$ a cluster $V \in \hat{\pi}_{\mathrm{SL}}(X, \epsilon')$ s.t. $U \subseteq V$. Hence, if $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ and $\epsilon \leq \epsilon'$, then there $\exists$ a map $\phi : \hat{\pi}_{\mathrm{SL}}(\mathcal{U}, \epsilon) \to \hat{\pi}_{\mathrm{SL}}(\mathcal{U}, \epsilon')$, which is a map of coverings.

There may not always exist a map of coverings $\phi : \hat{\pi}_{\mathrm{SL}}(\mathcal{U}, s) \to \hat{\pi}_{\mathrm{SL}}(\mathcal{U}, s')$, where $\mathcal{U}$ is some cover, when given two arbitary configurations s and s'. However, if s and s' satisfy the condition $\sup \circ s(\alpha) \leq \inf \circ s'(\alpha)\ \forall \alpha \in A$, then $\hat{\pi}_{\mathrm{SL}}(U_\alpha, s(\alpha)) \leq \hat{\pi}_{\mathrm{SL}}(U_\alpha, s'(\alpha))\, \forall \alpha \in A$.

Hence, there $\exists$ a (unique) map of coverings $\phi : \hat{\pi}_{\mathrm{SL}}(\mathcal{U}, s) \to \hat{\pi}_{\mathrm{SL}}(\mathcal{U}, s')$, which then, by Theorem 5.24, gives us a simplicial map $\mathcal{N} \circ \hat{\pi}_{\mathrm{SL}}(\mathcal{U}, s) \to \mathcal{N} \circ \hat{\pi}_{\mathrm{SL}}(\mathcal{U}, s')$.

References

[1] P. B. Bhattacharya, S. K. Jain, and S. R. Nagpaul. *Basic Abstract Algebra*. Cambridge University Press, Cambridge, second edition, 1994.

[2] N. Bourbaki. *Eléments de Mathématique - Algèbre (Chapitres 1 à 3)*.

[3] G. Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46:255–308, January 2009.

[4] G. Carlsson and F. Memoli. Persistent clustering and a theorem of j. kleinberg. 2008.

[5] G. Carlsson and V. D. Silva. Topological approximation by small simplicial complexes. Technical report, MISCHAIKOW, AND T. WANNER, 2003.

[6] G. Carlsson, A. Zomorodian, A. Collins, and L. Guibas. Persistence barcodes for shapes. In *SGP '04: Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 124–135, New York, NY, USA, 2004. ACM.

[7] V. de Silva. A weak definition of delaunay triangulation, 2003.

[8] V. de Silva and G. Carlsson. Topological estimation using witness complexes. In M. Alexa and S. Rusinkiewicz, editors, *Eurographics Symposium on Point-Based Graphics*. The Eurographics Association, 2004.

[9] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. *Discrete and Computational Geometry*, 28(4):511–533, July 2002.

[10] J. Ellson, E. R. Gansner, E. Koutsofios, S. C. North, and G. Woodhull. Graphviz - open source graph drawing tools. *Graph Drawing*, pages 483–484, 2001.

[11] J. W. Frances Kirwan. *An Introduction to Intersection Homology Theory*. Chapman & Hall/CRC, 2006.

[12] R. Ghrist. Barcodes: The persistent topology of data. Technical report, 2007.

[13] A. Hatcher. *Algebraic Topology*. Cambridge University Press, November 2001.

[14] A. Krowne and N. Egge. Planetmath: a collaborative, web-based mathematical encyclopedia, 2001.

[15] S. Kwon, Y.-H. Chu, H.-S. Yi, and C. Han. Dna microarray data analysis for cancer classification based on stepwise discriminant analysis and bayesian decision theory. 2001.

[16] J. M. Lee. *Introduction to Topological Manifolds (Graduate Texts in Mathematics)*. Springer, May 2000.

[17] I. Madsen and J. Tornehave. *From Calculus to Cohomology.* Cambridge University Press, Cambridge, UK, 1997.

[18] T. Marley. Graded rings and modules, 1993.

[19] G. McLachlan. Classification of microarray gene expression data. 2003.

[20] J. Munkres. *Topology.* Prentice Hall, December 1999.

[21] A. I. Saeed. Introduction to microarray analysis and mev. 2005.

[22] R. Simon, M. D. Radmacher, K. Dobbin, and L. M. McShane. Pitfalls in the Use of DNA Microarray Data for Diagnostic and Prognostic Classification. *J. Natl. Cancer Inst.*, 95(1):14–18, 2003.

[23] G. Singh, F. Memoli, and G. Carlsson. Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object Recognition. pages 91–100, Prague, Czech Republic, 2007. Eurographics Association.

[24] G. K. Smyth, Y. H. Yang, and T. Speed. Statistical issues in cdna microarray data analysis. pages 111–136. 2003.

[25] J. W. Vick. *Homology Theory: An Introduction to Algebraic Topology.* Springer, 2nd edition, January 1994.

[26] A. Zomorodian and G. Carlsson. Computing persistent homology. *Discrete Comput. Geom*, 33:249–274, 2005.

[27] A. Zomorodian and G. Carlsson. Localized homology. *Computational Geometry*, 41(3):126–148, November 2008.

[28] A. J. Zomorodian. Computing and comprehending topology: Persistence and hierarchical morse complexes, 2001.

Index