Arnau Mas

Contents

Preface

Lorem ipsum

Chapter 1

The problem

The present work is the result of an internship in the Interactive Augmented Modelling (IAM) at the Computer Vision Center (CVC). It is part of an ongoing collaboration, TOPiomics, between the IAM and the Radiomics Group at the Vall d'Hebron Institute of Oncology (VHIO).

This chapter provides some background and context for the state of the study and the field of radiomics more broadly, as well as a description of how the results of this thesis fit into the bigger picture.

1.1 An overview of radiomics

Radiomics is a relatively recent medical field. Its principal aim is to establish a relationship between a patient's response to treatment and what are known as *radiomic features* of a lesion. These are parameters systematically extracted from scans of a lesion such as a tumour or a nodule. The hypothesis is that there is information that can be extracted from this more systematic study of medical scans, which is not accessible through the more traditional qualitative observation by a medical professional. Furthermore, since imaging is much less invasive than other techniques such as biopsy, if reliable, radiomics could become a useful way of monitoring a patient's evolution and response.

1.1.1 Acquisition of radiomic data

The process of extracting radiomic features begins with the actual imaging of the lesion or tumour: a series of 2D sectional scans are obtained, from which a 3D reconstruction is assembled. It is then the work of a medical professional to manually delimit the contours of the actual lesion, which is known as the *region of interest* or ROI. This process is called *segmentation*. Once this is done, the actual work of computing the radiomic features takes

place. For example, one can gather statistical descriptors of the distribution of brightness of the pixels in the ROI, or calculate geometric parameters of the mesh defined by the boundary of the ROI such as volume, surface area or sphericity. There is a Python package, PyRadiomics, built specifically for the task of extracting radiomic features. [Gri+17] describes the whole list of parameters it can compute. The collection of these features is called the *radiomic signature*.

It is not obvious which of these features will best correlate with the response to treatment. Even worse, some of them might be highly dependent on the segmentaion of the ROI, so that if a different professional performs segments the ROI with slight variations, the value of the features changes wildly. It is desirable to select those featues which, first, are stable under variations in segmentation, called *reproducible*, and are best correlated to the response to treatment.

1.1.2 The nature of radiomic data

The aspect of radiomics that is more interesting to a mathematician is the work of attempting to extract patterns from the radiomic data. However, the data sets available have two significant problems: on the one hand, since all of these features are extracted from clinical cases, the size of available data is necessarily limited, and on the other hand the dimension of the data size can be very high since in the acquisition stage as many features as possible are computed. This means that traditional techniques from statistics and big data analysis which rely on large volumes of data are not well suited to radiomics.

1.2 Prior work

In view of the particular needs of radiomics, the TOPiomics project was developed with the aim of bridging the gap between the needs of medical professionals and the tools mathematicians have at their disposal. Specifically, as described in [GRP19; Gil19], the project intends to develop a method of *outlier detection*, i.e. patients with abnormal radiomic signatures. This is important because these abnormalities might indicate that existing treatments could fail with this patient and thus signal the need to develop specially tailored treatments. An early detection of these outliers is crucial for a positive evolution of the patient.

The idea then is to employ methods from *topological data analysis*, hence the name TOPiomics, for the detection of outliers. A first attempt described in [RBG20] develops an approach based on the *mutual k-nearest neighbours graph*, which is used to find clusters in the data and give a measure of "outlier-ness" to each patient in the data set. There is,

however, an ammount of parameter fiddling involved in this method (the k in k-nearest neighbours) to guarantee that the results are accurate and not an artifact of the choice of k. *Persistent homology* seems specially well-suited to solve this problem since its main application is the analysis of data at many scales. The work of this thesis then is to develop an outlier detector based on persistent homology.

Chapter 2

An overview of homology theory

In this chapter we give a brief explanation of the fundamental concepts involved in the homology theory, with special emphasis on those wich are relevant to persistent homology. A standard treatment of homology can be found in chapter 2 of [Hat01]. Introductions to persistent homology in particular can be found at [EH08; EM12; EM16].

2.1 Introduction

The standard elevator pitch for algebraic topology is as follows: classifying topological objects is hard, whereas algebraic objects (groups, rings, etc) are better understood, so algebraic topology provides tools to compute algebraic invariants from topological spaces, which aids in their study. The natural setting for these ideas is category theory, so that these "tools" become *functors* from the category of topological spaces and continuous maps, Top, to categories such as Grp, the category of groups and group morphisms, or Ab, the category of abelian groups and their morphisms[1].

One of this tools are homology groups, which are the fundamental object of study in the homology theory. At the most abstract level, a homology theory is a family of functors from (some subcategory of) Top to Ab with a series of natural transformations between them, subject to what are known as the *Eilenberg-Steenrod axioms*. For our purposes, however, such a general framework is not necessary, and we will simply describe one particular homology theory, which is known as simplicial homology. This theory is built in two steps. The first step has very much to do with the geometry of the space we are considering and can be seen as a functor from the category of *simplicial complexes*, Simp, to the category of *chain complexes of abelian groups*, $\mathsf{Ch}(\mathsf{Ab})$, as explained in Section 2.2.

[1] We will make some use of basic concepts from cateogry theory in this chapter, a wonderful book on the topic is [Rie16].

This is described in Section 2.2. Then there is a more algebraic step, in essence a functor from $\mathsf{Ch}(\mathsf{Ab})$ to Ab, as seen in Section 2.3. The entire theory is then the composition of these two functors.

2.2 The geometric side: simplices and simplicial complexes

As stated before, simplicial homology is restricted to simplicial complexes, which are, roughly speaking, topological spaces assembled out of *simplices*, which are the generalisation of triangles and tetrahedra to higher dimensions. These spaces can be described combinatorially, which makes them very useful for computation with datasets. There are ways to generalise the theory to broader classes of spaces, such as singular homology, but simplicial homology is sufficient for the analysis required.

2.2.1 Simplices

As stated before, the basic building block of the spaces we will be dealing with are simplicies.

Definition 2.1 (Simplex)**.** A simplex of dimension n, or simply an n-simplex, generated by $n+1$ points of $\mathbb{R}^d$ which do not lie in an affine subspace of dimension $n-1$, is their convex hull, i.e. the smallest convex set which contains them[2]. △

When a set of points generate a simplex they are called *geometrically independent*, see Figure 2.1.

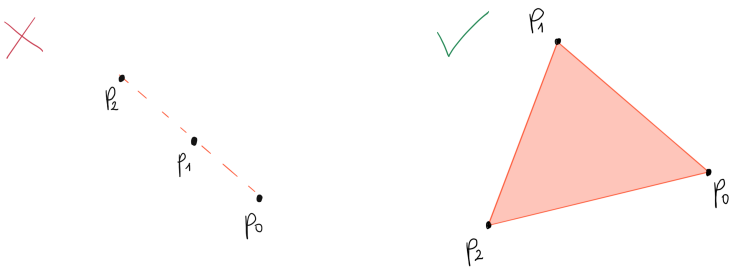

Figure 2.1: The three points on the left are not geometrically independent, whereas the three on the right are and so generate a 2-simplex.

The n-simplex generated by the standard basis of $\mathbb{R}^{n+1}$, $e_0, \ldots, e_n$[3], is called the *stan-*

[2]This means in particular that simplices of dimension n can only exist in spaces of ambient dimension at least n.

[3]Because an n-simplex is generated by $n+1$-points, it will be convenient to number things starting at 0, as opposed to 1 as is more common in mathematics.

dard n-simplex and written Δ^n. It can be shown that

$$\Delta^n = \left\{ (t_0, \dots, t_n) \in \mathbb{R}^{n+1} \mid \sum_{k=0}^{n} t_k = 1, \forall k \leq n \colon t_k \geq 0 \right\}.$$

The standard simplices provide a model for any other simplex. Indeed, if $\phi \colon \mathbb{R}^{n+1} \to \mathbb{R}^d$ is the linear map defined by $\phi(e_k) = p_k$ for $0 \leq k \leq n$ then $\phi(\Delta^n)$ is precisely the simplex generated by $p_0, \dots, p_n$. $(t_0, \dots, t_n)$ are called the *barycentric coordinates* of the point $\phi(t_0, \dots, t_n)$.

Orientation

For the purposes of homology, it is also important to keep track of the *orientation* of a simplex. We will use the idea of general simplices being the image of standard simplices to model this situation

Definition 2.2 (Ordered simplex)**.** An *ordered n-simplex* generated by $p_0, \dots, p_n \in \mathbb{R}^d$ is a map $\sigma \colon \Delta^n \to \mathbb{R}^d$ such that σ is the restriction to Δ^n of the linear map $\phi \colon \mathbb{R}^{n+1} \to \mathbb{R}^d$ given by $\phi(e_k) = p_k$, provided $p_0, \dots, p_k$ do indeed generate a simplex. △

Observe that giving an ordering to the vertices of a simplex completely determines an ordered simplex, thus we introduce the notation $[p_0, \dots, p_n]$ for an ordered simplex as defined above.

Consider now the natural action of the symmetric group, $\mathfrak{S}_n$, on $\mathbb{R}^n$ by permuting the elements of the standard basis. Then, a reordering of an ordered n-simplex σ is of the form $\tau \circ \sigma$ for some $\tau \in \mathfrak{S}_{n+1}$. Conversely, since the action of $\mathfrak{S}_n$ is free and transitive, for any two ordered simplices with the same vertices, σ_1 and σ_2 there will always exist a unique permutation $\tau \in \mathfrak{S}_n$ such that

$$\sigma_1 = \tau \circ \sigma_2.$$

If τ is even then σ_1 and σ_2 are said to have the same orientation, whereas if τ is odd then they are said to have opposite orientations. This means in particular that there are only two possible orientations for any simplex, which is consistent with the usual intuition for orientation. See Figure 2.2 for an example.

Given a simplex $\sigma = [p_0, \dots, p_n]$, the $n+1$ possible simplices we get by removing one of the generators —i.e. $[p_0, \dots, \hat{p}_k, \dots, p_n]$— are called the *faces* of σ. Observe that the orientation of a simplex induces an orientation on the faces which turns them into ordered simplices of dimension $n-1$.

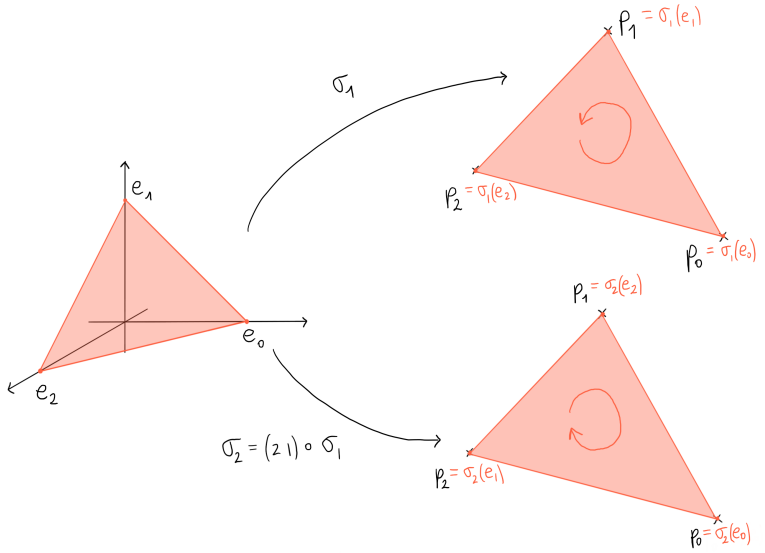

Figure 2.2: Two different orderings of a 2-simplex with the same vertices that determine different orientations.

2.2.2 Simplicial complexes

Simplicial complexes are spaces assembled out of properly glued together simplices, as the following definition makes precise.

Definition 2.3 (Simplicial complex)**.** A *simplicial complex* K is a collection of simplices of $\mathbb{R}^d$ such that

(i) if $\sigma \in K$ and $\tau \in K$ is a face of σ then $\tau \in K$,

(ii) for any $\sigma, \tau \in K$ then $\sigma \cap \tau \in K$[4].

△

The mental picture is that the simplices are only allowed to be glued along their faces.

The simplices of K are often called its *cells*. The collection of all of the cells of K of dimension n or lower is called its n-skeleton, $\text{skel}_n(K)$. The highest dimension of any of the cells of K is the *top dimension* of K.

A simplicial complex K determines a topological space, called its *underlying space*, which is the subset of $\mathbb{R}^d$ determined by the union of (the images of) all of the cells of K equipped with the subspace topology, which we will write as $|K|$. Pursuing the idea of realising more general topological spaces as underlying spaces of some simplicial complex is what leads to singular homology.

[4]This is an abuse of notation since what is really meant is $\sigma(\Delta^k) \cap \tau(\Delta^l)$.

2.2.3 Abstract simplicial complexes

Notice that a simplex is always determined by its vertices, and vice versa, except for, of course, whenever the vertices do not generate a simplex, in the sense of definition 2.1. This requirement comes from thinking of simplices and simplicial complexes as geometric objects. But if we don't think of them in this way, and simply allow for any $n+1$ vertices to determine an n-simplex they become strictly combinatorial objects. In this case, we also have to lift the second condition in the definition of a simplicial complex, which was related to how the simplices are assembled together as geometrical objects and therefore becomes meaningless in this new context. This more general complexes are called *abstract simplicial complexes* and are purely combinatorial objects. From this viewpoint, an n-simplex is now an (ordered) set of $n+1$ vertices, and its faces are just its subsets of n vertices. Ans so definition 2.3 is replaced by

Definition 2.4 (Abstract simplicial complex)**.** An *abstract simplicial complex* K is any set closed under the relation of inclusion, i.e., if $\sigma \in K$ then if $\tau \subseteq K$, it must be the case that $\tau \in K$. △

For our purposes, we will assume that all of the simplices that make up a complex are finite, that there is a maximum possible dimension and that complexes are at most countably infinite, thus avoiding problems related to size. This is not much of a restriction since the main application are datasets which are finite.

Notice that to specify a complex it suffices to exhibit its *maximal simplices*, i.e. those simplices which are not the face of any other simplex, or equivalently those which are maximal with respect to inclusion[5].

2.2.4 Special complexes

In the context of topological data analysis, one is often handed a bare point cloud and is faced with the task of generating an appropriate simplicial complex. We now describe various ways of doing this. For the remainder of this section, X will denote a finite subset of $\mathbb{R}^d$, which models the raw point cloud we begin with.

[5]They are guaranteed to exist by requiring that the dimensions of all simplices that make up the complex be bounded.

Čech complex

The ϵ-Čech complex of X, $C_\epsilon(X)$, is determined by the following prescription: $\sigma \subseteq X$ is in $C_\epsilon(X)$ if and only if

$$\bigcap_{p\in\sigma} B_\epsilon(p) \neq \emptyset$$

where $B_\epsilon(p)$ is the ball of radius ϵ centered at p.

This does determine a simplicial complex, since if a certain set of vertices is such that the intersection of all the ϵ-balls centered at them is nonempty, then so will the intersection of the corresponding balls for any subset be nonempty.

Vietoris-Rips complex

The prescription for the ϵ-Vietoris-Rips complex, $V_\epsilon(X)$, is as follows: $\sigma \subseteq X$ defines a simplex in $V_\epsilon(X)$ if and only if for every $p, q \in \sigma$

$$B_\epsilon(p) \cap B_\epsilon(p) \neq \emptyset.$$

In other words, a set of points generate a simplex whenever each of them is at distance at most 2ϵ[6] from the rest. This also shows that the Vietoris-Rips complex is indeed a complex.

Notice that the ϵ-Čech complex is always a subcomplex[7] of the ϵ-Vietoris-Rips complex. Indeed, if $[p_0, \ldots, p_n]$ is a simplex in $C_\epsilon(X)$, then for any $0 \leq i, j \leq n$

$$\emptyset \neq \bigcap_{k=0}^{n} \subseteq B_\epsilon(p_i) \cap B_\epsilon(p_j)$$

so that $[p_0, \ldots, p_n]$ is also a simplex in $V_\epsilon(X)$. The intuition is that since the condition for a set of points to determine a simplex in the Vietoris-Rips complex is weaker than for the Čech complex, the former will contain more simplices than the latter.

As a consequence of the Nerve Theorem (Theorem 2.4 in [Ghr14]), the underlying space of the Čech complex is homotopically equivalent to the union of the balls of radius ϵ centered at each point in the cloud. This is a guarantee that the homology of the Čech complex should reflect the homology of the point cloud (if we think the points in the cloud come from some underlying space). This comes at the cost of the Čech complex being more expensive to store. The Vietoris-Rips is in this way requires a tradeoff: on the one

[6] Some texts use a slightly different convention such that the vertices of a simplex in $V_\epsilon(X)$ are at distance at most ϵ (rather than 2ϵ) from each other

[7] A subcomplex of an abstract simplicial complex is a subset which is itself an abstract simplicial complex.

hand it is much less memory intensive than the Čech complex, but on the other hand the tie to the geometry of the point cloud is weaker, given by the inclusions

$$C_\epsilon(X) \subseteq V_\epsilon(X) \subseteq C_{2\epsilon}(X).$$

Clique complex

Complexes can also be built out of graphs. Given a graph G with vertex set V, one defines its *clique complex* with the followig prescription: $\sigma \subseteq V$ is a simplex in the clique complex if and only if it is a clique, i.e. a fully-connected subgraph of G. This is indeed a complex since the graph generated by any subset of vertices of a clique is itself a clique.

There is a relationship between the Vietoris-Rips complex and the clique complex. The 1-skeleton of the Vietoris-Rips complex (and of any other complex) determines a graph by taking the edges to be the 1-cells. Then its clique complex is exactly the same as the original Vietoris-Rips. Indeed, if $\sigma = \{p_0, \dots, p_n\}$ is a clique of $\operatorname{skel}_1 V(X)$ then there is an edge between any pair p_i, p_j. But this means $\{p_i, p_j\}$ is a 1-cell of $V(X)$, thus $B_\epsilon(p_i) \cap B_\epsilon(p_j) \neq \emptyset$. Since this is the case for any pair of vertices of σ, σ is an n-cell of $V(X)$, by definition. The converse is very similarly shown.

The filtration used in the detector is based on the clique complex of the mutual k-nearest neighbours graph of the point cloud.

2.3 The algebraic side: chain complexes and homology groups

So far we have dealt with the special kinds of spaces whose homologies we wish to compute. We now turn to the actual task of computing. The main algebraic idea which requires introduction is that of a chain complex, which plays an important role in homological algebra, the study of the algebraic details of homology theory. Out of the chain complexes we get the actual homology groups, by a quotienting operation we will describe.

2.3.1 Chain complexes

Definition 2.5 (Chain complex)**.** A *chain complex*[8] (of abelian groups) is a family of abelian groups $\{C_n\}_{n\in\mathbb{N}}$ together with a family of morphisms $\partial_n \colon C_n \to C_{n-1}$ such that $\partial_n \circ \partial_{n+1}$. We write (C_*, ∂_*) for the whole complex. △

[8]The word complex is being used for two different concepts: simplicial complexes and chain complexes. To avoid confusion we will adopt the convention that complex without qualifier refers to a simplicial complex whereas to refer to a chain complex we will always use both words.

The elements of C_n are called *n-chains*. The n-chains which lie in $\ker \partial_n$ are the *n-cycles*. And the chains in $\operatorname{im} \partial_{n+1}$ are called the *n-boundaries*. It follows from the definition of a chain complex that

$$\operatorname{im} \partial_{n+1} \subseteq \ker \partial_n$$

i.e. that every boundary is a cycle. This means the following definition makes sense.

Definition 2.6 (Homology groups)**.** The *n-th homology group* of the chain complex (C_*, ∂_*) is the quotient

$$H_n(C) := \ker \partial_n / \operatorname{im} \partial_{n+1}.$$

△

Two elements which are the same when projected onto the homology group are called *homologous*. From the definition it is immediate to see that two chains are homologous if they differ by a boundary, i.e. c_1 is homologous to c_2 if there exists d such that $c_1 = c_2 + \partial d$.

2.3.2 Simplicial chain complexes

The next step is building a chain complex out of the simplicial complexes we have built so far and computing its homology groups. These will then be the homology goups of the space.

This chain complex is called, perhaps confusingly, a simplicial chain complex. Given a simplicial complex K, the corresponding simplicial chain complex, which we now define, is written $(C_*(K), \partial_*)$. We need each $C_n(K)$ to be an abelian group, so an easy way to achieve this is to define $C_n(K)$ as the free abelian group generated by the n-cells of K. This is not quite it since we want the orientation to play nicely with the group structure. We add the additional condition that for any n-simplex σ and permutation $\tau \in \mathfrak{S}_n$,

$$\tau \circ \sigma = (-1)^\tau \sigma$$

where $(-1)^\tau$ is the sign of τ. That is, if we reorder the vertices of σ by an even permutation, so we don't change the orientation of σ, nothing changes. But if we reorder by an odd permutation, so that the orientation of σ changes, a minus sign is picked up.

In fact, once we have fixed an orientation for each of the cells of K we can treat $C_n(K)$ as a free group, so that an n-chain $c \in C_n(K)$ is of the form

$$\sum_{\sigma \in K_n} a_\sigma \sigma$$

for $a_\sigma \in \mathbb{Z}$.

Of course the other half of a chain complex are the boundary morphisms, which are defined as follows

Definition 2.7 (Boundary morphisms)**.** We define, for $n \geq 1$, the morphisms

$$\partial_n \colon C_n(K) \longrightarrow C_{n-1}(K)$$

by giving their action on a generator as

$$\partial_n[p_0, \ldots, p_n] := \sum_{k=0}^{n} (-1)^k [p_0, \ldots, \hat{p}_k, \ldots, p_n]$$

and extend them on any other chain by

$$\partial_n \left(\sum_{\sigma \in K_n} a_\sigma \sigma \right) := \sum_{\sigma \in K_n} a_\sigma \partial_n \sigma.$$

△

We need to show that all of this is indeed a chain complex, which is a consequence of the following result.

Lemma 2.8. *For any $n > 1$, $\partial_{n-1} \circ \partial_n = 0$.*

Proof. It suffices to show this on any generator, which is a simple calculation:

$$\begin{aligned}
(\partial_{n-1} \circ \partial_n)[p_0, \ldots, p_n] &= \partial_{n-1} \left(\sum_{k=0}^{n} (-1)^k [p_0, \ldots, \hat{p}_k, \ldots, p_n] \right) \\
&= \sum_{k=0}^{n} (-1)^k \partial_{n-1} [p_0, \ldots, \hat{p}_k, \ldots, p_n] \\
&= \sum_{k=0}^{n} (-1)^k \left(\sum_{l=0}^{k-1} (-1)^l [p_0, \ldots, \hat{p}_l, \ldots, \hat{p}_k, \ldots, p_n] \right. \\
&\quad \left. + \sum_{l=k+1}^{n} (-1)^{l-1} [p_0, \ldots, \hat{p}_k, \ldots, \hat{p}_l \ldots, p_n] \right) \\
&= \sum_{\substack{k=0 \\ l<k}}^{n} (-1)^k (-1)^l [p_0, \ldots, \hat{p}_l, \ldots, \hat{p}_k, \ldots, p_n] \\
&\quad - \sum_{\substack{k=0 \\ l>k}}^{n} (-1)^k (-1)^l [p_0, \ldots, \hat{p}_k, \ldots, \hat{p}_l, \ldots, p_n] \\
&= 0
\end{aligned}$$

because both summands in the last line are the same. □

Thus, if d is the top dimension of K, we have the simplicial chain complex

$$C_d(K) \xrightarrow{\partial_d} C_{d-1}(K) \longrightarrow \cdots \longrightarrow C_1(K) \xrightarrow{\partial_1} C_0(K) \longrightarrow 0$$

We will drop the subscripts from the boundary morphisms whenever they can be deduced from the context.

Interpretation

The chain groups are related to the concatenation of loops in a general topological space[9] Indeed, the sum of two simplices can be understood as their concatenation. Then, the boundary of a simplex is the sum of its faces. Except not quite, because if we simply used the induced orientation for the faces we would find that $\partial_n \circ \partial_{n+1} \neq 0$. Geometrically, what is happening is that with the induced orientation, the faces don't "go around nicely", so to say, and about half of them need their orientation flipped, which is what the alternating sign accounts for. Figure 2.3 shows this for the boundary of a 2-simplex. The more general

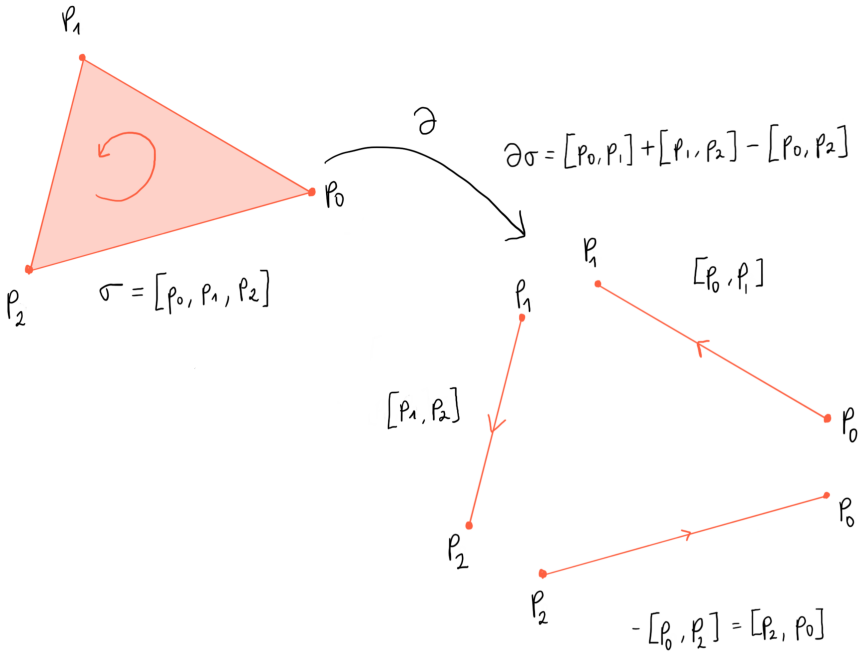

Figure 2.3: This is a representation of the boundary of a 2-simplex. Notice that the face $[p_0, p_2]$ is flipped, which results in the orientations of the boundary going around counterclockwise.

interpretation of homology is discussed in Section 2.4, however, notice that the cycles and boundaries in a simplicial chain complex are cycles and boundaries in the geometric sense and in fact this is the origin of the terminology.

Homology with different coefficients

Instead of considering the free abelian groups generated by the cells, we could think about the F-vector space they generate, for any field F, subject to the compatibility with orientation. Then we would get a chain complex of vector spaces. The homology groups

[9] As explained in [Hat01], the homology groups are in some sense the abelianisation of the fundamental groups.

of this complex would be F-vector spaces. In this case, one defines the *Betti numbers* as

$$\beta_n(X, F) = \dim H_n(X, F)$$

where $H_n(X, F)$ are the homology groups of the F-vector space chain complex.

The case of $F = \mathbb{F}_2$ will be of particular interest to us since the algorithm implemented later computes homology with coefficients in $\mathbb{F}_2$. In this case we have

$$\tau \circ \sigma = (-1)^\tau \sigma = \sigma$$

so that this homology is orientation blind in some sense. In particular, the alternating signs involved in the definition of ∂ vanish.

2.4 The meaning of homology

So we have described how to calculate, at least in principal, these homology groups, and the claim at the beginning of this chapter was that they give us useful information about the space at hand. the obvious question is then: what information exactly?

The first important point is to understand what homology *cannot* tell us. It can be shown that (singular) homology is invariant under homeomorphisms. But in fact it is also invariant under homotopy equivalence, which is a much weaker relationship than homeomorphism: a point is homotopically equivalent to $\mathbb{R}^n$. This means it cannot be used as a tool to classify spaces, at least not in the most general setting.

The easiest homology group to interpret is H_0, since it can be shown that β_0 is the number of connected components of the space. Higher homology groups have to do with n-dimensional holes in the space. Indeed, a cycle which is not the boundary of anything indicates the presence of some sort of "obstacle" in the space. What it means for two simplicial chains to be homologous is that they are both the boundary of a higher dimensional chain, which is very similar, yet weaker, then the notion of homotopy. Therefore, if there is a cycle which is not homologous to zero, it means it cannot, very roughly speaking, be continuously deformed to a point. For instance, β_1 is 1 for S^1, which makes sense because S^1 is in some sense the prototypical 1-dimensional hole. For S^2, $\beta_1 = 0$ but $\beta_2 = 1$, reflecting the fact that S^2 is "hollow".

2.5 The idea of persistence

So far we have seen various ways of extracting information related to the shape of our data using homology. All of this ways, however, carry with them an ammount of arbitrary

choice. For example, both the Vietoris-Rips and Čech complexes depend on a scale parameter ϵ, and the appropriate choice of ϵ is generally not clear from the data. The idea of persistent homology is to sidestep this problem altogether by considering the homology of the data at every scale to determine which are the features really reflect geometric aspects and which are byproducts of background noise. The first will be, roughly speaking, those which are present at a large range of scales, i.e. those which are *persistent*. We now formalise these ideas.

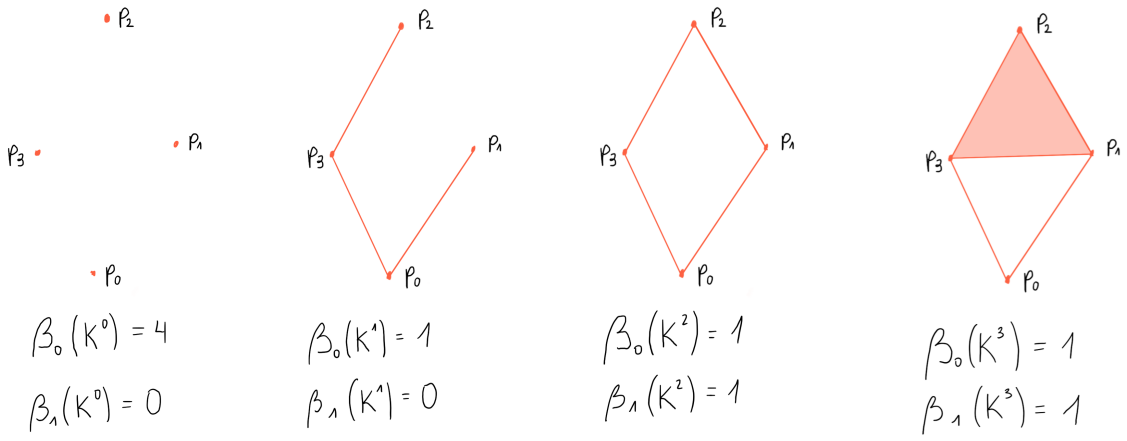

Figure 2.4: Four different steps of a filtration and their first two Betti numbers. At the first step, the complex consist of four isolated points. At the next step, three 1-simplices connect all the points but no combination of them is a cycle, so the dimension of $H_1(K^1)$ is still 0. At the next step, the simplex $[p_1, p_2]$ closes the cycle $c = [p_0, p_1] + [p_1, p_2] + [p_2, p_3] + [p_3, p_0]$ which is not the boundary of anything (there are no 2-simplices), and so the dimension of $H_1(K^2)$ becomes 1. Finally, a 2-cell is born, but this is not sufficient to kill the class of c. Indeed, $f_2^3(c) = c = [p_0, p_1] + [p_1, p_3] + [p_3, p_0] + \partial[p_1, p_2, p_3]$.

Definition 2.9 (Filtration)**.** A *filtration* is a family of simplicial complexes $\{K_i\}_{i=0}^n$ such that K_i is a subcomplex of K_{i+1},

$$K_0 \subseteq K_1 \subseteq \cdots \subseteq K_n$$

where we will mostly assume $K_0 = \emptyset$. △

The inclusions $\iota_i^j \colon K_i \hookrightarrow K_j$, because of the functoriality[10] of homology, give rise to maps between the homology groups at different steps of the filtration,

$$H_p(\iota_i^j) \colon H_p(K_i) \longrightarrow H_p(K_j)$$

which we will write as f_i^j for short. We say a certain class is *born at* i if it is not in the image of any f_k^i for any $k \leq i$. And we say it *dies at* j if it is in the kernel of f_i^j but not of f_i^k for $k < j$. The difference $j - i$ is called its *persistence* or *lifetime*.

[10] Again, see [Rie16] for a detailed introduction to category theory.

What this gives is a reflection at the level of homology of how the shape of the complex changes at every step of the filtration. See Figure 2.4 for a detailed example.

The information that persistent homology is often encoded into what is known as a *barcode*, which is the birth and death of every element of the homology groups that has lived at any of the steps of the filtration.

Chapter 3

The implementation of the detector

This chapter is a detailed look at how the outlier detector is actually implemented. We put special emphasis on the mathematical considerations behind each step and how they reflect on the actual code. The whole library is freely available at https://github.com/arnaumas/mknn_homology/.

3.1 Why mutual k-nearest neighbours graphs?

As stated in Section 1.2, the first approach to identify clusters and outliers in a dataset was by finding the maximal cliques of the mutual k-nearest neighbours graph, MkNN graph for short. Let's describe how it is constructed to understand why it is useful.

One starts with the nearest neighbours graph. Given a point cloud, C, its Nearest Neighbour directed graph is constructed according to the following prescription: there is an edge from p to q if and only if $d(p,q) = \min_{r \in C} d(p,r)$, so q is the closest point to p, its *nearest neighbour*. The reason the graph is directed is because the relation of being a nearest neighbour is not symmetric. Indeed, consider three points p_1, p_2, p_3 which all lie in a straight line in this order. Let $d_{12} = d(p_1, p_2)$ and $d_{23} = d(p_2, p_3)$ and suppose $d_{12} < d_{23}$. Then, p_2 is p_1's nearest-neighbour, and vice versa. Yet p_2 is also p_3's nearest neighbour but neither p_1 nor p_2 are a nearest neighbour of p_3.

This generalises to the k-nearest neighbours graph, in which each point is connected to its k-th nearest neighbours, that is, the k-th closest points to it. The resulting graph is still directed. We obtain the undirected *mutual* k-nearest neighbours graph by connecting two points if and only if there are edges between them in both directions, i.e. if they are *mutual* k-nearest neighbours.

Looking at Figure 3.1 we see the 2-nearest neighbours and mutual 2-nearest neighbours graphs for the same point cloud. In a kNN graph, it is always the case that every vertex

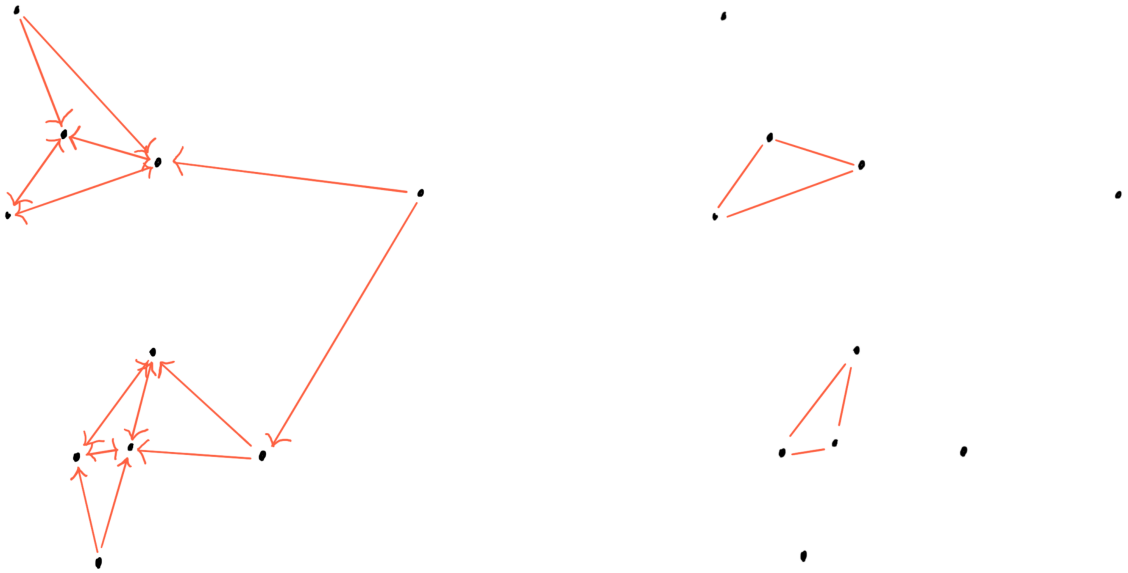

Figure 3.1: The 2NN and M2NN graphs for the same dataset. Note how the M2NN graph is undirected and has a lot less edges.

has k edges, which means that even points which are far away from any cluster are connected to some other points. This is not very useful if we wish to detect outliers. However, when taking the *mutual* version, many of this edges disappear. Indeed, now, for an outlier to become connected to a cluster a higher k will be required so that every point within the cluster has become connected to all of its immediate neighbours and is forced to "look" outside the cluster for additional neighbours. This idea of waiting for a higher k is exactly the one we exploit when combining the MkNN graph with persistent homology, for outliers will presumably become classes in H_0 with few elements and with a long lifetime.

3.2 The steps of the detector

The whole process from input to output is roughly divided in four steps. First, the data is shaped into an expected format and wrapped in a class which knows how to compute its MkNN graph for any k. Then, the filtration is built and suitably stored. Finally the actual computation of the persistence homology takes place. And then, based on the results of the persistent homology, each point is given an "outlier-ness" score.

3.2.1 Constructing the MkNN graph

The input is assumed to be in the form of a NumPy array of shape (N, d) where N is the number of points and d the number of features or equivalently the ambient dimension.

This array is wrapped in an instance of the class `Cloud` which implements a method to compute the MkNN graph for a given k.

The procedure to encode this graph is as follows. First, the distance matrix, D, for the data set is computed. If we index the points of the cloud by $C = \{p_i\}_{i=1}^n$, then this matrix contains the difference between every pair of points:

$$D_{ij} = d(p_i, p_j).$$

This can be done with NumPy and is implemented as a method of the `Cloud` class. Then, again using NumPy, the indices of each column of D are sorted into a new matrix such that the i-th column of this new matrix lists the points of C by increasing order of distance to p_i. In particular, the first k are p_i's k-th nearest neighbours. From this we can then construct the adjacency matrix of the kNN graph, A, which is in general not symmetric since the graph is directed. The adjacency matrix of the MkNN graph, M, is given by

$$M = A \odot A^\top$$

where $\odot$ denotes the entrywise prodct. Indeed, we have

$$M_{ij} = A_{ij} A_{ji}^\top = A_{ij} A_{ji}$$

so that M_{ij} is nonzero if and only if both A_{ij} and A_{ji} are nonzero, i.e. if there is an edge going from p_i to p_j and one from p_j to p_i. Furthermore

$$M_{ij} = A_{ij} A_{ji} = A_{ji} A_{ij} = M_{ji}$$

which means that M is symmetric and therefore the adjacency matrix of an undirected graph, as we claimed. This adjacency matrix is used to represent the MkNN graph as a NetworkX graph. NetworkX is a Python library which implements many algorithms from graph theory.

3.2.2 Building the filtration

The filtration used is based on the clique complex, see Section 2.2.4. For every k we have the clique complex of the corresponding MkNN graph. And letting k from 0 to an appropriate upper bound we get a filtration. That this is indeed a filtration is because the M$(k-1)$NN graph is a subgraph of the MkNN graph. Indeed, $k-1$ mutual nearest neighbours are in particular k mutual nearest neighbours. This means cliques of the first are cliques of the latter, thus simplices of the clique complex of the first are simplices of the clique complex of the latter, which proves the claim.

There are a couple caveats. First, we restrict the top dimension of the complex to the ambient dimension d, which means we will ignore any clique with more than $d+1$ vetices. This is because homology at a dimension higher than this does not have much geometric meaning (the points that would generate such simplicies are necessarily not geometrically independent). Secondly, the matter of the upper bound for k. A generous upper boud is $N-1$ which results in the MkNN graph being fully connected, and the homology of the resulting complex is trivial (zero in all dimensions save for H_0 which is of dimension 1). The problem is that the process of building the complex does not scale well with k in terms of time, so it seems desirable to seek a lower stopping point since most of the "interesting" fenomena will have, heuristically, already happened somewhere between $k = \lfloor N/2 \rfloor$ and $k = N$.

The work of constructing the filtration is packaged up in the `Filtration` class. This class is passed an instance of `Cloud`. The most straightforward way to represent a filtration is to simply store an ordered list of the *new* simplices, in this case cliques, which appear at every step. There are more efficient data structures that can be used, see [ALS12; Zom10], but since the data sets analysed are of small size, this more naive approach sufficed.

The NetworkX package has methods which can compute every clique in a graph. In a loop over k from 1 to the appropriate upper bound, the correspoinding MkNN graph is computed and a list of its cliques (of dimension less than or equal to d) is extracted. Of these, all those which are new are appended to a list. The `Simplex` class wraps a clique as a list of its points as well as keeping track of the step, k, at which the clique is born. In addition it implements methods used in the actual computation of the homology groups.

This list is then sorted by birth, such that cliques born earlier appear first, and cliques with the same birth are sorted by increasing dimension. This guarantees that any simplex is always preceded by its faces, which is required for later computations.

3.2.3 Computing the homology

The main idea

The algorithm used to compute the persistent homology is based on the one employed in [Cam18]. Informally, the idea is to start at the beginning of the filtration and to process every cimplex in order of appearance in the filtration. Whenever a new simplex is born one of two thing can happen: it either closes a cycle, thus a new class in a homology group appears, or it does not, in which case what used to be a cycle (because of how the filtration is sorted, the faces of a simplex always appear before it) is now the boundary of a simplex, and so it dies. In what folllows we make this precise. All of the following

arguments rely on the fact that we are computing homology for coefficients in $\mathbb{F}_2$.

First an observation. If we have a filtration $K^0 \subseteq \cdots \subseteq K^N$, we can always construct a new filtration, $L^0 \subseteq \cdots \subseteq L^M$, such that the difference between successive steps of L is a single simplex. Indeed, put $L^0 = K^0$ and say $K^i = K^{i-1} \cup \left\{\sigma_1^i, \ldots, \sigma_{l_i}^i\right\}$. Then define

$$\begin{aligned} L^1 &= L_0 \cup \left\{\sigma_1^1\right\} \\ L^k &= L^{(k-1)} \cup \left\{\sigma_k^1\right\} \\ L^{l_1+\cdots+l_q+k} &= L^{l_1+\cdots+l_q+(k-1)} \cup \left\{\sigma_k^{q+1}\right\}. \end{aligned}$$

For this to acutally be a filtration, the order in which we add the simplices has to be such that every simplex is always preceded by its faces. And note $K^i = L^{l_1+\cdots+l_i}$. So what we are doing is splitting every step of the filtration and creating a new filtration by adding simplicies one by one.

Consider a single step of L, going from L^i to L^{i+1} by adding a single p-simplex σ. All of the faces of σ, $w_0, \ldots, w_p$ are in L^1. In particular, the sum $w_0 + \cdots + w_p$ is a cycle. Write π_p^i for the projection from the p-th chain group of the i-th step of the filtration, $C_p(L^i)$, onto the p-th homology group at the i-th step, $H_p(L^i)$. Now one of two things can happen:

(i) if $\pi_p^i(w_0 + \cdots + w_n) = 0$ then $w_0 + \cdots + w_n$ is already the boundary of some other chain *in* L^i, say c. But in L^{i+1} we have $\partial\sigma = \partial c$, thus $\partial(\sigma + c) = 0$, i.e. $\sigma + c$ is a cycle. This is σ closing a cycle: think of c as being the container and σ being the lid. And this cycle cannot be the boundary of anything in L^{i+1} because σ is not the face of anything in L^{i+1} (simplices are precede by their faces). Thus a new element of $H_p(L^{i+1})$ has been born.

(ii) if $\pi_p^i(w_0 + \cdots + w_n)$ is not zero then $w_0 + \cdots + w_p$ is not the boundary of anything and $\pi_p^i(w_0 + \cdots + w_n)$ is a nonzero element of $H_{p-1}(L^i)$. But in L^{i+1}, $\partial\sigma = w_0 + \cdots + w_n$, thus σ kills this cycle.

Every cycle must be born and die at one of the steps of L, so by doing this for every single simplex of the whole filtration we will detect the birh and death of every cycle. Now, when going back to the smaller filtration K, it might very well be the case that several cycles are born and die in a single step of K, since we are collapsing several steps of L into 1. This is essentially what is done in the code: look at every simplex as a step in a larger filtration and then keep all of the cycles that survive for at least a step in the original filtration.

The actual computation

Let's now see how the idea just described can be implemented.

The classes `Simplex` and `Chain` can be used to calculate with the chain groups. As mentioned before, a `Simplex` simply wraps a clique as well as its birth. It also has the property `faces`, which returns a list of the faces of the simplex.

The elements of the chain group are linear combinations of simplices (of the same dimension), but since we are working over $\mathbb{F}_2$, this amounts to a list of simplices, which is what `Chain` stores[1]. Furthermore, this class implements the addition of chains which, again because the field we ar working over has characteristic 2, reduces to taking the symmetric difference of the lists of simplices of the two chains we are adding. Indeed, if we have two chains of the form $c_1 = \sum_{i=1}^{n} \epsilon_i \sigma_i$ and $c_2 = \sum_{i=1}^{n} \delta_i \sigma_i$ with $\epsilon_i, \delta_i \in \mathbb{F}_2$, then

$$c_1 + c_2 = \sum_{i=1}^{n} (\epsilon_i + \delta_i)\sigma_i$$

which means the coefficient of σ_i in $c_1 + c_2$ is $\epsilon_j + \delta_j$. And this will be 1 provided only one of ϵ_j or δ_j is equal to 1, and will be 0 whenever *both* ϵ_j and δ_j are 1 or 0. So σ_j will be present in $c_1 + c_2$ whenever it is present in c_1 or c_2, but not both.

This makes it very easy to implement the boundary morphisms. For a single simplex, wrap the list of faces inside a `Chain` object. And for a larger chain, add up the boundaries of each of its constituent simplices. Again, this works because we are taking coefficients from $\mathbb{F}_2$, so that, as explained in Section 2.3.2 the alternating signs that appear in the definition of the boundary all disappear.

The main issue with implementing the algorithm defined above is that we have no access to the projection π_p^i. What we do however is to implement what is basically a memoised version of it by keeping track of what each simplex is homologous[2] to in a dictionary. Its keys are every simplex in the filtration and the values are instances of the `HomologyClass` class. The dictionary is modified to guarantee transitivity, so that when the value of a key changes, all other keys that had that value are updated accordingly. Any time we encounter a new simplex we know it is not homologous to anything else, so it is assigned a homology class which has itself as a representative. Then, as more simplices are encountered, new identifications take place.

So then we loop over every simplex stored in the filtration and compute the sum of the homology classes of its faces. If the result is zero, it means a new cycle was closed.

[1] In fact we choose to work over $\mathbb{F}_2$ precisely because of this reason.

[2] Strictly speaking, the relationship of homology was only defined for cycles, but we can extend it to the quotient $C_p(K^i)/\operatorname{im}\partial$.

If it is not then an identification takes place. If σ is the p-simplex being processed and $w_0, \ldots, w_p$ are its faces then, after σ has been added, we have

$$\pi^i_{p-1}(w_0 + \cdots + w_p) = 0$$

so, we can solve for one of the faces as the sum of the rest, i.e.,

$$\pi^i_{p-1}(w_k) = \pi^i_{p-1}(w_0 + \cdots + \hat{w}_k + \cdots + w_p) = 0.$$

We do this for the youngest face of σ and update its homology class to one generated by the sum of all faces of σ except for it. This way we build up the projection map as we go along.

3.2.4 Detecting the outliers

What does it mean for a point to be an outlier? Intuitively, a point is an outlier if it is far removed from any cluster. We can use the knowledge of the persistent homology to precisely find out which of these points fit this criterion. As stated in Section 2.4, the 0th homolgy group contains the information on the number of connected components. We know that at the first step of the filtration all we have are the points themselves, so that the dimension of H_0 is the number of points. As more simplices begin to appear, some of these classes will begin to die as they start to merge with each other. Thus, if a point is an outlier, we expect the homology class it generates to live for a considerable time as it will take a relatively high value of k for the big clusters to start seeking neighbours outside of them, and to never gain a large number of representatives. We have access to both of these pieces of information, the lifetime of each homology class and the number of representatives it has when it dies. If we plot all of the points in the cloud in a lifetime vs. size chart, the outliers will be those closest to the lower right-hand corner, i.e. those which lived for a long time without merging with any other cluster.

Chapter 4

The results

In this chapter the detector is tested, first with a toy example and then with actual radiomic data provided by the VHIO.

4.1 A toy example

Let's test the detector with a toy dataset, which can be seen in Figure 4.1. This dataset

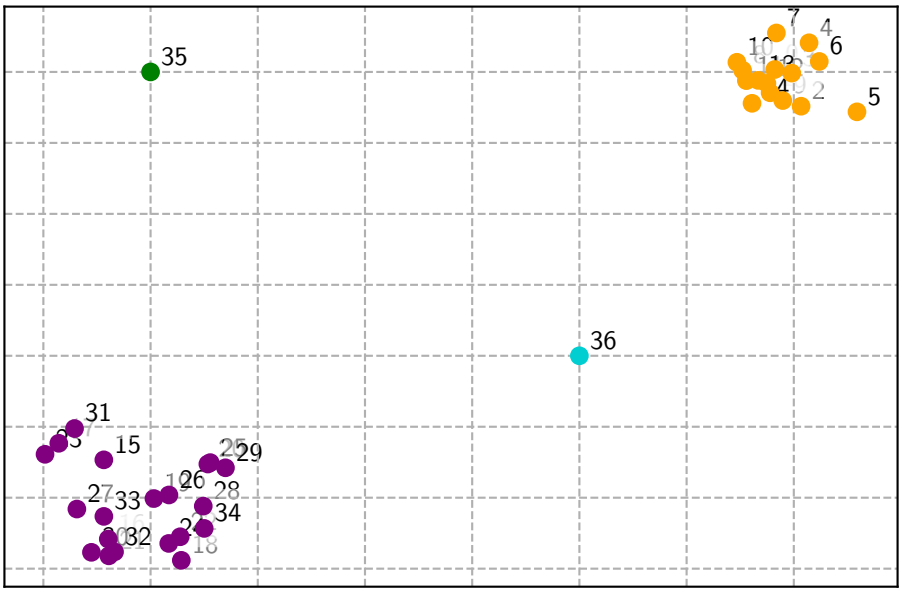

Figure 4.1: Custom point cloud to illustrate the working of the detector. This set has two clusters, the orange and purple points; and two outliers, the green and blue points.

has two large clusters and two clear outliers, which we expect to be able to detect. The result of running the detector with this set can be seen in Figure 4.2. On the lower left we see a lot of orange and purple dots. These reflect the fact that in the early stages of the

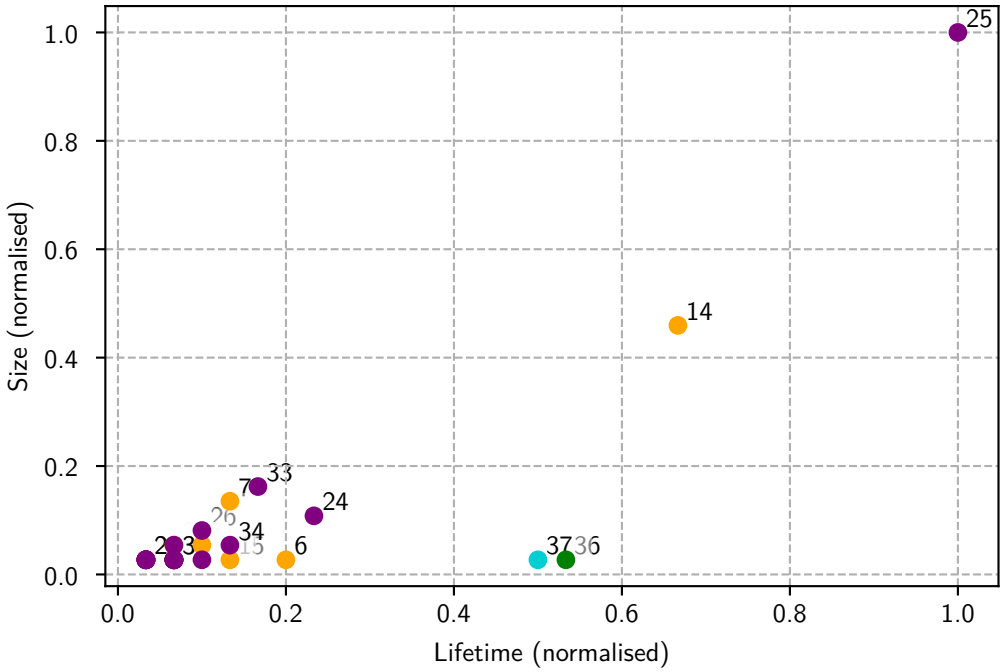

Figure 4.2: The result of running the detector on the point cloud of Figure 4.1. Both outliers are clearly visible as such, since they lie in the lower right of the plot, as is to be expected.

filtration, most of the classes in the clusters will quickly merge with their neighbours so they never reach large sizes. Then, once the two main clusters have become connected, no more deaths happen for some time. Then, roughly at the halfway point of the filtration, both outliers get absorbed into the two main clusters, as the blue and green dots represent. Then we see the death of the orange cluster as it merges into the purple cluster. In this situation, it makes sense that both outliers are absorbed first, since they are closer to the main clusters than the two main clusters are to each other. Finally, the purple cluster dies as the filtration ends, having absorbed every point.

4.2 A test with real radiomic signatures

A preliminary study of TOPiomics was an investigation into which radiomic features are best correlated to the response to a certain treatment [Lig+19]. Out of the whole range of features computed by PyRadiomics, 26 were found to have the highest correlation. The data from this study was kindly provided by VHIO to test the detector. It consists of the radiomic signature of 20 patients, as well as data related to the their genomics. The detector was run on both the radiomic and genomic feature spaces to see if there is a relationship between the clustering structure in both. The results can be seen in Figures 4.3 and 4.4. Points labeled with the same number represent the same patient.

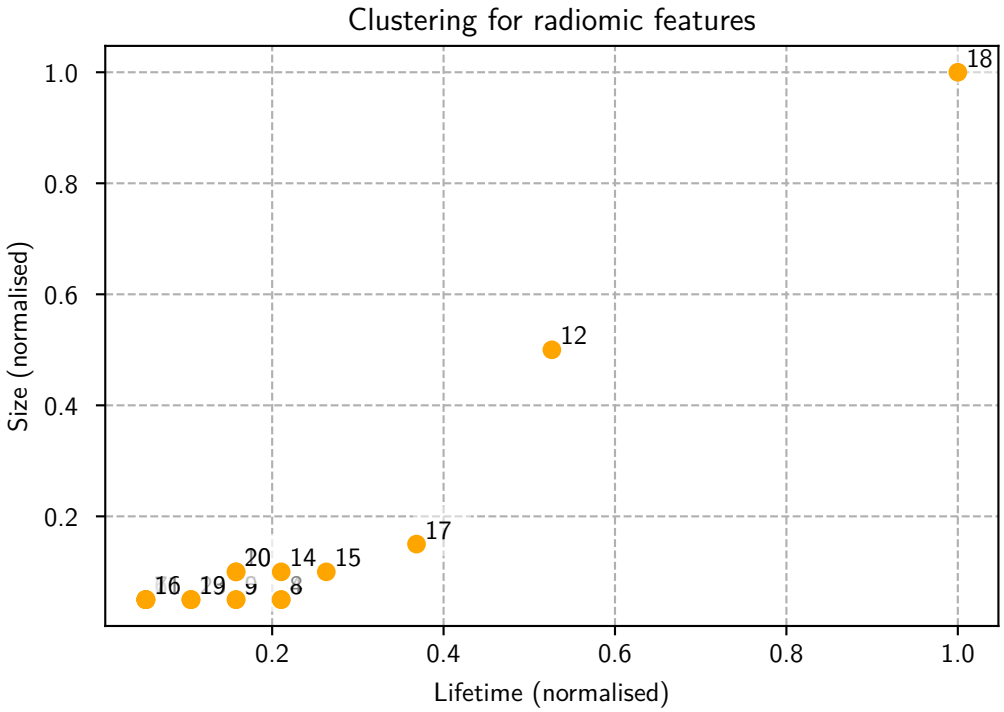

Figure 4.3: The clustering information obtained from the outlier detector on the test radiomic data.

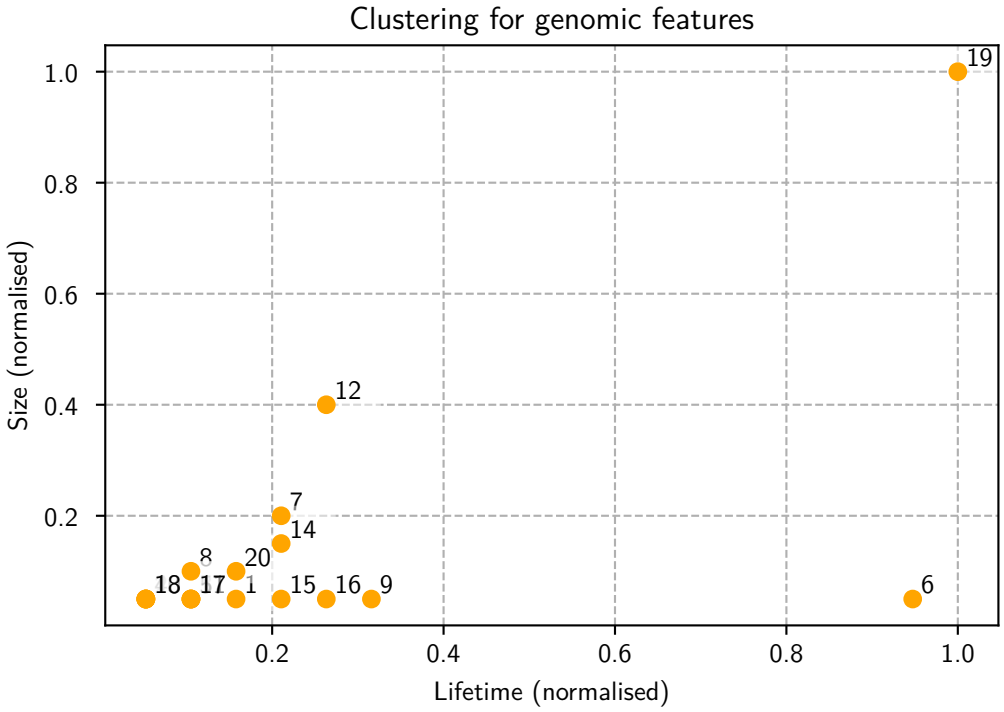

Figure 4.4: The clustering information obtained from the outlier detector on the test genomic data.

In the radiomic feature space we see a lot of clustering early on, and then a large cluster (12) which dies halfway through. This seems to indicate two main clusters. However no clear outliers can be seen. In the genomic space the situation is somewhat similar, with again a lot of clustering early on and a somewhat large cluster (12) that dies a little later, which again seems to indicate two main clusters. We also see a very clear outlier (6), that is not absorbed into the main cluster until the very end of the filtration. 9 could also be seen as an outlier, since it is absorbed into the main cluster after 12, but it does not persist for very long.

These results have been sent back to VHIO to determine what is the medical significance, if any, that they have.

Bibliography

[ALS12] D. Attali, A. Lieutier, and D. Salinas. "Efficient Data Structure for Representing and Simplifying Simplicial Complexes in High Dimensions". In: *International Journal of Computational Geometry & Applications* 22.04 (Aug. 2012). DOI: 10.1145/1998196.1998277.

[Cam18] Martín Campos. "Filtraciones en homología persistente mediante estimadores kernel de densidad". Universitat Autònoma de Barcelona, Facultat de Matemàtiques, June 2018.

[EH08] H. Edelsbrunner and J. Harer. "Persistent homology—a survey". In: *Discrete & Computational Geometry - DCG* 453 (Jan. 2008). DOI: 10.1090/conm/453/08802. URL: https://pub.ist.ac.at/~edels/Papers/2008-B-02-PersistentHomology.pdf.

[EM12] H. Edelsbrunner and D. Morozov. "Persistent homology: theory and practice". In: *Proceedings of the European Congress of Mathematics*. (July 2–7, 2012). Europ. Soc. of Mathematics. Krakow, Jan. 2012, pp. 31–50. URL: https://pub.ist.ac.at/~edels/Papers/2012-P-11-PHTheoryPractice.pdf.

[EM16] H. Edelsbrunner and D. Morozov. "Persistent Homology". In: *Handbook of Computational and Discrete Geometry, 3rd ed.* Ed. by J. E. Goodman, J. O'Rourke, and C. D. Tóth. Boca Raton, FL: CRC Press, 2016. Chap. 24. URL: https://pub.ist.ac.at/~edels/Papers/2016-B-01-PersDM.pdf.

[Ghr14] Robert Ghrist. *Elementary Algebraic Topology*. 1st ed. Createspace, 2014. URL: https://www.math.upenn.edu/~ghrist/notes.html.

[Gil19] D. Gil. *Topological Radiomics (TOPiomics): Early Detection of Genetic Abnormalities in Cancer Treatment Evolution*. The ATTRACT Project. 2019. URL: https://attract-eu.com/selected-projects/topological-radiomics-topiomics-early-detection-of-genetic-abnormalities-in-cancer-treatment-evolution.

[Gri+17] J. J. M. van Griethuysen et al. “Computational radiomics system to decode the radiographic phenotype”. In: *Cancer Research* 77.21 (2017), pp. 104–107. DOI: 10.1158/0008-5472.can-17-0339.

[GRP19] D. Gil, O. Ramos, and R. Perez. “Topological Radiomics (TOPiomics): Early Detection of Genetic Abnormalities in Cancer Treatment Evolution”. In: *Women in Geometry and Topology 2019, Abstracts Book*. (Sept. 25–27, 2019). Centre de Recerca Matemàtica. Bellaterra, 2019, pp. 28–30. URL: http://www.crm.cat/en/Activities/Curs_2019-2020/Documents/WomenInTopologyBooklet.pdf.

[Hat01] Allen Hatcher. *Algebraic Topology*. Cambridge University Press, 2001. URL: pi.math.cornell.edu/~hatcher/AT/AT.pdf.

[Lig+19] M. Ligero et al. “Selection of Radiomics Features based on their Reproducibility”. In: *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. (July 23–27, 2019). IEEE Engineering in Medicine and Biology Society. Berlin, 2019, pp. 403–408.

[RBG20] O. Ramos, A. Berenguel, and D. Gil. “A flexible outlier detector based on a topology given by graph communities”. Preprint. Feb. 2020. URL: https://arxiv.org/pdf/2002.07791.

[Rie16] Emily Riehl. *Cateogry Theory in Context*. Dover Publications, 2016. URL: http://www.math.jhu.edu/~eriehl/context.pdf.

[Zom10] A. Zomorodian. “The tidy set: A minimal simplicial set for computing homology of clique complexes”. In: *Proceedings of the 26th Annual Symposium on Computational Geometry*. (June 13–16, 2010). ACM. Snowbird, UT, 2010.