Topology, Metrics and Data: Computational Methods and Applications

by

Gabriele Beltramo

A thesis submitted to the University of London for the degree of
Doctor of Philosophy

School of Mathematical Sciences
Queen Mary, University of London
United Kingdom

March 2021

Abstract

The field of topological data analysis (TDA) combines computational geometry and algebraic topology notions for analyzing data. This thesis presents methods and efficient algorithms that extend the TDA toolset.

After introducing the needed background information about Euler characteristic curves and persistent homology, the former objects are extended to bi-dimensional filtrations. The result are Euler characteristic surfaces, which capture insights about data over a pair of parameters. Moreover, algorithms to compute these objects are described for both image and point data.

Persistent homology in ℓ_∞ metric is also studied. It is proven that in this setting Alpha and Čech filtration are not equivalent in general. On the other hand, two new filtrations — Alpha flag and Minibox — are defined and proven equivalent to Čech filtrations in homological dimensions zero and one. Algorithms for finding Minibox edges are described, and Minibox filtrations are empirically shown to speed up the computation of Čech persistence diagrams with computational experiments.

Then a new family of summary functions of persistence diagrams is defined, which is related to persistence landscapes. These are called cumulative landscapes and are used to vectorize the information contained in persistence diagrams. In particular, discretizations of these functions and their Fourier coefficients are used to obtain feature vectors that can be applied in supervised classification problems. The effectiveness of these feature vectors for the classification of data is compared against vectors obtained using persistence landscapes on two open-source datasets.

Finally, a novel method is described for the analysis of high-dimensional genomics data. Optimized metrics are defined on genomic vectors making use of a loss function. These are used in combination with a distance-based classification method, showing good performance compared to standard machine learning algorithms. Moreover, the structure of the given optimized metrics helps identify coordinates of the genomic vectors, which are most important for the classification task under study.

Acknowledgments

First of all, I would like to thank my primary supervisor, Professor Michael Farber, for giving me the opportunity to pursue a Ph.D. in a field about which I am very passionate and for always supporting me during these years. His incredible experience has been inspiring to me. I am grateful for all our fruitful conversations that helped me progress in my research.

I would also like to thank my second supervisor, Dr. Primoz Skraba, for all his guidance, his patience, and for sharing his knowledge with me. His feedback has been fundamental to my research and steered me in the right direction.

Thanks to Professor Trevor Graham and Dr. Kathleen Curtius for sharing with me their research cancer genomic data and for helping and supporting me in carrying out our research project.

Thanks to the School of Mathematical Sciences of Queen Mary University for believing in my project, and for supporting me financially during these years. I am also grateful for the incredible opportunity to join the Alan Turing Institute as part of the Enrichment Scheme during the second year of my degree.

I would also like to thank all the staff of the School Office for always being supportive and friendly in answering my many questions.

Finally, I would like to thank my family for always being there for me and for their encouragement throughout these years.

Table of Contents

Chapter 1

Introduction

The central objective of this thesis is to describe new computational methods for the analysis of datasets, including efficient algorithms for their application. This research is carried out in the framework of topological data analysis [Car09, EH10]. The focus is on using metric information on the data at hand to measure its geometric and topological features, which can then be used to define useful descriptors for its study. The increasing interest in this area is motivated by the growing amount of complex data that has become available in recent years, and the various application domains where topological data analysis has been shown to provide useful insights. These include collaboration networks [CH13], sensor networks [DSG07], neuroscience [BMM$^+$16, DHL$^+$16], robotics [BGK15], and many others.

Persistent homology is one of the main tools of topological data analysis. It studies the geometric and topological structure of datasets by combining concepts from the fields of computational geometry and algebraic topology. In particular, it defines persistence diagrams, which are multi-sets of points that compactly encode information about the "shape" of the data at hand over a range of parameters. To obtain persistence diagrams, functions defined on topological representations of data elements are used, which are often determined using metric information. The notion of persistence was independently developed by Frosini and Ferri [DFL03], Robins [Rob99], and the research group lead by Edelsbrunner [EKS83, EM94, ELZ02]. The property of persistence diagrams that makes them useful for real-world applications is their stability with respect to noise in the input data [CSEH07]. On the other hand, the high computational cost of the standard algorithm employed for their calculation represents a limit for their application. In cases where the size of data is an issue, it may be preferable to use Euler characteristic-based descriptors, for which more efficient algorithms are available [HW17].

The methods introduced in this work extend the field of topological data analysis. Euler characteristic curves are generalized to descriptors integrating the information of two filtrations. Then, filtrations are described for the computation of persistence diagrams of points in ℓ_∞ metric space. These new tools are applied to supervised classification problems on open-source data, through vectorizations obtained with a new type of summary function called cumulative landscape. Moreover, a technique is described for the analysis and classification of high-dimensional cancer genomic vectors, which uses optimized metrics on the given data.

The thesis is organized as follows. Chapter 2 gives a summary of the theoretical notions used throughout this work. The definitions of metric spaces and abstract complexes are stated for completeness. Filtrations of complexes are introduced and used to define Euler characteristic curves. This is the first descriptor presented and comes with an algorithm for its computation on image data. Next, the theory of persistent homology is summarized. The standard column algorithm for the computation of persistence diagrams is discussed, and an example is given to illustrate its application in practice. To conclude, some of the main types of proximity filtrations used in the field of persistent homology are introduced. These include Čech, Alpha, Delaunay-Čech, and Vietoris-Rips filtrations, which are compared in terms of their size and the persistence diagrams they produce.

In Chapter 3, the theory of multiparameter Euler characteristic descriptors is introduced. The goal is to extend to a bi-dimensional parameter space Euler characteristic curves, similarly to what was done in [CZ09] for multidimensional persistent homology. The obtained objects are called Euler characteristic surfaces, and novel algorithms are described for their computation for image and point data. The complexities of these algorithms are stated in Proposition 3.4.1 and Proposition 3.5.1. Moreover, several experiments, using both real and synthetically generated datasets, show that Euler characteristic surfaces can contain more information than multiple curves derived from the same data. Finally, we remark that the results and experiments discussed in this chapter are part of the preprint [BAG$^+$21].

Chapter 4 presents a number of results about the Čech persistent homology of points in ℓ_∞ metric space from [BS21]. The starting point is that Alpha and Čech filtration are equivalent in the Euclidean setting, as discussed in Section 2.4. However, the same proof technique does not work in the ℓ_∞ metric setting, and it is possible to show a counterexample to their equivalence using three-dimensional points. Nonetheless, it is possible to define two new types of filtrations — Alpha flag and Minibox — having the same persistence diagrams of Čech filtrations in homological dimensions zero and one. This equivalence is proven by means of Theorem 4.3.6 and Theorem 4.4.4, as well as using the properties of ℓ_∞-Delaunay edges. Both Alpha flag and Minibox filtrations are sequences of flag complexes, so they have the advantage of being fully determined by the edges they contain. On the other hand, Alpha filtrations in the Euclidean setting need all the simplices in the Delaunay complex of a set of points to be built. Furthermore, it is shown that for n randomly sampled points the number of Minibox edges on these points is proportional to $n \cdot \text{polylog}(n)$, while there are $\frac{n(n-1)}{2}$ Čech edges. Algorithms for finding Minibox edges in two, three, and higher dimensions are described, and used in computational experiments on random points. These show that the reduced number of simplices contained in Minibox filtrations helps decrease the time required and memory used to compute Čech persistence diagrams of points in ℓ_∞ metric space.

In Chapter 5 a new type of summary functions of persistence diagrams is introduced. These are called cumulative landscapes and are shown to preserve the information contained in persistence diagrams under genericity assumptions on the points of these latter objects. Moreover, cumulative landscapes can be used for vectorizing persistence diagrams using a single resolution parameter. The usefulness of the obtained vectors is evaluated on two supervised classification problems making use of open-source data. The average classification accuracy results of vectors obtained from Euler characteristic curves, Euler characteristic surfaces, persistence landscapes, and cumulative landscapes are compared. This comparison shows that in the case of image data Euler characteristic surfaces provide better results than any persistence diagram vectorization method. However, on both image and point data, Fourier coefficients of cumulative landscapes improve over the results of discretized persistence landscapes.

Finally, in Chapter 6 a classification problem is studied that involves high-dimensional cancer genomic vectors. The underlying assumption in this research is that genomic vectors are partitioned into two classes: those of low-risk and high-risk patients respectively. This information is used to define the loss function used to derive optimized metrics on a given dataset. These metrics are then used in combination with a distance-based classification method on the genomic vectors and compared against the average accuracy results obtained with logistic regression and nearest neighbourhoods classifiers. Experiments on both synthetically generated genomic and real-world vectors show that on this type of data the optimized distance-based classifier improves over the results of the standard machine learning algorithms mentioned above. Moreover, the local maximums of the weight functions used to define optimized metrics correspond to the coordinates of genomic vectors which are most informative for the classification problem at hand.

Chapter 2

Background

This chapter presents the basic definitions and results used throughout the thesis. After recalling metric spaces and definitions of abstract/geometric complexes, Euler characteristic curves are introduced together with an algorithm for their computation. Then an overview of persistent homology is given, followed by a discussion of the main types of proximity filtrations used in the field of topological data analysis. Several examples are included to illustrate Euler characteristic curves and the properties of persistence diagrams.

2.1 Preliminaries

To begin with, it is given a summary of the definitions and notation relative to metric spaces and abstract/geometric complexes.

Metric spaces. The general definition of metric is stated for completeness. The metrics used throughout this work are defined, followed by a discussion of balls and boxes derived from these.

Definition 2.1.1. A *metric* or *distance function* on a set X is a real-valued function $d_\bullet : X \times X \to \mathbb{R}$ such that for any $x, y, z \in X$:

(i) $d_\bullet(x, y) = 0$ if and only if $x = y$;

(ii) $d_\bullet(x, y) = d_\bullet(y, x)$;

(iii) $d_\bullet(x, z) \leq d_\bullet(x, y) + d_\bullet(y, z)$.

The pair $(X, d_\bullet)$ is a metric space.

Given $X = \mathbb{R}^d$ and $x = (x_1, x_2, \ldots, x_d), y = (y_1, y_2, \ldots, y_d) \in X$, we define three metrics:

- $d_1(x, y) = \sum_{i=1}^d |x_i - y_i|$;
- $d_2(x, y) = \sqrt{\sum_{i=1}^d (x_i - y_i)^2}$;
- $d_\infty(x, y) = \max_{i=1}^d |x_i - y_i|$.

We refer to d_2 as to the Euclidean metric, and to d_1 and d_∞ as the ℓ_1 and ℓ_∞ metrics respectively. Given a metric space $(\mathbb{R}^d, d_\bullet)$, the *open ball* of radius $r \geq 0$ and center $p \in \mathbb{R}^d$ is $B_r(p) = \{x \in \mathbb{R}^d \mid d_\bullet(x, p) < r\}$. The *closed ball* of radius $r \geq 0$ and center $p \in \mathbb{R}^d$ is denoted by $\overline{B_r(p)}$. The difference between the closed and open ball is the *boundary* $\partial\overline{B_r(p)}$. The ε-thickening of a set $A \subseteq \mathbb{R}^d$ is $\varepsilon(A) = \{p \in \mathbb{R}^d \mid \min_{a \in A} d_\bullet(a, p) \leq \varepsilon\}$, and from the definition of ball it follows $\varepsilon(B_r(p)) = B_{r+\varepsilon}(p)$. Moreover, a ball in ℓ_∞ metric space, i.e. $(\mathbb{R}^d, d_\infty)$, is the Cartesian product of d open intervals. This follows from the definition of d_∞ above, so that $B_r(p) = \prod_{i=1}^d I_i^p$, where $I_i^p = (p_i - r, p_i + r)$. As a *box* we refer to the Cartesian product of d intervals in $\mathbb{R}^d$, i.e. an axis-parallel hyperrectangle. Intervals and Cartesian products have the following properties.

- The intersection of a finite number of intervals is either empty or an interval.
- Cartesian products and intersections of a finite collection of sets commute, i.e. $(A \cap B) \times (C \cap D) = (A \times C) \cap (B \times D)$.

These follow from the definitions of intervals and Cartesian products, as given in [Mun00, Chapter 1], and allow us to derive the below properties of boxes, which we use in Chapter 4.

Proposition 2.1.2. *Let $\mathcal{B}$ be a finite collection of either open or closed boxes in $\mathbb{R}^d$.*

(i) *The intersection of the boxes in $\mathcal{B}$ is equal to the Cartesian product of the intersections of intervals defining these boxes. So this intersection is either empty or a box.*

(ii) *The intersection of the boxes in $\mathcal{B}$ is non-empty if and only if all the pairwise intersections of these boxes are non-empty.*

Proof. Given $\mathcal{B} = \{B^j\}_{j=1}^n$ and $B^j = \prod_{i=1}^d I_i^j$, if follows that

$$\bigcap_{j=1}^n B^j = \bigcap_{j=1}^n \prod_{i=1}^d I_i^j = \prod_{i=1}^d \bigcap_{j=1}^n I_i^j \tag{2.1}$$

from the properties of Cartesian products and intersections mentioned above. Point *(i)* follows because $\bigcap_{j=1}^{n} I_i^j$ is either empty or an interval for each $1 \leq i \leq d$. Thus by Equation (2.1) the intersection of all boxes is either empty or a box.

We prove the two directions of point *(ii)* separately. First, if $\bigcap_{j=1}^{n} B^j$ is non-empty, then also all pairwise intersections of boxes are non-empty because they contain this set. For the other direction, given that the pairwise intersections of boxes in $\mathcal{B}$ are non-empty, we need to show that $\bigcap_{j=1}^{n} B^j \neq \emptyset$. Equation (2.1) can be applied with $n = 2$ to any pair of boxes $B^{j_1}, B^{j_2} \in \mathcal{B}$, implying that $I_i^{j_1} \cap I_i^{j_2} \neq \emptyset$ for each $1 \leq i \leq d$ and $1 \leq j_1, j_2 \leq n$. Then, defined a_i' and b_i' as the maximum of left endpoints and the minimum of right endpoints of the intervals $\{I_i^j\}_{j=1}^{n}$ for each $1 \leq i \leq d$, it must be that $a_i' \leq b_i'$, otherwise the intervals realizing these minimum and maximum values would have an empty intersection. Thus $I_i' = (a_i', b_i') \subseteq I_i^j$ for each $1 \leq i \leq d$ and $1 \leq j \leq n$, by definition of a_i' and b_i', and $\bigcap_{j=1}^{n} I_i^j \supseteq I_i' \neq \emptyset$ for each $1 \leq i \leq d$. Finally $\bigcap_{j=1}^{n} B^j$ is non-empty because it contains $\prod_{i=1}^{d} I_i'$. □

Complexes. This thesis presents several results involving computational methods that can be applied to real-world data. Two of the main types of datasets to which these methods apply are finite point sets and images. These are dealt with by representing them as a collection of complexes, which are hierarchical structures of combinatorial/-geometric objects.

Definition 2.1.3. An *abstract complex* K is a finite collection of finite sets such that if $\tau \in K$ and $\sigma \subseteq \tau$, then $\sigma \in K$. A finite set in K is called a *simplex.* The dimension of a simplex is equal to its cardinality minus one. The dimension of K is the maximum dimension of any of its simplices.

Definition 2.1.4. Let K be an abstract complex.

- A subcollection of elements of K is a *subcomplex* if is itself a complex.
- The *closure* $\mathrm{Cl}(K')$ of a subcollection K' of elements of K is the smallest subcomplex of K containing K'.
- The *star* of of σ in K is $\mathrm{St}(\sigma) = \{\tau \in K \mid \sigma \subseteq \tau\}$. Note that the star of σ is not a complex in general, while $\mathrm{Cl}(\mathrm{St}(\sigma))$ is.
- The *link* of σ in K is $\mathrm{Lk}(\sigma) = \{\tau \in \mathrm{Cl}(\mathrm{St}(\sigma)) \mid \sigma \cap \tau = \emptyset\}$.

The definitions above are purely combinatorial. Because of this, abstract complexes can be translated into the matrices used by the algorithm of Section 2.3. However, the sets in K do not correspond to geometric objects that can be visualized. To model the structure of complexes derived from data, the following definitions are used instead.

Definition 2.1.5. The *convex hull* of a finite point set $S = \{p_i\}_{i=1}^n$ is the set of convex combinations of its points of the form $\alpha_1 p_1 + \alpha_2 p_2 + \ldots + \alpha_n p_n$, where $\alpha_i \geq 0$ for each i, and $\sum_{i=1}^n \alpha_i = 1$.

Definition 2.1.6. A *geometric simplex* σ is the convex hull of $k+1$ affinely independent points $\{p_i\}_{i=1}^{k+1} \subset \mathbb{R}^d$. The dimension of σ is k. A *face* σ' of σ is the convex hull of k points in $\{p_i\}_{i=1}^{k+1}$. The set of all faces of the simplex σ is its *boundary* $\partial\sigma$.

Geometric simplices of dimension from zero to three are vertices, edges, triangles, and tetrahedra. Higher-dimensional geometric simplices generalize these objects.

Definition 2.1.7. A *geometric simplicial complex* $|K|$ is a finite set of geometric simplices such that any face of a geometric simplex in $|K|$ is also in $|K|$, and the intersection of any two geometric simplices in $|K|$ is either empty or a geometric simplex of $|K|$. The dimension of $|K|$ is the maximum dimension of any of its geometric simplices.

Definition 2.1.8. An *elementary interval* is a subset $I \subset \mathbb{R}$ of the form $[n, n+1]$ or $[n, n]$, where $n \in \mathbb{Z}$. The second type of elementary intervals are said to be degenerate.

Definition 2.1.9. Let $\mathcal{I} = \{I_k\}_{i=1}^k$ be elementary intervals, of which $l \leq k$ are degenerate. The Cartesian product $C = \prod_{i=1}^k I_i$ is an *elementary cube* of dimension $k-l$. Given two elementary cubes C' and C such that $C' \subseteq C$, then C' is a *face* of C. The *boundary* ∂C of C is the set of all its faces.

A zero-dimensional elementary cube is a vertex, a one-dimensional elementary cube is an edge, a two-dimensional elementary cube a square, and a three-dimensional one a cube.

Definition 2.1.10. A cubical complex $|K|$ is a finite set of elementary cubes, such that the boundary of every elementary cube in $|K|$ is also in $|K|$. The dimension of $|K|$ is the maximum dimension of any of its elementary cubes.

Definition 2.1.11. Let K be an abstract complex. Given an embedding of the zero-dimensional simplices of K as an affinely independent set of points in some $\mathbb{R}^d$, the *geometric realization* $|K|$ of K is the collection of convex hulls of the embedded finite point sets corresponding to the simplices of K.

2.2 Euler Characteristic Curves

Given an abstract complex K, its structure can be used to define useful topological invariants [Hat02].

Definition 2.2.1. Let K be an abstract complex, and k_n denote the number of n-dimensional sets in K. The *Euler characteristic* of K is the alternating sum

$$\chi(K) = k_0 - k_1 + k_2 - k_3 + \dots \tag{2.2}$$

This is one of the invariants used in this work. It has the advantage of being efficient to compute algorithmically, as it only requires counts of elements of K.

Sublevel sets filtrations. In the context of topological data analysis [EH10, Chapter 7], the following objects are used to model real-world data parameterized by one real variable.

Definition 2.2.2. A *filtration* of an abstract complex K parameterized by $\mathcal{R}$ is a nested sequence of subcomplexes

$$K_{\mathcal{R}} = \{K_{r_0} \subseteq K_{r_1} \subseteq \dots \subseteq K_{r_m}\}, \tag{2.3}$$

where $\mathcal{R} = \{r_i\}_{i=0}^m$ is a monotonically increasing set of real values, and $K_{r_i} \subseteq K$ for each $0 \leq i \leq m$.

Definition 2.2.3. Let K be an abstract complex. A real-value function $h : K \to \mathbb{R}$ is a *filtering function* on K if for each $\sigma, \tau \in K$ such that $\sigma \subseteq \tau$, then $h(\sigma) \leq h(\tau)$.

The definition above ensures that the sublevel sets $h^{-1}((-\infty, r])$ for $r \in \mathbb{R}$, are subcomplexes of K. Moreover, given any pair of values $r_1, r_2 \in \mathbb{R}$, if $r_1 \leq r_2$, then $h^{-1}((-\infty, r_1]) \subseteq h^{-1}((-\infty, r_2])$, which allows for the following definition.

Definition 2.2.4. Let K be an abstract complex and $h : K \to \mathbb{R}$ a filtering function on K. The *sublevel sets filtation* of K induced by h on a sets of monotonically increasing real values $\mathcal{R} = \{r_i\}_{i=0}^m$ is the nested sequence of subcomplexes

$$K_{\mathcal{R}}^h = \left\{h^{-1}((-\infty, r_0]) \subseteq h^{-1}((-\infty, r_1]) \subseteq \dots \subseteq h^{-1}((-\infty, r_m])\right\}. \tag{2.4}$$

The idea is to associate a sublevel sets filtration to each element in a given dataset. The Euler characteristic, or some other invariant, of the complexes in these filtrations, can then be used to characterize the elements in the dataset [HW17].

Definition 2.2.5. Let $K_{\mathcal{R}}$ be a filtration of an abstract complex K on a set of monotonically increasing real values $\mathcal{R} = \{r_i\}_{i=0}^{m}$. The *Euler characteristic curve* of $K_{\mathcal{R}}$ is the vector of integer values

$$\mathrm{C}(K_{\mathcal{R}}) = \big[\chi(K_{r_0}), \chi(K_{r_1}), \ldots, \chi(K_{r_m})\big], \tag{2.5}$$

where $\chi(K_{r_i})$ is the Euler characteristic of the subcomplexes in the filtration of K.

Euler characteristic curves of images. A *gray-scale image* $M \in \mathbb{N}^{n_1 \times n_2}$ is a n_1-by-n_2 matrix of integer values in the range $[0, m] \subseteq \mathbb{Z}$, i.e. the element in position (s, t) of M is the pixel intensity $v_{s,t} \in [0, m]$. In the following discussion, we describe a method that can be used to obtain Euler characteristic curves out of images. This is extended to pair of images in Chapter 3, and applied to texture images in Chapter 5.

Given a gray-scale image M, its cubical complex $|K_M|$ is defined as the set of elementary squares $[s, s+1] \times [t, t+1]$ for each pixel $v_{s,t}$, together with all their faces and vertices. The abstract cubical complex K_M of M is the one whose geometric realization equals $|K_M|$, and where squares $[s, s+1] \times [t, t+1]$ correspond to sets $\sigma_{s,t}$.

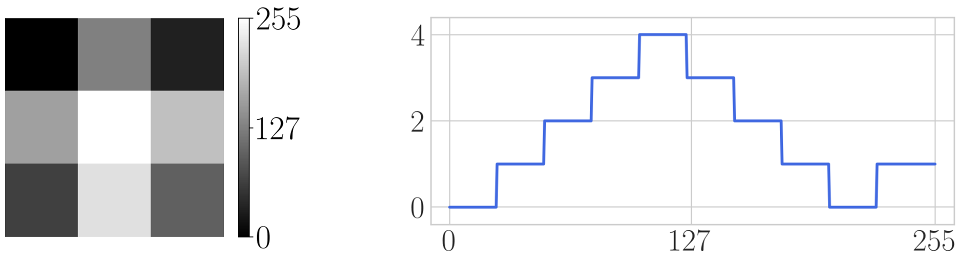
Definition 2.2.6. Let M be a gray-scale image and K_M its associated abstract cubical complex. The *pixel intensity filtering function* $h_M : K_M \to [0, m] \subseteq \mathbb{Z}$ of M is defined by setting $h_M(\sigma_{s,t}) = v_{s,t}$ for each $\sigma_{s,t} \in K$ and $h_M(\sigma') = \min_{\sigma_{s,t} \supseteq \sigma'} h_M(\sigma_{s,t})$ for each $\sigma' \subseteq \sigma_{s,t}$.

Definition 2.2.7. Let M be a gray scale image with values in $[0, m]$ and h_M its pixel intensity filtering function. Given the sublevel sets filtration $K_{\mathcal{R}}^{h_M}$, with $\mathcal{R}$ consisting of the integer values in $[0, m]$, the *Euler characteristic curve* C_M of M is defined as $\mathrm{C}(K_{\mathcal{R}}^{h_M})$, which is a vector of $m+1$ integers whose i-th entry is $\chi\big(h_M^{-1}([0, i-1])\big)$.

The above definition of Euler characteristic curve of an image is illustrated with an example. Given the matrix

$$M = \begin{pmatrix} 25 & 125 & 50 \\ 150 & 225 & 175 \\ 75 & 200 & 100 \end{pmatrix}, \tag{2.6}$$

which is as a gray-scale image with values in $[0, 255]$, Figure 2.1 displays it and gives a plot of its Euler characteristic curve vector C_M in the form of a piecewise constant continuous curve with domain $[0, 255]$. A visualization of the sublevel sets filtration $K_{\mathcal{R}}^{h_M}$ of the abstract simplicial complex K_M of M is given in Figure 2.2. This shows only the complexes for the values at which new elementary squares are added. Note that the



(a)

(b)

Figure 2.1: **(a)** Gray-scale image of example matrix M. **(b)** Euler characteristic curve of image in (a).

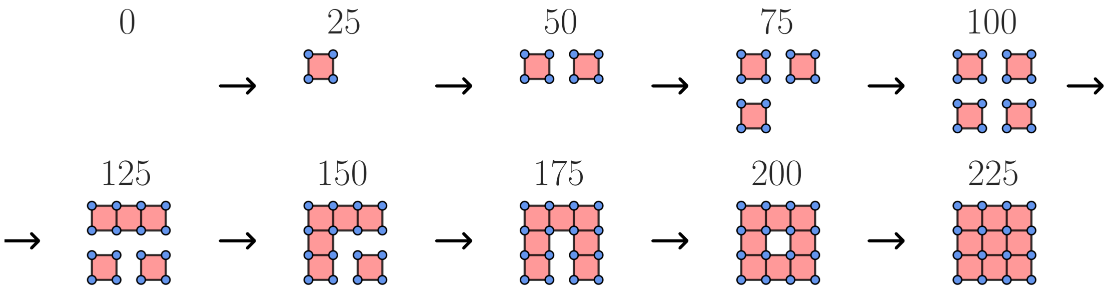


Figure 2.2: Sublevel sets filtration used to obtain the Euler characteristic curve in Figure 2.1b.

Algorithm 2.1 Euler characteristic curve of images.

```
Input: image matrix $M$ and range $[0, m]$.

$\mathrm{C}_M \leftarrow$ zeros array of length $m + 1$
for $v_{s,t}$ in $M$ do
    for each face $\sigma \in K_M$ introduced at value $v_{s,t}$ in $K_{\mathcal{R}}^{h_M}$ do
        $\mathrm{C}_M[v_{s,t}] = (-1)^{dim(\sigma)}$
    end for
end for
$\mathrm{C}_M = \left[\mathrm{C}_M[0], \mathrm{C}_M[0] + \mathrm{C}_M[1], \ldots, \sum_{i=0}^{m} \mathrm{C}_M[i]\right]$
return $\mathrm{C}_M$
```

Euler characteristic χ of the complexes in Figure 2.2 equals the number of connected components in the complexes minus the number of holes they contain.

Algorithm. Let K_M be the abstract cubical complex of an image M, and $h_M : K_M \to [0, m]$ its filtering function. The Euler characteristic curve of M can be computed with Algorithm 2.1, where $dim(\sigma)$ stands for the dimension of σ. This has $O(n + m)$

complexity, where n is the number of pixels in M and m the number of pixel intensity values. The $O(m)$ contribution to this complexity comes from the cumulative sum on line 7 of Algorithm 2.1. A full discussion of Algorithm 2.1, including computational experiments and its streaming version generalizing the input to d-dimensional images, can be found in [HW17].

In Chapter 3 Euler characteristic curves are generalized to objects encoding the information provided by a pair of filtering functions, and algorithms are described for their computation both for image and point data. This way the elements in a dataset can be characterized based on multiple features at the same time. For example, in the case of image data, pixel intensities and the values of a gradient on the image can be used. For point data, distances between points and estimates of local densities can be combined.

2.3 Persistent Homology

In the previous section the concept of filtration, Definition 2.2.2, was introduced to be used in conjunction with abstract cubical complexes derived from image data, and obtain a vector of Euler characteristic numbers. In this section, a topological invariant of abstract simplicial complexes is introduced. The idea is again to characterize the elements in a dataset with the way this invariant changes on the subcomplexes of a filtration. This way the structural information of these elements is compactly encoded on a range of parameters. The result of this procedure is a set of so-called *persistence diagrams* for each element in the dataset. In this work, these objects are going to be primarily applied to filtrations defined on finite point sets in $\mathbb{R}^d$. The different ways of associating such a filtration to a finite set of points are described in Section 2.4. Here the focus is on the theory underlying persistence diagrams, their properties, and the way they are computed.

Simplicial Homology. The Euler characteristic of K gives a summary of its structure based on a combination of connected components and holes, as it was observed in the example given in the previous section. Homology is a more powerful invariant, which distinguishes between connectedness and “holes” in different dimensions. The following discussion reviews the basics of simplicial homology theory. Additional information can be found in [Hat02].

Definition 2.3.1. An *oriented k-simplex* $[\sigma]$ is a k-dimensional simplex σ with an ordering of its elements such that two orientations are equal if the two underlying orderings differ by an even permutation.

Definition 2.3.2. Let K be an abstract simplicial complex. A *k-chain* c is a formal sum of oriented k-simplices, i.e. $c = \sum_i \alpha_i[\sigma_i]$, where the α_i are coefficients in a field $\mathbb{F}$ and the $[\sigma_i]$ are the oriented k-simplices of K.

Definition 2.3.3. The *group of k-chains* C_k of an abstract simplicial complex K is the set of its k-chains together with the addition operation defined by $c_1 + c_2 = \sum_i(\alpha_i^1 + \alpha_i^2)[\sigma_i]$ for any pair of k-chains $c_1 = \sum_i \alpha_i^1[\sigma_i]$ and $c_2 = \sum_i \alpha_i^2[\sigma_i]$.

Remark. Because of the use of coefficients in a field $\mathbb{F}$, the chain groups defined above are vector spaces. Using $\mathbb{F}$ instead of $\mathbb{Z}$ is required for defining persistence diagrams on K.

Definition 2.3.4. Let $[\sigma] = [p_1, p_2, \ldots, p_{k+1}]$ denote an ordered k-simplex in K, and $-[p_1, p_2, \ldots, p_{k+1}]$ the same simplex with orientation reversed. The *boundary operator* of σ is

$$\partial_k(\sigma) = \sum_{i=1}^{k+1}(-1)^{i+1}[p_1, \ldots, \hat{p_i}, \ldots, p_{k+1}], \tag{2.7}$$

where $[p_1, \ldots, \hat{p_i}, \ldots, p_{k+1}]$ is the oriented face of σ with $\hat{p_i}$ missing.

The boundary operator is an homomorphism of k-chains into $(k-1)$-chains, because $\partial_k(c_1 + c_2) = \partial_k c_1 + \partial_k c_2$. Moreover, an important property of the boundary operator [Hat02, Lemma 2.1] is that the composition $\partial_k\partial_{k+1} : C_{k+1} \to C_k \to C_{k-1}$ is the zero homomorphism for each $k \in \mathbb{N}$. Defined *k-cycles* $Z_k(K) = \mathrm{Ker}(\partial_k)$, and *$k$-boundaries* $B_k(K) = \mathrm{Im}(\partial_{k+1})$, the mentioned property of the boundary operator guarantees that $B_k(K) \subseteq Z_k(K)$ for each $k \geq 1$. In case $k = 0$, it is assumed that $\partial_0 : C_0 \to 0$ is the zero homomorphism, so that $B_0(K) \subseteq Z_k(K) = C_0$.

Definition 2.3.5. The *k-th homology group* of K is the quotient group

$$H_k(k) = \frac{Z_k(K)}{B_k(K)}. \tag{2.8}$$

The *k-th Betti number* $\beta_k(K)$ is the rank on $H_k(K)$. Elements of $H_k(K)$ are called *homology classes*, and two k-cycles mapped into the same homology class by the quotient operation are said to be *homologous*.

Remark. In case it is defined $\partial_0 : C_0 \to \mathbb{F}$ by setting $\partial_0(\sum_i \alpha_i\sigma_i) = \sum_i \alpha_i$, the quotient groups above are called the *reduced homology groups* $\tilde{H}_k(K)$ of K.

The zeroth Betti number $\beta_0(K)$ corresponds to the number of connected components of the geometric realization $|K|$ of K [Arm13, Theorem 8.2]. Moreover, if $|K|$ is homeomorphic to the unit n-sphere $S^n = \{x \in \mathbb{R}^{n+1} : ||x||_2 = 1\}$, then $\beta_k(K) = 0$ for $1 \leq k \leq n-1$, and $\beta_n(K) = 1$. In general, $\beta_n(K)$ equals the number of n-dimensional holes in the geometric realization of K. Betti numbers and the Euler characteristic of a complex K are related by the following equation, as stated by Theorem 2.44 in [Hat02].

$$\chi(K) = \sum_{k=0}^{\infty} (-1)^i \beta_k(K). \tag{2.9}$$

Persistence Diagrams. The information captured by homology groups, and in particular their rank, is useful for characterizing K. Furthermore, Equation (2.9) ensures that Betti numbers provide more information than Euler characteristic. Section 2.4 presents various ways of associating a finite set of points S to a sequence of abstract simplicial complexes, i.e. a filtration. The idea is that the complexes in a filtration encode the topological and geometric structure of S at different scales. A possible strategy for characterizing filtrations is to compute the Betti numbers of their subcomplexes. This would be similar to how Euler characteristic curves are associated to K in Section 2.2. The approach presented here captures even more information by tracking how homology classes appear and disappear in the given filtration, i.e. how long they persist. The following discussion summarizes the main definitions and results of the theory of persistent homology. An expanded discussion of these concepts can be found in [EH10, Chapter 7].

Definition 2.3.6. Let $K_{\mathcal{R}}$ be a filtration of an abstract simplicial complex K, where $\mathcal{R} = \{r_i\}_{i=0}^{m}$ is a monotonically increasing set of real values. The *k-th persistence* module of $K_{\mathcal{R}}$ is

$$\mathbb{M}_k(K_{\mathcal{R}}) = \left\{ H_k(K_{r_0}) \to H_k(K_{r_1}) \to \ldots \to H_k(K_{r_m}) \to H_k(K_{r_{m+1}}) \right\}, \tag{2.10}$$

where $r_{m+1} = +\infty$ and $K_{+\infty} = K$.

Because homology was defined with coefficients in a field $\mathbb{F}$, persistence modules can be put in bijection with sets of intervals on the values $\{r_i\}_{i=0}^{m} \cup \{+\infty\}$. This result is presented as given in [Oud15, Chapter 1], for the case of finite persistence modules containing only finite-dimensional homology groups.

Theorem 2.3.7. *Every persistence module $\mathbb{M}_k(K_{\mathcal{R}})$ is decomposable as a direct sum*

$$\mathbb{M}_k(K_{\mathcal{R}}) = \bigoplus_{l \in L} \mathbb{I}_k^l[r_i, r_j] \tag{2.11}$$

where $\mathbb{I}_k^l[r_i, r_j]$ is the indecomposable interval module

$$\overbrace{0 \xrightarrow{0} 0 \cdots 0}^{[r_1,r_{i-1}]} \xrightarrow{0} \overbrace{\mathbb{F} \xrightarrow{1} \mathbb{F} \cdots \mathbb{F} \xrightarrow{1} \mathbb{F}}^{[r_i,r_{j-1}]} \xrightarrow{0} \overbrace{0 \xrightarrow{0} 0 \cdots 0 \xrightarrow{0} 0}^{[r_j,r_{m+1}]} \tag{2.12}$$

Moreover, the decomposition in Equation (2.11) *is unique up to isomorphism and permutation of its terms.*

The proof of this theorem follows from the Krull-Remak-Schmidt principle, and Gabriel's theorem [Gab72] applied to the special case of persistence modules. Each indecomposable interval $\mathbb{I}_k^l[r_i, r_j]$ represents an homology class $[\gamma]$ *created* at r_i and *deleted* at r_j. A simplex σ_i added going from $K_{r_{i-1}}$ to K_{r_i}, that creates a k-cycle representing $[\gamma]$, is a *positive* simplex. On the other hand, a simplex τ_j added going from $K_{r_{j-1}}$ to K_{r_j}, that creates a k-boundary of $[\gamma]$, is a negative simplex. Together (σ_i, τ_j) form a *persistence pair*, and the *persistence* of $[\gamma]$ is $r_j - r_i$. This difference quantifies the importance of the connected component/k-hole represented by $[\gamma]$ in $K_{\mathcal{R}}$.

Definition 2.3.8. Let $K_{\mathcal{R}}$ be a filtration and $\mathbb{M}_k(K_{\mathcal{R}})$ its persistence module. The *k-th persistence diagram* of $K_{\mathcal{R}}$ is the multi-set of points

$$\mathrm{Dgm}_k(K_{\mathcal{R}}) = \left\{(r_i, r_j) \in \overline{\mathbb{R}}^2 \mid \mathbb{I}_k^l[r_i, r_j] \text{ is indecomposable interval of } \mathbb{M}_k(K_{\mathcal{R}})\right\}, \quad (2.13)$$

where $\overline{\mathbb{R}}^2 = (\mathbb{R} \cup \{+\infty\})^2$ is the extended plane.

Given a d-dimensional abstract complex K, any parameterized filtration $K_{\mathcal{R}}$ has non-trivial persistence diagrams in homological dimensions $0 \leq k \leq d-1$. This collection of multisets of points encodes the information about creation and deletion of connected components and k-holes in $K_{\mathcal{R}}$.

Persistent Homology Algorithm. Given a sublevel set filtration $K_{\mathcal{R}}^h$ of a complex K, its k-th persistence diagram can be computed by reducing a matrix $D_k \in (\mathbb{F})^{m_k \times m_{k+1}}$, where m_k is the number of k-simplices in K for any $k \geq 0$. To obtain this matrix, the k and $(k+1)$-simplices of K are first sorted on their h values, i.e. $\sigma_i \prec \sigma_j$ if $h(\sigma_i) \leq h(\sigma_j)$ for $\sigma_i, \sigma_j \in K$. Ties are broken arbitrarily. The result are the sorted lists of k-simplices $(\sigma_0, \sigma_1, \ldots, \sigma_{m_k-1})$ and $(k+1)$-simplices $(\tau_0, \tau_1, \ldots, \tau_{m_{k+1}-1})$. The elements of the matrix D_k are defined by setting

$$D_k[i][j] = \begin{cases} 1 & \text{if } \sigma_i \in \partial_{k+1}([\tau_j]), \\ -1 & \text{if } -\sigma_i \in \partial_{k+1}([\tau_j]), \\ 0 & \text{otherwise}, \end{cases} \quad (2.14)$$

for each $0 \leq i \leq m_k - 1$ and $0 \leq j \leq m_{k+1} - 1$. This way the j-th column of D_k represents the boundary of τ_j.

We present the standard persistent homology algorithm [DSMVJ11]. Its pseudocode

Algorithm 2.2 Standard persistent homology algorithm.

```
Input: matrix $D_k$.

$R_k \leftarrow D_k$
for $j = 0$ to $m_{k+1} - 1$ do
    while $\exists\, j' < j$ such that $low_{R_k}(j') = low_{R_k}(j)$ do
        $c \leftarrow R_k[low_{R_k}(j)][j] / R_k[low_{R_k}(j')][j']$
        for $i = 0$ to $m_k - 1$ do
            $R_k[i][j] = R_k[i][j] - c \cdot R_k[i][j']$
        end for
    end while
end for
return $R_k$
```

is given in Algorithm 2.2. This outputs an upper triangular matrix R_k iterating on the columns of D_k from left to right. It makes use of $low_{R_k}(j)$, which is the row index of the lowest non-zero element in the j-th column of R_k, or is undefined if this column contains only zeros. The matrix R_k is characterized by the fact of being reduced, i.e. any two non-zero columns j_1 and j_2 are such that $low_{R_k}(j_1) \neq low_{R_k}(j_2)$, and of being obtained with column operations from left to right. Moreover, R_k defines a collection of pairs $\{(i, j) : i = low_{R_k}(j)\}$, which the Pairing Lemma of [EH10, Chapter 7] ensures to be independent of the final reduced form of D_k. These pairs of indices correspond to pairs of simplices (σ_i, τ_j) representing the creation and deletion of a k-homology class, and so to a point $(h(\sigma_i), h(\tau_j))$ in $\mathrm{Dgm}_k(K_{\mathcal{R}})$. Besides, if row i in D_k does not contain the $low_{R_k}(j)$ element of any column j, and either $k = 0$ or column i in D_{k-1} contains only zeros, then σ_i represents a k-homology class created at $h(\sigma_i)$ that is never deleted, and so a point $(h(\sigma_i), +\infty)$ in $\mathrm{Dgm}_k(K_{\mathcal{R}})$.

Algorithm 2.2 has a worst-case running time of $O(m_k m_{k+1}^2)$, as it loops twice of the columns of R_k and once on its rows. To obtain the persistence diagrams of $K_{\mathcal{R}}$ up to homological dimension k, the matrices D_0, D_1, ..., D_k need to be reduced with Algorithm 2.2. Then the points in the persistence diagrams are derived using the filtering function h as described above.

A substantial amount of work has been done to improve the computational complexity of persistent homology algorithm, with a large number of results [BKR14a, BKR14b, CK11, DSMVJ11, MN13, WCV12] which have greatly sped up computations in practice [OPT$^+$17]. For instance, in [MMS11] the complexity of Algorithm 2.2 it is improved to $O(n^w)$ for the computation of zigzag persistent homology, where n is the number of simplices of K and $w \approx 2.376$ if using the Coppersmith–Winograd algorithm for

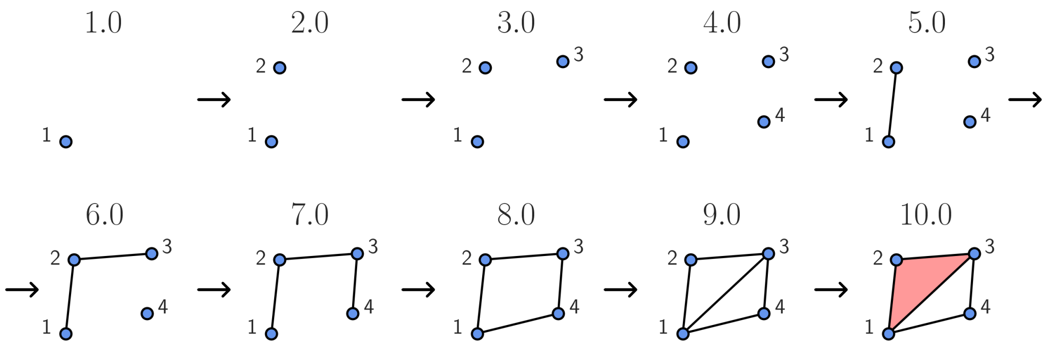


Figure 2.3: Filtration of a complex on four vertices. The h value of the simplex added at each step is shown above each K_{r_i}.

matrix multiplication [CW90]. However, it has also been observed that smaller complexes generally result in faster computation. For example, Alpha filtrations are introduced in Section 2.4 to reduce the number of simplices that need to be taken into consideration for the computation of the Čech persistence diagrams of points in Euclidean metric space. Similarly, the filtrations described in Chapter 4 help in reducing the size of Čech filtrations for points in ℓ_∞ metric space.

To conclude this discussion of the standard persistent homology algorithm, it is worth mentioning that the field of coefficients $\mathbb{F}$ is often assumed to be $\mathbb{Z}_2 = \mathbb{Z}/2\mathbb{Z}$. For instance, this is the case for the original persistent homology algorithm described in [ELZ02]. Furthermore, using $\mathbb{Z}_2$ allows to ignore orientations of simplices. Thus it is not necessary to compute c on line 4 of Algorithm 2.2, and the elements of two columns of R_k can be summed modulo two arithmetic.

Example: Persistence diagrams of filtration on four vertices. Let $K_{\mathcal{R}}$ be the filtration containing four vertices, five edges, and one triangle in Figure 2.3. These simplices are parameterized by values from 1 to 10. In particular the list of sorted vertices is ([1], [2], [3], [4]) with filtrations values (1, 2, 3, 4), the list of sorted edges ($[1,2]$, $[2,3]$, $[3,4]$, $[1,4]$, $[1,3]$) with filtration values (5, 6, 7, 8, 9), and the list of sorted triangles ([1, 2, 3]) with value (10). Using coefficients in $\mathbb{Z}_2$, these result in the matrices

$$D_0 = \begin{array}{cccccc} & [1,2] & [2,3] & [3,4] & [1,4] & [1,3] \\ [1] & 1 & 0 & 0 & 1 & 1 \\ [2] & 1 & 1 & 0 & 0 & 0 \\ [3] & 0 & 1 & 1 & 0 & 1 \\ [4] & 0 & 0 & 1 & 1 & 0 \end{array} \quad \text{and} \quad D_1 = \begin{array}{cc} & [1,2,3] \\ [1,2] & 1 \\ [2,3] & 1 \\ [3,4] & 0 \\ [1,4] & 0 \\ [1,3] & 1 \end{array},$$

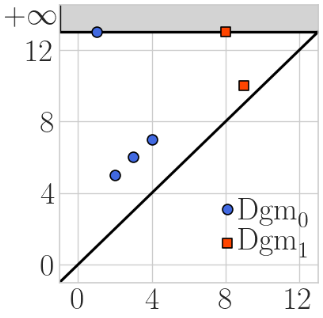


Figure 2.4: Persistence diagrams of filtration in Figure 2.3.

where rows and columns are labeled with their corresponding simplex. Algorithm 2.2 can be used to reduce D_0, obtaining

$$R_0 = \begin{array}{c} \\ [1] \\ [2] \\ [3] \\ [4] \end{array} \begin{array}{ccccc} [1,2] & [2,3] & [3,4] & [1,4] & [1,3] \\ \end{array}$$

$$R_0 = \begin{array}{c|ccccc} & [1,2] & [2,3] & [3,4] & [1,4] & [1,3] \\ \hline [1] & 1 & 0 & 0 & 0 & 0 \\ [2] & 1 & 1 & 0 & 0 & 0 \\ [3] & 0 & 1 & 1 & 0 & 0 \\ [4] & 0 & 0 & 1 & 0 & 0 \end{array}.$$

On the other hand, $D_1 = R_1$ is already reduced. The pairs $i = low_{R_k}(j)$ of the reduced matrices give the pairs of simplices prducing the persistence diagrams in homological dimensions zero and one of $K_{\mathcal{R}}$, which are displayed in Figure 2.4. Note that $\mathrm{Dgm}_0(K_{\mathcal{R}})$ contains one point at infinity, corresponding to the connected component created by [1] at filtration value 1.0. Similarly, $\mathrm{Dgm}_1(K_{\mathcal{R}})$ contains $(8.0, +\infty)$, which corresponds to the 1-cycle created by $[1, 4]$.

Bottleneck distance and stability. In order for persistence diagrams to be used to distinguish between elements of a dataset, a notion of dissimilarity between diagrams needs to be introduced.

Definition 2.3.9. Let $K^1_{\mathcal{R}}$ and $K^2_{\mathcal{R}}$ be two parameterized filtrations, and $\Delta = \big\{(x, x) \in \overline{\mathbb{R}}^2$ with infinite multiplicity : $x \in \mathbb{R} \cup \{+\infty\}\big\}$ the diagonal counted with infinite multiplicity. The *bottleneck distance* between the persistence diagrams of these filtrations is

$$d_B\big(\mathrm{Dgm}_k(K^1_{\mathcal{R}}), \mathrm{Dgm}_k(K^2_{\mathcal{R}})\big) = \inf_{\eta: X \to Y} \sup_{x \in X} d_\infty(x, \eta(x)), \tag{2.15}$$

where the infimum is taken over the set of all possible bijections $\eta : X \to Y$, from $X = \mathrm{Dgm}_k(K^1_{\mathcal{R}}) \cup \Delta$ into $Y = \mathrm{Dgm}_k(K^2_{\mathcal{R}}) \cup \Delta$.

The bottleneck distance makes use of a matching η of the points in two diagrams. The diagonal Δ is added with infinite multiplicity so that X and Y have the same cardinality. These additional points can be thought as of features having persistence equal to zero. Moreover, for sublevel sets filtrations, bottleneck distance has the following important stability property, first described in [CSEH07].

Theorem 2.3.10 (Stability Theorem for Filtrations [EH10])**.** *Let K be a d-dimensional abstract simplicial complex and $h_1 : K \to \mathbb{R}$ and $h_2 : K \to \mathbb{R}$ two filtering functions. The persistence diagrams of the sublevel sets filtrations of h_1 and h_2 on a set of monotonically increasing real values $\mathcal{R}$ satisfy*

$$d_B\big(Dgm_k(K^1_{\mathcal{R}}), Dgm_k(K^2_{\mathcal{R}})\big) \leq \|h_1 - h_2\|_\infty, \tag{2.16}$$

for each $0 \leq k \leq d-1$, where $\|h_1 - h_2\|_\infty = \sup_{\sigma \in K} |h_1(\sigma) - h_2(\sigma)|$.

This guarantees that small changes in the filtering functions are reflected in small perturbations of the points of the persistence diagrams.

2.4 Proximity Filtrations

In Section 2.2 it is shown how to define a sequence of abstract cubical complexes given a gray-scale image M. In the following discussion are described various methods for defining a sequence of abstract simplicial complexes on a finite set of point S. These filtrations model the structure of S on a range of scales. In particular, a distance $d_\bullet$ on the points of S is used to define the different sublevel sets filtrations of this section. This way the proximity of points in S is reflected in the local connectedness and presence of k-holes in the filtrations subcomplexes K_{r_i}. Persistence diagrams can then be used to compactly encode this information.

Čech filtrations. The first type of filtration uses intersections of balls centered in the points of S to model the topological and geometric structure of this set of points. Its subcomplexes are an instance of the following general concept.

Definition 2.4.1. The *nerve* of a finite collection of open or closed sets $\{A_i\}_{i \in I}$ is the abstract simplicial complex

$$\mathrm{Nrv}(\{A_i\}_{i \in I}) = \Big\{\sigma \subseteq I \mid \bigcap_{i \in \sigma} A_i \neq \emptyset\Big\}. \tag{2.17}$$

Definition 2.4.2. Let S be a finite set of points in $(\mathbb{R}^d, d_\bullet)$. The *Čech complex* with radius r of S is

$$K_r^{\check{C}} = \Big\{\sigma \subseteq S \mid \bigcap_{p\in\sigma} \overline{B_r(p)} \neq \emptyset\Big\}. \tag{2.18}$$

Given $\mathcal{R} = \{r_i\}_{i=0}^m$ to be a finite set of monotonically increasing real values, the *Čech filtration* $K_{\mathcal{R}}^{\check{C}}$ of S is

$$K_{r_0}^{\check{C}} \subseteq K_{r_1}^{\check{C}} \subseteq \ldots \subseteq K_{r_m}^{\check{C}}. \tag{2.19}$$

We use closed balls to define Čech complexes for consistency with the definitions of Alpha flag and Minibox complexes in Chapter 4. Note that using either open or closed balls results in the same ordering of simplices of K in Čech filtrations. Thus the input matrix D_k of Algorithm 2.2 is unaffected by this choice, as well as the resulting Čech persistence diagrams. Moreover, the following version of the Nerve Theorem can be used to establish a connection between the topology of $K_r^{\check{C}}$ and the finite union $\bigcup_{p\in S} \overline{B_r(p)}$.

Theorem 2.4.3 (Theorem 10.7 [GGL95]). *Let X be a triangulable space and $\{A_i\}_{i\in I}$ a locally finite family of open subsets (or a finite family of closed subsets) such that $X = \bigcup_{i\in I} A_i$. If every non-empty intersection $A_{i_1} \cap A_{i_2} \cap \ldots \cap A_{i_t}$ is contractible, then X and the nerve $Nrv(\{A_i\}_{i\in I})$ are homotopy equivalent.*

Čech complexes have the following properties.

- The Čech complex $K_r^{\check{C}}$ and the union of closed balls $\bigcup_{p\in S} \overline{B_r(p)}$ are homotopy equivalent, by Theorem 2.4.3 applied to the finite family of closed balls $\{\overline{B_r(p)}\}_{p\in S}$.
- If $r \in \mathbb{R}$ is greater than the radius of the minimal enclosing ball of the points of S, then $K_r^{\check{C}}$ contains all the simplices on the points of S, that is to say $K_r^{\check{C}} = K^n$ the full complex on S, having $\binom{n}{k+1}$ k-simplices for $k \geq 0$ where $n = |S|$.
- The Čech filtration can be seen as a sublevel sets filtration of the full complex K^n on S. Its filtering function is $h_{\check{C}} : K^n \to \mathbb{R}$, defined by setting $h_{\check{C}}(\sigma) = \inf_{x\in\mathbb{R}^d} \max_{p\in\sigma} d_\bullet(x,p)$ for each $\sigma \in K^n$. Moreover, $h_{\check{C}}(\sigma)$ equals the smallest enclosing ball radius of σ.

Given $K_{\mathcal{R}}^{\check{C}}$, to produce the sorted list of k-simplices used by Algorithm 2.2 the smallest enclosing ball radiuses of simplices in $K_{r_m}^{\check{C}}$ need to be computed and sorted. For this preprocessing step the miniball algorithm of [Gär99] can be used. Finally, Čech persistence diagrams are computed up to some fixed homological dimension $\bar{k} \geq 0$. This requires to operate on $\binom{n}{\bar{k}+2}$ $(\bar{k}+1)$-dimensional simplices, i.e. $\Theta(n^{\bar{k}+2})$ simplices. Thus, the value of $\bar{k}$ is typically chosen to be less than or equal to 2, because of the complexity of the persistent homology algorithm and the number of simplices in Čech filtrations.

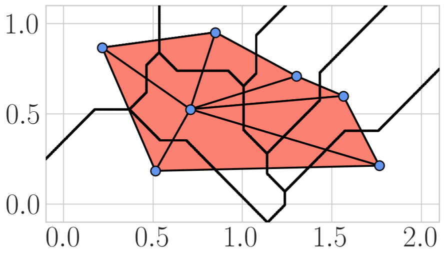


Figure 2.5: Boundaries of ℓ_∞-Voronoi regions of points in $\mathbb{R}^2$, and corresponding ℓ_∞-Delaunay triangulation.

Voronoi diagrams and Delaunay triangulations. The following discussion introduces geometric constructions that can be used to limit the number of simplices in Čech filtrations, and still obtain the desired persistence diagrams. In particular, Voronoi diagrams and Delaunay triangulations are defined for points in a general metric space. These have been extensively studied in computational geometry [dBCvKO08], primarily for Euclidean space. See [AKL13] for a reference for general Voronoi diagrams and Delaunay triangulations.

Definition 2.4.4. Let S be a finite set of points in $(\mathbb{R}^d, d_\bullet)$. The *Voronoi region* of a point $p \in S$ is

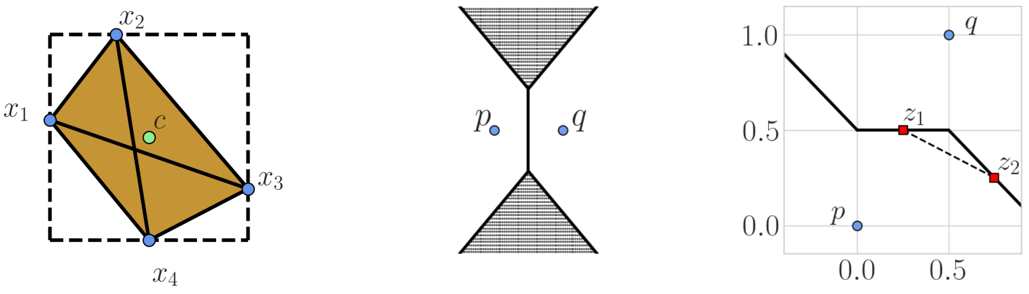
$$V_p = \Big\{ x \in \mathbb{R}^d \mid d_\bullet(p,x) \leq d_\bullet(q,x),\ \forall q \in S \Big\}. \tag{2.20}$$

The set of Voronoi regions $\{V_p\}_{p\in S}$ is the *Voronoi diagram* of S.

Definition 2.4.5. The *Delaunay complex* of a finite set of points $S \subseteq (\mathbb{R}^d, d_\bullet)$ is the abstract simplicial complex

$$K^D = \Big\{ \sigma \subseteq S \mid \bigcap_{p\in\sigma} V_p \neq \emptyset \Big\}. \tag{2.21}$$

In the remainder of this section, the focus is on ℓ_∞-Voronoi regions and ℓ_∞-Delaunay complexes, as their properties are used in Chapter 4. An example of such objects, for points in $\mathbb{R}^2$, is given in Figure 2.5. To begin with, it is worth noting that the structure of intersections of Voronoi regions defined using general polyhedral distances may be degenerate. For instance, as in the Euclidean case, $d+2$ points in $\mathbb{R}^d$ can have ℓ_∞-Voronoi regions with a non-empty intersection. This is illustrated by the four points in $\mathbb{R}^2$ of Figure 2.6a. Moreover, without assuming any hypothesis on S the intersection of two ℓ_∞-Voronoi regions can be a d-dimensional subset of $\mathbb{R}^d$. For example, given two



(a) (b) (c)

Figure 2.6: **(a)** Four points in $\mathbb{R}^2$ whose Delaunay complex is three-dimensional. **(b)** Degenerate intersection of ℓ_∞-Voronoi regions of two collinear points p and q, meaning that the segment $\overline{pq}$ is either horizontal or vertical. **(c)** ℓ_∞-Voronoi regions are not convex.

collinear points p and q in $\mathbb{R}^2$, i.e. p and q share a coordinate in $\mathbb{R}^2$, $V_p \cap V_q$ is the union of a line segment and two cones, the shaded areas in Figure 2.6b. We introduce the concept of bisector, and then describe constraints that can be imposed on the points of S to avoid such degenerate cases.

Definition 2.4.6. Let S be a finite set of points in $(\mathbb{R}^d, d_\infty)$. The *bisector* of a subset $\sigma \subseteq S$ is

$$\mathrm{bis}_\sigma = \left\{ x \in \mathbb{R}^d \mid d_\bullet(p,x) = d_\bullet(q,x) \text{ for } p,q \in \sigma \right\}. \tag{2.22}$$

Remark. We have $\bigcap_{p\in\sigma} V_p \subset \mathrm{bis}_\sigma$ by definition of Voronoi region and bisector. So showing that bis_σ is non-degenerate implies that $\bigcap_{p\in\sigma} V_p$ is also non-degenerate.

In [CJS19] the structure of bisectors of polyhedral distances is studied in light of different types (weak and strong) of general position assumptions. In particular, by Proposition 3.1 of [CJS19], it follows that the bisector between any two points in any $(\mathbb{R}^d, d_\infty)$ is a polyhedral complex.

In this thesis, we use different definitions of general position for point sets in different dimensions. In particular, more conditions are imposed on points in dimension two, so that in this case S is in weak general position, as defined in [CJS19].

Definition 2.4.7. Let S be a finite set of points in $(\mathbb{R}^d, d_\infty)$. The set S is in *general position* if the distances between pairs of points of S are all distinct. Moreover, for $d = 2$, it is required that no four points in S lie on the boundary of a square, no three points are collinear, and no two points have the same x or y coordinate.

Remark. The general position of S ensures that the intersection of three ℓ_∞-Voronoi regions in $\mathbb{R}^2$ is either empty or a point, by Corollary 3.19 of [CJS19].

The following result describes the bisector of a pair of points in general position in the plane.

Proposition 2.4.8. *Let p, $q \in (\mathbb{R}^2, d_\infty)$ be in general position. The bisector of p and q is the union of the line segment $A_e^{\bar{r}} = \partial\overline{B_{\bar{r}}(p)} \cap \partial\overline{B_{\bar{r}}(q)}$, where $\bar{r} = \frac{d_\infty(p,q)}{2}$, and two half-infinite lines with slope either 1 or -1, the initial points of which are the endpoints of $A_e^{\bar{r}}$.*

Proof. Let $e = \{p, q\}$. The bisector bis_e of p and q is the set of equidistant points from p and q by definition, i.e. $\mathrm{bis}_e = \bigcup_{r>0} \partial\overline{B_r(p)} \cap \partial\overline{B_r(q)}$. Moreover, both $\partial\overline{B_r(p)}$ and $\partial\overline{B_r(q)}$ are the boundaries of axis-parallel squares in the plane with sides of length $2r$, by definition of d_∞. These have an empty intersection for any $r < \bar{r}$, where $\bar{r} = \frac{d_\infty(p,q)}{2}$. On the other hand, $A_e^{\bar{r}} = \partial\overline{B_{\bar{r}}(p)} \cap \partial\overline{B_{\bar{r}}(q)}$ is a horizontal or vertical line segment, because the axis-parallel squares intersect along a face for $r = \bar{r}$. For example, in Figure 2.6c $A_e^{\bar{r}}$ is the line segment $[0.0, 0.5] \times [0.5, 0.5] \subseteq \mathbb{R}^2$. In case $r = \bar{r} + \varepsilon > \bar{r}$, the intersection $\partial\overline{B_{\bar{r}+\varepsilon}(p)} \cap \partial\overline{B_{\bar{r}+\varepsilon}(q)}$ consists of exactly two points $a^\varepsilon = (a_x^\varepsilon, a_y^\varepsilon)$, $b^\varepsilon = (b_x^\varepsilon, b_y^\varepsilon) \in \mathbb{R}^2$ for each $\varepsilon > 0$, by the general position assumption on p and q. Given the endpoints $c = (c_x, c_y)$ and $d = (d_x, d_y) \in \mathbb{R}^2$ of $A_e^{\bar{r}}$, we have the below equations, which follow from the structure of the intersections of boundaries of the axis-parallel squares $\partial\overline{B_{\bar{r}+\varepsilon}(p)}$ and $\partial\overline{B_{\bar{r}+\varepsilon}(q)}$:

$$a_x^\varepsilon = c_x \pm \varepsilon, \tag{2.23}$$

$$a_y^\varepsilon = c_y \pm \varepsilon, \tag{2.24}$$

$$b_x^\varepsilon = d_x \pm \varepsilon, \tag{2.25}$$

$$b_y^\varepsilon = d_y \pm \varepsilon. \tag{2.26}$$

Note that ε is added or subtracted depending on the relative positioning of p and q. Because the choice of sign in these equations is fixed for any $\varepsilon > 0$, we have that $L_1 = \bigcup_{\varepsilon>0} a^\varepsilon$ and $L_2 = \bigcup_{\varepsilon>0} b^\varepsilon$ are two half-infinite lines with c and d as initial points respectively, and that L_1 and L_2 have slope either 1 or -1. We conclude that $\mathrm{bis}_e = \bigcup_{r>0} \partial\overline{B_r(p)} \cap \partial\overline{B_r(q)} = A_e^{\bar{r}} \cup L_1 \cup L_2$. □

Definition 2.4.9. The *Delaunay triangulation* of a finite set of points S in general position in $(\mathbb{R}^2, d_\infty)$ is the geometric realization of the Delaunay complex K^D of S, which is the set of convex hulls of simplices of K^D.

Finally, ℓ_∞-Voronoi regions are shown to be generally non-convex. To see this consider

$p = (0,0)$ and $q = \left(\frac{1}{2}, 1\right)$ in $\mathbb{R}^2$ and the intersection of their ℓ_∞-Voronoi regions, as in Figure 2.6c. These are such that $z_1 = \left(\frac{1}{4}, \frac{1}{2}\right), z_2 = \left(\frac{3}{4}, \frac{1}{4}\right) \in V_p, V_q$, but the middle point on the line segment from z_1 to z_2 is $\frac{z_1+z_2}{2} = \left(\frac{1}{2}, \frac{3}{8}\right)$ which belongs to V_p only. Thus V_q is not convex so that the standard way of proving the equivalence of Čech filtrations and the next type of filtration introduced in this section does not work in ℓ_∞ metric.

Alpha filtrations. The Voronoi regions of S can be used to filter-out high-dimensional simplices from Čech complexes.

Definition 2.4.10. Let S be a finite set of points in $(\mathbb{R}^d, d_\bullet)$. The *Alpha complex* with radius r of S is

$$K_r^A = \Big\{\sigma \subseteq S \mid \bigcap_{p\in\sigma} \left(\overline{B_r(p)} \cap V_p\right) \neq \emptyset\Big\}. \tag{2.27}$$

Given $\mathcal{R} = \{r_i\}_{i=0}^m$ to be a finite set of monotonically increasing real values, the *Alpha filtration* $K_\mathcal{R}^A$ of S is

$$K_{r_0}^A \subseteq K_{r_1}^A \subseteq \dots K_{r_m}^A. \tag{2.28}$$

The idea is to remove "redundant" simplices from $K_r^{\check{C}}$ and preserve its equivalence with the union of closed balls centered in the points of S. This might be possible because $\bigcup_{p\in S} \overline{B_r(p)} = \bigcup_{p\in S} \left(\overline{B_r(p)} \cap V_p\right)$ for any $r \in \mathbb{R}$. In practice, this works only for points in Euclidean distance. In this case, it is known that the filtration $K_\mathcal{R}^A$ produces the same persistence diagrams of $K_\mathcal{R}^{\check{C}}$. This is proven by means of the Nerve Theorem 2.4.3, which applies because K_r^A is the nerve of the collection $\left\{\overline{B_r(p)} \cap V_p\right\}_{p\in S}$, the elements of which are all convex and closed, assuming the Euclidean distance d_2 is used. Importantly, convex sets are contractible, as well as any intersection of a finite number of convex sets. Thus K_r^A and $K_r^{\check{C}}$ are homotopy equivalent for each $r \in \mathbb{R}$, and their homology groups isomorphic. The equivalence of $K_\mathcal{R}^A$ and $K_\mathcal{R}^{\check{C}}$ follows from the next theorem.

Theorem 2.4.11 (Persistence Equivalence Theorem [EH10])**.** *Consider two sequences of homology groups with coefficients in a field connected by homomorphisms $\phi_i : U_i \to V_i$*

$$\begin{tikzcd} U_0 \arrow[r] \arrow[d] & U_1 \arrow[r] \arrow[d] & \dots \arrow[r] & U_m \arrow[r] \arrow[d] & U_{m+1} \arrow[d] \\ V_0 \arrow[r] & V_1 \arrow[r] & \dots \arrow[r] & V_m \arrow[r] & V_{m+1}. \end{tikzcd} \tag{2.29}$$

If the ϕ_i are isomorphisms and all square commute, then the persistence diagrams defined by the U_i are the same as those defined by the V_i.

See [EH10, Section 3.4] for more details on Alpha complexes.

In conclusion, Alpha filtrations can be used to speed up the computation of Čech persistent homology of $S \subseteq (\mathbb{R}^d, d_2)$, because $K^A_{r_i} \subseteq K^D$ for each $r_i \in R$ and the Delaunay complex of S contains only a subset of the simplices of the full complex K^n. In particular, K^D is expected to contain $O(n^{\lceil \frac{d}{2} \rceil})$ d-dimensional simplices if S consists of n points in $\mathbb{R}^d$ [HB08].

Delaunay-Čech filtrations. Alpha complexes use Voronoi regions to constrain the intersection of the closed balls centered in the points of S. The simplices of the Delaunay complex K^D can also be used directly.

Definition 2.4.12. Let S be a finite set of points in $(\mathbb{R}^d, d_\bullet)$. The *Delaunay-Čech complex* with radius r of S is

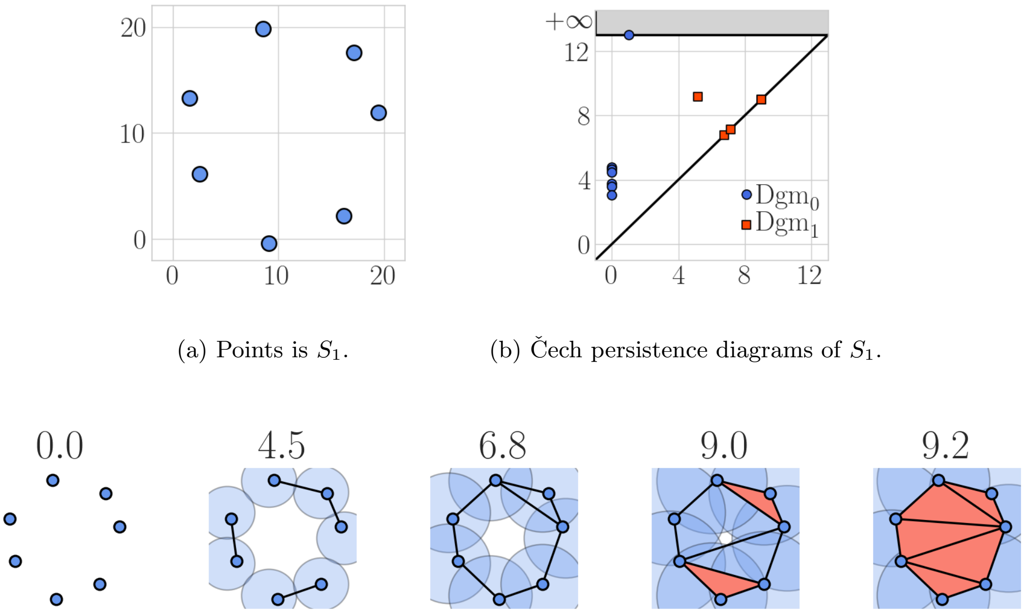
$$K^{D\check{C}}_r = \Big\{\sigma \subseteq K^D \mid \bigcap_{p\in\sigma} \overline{B_r(p)} \neq \emptyset\Big\}. \tag{2.30}$$

Given $\mathcal{R} = \{r_i\}_{i=0}^m$ to be a finite set of monotonically increasing real values, the *Delaunay-Čech filtration* $K^{D\check{C}}_{\mathcal{R}}$ of S is

$$K^{D\check{C}}_{r_0} \subseteq K^{D\check{C}}_{r_1} \subseteq \dots K^{D\check{C}}_{r_m}. \tag{2.31}$$

Simplices in $K^{D\check{C}}_r$ are parameterized by their minimal enclosing ball radius, as for Čech simplices. So Delaunay-Čech filtrations can be seen as sublevel sets filtrations of $h_{D\check{C}} : K^D \to \mathbb{R}$ with $h_{D\check{C}}(\sigma) = \inf_{x\in\mathbb{R}^d} \max_{p\in\sigma} d_\bullet(x,p)$ for each $\sigma \in K^D$. Given points in n-dimensional Euclidean space, in [BE17] it is proven that Alpha, Delaunay-Čech, Čech filtrations all produce the same persistence diagrams.

Example: Perturbed Delaunay-Čech filtration. The stability of persistence diagrams of Delaunay-Čech filtrations is illustrated with an example. Recall that the Stability Theorem 2.3.10 guarantees that the persistence diagrams of close filtering functions are going to be close. So, because of the definition of Delaunay-Čech filtering functions $h_{D\check{C}}$, infinitesimal perturbations of points sets induce infinitesimal perturbations of their persistence diagrams. Here, this property is empirically illustrated with an example. Let S_1 and S_2 consist of two distinct random perturbations of seven points disposed on a circle of radius 10 in $(\mathbb{R}^2, d_2)$, plotted in Figures 2.7a and 2.8a. Moreover, let K^D_1, K^D_2 be the Euclidean Delaunay complexes of S_1 and S_2, and $\mathcal{R}_1$, $\mathcal{R}_2$ the sets of monotonically increasing real values containing the minimal enclosing ball radiuses of all simplices in K^D_1 and K^D_2 respectively. Given these, the Euclidean Čech persistence diagrams of S_1 and S_2 can be computed using their Delaunay-Čech filtrations $K^{D\check{C}}_{\mathcal{R}_1}$ and $K^{D\check{C}}_{\mathcal{R}_2}$. Figures 2.7c and 2.8c show five of the subcomplexes in these filtrations.

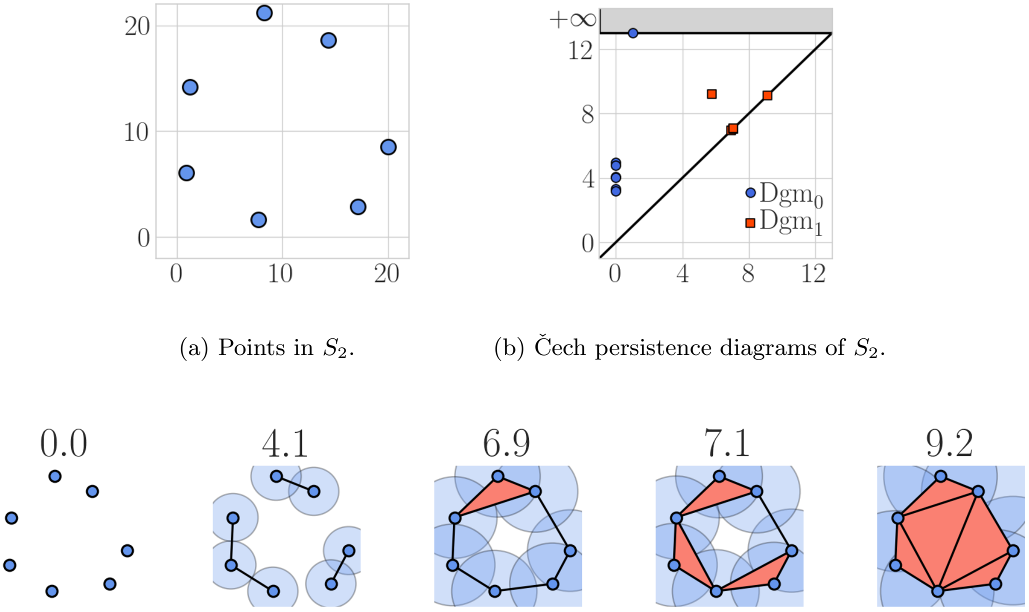


(a) Points is S_1.

(b) Čech persistence diagrams of S_1.

(c) Delaunay-Čech filtration of points in S_1.

Figure 2.7



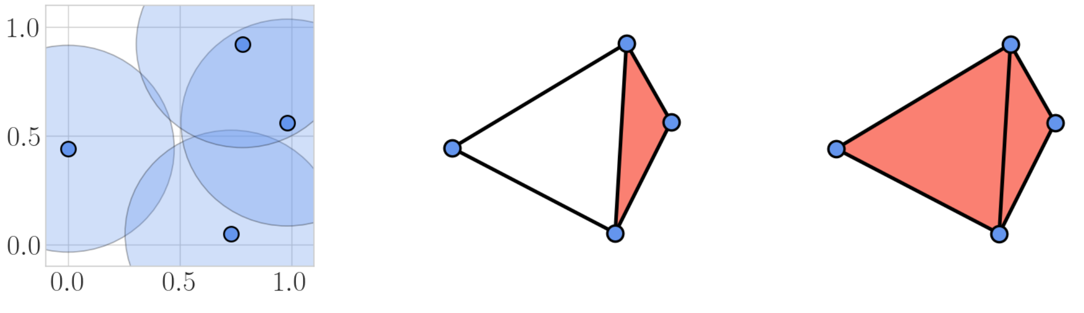
(a) Points in S_2.

(b) Čech persistence diagrams of S_2.

(c) Delaunay-Čech filtration of points in S_2.

Figure 2.8

The resulting Čech persistence diagrams are in Figures 2.7b and 2.8b. These were computed using the `gudhi` Python package [GUD21], which was also used to calculate



(a) Points in $\mathbb{R}^2$ (b) Čech complex (c) Vietoris-Rips complex

Figure 2.9: The intersections of disks in **(a)** produce the Čech and Vietoris-Rips complexes in **(b)** and **(c)** respectively.

the bottleneck distances between diagrams, obtaining $d_B(\mathrm{Dgm}_0(K^{D\check{C}}_{\mathcal{R}_1}), \mathrm{Dgm}_0(K^{D\check{C}}_{\mathcal{R}_2})) \approx 0.397$ and $d_B(\mathrm{Dgm}_1(K^{D\check{C}}_{\mathcal{R}_1}), \mathrm{Dgm}_1(K^{D\check{C}}_{\mathcal{R}_2})) \approx 0.626$.

Vietoris-Rips filtrations. An abstract simplicial complex K is a *flag complex* if K is the clique complex of its 1-skeleton, i.e. K contains a simplex σ if and only if it contains all the edges in σ. The final parameterized filtration introduced in this section consists of a sequence of flag complexes.

Definition 2.4.13. Let S be a finite set of points in $(\mathbb{R}^d, d_\bullet)$. The *Vietoris-Rips complex* with radius r of S is

$$K_r^{VR} = \left\{ \sigma \subseteq S \mid \max_{p,q \in \sigma} d_\bullet(p,q) < 2r \right\}. \tag{2.32}$$

Given $\mathcal{R} = \{r_i\}_{i=0}^m$ to be a finite set of monotonically increasing real values, the *Vietoris-Rips filtration* $K_{\mathcal{R}}^{VR}$ of S is

$$K_{r_0}^{VR} \subseteq K_{r_1}^{VR} \subseteq \ldots \subseteq K_{r_m}^{VR}. \tag{2.33}$$

Both Vietoris-Rips and Čech complexes are subcomplexes of the full complex K^n on S. The advantages of Vietoris-Rips complexes are that the parameters of their simplices are $\max_{p,q\in\sigma} d_\bullet(p,q)$ for each $\sigma \in K_r^{VR}$ (there is no need to compute smallest enclosing ball radiuses), and that their flag structure allows for shortcuts in the computation of their persistence diagrams (apparent and emergent persistence pairs of [Bau19]). Moreover, efficient software has been developed for the computation of Vietoris-Rips persistence diagrams [Bau19, TSBO18]. Hence, in case it is not possible to compute the Delaunay complex of S, for instance in high-dimensions where the complexity of K^D explodes, Vietoris-Rips persistence diagrams are usually preferred to Čech persistence diagrams as a way of encoding the structure of a set of points on a range of scales. The following proposition ensures that the sublevel sets filtrations from which Euclidean Čech and Vietoris-Rips complexes are obtained cannot be too dissimilar. By the Stability Theorem 2.3.10 this also gives a (multiplicative) bound on the bottleneck distance between their diagrams.

Proposition 2.4.14. *Let $K_r^{\check{C}}$ and K_r^{VR} be the Euclidean Čech and Vietoris-Rips complexes of $S \subseteq (\mathbb{R}^d, d_2)$ with radius $r \in \mathbb{R}$. The following nesting holds*

$$K_r^{\check{C}} \subseteq K_r^{VR} \subseteq K^{\check{C}}_{2r\sqrt{\frac{d}{2(d+1)}}} \subseteq K^{\check{C}}_{\sqrt{2}r}. \tag{2.34}$$

Proof. The first inclusion follows by the definition of Čech and Vietoris-Rips complexes. The second by Jung's theorem [DGK63] on a set of points in Euclidean space. Finally, note that $2r\sqrt{\frac{d}{2(d+1)}}$ converges to $2r\frac{1}{\sqrt{2}} = \sqrt{2}r$ from below for dimension d going to infinity, hence the third inclusion. □

Chapter 3

Euler Characteristic of Multiparameter Filtrations

Euler characteristic curves and persistence diagrams, defined in Sections 2.2 and 2.3, encode information about the structural changes of abstract complexes in a filtration. This is a parameterized sequence of nested complexes, obtained by taking sublevel sets of a real-values function $h : K \to \mathbb{R}$. In the case of images, an appropriate h can be defined using pixel intensity values. On the other hand, for finite point sets the radius of the smallest enclosing balls of geometric simplices can be used to obtain the Čech filtering function $h_{\check{C}}$. In both cases, a single parameter determines the final filtration.

In [CZ09] it was observed that many applications would benefit from studying the structural changes in families of complexes determined by multiple parameters. Values of radii, densities, and curvatures are mentioned as examples of parameters that could be combined to study the geometric structure and topological connectivity of datasets. Moreover, in the same paper, it was introduced the theory of multiparameter persistence and the rank invariant. Unfortunately, the computation of this invariant does not scale as well as for persistence diagrams (although efficient implementations exist for two-parameter persistence [LW15]), which restricts its possible applications.

Here it is proposed to utilize the Euler characteristic of families of complexes determined by two-parameters as an alternative to bi-dimensional persistence. This still allows getting insights about data on a multidimensional parameter space, while reducing the computational burden for obtaining them. The price to pay is a reduced amount of topological information extracted from the given nested family of complexes, as Euler characteristic is fully determined by ranks of homology groups, see Equation (2.9).

In this chapter, Euler characteristic curves are generalized to the Cartesian product of two parameterized filtrations. Then, algorithms are presented for the computation of such objects, both for image and point data. A Python package implementing these algorithms is also provided, which allows for their application in practice. To conclude, computational experiments are given to illustrate how multidimensional parameterizations capture information that would otherwise be lost by single-parameter Euler characteristic curves.

Note that, the novel results presented in this chapter are also part of the preprint [BAG$^+$21], where Euler characteristic is applied for the detection of diabetic retinopathy in retinal image.

3.1 Bi-filtrations and Euler Characteristic

The concept of sublevel sets filtration (see Definition 2.2.4) can be modified to make use of a pair of parameters for each simplex in an abstract complex K. The notation of Chapter 2 is adopted. Thus, a filtration of an abstract complex K is denoted by $K_{\mathcal{R}}$, where $\mathcal{R} = \{r_i\}_{i=0}^{m}$ is a monotonically increasing set of real values.

Definition 3.1.1. A *bi-filtration* of an abstract simplicial complex K parameterized by $\mathcal{R}_1$ and $\mathcal{R}_2$ is a grid of nested subcomplexes

$$K_{\mathcal{R}_1,\mathcal{R}_2} = \left\{ \begin{array}{ccccccc} K_{0,0} & \subseteq & K_{0,1} & \subseteq & \cdots & \subseteq & K_{0,m_2} \\ \rotatebox{90}{$\supseteq$} & & \rotatebox{90}{$\supseteq$} & & & & \rotatebox{90}{$\supseteq$} \\ K_{1,0} & \subseteq & K_{1,1} & \subseteq & \cdots & \subseteq & K_{1,m_2} \\ \rotatebox{90}{$\supseteq$} & & \rotatebox{90}{$\supseteq$} & & & & \rotatebox{90}{$\supseteq$} \\ \vdots & & \vdots & & \ddots & & \vdots \\ \rotatebox{90}{$\supseteq$} & & \rotatebox{90}{$\supseteq$} & & & & \rotatebox{90}{$\supseteq$} \\ K_{m_1,0} & \subseteq & K_{m_1,1} & \subseteq & \cdots & \subseteq & K_{m_1,m_2} \end{array} \right\}, \qquad (3.1)$$

where $\mathcal{R}_1 = \{r_i^1\}_{i=0}^{m_1}$ and $\mathcal{R}_2 = \{r_j^2\}_{j=0}^{m_2}$ are mononically increasing sets of real values, and $K_{i,j} \subseteq K$ for each $0 \leq i \leq m_1$ and $0 \leq j \leq m_2$.

Remark. The subcomplexes in Equation (3.1) are denoted with $K_{i,j}$ instead of $K_{r_i^1,r_j^2}$ to simplify notation.

The following definition is given as in [BAG$^+$21].

Definition 3.1.2. Let $K_{\mathcal{R}_1,\mathcal{R}_2}$ be a bi-filtration of an abstract complex K on the sets of monotonically increasing real values $\mathcal{R}_1 = \{r_i^1\}_{i=0}^{m_1}$ and $\mathcal{R}_2 = \{r_j^2\}_{j=0}^{m_2}$. The *Euler* *characteristic surface* of $K_{\mathcal{R}_1,\mathcal{R}_2}$ is the $(m_1+1)\times(m_2+1)$ matrix of integers

$$\mathrm{S}(K_{\mathcal{R}_1,\mathcal{R}_2}) = \begin{pmatrix} \chi(K_{0,0}), & \chi(K_{0,1}), & \cdots & \chi(K_{0,m_2}) \\ \chi(K_{1,0}), & \chi(K_{1,1}), & \cdots & \chi(K_{1,m_2}) \\ \vdots & \vdots & \ddots & \vdots \\ \chi(K_{m_1,0}), & \chi(K_{m_1,1}), & \cdots & \chi(K_{m_1,m_2}) \end{pmatrix}, \tag{3.2}$$

where $\chi(K_{i,j})$ is the Euler characteristic of the subcomplexes in the bi-filtration of K.

As done in Chapter 2 with filtering functions, we use sublevel sets of appropriate functions to define bi-filtrations on data.

Definition 3.1.3. Let K be an abstract complex. A function $\mathbf{h} : K \to \mathbb{R}^2$ is a *bi-filtering function* on K if for each $\sigma, \tau \in K$ such that $\sigma \subseteq \tau$, then $\mathbf{h}(\sigma)_1 \leq \mathbf{h}(\tau)_1$ and $\mathbf{h}(\sigma)_2 \leq \mathbf{h}(\tau)_2$, where $\mathbf{h}(\sigma) = (\mathbf{h}(\sigma)_1, \mathbf{h}(\sigma)_2)$, $\mathbf{h}(\tau) = (\mathbf{h}(\tau)_1, \mathbf{h}(\tau)_2)$.

Definition 3.1.4. Let K be an abstract complex and $\mathbf{h} : K \to \mathbb{R}^2$ a bi-filtering function on K. The *sublevel sets bi-filtration* of K induced by $\mathbf{h}$ on two sets of monotonically increasing real-values $\mathcal{R}_1 = \{r_i^1\}_{i=0}^{m_1}$, $\mathcal{R}_2 = \{r_j^2\}_{j=0}^{m_2}$ is the bi-filtration $K^{\mathbf{h}}_{\mathcal{R}_1,\mathcal{R}_2}$ such that

$$K_{i,j} = \mathbf{h}^{-1}\big((-\infty, r_i^1] \times (-\infty, r_j^2]\big), \tag{3.3}$$

for each $0 \leq i \leq m_1$ and $0 \leq j \leq m_2$.

Remark. If $\mathbf{h}$ is defined by means of two filtering functions $h_1 : K \to \mathbb{R}$ and $h_2 : K \to \mathbb{R}$, i.e. $\mathbf{h}(\sigma) = (h_1(\sigma), h_2(\sigma))$ for each $\sigma \in K$, then Equation (3.3) is equivalent at $K_{i,j} = h_1^{-1}\big((-\infty, r_i^1]\big) \cap h_2^{-1}\big((-\infty, r_j^2]\big)$ by the definition of Cartesian product.

Euler characteristic surfaces of pairs of images. Given a pair of gray-scale images M_1 and M_2, with the same size $n_1 \times n_2$ and values in $[0, m] \subseteq \mathbb{N}$, the method described in Section 2.2 can be used to obtain the pixel intensity filtering functions h_{M_1} and h_{M_2} of M_1 and M_2 respectively. Defined $\mathbf{h} : K_{M_1} \to \mathbb{R}$ by setting $\mathbf{h}(\sigma) = (h_{M_1}(\sigma), h_{M_2}(\sigma))$ for each $\sigma \in K_{M_1}$[1], it follows that $\mathbf{h}$ is a bi-filtering function. The *Euler characteristic surface* S_{M_1,M_2} of the pair of images M_1 and M_2 is defined as $\mathrm{S}(K^{\mathbf{h}}_{\mathcal{R}_1,\mathcal{R}_2})$, where both $\mathcal{R}_1$ and $\mathcal{R}_2$ coincide with the set of integer values in $[0, m]$. Note that the last column and last row of $\mathrm{S}(K^{\mathbf{h}}_{\mathcal{R}_1,\mathcal{R}_2})$ are equal to the Euler characteristic curves $\mathrm{C}(K^{h_{M_1}}_{\mathcal{R}_1})$ and $\mathrm{C}(K^{h_{M_2}}_{\mathcal{R}_2})$ respectively, by the remark above. So the Euler characteristic surface contains all the topological information of $\mathrm{C}(K^{h_1}_{\mathcal{R}_1})$ and $\mathrm{C}(K^{h_2}_{\mathcal{R}_2})$, plus the information coming from intersections of sublevel sets of h_{M_1} and h_{M_2}. In case a pair of three-dimensional

[1] The abstract cubical complexes K_{M_1} and K_{M_2} coincide because M_1 and M_2 are both n_1-by-n_2 matrices.

images M_1, M_2 is given, it is assumed that K_{M_1} and K_{M_2} are three-dimensional abstract cubical complexes whose geometric realizations contain an elementary cube $[s, s+1] \times [t, t+1] \times [u, u+1]$ for each voxel $v_{s,t,u} \in [0, m]$ in M_1 and M_2. Moreover, the voxel intensity filtering functions h_{M_1} and h_{M_2} are the natural extensions of pixel intensity filtering functions, setting the value of top-dimensional elements in K_{M_1} and K_{M_2} to the corresponding voxel intensities.

In Sections 3.4 and 3.5, we describe novel algorithms for computing Euler characteristic surfaces of two and three-dimensional data. Before discussing these, we study the invariance properties of Euler characteristic curves and surfaces, as well as their stability with respect to perturbations in the input data. Furthermore, we investigate the structure of expected Euler characteristic surfaces of random images in order to show that they can contain more information than Euler characteristic curves.

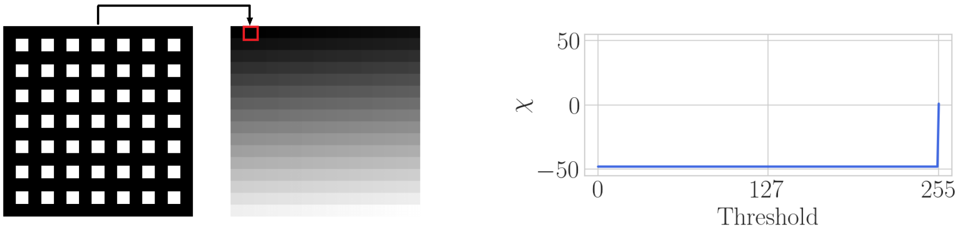
3.2 Properties of Euler Characteristic Curves and Surfaces

By Equation 2.9 in Chapter 2, we know that the Euler characteristic $\chi(K)$ of an abstract complex K is determined by the ranks of the homology groups of K. So, $\chi(K)$ is determined by the homotopy type of K, as homotopy equivalent complexes have isomorphic homology groups [Hat02]. Thus, we conclude that Euler characteristic curves and surfaces are invariant up to homotopy equivalence of the subcomplexes of filtrations $K_{\mathcal{R}}$ and bi-filtrations $K_{\mathcal{R}_1,\mathcal{R}_2}$.

When dealing with real-world applications, Euler characteristic curves and surfaces are derived from sublevel sets (bi-)filtrations. Hence, we are interested in the way in which Euler characteristic changes, given a perturbation of the input data. In this context, it would be desirable to prove a result equivalent to the Stability Theorem 2.3.10 of persistent homology. However, we show with a counterexample that such a result cannot be obtained.

Counterexample: "close" gray-scale images with different Euler characteristic curves. We show the existence of gray-scale images M_1 and M_2, of arbitrary size, with pixel values in $[0, 255]$ such that $||h_{M_1} - h_{M_2}||_\infty = 1$, where h_{M_1} and h_{M_2} are the pixel intensity filtering functions of M_1 and M_2. Moreover, we show that, by increasing the size of M_1 and M_2, the difference between their Euler characteristic curves goes to infinity, i.e. $||\mathrm{C}_{M_1} - \mathrm{C}_{M_2}||_\infty \to +\infty$.

We start by defining M_1 and M_2. Given two odd integers n_1 and n_2, we set their sizes to $(16 \cdot n_1) \times (16 \cdot n_2)$. The idea is to make M_1 and M_2 into the union of 256 rectangular matrices, each of size $n_1 \times n_2$. To simplify the exposition, we define Z to be



(a) (b)

Figure 3.1: A gray-scale image, in **(a)**, the Euler characteristic curve of which, in **(b)**, equals a negative constant on $[0, 254]$, which decreases by increasing the size of the image.

a zero matrix of size $n_1 \times n_2$, and H to be a 'holes' matrix of size $n_1 \times n_2$ such that

$$H[i][j] = \begin{cases} 1 \text{ if } i \text{ and } j \text{ are odd,} \\ 0 \text{ otherwise.} \end{cases} \tag{3.4}$$

It should be noted that, by thresholding the matrix H at level 0, we obtain a binary image like the one on the left in Figure 3.1a. This corresponds to a two-dimensional abstract cubical complex with one connected component and $\frac{n_1-1}{2} \cdot \frac{n_2-1}{2}$ one-dimensional holes. Moreover, we define $Z^{(k)}$ and $H^{(k)}$ as the $n_1 \times n_2$ matrices such that $Z^{(k)}[i][j] = Z[i][j]+k$ and $H^{(k)}[i][j] = H[i][j] + k$ for each $0 \le i \le n_1 - 1$ and $0 \le j \le n_2 - 1$.

Given these matrices, we first define

$$M_1[i \cdot n_1 : (i+1) \cdot n_1 - 1][j \cdot n_2 : (j+1) \cdot n_2 - 1] = H^{(j+16i-1)}, \tag{3.5}$$

$$M_2[i \cdot n_1 : (i+1) \cdot n_1 - 1][j \cdot n_2 : (j+1) \cdot n_2 - 1] = Z^{(j+16i-1)}, \tag{3.6}$$

for each $0 \le i \le 15$ and $0 \le j \le 15$, where $[i \cdot n_1 : (i+1) \cdot n_1 - 1]$ and $[j \cdot n_2 : (j+1) \cdot n_2 - 1]$ stand for all the indices from $i \cdot n_1$ to $(i+1) \cdot n_1 - 1$ and from $j \cdot n_2$ to $(j+1) \cdot n_2 - 1$. Finally, we set $M_1[0 : n_1 - 1][0 : n_2 - 1] = Z$ and $M_2[0 : n_1 - 1][0 : n_2 - 1] = Z$, so that all the elements of M_1 and M_2 are values in $[0, 255]$. Given $n_1 = n_2 = 15$, Figure 3.1a shows M_1 and the result of thresholding at level k one of its $H^{(k)}$ submatrices. Its corresponding Euler characteristic curves is given in Figure 3.1b.

We have that $||h_{M_1} - h_{M_2}||_\infty = 1$ because this distance equals the maximum absolute value difference between any two pixels at the same position in $H^{(k)}$ and $Z^{(k)}$. On the other hand, the value of $||\mathrm{C}_{M_1} - \mathrm{C}_{M_2}||_\infty$ increases with the size of M_1 and M_2. This follows, because by definition of M_1 and M_2:

- The subcomplexes $h_{M_1}^{-1}\big((-\infty, v]\big)$ contain one connected component and $\frac{n_1-1}{2} \cdot \frac{n_2-1}{2}$

one-dimensional holes for each $v \in [0, 254]$, and only one connected component for $v = 255$.

- The subcomplexes $h^{-1}_{M_2}\big((-\infty, v]\big)$ contain one connected component and no one-dimensional holes for each $v \in [0, 255]$.

Thus, C_{M_1} is equal to $1 - \frac{n_1-1}{2} \cdot \frac{n_2-1}{2}$ on the range $[0, 254]$, while C_{M_2} is equal to 1 on the same range. Importantly, the value of the negative constant $1 - \frac{n_1-1}{2} \cdot \frac{n_2-1}{2}$ depends on the size of M_1 and M_2, so that, by increasing n_1 and n_2, the value of $||\mathrm{C}_{M_1} - \mathrm{C}_{M_2}||_\infty$ can be made arbitrarily big.

Remark. Given the zero matrix M_3 of size $(16 \cdot n_1) \times (16 \cdot n_2)$, the Euler characteristic surfaces $S_{M_1 M_3}$ and $S_{M_2 M_3}$ can be used to extend the counterexample described above to bi-filtrations.

We conclude that, given a fixed difference in the (bi-)filtrations producing Euler characteristic curves and surfaces of images, these can be arbitrarily different. Therefore, it is not possible to prove a general Stability Theorem in this setting.

3.3 Euler Characteristic Surfaces of Random Images

Here we show with an example that Euler characteristic surfaces of pairs of images can contain useful information for distinguishing between different classes in a dataset, while the Euler characteristic curves of the same images do not. We introduce a method that can be used to obtain a family of pairs of random gray-scale images. While all such images have the same expected Euler characteristic curve, we provide an analytical expression of the expected values of the entries $\chi(K_{i,j})$ in Equation (3.2), which are not constant for different pairs in the family.

Fixed the sizes $n_1, n_2 \in \mathbb{N}$ and a probability $p \in [0, 1] \subseteq \mathbb{R}$, a pair of random gray-scale images M_1^p, $M_2^p \in \mathbb{N}^{n_1 \times n_2}$ can be generated with the following method. To define each pair of pixels of M_1^p and M_2^p at position (s, t), denoted by $M_1^p[s][t]$ and $M_2^p[s][t]$, three random values $x, v_1, v_2 \in \mathbb{R}$ are sampled from independent uniform distributions $\mathcal{U}(0, 1)$, $\mathcal{U}(0, 256)$, $\mathcal{U}(0, 256)$. The value of $0 < x < 1$ is used to to decide if the (s, t) pixels in M_1^p and M_2^p are set equal or not. In practice, if $x \leq p$, then $M_1^p[s][t] = M_2^p[s][t] = \lfloor v_1 \rfloor$. Otherwise, $M_1^p[s][t] = \lfloor v_1 \rfloor$ and $M_2^p[s][t] = \lfloor v_2 \rfloor$. Thus, for each position (s, t) pixels are set to the same random integer with probability p, and to independent random integers with probability $(1 - p)$. Hence, by sampling multiple values of x, v_1, and p for each position (s, t), it is possible to obtain a set of pairs of random gray-scale images with value in $[0, 255]$. Moreover, M_1^p and M_2^p have the same expected Euler characteristic curves for any $p \in [0, 1]$, because pixel values are sampled from the same uniform distributions $\mathcal{U}(0, 256)$.

On the other hand, defined $\mathbf{h} = (h_{M_1^p}, h_{M_2^p})$, it is possible to show that the expected Euler characteristic surface of the pairs M_1^p, M_2^p are different for different choices of the probability parameter p. In particular, we derive an analytical expression for the elements of the expected Euler characteristic surface $\mathrm{S}_{M_1^p, M_2^p}$, which differs for each $0 \leq p \leq 1$ [BAG+21].

Proposition 3.3.1. *Let M_1^p, M_2^p be two random gray-scale images of size $n_1 \times n_2$, generated with the method described above, where p is a real value in $[0, 1]$. Given the sublevel sets bi-filtrations $K^{\mathbf{h}}_{\mathcal{R}_1, \mathcal{R}_2}$ of M_1^p and M_2^p, the expected values of the elements of the Euler characteristic surface $S_{M_1^p, M_2^p}$ are*

$$\begin{aligned} E[\chi(K_{i,j})] = &(n_1 - 1)(n_2 - 1) \cdot \left[1 - (1 - P(\sigma_{s,t} \in K_{i,j})^4)\right] \\ &+ (n_1(n_2 + 1) + n_2(n_1 + 1) - 4) \cdot \left[1 - (1 - P(\sigma_{s,t} \in K_{i,j})^2)\right] \\ &+ (n_1 n_2 + 2n_1 + 2n_2 + 4) \cdot P(\sigma_{s,t} \in K_{i,j}), \end{aligned}$$

where $P(\sigma_{s,t} \in K_{i,j}) = \min\{i, j\} \cdot p + i \cdot j \cdot (1 - p)$

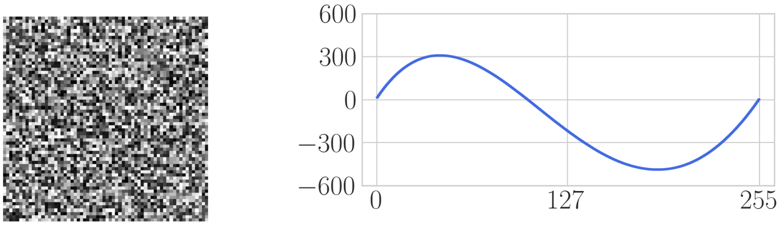
Proof. First, it is observed that the expected value of $\chi(K_{i,j})$, i.e. an element of the matrix $\mathrm{S}_{M_1^p, M_2^p}$, is completely determined by the expected number of vertices, edges, and squares in $K_{i,j} \subseteq K_{M_1^p} = K_{M_2^p}$. Moreover, it is known that a vertex is in $K_{i,j}$ if and only if at least one of the squares that include it is in $K_{i,j}$, and the same holds for edges. So given the expected probability $P(\sigma_{s,t} \in K_{i,j})$ of a square belonging to $K_{i,j}$, the expected probabilities of having vertices and edges in $K_{i,j}$ can be derived as well. From the definition of M_1^p and M_2^p and $\mathbf{h}$, it follows

$$\begin{aligned} P(\sigma_{s,t} \in K_{i,j}) = \; & P\Big(h_{M_1^p}(\sigma_{s,t}) < i \text{ and } h_{M_2^p}(\sigma_{s,t}) < j\Big) \cdot p \\ & + P\Big(h_{M_1^p}(\sigma_{s,t}) < i\Big) \cdot P\Big(h_{M_2^p}(\sigma_{s,t}) < j\Big) \cdot (1 - p) \\ = \; & \min\{i, j\} \cdot p + i \cdot j \cdot (1 - p), \end{aligned} \tag{3.7}$$

because the values of $h_{M_1^p}(\sigma_{s,t})$ and $h_{M_2^p}(\sigma_{s,t})$ are uniformly distributed in $[0, 255]$ and $0 \leq i, j \leq 255$. Then, because the values of different pixels are independent of each other, the probability that a vertex or edges σ' belongs to $K_{i,j}$ is

$$1 - \Big(1 - P(\sigma_{s,t} \in K_{i,j})^k\Big), \tag{3.8}$$

where k is the number of squares $\sigma_{s,t}$ containing σ'. Besides, it is known that in the



(a)

(b)

Figure 3.2: **(a)** A 64-by-64 random gray scale image.**(b)** Plot of the expected Euler characteristic curve of either M_1^p or M_2^p for any $0 \leq p \leq 1$.

$n_1 \times n_2$ abstract cubical complex $K_{255,255}$ there are:

- $(n_1 - 1)(n_2 - 1)$ internal vertices contained in four squares each;
- $2(n_1 - 1) + 2(n_2 - 1)$ boundary vertices contained in two squares each;
- 4 corner vertices contained in one square only;
- $n_1(n_2 + 1) + n_2(n_1 + 1) - 2n_1 - 2n_2$ internal edges contained in two squares each;
- $2n_1 + 2n_2$ boundary edges contained in one square only;
- and $n_1 n_2$ squares.

Finally, combining the expression in Equation (3.8) with the number of elements in $K_{255,255}$ above, the expected value of $\chi(K_{i,j})$ is

$$\begin{aligned} E[\chi(K_{i,j})] = {} & (n_1 - 1)(n_2 - 1) \cdot \left[1 - (1 - P(\sigma_{s,t} \in K_{i,j})^4)\right] \\ & + (n_1(n_2 + 1) + n_2(n_1 + 1) - 4) \cdot \left[1 - (1 - P(\sigma_{s,t} \in K_{i,j})^2)\right] \\ & + (n_1 n_2 + 2n_1 + 2n_2 + 4) \cdot P(\sigma_{s,t} \in K_{i,j}), \end{aligned} \tag{3.9}$$

where $P(\sigma_{s,t} \in K_{i,j}) = \min\{i, j\} \cdot p + i \cdot j \cdot (1 - p)$. □

Fixed $n_1 = 64$ and $n_2 = 64$, the expected Euler characteristic surfaces for $p = 0.1$ and $p = 0.8$, determined by Equation (3.3.1), are represented as contour plots in Figures 3.3a and 3.3b. Note that in this setting expected Euler characteristic curves are non-informative for distinguishing between random images generated using any $0 \leq p \leq 1$, as these always coincide with the curve in Figure 3.2b. On the other hand, expected Euler characteristic surfaces are different for each $0 \leq p \leq 1$. To further illustrate this, in Figure 3.3c it is given the contour plot of the absolute value of the difference of the expected surfaces for $p = 0.1$ and $p = 0.8$.

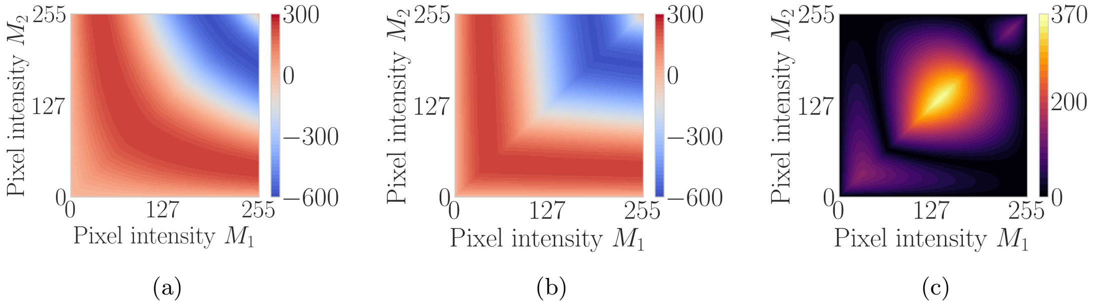


Figure 3.3: In **(a)** and **(b)** the contour plots of expected Euler characteristic surfaces of pairs of random images M_1^p, M_2^p with $p = 0.1$ and $p = 0.8$ respectively. In **(c)** the contour plot of the absolute value of the difference of the Euler characteristic surfaces in (a) and (b).

3.4 Algorithm for Image Data

In this and the next section, we describe novel algorithms that can be used to compute Euler characteristic surfaces. These are part of the results presented in the preprint [BAG⁺21]. Here it is discussed the case of image data, while in the following section the one of points data. In particular, it is given an algorithm to compute the Euler characteristic surface S_{M_1,M_2} of a sublevel sets bi-filtration of $\mathbf{h} : K_{M_1} \to \mathbb{R}^2$, which is defined by setting $\mathbf{h}(\sigma) = (h_{M_1}(\sigma), h_{M_2}(\sigma))$ for each $\sigma \in K_{M_1}$, where h_{M_1} and h_{M_2} are the pixel intensity filtering functions of two gray-scale images M_1, M_2 (see Definition 2.2.6). Algorithm 3.1 takes as inputs a pair of two or three-dimensional gray-scale images M_1, M_2 and a vector of precomputed Euler characteristic changes, and returns the matrix of Euler characteristic values S_{M_1,M_2}.[2] The correctness and running time of this algorithm are discussed below, while an implementation is provided by the `euchar` Python package, which is applied to real-world data in the final section of this chapter.

Discussion. It follows from the definition of $\mathbf{h}$ and Cartesian product that $K_{i,j} = \mathbf{h}^{-1}\big((-\infty, r_i^1] \times (-\infty, r_j^2]\big)$ is equivalent to $K_{i,j} = h_{M_1}^{-1}\big((-\infty, r_i^1]\big) \cap h_{M_2}^{-1}\big((-\infty, r_j^2]\big)$. So the j-th column of S_{M_1,M_2} equals the Euler characteristic curve of h_{M_1} with K_{M_1} restricted to its top-dimensional cubes $\bar{\sigma}$ such that $h_{M_2}(\bar{\sigma}) \leq j$, because of the intersection with the cubical complex $h_{M_2}^{-1}\big((-\infty, r_j^2]\big)$. Thus, a possible approach for computing the Euler characteristic surface of the sublevel sets bi-filtration of $\mathbf{h}$ is to apply Algorithm 2.1 for Euler characteristic curves to the restriction of h_{M_1} to $h_{M_2}^{-1}\big((-\infty, r_j^2]\big)$ for each $0 \leq j \leq m_2$, i.e. obtaining each column of S_{M_1,M_2} separately. We refer to this as the naïve approach, the correctness of which follows from the one of the Euler characteristic curve algorithm. For two-dimensional (three-dimensional) images, it has a running time of $O(nm_2 + m_1m_2)$, where n is the number of pixels (voxels) in M_1 and M_2.

[2] This is restricted to two and three dimensions because of practical limitations due to the size of the input vector of Euler characteristic changes for higher-dimensions.

To further improve the efficiency of real-world implementations, Algorithm 3.1 makes use of the following two strategies:

(i) Precompute the possible Euler characteristic changes produced by adding a top-dimensional $\bar{\sigma}$ into any $K_{i,j}$, and use these to increase or decrease the values of S_{M_1,M_2};

(ii) Loop on each top-dimensional $\bar{\sigma}$ only once, by modifying all columns of S_{M_1,M_2} where $\bar{\sigma}$ produces the same change at the same time.

In the following discussion, points *(i)* and *(ii)* above are shown to preserve the correctness of the naïve approach computing columns of S_{M_1,M_2} independently.

Using Euler characteristic changes as suggested in *(i)* is possible because the process of going from the empty abstract cubical complex to $K_{M_1} = K_{255,255}$ can be decomposed into steps at which a single $\bar{\sigma}$ and its subfaces are added. This follows from the definition of the filtering functions h_{M_1} and h_{M_2} in terms of pixel (voxel) intensity values. Furthermore, at each such step, the change $\Delta\chi^{\bar{\sigma}}$ in Euler characteristic of the current cubical complex is completely determined by the structure of elements adjacent to $\bar{\sigma}$. More precisely, defined the *neighbourhood* $N^{\bar{\sigma}}$ of $\bar{\sigma}$ to be the set of elementary cubes that intersect it, by Definition 2.2.1 $\Delta\chi^{\bar{\sigma}}$ only depends on the numbers of elementary cubes added into $N^{\bar{\sigma}}$ when $\bar{\sigma}$ is added. So all possible Euler characteristic changes can be precomputed because there is a finite number of neighbourhoods $N^{\bar{\sigma}}$.[3] In particular, there are $2^{(3^d-1)}$ such neighbourhoods in dimension d, meaning that there are 256 Euler characteristic changes to precompute for two-dimensional images and $67,108,864$ changes for three-dimensional images. For $d = 4$, the number of possible neighbourhoods is already a 25 digits integer, making the computation and storage of their corresponding changes impractical. Hence Equation (2.2) can be used to compute all the Euler characteristic changes for $d = 2$ and $d = 3$, which can then be stored in a vector *preCompChanges* using the binary representation of neighbourhoods to index them. For example, consider the neighbourhood in Figure 3.4a corresponding to the binary matrix

$$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}, \tag{3.10}$$

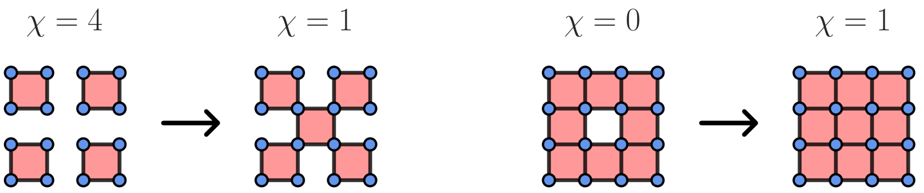
and in turn to the binary sequence 10100101. Its Euler characteristic change is -3 and the decimal representation of its binary sequence 165. Thus -3 is stored as the 165-th element of *preCompChanges*.

[3] For two-dimensional images, $N^{\bar{\sigma}}$ is a set of 8 squares and their subfaces, while for three-dimensional images it is a set of 26 cubes and their subfaces.

Algorithm 3.1 Euler characteristic surface of bi-filtration on a pair of images.

Input: gray-scale images M_1, M_2, $\mathbf{h} : K \to [0, m_1] \times [0, m_2] \subseteq \mathbb{Z}^2$, and the pre-computed vector *preCompChanges*.

```
 1: Add a one pixel (voxel) thick outer layer to images, so that the new boundary pixels
    (voxels) are mapped by $\mathbf{h}$ into $(m_1 + 1, m_2 + 1)$
 2: $\mathrm{S}_{M_1,M_2} \leftarrow$ $(m_1 + 1) \times (m_2 + 1)$ zeros matrix
 3: for each top-dimensional cube $\bar{\sigma}$ in $K_{M_1}$ do
 4:    $r^1_i, r^2_j \leftarrow h_{M_1}(\bar{\sigma}), h_{M_2}(\bar{\sigma})$
 5:    $neigh_1, neigh_2 \leftarrow h_{M_1}, h_{M_2}$ values in neighbourhood of $\bar{\sigma}$
 6:    $thresholds_2 \leftarrow$ sorted values in $neigh_2$ greater than $r^2_j$, union $m_2 + 1$
 7:    $N^{\bar{\sigma}}_1 \leftarrow$ boolean matrix defined by $(neigh_1 \leq r^1_i)$ before $\bar{\sigma}$ and $(neigh_1 < r^1_i)$ after
       $\bar{\sigma}$
 8:    for $k = 1$ to $|thresholds_2|$ do
 9:       $N^{\bar{\sigma}}_2 \leftarrow$ boolean matrix defined by $(neigh_2 \leq thresholds_2[k - 1])$
10:       $N^{\bar{\sigma}} \leftarrow$ element-wise AND of $N^{\bar{\sigma}}_1$ and $N^{\bar{\sigma}}_2$
11:       $l \leftarrow$ decimal integer of binary representation of $N^{\bar{\sigma}}$
12:       for $\hat{j}$ = index of $thresholds_2[k - 1]$ to index of $thresholds_2[k] - 1$ do
13:          $\mathrm{S}_{M_1,M_2}[i][\hat{j}]$ += $preCompChanges[l]$
14:       end for
15:    end for
16: end for
17: $\mathrm{S}_{M_1,M_2} \leftarrow$ cumulative sum on columns of $\mathrm{S}_{M_1,M_2}$
18: return $\mathrm{S}_{M_1,M_2}$
```



(a)

(b)

Figure 3.4: Euler characteristic changes produced by adding an elementary cube of maximal dimension in a two-dimensional cubical complex. In **(a)** the change is equal to −3, while in **(b)** it is +1.

Point *(ii)* above is realized by the inner loop on lines $8 - 15$ of Algorithm 3.1, where $r^1_i = h_{M_1}(\bar{\sigma})$ and $r^2_j = h_{M_2}(\bar{\sigma})$ so that $K_{i,j}$ is the first complex including $\bar{\sigma}$. The idea is to use *preCompChanges* to update the i-th row of S_{M_1,M_2} at each iteration. This can be done because $K_{i,j} = h^{-1}_{M_1}\big((-\infty, r^1_i]\big) \cap h^{-1}_{M_2}\big((-\infty, r^2_j]\big)$, so $\chi(K_{i,j})$ and $\chi(K_{i,j+1})$ can differ by a change $\Delta\chi^{\bar{\sigma}}$ induced by $\bar{\sigma}$ if and only if $N^{\bar{\sigma}}$ in $K_{i,j+1}$ has changed, i.e. if there is a top-dimensional cube $\sigma' \in N^{\bar{\sigma}}$ such that $h_{M_2}(\sigma') = r^2_{j+1}$. But all such changes depend on the h_{M_2} values of top-dimensional cubes in $N^{\bar{\sigma}}$ greater than r^2_j. Sorting and storing these in $thresholds_2$ with $m_2 + 1$ appended, it follows that the ranges of $\hat{j}$-th columns of S_{M_1,M_2} such that $\hat{j}$ is between two consecutive values of $thresholds_2$ are such that the Euler characteristic change induced by adding $\bar{\sigma}$ is constant because $N^{\bar{\sigma}}$ does not change. So the elements of vector *preCompChanges* can be used on line 13 to update all $\hat{j}$ columns such that $\hat{j} \geq j$.

In conclusion, at the end of the loop on lines $3-16$, each entry $\mathrm{S}_{M_1,M_2}[i][j]$ equals the change $\chi(K_{i,j}) - \chi(K_{i-1,j})$, because all changes $\Delta\chi^{\bar{\sigma}}$ induced by the top-dimensional $\bar{\sigma}$ in $K_{i,j} \setminus K_{i-1,j}$ have been considered. After the cumulative sum on columns of S_{M_1,M_2}, it follows that

$$\begin{aligned} \mathrm{S}_{M_1,M_2}[i][j] =& \Big(\chi(K_{0,j}) - \chi(\emptyset)\Big) + \ldots + \Big(\chi(K_{i,j}) - \chi(K_{i-1,j})\Big) \\ =& \chi(K_{i,j}) - \chi(\emptyset) = \chi(K_{i,j}), \end{aligned} \tag{3.11}$$

which is the required Euler characteristic surface entry.

Proposition 3.4.1. *Let M_1 and M_2 be two-dimensional (three-dimensional) gray-scale images with the same size, and values in $[0, m_1]$ and $[0, m_2]$ respectively. The Euler characteristic surface $S_{M_1 M_2}$ of the pair M_1, M_2 can be computed with Algorithm 3.1, which has worst-case complexity $O(nm_2+m_1m_2)$, where n is the number of pixels (voxels) in M_1 and M_2.*

Proof. The above discussion proves the correctness of Algorithm 3.1 for the computation of $\mathrm{S}_{M_1 M_2}$. The outer loop on line 3 iterates on the n pixels (voxels) of M_1 and M_2, while the inner loop on lines $8-15$ takes $O(m_2)$ operations in the worst case to update an entire row. Finally, the cumulative sum on line 17 takes $O(m_1m_2)$ operations. So, the worst-case complexity of Algorithm 3.1 is $O(nm_2 + m_1m_2)$. □

Remark. Compared to computing m_2 Euler characteristic curves as proposed by the naïve approach at the beginning of this section, the neighbourhood $N^{\bar{\sigma}}$ is computed only once for ranges of columns where it does not change. Moreover, the entries of S_{M_1,M_2} are incremented and decremented without having to count subfaces of top-dimensional cubes in $N^{\bar{\sigma}}$.

3.5 Algorithm for Point Data

Given a finite set set of points X and an abstract simplicial complex K on X, Algorithm 3.2 computes the Euler characteristic surface $\mathrm{S}^{\mathbf{h}}_{\mathcal{R}_1,\mathcal{R}_2}$ of the sublevel sets bi-filtration $K^{\mathbf{h}}_{\mathcal{R}_1,\mathcal{R}_2}$ of a given $\mathbf{h} = (h_1, h_2) : K \to \mathbb{R}^2$ on two sets of monotonically increasing

Algorithm 3.2 Euler characteristic surface of bi-filtration on finite point set.

Input: abstract simplicial complex K, $\mathbf{h} = (h_1, h_2) : K \to \mathbb{R}^2$, and sorted values in $\mathcal{R}_1$ and $\mathcal{R}_2$.

```
 1: $\mathrm{S}^{\mathbf{h}}_{\mathcal{R}_1,\mathcal{R}_2} \leftarrow (m_1 + 1) \times (m_2 + 1)$ zeros matrix
 2: for each simplex $\sigma$ in $K$ do
 3:     $v_1, v_2 \leftarrow h_1(\sigma),\ h_2(\sigma)$
 4:     $r^1_i, r^2_j \leftarrow$ minimum values greater than $v_1$, $v_2$ in $\mathcal{R}_1$, $\mathcal{R}_2$ with binary search
 5:     for $\hat{j} = j$ to $m_2$ do
 6:         $\mathrm{S}^{\mathbf{h}}_{\mathcal{R}_1,\mathcal{R}_2}[i][\hat{j}] \leftarrow (-1)^{dim(\sigma)}$
 7:     end for
 8: end for
 9: $\mathrm{S}^{\mathbf{h}}_{\mathcal{R}_1,\mathcal{R}_2} \leftarrow$ cumulative sum on columns of $\mathrm{S}^{\mathbf{h}}_{\mathcal{R}_1,\mathcal{R}_2}$
10: return $\mathrm{S}^{\mathbf{h}}_{\mathcal{R}_1,\mathcal{R}_2}$
```

real values $\mathcal{R}_1 = \{r^1_i\}_{i=0}^{m_1}$ and $\mathcal{R}_2 = \{r^2_j\}_{j=0}^{m_2}$. The euchar Python package provides an implementation of this algorithm, which is applied in the next section.

Discussion. In this case, when a simplex σ is added into a $K_{i,j} = h_1^{-1}\big((-\infty, r^1_i]\big) \cap h_2^{-1}\big((-\infty, r^2_j]\big)$ its neighbourhood does not have a fixed structure. Thus it is not possible to precompute Euler characteristic changes as in Algorithm 3.1. However, if $\sigma \in K_{i,j}$, then $\sigma \in K_{i,\hat{j}}$ for each $\hat{j} \geq j$. So the change in Euler characteristic $(-1)^{dim(\sigma)}$, produced by adding σ into $K_{i,j}$, also applies to $K_{i,\hat{j}}$ for each $\hat{j} \geq j$. This property is used on line 6 of Algorithm 3.2 to update the i-th row of $\mathrm{S}^{\mathbf{h}}_{\mathcal{R}_1,\mathcal{R}_2}$ for each σ. It follows that at the end of the loop on lines $2 - 8$ each entry $\mathrm{S}^{\mathbf{h}}_{\mathcal{R}_1,\mathcal{R}_2}[i][j]$ equals $\chi(K_{i,j}) - \chi(K_{i-1,j})$, and the cumulative sum on columns of on line 9 returns the Euler characteristic surface of $K^{\mathbf{h}}_{\mathcal{R}_1,\mathcal{R}_2}$.

Proposition 3.5.1. *Let $K^{\mathbf{h}}_{\mathcal{R}_1,\mathcal{R}_2}$ be a sublevel sets bi-filtration of an abstract simplicial complex K. The Euler characteristic surface $S^{\mathbf{h}}_{\mathcal{R}_1,\mathcal{R}_2}$ of $K^{\mathbf{h}}_{\mathcal{R}_1,\mathcal{R}_2}$ can be computed with Algorithm 3.2, which has worst-case complexity $O(n(\log_2(m_1) + m_2) + m_1 m_2)$, where n is the number of simplices of K, and m_1 and m_2 the numbers of values in $\mathcal{R}_1$ and $\mathcal{R}_2$ respectively.*

Proof. The correctness of the algorithm follows from the above discussion. The outer loop on line 2 iterates on the n simplices in K. Then, within this loop the indices i and j are found with binary search, taking $O(\log_2(m_1) + \log_2(m_2))$ operations, and used to update row i of $\mathrm{S}^{\mathbf{h}}_{\mathcal{R}_1,\mathcal{R}_2}$ with at most m_2 operations in the inner loop on lines $5 - 7$. Thus, the complexity of lines $2 - 8$ is $O(n \cdot (\log_2(m_1) + m_2))$. Finally, the cumulative sum of line 9 takes $O(m_1 m_2)$ operations. So, the worst-case complexity of Algorithm 3.2

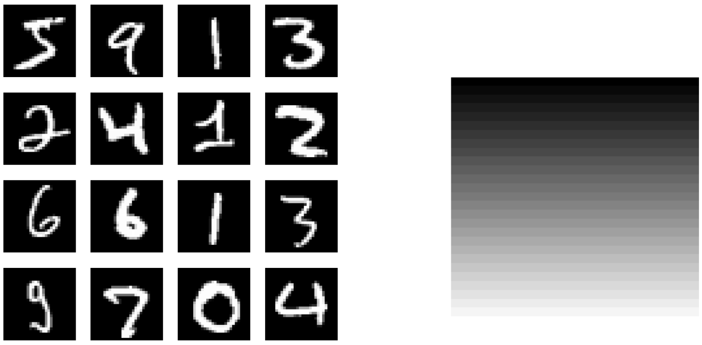
is $O(n(log_2(m_1) + m_2) + m_1 m_2)$, where n is the number of simplices in K, and m_1 and m_2 the number of values in $\mathcal{R}_1$ and $\mathcal{R}_2$ respectively. □

3.6 Experiments

Several experiments are presented that illustrate the additional information encoded by Euler characteristic surfaces compared to Euler characteristic curves. For image and point data, it is found that regions of the bi-dimensional parameter space, onto which bi-filtrations are defined, are useful in distinguishing between elements belonging to different classes of a given dataset. Algorithms 3.1 and 3.2 are used to compute the Euler characteristic surfaces of two and three-dimensional gray-scale images, and finite point sets in $\mathbb{R}^2$, by means of the implementations provided by the `euchar` Python package.

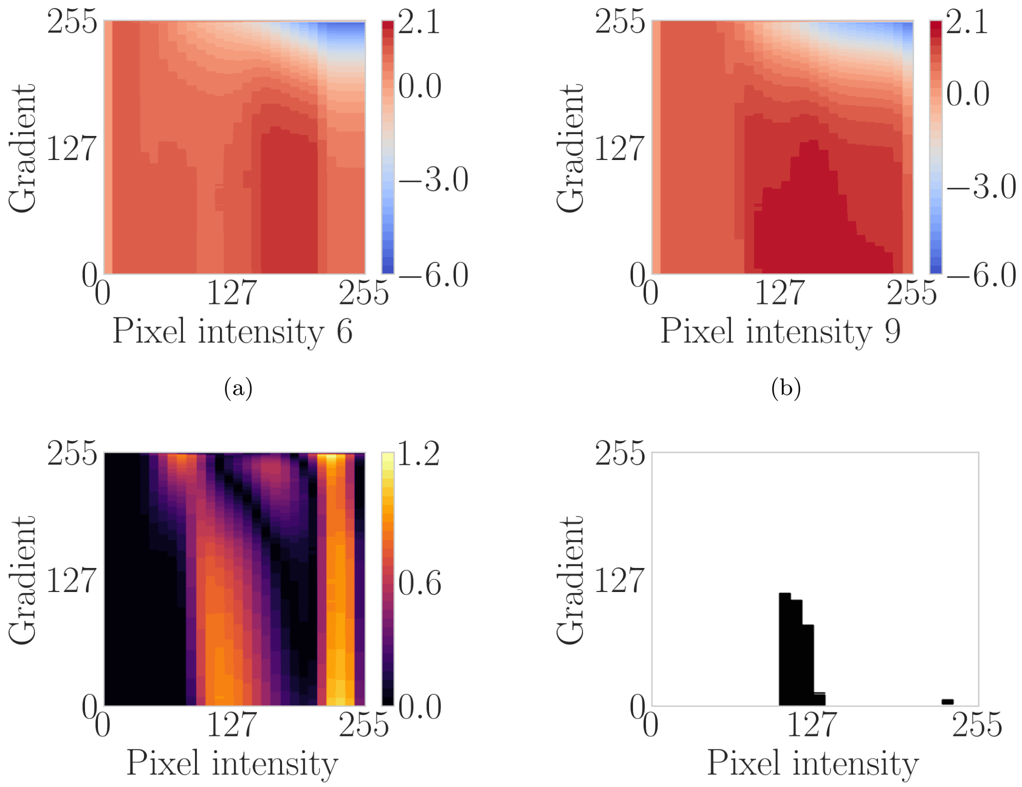
Handwritten digits images. The `MNIST` dataset of handwritten digits is an open-source collection of 28×28 gray-scale images with values in $[0, 255]$, see Figure 3.5a. It contains $60,000$ training and $10,000$ test images and is a standard tool used in benchmarking pattern recognition and machine learning algorithms [LBBH98].

In this setting Euler characteristic curves are expected to be non-informative in discriminating between some classes of images. For instance, take the sets of images representing a 6 and a 9 respectively, their average Euler characteristic curves cannot be used to distinguish between them. This happens because these two sets of `MNIST` images represent the same shape up to a rotation so that their pixel intensity sublevel sets have almost identical expected Euler characteristics. Luckily, the second parameterization used to define Euler characteristic surfaces can be used to account for this problem. Given the 28×28 top-down uniform gradient image G displayed in Figure 3.5b, the Euler characteristic surfaces of the pairs (M, G) for each `MNIST` image M were computed with Algorithm 3.1. The idea is that this gradient should help discriminate between the same shapes rotated by 180 degrees. The elementwise averages of surfaces representing the digits 6 and 9 are in Figures 3.6a and 3.6b respectively, and their difference in Figure 3.6c. Finally, Figure 3.6d shows the regions of the bi-dimensional parameter space $[0, 255] \times [0, 255]$ where the one standard deviation thickenings of these average surfaces are disjoint. These are indices (i, j) where the elementwise average minus one standard deviation of the surfaces representing a 6 is greater than the elementwise average plus one standard deviation of surfaces representing a 9, or vice versa. In this case, the average values of $\chi(K_{-,255})$ and $\chi(K_{255,-})$, i.e. average Euler characteristic curves, are such that their one standard deviations thickenings are not disjoint, while this is true in other regions of the bi-parameter space of Euler characteristic surfaces. Thus, utilizing the



(a) (b)

Figure 3.5: **(a)** `MNIST` images. **(b)** Top-down gradient image G.



(a) (b)

(c) (d)

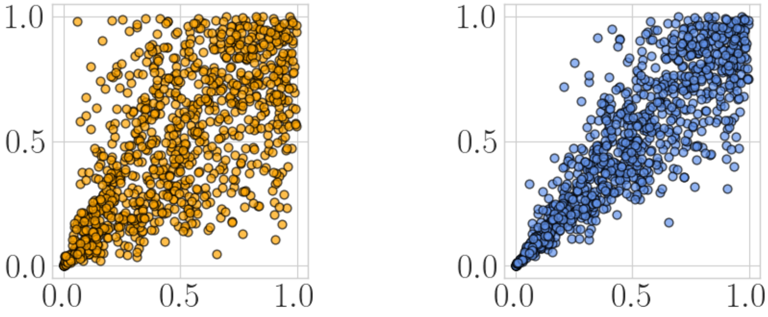
Figure 3.6: **(a)** Contour plots of average Euler characteristic surfaces of pairs (M, G), where G is the gradient in Figure 3.5b, and M is a `MNIST` images representing a 6. **(b)** Same as in (a), but for `MNIST` images representing a 9. **(c)** Absolute value of the difference of the average surfaces in (a) and (b). **(d)** The black areas are the regions of the parameter space where the one standard deviation thickenings of the average surfaces are disjoint.

gradient image G, it is possible to capture information that would otherwise be missed by single parameter Euler characteristic curves.

Random images with copula distributions. In Section 3.1 it is given an analytical expression, Equation (3.3.1), for the expected values of entries of Euler characteristic surfaces of $n_1 \times n_2$ random images. In that case a single parameter $0 \leq p \leq 1$ is used to regulate the strength of dependence within pairs of images. More generally, a pair of random images M_1, M_2 can be generated by sampling points in $\mathbb{R}^2$ according to a given bivariate distribution and setting the values of the entries of M_1 and M_2 equal to the coordinates of these randomly generated points. In the following, it is shown that average Euler characteristic surfaces of pairs of random three-dimension images generated from two given bivariate distributions are different, while their Euler characteristic curves are not.

A standard tool to define classes of multidimensional distributions are copula functions [Nel06], which can be used to join univariate marginal distribution functions. Here, the family of Clayton Archimedean copulas, with generator functions $\phi(t) = (t^{-\theta} - 1)/\theta$ for $\theta \in [-1 + \infty)$, is chosen and used to join a pair of univariate uniform distributions $\mathcal{U}(0, 1)$. The result is a collection of bivariate distributions parameterized by $\theta \in [-1, +\infty)$. In practice, the `copula` [HKMY20, Yan07] R package was used to sample random points from the two bivariate Clayton copula distributions with uniform marginals and $\theta = 1$ and $\theta = 5$ respectively. See Figure 3.7 for examples of such points. Then, pairs of three-dimensional $16 \times 16 \times 16$ gray-scale images M_1^θ, M_2^θ were generated by setting their entries to the coordinate values of sampled points. The expected Euler characteristic curve of any image M_1^θ or M_2^θ is constant because the bivariate distributions from which voxel intensity values are obtained have the same uniform marginals. On the other hand, average Euler characteristic surfaces of sublevel sets bi-filtrations of $\mathbf{h} = (h_{M_1^\theta}, h_{M_2^\theta})$, computed with Algorithm 3.1 over 50 pairs of random images, are different. Contour plots of these surfaces for $\theta = 1$ and $\theta = 5$ are in Figures 3.8a and 3.8b, and the absolute value of their difference in Figure 3.8c. Furthermore, the black area in Figure 3.8d represents the indices $(i, j) \in [0, 255] \times [0, 255]$ where the one standard deviation thickenings of the average surfaces are disjoint. Thus, as in the case of `MNIST` images above, indices such that either $i \neq 255$ or $j \neq 255$ are useful in distinguishing between the two given classes of data, while Euler characteristic curves are non-informative.

Poisson and Hawkes spatial processes. For this last experiment, average Euler characteristic surfaces of finite sets of points are compared. In particular, a homogeneous Poisson process and a Hawkes cluster process are used for generating random points in the unit square $[0, 1] \times [0, 1] \subseteq \mathbb{R}^2$ as described in [KB13]. The intensity parameter of the first process is set to $\lambda = 200$, while for the cluster process the intensity is $\lambda = 140$,



(a) (b)

Figure 3.7: Random points sampled in the unit square from bivariate distributions derived from a Clayton copula function with uniform marginals $\mathcal{U}(0,1)$. For the points in **(a)** the Clayton copula parameter is set to $\theta = 1$, while in **(b)** it is set to $\theta = 5$.

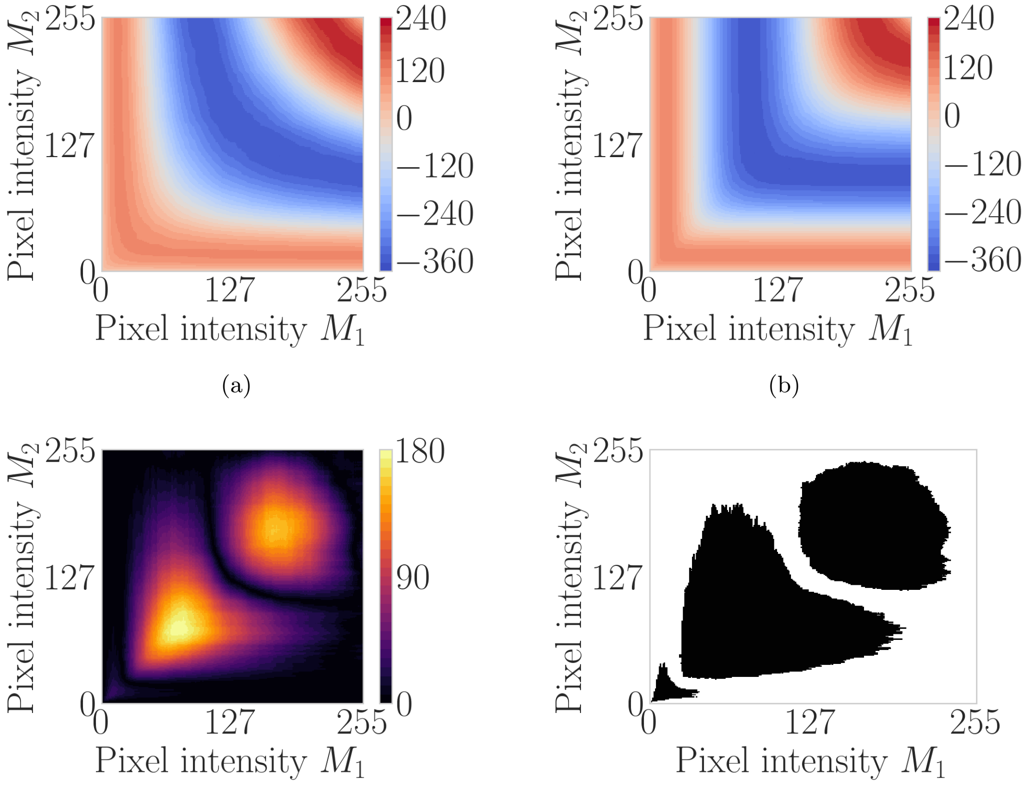


Figure 3.8: **(a)** Contour plots of average Euler characteristic surfaces of sublevel sets bi-filtration of pairs (M_1^θ, M_2^θ) for $\theta = 1$. **(b)** Same as in (a), but for $\theta = 5$. **(c)** Absolute value of the difference of the average surfaces in (a) and (b). **(d)** The black areas are the regions of the parameter space where the one standard deviation thickenings of the average surfaces are disjoint.

and the two parameters used in the definition of the offspring intensity function

$$\varrho(x_1, x_2) = \frac{\alpha}{2\pi\beta^2} \exp\left(-\frac{1}{2\beta^2}(x_1^2 + x_2^2)\right), \tag{3.12}$$

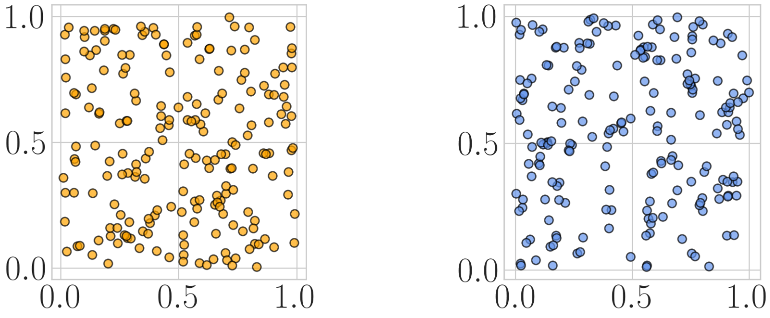
are set to $\alpha = 0.3$, $\beta = 0.02$. Figure 3.9b provides examples of finite point sets obtained from such spatial point processes. On these, Euler characteristic surfaces can be computed by defining an appropriate bi-filtering function $\mathbf{h}$ and sets $\mathcal{R}_1$, $\mathcal{R}_2$. In this case, it is used $\mathbf{h} = (h_1, h_2) : K^D \to \mathbb{R}^2$, where K^D is the Delaunay complex of the given finite point set, h_1 encodes information about local densities at points, and h_2 maps simplices to the radius of their minimal enclosing ball. An estimate of the inverse of the local density is obtained using the root mean square of the distances to its nearest-neighbours, that is to say

$$\text{dens}^k_{\text{inv}}(p) = \sqrt{\frac{d_1^2 + d_2^2 + \ldots + d_k^2}{k}}, \tag{3.13}$$

where d_i is the distance from any point p to its i-th nearest neighbour. In practice, it is set $h_1(\sigma) = \max_{p\in\sigma} \text{dens}^6_{\text{inv}}(p)$ and $h_2(\sigma) = h_{D\check{C}}(\sigma)$ for each $\sigma \in K^D$, where $h_{D\check{C}}$ is the Delaunay-Čech filtering function of Section 2.4. Besides, the values in $\mathcal{R}_1$ and $\mathcal{R}_2$ are defined so to subdivide the ranges $[0, \max_{\sigma\in K^D} h_1(\sigma)]$ and $[0, \max_{\sigma\in K^D} h_2(\sigma)]$ into 200 intervals of equal length. Finally, Euler characteristic surfaces are computed with Algorithm 3.2. A contour plot of the average Euler characteristic surface of $\mathbf{h}$, over 50 different point sets obtained from the Poisson process, is in Figure 3.10a. The same, but for the Hawkes cluster process, is in Figure 3.10b. As for the previous two experiments, the absolute value of the difference of these two surfaces is computed, and displayed in Figure 3.10c. Regions of the parameter space where the one standard deviation thickenings of the average surfaces are disjoint are represented by black areas in Figure 3.10d. In this case, some of the average values of $\chi(K_{-,200})$ and $\chi(K_{200,-})$, corresponding to those of Euler characteristic curves of h_1 and h_2 above, fall in regions of the parameter space where average surfaces are disjoint. However, there exist pairs of indices $i, j \in [0, 200] \times [0, 200]$ at which one standard deviation thickenings of the average surfaces are disjoint, while the same does not hold for any $(i, 200)$ and $(200, j)$. For example, this happens for (i, j) such that $(r^1_i, r^2_j) = (0.07, 0.08)$, which is the point marked by a red cross in Figure 3.10d. So Euler characteristic surfaces of $\mathbf{h}$ capture information that is not available in the Euler curve of h_1 nor in the Euler curve of h_2.

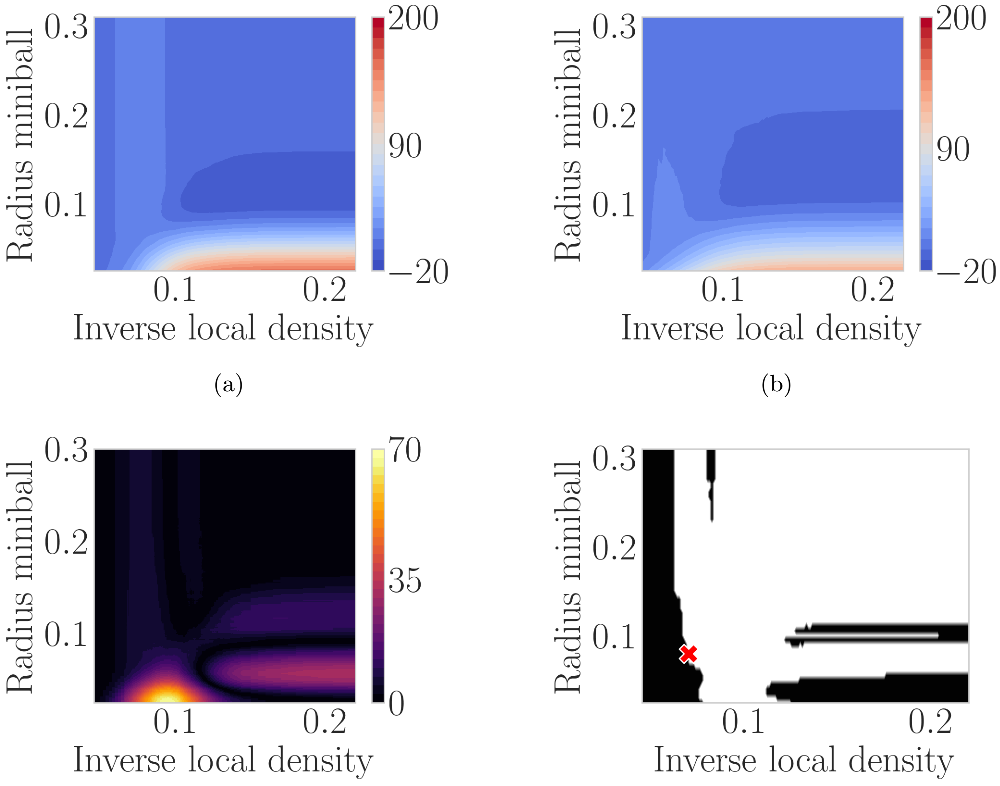
3.7 Discussion

The main contribution of this chapter is the introduction of Euler characteristic surfaces, which extend Euler characteristic curves to bi-filtrations, i.e. Cartesian products of



(a) (b)

Figure 3.9: **(a)** Points obtained from a homogeneous Poisson process with intensity $\lambda = 200$. **(b)** Points obtained from a Hawkes cluster process with intensity $\lambda = 140$ and offspring intensity parameters $\alpha = 0.3$ and $\beta = 0.02$.



(a) (b)

(c) (d)

Figure 3.10: **(a)** Contour plots of average Euler characteristic surfaces of $\mathbf{h} = (h_1, h_2) : K^D \to \mathbb{R}^2$ defined on the Delaunay complex of random points obtained from a homogeneous Poisson process. **(b)** Same as in (a), but for points obtained from a Hawkes cluster process. **(c)** Absolute value of the difference of the average surfaces in (a) and (b). **(d)** The black areas are the regions of the parameter space where the one standard deviation thickenings of the average surfaces are disjoint.

single-parameter filtrations. These can be used to characterize data over bi-dimensional parameter spaces. In particular, it is possible to obtain insights on the pairs of parameters that better distinguish between different types of data, that is to say the parameters maximising the difference in Euler characteristic between subcomplexes of different bi-filtrations. To illustrate this, we give various experiments on both real and synthetic data. These show how Euler characteristic surfaces identify regions of pairs of parameters discriminating between elements in different classes of a dataset, which would not be detected by Euler characteristic curves.

Furthermore, Algorithm 3.1 and Algorithm 3.2 are presented for the computation of Euler characteristic surfaces of image and point data. These have a worst-case running time of $O(nm_2 + m_1m_2)$ and $O(n(\log_2(m_1) + m_2) + m_1m_2)$ respectively, see Proposition 3.4.1 and Proposition 3.5.1. Note that the computation of these objects scales better than the one of persistence diagrams introduced in Chapter 2. In that case, Algorithm 2.2 takes $O(n_kn_{k+1}^2)$ time to compute the k-th persistence diagram of a filtration, where n_k is the number of k-simplices in K.

In Chapter 5 Euler characteristic surfaces are used to produce feature vectors from real-world data, which are then applied to classification tasks with standard machine learning methods. Moreover, these are compared against classification accuracy results obtained with feature vectors derived from persistence diagrams.

Chapter 4

Persistent Homology in ℓ_∞ Metric

This chapter studies the problem of computing the Čech persistent homology of a finite set of points S in ℓ_∞ metric space. The idea is to investigate whether or not filtrations of abstract simplicial complexes built out of nerves of ℓ_∞-balls (i.e. nerves of sets of axis-parallel hypercubes in a general dimension d) can be used to efficiently compute Čech persistence diagrams. In Euclidean metric space, it is known that Alpha filtrations can be used for such computations while restricting simplices to those of the Delaunay triangulation of S as discussed in Section 2.4. The main goals here are to find whether the same approach works in the ℓ_∞ metric setting, and possibly describe novel proximity filtrations that can be used to limit the size of Čech filtrations while producing the same persistence diagrams.

It should be noted that, the material presented in this chapter is part of [BS21]. Given a finite set of points $S \subseteq (\mathbb{R}^d, d_\infty)$, the contributions of this research project can be summarized as follows.

- Under genericity assumptions, i.e. the general position of S, Alpha complexes are proven to be equivalent to Čech complexes for points in two-dimensions, i.e. filtrations built with these complexes produce the same persistence diagrams. Moreover, it is given a counterexample of this equivalence for points in higher-dimensions.
- Alpha flag and Minibox filtrations are introduced and proven equivalent with Čech filtration in homological dimensions zero and one.
- Efficient algorithms are described for finding edges contained in Minibox complexes of two, three, and higher-dimensional points. In two dimensions, using a sweeping algorithm, it is shown a running time bound of $O(n^2)$ (which is optimal). In three

dimensions, it is achieved a worst-case bound of $O(n^2 \log(n))$ by extending the two-dimensional algorithm. In higher dimensions, using orthogonal range queries, the proposed algorithm has complexity $O(n^2 \log^{d-1}(n))$.

- For randomly sampled points in $\mathbb{R}^d$ the expected number of Minibox edges is proportional to $\Theta\left(\frac{2^{d-1}}{(d-1)!} n \log^{d-1}(n)\right)$. This is an improvement over the quadratic number of edges contained in Čech complexes and results in smaller filtrations. Interestingly, this implies that Minibox complexes are only a polylogarithmic factor larger than Euclidean Delanauy complexes of random points.
- We provide experimental evidence for speedups in the computation of persistence diagrams by means of Minibox filtrations.

While there is not as large a body of work on complexes in ℓ_∞ metric, as there is for Euclidean metric, there are several relevant related studies. In particular, approximations of ℓ_∞-Vietoris-Rips complexes are studied in [CKR17]. Moreover, the equivalence of the different complexes in zero and one homology is related to the results of [HKS15]. In this work offset filtrations of convex objects in two and three-dimensional space are considered. As in our case, an equivalence of filtrations is proven in homological dimensions zero and one by restricting offsets with Voronoi regions. While this result holds for general convex objects, Minibox filtrations can be used to reduce the size of ℓ_∞-Čech filtration in dimensions higher than three. Moreover, the approach presented here, which tries to constrain the number of edges of filtrations, is similar in spirit to the preprocessing step via collapses of [BP20], but works directly on the geometry of the given finite point set S.

4.1 ℓ_∞-Delaunay Edges

A characterization of ℓ_∞-Delaunay edges is given in terms of witness points, which are defined below. In the next section, this is used to show that and Alpha complexes of two-dimensional points in ℓ_∞ metric are flag complexes, as well as to prove their equivalence.

Recall from Chapter 2 that a box is an axis-parallel hyperrectangle, i.e. the Cartesian product of d intervals in $\mathbb{R}^d$, and the ε-thickening of a set $A \subseteq \mathbb{R}^d$ is $\varepsilon(A) = \{p \in \mathbb{R}^d \mid \min_{a \in A} d_\infty(a, p) \leq \varepsilon\}$. In particular, a ℓ_∞-ball of radius r is a box with sides of length $2r$, such that its ε-thickening is a box with sides of length $2r + 2\varepsilon$. Moreover, the Delaunay complex of S if denoted by K^D, and the Alpha and Čech filtrations of S by $K^A_{\mathcal{R}}$ and $K^{\check{C}}_{\mathcal{R}}$ respectively.

The following properties of ε-thickenings are needed for the main result of this section.

Proposition 4.1.1. (i) *Let $I_1, I_2 \subseteq \mathbb{R}$ be two non-empty closed intervals. If $I_1 \cap I_2 \neq \emptyset$, then $\varepsilon(I_1 \cap I_2) = \varepsilon(I_1) \cap \varepsilon(I_2)$.*

(ii) *Let $B_1, B_2 \subseteq \mathbb{R}$ be two non-empty boxes. If $B_1 \cap B_2 \neq \emptyset$, then $\varepsilon(B_1 \cap B_2) = \varepsilon(B_1) \cap \varepsilon(B_2)$.*

(iii) *Taking ε-thickenings preserves inclusions.*

(iv) *Let $\mathcal{A} = \{A\}_{i\in I}$ be a finite collection of sets. The ε-thickening of the union of sets in $\mathcal{A}$ is equal to the union of the ε-thickenings of sets in $\mathcal{A}$.*

Proof. (i) We have $I_1 = [a_1, b_1]$ and $I_2 = [a_2, b_2]$, with $I_1 \cap I_2 \neq \emptyset$. So either one of the two intervals is contained in the other or they share a common subinterval. In the first case, we can suppose without loss of generality that $I_1 \subseteq I_2$. Then $\varepsilon(I_1 \cap I_2) = \varepsilon(I_1) = [a_1 - \varepsilon, b_1 + \varepsilon] = [a_1 - \varepsilon, b_1 + \varepsilon] \cap [a_2 - \varepsilon, b_2 + \varepsilon] = \varepsilon(I_1) \cap \varepsilon(I_2)$. In the latter case, we can assume without loss of generality that $I_1 \cap I_2 = [a_2, b_1]$, and it follows $\varepsilon(I_1 \cap I_2) = [a_2 - \varepsilon, b_1 + \varepsilon] = [a_1 - \varepsilon, b_1 + \varepsilon] \cap [a_2 - \varepsilon, b_2 + \varepsilon] = \varepsilon(I_1) \cap \varepsilon(I_2)$.

(ii) Follows from property *(i)* and the definition of box in terms of Cartesian products, because ε-thickenings are in ℓ_∞ metric.

(iii) Consider $A, B \subseteq \mathbb{R}^d$ such that $A \subseteq B$. Given any $x \in \varepsilon(A) \setminus A$, by the definition of ε-thickening there exists $a \in A$ such that $d_\infty(x, a) \leq \varepsilon$. Then $x \in \varepsilon(B)$, because $a \in B$ and $d_\infty(x, a) \leq \varepsilon$. So $\varepsilon(A) \setminus A \subseteq \varepsilon(B)$, and because $A \subseteq B \subseteq \varepsilon(B)$ it follows that $\varepsilon(A) \subseteq \varepsilon(B)$.

(iv) Given a set $A \subseteq \mathbb{R}^d$, its ε-thickening is equivalently defined as $\varepsilon(A) = \bigcup_{x\in A} \overline{B_\varepsilon(x)}$. Thus

$$\varepsilon\left(\bigcup_{i\in I} A_i\right) = \bigcup_{x\in\bigcup_{i\in I} A_i} \overline{B_\varepsilon(x)} = \bigcup_{i\in I}\bigcup_{x\in A_i} \overline{B_\varepsilon(x)} = \bigcup_{i\in I} \varepsilon(A_i) \tag{4.1}$$

□

The concept of witness points is introduced next. The idea is to define these as the points in the intersection of ℓ_∞-Voronoi regions that can be used to characterize an edge as either belonging or not to a ℓ_∞-Delaunay complex.

Definition 4.1.2. A *witness* point of $\sigma \subseteq S$ is a point $z \in \mathbb{R}^d$ such that $z \in \bigcap_{p\in\sigma} V_p \neq \emptyset$, where V_p is the Voronoi region of p, and $d_\infty(z, p) = \max_{q\in\sigma} \frac{d_\infty(p,q)}{2}$ for each $p \in \sigma$. The set of witness points of σ is denoted by $\mathcal{Z}_\sigma$.

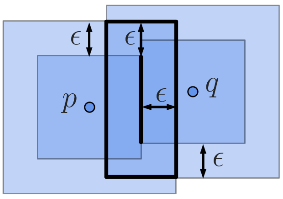


Figure 4.1: The ε-thickening of the non-empty intersection of two squares equals the intersection of the ε-thickenings of the squares.

The following result is presented as given in [BS21].

Proposition 4.1.3. *Let S be a finite set of points in $(\mathbb{R}^d, d_\infty)$ and $e = \{p, q\} \subseteq S$. Defined $A_e^r = \partial\overline{B_r(p)} \cap \partial\overline{B_r(q)}$ for $r > 0$, then $A_e^{\bar{r}} = \overline{B_{\bar{r}}(p)} \cap \overline{B_{\bar{r}}(q)}$, where $\bar{r} = \frac{d_\infty(p,q)}{2}$, which is a non-empty box. Moreover, the set of witness points of e is $\mathcal{Z}_e = A_e^{\bar{r}} \setminus \left(\bigcup_{y\in S\setminus e} B_{\bar{r}}(y)\right)$, and e is an edge of the ℓ_∞-Delaunay complex of S if and only if $\mathcal{Z}_e$ is non-empty.*

Proof. $A_e^{\bar{r}}$ is the intersection of the boundaries of the closed balls $\overline{B_{\bar{r}}(p)}$ and $\overline{B_{\bar{r}}(q)}$, which are axis-parallel hypercubes. So we have $A_e^{\bar{r}} \subseteq \overline{B_{\bar{r}}(p)} \cap \overline{B_{\bar{r}}(q)}$, because $\partial\overline{B_{\bar{r}}(p)} \subseteq \overline{B_{\bar{r}}(p)}$ and $\partial\overline{B_{\bar{r}}(q)} \subseteq \overline{B_{\bar{r}}(q)}$. Moreover $\overline{B_{\bar{r}}(p)} \cap \overline{B_{\bar{r}}(q)}$ is non-empty by definition of $\bar{r}$ and $\overline{B_{\bar{r}}(p)} \cap \overline{B_{\bar{r}}(q)} \subseteq \partial\overline{B_{\bar{r}}(p)} \cap \partial\overline{B_{\bar{r}}(q)} = A_e^{\bar{r}}$, because we can show a contradiction if $\left(\overline{B_{\bar{r}}(p)} \cap \overline{B_{\bar{r}}(q)}\right) \setminus A_e^{\bar{r}}$ is non-empty. In particular, given $y \in \left(\overline{B_{\bar{r}}(p)} \cap \overline{B_{\bar{r}}(q)}\right) \setminus A_e^{\bar{r}}$, then $d_\infty(y, p) \le \bar{r}$, $d_\infty(y, q) \le \bar{r}$, and at least one of these two distances must be strictly less than r, i.e. $d_\infty(y, p) < \bar{r}$ or $d_\infty(y, q) < \bar{r}$. Applying the triangular inequality to these distances it follows $\bar{r} + \bar{r} > d_\infty(p, y) + d_\infty(q, y) \ge d_\infty(p, q) = 2\bar{r}$, which is the desired contradiction. Thus $A_e^{\bar{r}} = \overline{B_{\bar{r}}(p)} \cap \overline{B_{\bar{r}}(q)}$ is a non-empty box, which is the Cartesian product of the intervals defining $\overline{B_{\bar{r}}(p)}$ and $\overline{B_{\bar{r}}(q)}$, because Cartesian products and intersections commute.

Furthermore

$$\begin{aligned} A_e^{\bar{r}+\varepsilon} &= \partial\overline{B_{\bar{r}+\varepsilon}(p)} \cap \partial\overline{B_{\bar{r}+\varepsilon}(q)} \\ &\subseteq \overline{B_{\bar{r}+\varepsilon}(p)} \cap \overline{B_{\bar{r}+\varepsilon}(q)} \\ &= \varepsilon(\overline{B_{\bar{r}}(p)}) \cap \varepsilon(\overline{B_{\bar{r}}(q)}) = \varepsilon(A_e^{\bar{r}}), \end{aligned} \tag{4.2}$$

because Proposition 4.1.1 *(ii)* can be applied to $\varepsilon(A_e^{\bar{r}}) = \varepsilon(\overline{B_{\bar{r}}(p)} \cap \overline{B_{\bar{r}}(q)})$, see Figure 4.1. Hence $A_e^{\bar{r}+\varepsilon} \subseteq \varepsilon(A_e^{\bar{r}})$ for any $\varepsilon \ge 0$, which is used below to prove the desired property of

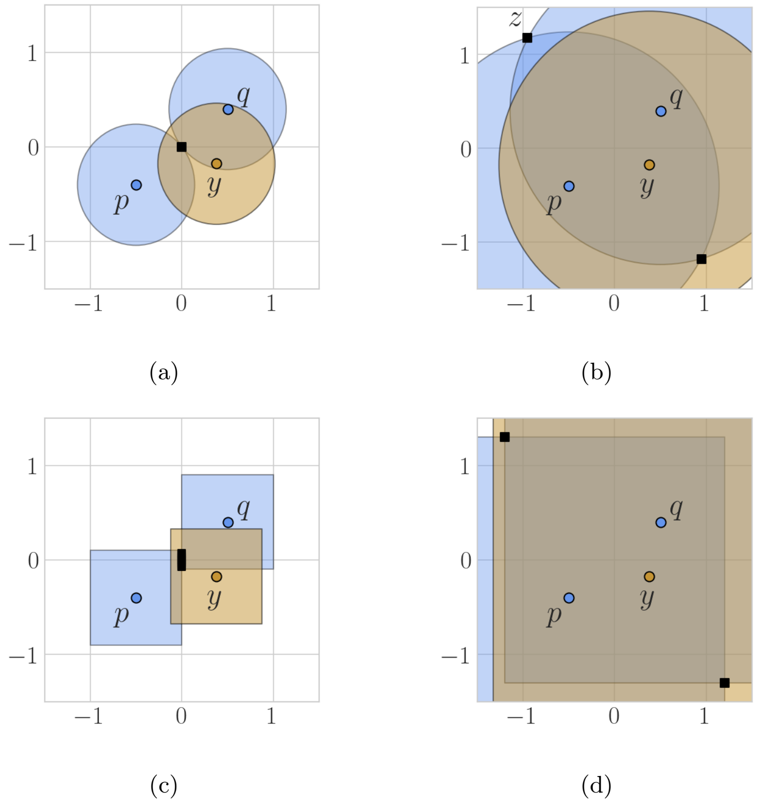


Figure 4.2: In **(a)** Euclidean balls centered in p and q intersect in a point which is covered by the ball centered in y. As the radius grows in **(b)** this intersection is not covered by the ball centered in y, so that $z \in V_p \cap V_q$ and $e = \{p, q\} \in K^D$. In **(c)** ℓ_∞-balls centered in p, q intersect in $A_e^{\bar{r}}$ which is covered by the ℓ_∞-ball centered in y. Again the radius grows in **(d)** but in this case the ℓ_∞-ball centered in y covers $A_e^{\bar{r}+\varepsilon}$.

witness points by contradiction.

First note that $\mathcal{Z}_\sigma = A_e^{\bar{r}} \setminus \big(\bigcup_{y \in S \setminus e} B_{\bar{r}}(y)\big)$ by definition of witness point, $A_e^{\bar{r}}$ and $\bar{r}$. The two directions of the equivalence are proven separately.

($\Rightarrow$) The pair $e = \{p, q\}$ is a Delaunay edge, so $V_p \cap V_q \neq \emptyset$. Equivalently there exist $\varepsilon \geq 0$ and $z \in \mathbb{R}^d$ such that $z \in A_e^{\bar{r}+\varepsilon} \setminus \big(\bigcup_{y \in S \setminus e} B_{\bar{r}+\varepsilon}(y)\big)$, where $\bar{r} = \frac{d_\infty(p,q)}{2}$, because

$$V_p \cap V_q = \bigcup_{\varepsilon \geq 0} A_e^{\bar{r}+\varepsilon} \setminus \Big(\bigcup_{y \in S \setminus e} B_{\bar{r}+\varepsilon}(y)\Big). \tag{4.3}$$

Suppose that $A_e^{\bar{r}}$ is covered by $\bigcup_{y \in S \setminus e} B_{\bar{r}}(y)$, i.e. $\mathcal{Z}_e$ is empty. Then $A_e^{\bar{r}+\varepsilon} \subseteq \varepsilon(A_e^{\bar{r}})$ from Equation (4.2), and applying points *(iii)* and *(iv)* of Proposition 4.1.1 the following sequence of inclusions is obtained

$$A_e^{\bar{r}+\varepsilon} \subseteq \varepsilon\left(A_e^{\bar{r}}\right) \subseteq \varepsilon\left(\bigcup_{y\in S\setminus e} B_{\bar{r}}(y)\right) = \bigcup_{y\in S\setminus e} B_{\bar{r}+\varepsilon}(y), \tag{4.4}$$

for any $\varepsilon \geq 0$. Thus $A_e^{\bar{r}+\varepsilon} \subseteq \bigcup_{y\in S\setminus e} B_{\bar{r}+\varepsilon}(y)$, which contradicts the existence of $z \in A_e^{\bar{r}+\varepsilon} \setminus \left(\bigcup_{y\in S\setminus e} B_{\bar{r}+\varepsilon}(y)\right)$ for any $\varepsilon \geq 0$.

($\Leftarrow$) Any point in $\mathcal{Z}_e \neq \emptyset$ belongs to $V_p \cap V_q$, so that $e \in K^D$. □

Figure 4.2 illustrates the inclusions in Equation (4.4). Moreover, it give an example showing that the same inclusions do not hold in the Euclidean case. The above result allows to determine if a pair of points forms an edge in the ℓ_∞-Delaunay complex K^D of S by checking whether $A_e^{\bar{r}}$ is covered or not by a union of ℓ_∞-balls.

4.2 Alpha Complexes

Given a finite set of points S in Euclidean space, it is known that the Alpha filtration $K_{\mathcal{R}}^A$ produces the same persistence diagrams of the Čech filtration $K_{\mathcal{R}}^{\check{C}}$, see Section 2.4. Moreover, $K_{\mathcal{R}}^A$ restricts the simplices to those of the Delaunay complex K^D, thus speeding up the computation of the Čech persistence diagrams of $S \subseteq (\mathbb{R}^d, d_2)$. Nonetheless, this requires finding the $O(n^{\lceil \frac{d}{2} \rceil})$ top-dimensional simplices of K^D, which can be done efficiently only in low-dimensions [HB08]. In this section, Alpha filtrations of points in ℓ_∞ metric are proven to be equivalent to Čech filtrations for $d = 2$. Moreover, in two-dimensions Alpha filtrations are shown to be sequences of flag complexes, so that they are completely determined by ℓ_∞-Delaunay edges. Counterexamples of both these properties are given for higher-dimensional points.

Alpha Filtrations in $\mathbb{R}^2$. For the following two results, the two-dimensional finite set of points S is assumed to be in general position as defined in Chapter 2, i.e. pairwise distance between points are distinct, no four points lie on the boundary of a square, no three points are collinear, and no two points have same x or y coordinates. The following novel result is stated as in Section 3 of the preprint [BS21].

Theorem 4.2.1. *Let S be a finite set of points in $(\mathbb{R}^2, d_\infty)$ in general position. The Alpha and Čech filtrations of S are equivalent, i.e. produce the same persistence diagrams.*

Proof. Alpha complexes K_r^A are nerves of collections of closed sets $\{\overline{B_r(p)} \cap V_p\}_{p\in S}$ for $r \in \mathbb{R}$. We show that any intersection of k elements in any such collection is either empty or contractible.

- $k = 2$. Let p, q be two points of S, and $\bar{r} = \frac{d_\infty(p,q)}{2}$. We show that

$$L = \overline{B_r(p)} \cap V_p \cap \overline{B_r(q)} \cap V_q, \tag{4.5}$$

  is either empty or contractible. In $\mathbb{R}^2$ we have that $A_e^{\bar{r}} = \overline{B_{\bar{r}}(p)} \cap \overline{B_{\bar{r}}(q)}$ is a line segment of length strictly less than $2\bar{r}$, by our general position assumption. If this line segment is covered by $\bigcup_{y \in S \setminus \{p,q\}} B_{\bar{r}}(y)$, then by Proposition 4.1.3 we have that $V_p \cap V_q$ is empty, so that L is empty. Moreover L is empty if $r < \bar{r}$, because $\overline{B_r(p)} \cap \overline{B_r(q)}$ is.

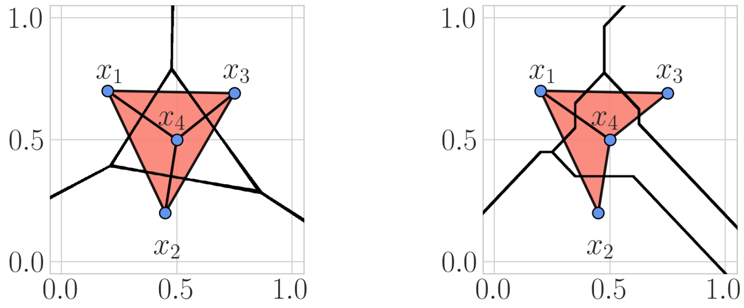
  On the other hand, if $r \geq \bar{r}$ and $A' = A_e^{\bar{r}} \setminus \bigcup_{y \in S \setminus \{p,q\}} B_{\bar{r}}(y)$ is a non-empty line segment, we show that L is contractible. First, we define a deformation retraction ϕ of $V_p \cap V_q$ onto A' as the Euclidean projection of $(V_p \cap V_q) \setminus A'$ onto $(V_p \cap V_q) \cap A'$. This can be done because $(V_p \cap V_q) \setminus A'$ contains a maximum of two line segments, defined by the union of points in $\partial\overline{B_{\bar{r}+\varepsilon}(p)} \cap \partial\overline{B_{\bar{r}+\varepsilon}(q)}$ not contained in $\bigcap_{y \in S \setminus \{p,q\}} B_{\bar{r}+\varepsilon}(y)$ for any $\varepsilon > 0$. For instance, consider the bisector $V_p \cap V_q$ in Figure 2.6c given in Chapter 2 to illustrate the non-convexity of ℓ_∞-Voronoi regions. In this case, ϕ retracts the two line segments oriented at a forty-five degree angle onto the horizontal line segment, which equals $A_e^{\bar{r}} = A'$. Moreover, ϕ restricts to L, by the convexity of $\overline{B_r(p)} \cap \overline{B_r(q)}$ for any $r > 0$, and the fact that this contains A' for $r \geq \bar{r}$. Hence L has the same homotopy type of A', which is a line segment, and so is contractible.

- $k = 3$. These intersections can either be empty or contain a single point by the general position of S.

- $k > 3$. Any such intersection is empty, again by the general position of S.

Thus we can apply the Nerve Theorem 2.4.3, obtaining that $X = \bigcup_{p \in S} \left(\overline{B_r(p)} \cap V_p\right)$ and K_r^A are homotopy equivalent for any $r \in \mathbb{R}$. Besides $X = \bigcup_{p \in S} \overline{B_r(p)}$, and by applying the Nerve Theorem to the collection $\{\overline{B_r(p)}\}_{p \in S}$, we have that X is homotopy equivalent to $K_r^{\check{C}}$ as well. So $K_r^A \simeq K_r^{\check{C}}$ for any $r \in \mathbb{R}$, and the desired equivalence of Alpha and Čech filtrations follows by applying the Persistence Equivalence Theorem 2.4.11. □

This is similar to the results of [HKS15], which proves that the nerve of offsets of convex shapes is equivalent to the union of the shapes for zero and one-dimensional homology in two and three dimensions. Our argument using general position implies that no higher-dimensional homology can appear in the nerve. In particular, the theorem implies that Alpha filtrations of two-dimensional points produce equivalent persistence diagrams to Čech filtrations. Hence, the above result ensures that the two-dimensional homology of Alpha complexes of $S \subseteq \mathbb{R}^2$ is trivial, because it equals the one of the two-



(a) (b)

Figure 4.3: Voronoi diagrams and Delaunay triangulations of four points in $\mathbb{R}^2$, with Euclidean and ℓ_∞ metric in **(a)** and **(b)** respectively.

dimensional sets $\bigcup_{p\in S}\overline{B_r(p)}$. At the end of this section, it is shown that in general this is not the case for three-dimensional points, and so for any set of points in dimension $d \geq 3$.

In order to construct the Alpha filtration of $S \subseteq (\mathbb{R}^2, d_\infty)$ in general position, the ℓ_∞-Delaunay triangulation of S is needed. Its simplices can be found with the $O(n\log(n))$ plane-sweep algorithm of [SDT91], but it is also necessary to find the radius parameter r_i of each simplex $\sigma \in K^D$ to build the Alpha filtration, i.e. the minimum $r_i > 0$ such that $\bigcap_{p\in\sigma}\left(\overline{B_{r_i}(p)} \cap V_p\right) \neq \emptyset$. Luckily, from the next result if follows that this is $\frac{\max_{p,q\in\sigma} d_\infty(p,q)}{2}$ for each $\sigma \in K^D$, i.e. half the edge length of the longest edge in σ. Thus information about ℓ_∞-Delaunay edges is all that is needed to build Alpha filtrations of points in $\mathbb{R}^2$, and compute their persistence diagrams. Figure 4.3 illustrates the differences between Euclidean and ℓ_∞-Delaunay triangulations. The following result is presented as given in [BS21].

Proposition 4.2.2. *Let S be a finite set of points in general position in $(\mathbb{R}^2, d_\infty)$ and $r \geq 0$. Both the Delaunay complex K^D and the Alpha complex K_r^A of S are flag complexes. Moreover, given an edge $e = \{p, q\} \in K^D$, then $e \in K_r^A$ if and only if $\frac{d_\infty(p,q)}{2} \leq r$.*

Proof. Consider three points $x_1, x_2, x_3 \subseteq S$, such that $\{x_1, x_2\}$, $\{x_1, x_3\}$ and $\{x_2, x_3\}$ are ℓ_∞-Delaunay edges. Without loss of generality, we can assume $\{x_1, x_2\}$ to be the longest edge. Defined $\bar{r} = \frac{d_\infty(x_1,x_2)}{2}$, and $A^{\bar{r}}_{x_1x_2} = \partial\overline{B_{\bar{r}}(x_1)} \cap \partial\overline{B_{\bar{r}}(x_2)}$, by Proposition 4.1.3 we have that $A^{\bar{r}}_{x_1x_2} = \overline{B_{\bar{r}}(x_1)} \cap \overline{B_{\bar{r}}(x_2)}$. This is a non-empty axis-parallel line segment of length less than $2\bar{r}$ by the general position assumption. Moreover, by definition of $\bar{r}$, the intersections $\overline{B_{\bar{r}}(x_1)} \cap \overline{B_{\bar{r}}(x_2)}$, $\overline{B_{\bar{r}}(x_1)} \cap \overline{B_{\bar{r}}(x_3)}$, and $\overline{B_{\bar{r}}(x_2)} \cap \overline{B_{\bar{r}}(x_3)}$ are non-empty. So the intersection $A^{\bar{r}}_{x_1x_2} \cap \overline{B_{\bar{r}}(x_3)} \neq \emptyset$ by property *(ii)* of Proposition 2.1.2.
If $A^{\bar{r}}_{x_1x_2} \setminus \overline{B_{\bar{r}}(x_3)} = \emptyset$, then $A^{\bar{r}}_{x_1x_2}$ is covered by $B_{\bar{r}}(x_3)$, which is in contradiction with

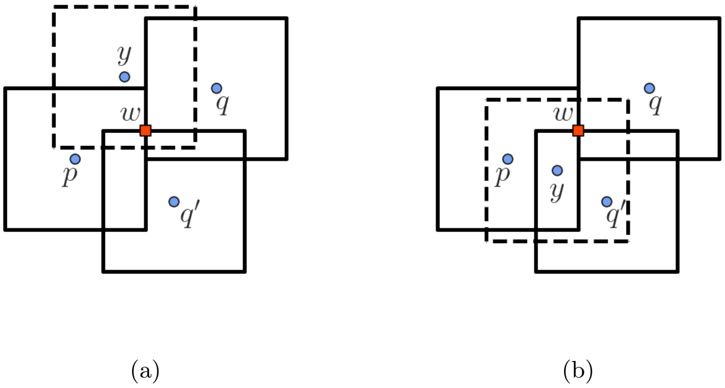


Figure 4.4: Illustration of the last two cases of the proof of Proposition 4.2.2. In **(a)** the red square marker represents point $(a_1, \hat{b}_2)$ on $A^{\bar{r}}_{x_1x_2}$, which is covered by $B_{\bar{r}}(y')$ from above. In **(b)** the same point is covered by $B_{\bar{r}}(y')$ from below. In both **(a)** and **(b)** the boundary of $B_{\bar{r}}(y')$ is drawn as a dashed line.

$\{x_1, x_2\}$ being a Delaunay edge from Proposition 4.1.3.
On the other hand, if $A^{\bar{r}}_{x_1x_2} \setminus \overline{B_{\bar{r}}(x_3)} \neq \emptyset$, then the line segment $A^{\bar{r}}_{x_1x_2}$ must intersect the boundary of the square $\overline{B_{\bar{r}}(x_3)}$. Defined $\tau = \{x_1, x_2, x_3\}$ and $A^{\bar{r}}_\tau = A^{\bar{r}}_{x_1x_2} \cap \partial\overline{B_{\bar{r}}(x_3)}$, we have that the set of witness points of τ is $\mathcal{Z}_\tau = A^{\bar{r}}_\tau \setminus \left(\bigcup_{y\in S\setminus\tau} B_{\bar{r}}(y)\right)$. Hence, if $\mathcal{Z}_\tau$ is non-empty, we can conclude that the Delaunay complex of S is a clique complex from the definition of witness point. We suppose by contradiction that $\mathcal{Z}_\tau = \emptyset$, and show that in every possible case one between $\{x_1, x_2\}$, $\{x_1, x_3\}$, and $\{x_2, x_3\}$ cannot be a ℓ_∞-Delaunay edge.

We know that the axis-parallel square $\overline{B_{\bar{r}}(x_3)}$ intersects $A^{\bar{r}}_{x_1x_2}$ without covering it, so that $A^{\bar{r}}_\tau$ is a point by our general position assumption. To simplify the exposition, we assume without loss of generality $A^{\bar{r}}_{x_1x_2}$ to be a vertical line segment in $\mathbb{R}^2$, and $\overline{B_{\bar{r}}(x_3)}$ to be intersecting $A^{\bar{r}}_{x_1x_2}$ from below. More precisely, given $x_1 = (x_1^1, x_2^1)$, $x_2 = (x_1^2, x_2^2)$, and $x_3 = (x_1^3, x_2^3)$, we assume $d_\infty(x_1, x_2) = |x_1^1 - x_1^2| = 2\bar{r} \geq |x_2^1 - x_2^2|$, and that $x_2^3 \leq \min\{x_2^1, x_2^2\}$. This implies

$$A^{\bar{r}}_\tau \subseteq A^{\bar{r}}_{x_1x_2} \cap \overline{B_{\bar{r}}(x_3)} = [a_1, a_1] \times [a_2, \hat{b}_2]$$

where $a_1 = \max\{x_1^1, x_1^2\} - \bar{r}$, $a_2 = \max\{x_2^1, x_2^2\} - \bar{r}$, $b_2 = \min\{x_2^1, x_2^2\} + \bar{r}$, and $\hat{b}_2 = x_2^3 + \bar{r}$. So $A^{\bar{r}}_\tau = (a_1, \hat{b}_2)$, and because we are assuming by contradiction that $A^{\bar{r}}_\tau$ is covered by balls of radius $\bar{r}$ centered in the points of $S \setminus \tau$, there exists $y' \in S \setminus \tau$ such that $(a_1, \hat{b}_2) \in B_{\bar{r}}(y')$. Finally, either $\overline{B_{\bar{r}}(y')}$ intersects $A^{\bar{r}}_{x_1x_2}$ from above or from below. These two cases are illustrated in Figure 4.4, where the boundary of $\overline{B_{\bar{r}}(y')}$ is represented as a dashed line, and the point $A^{\bar{r}}_\tau = (a_1, \hat{b}_2)$ as a red square marker. In the former case $\overline{B_{\bar{r}}(x_3)} \cup \overline{B_{\bar{r}}(y')}$ covers $A^{\bar{r}}_{x_1x_2}$, which is in contradiction with $\{x_1, x_2\}$ being a ℓ_∞-

Delaunay edge, by Proposition 4.1.3. In the latter case, given $y' = (y'^1, y'^2)$, we have $\min\{x_1^1, x_2^1\} < y'^1 < \max\{x_1^1, x_2^1\}$, and $x_3^2 < y'^2 < \min\{x_1^2, x_2^2\}$, because $\overline{B_{\bar{r}}(y')}$ intersects $A_{x_1x_2}^{\bar{r}}$ without covering it, and contains $(a_1, \hat{b}_2)$. Finally, the location of y' prevents either $\{x_1, x_3\}$ or $\{x_2, x_3\}$ from being a ℓ_∞-Delaunay edge. This follows from Proposition 4.1.3 because $\overline{B_{\bar{r}}(y')}$ covers either $A_{x_1x_3}^{\bar{r}_{13}}$ or $A_{x_2x_3}^{\bar{r}_{23}}$, where $\bar{r}_{13} = \frac{d_\infty(x_1,x_3)}{2}$ and $\bar{r}_{12} = \frac{d_\infty(x_1,x_2)}{2}$. Thus in every possible case the set $\mathcal{Z}_\tau$ must be non-empty, and K^D is a flag complex.

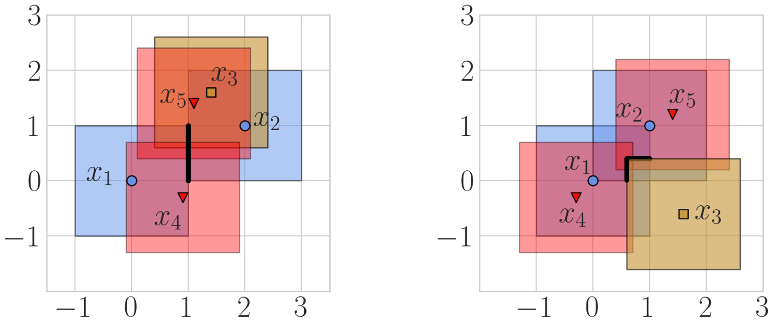
To conclude it is shown that K_r^A is also a flag complex. By Proposition 4.1.3 any ℓ_∞-Delaunay edge $e = \{p, q\}$ is added into the Alpha filtration at $\bar{r} = \frac{d_\infty(p,q)}{2}$. Moreover, when the longest edge of any Delaunay triangle τ is added at radius $\bar{r}$, also τ is added in $K_{\bar{r}}^A$, because from the discussion above there exist a point $A_\tau^{\bar{r}}$ at distance $\bar{r}$ from the vertices of τ, which is a witness of this triangle. □

Counterexample: Alpha complexes are not flag in higher dimensions. A counterexample to Proposition 4.2.2 is given for points in dimension three. Given $S = \{x_i\}_{i=1}^5 \subseteq (\mathbb{R}^3, d_\infty)$, where $x_1 = [0,0,0]$, $x_2 = [2,1,1]$, $x_3 = [1.4, 1.6, -0.6]$, $x_4 = [0.9, -0.3, -0.3]$, and $x_5 = [1.1, 1.4, 1.2]$, it is shown that the Alpha complex K_1^A of S is not a flag complex. In practice, the existence of witness points is used to prove that $\{x_1, x_2\}, \{x_1, x_3\}, \{x_2, x_3\} \in K_1^A$, and $\{x_1, x_2, x_3\} \notin K_1^A$. One can check that:

- $(1, 0, 1)$ is a witness of $\{x_1, x_2\}$ at distance 1 from x_1 and x_2.
- $(0.8, 0.8, 0.0)$ is a witness of $\{x_1, x_3\}$ at distance 0.8 from x_1 and x_3.
- $(1.5, 1.5, 0.2)$ is a witness of $\{x_2, x_3\}$ at distance 0.8 from x_2 and x_3.

Thus the pairs $\{x_1, x_2\}$, $\{x_1, x_3\}$, $\{x_2, x_3\}$ are edges of the Delaunay complex K^D, and edges of K_1^A by Proposition 4.2.2. On the other hand $\tau = \{x_1, x_2, x_3\}$ is not a triangle in K^D, and so does not belong to any Alpha complex. This follows from the fact that $A_\tau^1 = \partial\overline{B_1(x_1)} \cap \partial\overline{B_1(x_2)} \cap \partial\overline{B_1(x_3)}$ is formed by the two line segments, plotted as thickened lines in Figure 4.5, with endpoints $(1, 0.6, 0)$, $(1, 0.6, 0.4)$ and $(1, 0.6, 0.4)$, $(1, 1, 0.4)$, which are covered by $B_1(x_4) \cup B_1(x_5)$. The ε-thickenings of these line segments contain $A_\tau^{1+\varepsilon}$ for any $\varepsilon \geq 0$, by the properties of ε-thickenings used in the proof of Proposition 4.1.3. In turn, the ε-thickenings of the two line segments are contained in $\varepsilon(B_1(x_4) \cup B_1(x_4)) = B_{1+\varepsilon}(x_4) \cup B_{1+\varepsilon}(x_5)$. This implies that it does not exist a point $z \in V_{x_1} \cap V_{x_2} \cap V_{x_3}$, as this would require $A_\tau^{1+\varepsilon} \setminus \big(B_{1+\varepsilon}(x_4) \cup B_{1+\varepsilon}(x_5)\big)$ to be non-empty for some $\varepsilon \geq 0$.

Counterexample: Non-equivalence in higher dimensions. We conclude this section by providing a counterexample to the equivalence of Alpha and Čech filtrations in homological dimension higher than two. This is shown with a configuration of eight



(a) Projection along x and y axes.

(b) Projection along y and z axes.

Figure 4.5: Five points in $\mathbb{R}^3$ realising a counterexample to Delaunay complexes being flag complexes in dimensions higher than two. Projections along two pairs of axes are given. The thickened line segments represent $A^1_\tau = \partial B_1(x_1) \cap \partial B_1(x_2) \cap \partial B_1(x_3)$.

Table 4.1: Coordinates of points $S \subseteq (\mathbb{R}^3, d_\infty)$ giving a counterexample to the equivalence of Alpha and Čech filtrations in dimension higher than 2.

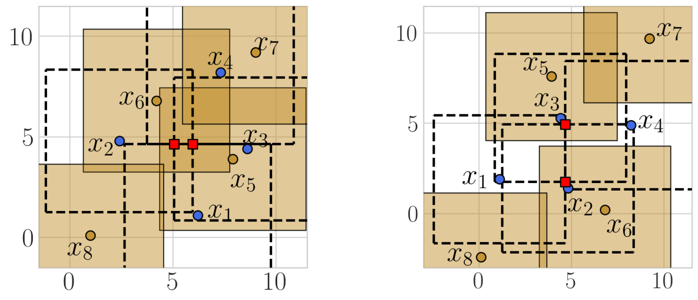
	x	y	z
x_1	6.2	1.1	1.9
x_2	2.4	4.8	1.4
x_3	8.6	4.4	5.3
x_4	7.3	8.2	4.9
x_5	7.9	3.9	7.6
x_6	4.2	6.8	0.2
x_7	9.0	9.2	9.7
x_8	1.0	0.1	-2.4

points $S = \{x_i\}_{i=1}^8 \subseteq \mathbb{R}^3$, the coordinate of which are listed in Table 4.1.

The points in S are such that their Delaunay complex contains the four faces of the tetrahedron $\{x_1, x_2, x_3, x_4\}$, but not the tetrahedron itself. This way the Alpha complexes of S never contain $\{x_1, x_2, x_3, x_4\}$ as a simplex, but for a big enough radius parameter they contain its the four faces. Moreover, the Delaunay complex of S also does not contain other tetrahedra that fill in the two-dimensional void created by the faces of $\{x_1, x_2, x_3, x_4\}$.

The points in this S were found by randomly sampling many sets of eight points in $\mathbb{R}^3$, and testing whether their Alpha and Čech persistence diagrams were equal. The existence of such a counterexample can be thought of as a consequence of the non-convexity of general ℓ_∞-Voronoi regions, even if one may hope the nerve of general Voronoi regions to be well behaved enough to prevent this from happening.

One can check that there are six tetrahedra belonging to the Delaunay complex K^D



(a) Projection along x and y axes.

(b) Projection along y and z axes.

Figure 4.6: Counterexample to the equivalence of Alpha and Čech persistent homology in ℓ_∞ metric. The two circumcenters of the tetrahedron $\{x_1, x_2, x_3, x_4\}$ are the red square markers. The boundaries of cubes centered in the vertices of $\{x_1, x_2, x_3, x_4\}$ are shown as dashed lines.

of S: $\{x_1, x_2, x_3, x_5\}$, $\{x_1, x_2, x_3, x_6\}$, $\{x_1, x_2, x_4, x_5\}$, $\{x_1, x_3, x_4, x_6\}$, $\{x_2, x_3, x_4, x_5\}$, and $\{x_2, x_3, x_4, x_6\}$. This can be done by finding the circumcenters of any four given points, and checking that the circumspheres of these (which in this case are cubes) do not contain any of the other points. It is important to note that in ℓ_∞ metric four three-dimensional points might have two distinct circumcenters. For instance this is the case for $\{x_1, x_2, x_3, x_4\}$, the circumcenters of which are represented as red square markers in Figure 4.6, having coordinates $w_1 = (5.95, 4.65, 1.75)$ and $w_2 = (5.05, 4.65, 4.95)$. On the other hand, in Euclidean metric four affinely independent three-dimensional points have exactly one circumcenter. Moreover, w_1 and w_2 are not witnesses of $\{x_1, x_2, x_3, x_4\}$, because they are closer to x_5 and x_6 than to the vertices of this tetrahedron. Thus $\{x_1, x_2, x_3, x_4\} \notin K^D$. Regarding the faces of $\{x_1, x_2, x_3, x_4\}$, one can check that:

- $(5.5, 4.2, 3.9)$ is a witness of $\{x_1, x_2, x_3\}$ at distance 3.1 from x_1, x_2, and x_3.
- $(4.05, 4.65, 4.95)$ is a witness of $\{x_1, x_2, x_4\}$ at distance 3.55 from x_1, x_2, and x_4.
- $(8.75, 4.65, 1.75)$ is a witness of $\{x_1, x_3, x_4\}$ at distance 3.55 from x_1, x_3, and x_4.
- $(5.5, 5.1, 3.9)$ is a witness point of $\{x_2, x_3, x_4\}$ at distance 3.1 from x_2, x_3, and x_4.

The tetrahedra belonging to the Delaunay complex of S (listed in the above discussion) do not create a boundary to the two-dimensional homology class created by adding $\{x_1, x_2, x_3\}$, $\{x_1, x_2, x_4\}$, $\{x_1, x_3, x_4\}$, and $\{x_2, x_3, x_4\}$ into K^A_r, for $r > 0$ big enough. Thus the two-dimensional persistence diagram of the Alpha filtration of S has a point at infinity, i.e. an homology class that never dies. On the other hand, the two-dimensional persistence diagrams of the Čech filtration of S cannot have such a point, because Čech complexes have trivial homology for a big enough radius.

4.3 Alpha Flag Complexes

In the previous section, we have seen that Alpha filtrations can be used to compute Čech persistence diagrams of points in $\mathbb{R}^2$. On the other hand, already in three dimensions there exists a set of points S having different Alpha and Čech persistence diagrams in homological dimensions two. Moreover, for points in $(\mathbb{R}^2, \ell_\infty)$ Alpha and Čech filtrations are sequences of flag complexes. In particular a simplex σ belongs to $K_r^{\check{C}}$ if and only if $\max_{p,q\in\sigma} d_\infty(p,q) \leq 2r$. The new family of complexes defined here has the same properties.

Definition 4.3.1. The *Alpha flag complex* of S with radius r is

$$K_r^{AF} = \big\{\sigma \subseteq S \mid \max_{p,q\in\sigma} d_\infty(p,q) \leq 2r \text{ and } \{p,q\} \in K^D \text{ for each } \ p,q \in \sigma\big\}.$$

In this section, we prove that Alpha flag and Čech persistence diagrams coincide in homological dimensions zero and one. In particular, we think of Čech filtrations as a sequence of complexes where a single edge is added when going from $K_{r_i}^{\check{C}}$ to $K_{r_{i+1}}^{\check{C}}$. It is proven that at each such step the zero and one-dimensional homology groups of Alpha flag and Čech complexes remain isomorphic. To deal with the problem of multiple edges having equal length, we assume that the ℓ_∞ distances between pairs of points of S are all distinct, i.e. S is in general position. In case this property does not hold, the finite set of points S can be infinitesimally perturbed to obtain it. Importantly, the Stability Theorem 2.3.10, guarantees that the persistence diagrams of the original and perturbed points are close in bottleneck distance.

From now on the field $\mathbb{F}$ is omitted when referring to the homology of complexes to simplify notation, and a pair of points $\{p,q\} \subseteq S$ is said to be a non-Delaunay edge if it does not belong to the ℓ_∞-Delaunay complex of S. We start by presenting supporting results used in the proofs of the main two theorems.

Proposition 4.3.2. *Let B_1 and B_2 be two boxes in $\mathbb{R}^d$. If $B_1 \cap B_2$ is non-empty, then the Euclidean projection $\pi_{B_1} : B_1 \to B_2$, defined by mapping each $x \in B_1$ to its closest points in Euclidean distance on B_2, is such that $\pi_{B_1}(B_1) \subseteq B_1 \cap B_2$.*

Proof. Let $B_1 = \prod_{i=1}^d [a_i^{B_1}, b_i^{B_1}]$ and $B_2 = \prod_{i=1}^d [a_i^{B_2}, b_i^{B_2}]$ such that $B_1 \cap B_2 \neq \emptyset$. Because Cartesian products and intersections of intervals commute, defined $[\bar{a}_i, \bar{b}_i] = [a_i^{B_1}, b_i^{B_1}] \cap [a_i^{B_2}, b_i^{B_2}]$, we have that $[\bar{a}_i, \bar{b}_i] \neq \emptyset$ for each $1 \leq i \leq d$, and $B_1 \cap B_2 = \prod_{i=1}^d [\bar{a}_i, \bar{b}_i]$.

Given $x \in B_1$, we suppose by contradiction that $y = \pi_{B_1}(x) \in B_2$ is such that $y \notin B_1 \cap B_2$. Thus $y \notin \prod_{i=1}^d [\bar{a}_i, \bar{b}_i]$, and there exists $1 \leq \hat{i} \leq d$ such that $y_{\hat{i}} \notin [\bar{a}_{\hat{i}}, \bar{b}_{\hat{i}}]$.

The intervals $[a_{\hat{i}}^{B_1}, b_{\hat{i}}^{B_1}]$ and $[a_{\hat{i}}^{B_2}, b_{\hat{i}}^{B_2}]$ can intersect in four possible ways:

(i) $[a_{\hat{i}}^{B_1}, b_{\hat{i}}^{B_1}]$ intersects $[a_{\hat{i}}^{B_2}, b_{\hat{i}}^{B_2}]$ on the left, i.e. $a_{\hat{i}}^{B_1} \leq a_{\hat{i}}^{B_2} \leq b_{\hat{i}}^{B_1} \leq b_{\hat{i}}^{B_2}$. Thus $a_{\hat{i}}^{B_1} \leq x_{\hat{i}} \leq b_{\hat{i}}^{B_1} < y_{\hat{i}}$, and we define $y' = [y_1, \ldots, b_{\hat{i}}^{B_1}, \ldots, y_d]$;

(ii) $[a_{\hat{i}}^{B_1}, b_{\hat{i}}^{B_1}]$ intersects $[a_{\hat{i}}^{B_2}, b_{\hat{i}}^{B_2}]$ on the right, i.e. $a_{\hat{i}}^{B_2} \leq a_{\hat{i}}^{B_1} \leq b_{\hat{i}}^{B_2} \leq b_{\hat{i}}^{B_1}$. Thus $y_{\hat{i}} < a_{\hat{i}}^{B_1} \leq x_{\hat{i}} \leq b_{\hat{i}}^{B_1}$, and we define $y'' = [y_1, \ldots, a_{\hat{i}}^{B_1}, \ldots, y_d]$;

(iii) $[a_{\hat{i}}^{B_1}, b_{\hat{i}}^{B_1}]$ is contained in $[a_{\hat{i}}^{B_2}, b_{\hat{i}}^{B_2}]$, i.e. $a_{\hat{i}}^{B_2} \leq a_{\hat{i}}^{B_1} \leq b_{\hat{i}}^{B_1} \leq b_{\hat{i}}^{B_2}$. Thus $a_{\hat{i}}^{B_1} \leq x_{\hat{i}} \leq b_{\hat{i}}^{B_1} < y_{\hat{i}}$ or $y_{\hat{i}} < a_{\hat{i}}^{B_1} \leq x_{\hat{i}} \leq b_{\hat{i}}^{B_1}$, and in the first case we define $y' = [y_1, \ldots, b_{\hat{i}}^{B_1}, \ldots, y_d]$ and in the second $y'' = [y_1, \ldots, a_{\hat{i}}^{B_1}, \ldots, y_d]$;

(iv) $[a_{\hat{i}}^{B_1}, b_{\hat{i}}^{B_1}]$ contains $[a_{\hat{i}}^{B_2}, b_{\hat{i}}^{B_2}]$, i.e. $a_{\hat{i}}^{B_1} \leq a_{\hat{i}}^{B_2} \leq b_{\hat{i}}^{B_2} \leq b_{\hat{i}}^{B_1}$.

In case *(iv)* we have a contradiction as

$$y_{\hat{i}} \in [a_{\hat{i}}^{B_2}, b_{\hat{i}}^{B_2}] = [\bar{a}_{\hat{i}}, \bar{b}_{\hat{i}}] \not\ni y_{\hat{i}}.$$

In the other three cases, taken either y' or y'' we have

$$d_2(x, y') = \sqrt{(x_{\hat{i}} - b_{\hat{i}}^{B_1})^2 + \sum_{i=1, i\neq\hat{i}}^{d} (x_i - y_i)^2} < \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2} = d_2(x, y), \tag{4.6}$$

$$d_2(x, y'') = \sqrt{(x_{\hat{i}} - a_{\hat{i}}^{B_1})^2 + \sum_{i=1, i\neq\hat{i}}^{d} (x_i - y_i)^2} < \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2} = d_2(x, y). \tag{4.7}$$

because $(x_{\hat{i}} - b_{\hat{i}}^{B_1})^2 < (x_{\hat{i}} - y_{\hat{i}})^2$ in Equation (4.6), and $(x_{\hat{i}} - a_{\hat{i}}^{B_1})^2 < (x_{\hat{i}} - y_{\hat{i}})^2$ in Equation (4.7). The proof follows because this contradicts y being the closest point in Euclidean distance to x in B_2. □

Proposition 4.3.3. *Let S be a finite set of points in $(\mathbb{R}^d, d_\infty)$. Given $e = \{p, q\} \subseteq S$, we have that Nrv$(\{\overline{B_{\bar{r}}(y)}\}_{y\in\bar{\mathcal{Y}}})$ has the homotopy type of $A_e^{\bar{r}}$, where $\bar{r} = \frac{d_\infty(p,q)}{2}$ and $\bar{\mathcal{Y}} = \{y \in S \mid d_\infty(y, p) < 2\bar{r}$ and $d_\infty(y, q) < 2\bar{r}\}$, and so is contractible.*

Proof. From the Nerve Theorem 2.4.3 it follows that $\mathrm{Nrv}(\{\overline{B_{\bar{r}}(y)}\}_{y\in\bar{\mathcal{Y}}})$ and $\bigcup_{y\in\bar{\mathcal{Y}}} \overline{B_{\bar{r}}(y)}$ are homotopy equivalent, because ℓ_∞-balls are convex so that their intersections are either empty or contractible. Next, we show how to define a deformation retraction

$$\phi : \left(\bigcup_{y\in\bar{\mathcal{Y}}} \overline{B_{\bar{r}}(y)} \right) \times [0, 1] \to A_e^{\bar{r}}. \tag{4.8}$$

Given ϕ, we have that the set $\bigcup_{y \in \bar{\mathcal{Y}}} \overline{B_{\bar{r}}(y)}$ has the homotopy type of $A_e^{\bar{r}}$, which is contractible by its convexity. To obtain ϕ, we first define $\phi_y : \overline{B_{\bar{r}}(y)} \times [0,1] \to A_e^{\bar{r}}$ for each $y \in \bar{\mathcal{Y}}$. Given the Euclidean projection $\boldsymbol{\pi}_{\overline{B_{\bar{r}}(y)}} : \overline{B_{\bar{r}}(y)} \to A_e^{\bar{r}}$, we set

$$\phi_y(x,t) = (1-t) \cdot x + t \cdot \boldsymbol{\pi}_{\overline{B_{\bar{r}}(y)}}(x), \tag{4.9}$$

for every $x \in \overline{B_{\bar{r}}(y)}$ and $t \in [0,1]$. From Proposition 4.3.2 we have $\boldsymbol{\pi}_{\overline{B_{\bar{r}}(y))}}(x) \in \overline{B_{\bar{r}}(y)} \cap A_e^{\bar{r}}$, so that the straight line segment from x to $\boldsymbol{\pi}_{B_{\bar{r}}(y)}(x)$ is fully contained in $B_{\bar{r}}(y)$, by the convexity of this set. Thus ϕ_y is well-defined and continuous by the continuity of $\boldsymbol{\pi}_{B_{\bar{r}}(y)}$. Then we set

$$\phi(x,t) = \phi_{\hat{y}}(x,t), \tag{4.10}$$

for every $x \in \bigcup_{y \in \bar{\mathcal{Y}}} B_{\bar{r}}(y)$ and $t \in [0,1]$, with $\hat{y} \in \bar{\mathcal{Y}}$ such that $x \in B_{\bar{r}}(\hat{y})$. This might not be well-defined, because for a given x all the $\phi_{\hat{y}}$ corresponding to a point in $\bar{\mathcal{Y}}^x = \{\hat{y} \in \bar{\mathcal{Y}} \mid x \in B_{\bar{r}}(\hat{y})\}$ can be used to define $\phi(x,t)$ for any $t \in [0,1]$. Luckily, given $R = \bigcap_{\hat{y} \in \bar{\mathcal{Y}}^x} B_{\bar{r}}(\hat{y})$, which is a box containing x, Proposition 4.3.2 guarantees that $\boldsymbol{\pi}_R : R \to A_e^{\bar{r}}$ is such that $\boldsymbol{\pi}_R(R) \subseteq R \cap A_e^{\bar{r}}$. Thus ϕ is well-defined because the straight line segment defined by $(1-t) \cdot x + t \cdot \boldsymbol{\pi}_R(x)$ for $t \in [0,1]$ is contained within R, again by convexity. Furthermore, ϕ is continuous by the continuity of the Euclidean projections, and is a deformation retraction onto $A_e^{\bar{r}}$ because $A_e^{\bar{r}} \subseteq \bigcup_{y \in \bar{\mathcal{Y}}} B_{\bar{r}}(y)$. □

Proposition 4.3.4. *Let K_1 and K_2 be two abstract simplicial complexes such that $K_1 \subseteq K_2$. If there is only one edge e contained in K_2 and not in K_1, and it exists a triangle $\tau \in K_2$ of which e is a face, then $H_1(K_2)$ cannot contain an homology class $[\gamma]$ not in $H_1(K_1)$.*

Proof. Any 1-cycle representing an homology class $[\gamma]$ such that $[\gamma] \in H_1(K_2)$ and $[\gamma] \notin H_1(K_1)$ must contain e. But given $e = \{p,q\}$ and $\tau = \{p,q,y\}$, any such 1-cycle would be homologous to a formal sum containing $\{p,y\}$ and $\{y,q\}$ in place of e. Thus it would exist a 1-cycle representing $[\gamma]$ containing edges in K_1 only, which is in contradiction with $[\gamma] \notin H_1(K_1)$. □

The following result is presented as given in [BS21].

Theorem 4.3.5. *Let S be a finite set of points in $(\mathbb{R}^d, d_\infty)$ in general position, and $K_r^{\check{C}}$ the Čech complex of S with radius $r > 0$. If $e = \{p,q\} \subseteq S$ is a non-Delaunay edge contained in $K_r^{\check{C}}$, then $H_k(K_r^{\check{C}} \setminus St(e))$ and $H_k(K_r^{\check{C}})$ are isomorphic in homological dimensions zero and one.*

Proof. We can apply the reduced Mayer-Vietoris sequence, as given in [Spa12, Section 4.6], with $A = \mathrm{Cl}(\mathrm{St}(e)) \subseteq K_r^{\check{C}}$ and $B = K_r^{\check{C}} \setminus \mathrm{St}(e)$, because $A \cap B = \mathrm{Cl}(\mathrm{St}(e)) \setminus \mathrm{St}(e) \neq \emptyset$. It follows

$$\cdots \tilde{H}_k(A \cap B) \to \tilde{H}_k(A) \oplus \tilde{H}_k(B) \to \tilde{H}_k(A \cup B) \to \tilde{H}_{k-1}(A \cap B) \cdots$$
$$\Downarrow$$
$$\cdots \tilde{H}_k(\mathrm{Cl}(\mathrm{St}(e)) \setminus \mathrm{St}(e)) \to \tilde{H}_k(K_r^{\check{C}} \setminus \mathrm{St}(e)) \to \tilde{H}_k(K_r^{\check{C}}) \to \tilde{H}_{k-1}(\mathrm{Cl}(\mathrm{St}(e)) \setminus \mathrm{St}(e)) \cdots$$

where $\tilde{H}_k(A)$ cancels out, because it is trivial by definition of $A = \mathrm{Cl}(\mathrm{St}(e))$. Thus showing that $\tilde{H}_k(\mathrm{Cl}(\mathrm{St}(e)) \setminus \mathrm{St}(e))$ is trivial in homological dimensions k and $k-1$, implies that $\tilde{H}_k(K_r^{\check{C}} \setminus \mathrm{St}(e)) \to \tilde{H}_k(K_r^{\check{C}})$ is an isomorphism, from the exactness of the Mayer-Vietoris sequence above.

By definition of nerve, and Proposition 2.1.2 *(ii)*, it follows that $A = \mathrm{Cl}(\mathrm{St}(e)) = \mathrm{Nrv}\big(\{\overline{B_r(y)}\}_{y \in \mathcal{Y}}\big) \subseteq K_r^{\check{C}}$, where

$$\mathcal{Y} = \{y \in S |\ d_\infty(y,p) \leq 2r \text{ and } d_\infty(y,q) \leq 2r\}. \tag{4.11}$$

Defined $A_e^{\bar{r}} = \partial\overline{B_{\bar{r}}(p)} \cap \partial\overline{B_{\bar{r}}(q)}$, where $\bar{r} = \frac{d_\infty(p,q)}{2} \leq r$, we have that $A_e^{\bar{r}}$ is covered by $\bigcup_{y \in S \setminus e} B_{\bar{r}}(y)$ by Proposition 4.1.3. We can restrict this union of open balls to those centered in the points of

$$\bar{\mathcal{Y}} = \{y \in S |\ d_\infty(y,p) < 2\bar{r} \text{ and } d_\infty(y,q) < 2\bar{r}\} \subseteq \mathcal{Y}, \tag{4.12}$$

because $B_{\bar{r}}(y) \cap A_e^{\bar{r}} = \emptyset$ if $y \notin \bar{\mathcal{Y}}$. So $A_e^{\bar{r}}$ must be covered by $\bigcup_{y \in \bar{\mathcal{Y}}} B_{\bar{r}}(y)$ and

$$\mathrm{Nrv}\big(\{\overline{B_{\bar{r}}(y)}\}_{y \in \bar{\mathcal{Y}}}\big) \subseteq \mathrm{Nrv}\big(\{\overline{B_r(y)}\}_{y \in \mathcal{Y}}\big) \setminus \mathrm{St}(e) = \mathrm{Cl}(\mathrm{St}(e)) \setminus \mathrm{St}(e) \subseteq K_r^{\check{C}}.$$

By Proposition 4.3.3, the nerve $\mathrm{Nrv}\big(\{\overline{B_{\bar{r}}(y)}\}_{y \in \bar{\mathcal{Y}}}\big)$ has the homotopy type of $A_e^{\bar{r}}$, and so trivial homology. Then, given the simplices in $\mathrm{Cl}(\mathrm{St}(e)) \setminus \mathrm{St}(e)$ and not in $\mathrm{Nrv}\big(\{\overline{B_{\bar{r}}(y)}\}_{y \in \bar{\mathcal{Y}}}\big)$, we prove that adding them into $\mathrm{Nrv}\big(\{\overline{B_{\bar{r}}(y)}\}_{y \in \bar{\mathcal{Y}}}\big)$ does not alter its zero and one-dimensional homology.

Regarding zero-dimensional homology we know that $\mathrm{Nrv}\big(\{\overline{B_{\bar{r}}(y)}\}_{y \in \bar{\mathcal{Y}}}\big)$ consists of one connected component. Also, the vertices in $\mathrm{Cl}(\mathrm{St}(e)) \setminus \mathrm{St}(e)$ not in $\mathrm{Nrv}\big(\{\overline{B_{\bar{r}}(y)}\}_{y \in \bar{\mathcal{Y}}}\big)$, that could potentially create a homology class in $\tilde{H}_0(\mathrm{Cl}(\mathrm{St}(e)) \setminus \mathrm{St}(e))$, are the points in $\mathcal{Y} \setminus \bar{\mathcal{Y}}$. We have that $p, q \in \mathcal{Y} \setminus \bar{\mathcal{Y}}$, and these are fully connected to the points in $\bar{\mathcal{Y}}$, so do not create any connected component. Moreover, all other points in $\mathcal{Y} \setminus \bar{\mathcal{Y}}$ are connected to both p and q by definition of $\mathcal{Y}$. So $\mathrm{Cl}(\mathrm{St}(e)) \setminus \mathrm{St}(e)$ cannot contain a connected component not in $\mathrm{Nrv}\big(\{\overline{B_{\bar{r}}(y)}\}_{y \in \bar{\mathcal{Y}}}\big)$, and $\tilde{H}_0(\mathrm{Cl}(\mathrm{St}(e)) \setminus \mathrm{St}(e))$ must be trivial.

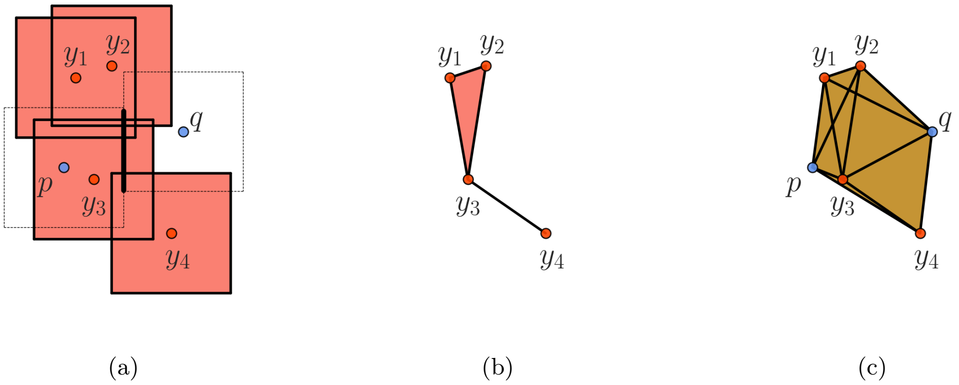


Figure 4.7: **(a)** Balls centered in the points of $\bar{\mathcal{Y}} = \{y_1, y_2, y_3, y_4\}$ covering $A_e^{\bar{r}}$. **(b)** $K_0 = \mathrm{Nrv}\big(\{\overline{B_{\bar{r}}(y)}\}_{y\in\bar{\mathcal{Y}}}\big)$. **(c)** K_1, i.e. the union of the cones from K_0 to p and q.

For one-dimensional homology, we define

$$K_0 = \mathrm{Nrv}\big(\{\overline{B_{\bar{r}}(y)}\}_{y\in\bar{\mathcal{Y}}}\big) \text{ and } K_n = \mathrm{Nrv}\big(\{\overline{B_r(y)}\}_{y\in\mathcal{Y}}\big) \setminus \mathrm{St}(e), \tag{4.13}$$

and show the existence of a filtration $K_0 \subseteq K_1 \subseteq \ldots \subseteq K_n$, such that at each step $K_i \subseteq K_{i+1}$ no one-dimensional homology class is created. We start by defining K_1 as the union of the cones from K_0 to p and q. Subfgures 4.7b and 4.7c illustrate this step for the points in Figure 4.7a. So going from K_0 to K_1 one-dimensional homology remains trivial because adding these cones cannot create any new 1-cycle. Then we add the points of $y' \in \mathcal{Y} \setminus (\bar{\mathcal{Y}} \cup \{p, q\})$ into K_1 one by one, obtaining a new complex K_{i+1} of the filtration above each time. Furthermore, at each such step $K_i \subseteq K_{i+1}$, we also add two triangles $\{p, y', \bar{y}\}$ and $\{q, y', \bar{y}\}$, where $\bar{y} \in \bar{\mathcal{Y}}$. This can be done because $A_e^{\bar{r}}$ is covered by $\bigcup_{y\in\bar{\mathcal{Y}}} B_{\bar{r}}(y)$, so that there exist $\bar{y} \in \bar{\mathcal{Y}}$ such that $\overline{B_r(y')} \cap \overline{B_r(\bar{y})} \supseteq \overline{B_r(y')} \cap \overline{B_{\bar{r}}(\bar{y})} \neq \emptyset$, because $\overline{B_r(y')} \cap A_e^{\bar{r}} \neq \emptyset$. Hence both $\overline{B_r(p)} \cap \overline{B_r(y')} \cap \overline{B_r(\bar{y})}$ and $\overline{B_r(q)} \cap \overline{B_r(y')} \cap \overline{B_r(\bar{y})}$ must be non-empty, by Proposition 2.1.2 *(ii)*, so that $\{p, y', \bar{y}\}, \{q, y', \bar{y}\} \in K_r^{\check{C}}$. Thus by Proposition 4.3.4 no one-dimensional homology class is created going from K_i to $K_{i+1} = K_i \cup \{y'\} \cup \{p, y'\} \cup \{q, y'\} \cup \{p, y', \bar{y}\} \cup \{q, y', \bar{y}\}$. We denote the K_{i+1} having $\mathcal{Y}$ as its set of vertices by K_{n-1}. Finally, we add all the simplices in $K_n \setminus K_{n-1}$ in the last filtration step. Again we can apply Proposition 4.3.4, because for each edge $\{y', y''\}$, with $y', y'' \in \mathcal{Y}$ added into K_n, there must be a triangle $\{p, y', y''\} \in K_n$ by definition of $\mathcal{Y}$. Hence we can conclude that K_n has trivial reduced one-dimensional homology, i.e. $\tilde{H}_1(\mathrm{Cl}(\mathrm{St}(e)) \setminus \mathrm{St}(e))$ is trivial.

The proof follows from the exactness of the reduced Mayer-Vietoris sequence as mentioned above, and the fact that isomorphisms in reduced homology translate into isomorphisms in non-reduced homology. □

The following result is one of the main contributions of this chapter. It is also discussed in Section 4 of the preprint [BS21].

Theorem 4.3.6. *Let S be a finite set of points in $(\mathbb{R}^d, d_\infty)$ in general position. Given $r > 0$ and $\varepsilon > 0$ such that $K^{\check{C}}_{r+\varepsilon}$ contains exactly one edge not in $K^{\check{C}}_r$, if $i^k_r : H_k(K^{AF}_r) \to H_k(K^{\check{C}}_r)$ is an isomorphism, then $i^k_{r+\varepsilon} : H_k(K^{AF}_{r+\varepsilon}) \to H_k(K^{\check{C}}_{r+\varepsilon})$ is also an isomorphism for $k = 0, 1$.*

Proof. Let $e = \{p, q\} \subseteq S$ be the only edge added to $K^{\check{C}}_r$ by increasing the radius parameter of $\varepsilon > 0$. Then either e is a ℓ_∞-Delaunay edge, so that $e \in K^{AF}_r$ and $e \in K^{\check{C}}_r$, or e is non-Delaunay edge, so that $e \notin K^{AF}_r$ and $e \in K^{\check{C}}_r$. We split the proof in two parts, dealing with these two cases separately.

We use the notation of Proposition 4.1.3, meaning that $r < \bar{r} = \frac{d_\infty(p,q)}{2} \leq r + \varepsilon$ and $A^{\bar{r}}_e = \overline{B_{\bar{r}}(p)} \cap \overline{B_{\bar{r}}(q)}$. Also, as in the proof of Theorem 4.3.5, we define $\bar{\mathcal{Y}} = \{y \in S \mid d_\infty(y, p) < 2\bar{r}$ and $d_\infty(y, q) < 2\bar{r}\}$, so that if e is a non-Delaunay edge, then $A^{\bar{r}}_e$ must be covered by $\bigcup_{y \in \bar{\mathcal{Y}}} B_{\bar{r}}(y)$.

CASE 1: e is ℓ_∞-Delaunay
For $r > 0$ the complexes K^{AF}_r and $K^{\check{C}}_r$ contain the same vertices by definition. Also, because the homomorphism induced by the inclusion of complexes $H_0(K^{AF}_r) \to H_0(K^{\check{C}}_r)$ is an isomorphism, K^{AF}_r and $K^{\check{C}}_r$ have the same connected components. Thus after e is added in both K^{AF}_r and $K^{\check{C}}_r$ either connected components do not change or the same connected component is merged in both. In the first case zero-dimensional homology remains unchanged, while in the second case the same zero-dimensional homology class is deleted in $H_0(K^{AF}_r)$ and $H_0(K^{\check{C}}_r)$. In both cases $i^0_r : H_0(K^{AF}_{r+\varepsilon}) \to H_0(K^{\check{C}}_{r+\varepsilon})$ is an isomorphism induced by the inclusion $K^{AF}_{r+\varepsilon} \subseteq K^{\check{C}}_{r+\varepsilon}$.

We now look at one-dimensional homology. Adding a single edge e and the cliques it forms into K^{AF}_r and $K^{\check{C}}_r$ can result in the creation or deletion of one-dimensional homology classes. We further split this case into two subcases.

1. The edge e adds nothing but itself to the Alpha flag complex K^{AF}_r.

2. The edge e adds itself and one or more triangles to the Alpha flag complex K^{AF}_r.

SUBCASE 1.1
We start by proving that the edge e is the only simplex added into $K^{\check{C}}_r$ as well. To show this, suppose by contradiction that increasing the radius parameter from r to $r+\varepsilon$ results into adding e and a triangle $\{p, q, y\}$ into $K^{\check{C}}_r$. This means $\{p, y\}, \{q, y\} \in K^{\check{C}}_r$, so that they are strictly shorter than $\{p, q\}$ from our hypothesis on distances between pairs of

points in S, i.e. general position. Given $0 < 2\delta < 2\bar{r} - \max\{d_\infty(p,y), d_\infty(q,y)\}$, we have $d_\infty(p,y) < 2\bar{r} - 2\delta$ and $d_\infty(q,y) < 2\bar{r} - 2\delta$. Moreover $d_\infty(p,q) = 2\bar{r}$, so the three axis-parallel hypercubes $\overline{B_{\bar{r}}(p)}$, $\overline{B_{\bar{r}}(q)}$, and $\overline{B_{\bar{r}-\delta}(y)}$ have non-empty pairwise intersections. Their triple intersection is also non-empty, by Proposition 2.1.2 *(ii)*, and it follows that $A_e^{\bar{r}} \cap \overline{B_{\bar{r}}(y)} = \overline{B_{\bar{r}}(p)} \cap \overline{B_{\bar{r}}(q)} \cap \overline{B_{\bar{r}}(y)} \neq \emptyset$. Hence the set of points $\mathcal{Y}$ contains at least one point, and because e is a ℓ_∞-Delaunay edge, we have that $A_e^{\bar{r}} \setminus \big(\bigcup_{y\in\bar{\mathcal{Y}}} B_{\bar{r}}(y)\big)$ is non-empty. Thus the closed set $\big(\bigcup_{y\in\bar{\mathcal{Y}}} B_{\bar{r}}(y)\big)^c$ needs to intersect $A_e^{\bar{r}}$, which is a closed box. So there exist a point z of $A_e^{\bar{r}}$ belonging to the boundary of the closure of $\bigcup_{y\in\bar{\mathcal{Y}}} B_{\bar{r}}(y)$, otherwise $A_e^{\bar{r}}$ would need to be disconnected, i.e. $A_e^{\bar{r}} \cap \partial\overline{\big(\bigcup_{y\in\mathcal{Y}} B_{\bar{r}}(y)\big)} \neq \emptyset$. Furthermore, $z \in A_e^{\bar{r}} \cap \big(\bigcup_{y\in\mathcal{Y}} \partial\overline{B_{\bar{r}}(y)}\big)$, because $\partial\overline{\big(\bigcup_{y\in\mathcal{Y}} B_{\bar{r}}(y)\big)} \subseteq \big(\bigcup_{y\in\mathcal{Y}} \partial\overline{B_{\bar{r}}(y)}\big)$. In conclusion there exist $z \in A_e^{\bar{r}} \cap \partial\overline{B_{\bar{r}}(y')}$ for some $y' \in \mathcal{Y}$, so $\{p,q,y'\}$ is a ℓ_∞-Delaunay triangle with z as a witness point, which belongs to $K_{r+\varepsilon}^{AF}$. This contradicts the hypothesis of Subcase 1.1, because the Alpha flag complex $K_{r+\varepsilon}^{AF}$ cannot contain any triangles of which $\{p,q\}$ is an edge. Thus, when increasing the radius parameter from r to $r+\varepsilon$, the edge $e = \{p,q\}$ is the only simplex added in both K_r^{AF} and $K_r^{\check{C}}$.

In general, adding a single edge to an abstract simplicial complex can result in either the deletion of a connected component or the creation of a one-dimensional homology class. The former of these two cases is dealt with the discussion of zero-dimensional homology above and does not affect one-dimensional homology. On the other hand, if e does not merge connected components in K_r^{AF}, then it also does not merge connected components in $K_r^{\check{C}}$, because as already discussed zero-dimensional homology remains isomorphic. Thus both $H_1(K_{r+\varepsilon}^{AF})$ and $H_1(K_{r+\varepsilon}^{\check{C}})$ contain a new homology class. In this case $i_{r+\varepsilon}^1 : H_1(K_{r+\varepsilon}^{AF}) \to H_1(K_{r+\varepsilon}^{\check{C}})$ is the isomorphism induced by the inclusion, which extends $i_r^1 : H_1(K_r^{AF}) \to H_1(K_r^{\check{C}})$ by mapping the one-dimensional homology class created by e in $K_{r+\varepsilon}^{AF}$ into the one created by e in $K_{r+\varepsilon}^{\check{C}}$.

Subcase 1.2

Adding $e = \{p,q\}$ to both K_r^{AF} and $K_r^{\check{C}}$ results in one or more triangles $\{\tau_j^{\bar{r}}\}_{j\in J}$ added to the Alpha flag complex $K_{r+\varepsilon}^{AF}$. Moreover, by the definition of flag complex, the same triangles are added to $K_{r+\varepsilon}^{\check{C}}$. Also, there might be triangles $\{\check{\tau}_j^{\bar{r}}\}_{j\in\check{J}}$ added to $K_{r+\varepsilon}^{\check{C}}$, which are not added to $K_{r+\varepsilon}^{AF}$. These $\{\check{\tau}_j^{\bar{r}}\}_{j\in\check{J}}$ contain $\{p,q\}$ as an edge, and at least one non-Delaunay edge among their other edges.

To begin with, we note that e does not create any one-dimensional homology class in $K_{r+\varepsilon}^{AF}$ and $K_{r+\varepsilon}^{\check{C}}$ by Proposition 4.3.4. It remains to prove that a one-dimensional homology class $[\gamma] \in H_1(K_r^{AF})$ is deleted at radius $r+\varepsilon$ if and only if $i_r^1([\gamma]) = [\check{\gamma}] \in H_1(K_r^{\check{C}})$ is also deleted.

The first direction holds because if a homology class is deleted in the Alpha flag complex, then the same formal sum of triangles is a boundary for the same homology class of the Čech complex.

For the opposite direction, let us suppose that $[\check{\gamma}] \in H_1(K^{\check{C}}_r)$ is deleted at radius $r+\varepsilon$, and that $[\gamma]$ remains open in the Alpha flag complex with radius $r+\varepsilon$. We can think of adding the triangles $\{\tau^{\bar{r}}_j\}_{j\in J}$ and $\{\check{\tau}^{\bar{r}}_j\}_{j\in \check{J}}$ one by one in $K^{\check{C}}_r$ in any order, obtaining a new $\check{K}_i \subseteq K^{\check{C}}_{r+\varepsilon}$ at each step. At some point, one of these must be creating a boundary deleting $[\check{\gamma}]$ in $\check{K}_i$. If this is a triangle $\tau^{\bar{r}}_j$ (containing Delaunay edges only), then its edges form a formal sum which is homologous to both $[\gamma]$ and $[\check{\gamma}]$. Moreover, $\tau^{\bar{r}}_j$ bounds this formal sums in both complexes, so that $[\gamma] \notin H_1(K^{AF}_{r+\varepsilon})$, which is a contradiction. On the other hand, if a non-Delaunay triangle $\check{\tau}^{\bar{r}}_j$ is creating a boundary deleting $[\check{\gamma}]$, we can apply Theorem 4.3.5 to one of the non-Delaunay edges $\check{e}$ of $\check{\tau}^{\bar{r}}_j$. We have a contradiction with the assumption of $\check{\tau}^{\bar{r}}_j$ deleting $[\check{\gamma}]$, because $K^{\check{C}}_{r+\varepsilon} \setminus \mathrm{St}(\check{e})$ and $K^{\check{C}}_{r+\varepsilon}$ need to have the same one-dimensional homology and $\check{\tau}^{\bar{r}}_j \in \mathrm{St}(\check{e})$.

In conclusion the same one-dimensional homology classes are deleted in both complexes by the same triangles, and so $i^1_{r+\varepsilon} : H_1(K^{AF}_{r+\varepsilon}) \to H_1(K^{\check{C}}_{r+\varepsilon})$ is an isomorphism induced by the inclusion $K^{AF}_{r+\varepsilon} \subseteq K^{\check{C}}_{r+\varepsilon}$.

CASE 2: e is non-Delaunay
By applying Theorem 4.3.5, we have that $H_k(K^{\check{C}}_{r+\varepsilon} \setminus \mathrm{St}(e)) \to H_k(K^{\check{C}}_{r+\varepsilon})$ is an isomorphism for $k = 0, 1$.

Finally, the diagram

$$\begin{tikzcd}
H_k(K^{AF}_r) \arrow[r, "\cong"] \arrow[d, "\cong"] & H_k(K^{\check{C}}_{r+\varepsilon} \setminus \mathrm{St}(e)) \arrow[d, "\cong"] \\
H_k(K^{AF}_{r+\varepsilon}) \arrow[r] & H_k(K^{\check{C}}_{r+\varepsilon})
\end{tikzcd} \tag{4.14}$$

obtained by applying the homology functor to the inclusion maps between complexes commutes, because $K^{\check{C}}_r = K^{\check{C}}_{r+\varepsilon} \setminus \mathrm{St}(e)$ and $K^{AF}_r = K^{AF}_{r+\varepsilon}$, proving that $H_k(K^{AF}_{r+\varepsilon}) \to H_k(K^{\check{C}}_{r+\varepsilon})$ is an isomorphism for $k = 0, 1$. □

Corollary 4.3.7. *Let S be a finite set of points in $(\mathbb{R}^d, d_\infty)$ in general position. Given a finite set of monotonically increasing real-values $\mathcal{R} = \{r_i\}_{i=1}^m$, the Alpha flag $K^{AF}_{\mathcal{R}}$ and Čech filtrations $K^{\check{C}}_{\mathcal{R}}$ of S have the same persistence diagrams in homological dimensions zero and one.*

Proof. Given the two parameterized filtrations $K^{AC}_{r_0} \subseteq K^{AC}_{r_1} \subseteq \ldots \subseteq K^{AC}_{r_m}$ and $K^{\check{C}}_{r_0} \subseteq K^{\check{C}}_{r_1} \subseteq \ldots \subseteq K^{\check{C}}_{r_m}$. We have that $H_k(K^{AF}_{r_i}) \to H_k(K^{\check{C}}_{r_i})$ is an isomorphism for each

$0 \leq i \leq m$ and $k = 0, 1$.

- For $r_i \leq 0$, $K^{AF}_{r_i}$ and $K^{\check{C}}_{r_i}$ are empty.
- For $r_i > 0$, we can think of $K^{AF}_{r_i}$ and $K^{\check{C}}_{r_i}$ as the result of adding one edge at a time, plus the cliques formed by edges, into K^{AC}_0 and $K^{\check{C}}_0$. Theorem 4.3.6 ensures that each new edge added preserves the isomorphism between the zero and one-dimensional homology groups of the Alpha flag and Čech complexes.

The proof follows by applying the Persistence Equivalence Theorem 2.4.11. □

The above result extends to a general ambient dimension d the equivalence of zero and one-dimensional persistence diagrams proven in [HKS15] for two and three-dimensional points.

4.4 Minibox Complexes

In this section, yet another family of complexes is introduced, which we prove to have the same property of Alpha flag complexes, i.e. they can be used to compute the Čech persistence diagrams of S in homological dimensions zero and one. We also discuss the expected number of edges these complexes contain. In the next section, we describe algorithms for finding these edges.

Definition 4.4.1. Let p, q be two points in $(\mathbb{R}^d, d_\infty)$. The *minibox* of p and q is

$$\text{Mini}_{pq} = \prod_{i=1}^{d} \big(\min\{p_i, q_i\}, \max\{p_i, q_i\}\big), \tag{4.15}$$

that is to say the interior of the minimal bounding box of p and q.

Proposition 4.4.2. *Let S be a finite set of points in $(\mathbb{R}^d, d_\infty)$, $e = \{p, q\}$ a pair of points of S, and Mini$_{pq}$ the minibox of p and q. If it exists $y \in S \setminus e$ such that $y \in$ Mini$_{pq}$, then e is not an edge of the ℓ_∞-Delaunay complex of S.*

Proof. Given $\bar{r} = \frac{d_\infty(p,q)}{2}$, we have $A^{\bar{r}}_e = \overline{B_{\bar{r}}(p)} \cap \overline{B_{\bar{r}}(q)}$ by Proposition 4.1.3. Equivalently $A^{\bar{r}}_e = \prod_{i=1}^{d}[b_i - \bar{r}, a_i + \bar{r}]$, where $a_i = \min\{p_i, q_i\}$ and $b_i = \max\{p_i, q_i\}$ for each $1 \leq i \leq d$. Then, given $y \in \text{Mini}_{pq}$, it follows that $a_i < y_i < b_i$ for each $1 \leq i \leq d$, implying $y_i - \bar{r} < b_i - \bar{r}$ and $a_i + \bar{r} < y_i + \bar{r}$. Thus $[b_i - \bar{r}, a_i + \bar{r}] \subset (y_i - \bar{r}, y_i + \bar{r})$ for each $1 \leq i \leq d$, and $A^{\bar{r}}_e \subset B_{\bar{r}}(y)$. The result follows applying Proposition 4.1.3. □

Definition 4.4.3. The *Minibox complex* of S with radius r is

$$K_r^M = \big\{\sigma \subseteq S \mid \max_{p,q\in\sigma} d_\infty(p,q) \leq 2r \text{ and } \mathrm{Mini}_{pq} \cap S = \emptyset \text{ for each } \ p,q \in \sigma\big\}.$$

The next theorem is another novel result, which is also presented in Section 5 of the preprint [BS21].

Theorem 4.4.4. *Let S be a finite set of points in $(\mathbb{R}^d, d_\infty)$ in general position. Given the Alpha flag K_r^{AF} and Minibox K_r^M complexes with radius r, then $H_k(K_r^{AF})$ and $H_k(K_r^M)$ are isomorphic in homological dimensions zero and one.*

Proof. We have $K_r^{AF} \subseteq K_r^M \subseteq K_r^{\check{C}}$, and we know that $H_k(K_r^{AF}) \to H_k(K_r^{\check{C}})$ is an isomorphism for $k = 0,1$ from the discussion in the proof of Corollary 4.3.7. Thus we have the following commutative diagrams, implying that $H_k(K_r^{AF}) \to H_k(K_r^M)$ is injective for $k = 0,1$ and any $r \in \mathbb{R}$.

$$\begin{tikzcd}
K_r^{AF} \arrow[rr, hook] \arrow[dr, hook] & & K_r^{\check{C}} & \Longrightarrow & H_k(K_r^{AF}) \arrow[rr, "\cong"] \arrow[dr, hook] & & H_k(K_r^{\check{C}}) \\
& K_r^M \arrow[ur, hook] & & & & H_k(K_r^M) \arrow[ur, two heads] &
\end{tikzcd} \tag{4.16}$$

To conclude our proof we need to show the surjectivity of this homomorphism for $k = 0,1$.

For $k = 0$, because K_r^{AF} and K_r^M have the same set of vertices, and K_r^M might contain more edges, it follows that K_r^{AF} has the same or more connected components than K_r^M. So in homological dimension zero the homomorphism induced by the inclusion $K_r^{AF} \subseteq K_r^M$, must be surjective.

To prove the surjetivity of $i_r^1 : H_1(K_r^{AF}) \to H_1(K_r^M)$, we show that for any $[\gamma] \in H_1(K_r^M)$ a 1-cycle γ representing it has to be homologous to a 1-cycle γ' containing only ℓ_∞-Delaunay edges of length less than or equal to $2r$, so that $i_r^1([\gamma']) = [\gamma]$.

Let γ be a 1-cycle in K_r^M representing $[\gamma] \in H_1(K_r^M)$, and $e = \{p,q\}$ the non-Delaunay edge in γ of maximum length. We have $A_e^{\bar{r}} = \overline{B_{\bar{r}}(p)} \cap \overline{B_{\bar{r}}(q)}$, where $\bar{r} = \frac{d_\infty(p,q)}{2}$ by Proposition 4.1.3. Defined $\bar{\mathcal{Y}} = \{y \in S \mid d_\infty(y,p) < 2\bar{r} \text{ and } d_\infty(y,q) < 2\bar{r}\}$, we equivalently have $\bar{\mathcal{Y}} = S \cap B_{2\bar{r}}(p) \cap B_{2\bar{r}}(q) = S \cap \bar{r}(A_e^{\bar{r}})$, because $\varepsilon(A_e^{\bar{r}}) = \varepsilon\big(\overline{B_{\bar{r}}(p)} \cap \overline{B_{\bar{r}}(q)}\big) = \overline{B_{\bar{r}+\varepsilon}(p)} \cap \overline{B_{\bar{r}+\varepsilon}(q)}$ by Proposition 4.1.1 *(ii)*. For points in $\mathbb{R}^2$, these sets are illustrated in Figure 4.8, where $A_e^{\bar{r}}$ is represented by a thickened vertical line between p and q. Moreover, given $c = \frac{p+q}{2}$, we have $\mathrm{Mini}_{pq} \subseteq \bar{r}(c) \subseteq \bar{r}(A_e^{\bar{r}})$, because $c \subseteq A_e^{\bar{r}}$, taking ε-thickenings preserves inclusions, and Mini_{pq} has sizes of length less than or equal to

$2\bar{r}$ and center c. Then, because e is not a Delaunay edge, $A_e^{\bar{r}}$ must be covered by the union of balls centered in the points of $S \setminus \{p, q\}$ by Proposition 4.1.3. Thus at least one $y \in S \setminus \{p, q\}$ is such that $B_{\bar{r}}(y)$ intersects $A_e^{\bar{r}}$, i.e. $\bar{\mathcal{Y}} \neq \emptyset$. Defined $\bar{y} \in \bar{\mathcal{Y}}$ to be the point realizing

$$\min_{y \in \bar{\mathcal{Y}}} d_\infty(y, \mathrm{Mini}_{pq}),$$

we have that $\mathrm{Mini}_{p\bar{y}}$ and $\mathrm{Mini}_{q\bar{y}}$ do not contain points in $S \setminus \{p, q, \bar{y}\}$, as we can show a contradiction otherwise. Suppose there exist either $y' \in S \setminus \bar{\mathcal{Y}}$ or $y'' \in \bar{\mathcal{Y}}$ belonging to one of these two miniboxes. Without loss of generality, we assume either $y' \subseteq \mathrm{Mini}_{p\bar{y}}$ or $y'' \subseteq \mathrm{Mini}_{p\bar{y}}$. In the former case we have $\mathrm{Mini}_{p\bar{y}} \subseteq \bar{r}(A_e^{\bar{r}})$, because p is on the boundary of $\bar{r}(A_e^{\bar{r}})$ and $\bar{y}$ in its interior. So $y' \in \bar{r}(A_e^{\bar{r}})$, implying that $y' \in \bar{\mathcal{Y}}$, which is a contradiction. In the latter case, it must be that $d_\infty(y'', \mathrm{Mini}_{pq}) < d_\infty(\bar{y}, \mathrm{Mini}_{pq})$ by definition of $\mathrm{Mini}_{p\bar{y}}$ and d_∞, which is in contradiction with $\bar{y}$ being the closest point of $\bar{\mathcal{Y}}$ to Mini_{pq}.

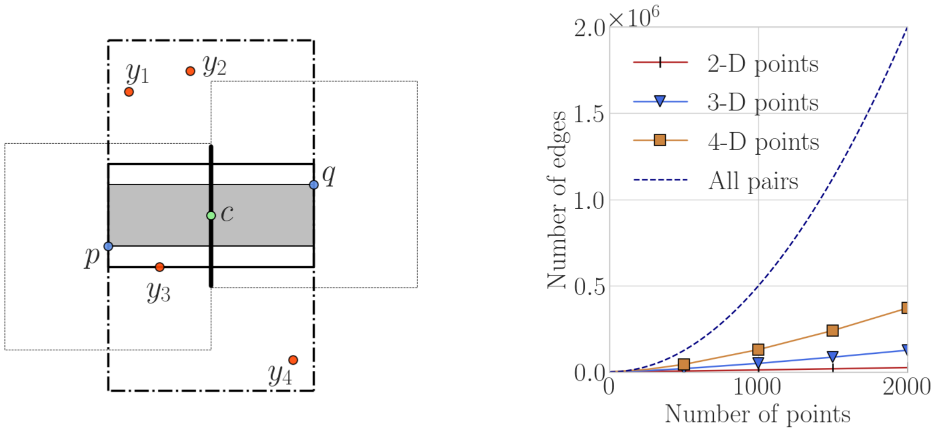
So there exists a vertex $\bar{y}$ of the Minibox complex connected to p and q by the edges $\{p, \bar{y}\}$ and $\{\bar{y}, q\}$. These are shorter than $2\bar{r}$ so that $\{p, \bar{y}\}, \{\bar{y}, q\} \subseteq K_r^M$. Swapping $\{p, \bar{y}\}$ and $\{\bar{y}, q\}$ for e in γ, we obtain a 1-cycle homologous to γ with the property of having a shorter longest non-Delaunay edge. This procedure can be repeated only a finite number of times, as we have a finite number of non-Delaunay edges, and at each iteration the maximum non-Delaunay edge length in the current 1-cycle decreases. When the procedure cannot be repeated, we have a 1-cycle γ' in K_r^M homologous to γ, containing only ℓ_∞-Delaunay edges. Hence γ' represents a one-dimensional homology class in the Alpha flag complex which is mapped into $[\gamma]$ by $i_r^1 : H_1(K_r^{AF}) \to H_1(K_r^M)$. □

Corollary 4.4.5. *Let S be a finite set of points in $(\mathbb{R}^d, d_\infty)$ in general position. Given a finite set of monotonically increasing real-values $\mathcal{R} = \{r_i\}_{i=1}^m$, the Alpha flag $K_{\mathcal{R}}^{AF}$ and Minibox filtrations $K_{\mathcal{R}}^M$ of S have the same persistence diagrams in homological dimensions zero and one.*

Proof. Follows from the Persistence Equivalence Theorem of 2.4.11 as for Corollary 4.3.7. □

Number of Minibox edges. We conclude this section by studying the number of edges that a Minibox complex K_r^M can contain. We are able to show that for randomly sampled points the expected number of empty miniboxes on the points of S is proportional to $n \cdot \mathrm{polylog}(n)$, where n is the number of points of S.

We start by noting that in the worst case a Minibox complex can contain $O(n^2)$



(a) (b)

Figure 4.8: **(a)** The pair (p, q) is not a Delaunay edge, but is a Minibox edge. Mini_{pq} is the gray region having p and q as two vertices. The set $\bar{\mathcal{Y}}$ consists of four y_i points contained in the rectangle $\bar{r}(A_e^{\bar{r}})$, whose boundary is represented by a dash-dot line. **(b)** Expected number of Minibox edges of randomly sampled points for $d = 2, 3, 4$, compared to the number of all possible edges (dashed line).

edges. For example the union of

$$S_1 = \left\{ p_i = \left(1 - \frac{i}{n}, 1 - \frac{i}{n}\right) \right\}_{i=1}^{n} \text{ and } S_2 = \left\{ q_j = \left(3 - \frac{j}{n}, 1 - \frac{j}{n}\right) \right\}_{j=1}^{n}, \tag{4.17}$$

is a set of $2n$ points in $\mathbb{R}^2$, on parallel line segments, such that all the miniboxes $\mathrm{Mini}_{p_i q_j}$ for $1 \leq i \leq j \leq n$ do not contain any point in $S_1 \cup S_2$. Thus the Minibox complex of $S_1 \cup S_2$ contains more than $\frac{n(n-1)}{2}$ points for a large enough radius parameter.

Next, given S to be a set of random points in $\mathbb{R}^d$, we can derive the expected number of edges contained in any maximal Minibox complex.

Definition 4.4.6. Let p and q be points in $\mathbb{R}^d$. We say that p *dominates* q if each of the coordinates of p is greater than the corresponding coordinate of q. Given a finite set of points $S \subseteq \mathbb{R}^d$, we say that p *directly dominates* q if p dominates q and there is no other point $y \in S$ such that p dominates y and y dominates q.

Proposition 4.4.7. *Let S be a finite set of uniformly distributed random points in the unit hypercube $[0, 1]^d \subseteq (\mathbb{R}^d, d_\infty)$. The expected number of edges contained in the maximal Minibox complex of S is $\Theta\left(\frac{2^{d-1}}{(d-1)!} n \log^{d-1}(n)\right)$, where n is the number of points of S.*

Proof. We have that if p directly dominates q, then $\mathrm{Mini}_{pq} \cap S = \emptyset$. On the other hand, $\mathrm{Mini}_{pq} \cap S = \emptyset$ does not imply that either p directly dominates q or q directly dominates

p. However, for each pair $\{p, q\}$ there exists a sequence of a maximum of d reflections about the coordinate hyperplanes that transforms S into a set of points such that q dominates p. There are 2^d possible such sequences of reflections, one for each orthant, and each produces a set of points S_k with a set of directly dominated pairs disjoint from those of the other S_ks. Moreover, if $\{p, q\}$ is not a directly dominated pair in any S_k for $1 \leq k \leq 2^d$, then $\mathrm{Mini}_{pq} \cap S$ must be non-empty. So if the expected number of directly dominated pairs in S_k is m, then the expected number of empty miniboxes on S is $2^{d-1} \cdot m$, because each edge $\{p, q\}$ is counted twice in the 2^d transformed point sets S_k. In [Kle86] it is shown that for n random points in a bounded region of $\mathbb{R}^d$ the expected number of directly dominated pairs is $\Theta\left(\frac{1}{(d-1)!} n \log^{d-1}(n)\right)$. Thus, in dimension d there are $\Theta\left(\frac{2^{d-1}}{(d-1)!} n \log^{d-1}(n)\right)$ pair of points $\{p, q\}$ such that $\mathrm{Mini}_{pq} \cap S = \emptyset$. The proof follows from the definition of Minibox complex. □

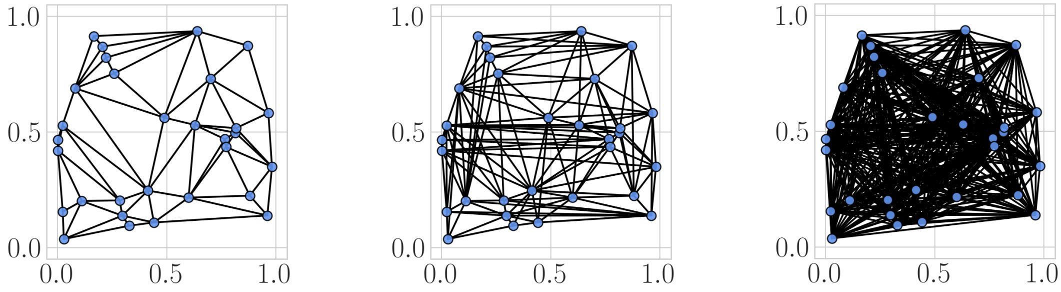
Figure 4.8b plots the expected number of minibox edges for random points in dimension $2 \leq d \leq 4$ with n in the range $[0, 2000]$. This is an empirical estimate obtained by randomly sampling points in the unit hypercube, and counting the number of edges found with the algorithms of the next section.

4.5 Algorithms

We present algorithms for finding all pairs of points $\{p, q\} \subseteq S$ such that $\mathrm{Mini}_{pq} \cap S$ is empty. By definition, these are all the edges a Minibox complex can contain. We study the two-dimensional, three-dimensional, and higher-dimensional cases separately. For $d = 2$ and $d = 3$, we present a plane-sweep and a space-sweep algorithm respectively. These maintain front data structures that can be used to efficiently determine whether $\mathrm{Mini}_{pq} \cap S$ is empty or not. For general dimension d, we see the problem of finding all empty miniboxes on S as of an offline orthogonal range emptiness problem with $\frac{n(n-1)}{2}$ range queries, and reference known results on range queries.

We also provide an implementation of these algorithms in the form of the `persty` Python package, the source code of which is available at github.com/gbeltramo/persty.

Points in two dimensions. We start by taking S to be a finite set of points in $(\mathbb{R}^2, d_\infty)$. For this case, we describe a $O(n^2)$ algorithm, whose pseudocode is given in Algorithm 4.1. This is worst-case optimal by the discussion on the number of Minibox edges at the end of Section 4.4. An example of the edges contained in the maximal Minibox complex of random points in the unit square $[0, 1] \times [0, 1] \subseteq \mathbb{R}^2$ is given in Figure 4.9b. This can be compared to the edges in Figures 4.9a and 4.9c showing the edges contained in the maximal Alpha flag and Čech complexes on the same points.



(a) Alpha flag edges.

(b) Minibox edges.

(c) Čech edges.

Figure 4.9: Comparison of Alpha flag (i.e. Delaunay), Minibox, and Čech edges of random points in $[0, 1] \times [0, 1] \subseteq \mathbb{R}^2$.

Algorithm 4.1 Minibox edges of a finite set of points S in $\mathbb{R}^2$.

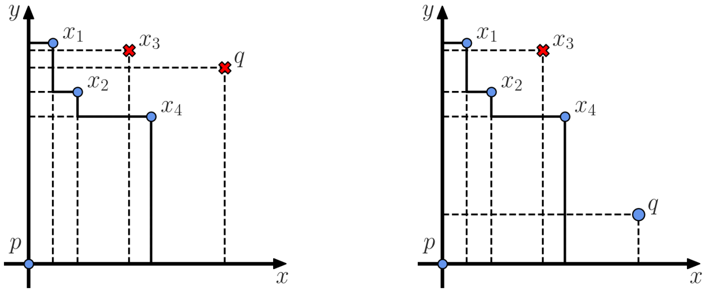
Input: array *points*, the finite set of points S in two dimensions.

```
 1: $edges \leftarrow$ empty list of two-tuples of integers
 2: Sort $points$ on their $x$-coordinate
 3: $front_\uparrow, front_\downarrow \leftarrow (p^0_x, +\infty), (p^0_x, -\infty)$, where $p^0 = points[0]$
 4: for $i = 0$ to $|S| - 1$ do
 5:     for $j = i + 1$ to $|S| - 1$ do
 6:         $p,\ q \leftarrow points[i],\ points[j]$
 7:         if Mini$_{pq}$ does not contain $front_\uparrow$ or $front_\downarrow$ then
 8:             Add $(i, j)$ to $edges$
 9:             Set $front_\uparrow = q$ if $p_y < q_y$, or $front_\downarrow = q$ if $p_y \geq q_y$
10:         end if
11:     end for
12: end for
13: return $edges$
```

The algorithm works by sweeping the plane form left to right for each point $p = (p_x, p_y)$, starting from p_x. In particular, it checks whether (p, q) is a Minibox edge for each point $q = (q_x, q_y)$ in the half plane $(p_x, +\infty) \times (-\infty, +\infty)$. This is done on line 7 of Algorithm 4.1. For this it uses a front, which consists of two points $front_\uparrow$ and $front_\downarrow$. These have the following properties:

- $front_\uparrow$ has a y-coordinate greater than p_y, while $front_\downarrow$ has a y-coordinate less than or equal to p_y;

- Defined X to be the set of points in S with x-coordinate in the range (p_x, q_x), $front_\uparrow$ has a y-coordinate smaller than any other point $x \in X$ such that $x_y > p_y$, and $front_\downarrow$ has a y-coordinate larger than any other point $x \in X$ such that $x_y \leq p_y$.



(a) (b)

Figure 4.10: Illustration of the plane-sweep algorithm for Minibox edges in $\mathbb{R}^2$, with $X = \{x_1, x_2, x_3, x_4\}$ and $front_\uparrow = x_4$. In **(a)** $\{p, q\}$ is not a Minibox edge, because $front_\uparrow \in \text{Mini}_{pq}$. In **(b)** Mini_{pq} is empty, so in this case $\{p, q\}$ is a Minibox edge.

These properties are true when $front_\uparrow$ and $front_\downarrow$ are defined on line 3 and are preserved by the update operation on line 9. Moreover, because these properties always hold, Mini_{pq} can be non-empty if and only if it contains either $front_\uparrow$ or $front_\downarrow$. For example, given p and q such that $q_y > p_y$, by definition Mini_{pq} is non-empty if and only if there exist of point $x \in X \subseteq S$ such that $x_y > p_y$ and q dominates x. But such a point exists only if q dominates $front_\uparrow$. Thus the check on line 7 determines if $\{p, q\}$ is a ℓ_∞-Delaunay edge. This is illustrated in Figure 4.10, where $X = \{x_1, x_2, x_3, x_4\}$ and $front_\uparrow = x_4$.

Proposition 4.5.1. *Let S be a set of finite points in $(\mathbb{R}^2, d_\infty)$. Algorithm 4.1 can be used to find the Minibox edges on S in $O(n^2)$ time.*

Proof. The correctness of Algorithm 4.1 is discussed above. It loops on all possible $\frac{n(n-1)}{2}$ edges, and at each iteration it needs $O(1)$ operations to check whether Mini_{pq} is empty and update $front_\uparrow$, $front_\downarrow$. Thus, it has complexity $O(n^2)$. □

Points in three dimensions. For a finite set of points S in three dimensions, we present Algorithm 4.2, which uses a space-sweep strategy.

Given $p = (p_x, p_y, p_z)$ and $q = (q_x, q_y, q_z)$ in S, we define the sweep-plane to be the yz-plane with origin (p_y, p_z). We have that a point $y \in \text{Mini}_{pq}$ must be such that its projection onto the sweep-plane belongs to the same quadrant as the projection of q. Hence, without loss of generality, we always assume the projection of q to belong to the first quadrant of the sweep-plane. In Algorithm 4.2 this reflects into the definition of p' and q' on line 8. The idea is to maintain a front data structure for each quadrant of the sweep-plane, and use it to test whether $\{p, q\}$ is a Minibox edge or not.

Algorithm 4.2 Minibox edges of a finite set of points S in three-dimensions.

Input: array *points*, the finite set of points S in $(\mathbb{R}^3, d_\infty)$.

```
 1: $edges \leftarrow$ empty list of two-tuples of integers
 2: Sort $points$ on their $x$-coordinate
 3: for $i = 0$ to $|S| - 1$ do
 4:    $fronts \leftarrow$ list of four empty red-black trees, one per quadrant
 5:    for $j = i + 1$ to $|S| - 1$ do
 6:       $p, q \leftarrow points[i], points[j]$
 7:       $p', q' \leftarrow (0, 0), (|q_y - p_y|, |q_z - p_z|)$ projections on the sweep-plane
 8:       $k \leftarrow$ index such that $(q_y, q_z)$ is in the $k$-th quadrant of the sweep-plane
 9:       if $fronts[k]$ is non-empty then
10:          $y' \leftarrow$ first element to the left of $q'$ in $fronts[k]$ bisecting on $q'_x$
11:          if $y'$ does not exist then
12:             Add $(i, j)$ to $edges$
13:             Delete the points in $fronts[k]$ that dominate $q'$, add $q'$ in $fronts[k]$
14:          else
15:             if $y' \notin \mathrm{Mini}_{p'q'}$ then
16:                Add $(i, j)$ to $edges$
17:                Delete the points in $fronts[k]$ that dominate $q'$, add $q'$ in $fronts[k]$
18:             end if
19:          end if
20:       else
21:          Add $(i, j)$ to $edges$, and add $q'$ to $fronts[k]$
22:       end if
23:    end for
24: end for
25: return $edges$
```

At each step of the inner loop on lines $5-23$, we have that $\{p, q\}$ is a Minibox edge if and only if $\mathrm{Mini}_{p'q'}$ does not contain a point y' in the sweep-plane. Because we restrict ourselves to the first quadrant, we only need to check whether or not q' dominates any y' projected from a $y \in S$ with y_x in the range (p_x, q_x). To speed this up we can store the points y' as we sweep on (p_x, q_x) in a red-black tree front, sorting them on their first coordinate, and then check if $\mathrm{Mini}_{p'q'}$ is empty by searching among the points in this front. In particular, we only store the points q' which are adding a Minibox edge, i.e. those that do not dominate points in the front. This happens on lines $13, 17, 21$ of Algorithm 4.2. The other points q'', dominating another point y' already in the front, are not needed. This is because if a future q' dominates q'', then it must also dominate y'. Furthermore, it may happen for q' to be dominated by points previously stored in the front. In this case, these are no longer needed, as for q'' above, and we can replace them with q', which happens on lines 13 and 17. A consequence of the way points are stored in and deleted from the red-black tree front is that these are the vertices of a staircase in the first quadrant of the sweep-plane, i.e. sorting the points using their first coordinates, their second coordinates are monotonically decreasing. This disposition of points is similar to those in the examples given for the two-dimensional case in Figure 4.10. The difference is that q' can be dominated by one of the points already in the front. To find y' dominated by q' in the front we can bisect on the first coordinate values of its points. This follows because if q' dominates any point in the front, then it also has to dominate the point in the front directly to its left, by the fact that the front is a staircase.

Proposition 4.5.2. *Let S be a set of finite points in $(\mathbb{R}^3, d_\infty)$. Algorithm 4.2 can be used to find the Minibox edges on S in $O(n^2 \log(n))$ time.*

Proof. The correctness of Algorithm 4.2 is discussed above. The inner loop may require to delete and add $O(n)$ points into a red-black tree, and to bisect on the same tree $O(n)$ times. Since either deleting, adding, or bisecting on a red-black tree requires $O(\log(n))$ operations, we conclude that the inner loop takes a total of $O(n \log(n))$ operations. Hence, Algorithm 4.2 has $O(n^2 \log(n))$ complexity. □

Points in higher dimensions. For points in general dimension $d \geq 4$, we propose different strategies, using a decreasing amount of additional storage, to test whether $\mathrm{Mini}_{pq} \cap S$ is empty for each pair of points in S.

For instance, high-dimensional range trees with fractional cascading [dBCvKO08, Section 5.6] can be used to answer orthogonal range emptiness queries in $O(\log^{d-1}(n))$ time, at the additional cost of $O(n \log^{d-1}(n))$ storage. By testing all pairs of points in S, we have a $O(n^2 \log^{d-1}(n))$ algorithm. Similarly, kd-trees [dBCvKO08, Section 5.2] can be used to answer the same query in $O(n^{1-\frac{1}{d}})$ time, only taking $O(n)$ additional storage, resulting in a $O(n^{3-\frac{1}{d}})$ algorithm for finding all the edges contained in any Minibox complex. Furthermore, we note that by the curse of dimensionality, if d becomes too big it might be faster to test each of the $\frac{n(n-1)}{2}$ pairs of points in S via a brute force strategy, searching all points in S sequentially. This results in a $O(dn^3)$ total time algorithm but does not require storing any additional data structure. The choice among these options depends on the amount of memory that can be spared for storing additional data structures. Moreover, we note that each of the above strategies could take advantage of parallel implementations using the independence of tests on each pair of points in S.

Finally, we also mention that in the Word RAM model of computation the offline orthogonal range counting algorithm of [CP10] can be used to find all empty miniboxes on S in constant dimension $d \geq 3$ in $O(n^2 \log^{d-2+\frac{1}{d}}(n))$. Anyway, as remarked in [CP10], for this algorithm to be applicable to floating-point numbers one needs to assume that the word size is at least as large as both $\log(n)$ and the maximum size of an input number.

4.6 Computational Experiments

In this final section, we present computational experiments giving empirical evidence of the speedup obtained by using Minibox filtrations in the calculation of zero and one-dimensional Čech persistence diagrams of S in ℓ_∞ metric. Moreover, we compute the persistence diagrams of Alpha flag, Minibox, and Čech filtrations obtained using randomly sampled points in $[0,1]^3 \subseteq (\mathbb{R}^3, d_\infty)$. These allow us to illustrate the similarities and dissimilarities between two-dimensional diagrams of these filtrations.

We use the implementation of the persistent homology algorithm provided by the `Ripser.py` [TSBO18] Python package, in combination with the algorithms of the `persty` Python package. All computations were run on a laptop with Intel Core i7-9750H CPU with six physical cores clocked at 2.60GHz with 16GB of RAM.

Size of Minibox filtrations. First, we study the expected size of Minibox filtrations versus the size of Čech filtrations. Our filtrations contain vertices, edges, and triangles because we only need to compute zero and one-dimensional persistence diagrams. So we have that the Čech filtration contains $\Theta(n^3)$ simplices. Given the edges in the maximal Minibox complex of S, the clique triangles on these can be found in $O(nk^2)$ time, where k is the maximum degree of any point in S, i.e. the maximum number of Minibox edges a point is contained in. Moreover $O(nk^2)$ is also an upper bound on the number of possible Minibox triangles, and by Proposition 4.4.7 it follows that the expected value of k for a uniformly distributed finite set of random points is $\Theta\left(\frac{2^{d-1}}{(d-1)!}\log^{d-1}(n)\right)$. Hence, we expect the Minibox filtration of S to contain fewer simplices compared to the Čech filtration. We give empirical evidence of this by calculating the expected number of Minibox simplices for 500, 1000, 1500, and 2000 uniformly distributed random points, averaging over five runs. Table 4.2 presents our results for Minibox filtrations in two, three and four dimensions. The number of simplices contained in the Čech filtrations are listed for comparison.

Running time and memory usage. Next, we explore the use of Minibox filtrations for the computation of Čech persistence diagrams of $S \subseteq (\mathbb{R}^d, d_\infty)$ in homological dimensions zero and one. As already mentioned, we make use of the `Ripser.py` package, which provides a Python interface to Ripser [Bau19] C++ code. In particular, we think of Minibox filtrations as sparse filtrations, and feed into the persistent homology algo-

Table 4.2: Average number of simplices contained in the Minibox and Čech filtrations for different input sizes.

	n = 500	n = 1000	n = 1500	n = 2000
Minibox 2D	0.01×10^6	0.03×10^6	0.05×10^6	0.07×10^6
Minibox 3D	0.17×10^6	0.50×10^6	0.91×10^6	1.38×10^6
Minibox 4D	1.19×10^6	4.50×10^6	9.41×10^6	15.65×10^6
Čech	20.83×10^6	166.67×10^6	562.50×10^6	1333.34×10^6

rithm a precomputed sparse matrix in coordinate format. We give timing and memory usage results for points in the range $[500, 32000]$ for Minibox filtrations, averaging over five runs. In the case of Čech filtrations, we limit our experiments to a maximum of 8000 points because of memory constraints. Moreover, we consider only points in $\mathbb{R}^2$, as results are similar in higher dimensions.

We list our results in Tables 4.3, 4.4, 4.5, and 4.6, where columns correspond to different sizes of the input points set S, and times are given in seconds. In particular, we use Algorithm 4.1 for edges in Table 4.3, Algorithm 4.2 for edges in Table 4.4, and a brute force algorithm for edges in Table 4.5. We also report the average total peak memory use in megabytes.[1]

In all the experiments, the reduced number of simplices of Minibox filtrations results in a substantial improvement in memory usage over Čech filtrations, and in a speedup in the computation of Dgm_0 and Dgm_1. This allows to increase the maximum size of inputs of the persistence algorithm, given a fixed amount of available memory. The price is having to precompute Minibox edges. We note that this computation could also take advantage of implementations parallelizing the inner loops of Algorithms 4.1 and 4.2, or the individual checks on edges of any brute force algorithm, as already mentioned in Section 4.5.

Table 4.3: Timing (seconds) and memory usage (MB) with Minibox filtrations of points in $\mathbb{R}^2$.

	500	1000	2000	4000	8000	16000	32000
Edges time	0.008	0.016	0.047	0.117	0.289	0.891	2.852
Sparse matrix time	0.023	0.070	0.141	0.312	0.734	1.562	3.406
$\text{Dgm}_{0,1}$ time	0.008	0.016	0.031	0.078	0.172	0.477	1.148
Total time	0.039	0.102	0.219	0.507	1.195	2.929	7.406
Peak Memory usage	2.92	5.52	11.51	25.15	53.50	112.1	246.3

[1] In Windows this was measured using the Win32 function `GetProcessMemoryInfo()` to obtain the `PeakWorkingSetSize` memory attribute of the Python process building sparse matrices and computing persistence diagrams.

Table 4.4: Timing (seconds) and memory usage (MB) with Minibox filtrations of points in $\mathbb{R}^3$.

	500	1000	2000	4000	8000	16000	32000
Edges time	0.062	0.188	0.586	2.047	7.500	27.89	110.6
Sparse matrix time	0.117	0.281	0.742	1.836	4.609	11.29	26.56
$\mathrm{Dgm}_{0,1}$ time	0.016	0.055	0.211	0.547	1.664	4.516	12.34
Total time	0.195	0.523	1.539	4.429	13.77	43.70	149.5
Peak memory usage	9.22	21.87	54.91	137.3	329.3	770.1	1848

Table 4.5: Timing (seconds) and memory usage (MB) with Minibox filtrations of points in $\mathbb{R}^4$.

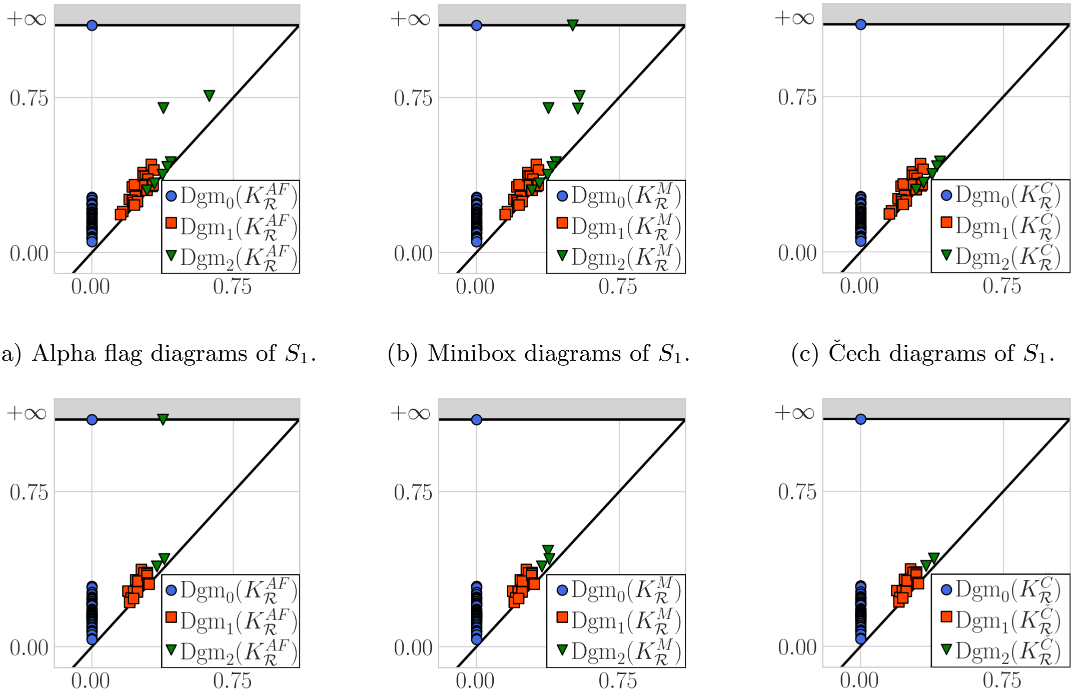
	500	1000	2000	4000	8000	16000	32000
Edges time	0.273	1.648	9.430	54.16	307.1	1657	8866
Sparse matrix time	0.258	0.727	2.055	6.250	15.68	43.52	107.8
$\mathrm{Dgm}_{0,1}$ time	0.070	0.227	0.797	2.539	9.320	27.02	107.3
Total time	0.601	2.601	12.281	62.95	332.1	1728	9081
Peak memory usage	19.19	51.18	155.4	410.4	1122	2841	7960

Table 4.6: Timing (seconds) and memory usage (MB) with Čech filtrations of points in $\mathbb{R}^2$.

	500	1000	2000	4000	8000
Sparse matrix	0.656	2.758	11.05	44.79	178.7
$\mathrm{Dgm}_{0,1}$	0.133	0.602	2.958	13.31	66.22
Total time	0.789	3.359	14.01	58.10	244.9
Peak memory usage	42.05	151.14	614.13	2532	10340

Differences in higher-dimensional diagrams. We present two examples of Alpha flag, Minibox, and Čech persistence diagrams, obtained from distinct $S_1, S_2 \subseteq (\mathbb{R}^d, d_\infty)$. These finite point sets were obtained by randomly sampling fifty points in $[0,1]^3 \subseteq \mathbb{R}^3$. The persistence diagrams were calculated with `Ripser.py` passing in the appropriate space matrix. For the Alpha flag case the edges belonging to the Delaunay complex of S_1 and S_2 were computed with a brute force strategy using the result of Proposition 4.1.3, i.e. checking if $A_e^{\bar{r}}$ is covered by $\bigcup_{y \in S \setminus e} B_{\bar{r}}(y)$ for each pair $p, q \in S$.

The first row in Figure 4.11 contains the diagrams of S_1. In this case $\mathrm{Dgm}_2(K_{\mathcal{R}}^M)$ contains a point at infinity, while $\mathrm{Dgm}_2(K_{\mathcal{R}}^{AF})$ does not. Furthermore, both contain additional off-diagonal points, which do not coincide. In the second row of Figure 4.11 we have the diagrams of S_2. In this case, it is $\mathrm{Dgm}_2(K_{\mathcal{R}}^{AF})$ that contains a point at infinity, while $\mathrm{Dgm}_2(K_{\mathcal{R}}^M)$ only has an additional off-diagonal point. This shows that it is possible to obtain Alpha flag and Minibox diagrams with off-diagonal points not contained in the corresponding Čech diagrams in homological dimensions higher than one. Furthermore, $\mathrm{Dgm}_2(K_{\mathcal{R}}^{AF})$ and $\mathrm{Dgm}_2(K_{\mathcal{R}}^M)$ are generally different and are not one a subset of the other.



(a) Alpha flag diagrams of S_1. (b) Minibox diagrams of S_1. (c) Čech diagrams of S_1.

(d) Alpha flag diagrams of S_2. (e) Minibox diagrams of S_2. (f) Čech diagrams of S_2.

Figure 4.11: Persistence diagrams of finite sets of three-dimensional points in ℓ_∞ metric space. Each row contains the diagrams of a different finite point set. These empirically show the equality of diagrams in dimensions zero and one, and illustrate the possible differences between diagrams of Alpha flag, Minibox, and Čech filtrations in homological dimension two.

4.7 Discussion

This chapter provides tools for the efficient computation of Čech persistence diagrams of a finite set of points S in ℓ_∞ metric space. The central idea is to make use of filtrations of flag complexes on S — Alpha flag and Minibox complexes — so that edges information is all that is needed to build them. This way only the simplices up to dimension $h+1$ need to be operated on if we are interested in computing persistence diagrams up to homological dimension h.

On the other hand, Alpha filtrations of points in Euclidean metric require finding the full Delaunay complex K^D of S. Algorithms for finding this K^D [HB08] make use of the empty circumsphere property of Delaunay top-dimensional simplices, of which there are $O(n^{\lceil \frac{d}{2} \rceil})$ for $S \subseteq \mathbb{R}^d$. Thus, it is not possible to determine which edges are in the Euclidean Delaunay complex without having to consider its d-dimensional simplices.

We prove the equivalence of Alpha flag, Minibox of Čech filtrations in homological dimensions zero and one of points in ℓ_∞ metric space, see Theorems 4.3.6 and

4.4.4. Moreover, it is shown that for n randomly sampled points Minibox filtrations are expected contain a number of edges proportinal to $n \cdot \text{polylog}(n)$, thus improving over the $\frac{n(n-1)}{2}$ edges of Čech filtrations. Algorithms are also described for finding minibox edges. For points in $\mathbb{R}^2$, it is given a $O(n^2)$ plane-sweep algorithm, while for points in $\mathbb{R}^3$ a $O(n^2 \log(n))$ space-sweep one. In dimension $d \geq 4$, the running time becomes $O(n^2 \log(n)^{d-1})$ using orthogonal range queries.

The final experiments section illustrates in practice the speedup obtained using Minibox filtrations. Examples are also given showing that for higher-dimensional homology Alpha flag and Minibox filtrations are related to Čech filtrations.

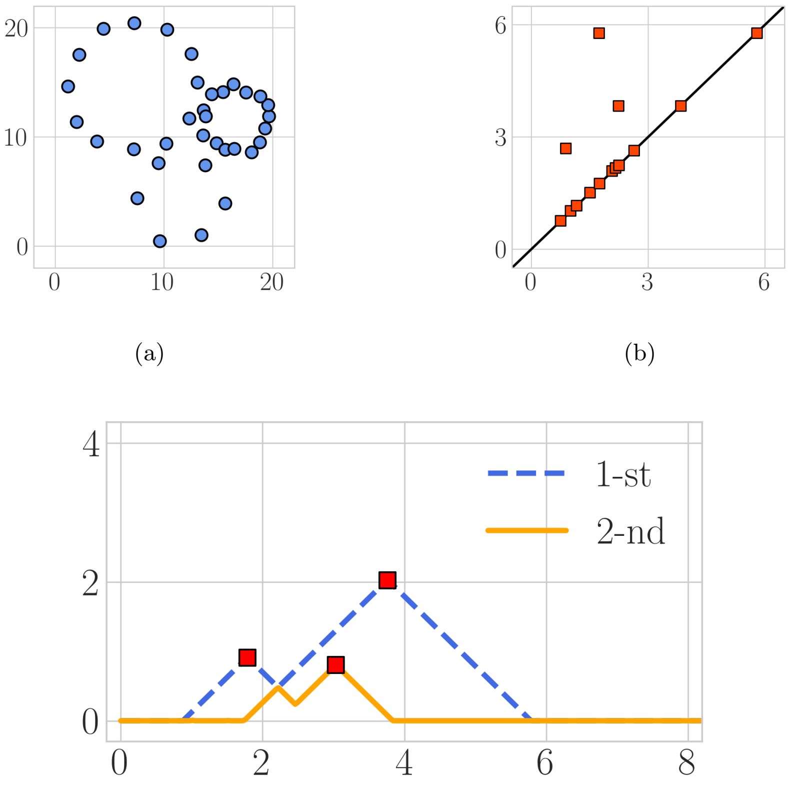
Future work could focus on determining whether alternative filtrations exist that can be used for computing Čech persistence diagrams in homological dimension two. Efficient algorithms for finding their simplices would also need to be described, as done in this chapter with edges of Minibox filtrations.

Chapter 5

Cumulative Landscapes for Supervised Classification

An application of the TDA descriptors considered in this thesis is their use as signatures of image and shape data. For example, persistence diagrams can be applied to classification problems involving three-dimensional shapes represented as a finite set of points in $\mathbb{R}^3$ [COO15]. A nice property of persistence diagrams is their stability with respect to small perturbations in the given input set of points, which makes them robust to noise present in real-world data. On the other hand, Euler characteristic curves and surfaces have the advantage of consisting of a finite number of numerical values. This allows for their direct application as feature vectors in the context of supervised classification problems. For this reason, a growing number of research papers has dealt with the problem of encoding the information of persistence diagrams into finite vectorial representations [Bub15, AEK$^+$17, RCIU19].

In this chapter, we propose two new methods for vectorizing persistence diagrams. The goal is to describe computationally efficient methods, which produce small feature vectors leading to good classification results. The efficacy of these is tested on supervised classification problems using open-source datasets. In particular, average accuracy scores are used to compare our methods to other TDA vectorization methods. Moreover, the algorithms described in Chapter 3 are used to compute Euler characteristic curves and Euler characteristic surfaces, which are made into feature vectors and added to the accuracy scores comparison of vectorizations of persistence diagrams.



(a)

(b)

(c)

Figure 5.1: **(a)** Points randomly sampled on three circles in $\mathbb{R}^2$. **(b)** Delaunay-Čech persistence diagram in homological dimension one of points in (a). **(c)** First two persistence landscape functions of the persistence diagram in (b).

5.1 Persistence Landscapes

A possible strategy is to map persistence diagrams into functions of one or more real variables, which can then be discretized by sampling their values. For instance *persistence landscapes* [Bub15] are defined as sequences of continuous piecewise linear functions.

Definition 5.1.1. Let $\mathrm{Dgm}_h(K_\mathcal{R}) = \{p_i = (b_i, d_i)\}_{i \in I}$ be a persistence diagram in homological dimension $h \geq 0$, and $T_1, T_2 \in \mathbb{R}$ be such that $T_1 \leq b_i \leq d_i \leq T_2$ for each $i \in I$. Defined the triangular function

$$\mathrm{tri}_i(t) = \begin{cases} t - b_i, & t \in \left[b_i, \frac{b_i+d_i}{2}\right] \\ d_i - t, & t \in \left[\frac{b_i+d_i}{2}, d_i\right] \\ 0, & \text{otherwise} \end{cases} \tag{5.1}$$

for each point $p_i = (b_i, d_i)$, the *persistence landscape* of $\mathrm{Dgm}_h(K_\mathcal{R})$ is the sequence of functions $\lambda_k(t) : [T_1, T_2] \to \mathbb{R}$ defined by

$$\lambda_k(t) = \text{kmax}_{i \in I} \text{tri}_i(t), \tag{5.2}$$

for $t \in [T_1, T_2]$ and $k \in \mathbb{N}$, where kmax is the kth-largest value in a set.

Example: Persistence landscapes of one-dimensional diagram. Figure 5.1c shows the first and second persistence landscape functions obtained from the persistence diagram $\text{Dgm}_1(K_{\mathcal{R}}^{D\check{C}})$ in Figure 5.1b. This was computed using the Delaunay-Čech filtration of the points in Figure 5.1a, introduced in Section 2.4. Notice that $\lambda_1(t)$ and $\lambda_2(t)$ are completely determined by the three points of maximum persistence in $\text{Dgm}_1(K_{\mathcal{R}}^{D\check{C}})$, which are also represented in Figure 5.1c.

Fixed a resolution parameter $m \in \mathbb{N}$, and defined $\Delta = T_2 - T_1$ and $\delta = \frac{\Delta}{m}$, any persistence landscape function $\lambda_k(t)$ can be discretized into the vector of real values

$$v_k = \left[\lambda_k\left(T_1\right), \lambda_k\left(T_1 + \delta\right), \ldots, \lambda_k\left(T_1 + (m-1)\delta\right), \lambda_k\left(T_2\right)\right].$$

Concatenating the vectors $\{v_k\}_{k=1}^{\bar{k}}$, corresponding to the first $\bar{k} \in \mathbb{N}$ functions in the sequence of a persistence landscape of $\text{Dgm}_h(K_{\mathcal{R}})$, it is obtained a vectorization of the given persistence diagram.[1] This procedure requires fixing the integer values of both the resolution m and the number of landscape functions $\bar{k}$. The optimal values of these parameters need to be determined each time discretized persistence landscapes are computed for a given dataset.

A concept related to persistence landscapes is the following, which was introduced in [CDSO14].

Definition 5.1.2. Let $\text{Dgm}_h(K_{\mathcal{R}}) = \{p_i = (b_i, d_i)\}_{i \in I}$ be a persistence diagram in homological dimension $h \geq 0$, and $T_1, T_2 \in \mathbb{R}$ be such that $T_1 \leq b_i \leq d_i \leq T_2$ for each $i \in I$. Fixed $p > 0$ and defined the weights $w_i = |d_i - b_i|^p$ for each $i \in I$, the *power-weighted silhouette* of $\text{Dgm}_h(K_{\mathcal{R}})$ is the function $\phi(t) : [T_1, T_2] \to \mathbb{R}$ defined by

$$\phi(t) = \frac{\sum_{i \in I} w_i \text{tri}_i(t)}{\sum_{i \in I} w_i}, \tag{5.3}$$

where $\text{tri}_i(t)$ is the triangular function of the point p_i in the diagram.

Any given silhouette $\phi(t)$ can be discretized by fixing a single resolution value m, but

[1] The concatenation of two vectors v_1 and v_2 containing n_1 and n_2 elements respectively is the vector with $n_1 + n_2$ elements whose first n_1 elements coincide with those of v_1, and the last n_2 elements with those of v_2.

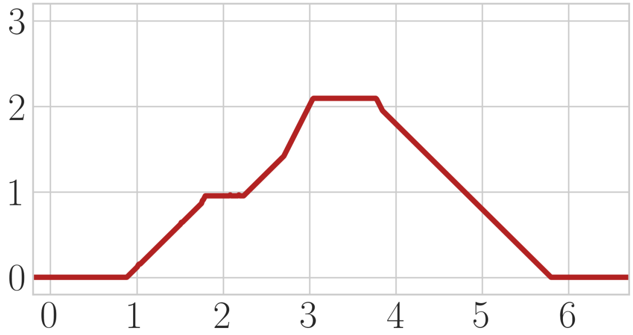


Figure 5.2: Cumulative landscape of $\mathrm{Dgm}_1(K_{\mathcal{R}}^{D\check{C}})$ is Figure 5.1b.

its definition requires to pick an application-specific optimal value of the power p.

Both in the case of persistence landscapes and silhouettes, a pair of parameters needs to be determined to derive feature vectors for supervised classification tasks. When dealing with a classification problem, the optimal values of these parameters can be chosen via k-fold cross-validation on a subset of the data at hand.

5.2 Cumulative Landscapes

Persistence landscapes and silhouettes are two instances of what is generally called a *summary function* of persistence diagrams. In this section, it is introduced a new type of summary function, which is related to power-weighted silhouettes.

Definition 5.2.1. Let $\mathrm{Dgm}_h(K_{\mathcal{R}}) = \{p_i = (b_i, d_i)\}_{i\in I}$ be a persistence diagram in homological dimension $h \geq 0$, and $T_1, T_2 \in \mathbb{R}$ be such that $T_1 \leq b_i \leq d_i \leq T_2$ for each $i \in I$. The *cumulative landscape* of $\mathrm{Dgm}_h(K_{\mathcal{R}})$ is the function $\Lambda(t) : [T_1, T_2] \to \mathbb{R}$ defined by

$$\Lambda(t) = \sum_{i\in I} \mathrm{tri}_i(t), \tag{5.4}$$

where $\mathrm{tri}_i(t)$ is the triangular function of the point p_i in the diagram, as given in Definition 5.1.1.

Like the summary functions introduced earlier in this chapter, cumulative landscapes are continuous and piecewise linear functions. Besides, as for power-weighted silhouettes, all that is required to vectorize cumulative landscapes is a fixed resolution parameter m, which can be used to sample $\Lambda(t)$ on its domain $[T_1, T_2]$. In particular, this is the only parameter to be optimized if using cumulative landscapes for supervised classification, because the dependence on the parameter p used to define the weight w_i in silhouettes has been removed. An example of a cumulative landscape is presented in Figure 5.2. This is the curve obtained by summing up the triangular functions associated with the points in the persistence diagram in Figure 5.1b. Moreover, under genericity assumptions on the coordinates of the points in $\mathrm{Dgm}_h(K_{\mathcal{R}})$, it can be shown that cumulative landscapes can be used to reconstruct the persistence diagrams defining them. Thus, with the appropriate hypotheses, the mapping into cumulative landscapes is information preserving.

Definition 5.2.2. A persistence diagram $\mathrm{Dgm}_h(K_{\mathcal{R}}) = \{p_i = (b_i, d_i)\}_{i \in I}$ is *generic* if the set $X = \bigcup_{i \in I} \left\{ b_i, d_i, \frac{b_i+d_i}{2} \right\}$ is such that

(i) $x \neq x'$ for each $x, x' \in X$;

(ii) $\frac{d_i - b_i}{2} \neq \left| \frac{b_i+d_i}{2} - x \right|$ for each $i \in I$ and $x \in X \setminus \{b_i, d_i\}$.

Proposition 5.2.3. *Let $\Lambda(t)$ be the cumulative landscape of $Dgm_h(K_{\mathcal{R}})$. If the persistence diagram $Dgm_h(K_{\mathcal{R}})$ is generic, then it is possible to reconstruct its set of points from $\Lambda(t)$.*

Proof. The idea is to use the first derivative $\Lambda'(t)$ of the cumulative landscape to identify the maximums of the triangular functions $\mathrm{tri}_i(t)$. This $\Lambda'(t)$ is a piecewise constant (and discontinuous) function with values in $\mathbb{Z}$, because it is the sum of

$$\mathrm{tri}_i'(t) = \begin{cases} 1, & t \in \left(b_i, \frac{b_i+d_i}{2} \right) \\ -1, & t \in \left(\frac{b_i+d_i}{2}, d_i \right) \\ 0, & \text{otherwise} \end{cases} \tag{5.5}$$

for each $i \in I$. Thus its value changes at the point of discontinuity $t = b_i$, $t = d_i$, and $t = \frac{b_i+d_i}{2}$ for each $i \in I$.

Given X as in Definition 5.2.2, let $\delta > 0$ be such that $\delta < \min_{x,x' \in X} |x - x'|$. By hypothesis $\mathrm{Dgm}_h(K_{\mathcal{R}})$ is generic, so property *(i)* of Definition 5.2.2 guarantees that

- $\Lambda'(b_i + \delta) - \Lambda'(b_i - \delta) = \Lambda'(d_i + \delta) - \Lambda'(d_i - \delta) = 1$,
- $\Lambda' \left(\frac{b_i+d_i}{2} + \delta \right) - \Lambda' \left(\frac{b_i+d_i}{2} - \delta \right) = -2$,

for each $i \in I$. Otherwise, some b_i, d_i, or $\frac{b_i+d_i}{2}$ would need to coincide. Hence, the midpoints of the triangular functions $\mathrm{tri}_i(t)$ are uniquely identified as the values at which $\Lambda'(t)$ decreases by 2. Then, for each midpoint $\frac{b_i+d_i}{2}$ the pair of values $t_1 = b_i$ and $t_2 = d_i$ is such that

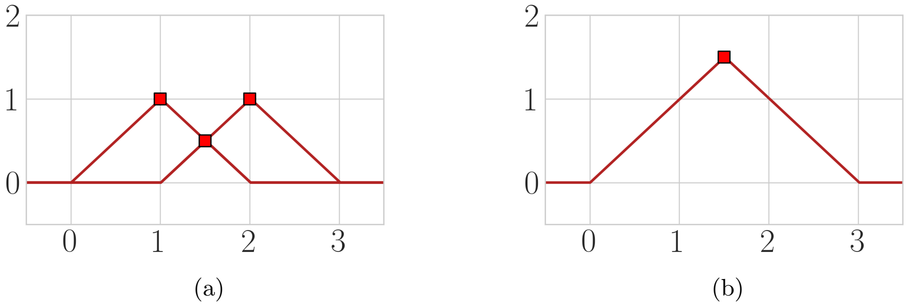


Figure 5.3: Two different sets of triangular functions whose sum is the same cumulative landscape. In **(a)** there are three functions with local maximums in $(1,1)$, $(1.5,0.5)$ and $(2,1)$; in **(b)** a single function with maximum in $(1.5,1.5)$.

- $\frac{b_i+d_i}{2} - t_1 = \frac{d_i-b_2}{2}$ and $t_2 - \frac{b_i+d_i}{2} = \frac{d_i-b_2}{2}$,
- $\Lambda'(t_1+\delta) - \Lambda'(t_1-\delta) = 1$,
- $\Lambda'(t_2+\delta) - \Lambda'(t_2-\delta) = 1$,

for each $i \in I$. Moreover, by point *(ii)* of Definition 5.2.2, there is only one pair of $t_1, t_2 \in [T_1, T_2]$ satisfying these properties. Thus the triangular function $\text{tri}_i(t)$, which is non-zero in the range $[t_1, t_2] = [b_i, d_i]$, must be in the sum defining $\Lambda(t)$. In conclusion, each value of t at which $\Lambda'(t)$ decreases by 2 uniquely identifies a point $(t_1, t_2) = (b_i, d_i)$ in the persistence diagram used to define $\Lambda(t)$. □

Note that if the given persistence diagram is not generic, then it may not be possible to decompose $\Lambda(t)$ into its triangular functions, and so obtain the points of the diagram. For instance, given the persistence diagram $\text{Dgm}_h(K_{\mathcal{R}}) = \{(0,2),(1,2),(1,3)\}$, its cumulative landscape could be decomposed both as the sum of the triangular functions with non-zero values in the intervals $[0,2]$, $[1,2]$, and $[1,3]$, and as the single triangular function with non zero-values in $[0,3]$. See Figure 5.3.

Example: Instability of cumulative landscapes. Given two finite set of points X and Y in $\mathbb{R}^d$, the Stability Theorem 2.3.10 of persistent homology ensures that small perturbations in X and Y result in small perturbations in the Čech persistence diagrams of X and Y. Here we show with an example that cumulative landscapes do not have the same property. In particular, we describe point sets in $\mathbb{R}^2$ at fixed Hausdorff distance $\varepsilon > 0$ such that the cumulative landscapes of their one-dimensional persistence diagrams are arbitrarily different.

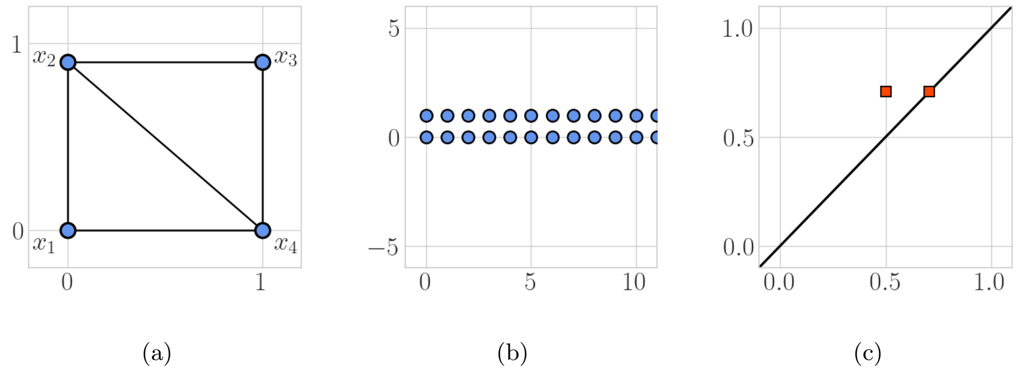


Figure 5.4: **(a)** Point set $Y^{(1)}$ in $\mathbb{R}^2$ together with the edges of its Delaunay triangulation. **(b)** Point set $Y^{(k)}$ in $\mathbb{R}^2$ producing an unstable cumulative landscape. **(c)** Čech persistence diagram in homological dimension one of the points in (a) and (b).

We define

$$X^{(k)} = \{(x, y) \in \mathbb{R}^2 \mid 0 \leq x \leq k \text{ and } 0 \leq y \leq 1 \text{ and } x, y \in \mathbb{N}\},$$
$$Y^{(k)} = \{(x, y - \varepsilon) \in \mathbb{R}^2 \mid 0 \leq x \leq k \text{ and } 0 \leq y \leq 1 \text{ and } x, y \in \mathbb{N}\},$$

where $\varepsilon \in (0, 1) \in \mathbb{R}$ is a fixed constant and $k \in \mathbb{N}$. We compute the Čech persistence diagrams of $X^{(k)}$ and $Y^{(k)}$ with Alpha complexes, as discussed in Section 2.4. This way, Delaunay triangulations on $X^{(k)}$ and $Y^{(k)}$, restrict the simplices contained in the filtrations used for this computation.

To begin with, we study the persistence diagrams in homological dimension one of $X^{(1)}$ and $Y^{(1)}$. Figure 5.4a shows the edges of a Delaunay triangulation of $Y^{(1)} = \{x_1, x_2, x_3, x_4\}$. The length of the diagonal edge $\{x_2, x_4\}$ is $\sqrt{(1-\varepsilon)^2 + 1}$, because $x_2 = (0, 1-\varepsilon) \in \mathbb{R}^2$. Moreover, both $\tau_1 = \{x_1, x_2, x_4\}$ and $\tau_2 = \{x_2, x_3, x_4\}$ are right-angled triangles with $\{x_2, x_4\}$ as hypotenuse, so τ_1 and τ_2 are added into the Alpha complex K_r^A of $Y^{(1)}$ with radius parameter $r = \frac{\sqrt{(1-\varepsilon)^2+1}}{2}$, because the circumcenter of right-angled triangles is the midpoint of their hypotenuse. Thus, we have that the 1-cycle containing $\{x_1, x_2\}$, $\{x_2, x_3\}$, $\{x_3, x_4\}$, and $\{x_1, x_4\}$ is created at $r = \frac{1}{2}$, and deleted at $r = \frac{\sqrt{(1-\varepsilon)^2+1}}{2}$ in the Alpha filtration of $Y^{(1)}$. Besides, there is a 1-cycle of zero persistence, which is both created and deleted at $r = \frac{\sqrt{(1-\varepsilon)^2+1}}{2}$. We conclude that

$$\left\{ \Big(\frac{1}{2}, \frac{\sqrt{(1-\varepsilon)^2+1}}{2}\Big), \Big(\frac{\sqrt{(1-\varepsilon)^2+1}}{2}, \frac{\sqrt{(1-\varepsilon)^2+1}}{2}\Big) \right\},$$

is the Čech persistence diagram in homological dimension one of $Y^{(1)}$. Similarly, the

Čech persistence diagram in homological dimension one of $X^{(1)}$ is

$$\left\{\Big(\frac{1}{2},\frac{\sqrt{2}}{2}\Big),\Big(\frac{\sqrt{2}}{2},\frac{\sqrt{2}}{2}\Big)\right\},$$

because the ε constant is missing from the points of $X^{(1)}$.

Then, the Čech persistence diagrams in homological dimension one of $X^{(k)}$ and $Y^{(k)}$ consist of the same points of those of $X^{(1)}$ and $Y^{(1)}$ above, but with the points having multiplicity $k-1$. This follows by the fact that a Delaunay triangulation on $2k$ points like those in Figure 5.4b contains $k-1$ pairs of triangles with the same properties of τ_1 and τ_2 above.

Finally, we write $\Lambda_{X^{(k)}}(t)$ and $\Lambda_{Y^{(k)}}(t)$ for the cumulative landscapes of the persistence diagrams in homological dimension one of $X^{(k)}$ and $Y^{(k)}$. We have

$$||\Lambda_{X^{(k)}}(t)-\Lambda_{Y^{(k)}}(t)||_\infty \geq (k-1)\cdot\Big(\frac{\sqrt{2}}{2}-\frac{\sqrt{(1-\varepsilon)^2+1}}{2}\Big), \tag{5.6}$$

because the triangular functions of the points $\left(\frac{1}{2},\frac{\sqrt{2}}{2}\right)$ and $\left(\frac{1}{2},\frac{\sqrt{(1-\varepsilon)^2+1}}{2}\right)$ differ by $\frac{\sqrt{2}}{2}-\frac{\sqrt{(1-\varepsilon)^2+1}}{2}$ at $t=\frac{\sqrt{(1-\varepsilon)^2+1}}{2}$, and there are $k-1$ such functions in the sums defining $\Lambda_{X^{(k)}}(t)$ and $\Lambda_{Y^{(k)}}(t)$. Thus, fixed any value of $\varepsilon\in(0,1)$, the value of $||\Lambda_{X^{(k)}}(t)-\Lambda_{Y^{(k)}}(t)||_\infty$ goes to infinity, with k going to infinity. We conclude that, a small perturbation of a set of points X may result in large differences in the cumulative landscapes derived from X. So, the stability property of persistence diagrams does not hold for cumulative landscapes.

5.3 Fourier Coefficients of Cumulative Landscapes

The simple structure of cumulative landscapes allows to reduce the dimensionality of the output vectors obtained by discretizing $\Lambda(t)$, without losing much of the structural information they encode. The idea is to think of $\Lambda(t):[T_1,T_2]\to\mathbb{R}$ as a periodic function of period $T=T_2-T_1$, and to use its Fourier coefficients $\tilde{a}_k$ and $\tilde{b}_k$ as the elements of the desired feature vectors. Analytical expressions for the values of these coefficients are derived in this section, and are applied in the next to classification problems.

Definition 5.3.1. Let $\Lambda(t):[T_1,T_2]\to\mathbb{R}$ be the cumulative landscape of a persistence diagrams $\mathrm{Dgm}_h(K_{\mathcal{R}})$ in homological dimension $h\geq 0$. The *periodic cumulative landscape* of $\mathrm{Dgm}_h(K_{\mathcal{R}})$ is the periodic function $\tilde{\Lambda}(t):\mathbb{R}\to\mathbb{R}$ defined by

$$\tilde{\Lambda}(t)\restriction_{[T_1+j\cdot T,T_2+j\cdot T]}=\Lambda(t-j\cdot T), \tag{5.7}$$

for $j \in \mathbb{Z}$, where $T = T_2 - T_1$.

The periodic cumulative landscape can be expanded into the *Fourier series*

$$s_{\bar{k}}(t) = \frac{\tilde{a}_0}{2} + \sum_{k=1}^{\bar{k}} \left(\tilde{a}_k \cdot \cos\left(\frac{2\pi k}{T} t\right) + \tilde{b}_k \cdot \sin\left(\frac{2\pi k}{T} t\right) \right), \tag{5.8}$$

where the coefficients $\tilde{a}_k$ and $\tilde{b}_k$ are

- $\tilde{a}_0 = \frac{2}{T} \int_{T_1}^{T_2} \tilde{\Lambda}(t) dt$;
- $\tilde{a}_k = \frac{2}{T} \int_{T_1}^{T_2} \tilde{\Lambda}(t) \cos\left(\frac{2\pi k}{T} t\right) dt$, for $1 \leq k \leq \bar{k}$;
- $\tilde{b}_k = \frac{2}{T} \int_{T_1}^{T_2} \tilde{\Lambda}(t) \sin\left(\frac{2\pi k}{T} t\right) dt$, for $1 \leq k \leq \bar{k}$;

as given in [Tol76, Section 1.6]. Furthermore, the following result (adapted to the notation of this section) guarantees that in our setting $s_{\bar{k}}(t)$ converges to $\tilde{\Lambda}(t)$ for $\bar{k}$ going to infinity.

Theorem 5.3.2 (Section 3.9 [Tol76])**.** *If $\tilde{\Lambda}(t)$ is an absolutely integrable function of period T which is piecewise smooth on the interval $[a, b]$, then for all t in $a < t < b$ the Fourier series $s_{+\infty}(t)$ of $\tilde{\Lambda}(t)$ converges to $\tilde{\Lambda}(t)$ at points of continuity and to the value*

$$\frac{\tilde{\Lambda}(t+0) + \tilde{\Lambda}(t-0)}{2}, \tag{5.9}$$

the arithmetic mean of the right-hand and left-hand limits, at points of discontinuity.

Given the cumulative landscape in Figure 5.2, its periodic version $\tilde{\Lambda}(t)$ is approximated with increasing accuracy by the Fourier series $s_2(t)$, $s_5(t)$, and $s_{20}(t)$, which are plotted in Figure 5.5 showing a single period $[T_1, T_2] = [0, 7]$.

In conclusion, fixed $\bar{k} \in \mathbb{N}$ large enough, the Fourier coefficients $\tilde{a}_0$, $\tilde{a}_k$, and $\tilde{b}_k$ encode most of the information of a cumulative landscape $\Lambda(t)$, and the vector

$$v_{\bar{k}}^F = \left[\tilde{a}_0, \tilde{a}_1, \tilde{a}_2, \ldots, \tilde{a}_{\bar{k}}, \tilde{b}_1, \tilde{b}_2, \ldots, \tilde{b}_{\bar{k}}\right]$$

can be used as a vectorial representation of the persistence diagram used to define $\Lambda(t)$. This contains $2\bar{k} + 1$ elements, which in practice (i.e. choosing parameters with k-fold cross-validation as done in the following section) results in smaller vectors compared to vectorizations of persistence landscapes and cumulative landscapes making use of resolution parameter m.

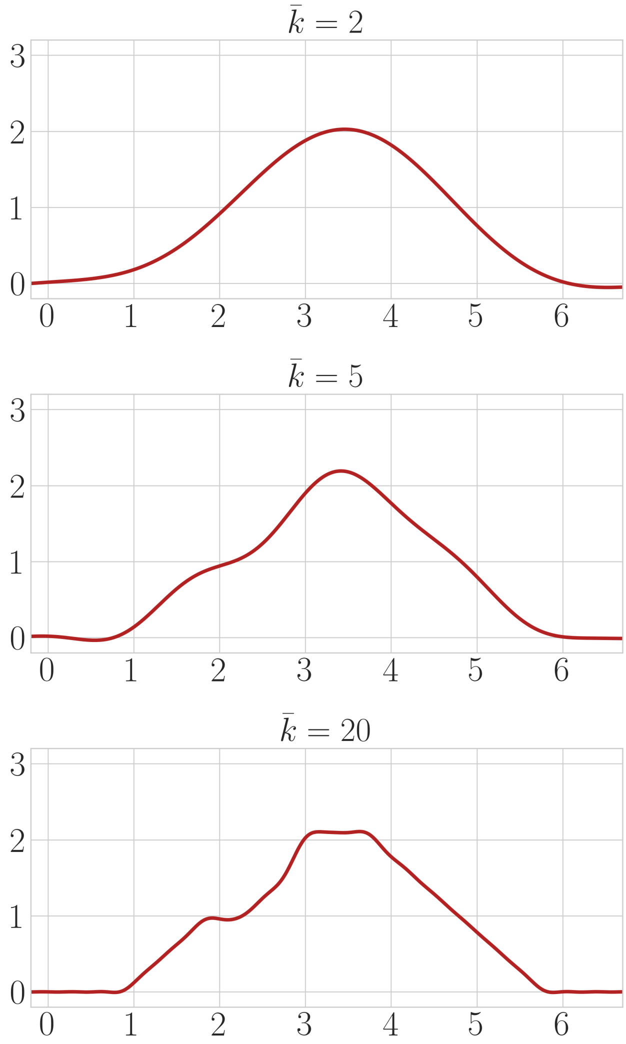


Figure 5.5: Fourier series of $\tilde{\Lambda}(t)$ derived from the cumulative landscape in Figure 5.2, for $\bar{k}$ fixed to three different values.

Analytical expressions of Fourier coefficients. The equations defining the Fourier coefficients $\tilde{a}_0$, $\tilde{a}_k$, and $\tilde{b}_k$ can be integrated to obtain their values explicitly in terms of mathematical expressions containing only trigonometric functions. Recall that $\Lambda(t) = \sum_{i\in I} \mathrm{tri}_i(t)$ is the cumulative landscape of $\mathrm{Dgm}_h(K_{\mathcal{R}}) = \{p_i = (b_i, d_i)\}_{i\in I}$ on the interval $[T_1, T_2]$ of length T.

Given that the integrals $\int_{T_1}^{T_2} \mathrm{tri}_i(t)dt = \int_{b_i}^{d_i} \mathrm{tri}_i(t)dt$ equal the area of the triangular functions $\mathrm{tri}_i(t)$ for each $i \in I$, which is $\frac{(d_i-b_i)^2}{4}$, it follows that

$$\begin{aligned}\tilde{a}_0 &= \frac{2}{T}\int_{T_1}^{T_2}\tilde{\Lambda}(t)dt = \frac{2}{T}\int_{T_1}^{T_2}\Lambda(t)dt = \frac{2}{T}\int_{T_1}^{T_2}\sum_{i\in I}\mathrm{tri}_i(t)dt \\ &= \frac{2}{T}\sum_{i\in I}\int_{T_1}^{T_2}\mathrm{tri}_i(t)dt = \frac{2}{T}\sum_{i\in I}\frac{(d_i-b_i)^2}{4} = \frac{\sum_{i\in I}(d_i-b_i)^2}{2T}.\end{aligned}$$

Next, the expression for $\tilde{a}_k$ can be derived by splitting $\int_{T_1}^{T_2}\mathrm{tri}_i(t)\cos\left(\frac{2\pi k}{T}t\right)dt$ in terms of its value on the intervals $[b_i, c_i]$ and $[c_i, d_i]$. The same works for $\tilde{b}_k$, for which it is only given the final expression.

$$\begin{aligned}\tilde{a}_k &= \frac{2}{T}\int_{T_1}^{T_2}\tilde{\Lambda}(t)\cos\left(\frac{2\pi k}{T}t\right)dt = \frac{2}{T}\int_{T_1}^{T_2}\Lambda(t)\cos\left(\frac{2\pi k}{T}t\right)dt \\ &= \frac{2}{T}\int_{T_1}^{T_2}\sum_{i\in I}\mathrm{tri}_i(t)\cos\left(\frac{2\pi k}{T}t\right)dt = \frac{2}{T}\sum_{i\in I}\int_{T_1}^{T_2}\mathrm{tri}_i(t)\cos\left(\frac{2\pi k}{T}t\right)dt \\ &= \frac{2}{T}\sum_{i\in I}\Bigg[\underbrace{\int_{b_i}^{\frac{b_i+d_i}{2}}(t-b_i)\cdot\cos\left(\frac{2\pi k}{T}t\right)dt}_{\textit{(i)}} + \underbrace{\int_{\frac{b_i+d_i}{2}}^{d_i}(d_i-t)\cdot\cos\left(\frac{2\pi k}{T}t\right)dt}_{\textit{(ii)}}\Bigg].\end{aligned}$$

The terms *(i)* and *(ii)* can be integrated by parts obtaining

(i) $\frac{d_i-b_i}{2}\cdot\frac{T}{2\pi k}\cdot\sin\left(\frac{b_i+d_i}{2}\frac{2\pi k}{T}\right)+\left(\frac{T}{2\pi k}\right)^2\cdot\cos\left(\frac{b_i+d_i}{2}\frac{2\pi k}{T}\right)-\left(\frac{T}{2\pi k}\right)^2\cdot\cos\left(b_i\frac{2\pi k}{T}\right)$

(ii) $-\left[\frac{d_i-b_i}{2}\cdot\frac{T}{2\pi k}\cdot\sin\left(\frac{b_i+d_i}{2}\frac{2\pi k}{T}\right)-\left(\frac{T}{2\pi k}\right)^2\cdot\cos\left(\frac{b_i+d_i}{2}\frac{2\pi k}{T}\right)+\left(\frac{T}{2\pi k}\right)^2\cdot\cos\left(d_i\frac{2\pi k}{T}\right)\right]$

Thus

$$\tilde{a}_k = \frac{T}{2\pi^2k^2}\sum_{i\in I}\Big[2\cos(\gamma_i k)-\cos(\beta_i k)-\cos(\delta_i k)\Big], \tag{5.10}$$

for each $k \geq 1$, where $c_i = \frac{b_i+d_i}{2}$, $\beta_i = b_i\frac{2\pi}{T}$, $\gamma_i = c_i\frac{2\pi}{T}$, $\delta_i = d_i\frac{2\pi}{T}$ for each $i \in I$. Similarly

$$\tilde{b}_k = \frac{T}{2\pi^2k^2}\sum_{i\in I}\Big[2\sin(\gamma_i k)-\sin(\beta_i k)-\sin(\delta_i k)\Big], \tag{5.11}$$

for each $k \geq 1$.

5.4 Supervised Classification Experiments

In this section vectorial representations of persistence diagrams derived from cumulative landscapes are applied to two supervised classification problems. The goal is to compare

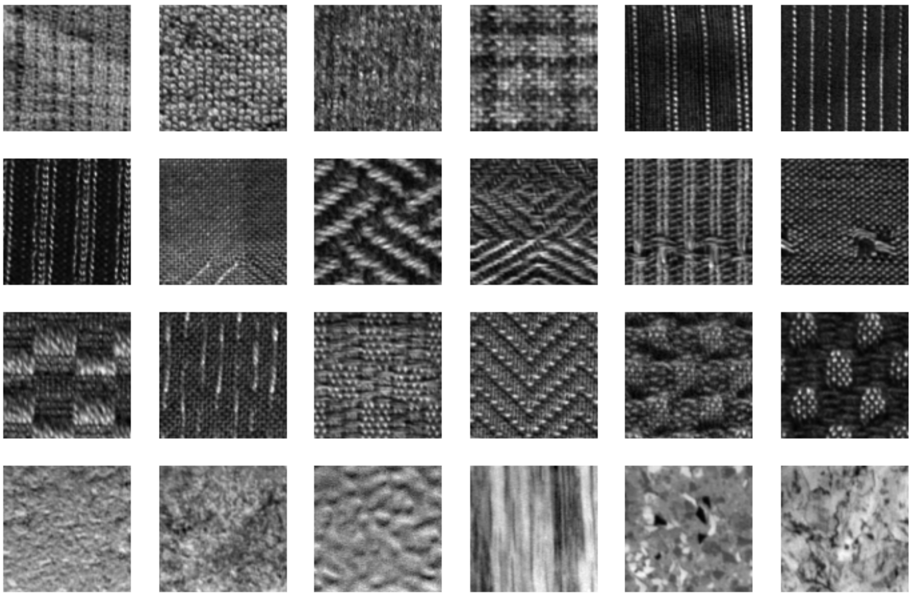


Figure 5.6: The 24 textures of the OUTEX_TC_00000 gray-scale image dataset.

the classification accuracy results obtained by applying standard machine learning algorithms with feature vectors derived with different methods from the same persistence diagrams. Moreover, accuracy results given by Euler characteristic feature vectors are also included in the final comparison tables.

All computations were performed on a laptop with Intel Core i7-9750H CPU with six physical cores clocked at 2.60GHz with 16GB of RAM.

Texture images. The first open-source dataset used to benchmark the effectiveness of TDA feature vectors is the OUTEX_TC_0000 test suite, which contains 480 gray-scale images of size 128 × 128 [OMP$^+$02]. These belong to 24 different classes, corresponding to as many types of textures, see Figure 5.6. The test suite also provides 100 random 50/50 test-train splits, with each class evenly represented by 10 images in the train and test data, that need to be used to compute average classification accuracy results.

To begin with, the Euler characteristic curves of pixel intensity values were computed for each image with Algorithm 2.1. Next, the discrete version of the Laplace operator [GW17] was applied to extract information about regions of high contrast in the OUTEX_TC_00000 images. In particular, the OpenCV [KB16] implementation was used with kernel

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

The result of this operation were Laplacian matrices of integers. Gray-scale absolute Laplacian images were produced by taking absolute values of the elements in these Lapacian matrices. Euler characteristic surfaces were computed with Algorithm 3.1 using the pairs of images obtained by taking each `OUTEX_TC_00000` image and its associate absolute Laplacian image. Finally, both Euler characteristic curves and surfaces were employed to produce feature vectors for classifying `OUTEX_TC_00000` images with standard machine learning algorithms. To obtained such vectors, step parameters $s_1, s_2 \in \mathbb{N}$ were chosen via 3-fold cross-validation on one-third of the data. The curves and surfaces were downsampled according to these steps by only keeping their elements in positions i and (i, j) such that $i \equiv 0 \pmod{s_1}$ and $j \equiv 0 \pmod{s_2}$.

In order to compete with the results of Euler characteristic surfaces, which integrate the information of two sublevel sets filtrations in one vector, multiple persistence diagrams were computed for each image. In practice, images were first downsampled to half their original width and height. Then, each of the resulting 64×64 images was mapped into a collection of four point clouds in $\mathbb{R}^3$. These were obtained from the downsampled images by thresholding their values at four different levels, and mapping the remaining pixels with value $v_{s,t}$ into the points $(s, t, v_{s,t})$ of $\mathbb{R}^3$. The four threshold levels were determined using the Euler characteristic curves of pixel intensities of `OUTEX_TC_00000` images. In particular, these were chosen to correspond to local maximums in the curve of average Euler characteristic changes of `OUTEX_TC_00000` images and to avoid producing empty point clouds. The result were the values $\{120, 127, 165, 255\}$. For an example of a collection of four point clouds derived from a single `OUTEX_TC_00000` image see Figure 5.7. The persistence diagrams in homological dimensions zero and one of these point clouds were computed with the `gudhi` [GUD21] Python package using both Delaunay-Čech filtrations with Euclidean distance, and Minibox filtrations with ℓ_∞ distance. This resulted in a collection of 8 diagrams for each `OUTEX_TC_00000` image, both for Delaunay-Čech and Minibox filtrations. All these diagrams were vectorized via persistence landscapes, cumulative landscapes, and Fourier coefficients of cumulative landscapes, as previously described in this section. The parameters $m, \bar{k} \in \mathbb{N}$ that needed to be fixed for this step were chosen applying 3-fold cross-validation to one-third of the `OUTEX_TC_00000` data. To conclude, the sets of 8 vectors corresponding to persistence landscapes, cumulative landscapes, and Fourier coefficients of cumulative landscapes were concatenated to produce the final persistence feature vectors.

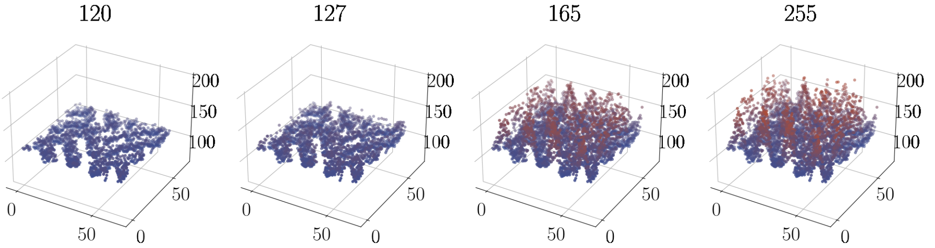


Figure 5.7: Four point clouds derived from one `OUTEX_TC_00000` image.

Table 5.1: Average classification accuracy results `OUTEX_TC_00000`

	Features preprocessing	
	-	Min-Max scaler
Euler char. curves - Intensity	$84.84 \pm 1.93\%$	$91.52 \pm 1.70\%$
Euler char. surfaces - (Intensity, Laplacian)	$\mathbf{97.20 \pm 0.89}\%$	$96.31 \pm 1.34\%$
Minibox pers. landcapes	$84.30 \pm 1.73\%$	$89.30 \pm 1.71\%$
Minibox cum. landscapes	$96.05 \pm 1.16\%$	$85.94 \pm 1.69\%$
Minibox Fourier coefficients	$95.45 \pm 1.22\%$	$95.44 \pm 1.23\%$
Delaunay-Čech pers. landcapes	$84.04 \pm 2.21\%$	$92.80 \pm 1.28\%$
Delaunay-Čech cum. landscapes	$94.53 \pm 1.61\%$	$88.04 \pm 1.89\%$
Delaunay-Čech Fourier coefficients	$95.84 \pm 1.13\%$	$96.40 \pm 1.23\%$

Classification on the 100 train-test splits provided by the test suite was performed using logistic regression [FHT09, Section 4.4] with ℓ_2 regularization and the `LIBLINEAR` [FCH+08] solver, as implemented by the **`scikit-learn`** [PVG+11] Python package. The inverse regularization strength parameter C was also determined with 3-fold cross-validation, at the same time of choosing the vectorization parameters m and $\bar{k}$, or the downsampling step in the case of Euler characteristic vectors. The average accuracy results for all feature vectors taken into consideration are in Table 5.1. The first column refers to results where no preprocessing on the features was used, and the second column to the average accuracy results obtained by applying a Min-Max scaler preprocessing step, i.e. transforming the training set so that each feature is in the range $[0, 1]$. The best results are attained by Euler characteristic surfaces without preprocessing, though similar performance is provided by Fourier coefficients of cumulative landscapes of Delaunay-Čech filtrations with Min-Max scaler preprocessing.

Three-dimensional human shapes. The second dataset considered in this section is the `SHREC 2014` collection of 400 real human 3D meshes, representing 40 human subjects in 10 different poses [PSR+14]. In practice, only the vertices of the meshes are used, producing points clouds in $\mathbb{R}^3$ on top of which Delaunay-Čech and Minibox filtrations are built in order to compute Euler characteristic curves/surfaces and persistence diagrams. Note that each shape in this dataset contains approximately $15,000$ vertices. The task

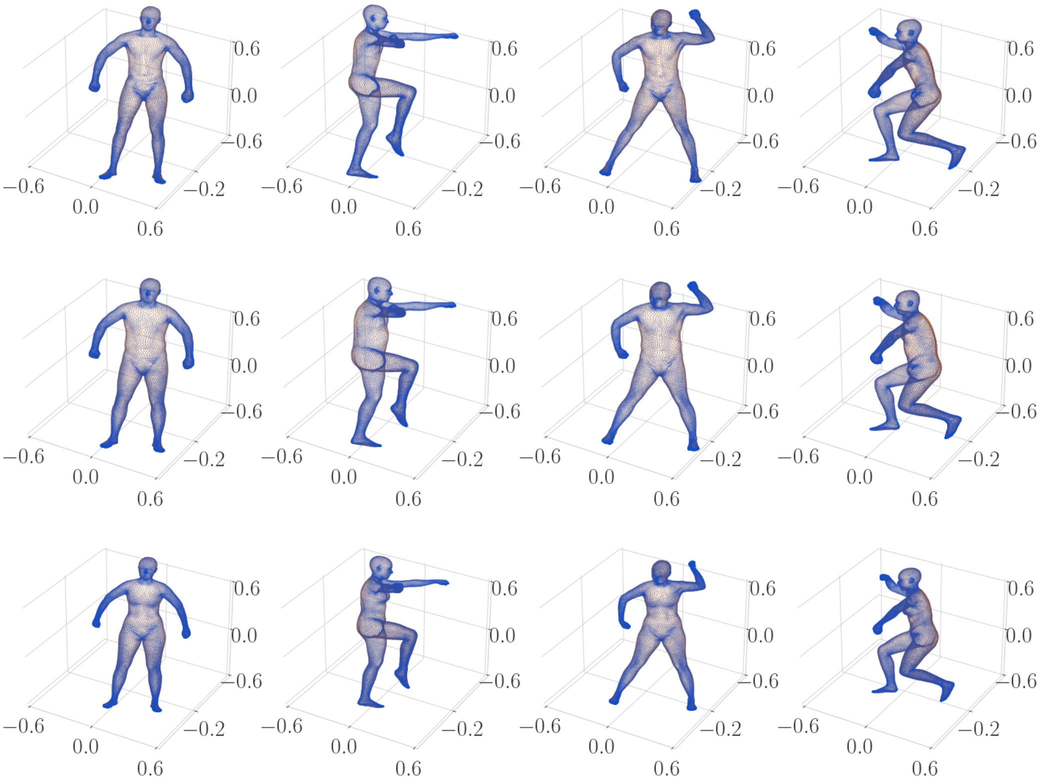


Figure 5.8: Each row displays a different human subject, and each column a different pose of the `SHREC 14` real dataset.

proposed by the `SHREC 2014` track was to perform classification of the 40 classes representing different human subjects, that is to say to distinguish between shapes in different rows in Figure 5.8. In this section, TDA feature vectors are applied to the problem of classifying the 10 different poses, i.e. different columns in Figure 5.8, as they are found to be not informative for the original task. This does not affect the conclusions of these experiments, as the focus here is on comparing the effectiveness of different vectorization methods on the same data.

As in the case of texture images above, first Euler characteristic curves and surfaces were computed for each point cloud. Delaunay-Čech filtrations of points in $\mathbb{R}^3$ were employed in the case of curves. The same filtrations, together with the estimate of the local density given by Equation (3.13) as a second parameter on Delaunay simplices, were used to compute Euler characteristic surfaces.

Again, persistence diagrams were computed for four different point clouds for each element in `SHREC 2014`. In this case, the already mentioned estimate of the local density was used to filter the human 3D shapes. The density thresholds chosen for all point clouds were $\{0.0046, 0.0092, 0.0140, 0.05\}$, which were picked based on the maximum

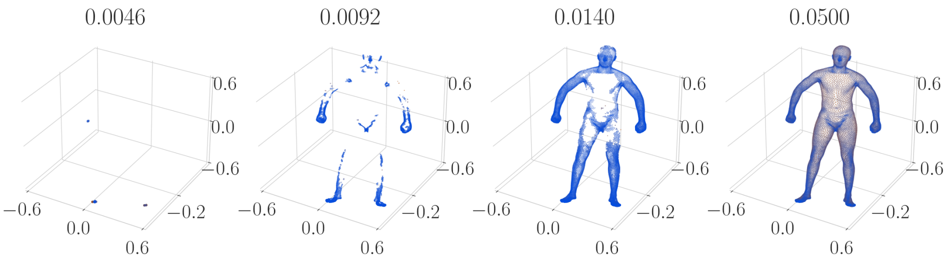


Figure 5.9: Four point clouds derived from one `SHREC 2014` human model.

Table 5.2: Average classification accuracy results `SHREC 2014`

	Features preprocessing	
	-	Min-Max scaler
Euler char. curves - Miniball	$59.09 \pm 4.25\%$	$75.95 \pm 3.62\%$
Euler char. surfaces - (Miniball, Density)	$69.39 \pm 3.61\%$	$67.01 \pm 3.54\%$
Minibox pers. landcapes	$88.98 \pm 2.04\%$	$91.17 \pm 2.83\%$
Minibox cum. landscapes	$88.52 \pm 2.80\%$	$88.11 \pm 3.37\%$
Minibox Fourier coefficients	$85.42 \pm 3.58\%$	$87.99 \pm 2.50\%$
Delaunay-Čech pers. landcapes	$88.41 \pm 2.41\%$	$91.25 \pm 2.31\%$
Delaunay-Čech cum. landscapes	$89.24 \pm 3.16\%$	$90.38 \pm 2.17\%$
Delaunay-Čech Fourier coefficients	$89.66 \pm 3.34\%$	$\mathbf{91.52 \pm 2.47}\%$

average Euler characteristic changes of Delaunay complexes of the filtered point clouds, as done with Euler characteristic of images above. On these collections of four point clouds, persistence diagrams in homological dimensions zero and one were computed as for `OUTEX_TC_00000` images. Furthermore, the same persistence diagrams vectorization methods (concatenating multiple vectors) and logistic regression classifier with ℓ_2 regularization were used.

Finally, the optimization of hyper-parameters (downsampling steps for Euler characteristic vectors, vectorization parameters m and $\bar{k}$, regularization parameter C) was performed on one-third of the data via 3-fold cross-validation. The remaining two-thirds of the data were instead used to compute average accuracy results over 10 repetitions of 3-fold cross-validation. These are given in Table 5.2, including both results with no preprocessing and with a Min-Max scaling step. In this case, Euler characteristic vectors are not as informative as those derived from persistence diagrams. The best result is given by using Fourier coefficients of cumulative landscapes of Delaunay-Čech filtrations with Min-Max scaler preprocessing.

5.5 Discussion

In this chapter the outputs of computational methods presented in this thesis are applied to supervised classification problems. The goal is to evaluate the effectiveness of existing vectorization methods for persistence diagrams versus those presented here. In particular, cumulative landscapes are introduced as an alternative to persistence landscapes and related objects. Moreover, analytical expressions for the Fourier coefficients of cumulative landscapes are given in terms of trigonometric functions, allowing for their direct computation. Finally, different vectorizations of persistence diagrams are applied to two classification problems on open-source datasets. In both cases, the Fourier coefficients perform better than other vectorization methods of persistence diagrams, while being competitive with Euler characteristic surfaces in the case of texture images.

Chapter 6

Optimal Metrics on Genomic Data

The previous chapter presented an application of topological data analysis methods to datasets of two-dimensional images and three-dimensional set of points. Here a classification problem involving high-dimensional data is studied. In particular, the focus is on genomic data of ulcerative colitis patients at risk of developing cancer, consisting of information about duplication and deletion of base pairs in the genome of such patients. This comes in the form of n vectors in $\mathbb{R}^d$ with $n \ll d$. The goal is to introduce a new method that can be used for cancer class prediction and to determine important genes/locations in chromosomes that identify these classes. The idea is to define a weighted metric optimized for a specific type of genomic vectors, and use the derived weighted distances for classification tasks. A dataset of 67 ulcerative colitis patients with low-grade dysplasia (LGD) is used to compare results obtained with this optimized metric against standard machine learning algorithms.

6.1 Related Work

The method described in the following section is related to studies on gene selection and classification of cancer data. For instance in [THNC02] the authors define shrunken centroids of genomic vectors classes, and use these in combinations with nearest-centorid classification [FHT09]. The non-zero values of the shrunken centroids identify the genes useful for selecting the class a patient belongs to. Standard machine learning methods can also be applied to the same problem. For example, the LASSO [FHT09, Section 3.4] machine learning shrinkage method can be used to both classify genomic vectors and select important genes based on the non-zero regularization coefficients it produces.

Moreover, the elastic net was introduced in [ZH05] to overcome the limitations of LASSO when working with high-dimensional data. In the same paper the elastic net was applied to leukemia cancer microarray data, so to classify cancer types and perform automatic gene selection.

6.2 Loss Function and Optimized Metric for Classification

In this section, it is introduced a novel approach to produce distance-based features out of genomic vectors, which can then be used for their classification. In particular, the following setting is considered. Let S be a set of genomic vectors of patients with or at risk of developing cancer, that is to say S is a finite set of n points in $\mathbb{R}^d$, with the dimension d being much greater than n. It is known that the elements of S can be partitioned into subsets S^1 and S^2, corresponding to the low-risk and high-risk patients respectively. Moreover, the elements of S^1 are locally clustered, as their genomic mutations show a reduced variability compared to S^2, which does not have the same property. In the next section, data from ulcerative colitis patients with low-grade dysplasia is used, and the sets S^1 and S^2 are determined by patients either progressing to high-grade dysplasia or not. Thus it is written S^{NP} for the set of non-progressor patients, and S^{P} for the set of progressor patients. The same notation is used in this section as well.

The goal is to define an algorithmic procedure capable of classifying a new and previously unseen genomic vector $p \in \mathbb{R}^d$ as coming from either a low-risk/non-progressor or high-risk/progressor patient. A possible approach is to apply known machine learning algorithms, using directly the elements $p \in S$ as feature vectors. This will be the baseline for the experiments in Section 6.3. Another possible strategy is to compute distances from the centroid of non-progressors S^{NP} and use these as features for classification. In this section an optimized weighted metric d_w is defined, so that the weight function w encodes information about which components of the elements $p \in S$ better discriminate between elements of S^{NP} and S^{P}.

Loss function. Let

$$g_{\mu,\sigma_1}(x) = \frac{1}{\sigma_1\sqrt{2\pi}} \exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{\sigma_1^2}\right) \tag{6.1}$$

be the Gaussian function of mean μ and standard deviation σ_1. Fixed a value of $\sigma_1 \in \mathbb{R}$, a weighted Euclidean distance is defined for each $\mu \in [1,d] \subseteq \mathbb{N}$. Given $p, q \in S$, the (μ,σ_1)-distance

$$d_{\mu,\sigma_1}(p,q) = \sqrt{\sum_{i=1}^{d} g_{\mu,\sigma_1}(i) \cdot (p_i - q_i)^2}, \tag{6.2}$$

weights the contributions of the differences $(p_i - q_i)^2$ based on how close i is to the chosen μ. Then, a loss function $L(\mu)$ is defined by summing up squared weighted Euclidean distances.

$$\begin{aligned} L(\mu) &= -\sum_{p \in S^{\mathrm{P}}} d^2_{\mu,\sigma_1}(c^{\mathrm{NP}}, p) \\ &= -\sum_{p \in S^{\mathrm{P}}} \Big[\sum_{i=1}^{d} g_{\mu,\sigma_1}(i) \cdot (c_i^{\mathrm{NP}} - p_i)^2 \Big] \\ &= -\sum_{p \in S^{\mathrm{P}}} \Big[\sum_{i=1}^{d} \frac{1}{\sigma_1 \sqrt{2\pi}} \exp\left(-\frac{1}{2} \frac{(i-\mu)^2}{\sigma_1^2} \right) \cdot (c_i^{\mathrm{NP}} - p_i)^2 \Big], \end{aligned} \tag{6.3}$$

where $c^{\mathrm{NP}} \in \mathbb{R}^d$ is the centroid of non-progressor vectors; i.e. $c_i^{\mathrm{NP}} = \sum_{p \in S^{\mathrm{NP}}} \frac{p_i}{n^{\mathrm{NP}}}$, with n^{NP} equal to the number of elements in S^{NP}. The idea is that c^{NP} represents well the elements of S^{NP}, which are known to be clustered. So the local minimums of $L(\mu)$ correspond to the values $\mu_j \in [1, d] \subseteq \mathbb{N}$ such that the weighted (μ_j, σ_1)-distances better discriminate between the progressors genomic vectors and the centroid of non-progressors. Moreover, the first derivative of the loss function with respect to μ is

$$\frac{\partial L}{\partial \mu}(\mu) = -\sum_{p \in S^{\mathrm{P}}} \Big[\sum_{i=1}^{d} \frac{(i-\mu)}{\sigma_1^3 \sqrt{2\pi}} \exp\left(-\frac{1}{2} \frac{(i-\mu)^2}{\sigma_1^2} \right) \cdot (c_i^{\mathrm{NP}} - p_i)^2 \Big]. \tag{6.4}$$

Hence the expression of $\frac{\partial L}{\partial \mu}$ can be used to find local minimums of $L(\mu)$ with the gradient descent method [BV04, Section 9.3]. Choosing different starting points for this algorithm in the domain $[1, d]$, it is possible to find all such local minimums $\{\mu_k\}_{k=1}^m$.

Optimized metric. An optimized weight w and metric d_w can are obtained using the loss $L(\mu)$ in Equation (6.3), and its set of local minimums. Let $\bar{\mu}$ be the vector of local minimums sorted by the absolute values of the losses $L(\mu_k)$; i.e. $|L(\bar{\mu}_1)| \geq |L(\bar{\mu}_2)| \geq \ldots \geq |L(\bar{\mu}_m)|$. Fixed an integer $\bar{k} \leq m$ and standard deviation value σ_2, the optimized weight is

$$w(i) = \sum_{k=1}^{\bar{k}} C_k \cdot g_{\mu_k,\sigma_2}(i), \tag{6.5}$$

where $C_k = \frac{|L(\mu_k)|}{\sum_{j=1}^{m} |L(\mu_j)|}$. This is the sum of the Gaussian functions given by the first $\bar{k}$ local minimums in $\bar{\mu}$, weighted by their loss values at the μ_ks. Finally, the optimized metric between two vectors $p, q \in S$ is

$$d_w(p, q) = \sqrt{\sum_{i=1}^{d} w(i) \cdot (p_i - q_i)^2}. \tag{6.6}$$

The distance $d_w(c^{\mathrm{NP}}, p)$ can be used to characterize $p \in S$.

Distance-based classification. The classification method used in combination with the optimized metric d_w is the following.

Let S be partitioned into a train and test subsets S_{train} and S_{test}. The corresponding train-test splits of the non-progressor and progressor vectors are $S^{\mathrm{NP}}_{\text{train}}$, $S^{\mathrm{NP}}_{\text{test}}$ and $S^{\mathrm{P}}_{\text{train}}$, $S^{\mathrm{P}}_{\text{test}}$. Fixed a threshold $t \in \mathbb{R}$, a genomic vector $p \in \mathbb{R}^d$ is classified as non-progressor if $d_w(c^{\mathrm{NP}}_{\text{train}}, p) \leq t$, and as progressor otherwise. Thus, the train and test accuracy functions $f_{\text{train}}(t) : \mathbb{R} \to [0, 1]$ and $f_{\text{test}}(t) : \mathbb{R} \to [0, 1]$ are defined by

$$f_{\text{train}}(t) = \frac{|\{p \in S^{\mathrm{NP}}_{\text{train}} \mid d_w(c^{\mathrm{NP}}_{\text{train}}, p) \leq t\}| + |\{p \in S^{\mathrm{P}}_{\text{train}} \mid d_w(c^{\mathrm{NP}}_{\text{train}}, p) \geq t\}|}{|S_{\text{train}}|}, \tag{6.7}$$

$$f_{\text{test}}(t) = \frac{|\{p \in S^{\mathrm{NP}}_{\text{test}} \mid d_w(c^{\mathrm{NP}}_{\text{test}}, p) \leq t\}| + |\{p \in S^{\mathrm{P}}_{\text{test}} \mid d_w(c^{\mathrm{NP}}_{\text{test}}, p) \geq t\}|}{|S_{\text{test}}|}. \tag{6.8}$$

The threshold $\hat{t}$ maximising the value of $f_{\text{train}}(t)$, i.e. $f_{\text{train}}(\hat{t}) = \max_{t \in \mathbb{R}} f_{\text{train}}(t)$, can be found by iterating on the distances $\{d_w(c^{\mathrm{NP}}_{\text{train}}, p)\}_{p \in S_{\text{train}}}$, because by definition these are the only thresholds at which $f_{\text{train}}(t)$ can increase or decrease its value. In conclusion, the training and test accuracies obtained by using d_w distances as features for classification are $f_{\text{train}}(\hat{t})$ and $f_{\text{test}}(\hat{t})$.

Example: Application to synthetic data. Here, the optimized distance classification method described above is applied to synthetic genomic data. This is a set S of 200 vectors with 4401 components, which are assigned random values on 28 fixed ranges of consecutive components. For instance, p_i is constant for $695 \leq i \leq 783$ and $2262 \leq i \leq 2504$ for each $p \in S$. The first half of the dataset models non-progressor vectors studied in the next section. These 100 vectors have real-valued components randomly sampled in the range $[-0.5, 0.5]$. The second half models progressor vectors. These have component values p_i assigned at random in $[-0.8, 0.8]$ with a higher variance between constant values in different ranges. Moreover, their p_is are decremented by 0.4 for each $695 \leq i \leq 793$, and incremented by 0.3 for each $2262 \leq i \leq 2504$, which are the two already mentioned ranges of constant components. The idea is that these are the regions of the genome that always get altered similarly in case a patient is likely to progress to high-grade dysplasia. Figure 6.1 presents two pairs of synthetic genomic vectors. The first column contains two synthetic "non-progressor" vectors and the second column two synthetic "progressors".

On this data it is possible to compute the value of the loss function for $\mu \in [1, 4401]$, as given by Equation (6.3), and find it local minimums with gradient descent using the

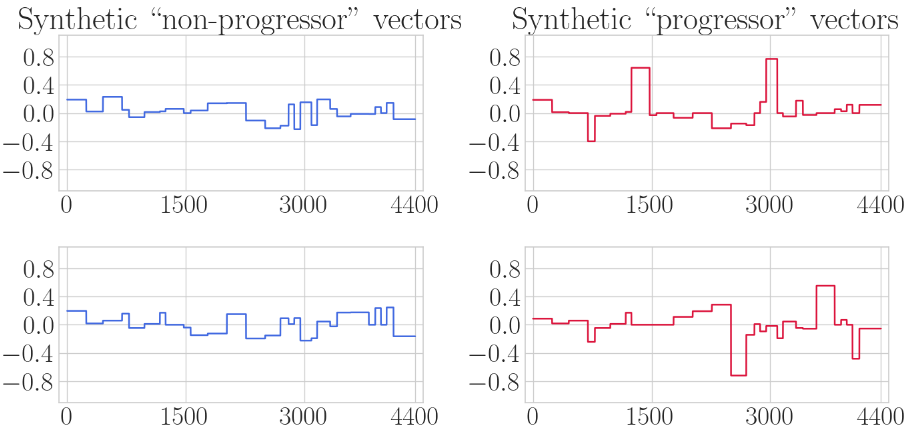


Figure 6.1: Synthetically generated genomic vectors plotted as piecewise constant curves on the range $[1, 4401]$. The column on the left shows two "non-progressor" vectors, and the column on the right two "progressor" vectors.

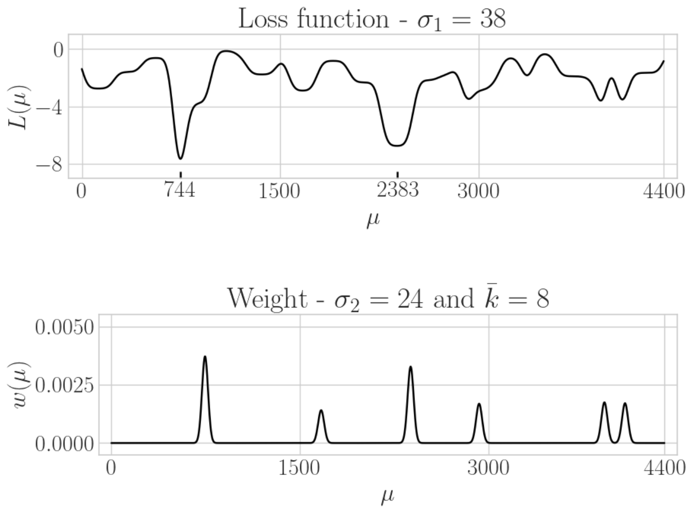


Figure 6.2: Loss function and optimized weight obtained by using 60% of the synthetically generate genomic vectors.

expression in Equation (6.4). For this, it is necessary to choose a standard deviation value σ_1. Then the weight w is determined by fixing the values of σ_2 and $\bar{k}$. The result is a weighted distance d_w, which can be used to classify patients and compute accuracy results with Equations (6.7) and (6.8).

A subset containing 40% of the elements of S is set aside to determine appropriate values of σ_1, σ_2, and $\bar{k}$. On this data, average test accuracy results over 100 train-test splits are used as a score to select a triplet $(\sigma_1, \sigma_2, \bar{k})$. In practice, this is done

Table 6.1: Classification accuracy results of synthetic genomic vectors over 100 train-test splits

	Avg. test accuracy
Distance-based classifier - Euclidean distance d_2	$77.55 \pm 4.09\%$
Distance-based classifier - optimized distance d_w	$\mathbf{93.28 \pm 3.27\%}$

by choosing the parameters maximizing average test accuracy results, searching over all possible combinations of integer standard deviations and number of local minimums, i.e. $(\sigma_1, \sigma_2, \bar{k}) \in \mathbb{N}^3$. This search results into picking $\sigma_1 = 38$, $\sigma_2 = 24$, and $\bar{k} = 6$, which are then used to compute the loss function and an optimized weight on the other 60% of S. Figure 6.2 provides a plot of this loss, whose two main local minimums correspond to the midpoints of the two ranges of components which were incremented and decremented while generating the data. Thus the weighted distance d_w derived from this loss correctly encodes values of μ which on average discriminate between "non-progressors" and "progressors". Lastly, the test accuracy over 100 train-test splits is calculated with Equation (6.8) on the second 60% of the data using d_w. This is compared against the average classification accuracy obtained on the same data with standard Euclidean distances, i.e. using $d_2(c^{\mathrm{NP}}, p)$ as features. Results are in Table 6.1 and show a clear advantage in using weighted distances d_w for this type of synthetically generated data. This example provides proof of concept of the distance-based classification method described above. It illustrates how it can be applied in practice, and it shows that patterns in the data at hand are detected in the local minimums of the loss function $L(\mu)$. In the next section, the same methodology used here is applied to real cancer genomic vectors.

6.3 Application to Low-Grade Dyspalasia Data

The genomic data under consideration in this final section comes from 67 ulcerative colitis patients with low-grade dysplasia [BCC+19, CABB+19], and consists of 269 vectors with 4401 real-valued components, representing the $\log_2$ values of copy number alterations (CNA)[1] in patients chromosomes. These vectors are the output of low-pass whole-genome sequencing of tissue samples obtained at different time-points from the patients. Due to their condition, the 67 patients are considered at risk of developing colorectal cancer (CRC). Moreover, it is known that 45 of them did not progress to high-grade dysplasia or cancer within five years, while the other 22 did. The goal is to be able to predict which patients are progressors based on the genomic vectors. Besides, it would also be important to identify which chromosomes regions (i.e. vectors components p_i) are important to distinguish between progressor and non-progressor patients.

[1] Measuring the amount of additional or missing genetic material found in chromosomes.

Table 6.2: Classification accuracy results of real-world genomic vectors over 100 train-test splits

	Avg. test accuracy
Nearest Neighbours classifier	$75.43 \pm 7.52\%$
Logistic regression - ℓ_2 regularization	$75.62 \pm 6.71\%$
Distance-based classifier - Euclidean distance d_2	$82.67 \pm 7.18\%$
Distance-based classifier - optimized distance d_w	$\mathbf{84.24 \pm 7.02}\%$

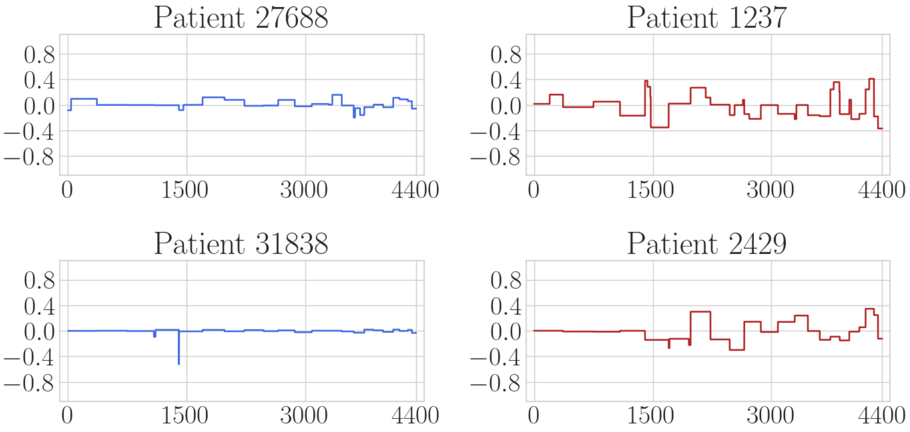


Figure 6.3: Genomic vectors of $\log_2$ CNA values of ulcerative colitis patients with low-grade dysplasia, plotted as piecewise constant curves on the range $[1, 4401]$. The columns on the left shows two vectors corresponding to non-progressor patients, and the column on the right two vectors corresponding to progressor patients.

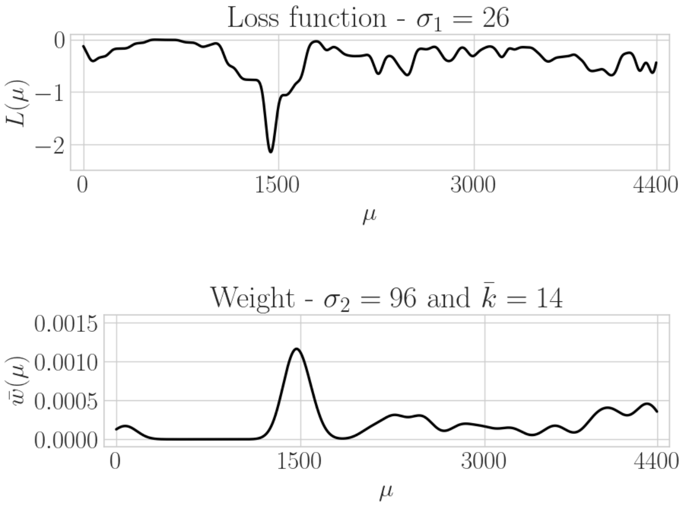


Figure 6.4: Loss function and optimized weight obtained by using 60% of the genomic vectors of $\log_2$ CNA values.

The optimized distance classification method of the previous section is applied, selecting a single vector per patient. This choice is made by keeping only the vector with the maximum norm, between the oldest ones of a patient, to use information from early genetic mutations. Four such vectors are shown in Figure 6.3 as piecewise constant curves. The column on the left contains data of two non-progressor patients and the column on the right of two progressor patients. Then, as in the example using synthetic genomic vectors, 40% of the data is set aside to determine the values of σ_1, σ_2, and $\bar{k}$ as those maximizing average test accuracy over 100 train-test splits. In this case the chosen parameters are $\sigma_1 = 26$, $\sigma_2 = 96$, and $\bar{k} = 14$. Given the second 60% of data, this results in the loss function $L(\mu)$ and optimized weight $w(\mu)$ plotted in Figure 6.4. Thus, a total of $\bar{k} = 14$ local minimums of the loss are identified as the locations $\{\mu_k\}_{k=1}^{\bar{k}}$ most useful in characterizing the differences between genomic vectors of low-risk and high-risk patients. In particular, the minimum found at $\hat{\mu} = 1438$ identifies that location as the most important for the binary classification task under study.

Finally, the distance-based classifier introduced in the previous section is applied. Both the Euclidean distance d_2 and the optimized distance d_w are used to test the effect of the weight $w(\mu)$ on classification results. As additional baselines to compare against, a nearest neighbours classifier and a logistic regression classifier with ℓ_2 regularization [FHT09] are employed on the second 60% of genomic vectors as well. For both of these, the `scikit-learn` [PVG+11] Python package implementations are used (choosing the `LIBLINEAR` [FCH+08] solver for logistic regression). Furthermore, the 40% of data which is set aside to determine the values of σ_1, σ_2, and $\bar{k}$, is also used to pick their hyperparameters: the number of nearest neighbours to use, and the inverse regularization strength parameter C. Average test accuracy results over 100 train-test splits are in Table 6.2. The distance-based classifiers produce the best results, with optimized distances improving over Euclidean distances on average.

6.4 Discussion

A distance-based method for the classification of high-dimensional genomic vectors is presented. This makes use of a loss and derived optimized weight functions, which allow identifying coordinates of the given genomic vectors useful for distinguishing between different classes of patients. Its efficacy in terms of average test accuracy results is shown both on synthetic and real-world data. On the latter, distance-based classification outperforms standard machine learning algorithms. Besides, it provides easily interpretable information regarding the coordinates of genomic vectors (that can be related to genes/chromosomes locations), which are the most informative in the classification problem at hand.

Chapter 7

Conclusion

The general concept underlying the field of topological data analysis, and in particular the theory of persistent homology, is that the geometric shape and topological structure of data can be used for its analysis. This approach is also at the basis of the computational methods presented in this thesis. The goal was to describe methods extending the range of tools available in this context, overcoming some of the existing limitations of persistent homology, and topological data analysis in general.

For instance, extending persistent homology to multidimensional parameter spaces is problematic due to the absence of a complete discrete invariant in this setting [CZ09]. In Chapter 3, we propose the use of Euler characteristic numbers, instead of ranks of persistent homology groups, to characterize bi-filtrations of complexes. This way it is possible to obtain a well-defined and compact representation of the topological information of a given bi-filtration. The idea is to generalize Euler characteristic curves and the algorithms for their computation. This results in matrices of numbers, Euler characteristic surfaces, which can be used to obtain insights about a two-dimensional parameter space. Notably, we provide novel algorithms for the computation of Euler characteristic surfaces of image and point data, the complexities of which are given in Proposition 3.4.1 and Proposition 3.5.1. A possible development of this research could be the generalization of our algorithms to higher-dimensional parameter spaces. Furthermore, it would be interesting to study which combinations of parameters are most effective in characterizing different types of data.

Another issue with the application of persistent homology is the complexity of algorithms for the computation of persistence diagrams. This problem is partially solved using Alpha filtrations, which reduce the number of simplices in Čech filtrations as seen in Section 2.4. However, these apply only to the case of points in Euclidean metric space. Thus, we try to obtain similar results in other metric spaces. In particular, we study the problem of defining alternative filtrations for the computation of Čech persistence diagrams of points in ℓ_∞ metric space in Chapter 4. Alpha flag and Minibox filtrations are introduced, which can be used for this task in homological dimensions zero and one. The main original results discussed in this chapter are Theorem 4.3.6 and Theorem 4.4.4, which are used to prove the equivalence of Alpha flag, Minibox, and Čech complexes. In addition to this, algorithms are described for finding the Minibox edges of a set of points, which is the only information needed to build Minibox filtrations. The complexities of these algorithms are given in Proposition 4.5.1 and Proposition 4.5.2. Furthermore, Proposition 4.4.7 shows that, for randomly sampled points, the expected number of simplices of Minibox filtrations is lower than the one of Čech filtrations. Thus, Minibox complexes can be seen as a tool for speeding up the computation of persistence diagrams. Future work could focus on improving the complexity of Minibox edges algorithms for points in high-dimensions. Moreover, the geometric property characterizing ℓ_∞-Delaunay edges (Proposition 4.1.3) could be applied to obtain efficient algorithms for finding these in ambient dimension three or higher. Besides, it may be possible to define other filtrations, with the same properties of the Alpha flag and Minibox ones, further reducing the expected number of simplices that need to be considered to compute persistence diagrams.

The application of topological data analysis invariants to supervised classification problems has recently received an increasing level of attention, with the introduction of several methods for discretizing the information of persistence diagrams [Bub15, AEK$^+$17, OPT$^+$17]. The goal is to map persistence diagrams into vectors that can be given as inputs to machine learning algorithms. In this context, a new type of summary function of persistence diagrams is introduced in Chapter 5, which we call cumulative landscape. Moreover, we derive analytical expressions of the Fourier coefficients of cumulative landscapes, which are given in Equations (5.10) and (5.11). These Fourier coefficients can then be used to produce feature vectors out of cumulative landscapes. Experiments on real-world data show that, compared to those obtained with persistence landscapes, the above-mentioned feature vectors can improve classification accuracy results. A possible future direction of work is to extend the approach involving Fourier coefficients. The sine and cosine functions used in the Fourier series form an orthogonal set, which is a basis for periodic cumulative landscapes. Alternative bases of function could be used to decompose cumulative landscapes, so as to employ the derived coefficients as features in classification tasks.

In case the data that needs to be analyzed consists of high-dimensional arrays of values, the methods discussed in the first part of this thesis do not directly apply. In

Chapter 6, we consider a classification problem of cancer genomic data, and describe a technique to define optimized metrics on it. These are used to classify the data with a distance-based classifier, which outperforms nearest neighbours and logistic regression classifiers. Thus, by only making use of metric information, we can describe a method improving over standard machine learning algorithms for a particular type of problem. Further research could focus on extending our method, which employs a loss function and optimized distance, to settings where more than two classes of genomic vectors are given. Moreover, the effectiveness of this approach on data of patients affected by different types of cancer than the one used in this thesis would be worth exploring.

References

[AEK$^+$17] Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. Persistence Images: A Stable Vector Representation of Persistent Homology. *The Journal of Machine Learning Research*, 18(1):218–252, 2017.

[AKL13] Franz Aurenhammer, Rolf Klein, and Der-Tsai Lee. *Voronoi Diagrams and Delaunay Triangulations.* World Scientific Publishing, Singapore, 2013.

[Arm13] Mark A. Armstrong. *Basic Topology.* Springer, New York, 2013.

[BAG$^+$21] Gabriele Beltramo, Rayna Andreeva, Ylenia Giarratano, Miguel O. Bernabeu, Rik Sarkar, and Primoz Skraba. Euler Characteristic Surfaces. *arXiv:2102.08260*, 2021.

[Bau19] Ulrich Bauer. Ripser: Efficient Computation of Vietoris-Rips Persistence Barcodes. *arXiv: 1908.02518*, 2019.

[BCC$^+$19] Ann-Marie Baker, William Cross, Kit Curtius, Ibrahim Al Bakir, Chang-Ho Ryan Choi, Hayley Louise Davis, Daniel Temko, Sujata Biswas, Pierre Martinez, Marc J. Williams, et al. Evolutionary History of Human Colitis-Associated Colorectal Cancer. *Gut*, 68(6):985–995, 2019.

[BE17] Ulrich Bauer and Herbert Edelsbrunner. The Morse Theory of Čech and Delaunay Complexes. *Transactions of the American Mathematical Society*, 369(5):3741–3762, 2017.

[BGK15] Subhrajit Bhattacharya, Robert Ghrist, and Vijay Kumar. Persistent Homology for Path Planning in Uncertain Environments. *IEEE Transactions on Robotics*, 31(3):578–590, 2015.

[BKR14a] Ulrich Bauer, Michael Kerber, and Jan Reininghaus. Clear and Compress: Computing Persistent Homology in Chunks. In *Topological Methods in Data Analysis and Visualization III*, pages 103–117. Springer, Cham, 2014.

[BKR14b] Ulrich Bauer, Michael Kerber, and Jan Reininghaus. Distributed Computation of Persistent Homology. In *Proceedings of the 16th Workshop on Algorithm Engineering and Experiments*, pages 31–38, 2014.

[BMM$^+$16] Paul Bendich, James S Marron, Ezra Miller, Alex Pieloch, and Sean Skwerer. Persistent Homology Analysis of Brain Artery Trees. *The Annals of Applied Statistics*, 10(1):198–218, 2016.

[BP20] Jean-Daniel Boissonnat and Siddharth Pritam. Edge Collapse and Persistence of Flag Complexes. In *Proceedings of the 36th International Symposium on Computational Geometry*, pages 19:1–19:15, 2020.

[BS21] Gabriele Beltramo and Primoz Skraba. Persistent Homology in ℓ_∞ Metric. *arXiv: 2008.02071*, 2021.

[Bub15] Peter Bubenik. Statistical Topological Data Analysis using Persistence Landscapes. *The Journal of Machine Learning Research*, 16(1):77–102, 2015.

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization.* Cambridge University Press, Cambridge, 2004.

[CABB$^+$19] Kit Curtius, Ibrahim Al Bakir, Ann-Marie Baker, Theo Clarke, Nadia Nasreddin, Maja Kopczynska, Meghan Agnew, Kane Smith, Morgan Moorghen, Manuel Rodriguez-Justo, et al. Quantifying Evolution of Early Dysplastic Lesions in Ulcerative Colitis Predicts Future Colorectal Cancer Risk. *Gastroenterology*, 156(6):S162–S163, 2019.

[Car09] Gunnar Carlsson. Topology and Data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.

[CDSO14] Frédéric Chazal, Vin De Silva, and Steve Oudot. Persistence Stability for Geometric Complexes. *Geometriae Dedicata*, 173(1):193–214, 2014.

[CH13] Corrie J. Carstens and Kathy J. Horadam. Persistent Homology of Collaboration Networks. *Mathematical Problems in Engineering*, ID 815035, 2013.

[CJS19] Francisco Criado, Michael Joswig, and Francisco Santos. Tropical Bisectors and Voronoi Diagrams. *arXiv: 1906.10950*, 2019.

[CK11] Chao Chen and Michael Kerber. Persistent Homology Computation with a Twist. In *Proceedings of the 27th European Workshop on Computational Geometry*, pages 192–200, 2011.

[CKR17] Aruni Choudhary, Michael Kerber, and Sharath Raghvendra. Improved Approximate Rips Filtrations with Shifted Integer Lattices. In *Proceedings of the 25th Annual European Symposium on Algorithms*, pages 28:1–28:13, 2017.

[COO15] Mathieu Carrière, Steve Y. Oudot, and Maks Ovsjanikov. Stable Topological Signatures for Points on 3D Shapes. In *Proceedings of the 13th Eurographics Symposium on Geometry Processing*, pages 1–12, 2015.

[CP10] Timothy M. Chan and Mihai Pătraşcu. Counting Inversions, Offline Orthogonal Range Counting, and Related Problems. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 161–173, 2010.

[CSEH07] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of Persistence Diagrams. *Discrete & Computational Geometry*, 37(1):103–120, 2007.

[CW90] Don Coppersmith and Shmuel Winograd. Matrix Multiplication via Arithmetic Progressions. *Journal of Symbolic Computation*, 9(3):251–280, 1990.

[CZ09] Gunnar Carlsson and Afra Zomorodian. The Theory of Multidimensional Persistence. *Discrete & Computational Geometry*, 42(1):71–93, 2009.

[dBCvKO08] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational Geometry: Algorithms and Applications.* Springer, Berlin, 2008.

[DFL03] Michele D'Amico, Patrizio Frosini, and Claudia Landi. Optimal Matching between Reduced Size Functions. *Università degli Studi di Modena e Reggio Emilia, Italy*, Technical report 35, 2003.

[DGK63] Ludwig Danzer, Branko Grunbaum, and Victor Klee. Helly's Theorem and its Relatives. In *Proceedings of Symposia in Pure Mathematics*, pages 101–180, 1963.

[DHL$^+$16] Pawel Dłotko, Kathryn Hess, Ran Levi, Max Nolte, Michael Reimann, Martina Scolamiero, Katharine Turner, Eilif Muller, and Henry Markram. Topological Analysis of the Connectome of Digital Reconstructions of Neural Microcircuits. *arXiv: 1601.01580*, 2016.

[DSG07] Vin De Silva and Robert Ghrist. Coverage in Sensor Networks via Persistent Homology. *Algebraic & Geometric Topology*, 7(1):339–358, 2007.

[DSMVJ11] Vin De Silva, Dmitriy Morozov, and Mikael Vejdemo-Johansson. Dualities in Persistent (Co)Homology. *arXiv: 1107.5665*, 2011.

[EH10] Herbert Edelsbrunner and John Harer. *Computational Topology: an Introduction.* American Mathematical Society, Providence, 2010.

[EKS83] Herbert Edelsbrunner, David Kirkpatrick, and Raimund Seidel. On the Shape of a Set of Points in the Plane. *IEEE Transactions on information theory*, 29(4):551–559, 1983.

[ELZ02] Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological Persistence and Simplification. *Discrete & Computational Geometry*, 28(4):511–533, 2002.

[EM94] Herbert Edelsbrunner and Ernst P. Mücke. Three-Dimensional Alpha Shapes. *ACM Transactions on Graphics*, 13(1):43–72, 1994.

[FCH$^+$08] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[FHT09] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer, New York, 2009.

[Gab72] Peter Gabriel. Unzerlegbare Darstellungen I. *Manuscripta mathematica*, 6(1):71–103, 1972.

[Gär99] Bernd Gärtner. Fast and Robust Smallest Enclosing Balls. In *Proceedings of the 7th Annual European Symposium on Algorithms*, pages 325–338, 1999.

[GGL95] Ronald L. Graham, Martin Grötschel, and László Lovász. *Handbook of Combinatorics, Volume II.* The MIT Press, Cambridge, 1995.

[GUD21] GUDHI Project. *GUDHI User and Reference Manual.* GUDHI Editorial Board, 3.4.1 edition, 2021.

[GW17] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing.* Pearson, New York, 2017.

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

[HB08] Samuel Hornus and Jean-Daniel Boissonnat. An Efficient Implementation of Delaunay Triangulations in Medium Dimensions. Research report 6743, INRIA, 2008.

[HKMY20] Marius Hofert, Ivan Kojadinovic, Martin Maechler, and Jun Yan. *copula: Multivariate Dependence with Copulas.* R package version 1.0-1, 2020.

[HKS15] Dan Halperin, Michael Kerber, and Doron Shaharabani. The Offset Filtration of Convex Objects. In *Proceedings of the 23rd Annual European Symposium on Algorithms*, pages 705–716, 2015.

[HW17] Teresa Heiss and Hubert Wagner. Streaming Algorithm for Euler Characteristic Curves of Multidimensional Images. In *Proceedings of the 17th International Conference on Computer Analysis of Images and Patterns*, pages 397–409, 2017.

[KB13] Dirk P. Kroese and Zdravko I. Botev. Spatial Process Generation. *arXiv: 1308.0399*, 2013.

[KB16] Adrian Kaehler and Gary Bradski. *Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library.* O'Reilly Media, Sebastopol, 2016.

[Kle86] Rolf Klein. Direct Dominance of Points. *International Journal of Computer Mathematics*, 19(3-4):225–244, 1986.

[LBBH98] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[LW15] Michael Lesnick and Matthew Wright. Interactive Visualization of 2-D Persistence Modules. *arXiv: 1512.00180*, 2015.

[MMS11] Nikola Milosavljević, Dmitriy Morozov, and Primoz Skraba. Zigzag Persistent Homology in Matrix Multiplication Time. In *Proceedings of the 27th Annual Symposium on Computational Geometry*, pages 216–225, 2011.

[MN13] Konstantin Mischaikow and Vidit Nanda. Morse Theory for Filtrations and Efficient Computation of Persistent Homology. *Discrete & Computational Geometry*, 50(2):330–353, 2013.

[Mun00] James R. Munkres. *Topology.* Prentice Hall, Upper Saddle River, 2000.

[Nel06] Roger B. Nelsen. *An Introduction to Copulas.* Springer, New York, 2006.

[OMP$^+$02] T. Ojala, T. Mäenpää, M. Pietikäinen, J. Viertola, J. Kyllönen, and S. Huovinen. Outex-New Framework for Empirical Evaluation of Texture Analysis Algorithms. In *Proceedings of the 16th International Conference* *on Pattern Recognition*, pages 701–706, 2002.

[OPT$^+$17] Nina Otter, Mason A. Porter, Ulrike Tillmann, Peter Grindrod, and Heather Harrington. A Roadmap for the Computation of Persistent Homology. *EPJ Data Science*, 6(17):1–38, 2017.

[Oud15] Steve Y. Oudot. *Persistence Theory: from Quiver Representations to Data Analysis.* American Mathematical Society, Providence, 2015.

[PSR$^+$14] D. Pickup, X. Sun, P. L. Rosin, R. R. Martin, Z. Cheng, Z. Lian, M. Aono, A. Ben Hamza, A. Bronstein, M. Bronstein, S. Bu, U. Castellani, S. Cheng, V. Garro, A. Giachetti, A. Godil, J. Han, H. Johan, L. Lai, B. Li, C. Li, H. Li, R. Litman, X. Liu, Z. Liu, Y. Lu, A. Tatsuma, and J. Ye. SCHREC'14 Track: Shape Retrieval of Non-Rigid 3D Human Models. In *Proceedings of the 7th Eurographics Workshop on 3D Object Retrieval*, pages 1–10, 2014.

[PVG$^+$11] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[RCIU19] Martin Royer, Frédéric Chazal, Yuichi Ike, and Yuhei Umeda. ATOL: Automatic Topologically-Oriented Learning. *arXiv: 1909.13472v1*, 2019.

[Rob99] Vanessa Robins. Towards Computing Homology from Finite Approximations. *Topology Proceedings*, 24:503–532, 1999.

[SDT91] Gary M. Shute, Linda L. Deneen, and Clark D. Thomborson. An $O(n\ log\ n)$ Plane-Sweep Algorithm for L_1 and L_∞ Delaunay Triangulations. *Algorithmica*, 6:207–221, 1991.

[Spa12] Edwin H. Spanier. *Algebraic Topology.* Springer, New York, 2012.

[THNC02] Robert Tibshirani, Trevor Hastie, Balasubramanian Narasimhan, and Gilbert Chu. Diagnosis of Multiple Cancer Types by Shrunken Centroids of Gene Expression. *Proceedings of the National Academy of Sciences (PNAS)*, 99(10):6567–6572, 2002.

[Tol76] Georgi P. Tolstov. *Fourier Series.* Dover Publications, New York, 1976.

[TSBO18] Christopher Tralie, Nathaniel Saul, and Rann Bar-On. Ripser.py: A Lean Persistent Homology Library for Python. *The Journal of Open Source Software*, 3(29):925, 2018.

[WCV12] Hubert Wagner, Chao Chen, and Erald Vuçini. Efficient Computation of Persistent Homology for Cubical Data. In *Topological Methods in Data Analysis and Visualization II*, pages 91–106. Springer, Berlin, 2012.

[Yan07] Jun Yan. Enjoy the Joy of Copulas: With a Package copula. *Journal of Statistical Software*, 21(4):1–21, 2007.

[ZH05] Hui Zou and Trevor Hastie. Regularization and Variable Selection via the

Elastic Net. *Journal of the Royal Statistical Society: Series B*, 67(2):301–320, 2005.