Extending persistent barcodes with information captured by persistent Laplacians

Joris Kirchner
Supervisor: Martina Vittorietti

Date: August 12, 2025

Image source: Elesey/Shutterstock

Abstract

Over the last decade, an increasing amount of data has become available for data analysts to understand. Datasets containing books, images, networks, or other types of data have been studied. A recent group of methods proposes to analyze samples in datasets based on a description of their shape. This group of methods is often referred to as Topological Data Analysis (TDA). In this thesis, an extension to the most commonly used TDA method, called Persistent Homology (PH), is proposed. PH only describes topological features, while this extension additionally allows for the description of geometric properties. The new information is obtained via the persistent Laplacian, a recently proposed operator that encodes the topological information of persistent homology in its kernel and geometric information in its non-zero spectrum. The persistent Laplacian contains a lot of information and extracting the relevant parts has not yet been standardized. In this thesis, a new operator, the persistent multiplicity operator, is proposed. The new operator summarizes the information of the persistent Laplacian such that it can easily be extracted and used to extend PH. This allows the many previously studied methods based on persistent homology to additionally describe geometric properties, as opposed to only topological features. For the multiplicity operator, the trace is analyzed and the features captured by it are discussed. Besides analyzing the sum of the eigenvalues, it is argued that individual eigenvalues could contain more information. However, these are deemed hard to understand. Therefore, an adjusted multiplicity operator is proposed that contains separately interpretable eigenvalues. Finally, the operators are used to classify handwritten digits from the MNIST dataset and to make statistical tests that can detect different generation processes of artificially made cross sections of crystalline structures.

Contents

1 Introduction

With the rise of Machine Learning and Artificial Intelligence in recent years, more and more data across various fields and applications has become available. Numerous methods and models have been proposed to interpret this data, to make predictions with the data and to visualize the data. These range from older methods such as linear regression and logistic regression to newer neural networks. These methods often focus on predicting a distribution in some high-dimensional space, often called feature space.

Instead of directly predicting the distribution in this feature space, one could look at first describing its properties. For example, in image data, often all pixel values of an image are used to create a very high dimensional representation of the image. However, only certain shapes within the image are often of importance.

This brings the category of methods from the field of Topological Data Analysis (TDA). These methods try to combine the abstract field of topology with the field of data science. Topology looks at shapes in abstract spaces and provides ways of describing these shapes. TDA methods convert data to such a shape, allowing the usage of the shape descriptions in methods such as linear regression [21]. Describing the shapes is done in terms of topological invariants, properties of the shape that remain intact under deformations like stretching, twisting or bending. These invariants include, number of connected components, number of loops and the number of voids.

The most commonly used method in TDA is called Persistent Homology (PH) [11]. It aims to describe samples in a dataset, by creating a continuously changing topological shape and analyzing its properties during this change. The result will be a set of intervals, referred to as a persistent barcode, that describe when certain properties appear and disappear. These barcodes can be analyzed using statistical methods to understand the differences between the samples.

PH has been successfully applied in many different fields [23]. Examples can be found in Oncology and the study of tumor behavior [8, 48], COVID-19 identification and mutation detection [6, 28], quantification of bone microstructure [38], protein engineering and folding [27, 39, 47], granular crystallization [41], text classification [22], the study of the cosmic microwave background temperature [37] etc. Two applications are highlighted.

The first of which is the study of handwritten number recognition. It aims to detect the number written in images of handwritten numbers. Examples of such images are shown in Figure 1.1. Describing these images with persistent homology, one could obtain a changing topological shape by scanning the images from top to bottom. The 4 in the figure would start with two connected components, while the 6 and 9 both start with only one, therefore a first distinction could immediately be found in terms of the topological invariants. Furthermore the 6 and the 9 could be differentiated by the moment the loop is fully scanned. When scanning from top to bottom, the 9 finishes its loop first, making it different in terms of the invariants compared to the 6. In [21] the authors apply the method to the MNIST dataset [2]. They conclude that the method can reduce the dimensionality of the data, while retaining a similar accuracy on the test set.

The second application is the classification of crystalline structures of alloys, see Figure 1.2a. The focus will be on the red lines in the figure, representing Kernel Average Misorientation (KAM). It is often

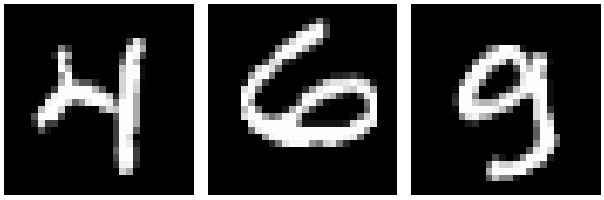
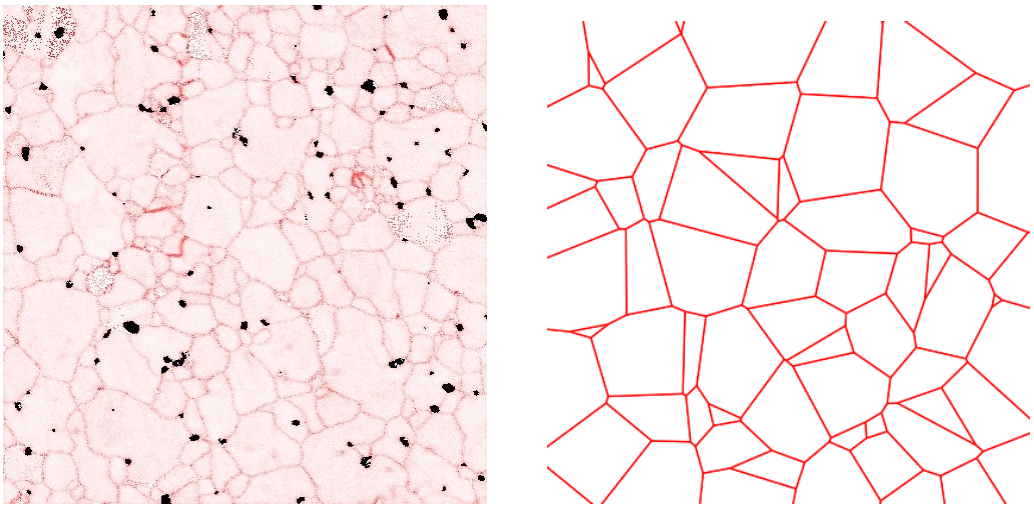


Figure 1.1: Examples of handwritten numbers from the MNIST dataset [2].



(a) Microstructure image showing the Kernel Average Misorientation (KAM) in red, black dots are carbides.

(b) Artificially generated cross section of a three dimensional Poisson Voronoi diagram.

Figure 1.2: Real microstructure image on the left and artificially generated cross section on the right.

believed that these lines as well as the carbides, contain information that describes physical properties of the material. Therefore, understanding these images may help the development of new alloys.

Relating the material properties to these structures would require many of these images, which would be time-consuming to obtain. Instead, it would be useful to artificially generate samples of these types of images. [43] propose a new test statistic, based on persistent homology to test whether a microstructure can be modeled as an artificially generated structure, see Figure 1.2b.

For this thesis, only the artificially generated cross sections are analyzed, where three different types of generation processes are discussed to see if they can be recognized by the TDA methods. The ideas behind the tests proposed in [43] can again be used. This would allow the application of persistent homology and therefore the analysis of topological features in the images.

Nevertheless, persistent homology does have some limitations. Because it only uses topological invariants, geometric information which can be affected by stretching, twisting or bending is not captured. For example, the authors of [21] note that for handwritten number recognition, their method would misclassify a 7 as a 3 if a horizontal center bar is added to the 7 or misclassify a 9 as a 4 if the top loop of the 9 is not fully closed. A 9 with an open loop can be bent into the 4 seen in Figure 1.1, therefore the topological invariants would be equal. Hence, analyzing these invariants does not allow us to differentiate between these two numbers and geometric information would be needed to do so.

An operator that captures the topological invariants, as well as some geometric properties, is the combinatorial Laplacian [24]. Originally defined as a generalization of the graph Laplacian, it encodes the topological features in the 0-eigenspace, while geometric properties can be found in the non-zero spectrum. Even though this operator has seen some success in certain applications [3, 20, 44], it does not follow the PH pipeline, as it only looks at a single shape and not a continuously changing one.

In order to track the changes in the combinatorial Laplacian over such a changing shape, persistent Laplacians were recently proposed [44]. It has been shown that this new operator contains all the information obtained from standard persistent homology, together with some extra information [33]. This extra information is often thought to describe geometric features, however little is known about its specific meaning [46]. Nevertheless, persistent Laplacians have already been used in multiple fields.

Successes have been noted in handwritten number recognition on the MNIST dataset [16], protein thermal stability [44], protein-ligand binding [34], and COVID-19 strain projections [13].

Another advantage of using the persistent Laplacian arises from the analysis of crystalline structures. In PH, all parts of the shape are of the same type. However, in Figure 1.2 carbides are also visible as black dots and are thought to affect the properties of the material. Including this information would require a distinction to be made between different points. Standard persistent homology is unable to make this distinction, instead weighted persistence would be needed to do so [40]. Including weighted persistence in the standard algorithm to compute persistent homology is not trivial. However, the persistent Laplacian can easily be adapted to include it.

In this thesis, the full spectrum of the persistent Laplacian is analyzed. This yields the introduction of a new operator, the persistent multiplicity operator, that summarizes the information captured by the persistent Laplacian. For many applications, such as machine learning, having more condensed information, as opposed to a lot of duplicate information, often yields better results [31, 30, 15]. It is therefore argued that this operator can yield a better performance in the previously discussed applications of the persistent Laplacian, as well as potential new ones. Furthermore, information from this operator can easily be extracted in a form similar to how persistent homology encodes information. This allows us to describe the operator as an enhancement of PH, which can make it more intuitive for someone unfamiliar with Laplacians. Moreover, it allows existing theory based on PH to describe additional features captured by the persistent Laplacian. To demonstrate the capabilities of this new operator, it is applied to the study of handwritten number recognition and the classification of crystalline structures.

In Chapter 2, the formal definitions of all the concepts used in the thesis are given. Having defined the persistent Laplacian, Chapter 3 starts by introducing a new way of visualizing the operator showing the duplicate information that seems to be present, giving rise to the new multiplicity operator. The next two sections in the chapter analyze the new operator algebraically, thereby showing some of the properties it captures. Afterwards a section discusses how to use the operator in practice, which also shows that existing theory can easily be adapted to include the new information. The chapter is concluded with a proposed algorithm to compute the new operator. Finally, the persistent multiplicity operator and its corresponding method are applied to the MNIST dataset [2], containing handwritten numbers, and some computer generated crystalline structures in Chapter 4.

Code used in the thesis can be found on `https://github.com/siroj99/Master_thesis`. To write the code, Microsoft Copilot was used. This is an Artificial Intelligence based system that generates lines of code or parts of lines based on the already written code. The user can then accept these lines, which means that some of the code was written with AI. Furthermore, Google Gemini was used to do a spelling and grammar check on the thesis. This was done by uploading the document and asking: ”Could you correct all spelling and grammar errors in the document? Just tell me where the mistakes are and how to correct them. Do not rewrite the entire document.”. It was therefore only used to highlight the sentences that contained errors. Changing the sentences was done by hand, one at a time and not all its suggestions were accepted. Besides these two usages, no further Artificial Intelligence was used to write the thesis.

2

Mathematical Background

The purpose of this chapter is to standardize the notation used in the thesis. The chapter is divided into five sections. The first section summarizes some abstract linear algebra concepts like the Moore-Penrose pseudo inverse, which are needed later on in the thesis. The second and third sections define some concepts used for the applications. Section 2.2 introduces Mapper, an algorithm that takes a point cloud and outputs a graph and Section 2.3 introduces Voronoi diagrams and ways of artificially generating crystalline structures. The fourth Section introduces homology and the field of persistent homology. Finally the chapter is concluded with a section on graph-, combinatorial- and persistent Laplacians. It describes how combinatorial Laplacians are commonly visualized and used. Furthermore, for persistent Laplacians, a short discussion on the computational complexity is noted.

2.1. Preliminaries

In this thesis, a few abstract concepts from the field of linear algebra are used. First, we define the Moore-Penrose pseudo inverse for real matrices. In contrast to the normal inverse, it is also defined for non-square or singular matrices. While it is a relatively well known operator, it may be useful to define it formally.

Definition 2.1.1 (Moore-Penrose pseudo inverse, [26])**.** The Moore-Penrose pseudo inverse of a matrix $A \in \mathbb{R}^{m\times n}$ is the $n \times m$ matrix $A^\dagger$, which satisfies:

$$AA^\dagger A = A \tag{2.1}$$
$$A^\dagger AA^\dagger = A^\dagger \tag{2.2}$$
$$(AA^\dagger)^T = AA^\dagger \tag{2.3}$$
$$(A^\dagger A)^T = A^\dagger A. \tag{2.4}$$

Often an alternative definition using the singular value decomposition is used that is equivalent to the previously discussed definition. Here, it is formulated as a theorem.

Theorem 2.1.1 ([25])**.** *For a matrix $A \in \mathbb{R}^{m\times n}$ with singular value decomposition $A = U\Sigma V^T$, where $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are both orthogonal matrices and $\Sigma \in \mathbb{R}^{m\times n}$ a diagonal matrix, the Moore-Penrose pseudo inverse $A^\dagger$ can be defined as $A^\dagger = V\Sigma^\dagger U^T$, where $\Sigma^\dagger \in \mathbb{R}^{n\times m}$ is a diagonal matrix, with*

$$(\Sigma^\dagger)_{i,i} = \begin{cases} \frac{1}{\Sigma_{i,i}} & \text{if } \Sigma_{i,i} \neq 0. \\ 0 & \text{otherwise.} \end{cases} \tag{2.5}$$

Besides these, three more properties are relevant throughout this thesis. They are formulated in the following theorem.

Theorem 2.1.2 ([26]). *For a matrix $A \in \mathbb{R}^{m\times n}$, with Moore-Penrose pseudo inverse $A^\dagger$, we have,*

$$A^\dagger = A^T(AA^T)^\dagger = (A^TA)^\dagger A^T \tag{2.6}$$

$$(AA^T)^\dagger = (A^T)^\dagger A^\dagger \tag{2.7}$$

$$\textit{For } a \in \mathbb{N}, \textit{let } B \in \mathbb{R}^{a\times n}, \textit{ then } \ker A \subseteq \ker B \iff B = BA^\dagger A. \tag{2.8}$$

Finally, the matrix $Q = A^\dagger A$ has some interesting properties. Q is a projection matrix as, by Equation (2.4), it is symmetric and using Equation (2.2), one can show it is idempotent, see Equation (2.9).

$$Q^2 = A^\dagger AA^\dagger A = A^\dagger A = Q \tag{2.9}$$

It is known that this matrix is the projection matrix that projects onto the complement of the kernel of A [36].

Besides the Moore-Penrose inverse, the notion of the Schur complement is also often used. Before we define the Schur complement, note that the definition stated here is for block matrices, such that the blocks consist only of adjacent rows and columns. A more general definition exists, where blocks are defined as sets of any rows and columns [9], however it will not be used in this thesis.

Definition 2.1.2 ([33], Schur complement). Let $M \in \mathbb{R}^{n\times n}$ be a block matrix $M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$, where block $D \in \mathbb{R}^{d\times d}$ is square. The (Generalized) Schur complement of D in M, is defined as,

$$M/D := A - BD^\dagger C. \tag{2.10}$$

Here, $D^\dagger$ denotes the Moore-Penrose pseudo inverse of D.

Finally, it is useful to recall two well-known identities for vector norms and inner products. For two vectors $x, y \in \mathbb{R}^n$, the Parallelogram law states,

$$||x + y||^2 + ||x - y||^2 = 2||x||^2 + 2||y||^2. \tag{2.11}$$

Secondly, the polarization identity states,

$$||x + y||^2 = ||x||^2 + ||y||^2 + 2\langle x, y\rangle. \tag{2.12}$$

Finally, these two can be combined to get the following equation,

$$2\langle x, y\rangle = ||x||^2 + ||y||^2 - ||x - y||^2. \tag{2.13}$$

2.2. Mapper

The Mapper method, introduced by [42], is a method that takes a point cloud and generates a graph with a similar structure. The resulting graph is often less complex than the entire point cloud, therefore it can be seen as a dimensionality reduction algorithm. After using Mapper, the graph can be analyzed instead of the full point cloud to perform regression, classification or any other desired task.

Instead of discussing the classical Mapper algorithm, a more recent adaptation is used, called Ball Mapper [17]. The classical algorithm is very sensitive to multiple parameters and choosing one set of these parameters that works on every image in the dataset would be very difficult, if not impossible. Ball mapper reduces the number of parameters to one single parameter, ε, the radius of the balls. The resulting graph can still vary significantly based on this parameter [32]. Nevertheless, needing to fit only one, greatly reduces the complexity.

To further explain the method, the handwritten numbers of the MNIST dataset are used, see Figure 2.1. These images are grayscale, therefore choosing some threshold parameter, we can select the pixels that have a value greater than this threshold to create a set of points. In the first step of Figure 2.1 this has been done for a hand drawn 6 with threshold equal to the average pixel intensity.

To obtain the locations of the balls, an ordering of the points needs to be chosen. In the example figure the points are ordered from top to bottom and left to right. This ordering affects where the balls appear. Starting with the first point, a ball is drawn of radius ε. Afterwards, a second ball is drawn around the next point of the chosen order that is not contained in the previously drawn ball. This is

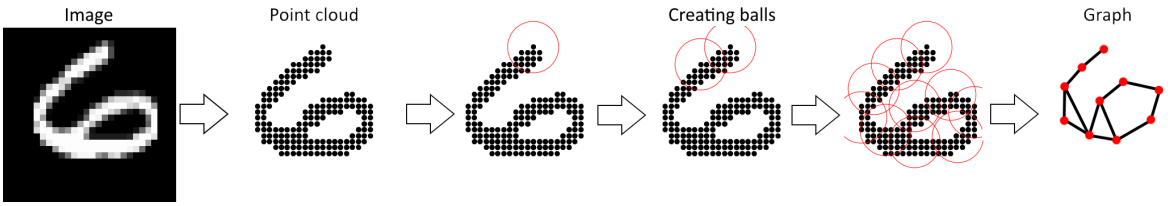


Figure 2.1: Using the Ball Mapper algorithm on the point cloud generated from an image of the MNIST dataset.

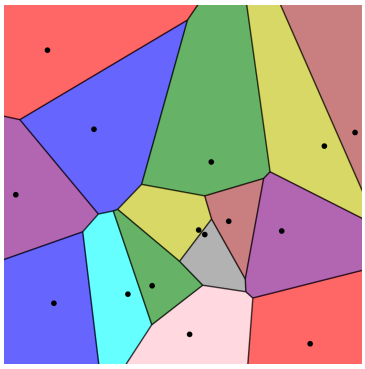


Figure 2.2: Example of a Voronoi diagram.

done iteratively until every point is contained in at least one ball. Every ball is considered a cluster, which contains a set of points.

Finally, the resulting graph is made by creating a vertex for each cluster. Edges between two vertices exist if at least one point is part of both the corresponding clusters. Note that vertices corresponding to balls that overlap, but do not have a point in the overlapping area, are not connected. Finally, note that while [17] states two algorithms, only the greedy algorithm is discussed and used.

2.3. Voronoi Diagrams

For the application in crystalline structures, the concept of Voronoi diagrams is needed. These are used in two separate steps. First, they allow the generation of images like Figure 1.2b. Secondly, when analyzing these images, they allow for an approximation that enables TDA methods. This approximation is found by taking the centers of the cells formed by the red lines and using these centers as a point cloud to create the Voronoi diagram, as is done in [43].

A Voronoi diagram can be made from a set of points by dividing the space into cells, where each cell corresponds to a region closest to a certain point, see Figure 2.2.

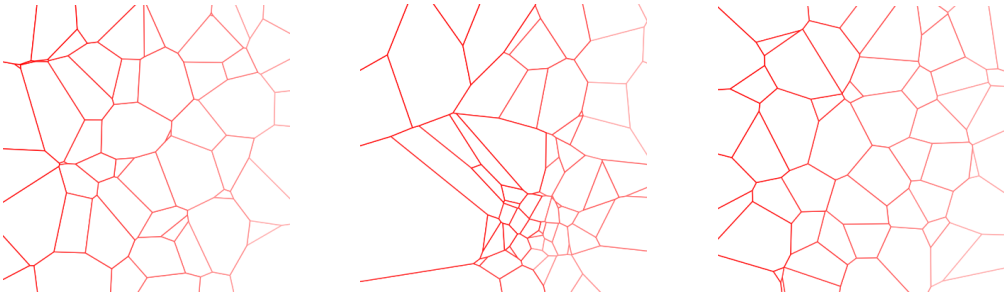
Definition 2.3.1 (Voronoi cell)**.** For a point cloud $\mathcal{P} \subset \mathbb{R}^d$ and a point $p \in \mathcal{P}$, the Voronoi cell of p is given by

$$V(p) := \{x \in \mathbb{R}^d : ||x - p|| \leq ||x - q|| \text{ for all } p \neq q \in \mathcal{P}\}. \tag{2.14}$$

Many ways exists to generate the artificial cross sections, however, only three are discussed here. The goal of the application is to detect the generation process of the images. Note that the methods do not necessarily create Voronoi diagrams, but are methods to sample point clouds. For the application of this thesis these point clouds are however only used to create Voronoi diagrams.

The first method is the Poisson Voronoi (PV) diagram. It is based on the Poisson point process. For a specified region $A \subseteq \mathbb{R}^d$ and an intensity λ it is required that for every subregion B of A, the number of points sampled in that region follows $Pois(\lambda \cdot \text{Vol}(B))$. For a rectangular region $A = [a_1, b_1] \times \cdots \times [a_d, b_d]$, this can be done by first sampling the number of points $n \sim Pois(\lambda \cdot \text{Vol}(A))$. Afterwards, uniformly sampling coordinates $x_i \sim U(a_i, b_i)$ for each point $p = \begin{bmatrix} x_1 & \dots & x_d \end{bmatrix}$.

While the microstructure of alloys is intrinsically a three dimensional object, it is visually observed in two dimensional cross-sectional data. Therefore, instead of sampling a two dimensional PV diagram, a three dimensional PV diagram is sampled and a cross section is taken. It can be shown that such



(a) Poisson Voronoi cross section. (b) Cluster cross section. (c) Hard core cross section.

Figure 2.3: Cross sections of three dimensional Voronoi diagrams. Each is sampled using a different point cloud generation method while the number of sampled points is kept to be between 225 and 275 over a unit cube.

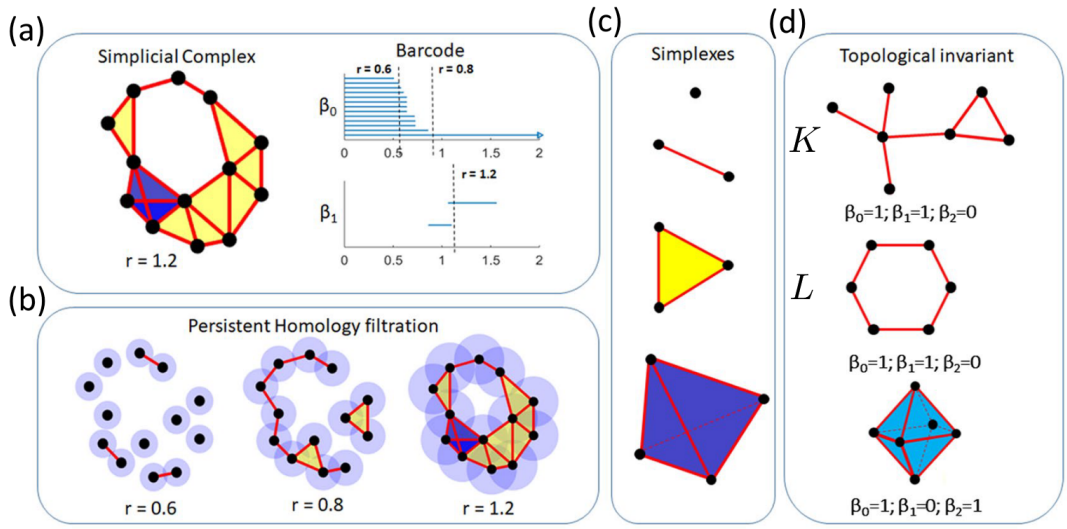


Figure 2.4: Visualization of the most important concepts in TDA. Figure from [4].

a cross section is not itself a PV diagram [43], therefore this is different from sampling a 2D diagram. See Figure 2.3a for an example of such a cross section.

Besides the PV diagram method, two more are analyzed. First, the cluster method, which instead of uniformly sampling all points, creates clusters of points, see Figure 2.3b. This is done by first sampling a pre-specified number of locations of clusters. Afterwards, the remaining points are sampled close to the clusters. In this thesis all cluster processes are generated with 3 clusters and the remaining points within a ball of radius 0.2 of each of the clusters.

Finally, the Hard Core (HC) method is used. This method does the opposite of the cluster process and samples points more evenly spread over the volume, see Figure 2.3c. It requires that every two points are at least a set distance away from one another. In this thesis the distance is chosen as 0.033, which allows for the total number of points to be between 225 and 275.

2.4. Homology theory

2.4.1. Introduction to Standard and Persistent Homology

In this section, an intuitive introduction to the concepts used in TDA is given. For a mathematically formal introduction, see the next two sections instead. For an intuitive understanding of many TDA methods it is often helpful to visualize them. In Figure 2.4 most of the important concepts are therefore drawn.

The idea of Topological Data Analysis is to use topology to describe datasets. However, most data is in the form of point clouds, which inherently do not have a topological shape. Nevertheless, these point clouds are often theorized to lie on some manifold. While this manifold is often unknown, it can

be approximated. One of the most common ways to do this, is to generate a simplicial complex [35].

Simplicial complexes consist of simplexes of different dimensions, see Figure 2.4. Looking at the simplexes section of the figure, from top to bottom, first a 0 dimensional simplex, or just 0-simplex is drawn, which is represented by a point. The second simplex is a 1-simplex, represented by a line segment. Afterwards, a 2-simplex is shown and represented by a filled triangle.

A collection of simplexes is called a simplicial complex if for every simplex, each lower dimensional simplex that is part of it is also in the collection. For example, a 2-simplex is a triangle between three points. In a simplicial complex, it is therefore required that these points are also in the complex, as well as the edges of the triangle. An example can be seen in block a of Figure 2.4.

Note that a simplicial complex consisting only of 0-simplexes and 1-simplexes can be viewed as a graph. See for example complex K in block d of Figure 2.4. Furthermore, every graph can be represented by a simplicial complex of only 0 and 1 dimensional simplices. Simplicial complexes can therefore be seen as an extension to graphs, where simplexes of dimension greater than 1 are added. Hence, for graph or network data, simplicial complexes can encode additional information in these higher dimensional simplices. Whenever additional information is available in graph or network data, it is therefore also common to extend the graph to a simplicial complex with higher dimensional simplices to allow more information to be captured by Persistent Homology. In [1] multiple methods are described to obtain the complex from a graph.

The same can be said when the data is in the form of a point cloud. Many different methods exist to generate a simplicial complex from a set of points. One of the most commonly used methods is to generate a Cech complex, see section b of Figure 2.4. Given a scalar r, a Cech complex is formed by drawing balls around every data point of radius r and forming a simplex whenever balls of different points touch. If two balls touch, a 1-simplex connecting the corresponding points is added to the simplicial complex, which happens in the figure at $r = 0.6$ twice. When three balls all overlap with each other, a filled triangle or 2-simplex is added, see $r = 0.8$ for two examples.

The topological invariants that are analyzed, are called Betti numbers, often denoted by β_q, with q the dimension. They are properties of a topological shape that remain intact under deformations like stretching, twisting or bending. The zero dimensional Betti number is, for example, equal to the number of connected components, while the one dimensional Betti number is the number of holes or loops. The final Betti number that has an intuitive explanation is β_2, which describes the number of cavities or voids. See section d of Figure 2.4 for a visualization.

While Betti numbers in a Cech complex can be analyzed for some fixed r, the resulting complex and therefore the Betti numbers, can vary greatly with respect to this parameter. See for example section b of Figure 2.4, where the number of connected components changes at every value of r. Therefore, instead of selecting one parameter, PH aims to look at all values of $r > 0$. Note that, by increasing r, the obtained simplicial complex will always contain more and more simplices. This is called a simplicial filtration.

When analyzing a filtration of simplicial complexes, the Betti numbers can change. If a 0-simplex is added, the number of connected components increases. We then say that a 0 dimensional feature is *born* at the corresponding value of the filtration parameter. When that 0-simplex is connected to the previously present simplices, the number of connected components decreases again, often described as a 0 dimensional feature *dying*. For each feature this therefore allows us to describe it with an interval of filtration parameters, where the feature exists within this interval.

These features for filtrations are described by persistent Betti numbers. The q dimensional persistent Betti number for a start time s and end time t in a filtration is equal to the number of q dimensional features that were born before s and died after t. Note that this is different from computing the difference between the non-persistent Betti numbers at times s and t. Computing this difference would not be able to differentiate between a feature dying in the interval and another being born, and one feature persisting for the entire interval.

The complete set of all of the intervals can be represented in the so called persistence barcode, see section a of Figure 2.4. If samples consist of multiple points, the barcode is thought to describe samples and can therefore be used to compare them. Furthermore, barcodes of whole datasets can be used to compare different datasets. It can be shown that small changes in the dataset also correspond to small changes in the barcode [5, 12, 19], therefore similar datasets should give similar barcodes, indicating that these barcodes are a good descriptor for datasets and samples.

2.4.2. Standard Homology

As was stated in the previous section, most of Topological Data Analysis (TDA) consists of analyzing simplicial complexes. So let us first formally define these. The notation used in this thesis is mostly the same as what the authors of [33] use and is most common among TDA papers.

Definition 2.4.1. (Simplex) For an ordered set of points $\mathcal{P}$, a q dimensional simplex σ, often just called q-simplex, is defined as an ordered subset of $\mathcal{P}$ of $q+1$ points, which adheres to the same ordering as $\mathcal{P}$.

Note that this definition is often referred to as an ordered simplex. Unordered simplices also exist, but are not used in this thesis, therefore ”simplex” will always refer to an ordered simplex. Visualizing simplices of dimension lower than 3 is possible, see section c of Figure 2.4. A 0-simplex is often represented by a point, a 1-simplex by an edge and a 2-simplex by a filled triangle. Higher dimensional simplices are not used in the thesis.

Definition 2.4.2. (Simplicial Complex) A simplicial complex K is a set of simplices, such that for any $\sigma \in K$ and $\tau \subseteq \sigma$, we have that $\tau \in K$.

We denote by S_q^K, the set of q-simplices in K and $n_q^K := |S_q^K|$, the number of q-simplices in K.

The definition stated here is the definition of an abstract simplicial complex. When drawn and therefore geometrically realized, it is also required that no two simplices intersect, except, possibly on their boundaries.

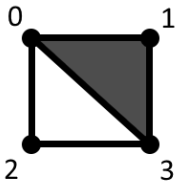


Figure 2.5: Example of a representation of a simplicial complex. Points indicate 0-simplexes, edges indicate 1-simplexes and the filled triangle indicates a 2-simplex.

Example 2.4.1. Consider the simplicial complex K drawn in Figure 2.5. We have the 0-simplices $S_0^K = \{\bar{0}, \bar{1}, \bar{2}, \bar{3}\}$. A bar is added in the notation to avoid confusion with scalars. Furthermore, we have the 1-simplices $S_1^K = \{01, 03, 13, 23, 02\}$, here short notation is used to indicate the 1-simplices, so $01 = [\bar{0}, \bar{1}]$. Finally, a 2-simplex is also present, $S_2^K = \{013\}$.

In order to do any calculations with the simplices, we need to define a vector space over them.

Definition 2.4.3. (Chain Group) The q-th chain group C_q^K is defined as the vector space over $\mathbb{R}$ with basis S_q^K.

For $c_1, c_2 \in C_q^K$, we can write $c_1 = \sum_{i=1}^{n_q^K} a_{1,i}\sigma_i$ and $c_2 = \sum_{i=1}^{n_q^K} a_{2,i}\sigma_i$ for some constants $a_{1,i}, a_{2,i} \in \mathbb{R}$ and simplices $\sigma_i \in S_q^K$. C_q^K is equipped with the inner product $\langle c_1, c_2 \rangle := \sum_{i=1}^{n_q^K} a_{1,i} \cdot a_{2,i}$.

Example 2.4.2. Let K be the simplicial complex drawn in Figure 2.5. Consider the chains $c_1 = 02 - 3 \cdot 03 + 2 \cdot 01 \in C_1^K$ and $c_2 = 23 - 03 - 2 \cdot 01 \in C_1^K$. The inner product between the two chains is $\langle c_1, c_2 \rangle = 1 \cdot 0 + 0 \cdot 1 + (-3) \cdot (-1) + 2 \cdot (-2) = -1$.

An important definition for homology is the boundary operator. It is an operator on the chain group that relates a q-simplex to its $q-1$-subsets.

Definition 2.4.4. (Boundary Operator) Let K be a simplicial complex and $q \in \mathbb{Z}_{\geq 0}$ such that $n_q^K > 0$. For a simplex $[p_0, \dots, p_q] = \sigma \in S_q^K$, the q-boundary operator $\partial_q^K : C_q^K \to C_{q-1}^K$ is defined as follows:

$$\partial_q^K(\sigma) := \sum_{i=0}^{q} (-1)^i d_i \sigma. \tag{2.15}$$

Where $d_i\sigma := [p_0, \dots, p_{i-1}, p_{i+1}, \dots, p_q]$, the $(q-1)$-simplex which omits point p_i from σ.

The most important and well-known property is that the boundary of the boundary is zero, $\partial_q^K \partial_{q+1}^K = 0$ [49]. This property is the basis of homology theory.

Using this boundary operator, we can define two subspaces of C_q^K.

Definition 2.4.5. (Cycles & Boundaries) For a simplicial complex K and some dimension $q \in \mathbb{Z}_{\geq 0}$, the cycles Z_q^K are defined by,

$$Z_q^K := \ker \partial_q^K. \tag{2.16}$$

Furthermore, the boundaries B_q^K are defined by,

$$B_q^K := \operatorname{Im} \partial_{q+1}^K \tag{2.17}$$

Note that $B_q^K \subseteq Z_q^K$, because, for some $b \in B_q^K$, there exists a $c \in C_{q+1}^K$ such that $\partial_q^K b = \partial_q^K \partial_{q+1}^K c = 0$, therefore $b \in Z_q^K$. This shows that the homology group in the next definition, is well defined.

Definition 2.4.6. (Homology group & Betti number) For a simplicial complex K and some dimension $q \in \mathbb{Z}_{\geq 0}$, the homology group H_q^K is defined by,

$$H_q^K := Z_q^K / B_q^K. \tag{2.18}$$

The dimension of this homology group is called the q-th Betti number of K, β_q^K. Formally,

$$\beta_q^K := \dim H_q^K = \dim Z_q^K - \dim B_q^K. \tag{2.19}$$

Example 2.4.3. Consider again the simplicial complex K drawn in Figure 2.5. We have a basis for the 1-cycles $Z_1^K = \{01 - 03 + 13, 02 - 03 + 23\}$. This can be checked by computing $\partial_1^K(01 - 03 + 13) = (\bar{1} - \bar{0}) - (\bar{3} - \bar{0}) + (\bar{3} - \bar{1}) = 0$ and $\partial_1^K(02 - 03 + 23) = 0$. Note that this basis is not unique as $\{01 - 03 + 13, 01 + 13 - 23 - 02\}$ is also valid. A basis for the 1-boundaries is unique as there is only one and can be written as $\operatorname{Im} \partial_2^K = \{\partial_2^K(013)\} = \{01 - 03 + 13\}$. We can now compute a basis of the 1-homology $H_1^K = \{[02 - 03 + 23]\}$, where $[\cdot]$ represents the equivalence class. Finally we can compute the dimension 1 Betti number by looking at the dimension of this homology group and conclude $\beta_1^K = 1$. Visually this can be seen by the one hole 023 that is present.

For two distinct q-simplices σ_1, σ_2 in a simplicial complex K, we call them upper adjacent, denoted by $\sigma_1 \overset{U}{\sim} \sigma_2$, if there is a $\tau \in K$ such that $\sigma_1, \sigma_2 \subset \tau$. This τ is often called their common upper simplex and σ_1 and σ_2 are said to be faces of τ. Furthermore, if the sign of the two simplices in the boundary $\partial_{q+1}^K(\tau)$ of τ is the same, we say that they are similarly oriented and otherwise dissimilarly oriented. Finally, the upper degree of a q-simplex σ, $\deg_U(\sigma)$, is the number of $(q+1)$-simplices in K for which σ is a face.

In the same way, we call two q-simplices $\sigma_1, \sigma_2 \in K$ lower adjacent, denoted by $\sigma_1 \overset{L}{\sim} \sigma_2$, if there is a $(q-1)$-simplex $\tau \in K$ such that $\tau \subset \sigma_1$ and $\tau \subset \sigma_2$. This τ is called their common lower simplex and is said to be a similar common lower simplex if the sign in the boundaries $\partial_q^K(\sigma_1)$ and $\partial_q^K(\sigma_2)$ is the same. Otherwise, τ is called a dissimilar common lower simplex.

In [24] it is proved that for a pair of simplices σ_1 and σ_2, the common lower simplex and the common upper simplex are both unique. Furthermore, they prove an important Corollary.

Corollary 2.4.1 ([24], Corollary 3.2.7)**.** *Let $q > 0$ be an integer. If two distinct q-simplices of a simplicial complex are upper adjacent, then they are also lower adjacent.*

Example 2.4.4. Consider the simplicial complex K drawn in Figure 2.5 and let the ordering of the point cloud be given by the names of the 0-simplices. The 1-simplices 03 and 13 are lower adjacent, because they both contain the 0-simplex $\bar{3}$, so we write $03 \overset{L}{\sim} 13$. The boundaries $\partial_1^K(03) = \bar{3} - \bar{0}$ and $\partial_1^K(13) = \bar{3} - \bar{1}$ both have a positive sign for $\bar{3}$, therefore $\bar{3}$ is a similar common lower simplex. The 1-simplices 01 and 13 are also lower adjacent as they share $\bar{1}$. However, the sign of $\bar{1}$ in the boundary $\partial_1^K(01) = \bar{1} - \bar{0}$ is different from the boundary of 13, therefore $\bar{1}$ is a dissimilar common lower simplex of the two.

All the discussed 1-simplices are part of the 2-simplex 013, therefore they are all upper adjacent to each other, so $03 \overset{U}{\sim} 13 \overset{U}{\sim} 01$. From the boundary $\partial_2^K(013) = 13 - 03 + 01$, we obtain that 13 and 01 are similarly oriented as their sign is the same. On the other hand the pairs 13 and 03, as well as 01 and 03, both have opposite sign and are therefore dissimilarly oriented.

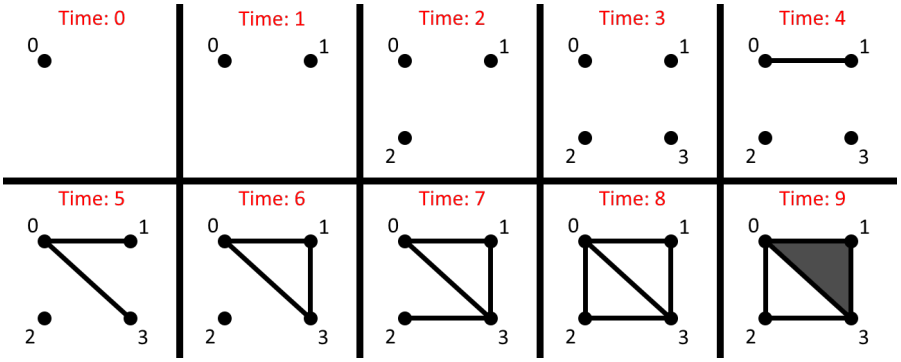


Figure 2.6: Example filtration of simplicial complexes. Points indicate 0-simplexes, edges indicate 1-simplexes and the filled triangle indicates a 2-simplex.

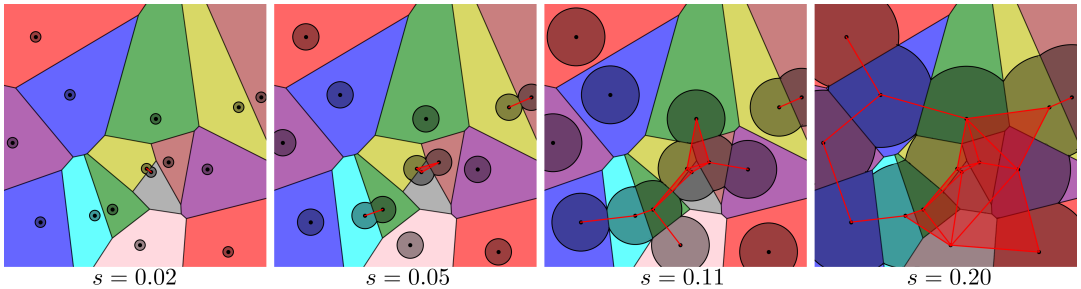


Figure 2.7: Example of an alpha filtration for different scale parameters s, using the point cloud of the Voronoi example in Figure 2.2. All points are always included in the simplicial complexes as 0-simplices.

2.4.3. Persistent Homology

Instead of looking at a single complex K, persistent homology looks at a set of complexes and analyzes the changing Betti numbers. In general, any set of complexes can be used [10], however for this thesis only filtrations are relevant.

Definition 2.4.7. (Filtration) A filtration is a set of simplicial complexes $\{K_0, K_1, ..., K_N\}$ for some $N > 0$, such that for any $0 \leq i < j \leq N$, we have $K_i \subseteq K_j$.

A filtration of two simplicial complexes K, L is called a simplicial pair, their inclusion is denoted by $K \hookrightarrow L$. For an example of a filtration, see Figure 2.6.

In practice, filtrations are often defined using some filtration parameter s, which does not need to be an integer. To simplify notation, K_s is often referred to as the simplicial complex that corresponds to this filtration parameter. Given a minimum parameter value $T_{min} \in \mathbb{R}$, a maximum parameter value $T_{max} \in \mathbb{R}$ and a finite discretization T_{disc} of the interval $[T_{min}, T_{max}]$, we call $\{K_s\}_{s \in T_{disc}}$ a filtration if $K_s \subseteq K_t$ for all $T_{min} \leq s < t \leq T_{max}$.

Finding a filtration

One of the most common ways of finding such a filtration is by forming the Cech filtration from a point cloud $\mathcal{P} \subset \mathbb{R}^d$. For a scale $s \geq 0$, it is defined by the intersection of closed balls around the points, where a closed ball is defined by $B_s(p) := \{x' \in \mathbb{R}^d : ||p - x'|| \leq s\}$. Formally we can write

$$\text{Cech}_s(\mathcal{P}) := \{\sigma \subseteq \mathcal{P} : \bigcap_{p \in \sigma} B_s(p) \neq \emptyset\}. \tag{2.20}$$

For an example of the Cech filtration, see block b of Figure 2.4.

Instead of a Cech filtration in [43] an alpha filtration is used. The alpha complex is similar to the Cech complex, but uses the Voronoi cells $V(p)$ from Definition 2.3.1. See Figure 2.7 for an example.

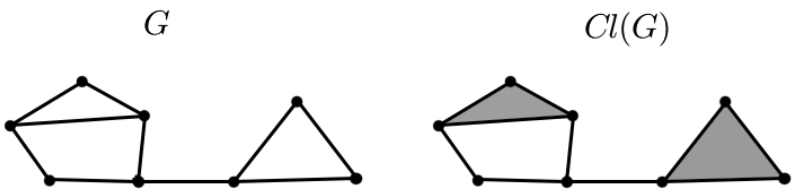


Figure 2.8: Left a graph G, with on the right the corresponding clique complex $Cl(G)$.

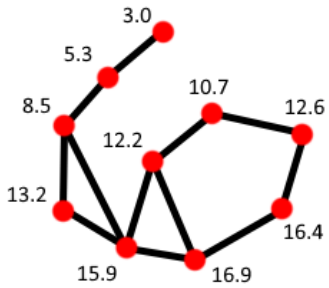


Figure 2.9: Ball Mapper output graph with vertices weighted according to average y-value of points in the corresponding cluster.

Definition 2.4.8 (Alpha Complex)**.** The alpha complex of a point cloud $\mathcal{P} \subset \mathbb{R}^d$ for a scale $s \geq 0$ is the simplicial complex

$$\text{Alpha}_s(\mathcal{P}) := \{\sigma \subseteq \mathcal{P} : \bigcap_{p \in \sigma} (V(p) \cap B_s(p)) \neq \emptyset\}, \tag{2.21}$$

where $B_s(p)$ is the closed ball around p with radius s.

In the applications of this thesis, images are converted into graphs where persistent homology is applied to these graphs, see Section 2.2. While a graph can immediately be viewed as a simplicial complex of only 0 and 1 dimensional simplices, it is common to instead also make use of higher dimensional simplices. Often the clique complex is used to this end, see Definition 2.4.9 and Figure 2.8. It requires the notion of a clique, which is a set of vertices in the graph that are all connected to each other. A k-clique is a set of k vertices that are all connected.

Definition 2.4.9 (Clique complex [1])**.** For a graph $G = (V, E)$, the clique complex $Cl(G)$ is a simplicial complex where all vertices in V are 0-simplices in $Cl(G)$ and every k-clique corresponds to a $(k-1)$-simplex in $Cl(G)$.

For persistent homology, a filtration of simplices is often required. Numerous methods have been proposed to find such a filtration from a graph [1]. One of these is the vertex-based clique filtration, which is probably the most intuitive for a vertex weighted graph. A vertex weighted graph $G = (V, E)$ is a graph, with V the vertices and E the edges, where we require a weight function $w : V \to \mathbb{R}$ on the vertices.

Definition 2.4.10 (Vertex-based clique filtration (VBCL), [1])**.** Let $G = (V, E)$ be an undirected weighted graph, with weight function $w : V \to \mathbb{R}$. For $\delta \in \mathbb{R}$, the 1-skeleton $G_\delta = (V_\delta, E_\delta) \subseteq G$ is defined as the subgraph of G, where $V_\delta := \{v \in V : w(v) \leq \delta\}$ and the edges $E_\delta := \{e = \{u, v\} \in E : \max(w(u), w(v)) \leq \delta\}$. The vertex-based clique filtration is defined as

$$\{Cl(G_\delta) \hookrightarrow Cl(G_{\delta'})\}_{0 \leq \delta \leq \delta'}. \tag{2.22}$$

Example 2.4.5. Consider the graph of the ball mapper output in Figure 2.1. As weight function on the vertices, we will choose the average y-value of points in the corresponding cluster, see Figure 2.9. Note that vertices in the figure can be lower, while having a higher weight because the average y-value of the points in the ball may be lower than the middle of the ball. The resulting Vertex-Based clique filtration can be seen in Figure 2.10.

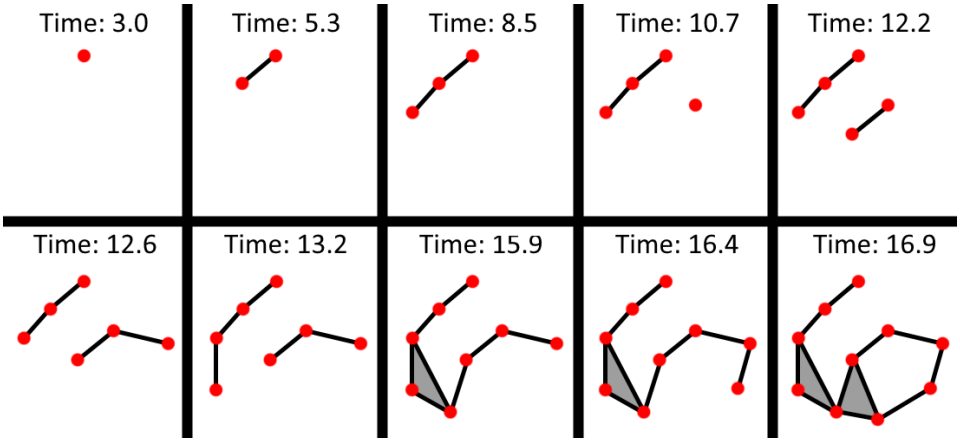


Figure 2.10: Vertex-Based Clique filtration of the weighted graph from Figure 2.9.

Analyzing the filtration

On a filtration, persistent homology looks at persistent Betti numbers. To formally define persistent Betti numbers, we need the notion of a simplicial map. Let K and L be two simplicial complexes, a simplicial map $f : K \to L$ is a function such that for every simplex $[p_0, \dots, p_q] \in S_q^K$ its image is a simplex of L, $[f(p_0), \dots, f(p_n)] \in S_q^L$, for every dimension q. A simplicial map induces a linear map in the chain complexes $f_\# : C_q^K \to C_q^L$, defined by,

$$f_\#(\sigma) := \begin{cases} f(\sigma) & \text{if } \dim f(\sigma) = \dim \sigma. \\ 0 & \text{otherwise.} \end{cases} \tag{2.23}$$

Furthermore, it is well known that a simplicial map also induces a well-defined map in homology $f_* : H_q^K \to H_q^L$ defined by $f_*([c]) = [f_\#(c)]$, where $[\cdot]$ is the equivalence class of the quotient space.

We are now ready to define the main features that are tracked using persistent homology.

Definition 2.4.11. (Persistent Betti Number) For a simplicial pair K, L, we have the inclusion map $\iota : K \hookrightarrow L$, defined by $\iota : x \mapsto x$. This inclusion map is a simplicial map and therefore induces a map in homology $f_*^{K,L} : H_q^K \to H_q^L$. The q-th persistent Betti number $\beta_q^{K,L}$ is defined as the rank of this map,

$$\beta_q^{K,L} := \dim \operatorname{Im} f_*^{K,L}. \tag{2.24}$$

To simplify notation for filtrations over a discretized interval, we denote $\beta_q^{s,t} := \beta_q^{K_s,K_t}$, for $s < t$.

The persistent Betti number $\beta_q^{s,t}$ describes the q-dimensional features that are present in K_s and still appear in K_t. Every combination of $s \leq t$ yields a persistent Betti number. However, this is often too much information and can be "summarized" by only looking at when features appear and when they disappear, often called *born* and *die*, respectively. This gives rise to the notion of persistent barcodes.

Definition 2.4.12. (Persistent Barcode) A persistent barcode of dimension q for a filtration $\{K_s\}_{0 \leq s \leq T}$ is the set of intervals $\{[s_i, t_i]\}_i$, where the multiplicity $\mu_q^{s_i,t_i}$ is positive. For an interval $[s, t]$, this multiplicity can be calculated with the Betti numbers as follows,

$$\mu_q^{s,t} := \left(\beta_q^{s,t-1} - \beta_q^{s,t}\right) - \left(\beta_q^{s-1,t-1} - \beta_q^{s-1,t}\right), \qquad \mu_q^{i,\infty} := \beta_q^{s,T} - \beta_q^{s-1,T}. \tag{2.25}$$

Each interval $[s, t]$ in the barcode captures a feature, where s is often referred to as the *birth* time of the feature and t the *death* time. See section a of Figure 2.4 for a common way of visualizing the barcode. Each bar corresponds to the "lifespan" of a certain feature in the filtration visualized in section b of the same figure.

The multiplicity equation (2.25) for finite intervals is split into two parts. The first part describes the number of features that were born before s and were still alive before t, but died at t, while the second part describes the features that were alive already before s and died at t. Subtracting these two parts yields the features that were born at exactly s and died at exactly t. Infinite intervals describe

the features that are still alive at the end of the filtration, therefore we only care that the they were born at exactly s, requiring only one subtraction.

An algorithm for computing the barcode is given in [49]. For a simplicial filtration of N complexes, where there is a single simplex added in each step and is started with a single simplex, the time complexity of the standard algorithm to compute the barcode for all relevant dimensions is at most $O(N^3)$ [16].

Besides the bar plot, visualizing the barcode is often done in a persistence diagram. This is a scatter plot, with on the x-axis the birth times s and on the y-axis, the death times t. The multiplicity is often not visualized, see Figure 2.11 for an example. For this thesis persistence diagrams are always used instead of bar plots.

One of the main results of persistent homology is the stability of these barcodes in terms of the bottleneck distance with respect to the filtration function [5, 12, 19]. The main point of PH is to analyze the shape of the data. If two filtration functions are very similar, but PH would give very different results for each function while the topology is not changed, we would mainly be able to detect the filtration function and not the actual shape of the data. Therefore, in order for the results of PH to be interpretable, this type of stability is needed.

To better understand the stability, we look at an intuitive interpretation of the bottleneck distance. The distance compares two persistence diagrams by matching points of the first diagram with points of the second diagram that are close. It is defined as the infimum cost of all possible matchings, where the cost of a matching is equal to the maximum distance between the matched points. Points can also be left unmatched, these have a cost equal to their distance to the diagonal. Therefore, points that are far away from the diagonal are deemed more relevant.

While this subject of stability is very important, in this thesis it is not discussed beyond this explanation. Instead, this is left for future research.

Example 2.4.6. We consider the filtration visualized in Figure 2.6. The persistence diagram corresponding to this filtration can be seen in Figure 2.11. Each point in this diagram corresponds to an interval of the barcodes. Some of these points are now discussed in more detail, however only an intuitive interpretation is discussed.

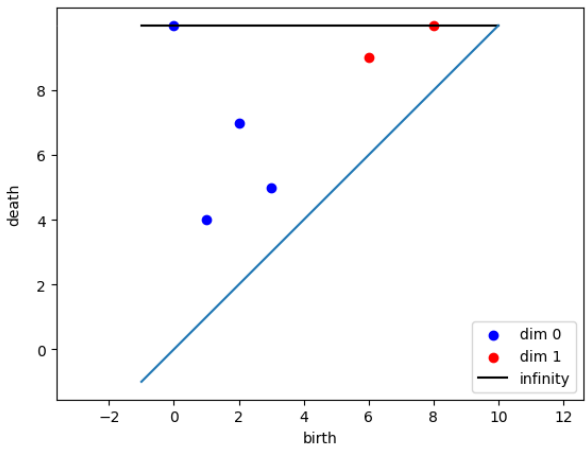


Figure 2.11: The persistence diagram corresponding to the filtration of Figure 2.6. The blue points correspond to the 0 dimensional features, while the red points corresponds to 1 dimensional features. Finally, points that lie on the black line have an infinite lifetime.

We first look at the point at $(0, \infty)$. Starting at time 0, we see one connected component, so $\beta_0^0 = 1$. In all subsequent timesteps this connected component is still present, therefore the persistent Betti number $\beta_0^{0,t} = 1$, for all $t \in [1, 9]$. Using the multiplicity equation (2.25), we get $\mu_0^{0,\infty} = 1$ as $s - 1$ does not exist in this case.

If we instead look at the connected component that is born at time 2, we can see that at time 7 this component gets merged with the component of point 0. If we were to calculate the multiplicity of this possible interval, we would get $\mu_0^{2,7} = \left(\beta_0^{2,6} - \beta_0^{2,7}\right) - \left(\beta_0^{1,6} - \beta_0^{1,7}\right) = (2-1) - (1-1) = 1 > 0$ and can therefore conclude that this interval belongs in the barcode.

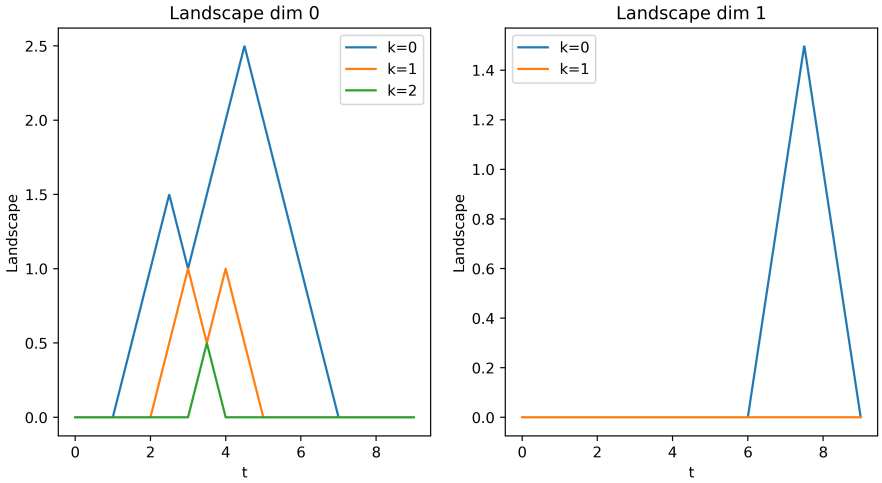


Figure 2.12: Persistence landscape based on the filtration visualized in Figure 2.6 and with corresponding persistence diagram of Figure 2.11.

For dimension 1, a hole appears at time 6, which gets filled and therefore dies, at 9, resulting in the multiplicity $\mu_1^{6,9} = 1$. Finally for this dimension, the hole that is born at time 8 does not die and therefore $\mu_1^{8,\infty} = 1$.

Persistence Landscapes

Like was said previously, two persistence diagrams can be compared using the bottleneck distance. However, the distance is not easy to compute and therefore comparing diagrams with lots of points can be time consuming. Furthermore, computing a ”mean diagram” is not well defined, even though it would be of great use to many statistical methods.

Persistence landscapes [7] were proposed as an alternative to persistence diagrams that solve some of these issues. They can be computed using a persistence diagram, PD, where only points of a single dimension q are considered $PD(q)$. Each point $p = (b, d)$ in the diagram is transformed into a simple linear function. Intuitively this function can be found by taking the persistence diagram and drawing a vertical and a horizontal line from the point p to the diagonal. Afterwards, *tipping* the diagram such that the diagonal is on the x-axis. This then yields a triangular function Λ_p for each point. Formally the functions are defined by,

$$\Lambda_p(t) = \begin{cases} t - b & \text{if } t \in [b, \frac{b+d}{2}]. \\ d - t & \text{if } t \in (\frac{b+d}{2}, d]. \\ 0 & \text{otherwise.} \end{cases} \tag{2.26}$$

Note that persistent points that correspond to infinite intervals do not have a well defined function and are therefore not encoded in the landscape.

The persistence landscape is a function defined over the set of all triangular function $\{\Lambda_p\}_{p \in PD(q)}$. In addition to taking a t, it requires some positive integer k and is defined by,

$$\lambda_{PD(q)}(k, t) = k \max_{p \in PD(q)} \Lambda_p(t), \tag{2.27}$$

where $k \max$ is the k-th largest value in the set.

An example of a landscape can be seen in Figure 2.12. Here the filtration of Figure 2.6 has been used. Comparing it to the corresponding persistence diagram, see Figure 2.11, we can see that in dimension 0, the two points furthest away from the diagonal are visible in $k = 0$, while the third point is only visible in the higher values of k. Like in the bottleneck distance, points further away from the diagonal are therefore deemed to correspond to more relevant features. To obtain the intervals of the barcode from such a landscape, one can look at the triangles. Each triangle corresponds to a features which is born at the value of t where the triangle first became positive and died where it returned to zero.

Comparing two diagrams can now be done by first computing the persistence landscapes and then integrating the squared difference of each $\lambda(k,\cdot)$. Formally, the distance between two persistence diagrams PD_1 and PD_2 can be defined by the following equation, using their corresponding landscapes λ_{PD_1} and λ_{PD_2},

$$d_{land}(PD_1, PD_2) = \sum_q \left[\sum_k \int_0^T (\lambda_{PD_1(q)}(k,t) - \lambda_{PD_2(q)}(k,t))^2 dt \right]^{\frac{1}{2}}. \tag{2.28}$$

The sum over k can be taken over all values of where one of the two still is non-zero. This requires $k\max$ to be equal to 0 if k is greater than the size of the set. On the other hand, the sum could also be taken up to some pre-defined integer. Furthermore, the sum over q is often taken as just a single q which is deemed most important.

2.5. Laplacians

2.5.1. Introduction to Combinatorial Laplacians

In this section an intuitive introduction to combinatorial Laplacians is given, see the next section for formal definitions. The combinatorial Laplacian is an operator on a topological shape. Unlike persistent homology, it is only defined for a single shape and not a filtration. Nevertheless, it is also used to interpret the shape of the data, therefore, it can be seen as a part of TDA. However, instead of only focusing on topological invariants, it also encodes geometric information. To highlight the importance of this information, first an example is discussed.

In section d of Figure 2.4, the simplicial complexes K and L have the same Betti numbers. Therefore, topological information is not enough to distinguish them. Nevertheless, visually they appear very different. Hence, they differ only in geometric information instead of the topological information of Betti numbers. The combinatorial Laplacian contains both types of information. The number of zero eigenvalues or dimension of its kernel, is equal to the non-persistent Betti number and geometric information is encoded in the remaining spectrum. It is worth noting that for the complexes in the figure, making a filtration and applying PH, might be able to distinguish them without geometric information. However, the example can still be used to explain the difference between the two types of information.

Combinatorial Laplacians are often denoted by Δ_q^K, where K is a simplicial complex and q the dimension to be analyzed. Without going into the details, a matrix representation can be found for this operator, denoted by $[\Delta_q^K]$. Let K and L be the simplicial complexes visualized in section d of Figure 2.4. The following matrix representations can be found.

$$[\Delta_0^K] = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & -1 & -1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 3 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & 2 \end{bmatrix}$$

$$[\Delta_0^L] = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & 0 & -1 & 2 \end{bmatrix}.$$

While an obvious difference between the entries of these two matrices can immediately be seen, it is most common to interpret the eigenvalues of these matrices instead. In order to generate these matrices, an ordering of the points was required. Depending on this order, the specific entries of the matrix representation can differ. On the other hand, it has been proven that the eigenvalues of the combinatorial Laplacian are independent of this choice [24]. Therefore these are mainly used instead. Furthermore, because the number of eigenvalues of the combinatorial Laplacian is equal to the number

of q-simplices, an aggregation function is needed to compare two complexes that differ in the number of q-simplices. Common functions are the minimum non-zero eigenvalue or the maximum eigenvalue [44, 46].

The eigenvalues of the 0 dimensional combinatorial Laplacian for the two discussed simplicial complexes rounded to two decimals, are given by,

$$\begin{aligned}\text{Spectrum}(\Delta_0^K) &= \{0, \quad 0.40, \quad 1, \quad 1, \quad 3, \quad 3.34, \quad 5.26\}\\ \text{Spectrum}(\Delta_0^L) &= \{0, \quad 1, \quad 1, \quad 3, \quad 3, \quad 4\}.\end{aligned}$$

Note that both complexes have exactly one zero-eigenvalue, representing the zero dimensional Betti number. Looking at the minimum non-zero eigenvalues, 0.40 is different from 1, therefore using the combinatorial Laplacian it can be concluded that the geometry of these simplicial complexes is indeed different.

In practice, often a simplicial complex needs to be made from data. The most common ways require a parameter to make this complex and the resulting simplicial complex can vary greatly in the choice of this parameter. Like in the case of persistent homology, it is therefore argued that instead of selecting one value of this parameter, a set of possible values needs to be analyzed. Extending the combinatorial Laplacian to this setting, yields the persistent Laplacian [44]. This operator is theorized to contain the information of the changes in the geometry of the resulting simplicial complex when varying the parameter. See Section 2.5.4 for a more detailed discussion on this operator and an example of its usage.

2.5.2. Formal Definition of the Combinatorial Laplacian

For a graph (V, E), with V the ordered set of vertices of the graph and $E \subseteq V \times V$ the set of edges, the graph Laplacian L is often defined as the difference between the degree matrix D and the adjacency matrix A, $L := D - A$. Here the $|V| \times |V|$ degree matrix D is a diagonal matrix that contains the degrees of all the vertices on the diagonal. The degree of a vertex is equal to the number of edges that have an end point in the vertex. Note that this coincides with the upper degree of a 0-simplex in a simplicial complex. Finally, the adjacency matrix A is a symmetric binary matrix where the entry on row i and column j contains a 1 when $(v_i, v_j) \in E$ and is otherwise 0.

This definition can be extended to simplicial complexes, called the combinatorial Laplacian.

Definition 2.5.1. (Combinatorial Laplacian) For a simplicial complex K, the q-combinatorial Laplacian $\Delta_q^K : C_q^K \to C_q^K$ is defined by

$$\Delta_q^K := \underbrace{\partial_{q+1}^K (\partial_{q+1}^K)^*}_{\Delta_{q,+}^K} + \underbrace{(\partial_q^K)^* \partial_q^K}_{\Delta_{q,-}^K}. \tag{2.29}$$

With $(\partial_{q+1}^K)^* : C_q^K \to C_{q+1}^K$, the Hermitian adjoint of the boundary operator ∂_{q+1}^K on the inner product space C_{q+1}^K. Furthermore, we define $\Delta_{q,+}^K$ and $\Delta_{q,-}^K$ as the up- and down-Laplacian, respectively.

Note that, for a chosen basis of C_q^K and C_{q+1}^K, we can write the matrix representation of $(\partial_q^K)^*$ as $[(\partial_q^K)^*] = [\partial_q^K]^T$, with $(\cdot)^T$ the normal matrix transpose. Furthermore, by convention $\partial_0^K := 0$ and therefore $\Delta_0^K = \partial_1^K (\partial_1^K)^*$.

For the remainder of the thesis, the notation $[F]$ for an operator F or a vector $F \in C_q^K$ is used to indicate the matrix representation of F. While ∂_q^K is a function from $C_q^K \to C_{q-1}^K$, the matrix representation $[\partial_q^K]$ is a function from $\mathbb{R}^{n_q^K} \to \mathbb{R}^{n_{q-1}^K}$. If not specified, the matrix representation is assumed to be in the trivial basis S_q^K.

An expression for every element of the matrix representation of the combinatorial Laplacian was found in [24]. This is formulated in the next Theorem.

Theorem 2.5.1 ([24], Theorem 3.3.4)**.** *Let K be a simplicial complex and let $S_q^K = \{\sigma_1, \sigma_2, \dots, \sigma_{n_q^K}\}$ denote the q-simplices of K. The entries of the matrix representation according to the basis formed by S_q^K of*

the q-combinatorial Laplacian, can be described as,

$$q > 0, [\Delta_q^K]_{i,j} = \begin{cases} \deg_U(\sigma_i) + q + 1, & \text{if } i = j. \\ 1, & \text{if } i \neq j, \sigma_i \overset{U}{\nsim} \sigma_j \text{ and } \sigma_i \overset{L}{\sim} \sigma_j \text{ with similar common lower simplex.} \\ -1, & \text{if } i \neq j, \sigma_i \overset{U}{\nsim} \sigma_j \text{ and } \sigma_i \overset{L}{\sim} \sigma_j \text{ with dissimilar common lower simplex.} \\ 0, & \text{if } i \neq j \text{ and either, } \sigma_i \overset{U}{\sim} \sigma_j \text{ or } \sigma_i \overset{L}{\nsim} \sigma_j. \end{cases} \tag{2.30}$$

$$q = 0, [\Delta_q^K]_{i,j} = \begin{cases} \deg_U(\sigma_i), & \text{if } i = j. \\ -1, & \text{if } i \neq j, \sigma_i \overset{U}{\sim} \sigma_j. \\ 0, & \text{otherwise.} \end{cases} \tag{2.31}$$

We can see that the entries of the combinatorial Laplacian may depend on the orientation of the simplices, which in the definitions given here, is completely defined by the chosen ordering of the points in the point set $\mathcal{P}$. However, it is proven [24] that the spectra of the q-combinatorial Laplacian are independent of this choice of orientation. Therefore, the chosen ordering does not affect any conclusions drawn from the spectra.

The reason, these Laplacians are interesting for TDA, is because of the next theorem. It shows that the information we are interested in, namely Betti numbers, is completely described by the 0-eigenspace of the Laplacian.

Theorem 2.5.2 ([18])**.** *For each* $q \in \mathbb{N}$, $\beta_q^K = \dim \ker \Delta_q^K$.

Example 2.5.1. Consider again the filtration visualized in Figure 2.6. We first look at the final simplicial complex K_9. To compute a matrix representation of the 1-combinatorial Laplacian $\Delta_1^{K_9}$, we could use Theorem 2.5.1 or use the definition. To illustrate the computation, we first compute it using the definition. To do this, we need the matrix representations of the boundary functions $\partial_1^{K_9}$ and $\partial_2^{K_9}$. Written in the canonical basis, we get,

$$\left[\partial_1^{K_9}\right] = \begin{array}{c} \\ \bar{0} \\ \bar{1} \\ \bar{2} \\ \bar{3} \end{array} \begin{array}{c} \begin{array}{ccccc} 01 & 03 & 13 & 23 & 02 \end{array} \\ \begin{pmatrix} -1 & -1 & 0 & 0 & -1 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix} \end{array}, \qquad \left[\partial_2^{K_9}\right] = \begin{array}{c} \\ 01 \\ 03 \\ 13 \\ 23 \\ 02 \end{array} \begin{array}{c} \begin{array}{c} 013 \end{array} \\ \begin{pmatrix} 1 \\ -1 \\ 1 \\ 0 \\ 0 \end{pmatrix} \end{array}.$$

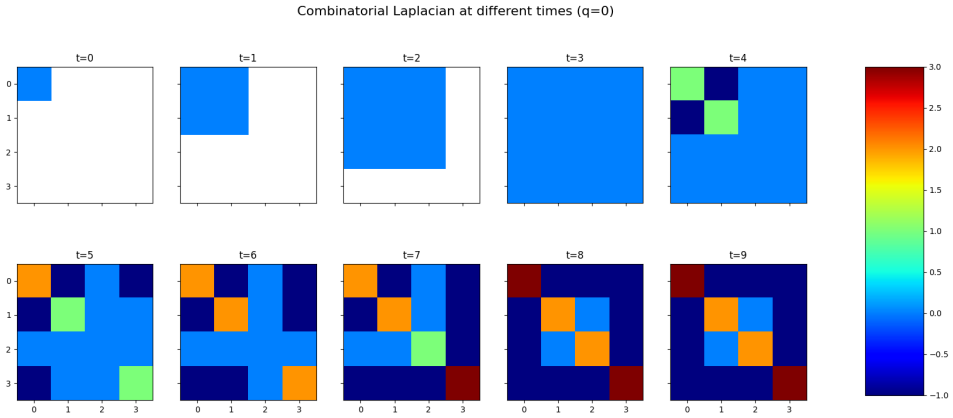
The combinatorial Laplacian then follows,

$$[\Delta_1^{K_9}] = \left[\partial_2^{K_9}\right]\left[\partial_2^{K_9}\right]^T + \left[\partial_1^{K_9}\right]^T\left[\partial_1^{K_9}\right] = \begin{pmatrix} 3 & 0 & 0 & 0 & 1 \\ 0 & 3 & 0 & 1 & 1 \\ 0 & 0 & 3 & 1 & 0 \\ 0 & 1 & 1 & 2 & -1 \\ 1 & 1 & 0 & -1 & 2 \end{pmatrix}.$$

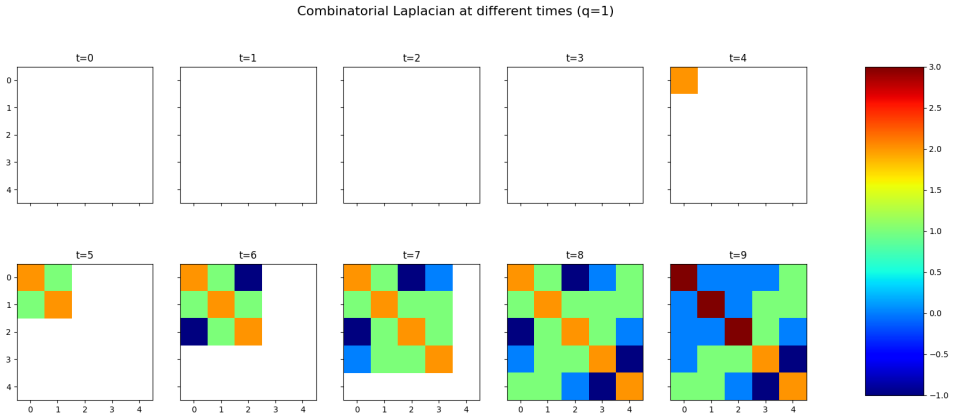
The eigenvalues of this matrix can be computed to obtain $\lambda_1 = 4, \lambda_2 = 4, \lambda_3 = 3, \lambda_4 = 2, \lambda_5 = 0$. Hence we see only one zero eigenvalue and can determine that $\beta_1^{K_9} = 1$.

2.5.3. Visualizing and using the combinatorial Laplacian

The most straight forward way of visualizing the combinatorial Laplacian is to make a heatmap of the matrix representation for each relevant time, see Figure 2.13. Even though we have a complete description of the combinatorial Laplacian using Equation (2.30), it still might be more intuitive to look at this heatmap. This is also done in [45]. They note that when the complex represents a more complete graph, so when almost all possible 1-simplices are added, the off-diagonal entries of the 0-combinatorial Laplacian converge to -1. This is also apparent from Equation (2.30) as for a complete graph, every 0-simplex will be upper adjacent to any other 0-simplex. Looking at the Laplacian for $t = 9$, we see



(a) Dimension 0.



(b) Dimension 1.

Figure 2.13: Heatmap of a matrix representation of the combinatorial Laplacian $\left[\Delta_q^{K_t}\right]$ for each time step t of the filtration visualized in Figure 2.6.

that only 1 and 2 are not yet connected as their entry is still 0. Finally, for dimension 1, we can see that the Laplacian converges to a diagonal matrix, which can also be concluded from Equation (2.30).

Using this representation, the changes in the matrix during the filtration become visible. However, an exact meaning of what the changes represent is still unclear and comparing two filtrations is still difficult as the matrices can vary in size. Furthermore, the specific entries in the matrix may depend on the chosen order of the point cloud. For this, we would like to look at the eigenvalues of each of the matrices, which are independent of the order of the points [24]. Because of Theorem 2.5.2, we know the number of 0-eigenvalues corresponds to the Betti number and therefore the topological information. However, persistent homology already captures this information. For the usage of the combinatorial Laplacian to make sense, we require additional information. Therefore, we mainly look at the non-zero eigenvalues.

To use the non-zero eigenvalues they are often aggregated into one value for each time step and each dimension [44, 46]. Common ways are to either sum the eigenvalues, take their average or compute the maximum or minimum non-zero eigenvalue. Using Equation (2.30) we can already say something about the properties described by the sum and average. This allows us to disregard them further and in the process clarify the information we are after.

The sum of the non-zero eigenvalues is equal to the trace of the matrix. For a simplicial complex K, we therefore get,

$$\sum \lambda\left(\left[\Delta_q^K\right]\right) = Tr\left[\Delta_q^K\right] = \begin{cases} 2n_1^K & \text{if } q = 0. \\ (q+2)n_{q+1}^K + (q+1)n_q^K & \text{if } q > 0. \end{cases} \tag{2.32}$$

Here it is used that the sum of all the upper degrees of the q-simplices is equal to $q+2$ times the number of $(q+1)$-simplices, $\sum_{\sigma \in S_q^K} \deg_U(\sigma) = (q+2)n_{q+1}^K$. This is a result of double counting. Each $(q+1)$-simplex contains $q+2$ q-simplices that are faces of it.

Note that instead of computing the combinatorial Laplacian and the trace or the sum of the eigenvalues, one could just look at the number of simplices instead and compute the same value. While this yields some geometric information, the information could be obtained in a computationally less complex way.

Now looking at the average of the non-zero eigenvalues, one could note that this is just the trace divided by the number of non-zero eigenvalues. Δ_q^K is a self-adjoint, non-negative operator and therefore the algebraic and geometric multiplicities of the eigenvalues are all equal. Resulting in the number of positive eigenvalues being equal to $n_q^K - \beta_q^K$. We can therefore conclude that using the average of the non-zero eigenvalues also encodes the topological information into the result. However, this means that compared to standard homology only the number of simplices is added as information. Using other algorithms to compute the Betti numbers and then adding the information of the number of simplices, should therefore be the same.

This leaves the minimum or maximum non-zero eigenvalues as reasonable aggregation functions. They are often plotted together with the Betti number [44, 46], see Figure 2.14 for the minimum eigenvalue. Note here that the minimum non-zero eigenvalue for dimension 0 and 1 is exactly the same. This is caused by the fact that there are no 2-simplices before time 9. The 0-combinatorial Laplacian is then equal to $[\partial_1]\,[\partial_1]^T$, while the 1-combinatorial Laplacian is equal to $[\partial_1]^T\,[\partial_1]$. For any matrix A, the non-zero eigenvalues of AA^T are the same as the non-zero eigenvalues of A^TA. Therefore, these Laplacians have the same non-zero eigenvalues before time 9. At time 9, the minimum non-zero eigenvalue could have been different, but apparently the method considers the 1 dimensional geometry unchanged.

2.5.4. Persistent Laplacian

Again, instead of looking at one simplicial complex and analyzing its properties, we would like to look at a filtration of complexes, allowing for more information to be captured. To use the combinatorial Laplacian in this setting, the persistent Laplacian was proposed [44].

We define the persistent Laplacian for a simplicial pair $K \hookrightarrow L$ as it can be naturally extended to a filtration by repeatedly using this definition. For that, consider the subspace

$$C_q^{K,L} := \{c \in C_q^L : \partial_q^L(c) \in C_{q-1}^K\} \subseteq C_q^L, \tag{2.33}$$

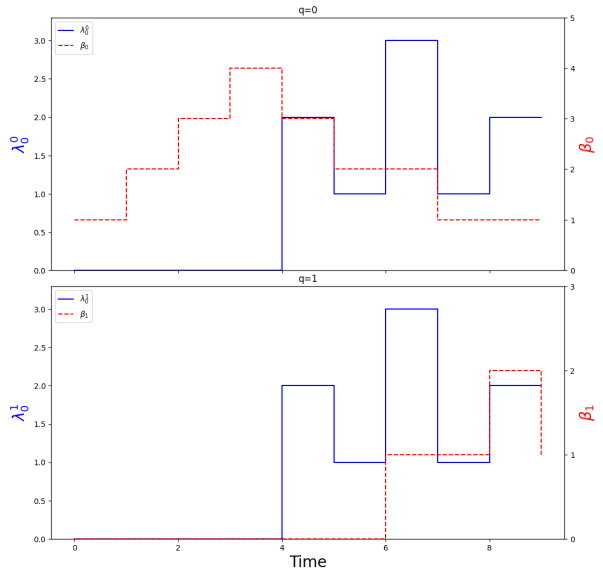


Figure 2.14: Results for the filtration in Figure 2.6. Minimum non-zero eigenvalue of the combinatorial Laplacian λ_0^q in blue, together with the Betti number β_q in red for each time step.

the q-chains in C_q^L that have a boundary in C_{q-1}^K. We use the notation $n_q^{K,L} := \dim C_q^{K,L}$. Finally, let the boundary $\partial_q^{K,L}$ denote the restriction of ∂_q^L to $C_q^{K,L}$.

Definition 2.5.2 (Persistent Laplacian, [44])**.** For a simplicial pair $K \hookrightarrow L$, the q-persistent Laplacian $\Delta_q^{K,L} : C_q^K \to C_q^K$ is defined by

$$\Delta_q^{K,L} = \underbrace{\partial_{q+1}^{K,L}(\partial_{q+1}^{K,L})^*}_{\Delta_{q,+}^{K,L}} + \underbrace{(\partial_q^K)^*\partial_q^K}_{\Delta_{q,-}^K}. \tag{2.34}$$

With $\Delta_{q,+}^{K,L}$ the up persistent Laplacian and $\Delta_{q,-}^K$, the same down-Laplacian from the combinatorial Laplacian of Definition 2.5.1.

Note that this is a self-adjoint, non-negative and compact operator on C_q^K and therefore its eigenvalues are real and non-negative.

A representation such as Theorem 2.5.1 has not been found yet for the persistent Laplacian, however, the kernel result was recently proven [33].

Theorem 2.5.3 ([33], Theorem 2.6)**.** *For each* $q \in \mathbb{N}$, $\beta_q^{K,L} = \dim\ker \Delta_q^{K,L}$.

Note that this means that the analysis of the persistent Laplacian is split into two objectives. First, we want to know the number of zero-eigenvalues, often called harmonic spectra, to describe the topological information. Secondly, the remaining non-zero-eigenvalues, often called non-harmonic spectra, could contain additional useful information. For the non-harmonic spectra, not much is currently known, however it is theorized that they contain geometric information instead of the topological information of the harmonic spectra [46].

In general, a matrix representation of the boundary matrix ∂_q^K is easy to compute as the canonical basis S_q^K can just be used. However, for the persistent Laplacian, we also require a matrix representation of the boundary $\partial_q^{K,L}$. Here, the basis is not necessarily trivial. A cycle c in C_q^L, may need to consist of a linear combination of multiple simplices of S_q^L, in order for the boundary to be in C_q^K. This complication is highlighted in the following example.

Example 2.5.2. We first highlight an example of an easy to compute persistent Laplacian. Consider the persistent Betti number $\beta_1^{6,9}$. In order to calculate this Betti number with the persistent Laplacian, we need a basis for $C_2^{6,9}$. Note that the boundary of the only 2-simplex in K_9 is contained in C_2^6. Therefore, $C_2^{6,9} = C_2^9$, which means that the canonical basis available for C_2^9 can also be used for $C_2^{6,9}$. Now the computation is very similar to the one for the combinatorial Laplacian done in Example 2.5.1.

A harder calculation is needed when trying to calculate the Betti number $\beta_0^{2,7}$. Here a basis for $C_1^{2,7}$ is not trivial as the boundaries of $03, 13$ and 23 are not contained in C_0^2. However, the boundaries of 01, $03-13$ and $03-23$ all are. From these vectors, an orthonormal basis needs to be made, which is not necessarily unique. One way of obtaining such a basis is using the Gram-Schmidt procedure on a linearly independent set. Using the set of boundaries mentioned before, note that 01 and $03-13$ are already orthogonal, however $03-13$ needs to be normalized. This yields $v_1 = 01$ and $v_2 = \frac{1}{\sqrt{2}}(03-13)$. Furthermore, $03-23$ is also already orthogonal to 01. Therefore, the only thing that remains to be done, is to orthogonalize v_2 and $03-23$ and normalize the outcome. Following Gram-Schmidt and writing the vectors in the canonical basis, we obtain,

$$\tilde{v_3} = \begin{matrix} 01 \\ 03 \\ 13 \\ 23 \end{matrix} \begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \\ 0 \end{pmatrix} - \left\langle \begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ -1 \end{pmatrix} \right\rangle \begin{pmatrix} 0 \\ 1 \\ 0 \\ -1 \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{1}{2} \\ \frac{1}{2} \\ -1 \end{pmatrix}.$$

Normalizing this vector yields the following basis $\{01, \frac{1}{\sqrt{2}}(03-13), \frac{1}{\sqrt{6}}(03+13-2\cdot 23)\}$. Using this basis for the matrix representation of the boundary $\partial_1^{2,7}$, we obtain

$$\left[\partial_1^{2,7}\right] = \begin{matrix} \\ \bar{0} \\ \bar{1} \\ \bar{2} \end{matrix} \begin{matrix} 01 & \frac{1}{\sqrt{2}}(03-13) & \frac{1}{\sqrt{6}}(03+13-2\cdot 23) \end{matrix} \begin{pmatrix} -1 & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} \\ 1 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} \\ 0 & 0 & \frac{2}{\sqrt{6}} \end{pmatrix}.$$

The persistent Laplacian then follows,

$$[\Delta_0^{2,7}] = \left[\partial_1^{2,7}\right]\left[\partial_1^{2,7}\right]^T = \frac{1}{3}\begin{pmatrix} 5 & -4 & -1 \\ -4 & 5 & -1 \\ -1 & -1 & 2 \end{pmatrix}.$$

Computing the eigenvalues of this matrix yields $\lambda_1 = 3, \lambda_2 = 1, \lambda_3 = 0$. Therefore, only one zero-eigenvalue is present and we can conclude $\beta_0^{2,7} = 1$.

Using the Schur complement

From Example 2.5.2, we can see that computing the persistent Laplacian can be quite complex. [33] propose an algorithm to determine the matrix representation of $\Delta_q^{K,L}$ without needing to do Gramm Schmidt. However, they note that this algorithm has a worst case time complexity of $O\left(n_q^L (n_{q+1}^L)^2 + (n_{q+1}^L)^3 + (n_q^K)^2\right)$, making it still quite slow.

To improve on this, they give a faster algorithm, that relies on the Schur complement. In [33], the authors prove that the persistent up-Laplacian $\Delta_{q,+}^{K,L}$ can be defined as a Schur complement of the combinatorial up-Laplacian of L. Before stating this theorem, we need some notation. For an integer n, we write $[n] = \{1, 2, \ldots, n\}$. Furthermore for a matrix $M \in \mathbb{R}^{n\times m}$, some set of row indices $I_r \subseteq [n]$ and some set of column indices $I_c \subseteq [m]$, we refer to $M(I_r, I_c)$ as the submatrix of M that contains the rows and columns with indices I_r and I_c, respectively.

Theorem 2.5.4 ([33], Theorem 4.6)**.** *Let $K \hookrightarrow L$ be a simplicial pair. Assume that $n_q^K < n_q^L$. Let $I_q^{K,L} := [n_q^L] \setminus [n_q^K]$. Then,*

$$[\Delta_{q,+}^{K,L}] = [\Delta_{q,+}^L]/[\Delta_{q,+}^L](I_q^{K,L}, I_q^{K,L}). \tag{2.35}$$

Note that in the case of $n_q^K = n_q^L$, we have that $C_q^K = C_q^L$ and therefore any boundary of a vector $c \in C_{q+1}^L$ is in C_q^K. Which means that $C_{q+1}^{K,L} = C_{q+1}^L$ and therefore the persistent up-Laplacian is equal to the combinatorial Laplacian of L, $\Delta_{q,+}^{K,L} = \Delta_{q,+}^L$.

Using the Schur complement representation of the persistent up-Laplacian, [33] note that the time complexity for computing the persistent Laplacian is reduced to worst case $O\left((n_q^L)^3 + n_{q+1}^L\right)$. Therefore, the new algorithm is faster when $n_q^L = O(n_{q+1}^L)$, which they claim is often the case as, for example, the Cech filtration adheres to this condition.

They also note that, for computing persistent Betti numbers, the rank of the null space needs to be determined, which can be done in $O\left((n_q^L)^3\right)$ time. Resulting in a total time complexity for the computation of the persistent Betti number using the persistent Laplacian of $O\left((n_q^L)^3 + n_{q+1}^L\right)$. They compare this to the standard algorithm for computing persistent Betti numbers, which they say has a time complexity of $O\left((n_q^L)^2 n_{q+1}^L + (n_{q-1}^K)^2 n_q^K\right)$ and conclude that the new algorithm is faster if again, $n_q^L = O(n_{q+1}^L)$.

On the other hand, the authors of [16] note that often the persistent Betti number for every combination of birth and death times in multiple dimensions, needs to be calculated. They write that doing this with the persistent Laplacian is time consuming. They compute that for a simplicial filtration of N complexes, where a single simplex is added in each step and the first complex contains only one simplex, the time complexity of the algorithm using the Schur complement representation of the persistent Laplacian is $O\left(N^4/(q+1)\right)$ with q the maximum dimension that needs to be calculated. Comparing this to the previously discussed standard algorithm time complexity of $O(N^3)$, they conclude that the new time complexity is asymptotically larger.

3

Extending persistent barcodes

To use the information captured by the Laplacian, in this chapter, a method to extend persistent barcodes is introduced. In order to do that, first a method to visualize the persistent Laplacian is proposed. This gives a visual argument for the interest in the persistent multiplicity operator, see Equation (3.1) and how it could be used to extend barcodes. In the next section, a new formulation of the persistent Laplacian is obtained and analyzed, which allows for easier algebraic analysis of the multiplicity operator in the succeeding section. Furthermore, it gives an interpretation of the features tracked by the persistent Laplacian. In the third section, the multiplicity operator is split into two parts and each part is individually analyzed. Finally, in this section the full multiplicity operator is discussed and analyzed, concluding with a discussion of its eigenvalues and their meaning, which gives rise to the adjusted multiplicity operator, see Equation (3.32). A proposed usage of these operators is then discussed in Section 3.4. To efficiently calculate the matrix representation of the multiplicity operator, an algorithm is proposed in the final section.

3.1. Visualizing the persistent Laplacian

The combinatorial Laplacian was first visualized using a heat map in Figure 2.13. The same can be done for the persistent Laplacian, but it requires plotting a lot of different images, see Figure 3.1 for dimension 0. Note that in this Figure, on the diagonal are the combinatorial Laplacians of Figure 2.13. While this representation is not very clear, it may provide some intuition on the functioning of the operator.

To make it more clear, we again summarize the matrix by a single number using its eigenvalues. The same previously discussed aggregation functions can be used. For the persistent Laplacian a representation such as Equation (2.30) has not yet been found. Therefore, looking at the sum or average of the non-zero eigenvalues may still provide more information than just the number of simplices. What we do know is that, by the same reasoning, the information added by the average over the sum is determined by the persistent Betti number and the number of simplices.

A few of these aggregation functions have been plotted in Figure 3.2. Here every matrix seen in Figure 3.1 is made into a rectangle with color equal to the aggregation function applied to the non-zero eigenvalues of the persistent Laplacian $\Delta_q^{s,t}$, with s and t equal to the x and y coordinates of the bottom left point of the rectangle, respectively. The intervals of the persistent barcode are also visualized as points to highlight their effect on the persistent Laplacian.

To clarify the representation, we look at an example. The point $(2, 7)$ in dimension 0, corresponds to the persistent interval of the connected component created by vertex 2 in the filtration of Figure 2.6. By looking at the rectangle to the top right of this point in Figure 3.2, we can see that the persistent Laplacian $\Delta_0^{2,7}$ has 2 non-zero eigenvalues. The minimum eigenvalue is equal to 1 and the maximum is equal to 3 making their sum equal to 4. Finally, we can see that the persistent Betti number $\beta_0^{2,7} = 1$.

Now looking at the plot of Figure 3.2 in more detail, we can see the effect of the standard persistent barcode on the Laplacian and the Betti numbers. In the right most plot of the figure, the intervals, represented by points, are when the Betti number changes with respect to the Betti numbers of the previous end time and the previous start time. This results in the points seemingly having effect on the

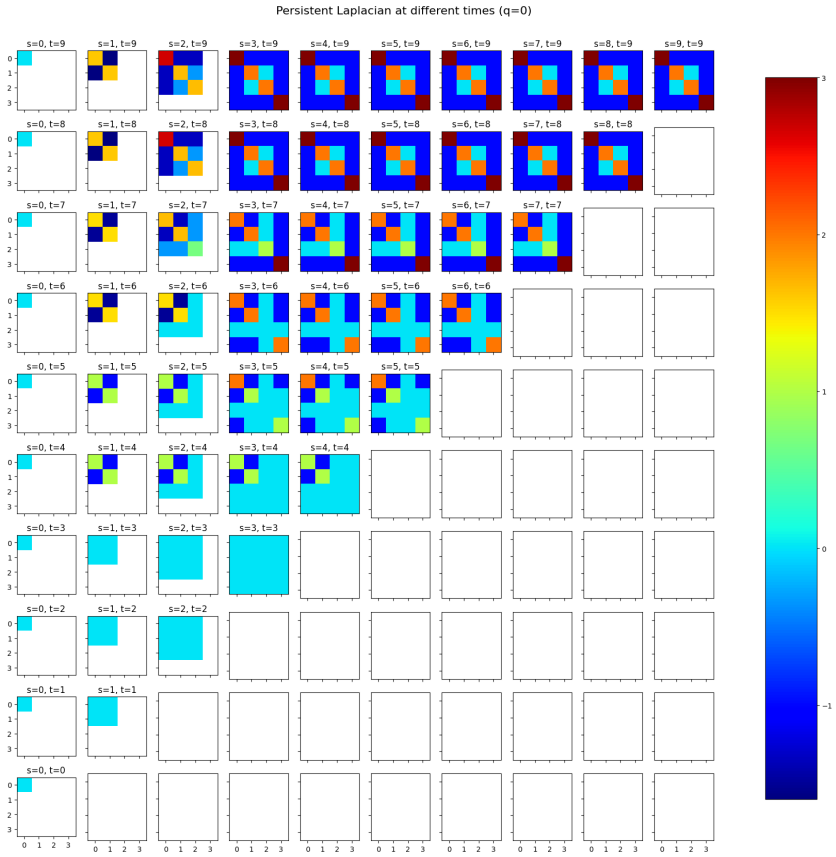


Figure 3.1: Heatmaps of a matrix representation of the 0-persistent Laplacian $\left[\Delta_0^{s,t}\right]$ for different start times s and end times t for the filtration visualized in Figure 2.6.

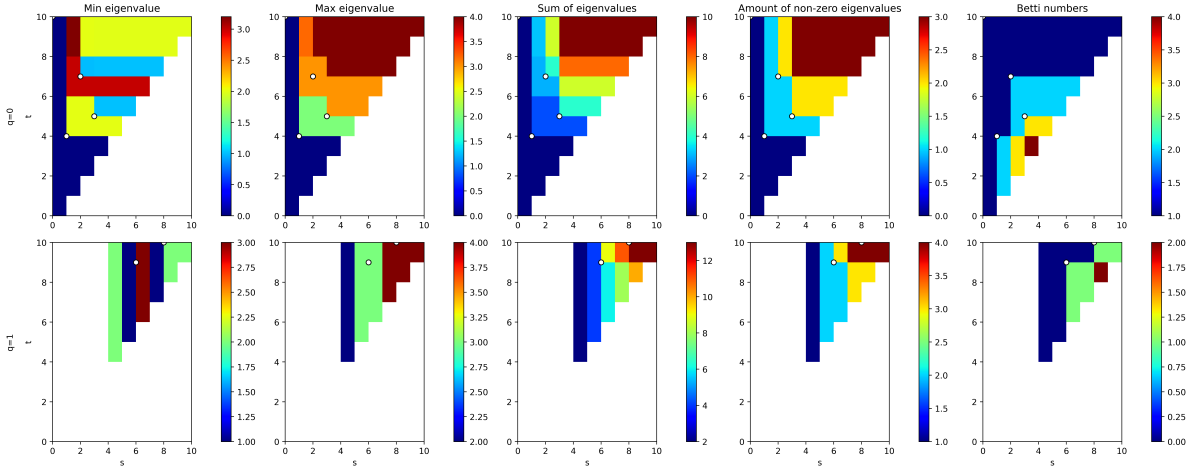


Figure 3.2: Representation of aggregation functions applied to the eigenvalues of the persistent Laplacian $\left[\Delta_q^{s,t}\right]$ for different start times s on the x-axis and end times t on the y-axis for the filtration visualized in Figure 2.6. White dots represent the standard persistent barcode.

rectangles below and to the right of it and in between. Whenever two of these ”effects” overlap the Betti number increases.

Something similar can be said when looking at the sum of the eigenvalues of the persistent Laplacian. Every point that corresponds to an interval affects the rectangles above and to the right of it and in between. Again there seems to be some extra effect whenever these areas overlap. The fact that these triangular patterns appear here is no surprise as whenever the Betti number decreases, a zero eigenvalue has to become positive. This results in more non-zero eigenvalues, so some effect on the sum should be expected.

What is more interesting is that there are more points that seem to satisfy the criteria of creating an effect. Looking at the plot of the sum of eigenvalues, in $(1, 6)$ and $(1, 8)$ a similar pattern is observed. Therefore, these points seem to be of relevance to the Laplacian as well. Although difficult to see now, it will become clear that these points refer to new paths appearing between the vertices. $(1, 6)$ looks at the path between the simplex that appeared at 1, so vertex 1, and the vertices that were present before it, in this case only 0. At 6 the path $03 - 13$ becomes available, so here we see an effect in the persistent Laplacian. At time 8, the path $02 + 23 - 13$ appears for the first time, therefore we also have an effect at $(1, 8)$.

In order to calculate the location of these points, the same operation that is used to find the persistent intervals from the persistent Betti numbers can be used. For persistent intervals, Equation (2.25) was used to find their multiplicity and only intervals having positive multiplicity are considered part of the barcode. Therefore, it is proposed to use the same equation on the persistent Laplacian, however because the effect is upward instead of downward in the Figure, the signs are swapped. This comes with several complications and the following equation is also not well defined yet. However, to give the idea, it is stated here.

$$M_q^{s,t} := \left(\Delta_q^{s,t} - \Delta_q^{s,t-1}\right) - \left(\Delta_q^{s-1,t} - \Delta_q^{s-1,t-1}\right). \tag{3.1}$$

Besides the effect in the plot, one more motivation is given for the relevance of this operator. For most methods in TDA, persistent Betti numbers are not directly used, instead only the intervals defined by the multiplicity equation are used. The topological interpretation of the persistent Laplacian $\Delta_q^{s,t}$ yields that the kernel corresponds to the persistent Betti number $\beta_q^{s,t}$. Therefore, it would make sense to not directly use the persistent Laplacian and instead first apply the multiplicity operation on it. This operation would be described by Equation (3.1).

Assuming this all works, we would get that the trace of the new multiplicity operator follows the multiplicity equation as it is linear. Therefore, computing this trace gives us the points we are looking for, together with some value. These can then be used to extend the standard persistent barcode.

3.2. Laplacians for simplicial pairs

In order to better understand the persistent Laplacian and the effect of the multiplicity operation, it is useful to rewrite the equation using the Schur complement formula of Theorem 2.5.4. The next Theorem introduces the matrices $B_{q,1}^{K,L}$ and $B_{q,2}^{K,L}$, which are important for the remainder of the thesis. After that, an interpretation of the new representation is discussed in the form of Corollary 3.2.3. This shifts the focus to understanding the kernel of $B_{q,2}^{K,L}$, which results in the formulation of Lemma 3.2.4.

One of the main concepts used to prove the results is to exploit the structure of the matrix representation of the boundary operator. For a simplicial pair $K \hookrightarrow L$ and some dimension q, let $I_{q+1}^{K,L} = [n_{q+1}^L] \setminus [n_{q+1}^K]$. Partitioning the matrix representation of $[\partial_{q+1}^L]$ into four blocks, we can write,

$$[\partial_{q+1}^L] = \begin{bmatrix} [\partial_{q+1}^L]([n_q^K], [n_{q+1}^K]) & [\partial_{q+1}^L](I_q^{K,L}, [n_{q+1}^K]) \\ [\partial_{q+1}^L]([n_q^K], I_{q+1}^{K,L}) & [\partial_{q+1}^L](I_q^{K,L}, I_{q+1}^{K,L}) \end{bmatrix}.$$

Note that by definition the top left block is exactly equal to $[\partial_{q+1}^L]([n_q^K], [n_{q+1}^K]) = [\partial_{q+1}^K]$. Furthermore, each column in the boundary matrix corresponds to a $(q+1)$-simplex in L, where the boundary of the simplex is described by the rows, such that each row corresponds to a q-simplex. Note that, because K and L are simplicial complexes, the boundary of a $(q+1)$-simplex in L, which was also part of K cannot contain q-simplices that were not in K. Therefore, columns corresponding to these $(q+1)$-simplices can only be non-zero in rows corresponding to q-simplices in K. Hence, the bottom left block of the

matrix must be zero $[\partial_{q+1}^L]([n_q^K], I_{q+1}^{K,L}) = 0$. Which yields the following structure,

$$[\partial_{q+1}^L] = \begin{bmatrix} [\partial_{q+1}^K] & [\partial_{q+1}^L](I_q^{K,L}, [n_{q+1}^K]) \\ 0 & [\partial_{q+1}^L](I_q^{K,L}, I_{q+1}^{K,L}) \end{bmatrix}.$$

Theorem 3.2.1. *Let K and L be simplicial complexes, such that $K \subseteq L$. The matrix representation of the persistent up-Laplacian can be expressed as*

$$[\Delta_{q,+}^{K,L}] = [\Delta_{q,+}^K] + B_{q,1}^{K,L}(I - (B_{q,2}^{K,L})^\dagger B_{q,2}^{K,L})(B_{q,1}^{K,L})^T, \tag{3.2}$$

where $B_{q,1}^{K,L} := [\partial_{q+1}^L]([n_q^K], I_{q+1}^{K,L})$, $B_{q,2}^{K,L} := [\partial_{q+1}^L](I_q^{K,L}, I_{q+1}^{K,L})$ and † the Moore-Penrose inverse.

Proof. Note that we can write the matrix representation of the $q+1$ boundary operator on L as:

$$[\partial_{q+1}^L] = \begin{bmatrix} [\partial_{q+1}^K] & B_{q,1}^{K,L} \\ 0 & B_{q,2}^{K,L} \end{bmatrix}. \tag{3.3}$$

We therefore get that the combinatorial up-Laplacian $[\Delta_{q,+}^L] = [\partial_{q+1}^L][\partial_{q+1}^L]^T = \begin{bmatrix} [\partial_{q+1}^K][\partial_{q+1}^K]^T + B_{q,1}^{K,L}(B_{q,1}^{K,L})^T & B_{q,1}^{K,L}(B_{q,2}^{K,L})^T \\ B_{q,2}^{K,L}(B_{q,1}^{K,L})^T & B_{q,2}^{K,L}(B_{q,2}^{K,L})^T \end{bmatrix}$. Next the Schur complement representation of Theorem 2.5.4 is used. We get,

$$\begin{aligned} [\Delta_{q,+}^{K,L}] &= [\Delta_{q,+}^L]/([\Delta_{q,+}^L](I_q^{K,L}, I_q^{K,L})) \\ &= [\partial_{q+1}^K][\partial_{q+1}^K]^T + B_{q,1}^{K,L}(B_{q,1}^{K,L})^T - B_{q,1}^{K,L}(B_{q,2}^{K,L})^T \left[B_{q,2}^{K,L}(B_{q,2}^{K,L})^T\right]^\dagger B_{q,2}^{K,L}(B_{q,1}^{K,L})^T \\ &= [\Delta_{q,+}^K] + B_{q,1}^{K,L}\left(I - (B_{q,2}^{K,L})^T \left[B_{q,2}^{K,L}(B_{q,2}^{K,L})^T\right]^\dagger B_{q,2}^{K,L}\right)(B_{q,1}^{K,L})^T. \end{aligned}$$

We now first use the property of the pseudo inverse described by Equation (2.7) and rewrite using $Q = (B_{q,2}^{K,L})^\dagger B_{q,2}^{K,L}$, to get,

$$\begin{aligned} [\Delta_{q,+}^{K,L}] &= [\Delta_{q,+}^K] + B_{q,1}^{K,L}\left(I - (B_{q,2}^{K,L})^T((B_{q,2}^{K,L})^\dagger)^T(B_{q,2}^{K,L})^\dagger B_{q,2}^{K,L}\right)(B_{q,1}^{K,L})^T \\ &= [\Delta_{q,+}^K] + B_{q,1}^{K,L}\left(I - Q^T Q\right)(B_{q,1}^{K,L})^T. \end{aligned}$$

Finally, the fact that Q is symmetric and idempotent, see Equations (2.4) and (2.9), can be used to obtain,

$$\begin{aligned} [\Delta_{q,+}^{K,L}] &= [\Delta_{q,+}^K] + B_{q,1}^{K,L}\left(I - Q\right)(B_{q,1}^{K,L})^T \\ &= [\Delta_{q,+}^K] + B_{q,1}^{K,L}(I - (B_{q,2}^{K,L})^\dagger B_{q,2}^{K,L})(B_{q,1}^{K,L})^T. \end{aligned}$$

□

As a direct consequence of the previous Theorem, we get the following Corollary for the full persistent Laplacian.

Corollary 3.2.2. *Let K and L be simplicial complexes, such that $K \subseteq L$. Then,*

$$[\Delta_q^{K,L}] = [\Delta_q^K] + B_{q,1}^{K,L}(I - (B_{q,2}^{K,L})^\dagger B_{q,2}^{K,L})(B_{q,1}^{K,L})^T, \tag{3.4}$$

where $B_{q,1}^{K,L} := [\partial_{q+1}^L]([n_q^K], I_{q+1}^{K,L})$, $B_{q,2}^{K,L} := [\partial_{q+1}^L](I_q^{K,L}, I_{q+1}^{K,L})$, $I_{q+1}^{K,L} = [n_{q+1}^L] \setminus [n_{q+1}^K]$ and † the Moore-Penrose inverse.

Proof. By definition, the persistent Laplacian is $[\Delta_q^{K,L}] = [\Delta_{q,-}^K] + [\Delta_{q,+}^{K,L}]$ and the combinatorial Laplacian is $[\Delta_q^K] = [\Delta_{q,-}^K] + [\Delta_{q,+}^K]$. Substituting Equation (3.2) into the definition of the persistent Laplacian, we obtain,

$$\begin{aligned} [\Delta_q^{K,L}] &= [\Delta_{q,-}^K] + [\Delta_{q,+}^K] + B_{q,1}^{K,L}(I - (B_{q,2}^{K,L})^\dagger B_{q,2}^{K,L})(B_{q,1}^{K,L})^T \\ &= [\Delta_q^K] + B_{q,1}^{K,L}(I - (B_{q,2}^{K,L})^\dagger B_{q,2}^{K,L})(B_{q,1}^{K,L})^T. \end{aligned}$$

□

In order to understand the new representation of Equation (3.4), we can analyze and rewrite the second part in the following Corollary. Here we see that this part of the equation is completely defined by the following set of vectors: $\{B_{q,1}^{K,L} v : v \in \ker B_{q,2}^{K,L}\}$.

Corollary 3.2.3. *Let $K \hookrightarrow L$ be a simplicial pair. Then,*

$$B_{q,1}^{K,L}(I - (B_{q,2}^{K,L})^{\dagger} B_{q,2}^{K,L})(B_{q,1}^{K,L})^T = \sum_{v \in [\ker B_{q,2}^{K,L}]} B_{q,1}^{K,L} v (B_{q,1}^{K,L} v)^T, \tag{3.5}$$

where $B_{q,1}^{K,L} := [\partial_{q+1}^{L}]([n_q^K], I_{q+1}^{K,L})$, $B_{q,2}^{K,L} := [\partial_{q+1}^{L}](I_q^{K,L}, I_{q+1}^{K,L})$, $I_{q+1}^{K,L} = [n_{q+1}^{L}] \setminus [n_{q+1}^{K}]$, † the Moore-Penrose inverse and $[\ker B_{q,2}^{K,L}]$ an orthonormal basis for the kernel of $B_{q,2}^{K,L}$.

Proof. Consider the singular value decomposition of $B_{q,2}^{K,L} = U \Sigma V^T$, then by Theorem 2.1.1, $(B_{q,2}^{K,L})^{\dagger} = V \Sigma^{\dagger} U^T$, where $\Sigma^{\dagger}$ is the transpose of Σ with every non-zero entry on the diagonal $(\Sigma^{\dagger})_{i,i} = \frac{1}{(\Sigma)_{i,i}}$. We can therefore write, $(B_{q,2}^{K,L})^{\dagger} B_{q,2}^{K,L} = V \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix} V^T$, with $r = \text{Rank}(B_{q,2}^{K,L})$. Now we can simplify

$$\begin{aligned} I - (B_{q,2}^{K,L})^{\dagger} B_{q,2}^{K,L} &= V(I - \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix}) V^T \\ &= \sum_{v \in [\ker B_{q,2}^{K,L}]} v (v)^T, \end{aligned}$$

where the last equality comes from the fact that we are using the vectors in V that correspond to zero singular values. V contains the eigenvectors of $(B_{q,2}^{K,L})^T B_{q,2}^{K,L}$, therefore the vectors that correspond to zero singular values correspond to 0-eigenvectors of $(B_{q,2}^{K,L})^T B_{q,2}^{K,L}$ and are therefore vectors of the kernel of $B_{q,2}^{K,L}$, as long as a orthonormal basis for the kernel is chosen. Finally, we get,

$$\begin{aligned} B_{q,1}^{K,L}(I - (B_{q,2}^{K,L})^{\dagger} B_{q,2}^{K,L})(B_{q,1}^{K,L})^T &= B_{q,1}^{K,L} \left(\sum_{v \in [\ker B_{q,2}^{K,L}]} v (v)^T \right) (B_{q,1}^{K,L})^T \\ &= \sum_{v \in [\ker B_{q,2}^{K,L}]} B_{q,1}^{K,L} v (B_{q,1}^{K,L} v)^T. \end{aligned}$$

□

From this Corollary it becomes clear that, to understand this part of the equation is to understand the set $\{B_{q,1}^{K,L} v : v \in \ker B_{q,2}^{K,L}\}$. In order to do that, we first focus on $\ker B_{q,2}^{K,L}$. The following Lemma tells us that this kernel describes the space of $(q+1)$-simplices in L, which have a boundary in K, but are not part of C_{q+1}^{K}.

Lemma 3.2.4. *Let $K \hookrightarrow L$ be a simplicial pair and let $B_{q,2}^{K,L} := [\partial_{q+1}^{L}](I_q^{K,L}, I_{q+1}^{K,L})$. Then,*

$$C_{q+1}^{K,L} \cong C_{q+1}^{K} \oplus \ker B_{q,2}^{K,L}. \tag{3.6}$$

Proof. For $c \in C_{q+1}^{L}$, we can represent c in the trivial orthonormal basis like $c = \sum_{i=1}^{n_{q+1}^{L}} a_i \sigma_i^{q+1}$ with $a_i \in \mathbb{R}$ and $\sigma_i^{q+1} \in S_{q+1}^{L}$ (the $q+1$ simplices of L). We denote by $c_K \in C_{q+1}^{K}$, the part of c in C_{q+1}^{K}. Using the previous representation, we get $c_K = \sum_{i=1}^{n_{q+1}^{K}} a_i \sigma_i^{q+1}$. Finally, we denote by $c_{K,L}$, the part of c that is not in C_{q+1}^{K}, which can be written as $c_{K,L} = \sum_{i=n_{q+1}^{K}+1}^{n_{q+1}^{L}} a_i \sigma_i^{q+1}$. It now remains to show that $c \in C_{q+1}^{K,L}$ if and only if $c_{K,L} \in \ker B_{q,2}^{K,L}$.

First, note that for $n_q^K < j \leq n_q^L$ and $n_{q+1}^K < i \leq n_{q+1}^L$, $B_{q,2}^{K,L}$ is defined like

$$(B_{q,2}^{K,L})_{j - n_q^K, i - n_{q+1}^K} = \left\langle \partial_{q+1}^{L} \sigma_i^{q+1}, \sigma_j \right\rangle.$$

We now look at its multiplication with $c_{K,L}$ by looking at the entries of the resulting vector:

$$(B_{q,2}^{K,L} \cdot [c_{K,L}])_{j-n_q^K} = \sum_{i=n_{q+1}^K+1}^{n_{q+1}^L} a_i \left\langle \partial_{q+1}^L \sigma_i^{q+1}, \sigma_j \right\rangle.$$

Hence, $c_{K,L} \in \ker B_{q,2}^{K,L}$ if and only if $\sum_{i=n_{q+1}^K+1}^{n_{q+1}^L} a_i \left\langle \partial_{q+1}^L \sigma_i^{q+1}, \sigma_j \right\rangle = 0$ for all $n_q^K < j \leq n_q^L$.

We now show that $c \in C_{q+1}^{K,L}$ is also equivalent to this. Applying the boundary operator to c yields

$$\begin{aligned}
\partial_{q+1}^L(c) &= \sum_{i=1}^{n_{q+1}^L} a_i \partial_{q+1}^L \sigma_i^{q+1} \\
&= \sum_{i=1}^{n_{q+1}^L} a_i \sum_{j=1}^{n_q^L} \left\langle \partial_{q+1}^L \sigma_i^{q+1}, \sigma_j^q \right\rangle \sigma_j^q \\
&= \sum_{j=1}^{n_q^L} \sum_{i=1}^{n_{q+1}^L} a_i \left\langle \partial_{q+1}^L \sigma_i^{q+1}, \sigma_j^q \right\rangle \sigma_j^q.
\end{aligned}$$

Therefore, $c \in C_q^{K,L}$ and equivalently $\partial_{q+1}^L(c) \in C_q^K$ if and only if $\sum_{i=1}^{n_{q+1}^L} a_i \left\langle \partial_{q+1}^L \sigma_i^{q+1}, \sigma_j^q \right\rangle = 0$ for all $n_q^K < j \leq n_q^L$. Finally, note that whenever $j > n_q^K$, we have $\left\langle \partial_{q+1}^L \sigma_i^{q+1}, \sigma_j^q \right\rangle = 0$ for $i \leq n_{q+1}^K$ as a $(q+1)$-simplex in K cannot have a boundary outside K. We therefore get the following equivalence statement.

$$c_{K,L} \in \ker B_{q,2}^{K,L} \iff \sum_{i=n_{q+1}^K+1}^{n_{q+1}^L} a_i \left\langle \partial_{q+1}^L \sigma_i^{q+1}, \sigma_j \right\rangle = 0, \forall j > n_q^K \iff c \in C_{q+1}^{K,L}.$$

□

Turning our attention again to the full set $\{B_{q,1}^{K,L} v : v \in \ker B_{q,2}^{K,L}\}$, we would like to understand the product $B_{q,1}^{K,L} v$. With the new interpretation of the kernel, we get that a vector $v \in \ker B_{q,2}^{K,L}$ can be extended to a chain $c \in C_{q+1}^{K,L}$ which contains no simplices of S_{q+1}^K. The matrix representation of this chain in the trivial basis would be $[c] = \begin{bmatrix} 0 \\ v \end{bmatrix}$. The effect of the boundary ∂_{q+1}^L on this chain c is then completely described by $B_{q,1}^{K,L}$ as using Equation (3.3), we can see that $[\partial_{q+1}^L][c] = \begin{bmatrix} B_{q,1}^{K,L} v \\ 0 \end{bmatrix}$. Here $B_{q,1}^{K,L} v$ can be interpreted as the boundary in K of this chain c. In other words the set $\{B_{q,1}^{K,L} v : v \in \ker B_{q,2}^{K,L}\}$, can be interpreted as new boundaries in K, which come from a combination of simplices in L.

We can now look at the implication of this representation for the trace of the persistent Laplacian,

$$\begin{aligned}
Tr(\Delta_q^{K,L}) &= Tr(\Delta_q^K) + \sum_{v \in [\ker B_{q,2}^{K,L}]} ||B_{q,1}^{K,L} v||_2^2 \\
&= \begin{cases} (q+2)n_{q+1}^K + (q+1)n_q^K + \sum_{v \in [\ker B_{q,2}^{K,L}]} ||B_{q,1}^{K,L} v||_2^2 & \text{if } q > 0. \\ 2n_1^K + \sum_{v \in [\ker B_{q,2}^{K,L}]} ||B_{q,1}^{K,L} v||_2^2 & \text{if } q = 0. \end{cases}
\end{aligned} \tag{3.7}$$

Here Equation (2.32) is used in the second equality.

We would therefore like to find an interpretation of $||B_{q,1}^{K,L} v||_2^2$, which can be done in certain situations. For example, see the two simplicial pairs in Figure 3.3. Note that their topological features are the same as they both start with 2 connected components where one dies in L. Furthermore, note that in L, they both contain a path from 0 to 1 consisting of 1-simplices. For any such path, we can find

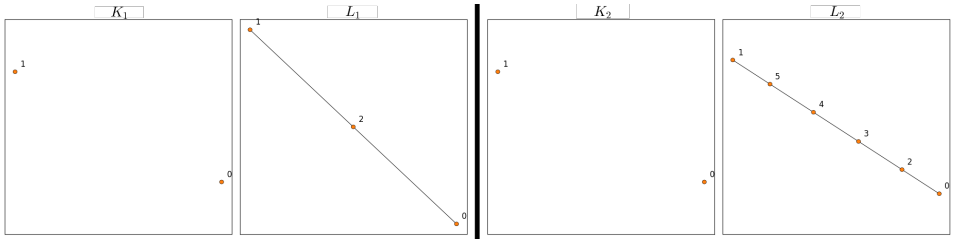


Figure 3.3: Two simplicial pairs $K_1 \hookrightarrow L_1$ and $K_2 \hookrightarrow L_2$.

a $c \in C_1^{K,L}$ by choosing the sign such that the simplices in the middle disappear in the boundary. For the $K_1 \hookrightarrow L_1$ the path would be $c = 02 - 12$, while for the second simplicial pair, we can find the path $c = 02 - 23 + 34 - 45 - 15$. None of these 1-simplices appeared already in K, therefore using Lemma 3.2.4, this path is described by a $v \in \ker B_{q,2}^{K,L}$. Like was said before, $B_{q,1}^{K,L} v$, is the boundary of this chain, which in both of these cases is just $\bar{1} - \bar{0}$, resulting in $||B_{q,1}^{K,L} v||_2^2 = 2$. However, v is not yet normalized. Here lies the difference between the two simplicial pairs. For pair $K_1 \hookrightarrow L_1$, $||v_1||_2 = \sqrt{2}$, while for the pair $K_2 \hookrightarrow L_2$, $||v_2||_2 = \sqrt{5}$. Multiplying with the normalization constant, we get for the first pair $Tr(\Delta_0^{K_1,L_1}) = \frac{2}{2} = 1$, while for the second pair, we get $Tr(\Delta_0^{K_2,L_2}) = \frac{2}{5}$ as the trace of the combinatorial Laplacian is zero in both cases. Hence, using the trace of the persistent Laplacian, we do see a difference between the two pairs.

We can make this case more general, by noting that for $q = 0$, any chain $c \in C_1^L$ corresponding to a path of 1-simplices in L between two 0-simplices of K that does not cross any other 0-simplices of K, has a boundary in K. Therefore, we have $c \in C_1^{K,L}$. However, if it does not contain any 1-simplices already in K, it does not have a part in C_1^K, which means it is completely described by a $v \in \ker B_{q,2}$ according to the previous lemma. The multiplication of v with $B_{q,1}^{K,L}$ then only contains the two 0-simplices in K the path connects. $||B_{q,1}^{K,L} v||_2^2$ is then equal to 2. If v comes from an orthonormal basis instead, it needs to be normalized first, resulting in $\tilde{v} = \frac{1}{||v||_2} v$, this causes $||B_{q,1}^{K,L} \tilde{v}||_2^2 = \frac{2}{||v||_2}$. Therefore, longer paths correspond to a smaller effect on the trace of the persistent Laplacian.

For dimensions higher than 0, similar reasoning is more difficult as paths are harder to define. For example, the boundary of a 2-simplex contains three 1-simplices. Making a path by connecting another 2-simplex to it only cancels one of these simplices. For a selection of these simplices to have a boundary in K requires a very specific structure as many 1-simplices need to be in K already. Therefore most analysis is done in dimension 0, however an interpretation probably exists for higher dimensions as well.

3.3. Persistent Laplacians in filtrations

Having looked at a different representation of the persistent Laplacian, we can now turn to applying it to the multiplicity equation (3.1). This is done in 3 steps. First we look at the subtraction $\Delta_q^{s,t} - \Delta_q^{s,t-1}$, which is relatively easy as both persistent Laplacians operate on the same space. The first subsection finds some properties of this subtraction, which results in the notion of the horizontal operator. Afterwards, the subtraction $\Delta_q^{s,t} - \Delta_q^{s-1,t}$ is discussed. While it is not explicitly part of the multiplicity equation, the idea of subtracting persistent Laplacians with different start times needs to be addressed. This is harder as $\Delta_q^{s,t}$ and $\Delta_q^{s-1,t}$ operate on a different subspaces. After proposing a way to solve this issue, the same structure of lemmas and theorem describing the properties is used to characterize the notion of the vertical operator. Finally the full multiplicity equation is used and its properties are again discussed.

In this section and for the rest of the thesis some new notation is used. For a filtration of simplicial complexes $\{K_t\}_{0 \le t \le T}$, where $K_t \hookrightarrow K_{t+1}$, we write $I_q^{s,t} := [n_q^t] \setminus [n_q^s]$. Furthermore, to simplify notation, we write $B_{q,1}^{s,t} := B_{q,1}^{K_s,K_t}$, $B_{q,2}^{s,t} := B_{q,2}^{K_s,K_t}$, $n_q^s = n_q^{K_s}$, $C_q^s := C_q^{K_s}$, $C_q^{s,t} := C_q^{K_s,K_t}$ and $\Delta_q^s := \Delta_q^{K_s}$.

3.3.1. Horizontal operator

In this subsection, we look at the subtraction $\mathcal{H}_q^{s,t} := \Delta_q^{s,t-1} - \Delta_q^{s,t}$. We look at a Lemma that simplifies the matrix representation of $\mathcal{H}_q^{s,t}$ equation. After that, we look at its implications.

Lemma 3.3.1. *In a simplicial filtration $\{K_t\}_{0\le t\le T}$, we have that*

$$[\mathcal{H}_q^{s,t}] = [\Delta_q^{s,t}] - [\Delta_q^{s,t-1}] = B_{q,1}^{s,t}\left(\begin{bmatrix}(B_{q,2}^{s,t-1})^\dagger B_{q,2}^{s,t-1} & 0\\ 0 & I_{n_{q+1}^t - n_{q+1}^{t-1}}\end{bmatrix} - (B_{q,2}^{s,t})^\dagger B_{q,2}^{s,t}\right)(B_{q,1}^{s,t})^T. \tag{3.8}$$

Proof. In a simplicial filtration, we can write,

$$B_{q,1}^{s,t} = \begin{bmatrix}[\partial_{q+1}^t]([n_q^s], I_{q+1}^{s,t-1}) & [\partial_{q+1}^t]([n_q^s], I_{q+1}^{t-1,t})\end{bmatrix} = \begin{bmatrix}B_{q,1}^{s,t-1} & [\partial_{q+1}^t]([n_q^s], I_{q+1}^{t-1,t})\end{bmatrix}. \tag{3.9}$$

We now apply equation (3.4) twice, where we see that $[\Delta_q^s]$ disappears. Furthermore, we extend $B_{q,1}^{s,t-1}$ to $B_{q,1}^{s,t}$, without changing the result, like:

$$\begin{aligned}[\Delta_q^{s,t}] - [\Delta_q^{s,t-1}] =& \begin{bmatrix}B_{q,1}^{s,t-1} & [\partial_{q+1}^t]([n_q^s], I_{q+1}^{t-1,t})\end{bmatrix}\left(I - (B_{q,2}^{s,t})^\dagger B_{q,2}^{s,t}\right)\begin{bmatrix}(B_{q,1}^{s,t-1})^T\\ ([\partial_{q+1}^t]([n_q^s], I_{q+1}^{t-1,t}))^T\end{bmatrix}\\ &- \begin{bmatrix}B_{q,1}^{s,t-1} & [\partial_{q+1}^t]([n_q^s], I_{q+1}^{t-1,t})\end{bmatrix}\left(I - \begin{bmatrix}(B_{q,2}^{s,t-1})^\dagger B_{q,2}^{s,t-1} & 0\\ 0 & I_{n_{q+1}^t - n_{q+1}^{t-1}}\end{bmatrix}\right)\begin{bmatrix}(B_{q,1}^{s,t-1})^T\\ ([\partial_{q+1}^t]([n_q^s], I_{q+1}^{t-1,t}))^T\end{bmatrix}\\ =& B_{q,1}^{s,t}\left(\begin{bmatrix}(B_{q,2}^{s,t-1})^\dagger B_{q,2}^{s,t-1} & 0\\ 0 & I_{n_{q+1}^t - n_{q+1}^{t-1}}\end{bmatrix} - (B_{q,2}^{s,t})^\dagger B_{q,2}^{s,t}\right)(B_{q,1}^{s,t})^T.\end{aligned}$$

□

In order to prove the main theorem in this setting, we need the following important Lemma.

Lemma 3.3.2. *In a simplicial filtration $\{K_t\}_{0\le t\le T}$, we have that*

$$(B_{q,2}^{s,t})^\dagger B_{q,2}^{s,t} = (B_{q,2}^{s,t})^\dagger B_{q,2}^{s,t}\begin{bmatrix}(B_{q,2}^{s,t-1})^\dagger B_{q,2}^{s,t-1} & 0\\ 0 & I_{n_{q+1}^t - n_{q+1}^{t-1}}\end{bmatrix} = \begin{bmatrix}(B_{q,2}^{s,t-1})^\dagger B_{q,2}^{s,t-1} & 0\\ 0 & I_{n_{q+1}^t - n_{q+1}^{t-1}}\end{bmatrix}(B_{q,2}^{s,t})^\dagger B_{q,2}^{s,t}. \tag{3.10}$$

Proof. In a simplicial filtration, we can write,

$$B_{q,2}^{s,t} = \begin{bmatrix}[\partial_{q+1}^t](I_q^{s,t-1}, I_{q+1}^{s,t-1}) & [\partial_{q+1}^t](I_q^{s,t-1}, I_{q+1}^{t-1,t})\\ 0 & [\partial_{q+1}^t](I_q^{t-1,t}, I_{q+1}^{t-1,t})\end{bmatrix} = \begin{bmatrix}B_{q,2}^{s,t-1} & [\partial_{q+1}^t](I_q^{s,t-1}, I_{q+1}^{t-1,t})\\ 0 & [\partial_{q+1}^t](I_q^{t-1,t}, I_{q+1}^{t-1,t})\end{bmatrix}. \tag{3.11}$$

Hence, for a vector $v \in \ker B_{q,2}^{s,t-1}$, extending the vector with zeros, yields:

$$B_{q,2}^{s,t} v = \begin{bmatrix}B_{q,2}^{s,t-1} & [\partial_{q+1}^t](I_q^{s,t-1}, I_{q+1}^{t-1,t})\\ 0 & [\partial_{q+1}^t](I_q^{t-1,t}, I_{q+1}^{t-1,t})\end{bmatrix}\begin{bmatrix}v\\ 0\end{bmatrix} = 0.$$

Now note that $v = \begin{bmatrix}v_1\\ v_2\end{bmatrix} \in \ker\begin{bmatrix}(B_{q,2}^{s,t-1})^T B_{q,2}^{s,t-1} & 0\\ 0 & I_{n_{q+1}^t - n_{q+1}^{t-1}}\end{bmatrix} \Rightarrow v_2 = 0 \Rightarrow v \in \ker B_{q,2}^{s,t-1}$, hence we have that,

$$\ker\begin{bmatrix}(B_{q,2}^{s,t-1})^T B_{q,2}^{s,t-1} & 0\\ 0 & I_{n_{q+1}^t - n_{q+1}^{t-1}}\end{bmatrix} \subseteq \ker B_{q,2}^{s,t}.$$

Finally, we use Equations (2.8) and (2.2) to get,

$$\begin{aligned}(B_{q,2}^{s,t})^\dagger B_{q,2}^{s,t} &= (B_{q,2}^{s,t})^\dagger B_{q,2}^{s,t}\begin{bmatrix}(B_{q,2}^{s,t-1})^T B_{q,2}^{s,t-1} & 0\\ 0 & I_{n_{q+1}^t - n_{q+1}^{t-1}}\end{bmatrix}^\dagger\begin{bmatrix}(B_{q,2}^{s,t-1})^T B_{q,2}^{s,t-1} & 0\\ 0 & I_{n_{q+1}^t - n_{q+1}^{t-1}}\end{bmatrix}\\ &= (B_{q,2}^{s,t})^\dagger B_{q,2}^{s,t}\begin{bmatrix}(B_{q,2}^{s,t-1})^\dagger B_{q,2}^{s,t-1} & 0\\ 0 & I_{n_{q+1}^t - n_{q+1}^{t-1}}\end{bmatrix}.\end{aligned}$$

The final equality in equation (3.10) can be obtained by taking the transpose on both sides and realizing that the relevant matrices are symmetric. □

We are now ready to state and prove the main theorem. Note that for a subspace $V \subseteq S$ of some vector space S , $V^\perp = \{v \in S : \langle v, w\rangle = 0\ \forall w \in V\}$ refers to the orthogonal complement of this vector space.

Theorem 3.3.3. *Let* $\Psi_h := \ker B_{q,2}^{s,t} \cap \left(\ker \begin{bmatrix} B_{q,2}^{s,t-1} & 0 \\ 0 & I_{n_{q+1}^t - n_{q+1}^{t-1}} \end{bmatrix}\right)^{\perp}$ *and* $[\Psi_h]$ *represent an orthonormal basis for* Ψ_h*. We get,*

$$[\mathcal{H}_q^{s,t}] = \sum_{v \in [\Psi_h]} B_{q,1}^{s,t} v \left(B_{q,1}^{s,t} v\right)^T. \tag{3.12}$$

Proof. We can use Lemmas 3.3.1 and 3.3.2, to get,

$$\begin{aligned}[\Delta_q^{s,t}] - [\Delta_q^{s,t-1}] &= B_{q,1}^{s,t} \left(\begin{bmatrix} (B_{q,2}^{s,t-1})^\dagger B_{q,2}^{s,t-1} & 0 \\ 0 & I_{n_{q+1}^t - n_{q+1}^{t-1}} \end{bmatrix} - \begin{bmatrix} (B_{q,2}^{s,t-1})^\dagger B_{q,2}^{s,t-1} & 0 \\ 0 & I_{n_{q+1}^t - n_{q+1}^{t-1}} \end{bmatrix} (B_{q,2}^{s,t})^\dagger B_{q,2}^{s,t} \right) (B_{q,1}^{s,t})^T \\ &= B_{q,1}^{s,t} \begin{bmatrix} (B_{q,2}^{s,t-1})^\dagger B_{q,2}^{s,t-1} & 0 \\ 0 & I_{n_{q+1}^t - n_{q+1}^{t-1}} \end{bmatrix} \left(I - (B_{q,2}^{s,t})^\dagger B_{q,2}^{s,t}\right) (B_{q,1}^{s,t})^T.\end{aligned}$$

Note that,

$$\begin{bmatrix} (B_{q,2}^{s,t-1})^\dagger B_{q,2}^{s,t-1} & 0 \\ 0 & I_{n_{q+1}^t - n_{q+1}^{t-1}} \end{bmatrix} = \begin{bmatrix} B_{q,2}^{s,t-1} & 0 \\ 0 & I_{n_{q+1}^t - n_{q+1}^{t-1}} \end{bmatrix}^\dagger \begin{bmatrix} B_{q,2}^{s,t-1} & 0 \\ 0 & I_{n_{q+1}^t - n_{q+1}^{t-1}} \end{bmatrix}.$$

Therefore it projects onto $\left(\ker \begin{bmatrix} B_{q,2}^{s,t-1} & 0 \\ 0 & I_{n_{q+1}^t - n_{q+1}^{t-1}} \end{bmatrix}\right)^{\perp}$. Furthermore, $I - (B_{q,2}^{s,t})^\dagger B_{q,2}^{s,t}$ projects onto $\left(\left(\ker B_{q,2}^{s,t}\right)^\perp\right)^\perp = \ker B_{q,2}^{s,t}$, see Section 2.1. Now note that the projection matrices commute because of Lemma 3.3.2. Therefore, their product is again a projection matrix that projects onto the intersection of the two spaces [36], $\Psi_h := \ker B_{q,2}^{s,t} \cap \left(\ker \begin{bmatrix} B_{q,2}^{s,t-1} & 0 \\ 0 & I_{n_{q+1}^t - n_{q+1}^{t-1}} \end{bmatrix}\right)^{\perp}$.

If we consider $[\Psi_h]$ an orthonormal basis for Ψ_h, we can represent this projection by VV^T, where the columns of V are the vectors in $[\Psi_h]$. This allows us to write,

$$\begin{aligned}[\Delta_q^{s,t}] - [\Delta_q^{s,t-1}] &= B_{q,1}^{s,t} \sum_{v \in [\Psi_h]} v v^T (B_{q,1}^{s,t})^T \\ &= \sum_{v \in [\Psi_h]} B_{q,1} v \left(B_{q,1} v\right)^T.\end{aligned}$$

□

From this Theorem, it becomes clear the matrix $\mathcal{H}_q^{s,t}$ is still symmetric and positive semi definite as it is a sum of outer products. Furthermore, the trace can be calculated as follows, with $||\cdot||_2$ the L^2 norm,

$$\begin{aligned}Tr([\mathcal{H}_q^{s,t}]) &= \sum_{v \in [\Psi_h]} Tr\left(B_{q,1}^{s,t} v \left(B_{q,1}^{s,t} v\right)^T\right) \\ &= \sum_{v \in [\Psi_h]} ||B_{q,1}^{s,t} v||_2^2.\end{aligned} \tag{3.13}$$

Besides this, we can now also say something about the eigenvectors. For a vector $c \in C_q^s$, we have in the trivial basis,

$$[\mathcal{H}_q^{s,t}][c] = \sum_{v \in [\Psi_h]} B_{q,1} v \left(B_{q,1} v\right)^T [c] = \sum_{v \in [\Psi_h]} \left\langle B_{q,1} v, [c] \right\rangle B_{q,1} v. \tag{3.14}$$

Hence the matrix representation of $\mathcal{H}_q^{s,t}$ applied to any vector c, results in a linear combination of vectors of the set $\left\{B_{q,1} v : v \in \left[\ker B_{q,2}^{s,t} \cap \left(\ker B_{q,2}^{s,t-1}\right)^\perp\right]\right\}$. If c would be an eigenvector of $\mathcal{H}_q^{s,t}$, we would need $[\mathcal{H}_q^{s,t}][c] = \lambda[c]$ for some $\lambda \geq 0$. If $\lambda > 0$, we can write $[c] = \sum_{v \in [\Psi_h]} \frac{\langle B_{q,1} v, [c] \rangle}{\lambda} B_{q,1} v$ and therefore $[c]$ would need to be a linear combination of vectors of this set. This allows us to obtain an upper bound

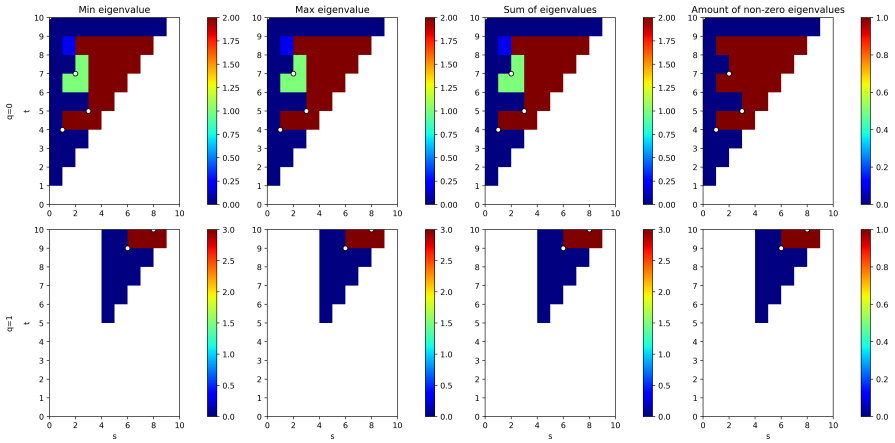


Figure 3.4: Plot of different aggregation functions applied to the eigenvalues of $\mathcal{H}_q^{s,t}$ for the filtration visualized in Figure 2.6. White dots represent the intervals of the standard persistent barcode.

on the number of positive eigenvalues equal to the number of linearly independent vectors of the set. Where the number of independent vectors is again bounded by the total number of vectors in the set.

In the same way as before in Section 3.1, we can plot the eigenvalues of $\mathcal{H}_q^{s,t}$, see Figure 3.4. Here we see that compared to Figure 3.2, the number of non-zero eigenvalues is reduced. For every rectangle with bottom left point (s,t), the effect of the rectangle below it, corresponding to $(s,t-1)$, has been removed. Now, we can clearly see what the new features are in t. However, these features may still have appeared before s already. For example, the rectangle corresponding to $(2,4)$ still has a non-zero eigenvalue of 2, while this effect originates in $(1,4)$.

Nevertheless, looking at the added information over the standard persistent barcode, we can see three more points that seem to be relevant. The points $(1,6)$ and $(1,8)$ were already noted in Section 3.1. However, now we see that $(2,8)$, $(3,6)$ and $(3,7)$ seem to contain some information as well as they are both different from their left adjacent rectangle. Looking at the visualization of the filtration in Figure 2.6 and using the same interpretation as before. These points can be explained by new paths of 1-simplices forming between the new 0-simplex of their starting time. In $(2,8)$, the 0-simplex $\bar{2}$ gets connected in a direct path to one of the 0-simplices that appeared before it. Similarly in $(3,6)$ and $(3,7)$, the 0-simplex $\bar{3}$ gets connected to a 0-simplex that appeared before it.

3.3.2. Vertical operator

In this section, the persistent up-Laplacian $\Delta_{q,+}^{s-1,t}$ is extended to operate on the space C_q^s instead of C_q^{s-1} such that the subtraction $\Delta_{q,+}^{s,t} - \Delta_{q,+}^{s-1,t}$ is well-defined. Extending the down-Laplacian is not discussed. This is due to the fact that in the multiplicity equation (3.1), twice Lemma 3.3.1 can be used, which means the down-Laplacians cancel out and are not needed to solve the multiplicity equation. However, for persistent Laplacians $\Delta_q^{s,t}$, where $s=t$, $\Delta_q^{s,t-1}$ does not exist and therefore the multiplicity equation is not defined. In these cases, including the down-Laplacian could be useful. This is left for future research.

Definition 3.3.1. For a filtration of simplicial complexes $\{K_t\}_{0\le t\le T}$ and $0<s<t\le T$, let $\iota : K_{s-1} \hookrightarrow K_s$ the inclusion map and $f_\iota : C_q^{s-1} \to C_q^s$ the induced linear map. Furthermore, let $(f_\iota)^* : C_q^s \to C_q^{s-1}$ the Hermitian adjoint of f_ι. The persistent up-Laplacian $\Delta_{q,+}^{s-1,t} : C_q^{s-1} \to C_q^{s-1}$ can be extended to $\widetilde{\Delta_{q,+}^{s-1,t}} : C_q^s \to C_q^s$ by $\widetilde{\Delta_{q,+}^{s-1,t}} = f_\iota \circ \Delta_{q,+}^{s-1,t} \circ (f_\iota)^*$.

Note that the matrix representation of this extension is given as follows, where $0_{n\times m} \in \mathbb{R}^{n\times m}$ represents the n times m zero matrix,

$$[\widetilde{\Delta_{q,+}^{s-1,t}}] = \begin{bmatrix} I_{n_q^{s-1}} \\ 0_{(n_q^s - n_q^{s-1})\times n_q^{s-1}} \end{bmatrix} [\Delta_{q,+}^{s-1,t}] \begin{bmatrix} I_{n_q^{s-1}} & 0_{n_q^{s-1}\times(n_q^s - n_q^{s-1})} \end{bmatrix} = \begin{bmatrix} [\Delta_{q,+}^{s-1,t}] & 0_{n_q^{s-1}\times(n_q^s - n_q^{s-1})} \\ 0_{(n_q^s - n_q^{s-1})\times n_q^{s-1}} & 0_{(n_q^s - n_q^{s-1})\times(n_q^s - n_q^{s-1})} \end{bmatrix}. \tag{3.15}$$

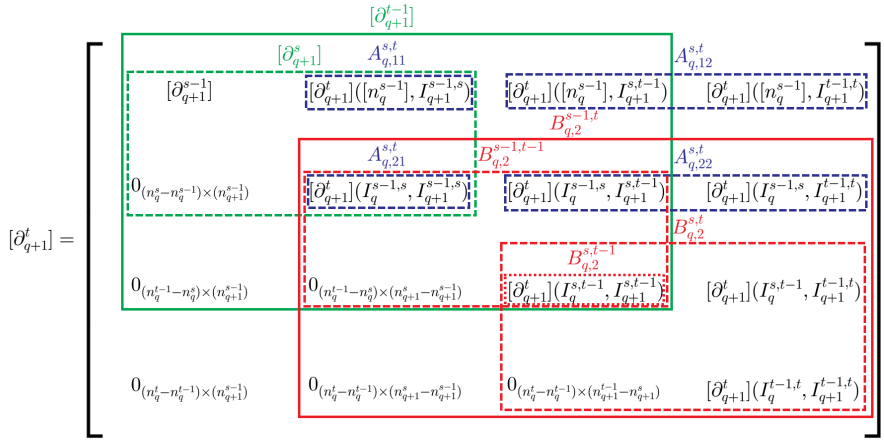


Figure 3.5: The boundary $[\partial_{q+1}^t]$ split into the different submatrices. Each block of the matrix that is zero is denoted by $0_{m\times n}$ for some m and n representing the number of rows and columns that are zero respectively.

For the rest of this section, we focus on the operator $\mathcal{V}_q^{s,t} := \Delta_{q,+}^{s,t} - \widetilde{\Delta_{q,+}^{s-1,t}}$. However, before discussing any of its properties, we need some notation to make the results more readable.

$$A_q^{s,t} := [\partial_{q+1}^t]([n_q^s], I_{q+1}^{s-1,t}) = \begin{bmatrix} A_{q,11}^{s,t} & A_{q,12}^{s,t} \\ A_{q,21}^{s,t} & A_{q,22}^{s,t} \end{bmatrix}, \tag{3.16}$$

$$A_{q,11}^{s,t} := [\partial_{q+1}^t]([n_q^{s-1}], I_{q+1}^{s-1,s}), \qquad A_{q,12}^{s,t} := [\partial_{q+1}^t]([n_q^{s-1}], I_{q+1}^{s,t}),$$

$$A_{q,21}^{s,t} := [\partial_{q+1}^t](I_q^{s-1,s}, I_{q+1}^{s-1,s}), \qquad A_{q,22}^{s,t} := [\partial_{q+1}^t](I_q^{s-1,s}, I_{q+1}^{s,t}),$$

$$B_{q,1}^{s-1,t} = \begin{bmatrix} A_{q,11}^{s,t} & A_{q,12}^{s,t} \end{bmatrix}, \qquad B_{q,1}^{s,t} = \begin{bmatrix} A_{q,12}^{s,t} \\ A_{q,22}^{s,t} \end{bmatrix}.$$

In order to better understand these matrices, we look at how they all come from the boundary $[\partial_{q+1}^t]$ in Figure 3.5.

Adhering to the same structure as the previous section, we can now state and prove the Lemma that simplifies the subtraction. Afterwards, we discuss the implications of this derivation.

Lemma 3.3.4. *In a simplicial filtration $\{K_t\}_{0\le t\le T}$, we have that*

$$[\mathcal{V}_q^{s,t}] = [\Delta_{q,+}^{s,t}] - [\widetilde{\Delta_{q,+}^{s-1,t}}] = A_q^{s,t}\left((B_{q,2}^{s-1,t})^\dagger B_{q,2}^{s-1,t} - \begin{bmatrix} 0 & 0 \\ 0 & (B_{q,2}^{s,t})^\dagger B_{q,2}^{s,t} \end{bmatrix}\right)(A_q^{s,t})^T. \tag{3.17}$$

Where $A_q^{s,t} := [\partial_{q+1}^t]([n_q^s], I_{q+1}^{s-1,t})$.

Proof. In a simplicial filtration, we have

$$[\partial_{q+1}^s] = \begin{bmatrix} [\partial_{q+1}^{s-1}] & A_{q,11}^{s,t} \\ 0 & A_{q,21}^{s,t} \end{bmatrix}.$$

This allows us to rewrite $[\Delta_{q,+}^{s,t}]$, using Theorem 3.2.1

$$\begin{aligned}
[\Delta_{q,+}^{s,t}] &= [\partial_{q+1}^s][\partial_{q+1}^s]^T + B_{q,1}^{s,t}(B_{q,1}^{s,t})^T - B_{q,1}^{s,t}(B_{q,2}^{s,t})^\dagger B_{q,2}^{s,t}(B_{q,1}^{s,t})^T \\
&= \begin{bmatrix} [\partial_{q+1}^{s-1}][\partial_{q+1}^{s-1}]^T + A_{q,11}^{s,t}\left(A_{q,11}^{s,t}\right)^T & A_{q,11}^{s,t}\left(A_{q,21}^{s,t}\right)^T \\ A_{q,21}^{s,t}\left(A_{q,11}^{s,t}\right)^T & A_{q,12}^{s,t}\left(A_{q,12}^{s,t}\right)^T \end{bmatrix} + \begin{bmatrix} A_{q,12}^{s,t}\left(A_{q,12}^{s,t}\right)^T & A_{q,12}^{s,t}\left(A_{q,22}^{s,t}\right)^T \\ A_{q,22}^{s,t}\left(A_{q,12}^{s,t}\right)^T & A_{q,22}^{s,t}\left(A_{q,22}^{s,t}\right)^T \end{bmatrix} - A_q^{s,t}\begin{bmatrix} 0 & 0 \\ 0 & (B_{q,2}^{s,t})^\dagger B_{q,2}^{s,t} \end{bmatrix}(A_q^{s,t})^T \\
&= \begin{bmatrix} [\Delta_{q,+}^{s-1}] & 0 \\ 0 & 0 \end{bmatrix} + A_q^{s,t}\left(I - \begin{bmatrix} 0 & 0 \\ 0 & (B_{q,2}^{s,t})^\dagger B_{q,2}^{s,t} \end{bmatrix}\right)(A_q^{s,t})^T.
\end{aligned}$$

Furthermore, note that we can write,

$$B_{q,2}^{s-1,t} = \begin{bmatrix} A_{q,21}^{s,t} & A_{q,22}^{s,t} \\ 0 & B_{q,2}^{s,t} \end{bmatrix}.$$

Therefore, $\ker B_{q,2}^{s-1,t} \subseteq \ker \begin{bmatrix} A_{q,21}^{s,t} & A_{q,22}^{s,t} \end{bmatrix}$. Using Equation (2.8), we get,

$$\begin{bmatrix} A_{q,21}^{s,t} & A_{q,22}^{s,t} \end{bmatrix} = \begin{bmatrix} A_{q,21}^{s,t} & A_{q,22}^{s,t} \end{bmatrix} (B_{q,2}^{s-1,t})^{\dagger} B_{q,2}^{s-1,t}. \tag{3.18}$$

Now using Theorem 3.2.1 and Equation (3.15), we can rewrite $[\widetilde{\Delta_{q,+}^{s-1,t}}]$,

$$\begin{aligned}
[\widetilde{\Delta_{q,+}^{s-1,t}}] &= \begin{bmatrix} [\Delta_{q,+}^{s-1}] + B_{q,1}^{s-1,t}(I - (B_{q,2}^{s-1,t})^{\dagger} B_{q,2}^{s-1,t})(B_{q,1}^{s-1,t})^{T} & 0 \\ 0 & 0 \end{bmatrix} \\
&= \begin{bmatrix} [\Delta_{q,+}^{s-1}] & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} A_{q,11}^{s,t} & A_{q,12}^{s,t} \\ 0 & 0 \end{bmatrix} (I - (B_{q,2}^{s-1,t})^{\dagger} B_{q,2}^{s-1,t}) \begin{bmatrix} (A_{q,11}^{s,t})^{T} & 0 \\ (A_{q,12}^{s,t})^{T} & 0 \end{bmatrix} \\
&= \begin{bmatrix} [\Delta_{q,+}^{s-1}] & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} A_{q,11}^{s,t} & A_{q,12}^{s,t} \\ A_{q,21}^{s,t} & A_{q,22}^{s,t} \end{bmatrix} (I - (B_{q,2}^{s-1,t})^{\dagger} B_{q,2}^{s-1,t}) \begin{bmatrix} (A_{q,11}^{s,t})^{T} & (A_{q,21}^{s,t})^{T} \\ (A_{q,12}^{s,t})^{T} & (A_{q,22}^{s,t})^{T} \end{bmatrix}.
\end{aligned}$$

This allows for the following computation of the difference,

$$[\Delta_{q,+}^{s,t}] - [\widetilde{\Delta_{q,+}^{s-1,t}}] = A_{q}^{s,t} \left((B_{q,2}^{s-1,t})^{\dagger} B_{q,2}^{s-1,t} - \begin{bmatrix} 0 & 0 \\ 0 & (B_{q,2}^{s,t})^{\dagger} B_{q,2}^{s,t} \end{bmatrix} \right) (A_{q}^{s,t})^{T}.$$

□

In this setting, we can prove a similar Lemma to Lemma 3.3.2. It is important in the final theorem of this section.

Lemma 3.3.5. *In a simplicial filtration $\{K_t\}_{0 \le t \le T}$, we have that*

$$\begin{bmatrix} 0 & 0 \\ 0 & (B_{q,2}^{s,t})^{\dagger} B_{q,2}^{s,t} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & (B_{q,2}^{s,t})^{\dagger} B_{q,2}^{s,t} \end{bmatrix} (B_{q,2}^{s-1,t})^{\dagger} B_{q,2}^{s-1,t} = (B_{q,2}^{s-1,t})^{\dagger} B_{q,2}^{s-1,t} \begin{bmatrix} 0 & 0 \\ 0 & (B_{q,2}^{s,t})^{\dagger} B_{q,2}^{s,t} \end{bmatrix}. \tag{3.19}$$

Proof. In the same manner as before in the horizontal operator, we rewrite the boundary matrix,

$$B_{q,2}^{s-1,t} = \begin{bmatrix} [\partial_{q+1}^{t}](I_{q}^{s-1,s}, I_{q+1}^{s-1,s}) & [\partial_{q+1}^{t}](I_{q}^{s-1,s}, I_{q+1}^{s,t}) \\ 0 & [\partial_{q+1}^{t}](I_{q}^{s,t}, I_{q+1}^{s,t}) \end{bmatrix} = \begin{bmatrix} [\partial_{q+1}^{t}](I_{q}^{s-1,s}, I_{q+1}^{s-1,s}) & [\partial_{q+1}^{t}](I_{q}^{s-1,s}, I_{q+1}^{s,t}) \\ 0 & B_{q,2}^{s,t} \end{bmatrix}.$$

We now have that,

$$v \in \ker B_{q,2}^{s-1,t} \Rightarrow v(I_{q+1}^{s,t}) \in \ker B_{q,2}^{s,t} \Rightarrow v(I_{q+1}^{s,t}) \in \ker (B_{q,2}^{s,t})^{T} B_{q,2}^{s,t} \iff v \in \ker \begin{bmatrix} 0 & 0 \\ 0 & (B_{q,2}^{s,t})^{T} B_{q,2}^{s,t} \end{bmatrix}. \tag{3.20}$$

And therefore $\ker B_{q,2}^{s-1,t} \subseteq \ker \begin{bmatrix} 0 & 0 \\ 0 & (B_{q,2}^{s,t})^{T} B_{q,2}^{s,t} \end{bmatrix}$, which allows us to use Equations (2.2) and (2.8) again, to get,

$$\begin{aligned}
\begin{bmatrix} 0 & 0 \\ 0 & (B_{q,2}^{s,t})^{\dagger} B_{q,2}^{s,t} \end{bmatrix} &= \begin{bmatrix} 0 & 0 \\ 0 & (B_{q,2}^{s,t})^{T} B_{q,2}^{s,t} \end{bmatrix}^{\dagger} \begin{bmatrix} 0 & 0 \\ 0 & (B_{q,2}^{s,t})^{T} B_{q,2}^{s,t} \end{bmatrix} \\
&= \begin{bmatrix} 0 & 0 \\ 0 & (B_{q,2}^{s,t})^{T} B_{q,2}^{s,t} \end{bmatrix}^{\dagger} \begin{bmatrix} 0 & 0 \\ 0 & (B_{q,2}^{s,t})^{T} B_{q,2}^{s,t} \end{bmatrix} (B_{q,2}^{s-1,t})^{\dagger} B_{q,2}^{s-1,t} \\
&= \begin{bmatrix} 0 & 0 \\ 0 & (B_{q,2}^{s,t})^{\dagger} B_{q,2}^{s,t} \end{bmatrix} (B_{q,2}^{s-1,t})^{\dagger} B_{q,2}^{s-1,t}.
\end{aligned}$$

Again, the final equality in equation (3.19), can be acquired by taking the transpose of this and realizing that the matrices are symmetric. □

Finally, we get a Theorem similar to Theorem 3.3.3.

Theorem 3.3.6. *Let* $\Psi_v := \ker \begin{bmatrix} 0_{(n_{q+1}^s - n_{q+1}^{s-1}) \times (n_{q+1}^s - n_{q+1}^{s-1})} & 0 \\ 0 & B_{q,2}^{s,t} \end{bmatrix} \cap \left(\ker B_{q,2}^{s-1,t}\right)^{\perp}$ *and* $[\Psi_v]$ *represent an orthonormal basis for* Ψ_v*. We get,*

$$[\mathcal{V}_q^{s,t}] = \sum_{v \in [\Psi_v]} A_q^{s,t} v \left(A_q^{s,t} v\right)^T. \tag{3.21}$$

Proof. The same approach as Theorem 3.3.3 is used. Using Lemmas 3.3.4 and 3.3.5, we can write,

$$[\Delta_{q,+}^{s,t}] - [\widetilde{\Delta_{q,+}^{s-1,t}}] = A_q^{s,t} (B_{q,2}^{s-1,t})^{\dagger} B_{q,2}^{s-1,t} \left(I - \begin{bmatrix} 0 & 0 \\ 0 & (B_{q,2}^{s,t})^{\dagger} B_{q,2}^{s,t} \end{bmatrix} \right) (B_{q,2}^{s-1,t})^{\dagger} B_{q,2}^{s-1,t} (A_q^{s,t})^T. \tag{3.22}$$

Note that,

$$\begin{bmatrix} 0_{(n_{q+1}^s - n_{q+1}^{s-1}) \times (n_{q+1}^s - n_{q+1}^{s-1})} & 0 \\ 0 & (B_{q,2}^{s,t})^{\dagger} B_{q,2}^{s,t} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & B_{q,2}^{s,t} \end{bmatrix}^{\dagger} \begin{bmatrix} 0 & 0 \\ 0 & B_{q,2}^{s,t} \end{bmatrix}.$$

Therefore $I - \begin{bmatrix} 0 & 0 \\ 0 & B_{q,2}^{s,t} \end{bmatrix}$ projects onto $\ker \begin{bmatrix} 0 & 0 \\ 0 & B_{q,2}^{s,t} \end{bmatrix}$. Furthermore, $(B_{q,2}^{s-1,t})^{\dagger} B_{q,2}^{s-1,t}$ projects onto $\left(\ker B_{q,2}^{s-1,t}\right)^{\perp}$, see Section 2.1. Now note that the projection matrices commute because of Lemma 3.3.5. Therefore, their product is again a projection matrix that projects onto the intersection of the two spaces [36], $\Psi_v := \ker \begin{bmatrix} 0 & 0 \\ 0 & B_{q,2}^{s,t} \end{bmatrix} \cap \left(\ker B_{q,2}^{s-1,t}\right)^{\perp}$.

If we consider $[\Psi_v]$ an orthonormal basis for Ψ_v, we can represent this projection by VV^T, where the columns of V are the vectors in $[\Psi_v]$. This allows us to write,

$$\begin{aligned} [\Delta_q^{s,t}] - [\widetilde{\Delta_q^{s-1,t}}] &= A_q^{s,t} \sum_{v \in [\Psi_v]} v v^T (A_q^{s,t})^T \\ &= \sum_{v \in [\Psi_v]} A_q^{s,t} v \left(A_q^{s,t} v\right)^T. \end{aligned}$$

□

Note that this Theorem again shows that $\mathcal{V}_q^{s,t}$ is a positive semi definite operator. We can plot the eigenvalues of the operator in the same way as before, see Figure 3.6. Here we can see that for every rectangle, instead of removing the effect of the rectangle below it, now the effect of the rectangle to the left of it has been removed. All the previously discussed points where the persistent up-Laplacian changes are also visible here. The goal now is to do both these operations and remove both the vertical as well as the horizontal effect to get an operator that has a non-zero eigenvalue on only the points where the persistent Laplacian changes. This is done in the next section.

3.3.3. Finite barcodes

In this section, we focus on the multiplicity equation for persistent Laplacians, see Equation (3.1). The same structure as the previous two sections is used, however some results can not be obtained in this setting. To obtain results in a similar fashion, the adjusted multiplicity operator is proposed together with a discussion on why it may be useful.

Note that in Equation (3.1), the down Laplacians all disappear. By definition of the persistent Laplacian, we have,

$$\begin{aligned} \left(\Delta_q^{s,t} - \Delta_q^{s,t-1}\right) - \left(\Delta_q^{s-1,t} - \Delta_q^{s-1,t-1}\right) &= \Delta_{q,-}^{s} + \Delta_{q,+}^{s,t} - \left(\Delta_{q,-}^{s} + \Delta_{q,+}^{s,t-1}\right) - \left(\Delta_{q,-}^{s-1} + \Delta_{q,+}^{s-1,t} - \left(\Delta_{q,-}^{s-1} + \Delta_{q,+}^{s-1,t-1}\right)\right) \\ &= \left(\Delta_{q,+}^{s,t} - \Delta_{q,+}^{s,t-1}\right) - \left(\Delta_{q,+}^{s-1,t} - \Delta_{q,+}^{s-1,t-1}\right). \end{aligned}$$

Therefore, down-Laplacians are not discussed in this section.

Because the multiplicity equation contains Laplacians with different start times, the extension of Definition 3.3.1 needs to be used. We obtain,

$$M_q^{s,t} := \left(\Delta_{q,+}^{s,t} - \Delta_{q,+}^{s,t-1}\right) - \left(\widetilde{\Delta_{q,+}^{s-1,t}} - \widetilde{\Delta_{q,+}^{s-1,t-1}}\right). \tag{3.23}$$

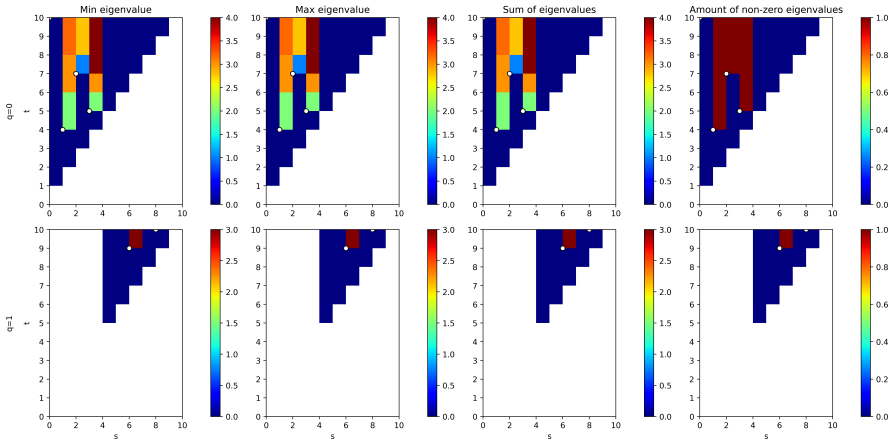


Figure 3.6: Plot of different aggregation functions applied to the eigenvalues of $\mathcal{V}_q^{s,t}$ for the filtration visualized in Figure 2.6. White dots represent the intervals of the standard persistent barcode.

Note that the extension is linear, therefore the subtraction between the two extended up persistent Laplacians can be done before extending. Formally, $\widetilde{\Delta_{q,+}^{s-1,t}} - \widetilde{\Delta_{q,+}^{s-1,t-1}} = \widetilde{\Delta_{q,+}^{s-1,t} - \Delta_{q,+}^{s-1,t-1}}$. This allows for the simplification of the matrix representation of the multiplicity operator in the next Theorem.

Theorem 3.3.7. *For a filtration of simplicial complexes $\{K_t\}_{0\le t\le T}$, let $0 < s < t < T$ be some start and end times and q a specified dimension. The matrix representation of the multiplicity operator from Equation* (3.23) *can be written as*

$$[M_q^{s,t}] = A_q^{s,t}\left(\begin{bmatrix} 0_{n_{q+1}^s - n_{q+1}^{s-1}} & 0 & 0 \\ 0 & (B_{q,2}^{s,t-1})^\dagger B_{q,2}^{s,t-1} & 0 \\ 0 & 0 & I_{n_{q+1}^t - n_{q+1}^{t-1}} \end{bmatrix} - \begin{bmatrix} 0_{n_{q+1}^s - n_{q+1}^{s-1}} & 0 \\ 0 & (B_{q,2}^{s,t})^\dagger B_{q,2}^{s,t} \end{bmatrix} - \begin{bmatrix} (B_{q,2}^{s-1,t-1})^\dagger B_{q,2}^{s-1,t-1} & 0 \\ 0 & I_{n_{q+1}^t - n_{q+1}^{t-1}} \end{bmatrix} + (B_{q,2}^{s-1,t})^\dagger B_{q,2}^{s-1,t}\right)(A_q^{s,t})^T. \quad (3.24)$$

Where $A_q^{s,t} = [\partial_{q+1}^t]([n_q^s], I_{q+1}^{s-1,t})$ and $0_n \in \mathbb{R}^{n\times n}$, the square zero matrix.

Proof. Note that $\ker \begin{bmatrix} (B_{q,2}^{s-1,t-1})^T B_{q,2}^{s-1,t-1} & 0 \\ 0 & I_{n_{q+1}^t - n_{q+1}^{t-1}} \end{bmatrix} \subseteq \ker B_{q,2}^{s-1,t} \subseteq \ker \begin{bmatrix} A_{q,21}^{s,t} & A_{q,22}^{s,t} \end{bmatrix}$, see the proofs of Lemmas 3.3.5 and 3.3.1. In addition to Equation (3.18), by Equation (2.8) we therefore have,

$$\begin{bmatrix} A_{q,21}^{s,t} & A_{q,22}^{s,t} \end{bmatrix} = \begin{bmatrix} A_{q,21}^{s,t} & A_{q,22}^{s,t} \end{bmatrix} \begin{bmatrix} (B_{q,2}^{s-1,t-1})^\dagger B_{q,2}^{s-1,t-1} & 0 \\ 0 & I_{n_{q+1}^t - n_{q+1}^{t-1}} \end{bmatrix}.$$

Hence, we obtain the following,

$$\begin{bmatrix} 0 & 0 \\ A_{q,21}^{s,t} & A_{q,22}^{s,t} \end{bmatrix} \left(\begin{bmatrix} (B_{q,2}^{s-1,t-1})^\dagger B_{q,2}^{s-1,t-1} & 0 \\ 0 & I_{n_{q+1}^t - n_{q+1}^{t-1}} \end{bmatrix} - (B_{q,2}^{s-1,t})^\dagger B_{q,2}^{s-1,t} \right) \begin{bmatrix} 0 & (A_{q,21}^{s,t})^T \\ 0 & (A_{q,22}^{s,t})^T \end{bmatrix} = 0.$$

Using equation (3.8) twice and using the notation of equation (3.16), we obtain

$$\begin{aligned}
[M_q^{s,t}] =& (B_{q,1}^{s,t}\left(\begin{bmatrix}(B_{q,2}^{s,t-1})^\dagger B_{q,2}^{s,t-1} & 0\\ 0 & I_{n_{q+1}^t - n_{q+1}^{t-1}}\end{bmatrix} - (B_{q,2}^{s,t})^\dagger B_{q,2}^{s,t}\right)(B_{q,1}^{s,t})^T \\
&- \begin{bmatrix} I_{n_q^{s-1}} \\ 0\end{bmatrix}\left(B_{q,1}^{s-1,t}\left(\begin{bmatrix}(B_{q,2}^{s-1,t-1})^\dagger B_{q,2}^{s-1,t-1} & 0\\ 0 & I_{n_{q+1}^t - n_{q+1}^{t-1}}\end{bmatrix} - (B_{q,2}^{s-1,t})^\dagger B_{q,2}^{s-1,t}\right)(B_{q,1}^{s-1,t})^T\right)\begin{bmatrix} I_{n_q^{s-1}} & 0\end{bmatrix} \\
=& \begin{bmatrix} A_{q,11}^{s,t} & A_{q,12}^{s,t} \\ A_{q,21}^{s,t} & A_{q,22}^{s,t}\end{bmatrix}\left(\begin{bmatrix}0 & 0 & 0\\ 0 & (B_{q,2}^{s,t-1})^\dagger B_{q,2}^{s,t-1} & 0\\ 0 & 0 & I\end{bmatrix} - \begin{bmatrix}0 & 0\\ 0 & (B_{q,2}^{s,t})^\dagger B_{q,2}^{s,t}\end{bmatrix}\right)\begin{bmatrix}(A_{q,11}^{s,t})^T & (A_{q,21}^{s,t})^T\\ (A_{q,12}^{s,t})^T & (A_{q,22}^{s,t})^T\end{bmatrix} \\
&- \begin{bmatrix} A_{q,11}^{s,t} & A_{q,12}^{s,t} \\ A_{q,21}^{s,t} & A_{q,22}^{s,t}\end{bmatrix}\left(\begin{bmatrix}(B_{q,2}^{s-1,t-1})^\dagger B_{q,2}^{s-1,t-1} & 0\\ 0 & I_{n_{q+1}^t - n_{q+1}^{t-1}}\end{bmatrix} - (B_{q,2}^{s-1,t})^\dagger B_{q,2}^{s-1,t}\right)\begin{bmatrix}(A_{q,11}^{s,t})^T & (A_{q,21}^{s,t})^T\\ (A_{q,12}^{s,t})^T & (A_{q,22}^{s,t})^T\end{bmatrix}.
\end{aligned}$$

□

The previous theorem shows that applying the multiplicity equation to the persistent Laplacians translates to applying the same equation for the $B_{q,2}$ matrices. To simplify notation, these projection matrices are written in the following form,

$$P_q^{s,t} := \begin{bmatrix} 0_{n_{q+1}^s - n_{q+1}^{s-1}} & 0 \\ 0 & (B_{q,2}^{s,t})^\dagger B_{q,2}^{s,t}\end{bmatrix}, \qquad P_q^{s,t-1} := \begin{bmatrix} 0_{n_{q+1}^s - n_{q+1}^{s-1}} & 0 & 0 \\ 0 & (B_{q,2}^{s,t-1})^\dagger B_{q,2}^{s,t-1} & 0 \\ 0 & 0 & I_{n_{q+1}^t - n_{q+1}^{t-1}}\end{bmatrix}, \tag{3.25}$$

$$P_q^{s-1,t} := (B_{q,2}^{s-1,t})^\dagger B_{q,2}^{s-1,t}, \qquad P_q^{s-1,t-1} := \begin{bmatrix}(B_{q,2}^{s-1,t-1})^\dagger B_{q,2}^{s-1,t-1} & 0 \\ 0 & I_{n_{q+1}^t - n_{q+1}^{t-1}}\end{bmatrix}.$$

In this notation, we have $[M_q^{s,t}] = A_q^{s,t}\left(P_q^{s,t-1} - P_q^{s,t} - P_q^{s-1,t-1} + P_q^{s-1,t}\right)(A_q^{s,t})^T$. Important here is that the superscript is somewhat ill defined as, for example, $P_0^{1,2}$ and $P_0^{1,3-1}$ could have different shapes. Furthermore, note that the projection matrix $P_q^{s,t}$ projects onto the complement of the kernel of $B_{q,2}^{s,t}$ extended to be on the same space as $B_{q,2}^{s-1,t}$, see Section 2.1. Finally, it is well known that the kernel of a projection matrix is the orthogonal complement of the image. Therefore, without concern for the extension of the subspace, we get, $\ker P_q^{s,t} = \left((\ker B_{q,2}^{s,t})^\perp\right)^\perp = \ker B_{q,2}^{s,t}$.

For these projection matrices, we can obtain a few calculation rules using Lemmas 3.3.5 and 3.3.2.

$$P_q^{s,t} = P_q^{s,t}P_q^{s,t-1} = P_q^{s,t-1}P_q^{s,t}, \tag{3.26}$$

$$P_q^{s,t} = P_q^{s,t}P_q^{s-1,t} = P_q^{s-1,t}P_q^{s,t}, \tag{3.27}$$

$$P_q^{s,t} = P_q^{s,t}P_q^{s-1,t-1} = P_q^{s-1,t-1}P_q^{s,t}. \tag{3.28}$$

Note here that there is no rule for the product $P_q^{s,t-1}P_q^{s-1,t}$, which becomes important later on in the thesis.

Besides these equations, we can find a connection between the kernels of three of the projection matrices, which provide some insight into the function they have. This is formulated in the following lemma.

Lemma 3.3.8. *We have,*

$$\ker P_q^{s-1,t-1} = \ker P_q^{s,t-1} \cap \ker P_q^{s-1,t}. \tag{3.29}$$

Proof. Let $v \in \ker P_q^{s-1,t-1}$, from the proofs of Lemmas 3.3.2 and 3.3.5, we get that $v \in \ker P_q^{s,t-1}$ and $v \in \ker P_q^{s-1,t}$, therefore $\ker P_q^{s-1,t-1} \subseteq \ker P_q^{s,t-1} \cap \ker P_q^{s-1,t}$.

Now let $v \in \ker P_q^{s,t-1} \cap \ker P_q^{s-1,t}$, we again exploit the structure of the boundary matrices. We can write,

$$B_{q,2}^{s-1,t} = \begin{bmatrix} [\partial_q^t](I_q^{s-1,s}, I_{q+1}^{s-1,s}) & [\partial_q^t](I_q^{s-1,s}, I_{q+1}^{s,t}) & [\partial_q^t](I_q^{s-1,s}, I_{q+1}^{t-1,t}) \\ 0 & B_{q,2}^{s,t-1} & [\partial_q^t](I_q^{s,t-1}, I_{q+1}^{t-1,t}) \\ 0 & 0 & [\partial_q^t](I_q^{t-1,t}, I_{q+1}^{t-1,t})\end{bmatrix},$$

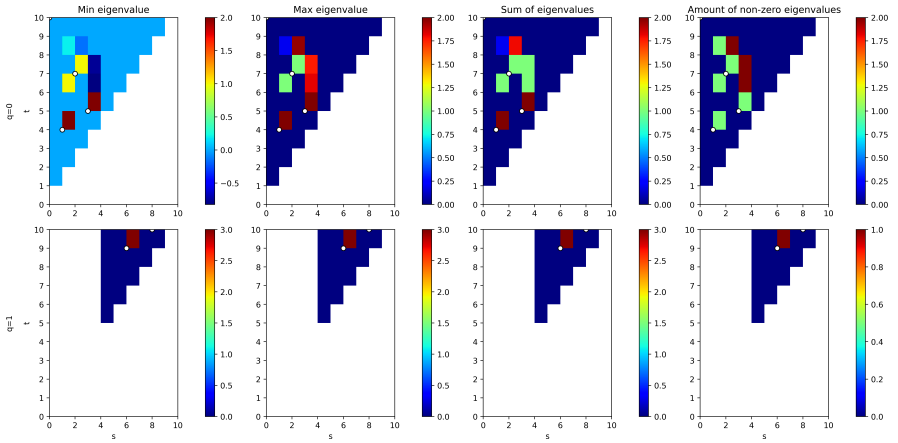


Figure 3.7: Plot of different aggregation functions applied to the eigenvalues of $M_q^{s,t}$ for the filtration visualized in Figure 2.6. White dots represent the intervals of the standard persistent barcode.

where the highlighted part of the matrix is $B_{q,2}^{s-1,t-1}$. Now note that $v \in \ker P_q^{s-1,t} = \ker B_{q,2}^{s-1,t}$. Furthermore, $v(I_{q+1}^{t-1,t}) = 0$ as $v \in \ker P_q^{s,t-1} = \ker \begin{bmatrix} 0 & 0 & 0 \\ 0 & (B_{q,2}^{s,t-1})^\dagger B_{q,2}^{s,t-1} & 0 \\ 0 & 0 & I_{n_{q+1}^t - n_{q+1}^{t-1}} \end{bmatrix}$. Therefore, we obtain $\begin{bmatrix} B_{q,2}^{s-1,t-1} & 0 \\ 0 & I \end{bmatrix} v = 0$. Hence, $v \in \ker P_q^{s-1,t-1}$. □

Note that a direct consequence of this lemma is that $(\ker P_q^{s-1,t-1})^\perp = (\ker P_q^{s,t-1})^\perp + (\ker P_q^{s-1,t})^\perp$. Furthermore, the lemma tells us that the features that are part of $(s-1,t-1)$ are exactly the features that are both in $(s,t-1)$ and $(s-1,t)$. If $P_q^{s,t-1}$ and $P_q^{s-1,t}$ would commute, their product would equal $P_q^{s-1,t-1}$, however this is not always true.

Now the same plots as before in Figures 3.4 and 3.6 can be made, see Figure 3.7. Here it can be seen that the multiplicity operator for dimension 0 is non-zero on exactly the previously discussed points. However, for the points $(2,8), (3,6)$ and $(3,7)$, the minimum eigenvalue is negative. Furthermore, on these points, there are two eigenvalues present. This is because, in these points, there is both a vertical as well as a horizontal effect. It also shows that the matrix is not necessarily positive semi definite as it can have a negative eigenvalue.

Because the matrix is no longer positive semi definite, finding a representation of $[M_q^{s,t}]$ in terms of outer products of a certain set of vectors, like in Theorems 3.3.6 and 3.3.3 is not possible. However, we can still try to formulate a similar expression with the previously defined calculation rules,

$$\begin{aligned}[M_q^{s,t}] &= A_q^{s,t}\left(\left(I - P_q^{s,t}\right)P_q^{s,t-1} - \left(I - P_q^{s-1,t}\right)P_q^{s-1,t-1}\right)(A_q^{s,t})^T \\ &= \sum_{w \in [\ker P_q^{s,t} \cap (\ker P_q^{s,t-1})^\perp]} A_q^{s,t} w (A_q^{s,t} w)^T - \sum_{v \in [\ker P_q^{s-1,t} \cap (\ker P_q^{s-1,t-1})^\perp]} A_q^{s,t} v (A_q^{s,t} v)^T. \end{aligned} \tag{3.30}$$

Here we can see that instead of being a sum of outer products of a single set of vectors, it is now the difference between sums of two different sets. Namely $V_{-1} := \{A_q^{s,t} v : v \in \ker P_q^{s-1,t} \cap (\ker P_q^{s-1,t-1})^\perp\}$ and $V := \{A_q^{s,t} w : w \in \ker P_q^{s,t} \cap (\ker P_q^{s,t-1})^\perp\}$. It can be determined that when the second set is empty, the first set is as well, the proof of this is left for future research. Therefore, when V_{-1} is empty $[M_q^{s,t}]$ can be written as one sum and the matrix is positive semi definite. However, this does not need to be the case. In Figure 3.7 on the points $(2,8), (3,6)$ and $(3,7)$, the two sets are both non-empty causing the eigenvalues and eigenvectors of the two sets to interact, which makes them difficult to interpret. This also causes additional non-zero eigenvalues to appear. The filtration of Figure 2.6 only differs by one simplex in each step, however at the aforementioned points, there are two non-zero eigenvalues. Therefore, the eigenvalues no longer correspond to specific features.

If we instead look at the trace, using the calculation rules of the projection matrices, the following expression can be found which has been formulated into a Corollary.

Corollary 3.3.9. *The trace of the matrix representation of the multiplicity operator $M_q^{s,t}$ can be written as,*

$$Tr([M_q^{s,t}]) = \sum_{w \in [\ker P_q^{s,t} \cap \left(\ker P_q^{s-1,t-1}\right)^\perp]} ||A_q^{s,t} w||_2^2 - ||A_q^{s,t}(I - P_q^{s,t-1})w||_2^2 - ||A_q^{s,t}(I - P_q^{s-1,t})w||_2^2. \quad (3.31)$$

Proof. Let $P = P_q^{s-1,t-1} - P_q^{s,t}$, using Theorem 3.3.7 and the calculation rules of the projection matrices (3.26), we get

$$\begin{aligned}[M_q^{s,t}] &= A_q^{s,t}\left(P_q^{s,t-1} - P_q^{s,t} - P_q^{s-1,t-1} + P_q^{s-1,t}\right)(A_q^{s,t})^T \\ &= A_q^{s,t}\left(P - (I - P_q^{s,t-1})P(I - P_q^{s,t-1}) - (I - P_q^{s-1,t})P(I - P_q^{s-1,t})\right)(A_q^{s,t})^T.\end{aligned}$$

Note that P projects onto $\ker P_q^{s,t} \cap \left(\ker P_q^{s-1,t-1}\right)^\perp$, therefore in the same way as before, it can be represented as a sum.

$$[M_q^{s,t}] = \sum_{w \in [\ker P_q^{s,t} \cap \left(\ker P_q^{s-1,t-1}\right)^\perp]} A_q^{s,t}\left(ww^T - (I - P_q^{s,t-1})ww^T(I - P_q^{s,t-1}) - (I - P_q^{s-1,t})ww^T(I - P_q^{s-1,t})\right)(A_q^{s,t})^T.$$

This yields the following equation for the trace

$$Tr([M_q^{s,t}]) = \sum_{w \in [\ker P_q^{s,t} \cap \left(\ker P_q^{s-1,t-1}\right)^\perp]} ||A_q^{s,t} w||_2^2 - ||A_q^{s,t}(I - P_q^{s,t-1})w||_2^2 - ||A_q^{s,t}(I - P_q^{s-1,t})w||_2^2.$$

□

Therefore, the trace can be interpreted as follows. Each w represents an effect that was present at (s,t) but not at $(s-1,t-1)$. The trace sums the *impacts* of these effects $||A_q^{s,t} w||_2^2$, but, for each w, it removes the effect it had on step $(s,t-1)$ and $(s-1,t)$ by subtracting $||A_q^{s,t}(I - P_q^{s,t-1})w||_2^2$ and $||A_q^{s,t}(I - P_q^{s-1,t})w||_2^2$ respectively.

Finally, note that in dimension 1, the multiplicity operator is 0 on all points except for $(6,9)$. While the persistent Laplacian did change on more points, see Figure 3.2, the changes in this dimension mainly came from the down Laplacian as no 2-simplices exist before $t = 9$. Note that these changes start at the diagonal, so from the combinatorial Laplacian. At the diagonal only a vertical operator could be made. Like was said in the previous section, for the vertical operator, only the up-persistent Laplacian is used, which means these changes are not captured by the multiplicity operator. As the only effect that is not captured by the multiplicity operator originates from the diagonal, or when $s = t$, we note that this is equivalent to saying that the information of the combinatorial Laplacian is not captured by the multiplicity operator. Adding this information to a model based on the multiplicity operator is attempted in the application of the MNIST dataset in Section 4.1.2.

Now a decision can be made on how to continue. Either the previous multiplicity operator is used and only the trace is interpreted, or a new operator can be defined that solves some of the issues. One could look for an operator that contains no more non-zero eigenvalues than the number of simplices appearing in a certain step. For the remainder of this section, a new operator is discussed that achieves this, however the trace is altered in some points. An operator that has the same trace and less non-zero eigenvalues has not been found.

Interpretable eigenvalues

Our goal is now to create a matrix that has non-zero eigenvalues on the same points as $M_q^{s,t}$, but with "interpretable" and positive eigenvalues. Formally, we seek a matrix $\tilde{M}_q^{s,t}$, that satisfies the following criteria:

1. $M_q^{s,t} = 0 \implies \tilde{M}_q^{s,t} = 0$.
2. The eigenvalues of $\tilde{M}_q^{s,t}$ are real and non-negative.

3. The number of positive eigenvalues should correspond with the number of new features, which cannot exceed the number of simplexes being added. Formally, let $n := n_{q+1}^t - n_{q+1}^{t-1}$, so exactly n $(q+1)$-simplices are added at time t. We require $\dim \mathrm{Im}\,(\tilde{M}_q^{s,t}) \leq n$.

Theorem 3.3.10. *The matrix $\tilde{M}_q^{s,t}$ defined below, satisfies criteria 1 and 2. Furthermore, it satisfies criterion 3 if it is diagonalizable.*

$$\tilde{M}_q^{s,t} := A(P^{s,t-1}(I - P^{s,t})P^{s-1,t})A^T = A(P^{s,t-1}P^{s-1,t} - P^{s,t})A^T. \tag{3.32}$$

Proof. **Criterion 1:** We can rewrite the representation found in Theorem 3.3.7 and assume it is zero to get,

$$\begin{aligned}
[M_q^{s,t}] = A_q^{s,t}(I - P_q^{s,t})\left(P_q^{s,t-1} + P_q^{s-1,t} - P_q^{s-1,t-1}\right)(A_q^{s,t})^T = 0 \\
\implies A_q^{s,t}(I - P_q^{s,t})\left(P_q^{s,t-1} + P_q^{s-1,t} - P_q^{s-1,t-1}\right) = 0. \\
\implies A_q^{s,t}(I - P_q^{s,t})P_q^{s,t-1} = A_q^{s,t}(I - P_q^{s,t})\left(P_q^{s-1,t} - P_q^{s-1,t-1}\right)
\end{aligned}$$

Furthermore, for $\tilde{M}_q^{s,t}$ a similar representation can be found,

$$\begin{aligned}
\tilde{M}_q^{s,t} &= A_q^{s,t}(I - P_q^{s,t})P^{s,t-1}P^{s-1,t}(A_q^{s,t})^T \\
&= A_q^{s,t}(I - P_q^{s,t})\left(P_q^{s-1,t} - P_q^{s-1,t-1}\right)P_q^{s-1,t}(A_q^{s,t})^T = 0.
\end{aligned}$$

Criterion 2: We can write

$$\tilde{M}_q^{s,t} = A\tilde{P}^{s,t-1}\tilde{P}^{s-1,t}A^T,$$

with $\tilde{P}^{s,t-1} := P^{s,t-1}(I - P^{s,t})$, the orthogonal projection matrix corresponding to the projection onto $\ker P^{s,t} \cap (\ker P^{s,t-1})^\perp$ and $\tilde{P}^{s-1,t} := P^{s-1,t}(I - P^{s,t})$ corresponding to the projection onto $\ker P^{s,t} \cap (\ker P^{s-1,t})^\perp$.

First note that $\tilde{P}^{s,t-1}$ and $\tilde{P}^{s-1,t}$ are orthogonal projection matrices and therefore positive semi-definite. We define two cases:

Case 1: $\tilde{P}^{s-1,t}$ is non-singular. This means that we can write it as $\tilde{P}^{s-1,t} = P^{1/2}P^{1/2}$ for some invertible and symmetric $P^{1/2}$. The matrix $X := \tilde{P}^{s,t-1}\tilde{P}^{s-1,t}$ is then similar to $X' := P^{1/2}\tilde{P}^{s,t-1}P^{1/2}$, as $X' = P^{1/2}XP^{-1/2}$. X therefore shares the same eigenvalues and eigenvectors as X'. Furthermore, X' is congruent to $\tilde{P}^{s,t-1}$, which means that for some x, we have $x^TX'x = x^TP^{1/2}\tilde{P}^{s,t-1}P^{1/2}x = (P^{1/2}x)^T\tilde{P}^{s,t-1}(P^{1/2}x) \geq 0$. Therefore, X' is also positive semi-definite and has non-negative eigenvalues, which in turn shows that X has non-negative eigenvalues. $\tilde{M}_q^{s,t}$ is congruent to X as $\tilde{M}_q^{s,t} = AXA^T$ and therefore $\tilde{M}_q^{s,t}$ is also positive semi-definite and has real positive eigenvalues.

Case 2: $\tilde{P}^{s-1,t}$ is singular. Now instead consider $P_\varepsilon := \tilde{P}^{s-1,t} + \varepsilon I$ and $M_\varepsilon := A\tilde{P}^{s,t-1}P_\varepsilon A^T$. Because P_ε is non-singular for any $\varepsilon > 0$, we can use the same argument as before and conclude that M_ε has real non-negative eigenvalues. Now using the property that eigenvalues of a matrix are continuous [29] and $\lim_{\varepsilon\to 0^+} M_\varepsilon = \tilde{M}_q^{s,t}$, we obtain the required result that the eigenvalues of $\tilde{M}_q^{s,t}$ are real and non-negative.

Criterion 3: We start by proving that every eigenvector v of $\tilde{M}_q^{s,t}$ is a linear combination of the vectors $V := \{Aw : w \in [\ker P^{s,t} \cap (\ker P^{s,t-1})^\perp]\}$.

Let v be an eigenvector of $\tilde{M}_q^{s,t}$, with corresponding eigenvalue λ. We get,

$$\begin{aligned}
\tilde{M}_q^{s,t}v &= \sum_{w\in[\ker P^{s,t}\cap(\ker P^{s,t-1})^\perp]} Aww^T\tilde{P}^{s-1,t}A^Tv \\
&= \sum_{w\in[\ker P^{s,t}\cap(\ker P^{s,t-1})^\perp]} Aw\langle A\tilde{P}^{s-1,t}w, v\rangle = \lambda v \\
\iff v &= \sum_{w\in[\ker P^{s,t}\cap(\ker P^{s,t-1})^\perp]} \frac{\langle A\tilde{P}^{s-1,t}w, v\rangle}{\lambda} Aw.
\end{aligned}$$

Because we have assumed that $\tilde{M}_q^{s,t}$ is diagonalizable, the eigenvectors are linearly independent. Which means that the number of eigenvectors with non-negative eigenvalues is upper bounded by $\dim Span(V)$.

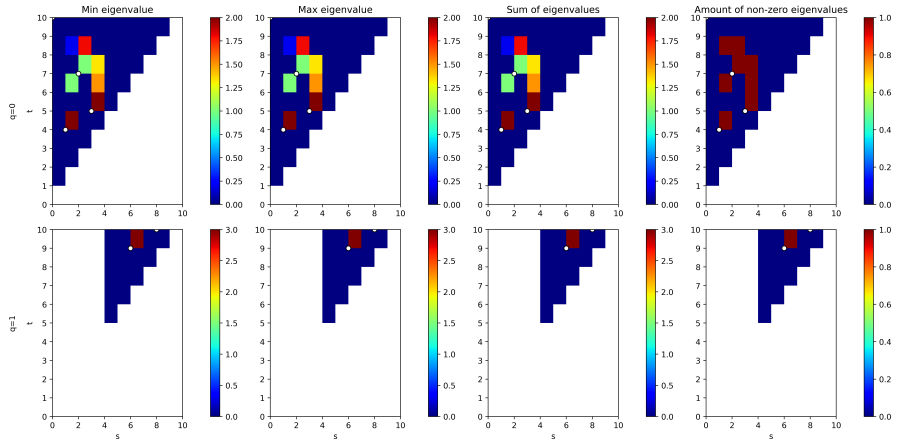


Figure 3.8: Plot of different aggregation functions applied to the eigenvalues of $\tilde{M}_q^{s,t}$ for the filtration visualized in Figure 2.6. White dots represent the intervals of the standard persistent barcode.

It now remains to show that the dimension of this span is less than or equal to $(n_{q+1}^t - n_{q+1}^{t-1})$. Note that the number of vectors in V is equal to $\dim\left(\ker P^{s,t} \cap (\ker P^{s,t-1})^\perp\right) = \dim\ker P^{s,t} - \dim\ker P^{s,t-1}$ because of Lemma 3.3.2. We have that,

$$\begin{aligned}\dim\left(\ker P^{s,t} \cap (\ker P^{s,t-1})^\perp\right) &= \dim\ker P^{s,t} - \dim\ker P^{s,t-1} \\ &= \dim\ker B_{q,2}^{s,t} - \dim\ker B_{q,2}^{s,t-1} \\ &= \dim C_{q+1}^{s,t} - \dim C_{q+1}^{s,t-1}.\end{aligned}$$

Here, the last equality follows from Lemma 3.2.4. Because every $c \in C_{q+1}^t$ can at most create one $\tilde{c} \in C_{q+1}^{s,t}$, we have that

$$\begin{aligned}\dim\left(\ker P^{s,t} \cap (\ker P^{s,t-1})^\perp\right) &= \dim C_{q+1}^{s,t} - \dim C_{q+1}^{s,t-1} \\ &\leq n_{q+1}^t - n_{q+1}^{t-1}.\end{aligned}$$

The dimension of $Span(V)$ is less than or equal to the number of vectors in V, which means that we have found the required bound.

□

Note that, while it has not been proven that $\tilde{M}_q^{s,t}$ is diagonalizable, in practice it has always been the case.

Using this modified multiplicity matrix on the example filtration from Figure 2.6, we obtain Figure 3.8. Here we can see that now only one non-zero eigenvalue is present for each of the points (s,t). Furthermore, the trace of the matrix is the same as the trace of $[M_q^{s,t}]$ for all points where that matrix had only one non-zero eigenvalue. The points $(2,8), (3,6)$ and $(3,7)$, which had both a vertical and a horizontal effect, do have a different trace, however.

To understand why this is the case, we write $\tilde{M}_q^{s,t}$ in terms of the matrix representation of $M_q^{s,t}$,

$$\begin{aligned}\tilde{M}_q^{s,t} &= A_q^{s,t}\left(P_q^{s,t-1}P_q^{s-1,t} - P_q^{s,t}\right)(A_q^{s,t})^T \\ &= A_q^{s,t}\left(P_q^{s-1,t} + P_q^{s,t-1} - P_q^{s-1,t-1} - P_q^{s,t} + (I - P_q^{s,t-1})(P_q^{s-1,t-1} - P_q^{s,t})(I - P_q^{s-1,t})\right)(A_q^{s,t})^T \\ &= [M_q^{s,t}] + A_q^{s,t}\left((I - P_q^{s,t-1})(P_q^{s-1,t-1} - P_q^{s,t})(I - P_q^{s-1,t})\right)(A_q^{s,t})^T.\end{aligned}$$

This shows that the adjusted multiplicity operator is equal to the standard multiplicity operator plus some matrix. This added matrix contains the product $P_q^{s,t-1}P_q^{s-1,t}$, which is not necessarily a projection matrix. Therefore, it is hard to interpret its function.

Nevertheless, we can still look at the trace to see if this has any interpretability. Using Equation

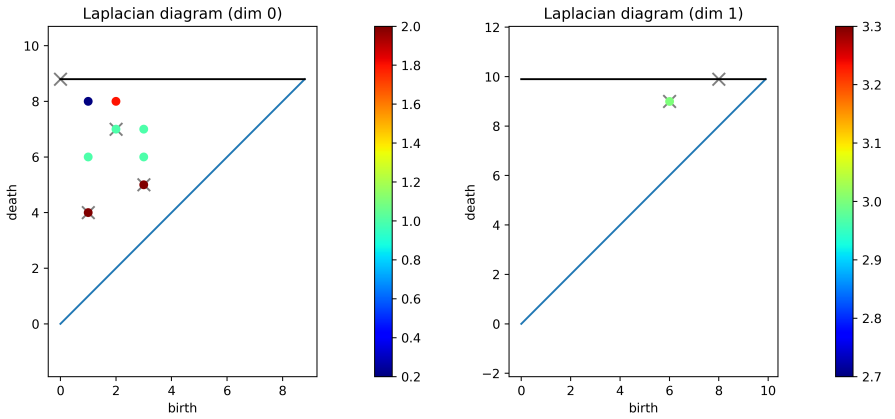


Figure 3.9: Colored persistence diagram containing the information of the trace of the multiplicity operator $[M_q^{s,t}]$ applied to the filtration of Figure 2.6. Crosses correspond to points in the standard persistence diagram of Figure 2.11. Previous representation of this trace can be seen in Figure 3.7.

(3.31) and the singular value decomposition of $P_q^{s-1,t-1} - P_q^{s,t}$, we have,

$$\begin{aligned} Tr(\tilde{M}_q^{s,t}) &= Tr([M_q^{s,t}]) + A_q^{s,t}\left((I - P_q^{s,t-1})(P_q^{s-1,t-1} - P_q^{s,t})(I - P_q^{s-1,t})\right)(A_q^{s,t})^T) \\ &= Tr([M_q^{s,t}]) + \sum_{w \in [\ker P_q^{s,t} \cap \left(\ker P_q^{s-1,t-1}\right)^{\perp}]} \langle A_q^{s,t}(I - P_q^{s,t-1})w, A_q^{s,t}(I - P_q^{s-1,t})w\rangle. \end{aligned} \tag{3.33}$$

Note that the added part over the trace of the standard multiplicity operator, is again a sum over the features from (s,t), which did not appear in $(s-1,t-1)$. It sums for each of the features the inner product between the horizontal and vertical effect of that feature. Therefore, if one of these effects is 0, it does nothing. Furthermore, it can be conluded that the trace of $\tilde{M}_q^{s,t}$ is the same as the trace of $[M_q^{s,t}]$ if features in (s,t) either come from a vertical effect or a horizontal effect and not both.

3.4. Using the multiplicity operators

In order to apply the multiplicity operator in practice, the spectra corresponding to two different filtrations need to be compared. Being able to compare two filtrations then yields a type of distance between two shapes. For the handwritten digit dataset, this distance could be used to check whether images of a certain digit have a smaller distance to each other than to other digits. If this is true, this distance could then be used to classify the numbers. In this section, a proposed distance is therefore given.

First a new visualization of the multiplicity operator is discussed. The operator is only non-zero at some points (b,d), we can therefore represent it as a colored persistence diagram. Colors are needed because the points still have a certain weight. Note that, like discussed, using the standard multiplicity operator $[M_q^{b,d}]$ only an interpretation for the trace was found and should therefore be used. This means that every combination of start and end times (b,d) only corresponds to one point. However, if one uses the modified multiplicity matrix $\tilde{M}_q^{s,t}$, specific eigenvalues can have an interpretable meaning. Nevertheless, using these separate eigenvalues in some way has not been done yet and is left for future research.

Plotting the trace of $[M_q^{s,t}]$ in a colored persistence diagram is done in Figure 3.9. From now on, we refer to such a diagram as a persistent Laplacian diagram. Note that this figure contains the exact same information as the ”Sum of eigenvalues” part of Figure 3.7. However, now it can more easily be compared to the standard persistence diagram as it is also a point cloud. This also allows for the usage of methods based on persistence diagrams, where only a slight adjustment needs to be made regarding the weight.

3.4.1. Landscapes for the multiplicity operators

One of these methods is the persistence landscape, see Section 2.4.3. It provides an easy way of comparing two diagrams by representing the diagrams as landscapes. For each point in the persistence

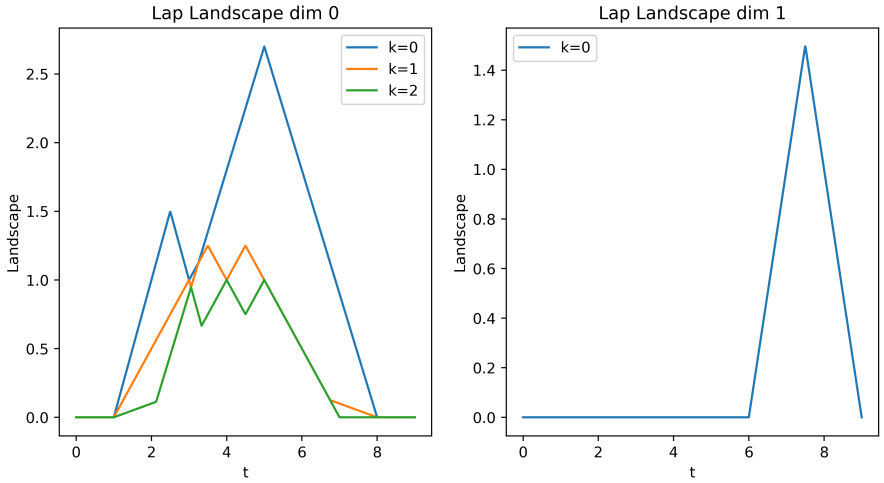


Figure 3.10: Persistent Laplacian landscape for $k \in \{0, 1, 2\}$, based on the filtration visualized in Figure 2.6 and corresponding persistent Laplacian diagram 3.9. See also Figure 2.12 for the corresponding persistence landscape.

diagram $p = (b, d)$ it formulates a simple function based on that point Λ_p. Adapting this method to work for the Laplacian diagram from the multiplicity operator, requires reformulating the functions Λ_p^l to work for points with a weight $p = (b, d, w)$.

It is proposed that this weight can be incorporated into the functions, using the following equation:

$$\Lambda_p^l(t) := \begin{cases} \frac{w}{q+2}(t-b) & \text{if } t \in [b, \frac{b+d}{2}]. \\ \frac{w}{q+2}(d-t) & \text{if } t \in (\frac{b+d}{2}, d]. \\ 0 & \text{otherwise.} \end{cases} \tag{3.34}$$

Where q is the dimension of the considered points. Note that this formulation is just $\Lambda_p^l(t) := \frac{w}{q+2}\Lambda_p(t)$. Therefore, higher weighted points correspond to larger function values. This is motivated by Equation 3.31 as here it was concluded that the trace is higher when more effect came from the step (s, t) and lower when some effect already appeared in $(s-1, t)$ or $(s, t-1)$. Dividing the weight by $q+2$ is done to be able to compare two diagrams of different dimensions. Because the same division is done for every point, it should not affect the comparison of two diagrams with the same dimension.

This allows for the same structure as before, where a persistent Laplacian landscape λ^l is a function defined over the set of all triangular functions $\{\Lambda_p^l\}_{p \in LD}$ for a persistent Laplacian diagram LD. Again, it also requires a positive integer k.

$$\lambda_{LD}^l(k, t) = k \max_{p \in LD} \Lambda_p^l(t). \tag{3.35}$$

In Figure 3.10 the persistent Laplacian landscape corresponding to the filtration of Figure 2.6 is plotted. Here only $k = 0, 1, 2$ are shown to more easily compare it to the persistence landscape of Figure 2.12. Note that these landscapes are quite similar as they both contain two large peaks in $k = 0$. However, the largest peak of the persistence landscape corresponds to the feature $(2, 7)$, while the largest peak of the persistent Laplacian landscape corresponds to $(2, 8)$ as can be seen by the value of t where this peak stops. At $t = 7$, the 0 dimensional feature corresponding to point 2 dies. However, it is still "far" away from 0 and 1, while at $t = 8$, the point connects directly to the main component.

3.5. Efficient algorithm

Like was said in Section 2.5.4, while computing one persistent Laplacian is not very time consuming, computing them for every combination of start times and end times can be. The formulation of the multiplicity operator in Theorem 3.3.7 does help with reducing the number of matrix multiplications, however it is still required to find the pseudo inverses of all the $B_{q,2}^{s,t}$ matrices. Therefore, in this section a method is proposed that finds $(B_{q,2}^{s,t})^\dagger B_{q,2}^{s,t}$ for each combination of start times s and end times t.

Afterwards, calculating the multiplicity operator or the modified multiplicity operator $\tilde{M}_q^{s,t}$ only requires 2 or 3 matrix multiplications respectively.

In a filtration, subsequent $B_{q,2}$ matrices differ only slightly. Therefore, it seems unnecessary to recalculate the pseudo inverse every time. From Figure 3.5, we can see that subsequent steps in the end time t follow the following pattern,

$$B_{q,2}^{s,t} = \begin{bmatrix} B_{q,2}^{s,t-1} & [\partial_{q+1}^t](I_q^{s,t-1}, I_{q+1}^{t-1,t}) \\ 0 & [\partial_{q+1}^t](I_q^{t-1,t}, I_{q+1}^{t-1,t}) \end{bmatrix}.$$

Assuming $(B_{q,2}^{s,t-1})^\dagger$ is known, we would like to find a formula for $(B_{q,2}^{s,t})^\dagger$. This is done in two steps. First the pseudo inverse of $\begin{bmatrix} B_{q,2}^{s,t-1} \\ 0 \end{bmatrix}$ is found, formulated in Lemma 3.5.1. With this inverse, the full $(B_{q,2}^{s,t})^\dagger$ can be acquired, which is done with Greville's form [14]. Finally the full algorithm can be formulated that deals with when each method needs to be used, see Algorithm 1.

Lemma 3.5.1. *For a matrix $A \in \mathbb{R}^{m\times n}$, the pseudo inverse of $\begin{bmatrix} A \\ 0 \end{bmatrix}$ is given by,*

$$\begin{bmatrix} A \\ 0 \end{bmatrix}^\dagger = \begin{bmatrix} A^\dagger & 0 \end{bmatrix}. \tag{3.36}$$

Proof. Following definition 2.1.1, we check the properties.

$$\begin{bmatrix} A \\ 0 \end{bmatrix}\begin{bmatrix} A^\dagger & 0 \end{bmatrix}\begin{bmatrix} A \\ 0 \end{bmatrix} = \begin{bmatrix} AA^\dagger A \\ 0 \end{bmatrix} = \begin{bmatrix} A \\ 0 \end{bmatrix},$$

$$\begin{bmatrix} A^\dagger & 0 \end{bmatrix}\begin{bmatrix} A \\ 0 \end{bmatrix}\begin{bmatrix} A^\dagger & 0 \end{bmatrix} = \begin{bmatrix} A^\dagger AA^\dagger & 0 \end{bmatrix} = \begin{bmatrix} A^\dagger & 0 \end{bmatrix},$$

$$\left(\begin{bmatrix} A \\ 0 \end{bmatrix}\begin{bmatrix} A^\dagger & 0 \end{bmatrix}\right)^T = \begin{bmatrix} (AA^\dagger)^T & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} AA^\dagger & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} A \\ 0 \end{bmatrix}\begin{bmatrix} A^\dagger & 0 \end{bmatrix},$$

$$\left(\begin{bmatrix} A^\dagger & 0 \end{bmatrix}\begin{bmatrix} A \\ 0 \end{bmatrix}\right)^T = (A^\dagger A)^T = A^\dagger A = \begin{bmatrix} A^\dagger & 0 \end{bmatrix}\begin{bmatrix} A \\ 0 \end{bmatrix}.$$

□

With the ability to extend the pseudo inverse over the added rows, now adding columns is discussed. The used method uses an old theorem from Greville, which provides a method to obtain the pseudo inverse of $A_k = \begin{bmatrix} A_{k-1} & a_k \end{bmatrix}$ for some $A_{k-1} \in \mathbb{R}^{m\times n}$ and some vector $a_k \in \mathbb{R}^m$, provided the pseudo inverse of A_{k-1} is known. Iteratively using the method then yields an algorithm to add any number of columns to the matrix and to compute the pseudo inverse. The method for one column can be found in Theorem 3.5.2.

Theorem 3.5.2 (Greville's form [14])**.** *For a matrix $A_{k-1} \in \mathbb{R}^{m\times n}$ and a vector $a_k \in \mathbb{R}^m$, with known pseudo inverse $A_{k-1}^\dagger$. Let $d_k := A_{k-1}^\dagger a_k$ and $c_k := a_k - A_{k-1}d_k$. Furthermore, define*

$$b_k = \begin{cases} c_k^\dagger & \text{if } c_k \neq 0. \\ \frac{1}{1+d_k^T d_k} d_k^T A_{k-1} & \text{if } c_k = 0. \end{cases} \tag{3.37}$$

Then

$$\begin{bmatrix} A_{k-1} & a_k \end{bmatrix}^\dagger = \begin{bmatrix} A_{k-1}^\dagger - d_k b_k \\ b_k \end{bmatrix}. \tag{3.38}$$

Note that the pseudo inverse of c_k is relatively easy to obtain, as for a vector, the pseudo inverse is given by $c_k^\dagger = \frac{1}{||c_k||_2^2} c_k^T$.

Using Lemma 3.5.1 first and then iteratively adding columns with Theorem 3.5.2, we obtain an algorithm to compute $B_{q,2}^{s,t}$ using the pseudo inverse of $B_{q,2}^{s,t-1}$. Thereby reducing the number of pseudo

inverses to be calculated to the number of time steps, as for every s still one pseudo inverse needs to be calculated. The full algorithm to compute $B_{q,2}^{s,t}$ for every s and t is formulated in Algorithm 1. This algorithm also deals with the fact that $B_{q,2}^{s,t}$ does not exist if $n_{q+1}^{s} = n_{q+1}^{t}$.

Finally, note that time complexity can be further reduced by realizing that $M_q^{s,t} = 0$ if $n_{q+1}^{t} = n_{q+1}^{t-1}$ or if $n_q^s = n_q^{s-1}$. By Theorem 3.3.10, this also means that $\tilde{M}_q^{s,t} = 0$. Therefore, in these cases, the required matrix multiplications to obtain the matrices are not needed and no eigenvalues need to be found. However, $(B_q^{s,t})^\dagger$ still needs to be computed as it is needed for $(B_q^{s,t+1})^\dagger$ if it exists.

Algorithm 1: Computing the pseudo inverses

Data: $[\partial_q], [\partial_{q+1}]$
Result: $\{(B_{q,2}^{s,t})^\dagger B_{q,2}^{s,t}\}_{T_{start} \le s,t \le T_{end}}$
for $s \leftarrow T_{start}$ *to* T_{end} **do**
  Obtained_B22 ← False;
  for $t \leftarrow s$ *to* T_{end} **do**
    if *Obtained_B22 = False* **then**
      if $n_{q+1}^s = n_{q+1}^t$ **then**
        $(B_{q,2}^{s,t})^\dagger B_{q,2}^{s,t} \leftarrow []$
      else
        if $n_q^s = n_q^t$ **then**
          $(B_{q,2}^{s,t})^\dagger B_{q,2}^{s,t} \leftarrow [0]$;
        else
          $A_{k-1} \leftarrow [\partial_{q+1}](I_q^{s,t}, I_{q+1}^{s,t})$;
          $A_{k-1}^\dagger \leftarrow (A_{k-1})^\dagger$ $(B_{q,2}^{s,t})^\dagger B_{q,2}^{s,t} \leftarrow A_{k-1}^\dagger A_{k-1}$;
          $t_A \leftarrow t$;
          Obtained_B22 ← True;
        end
      end
    else
      Changed_B22=False;
      if $n_q^t > n_q^{t_A}$ **then**
        $A_{k-1} \leftarrow \begin{bmatrix} A_{k-1} \\ 0 \end{bmatrix}$;
        $A_{k-1}^\dagger \leftarrow \begin{bmatrix} A_{k-1}^\dagger \\ 0 \end{bmatrix}$;
        Changed_B22=True;
      end
      if $n_{q+1}^t > n_{q+1}^{t_A}$ **then**
        for $c \leftarrow n_{q+1}^{t_A}$ *to* n_{q+1}^t **do**
          $a_k \leftarrow [\partial_{q+1}](I_q^{s,t}, c) d_k \leftarrow A_{k-1}^\dagger a_k$;
          $c_k \leftarrow a_k - A_{k-1} d_k$;
          $b_k \leftarrow \begin{cases} c_k^T / ||c_k||_2^2, & \text{if } ||c_k||_2^2 \neq 0 \\ (1 + ||d_k||_2^2)^{-1} d_k^T A_{k-1}^\dagger & \text{otherwise} \end{cases}$;
          $A_{k-1}^\dagger \leftarrow \begin{bmatrix} A_{k-1}^\dagger - d_k b_k \\ b_k \end{bmatrix}$;
          $A_{k-1} \leftarrow \begin{bmatrix} A_{k-1} & a_k \end{bmatrix}$;
          Changed_B22=True;
        end
      end
      if *Changed_B22* **then**
        $t_A \leftarrow t$;
        $(B_{q,2}^{s,t})^\dagger B_{q,2}^{s,t} \leftarrow A_{k-1}^\dagger A_{k-1}$;
      else
        $(B_{q,2}^{s,t})^\dagger B_{q,2}^{s,t} \leftarrow (B_{q,2}^{s,t-1})^\dagger B_{q,2}^{s,t-1}$;
      end
    end
  end
end

4

Applications

In this chapter two possible applications of the persistent Laplacian are discussed, where the use of standard persistent homology is compared with using the multiplicity operators defined in Section 3.3.3.

4.1. MNIST Dataset

In this section the multiplicity operator is applied to the MNIST dataset [2]. The dataset contains 70.000 images, but in this thesis just a random sample of 7000 images is used. Each image is made into a graph using the Ball Mapper method discussed in Section 2.2. The vertices of the resulting graph are given a weight equal to the average x value or y value of the points in its corresponding cluster. Thereby creating a vertex weighted graph $G = (V, E)$ with weight function $w : V \to \mathbb{R}_{\geq 0}$. This allows us to use the vertex-based clique filtration (VBCL), see Definition 2.4.10.

For the 6 represented in Figure 2.1 the resulting filtration is plotted in Figure 4.1. Here the average y value is used as weight for the vertices in the Ball Mapper graph. Points on top of the image have the lowest y-values, therefore it can be seen that first the stem of the 6 appears in the filtration and the cycle is closed only at the very end. Note that the filtration is different from the filtration of Figure 2.10, because the radii of the balls are different.

Using this filtration, we can apply standard persistent homology. The landscape corresponding to the filtration is shown in Figure 4.2. Here it can be seen that only one finite interval is found in the barcode as only one non-zero function is present across both dimensions. While a cycle appears in the filtration of the 6, it is not shown in the landscape as it does not die. The function that is present in the landscape is due to the fact that the drawing is tilted a bit, such that at some values of y, there are two connected components.

If we instead look at the Laplacian landscape, see Figure 4.3, we can see a bit more information. Because the persistent Laplacian encodes geometric information, it changes more often and the landscape contains more noise. Therefore visually the diagrams become harder to interpret. However, what we can see from the figure is that the second highest peak is maximal at the exact same time as the peak in the persistence landscape. Furthermore, after the two connected components merge, topologically nothing more happens, but in the Laplacian landscape, we do see some more features. The final function of the landscape is zero at $t = 19$, which means some of these features correspond to the cycle. The dimension 1 landscape also contains two features, however these are thought to mainly be due to noise and no interpretation was found.

4.1.1. Building a first classifier

In order to build a classifier, for each image in the dataset, both a persistence landscape as well as a Laplacian landscape are computed. The images are split into a training set and a test set. The training set contains 80% of the images (5600 images), while the test set contains the remaining 20% (1400 images). Landscapes of the training set are used to compute a mean landscape for each digit, like is done in [43]. If $\lambda_{q,i}$ is the landscape corresponding to image i of the training set, the mean landscape

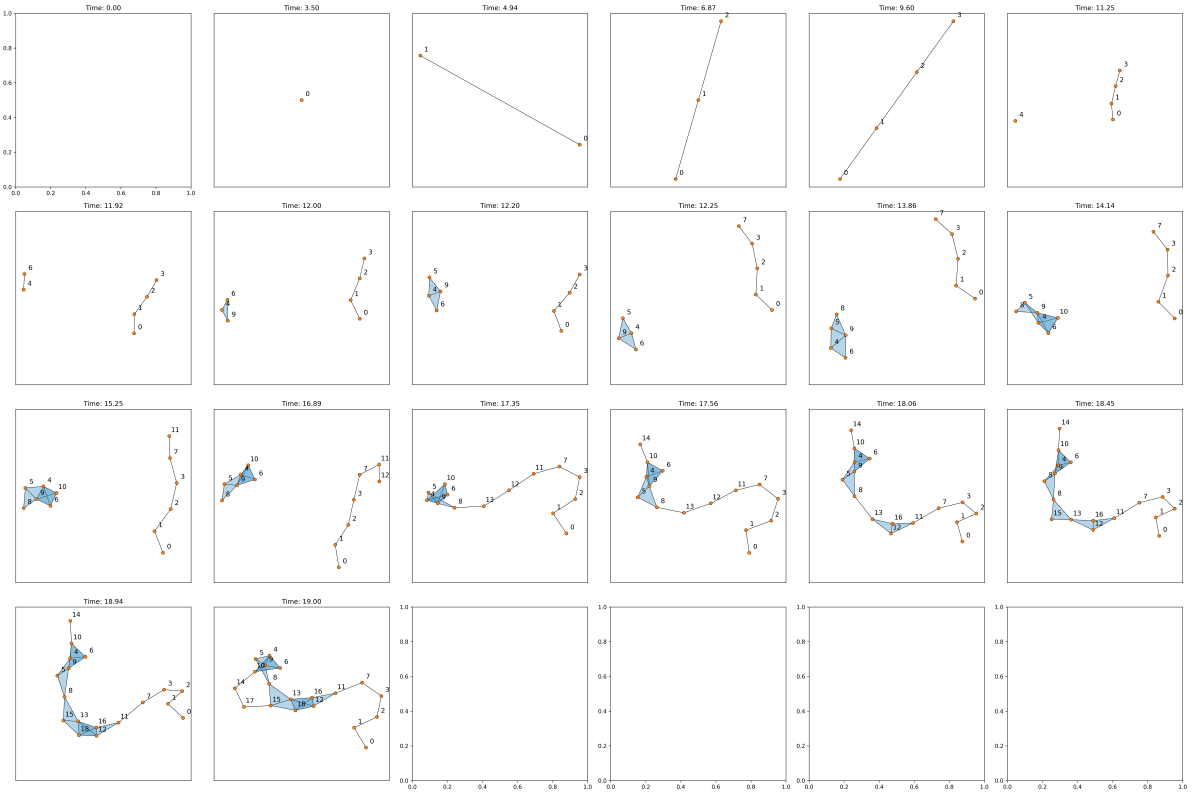


Figure 4.1: Vertex-based clique filtration of the graph output by Ball Mapper, with $\varepsilon = 3$, on the point cloud in Figure 2.1. Weights of the vertices of the graph are given by the average y value of the points in the corresponding cluster. Points at the top of the figure have the lowest y value. Only the timesteps where a simplex is added are shown.

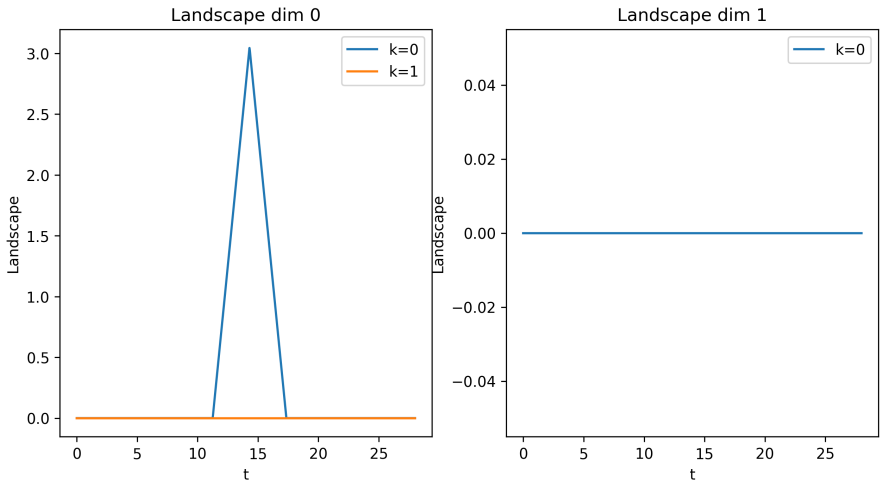


Figure 4.2: Persistence landscape for the filtration of Figure 4.1.

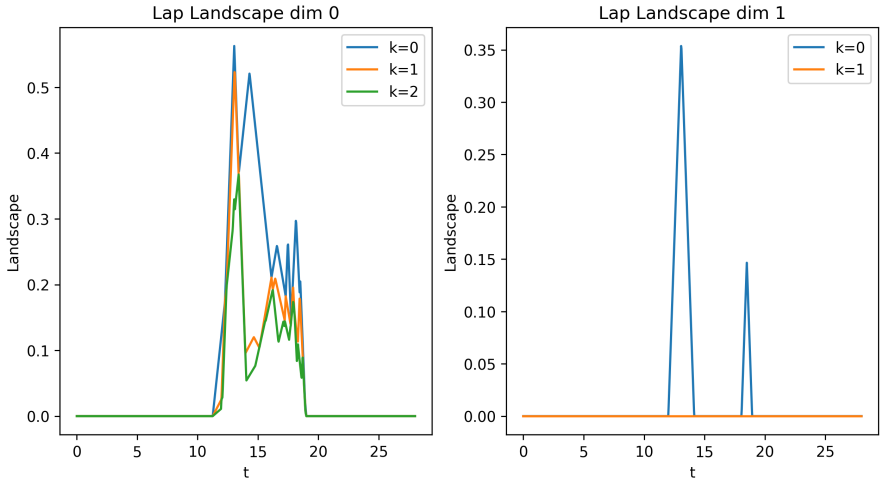


Figure 4.3: Laplacian landscape for the filtration of Figure 4.1.

$\bar{\lambda}_{q,d}$ corresponding to digit d is given by

$$\bar{\lambda}_{q,d}(k,t) = \frac{1}{n}\sum_{i=1}^{n(d)} \lambda_{q,i}(k,t), \tag{4.1}$$

where $n(d)$ is the amount of samples of digit d in the training set. The same can be done for Laplacian landscapes to get

$$\bar{\lambda}^l_{q,d}(k,t) = \frac{1}{n}\sum_{i=1}^{n(d)} \lambda^l_{q,i}(k,t). \tag{4.2}$$

Because these mean landscapes are again a landscape, for each digit, we can plot them to see how they look, see Figure 4.4. Here we can see a clear difference in the average landscape of the different digits. For the persistence landscapes, only $k = 0$ yields something relevant, however mostly within one standard deviation of 0. Only the 4 has values of t, where it is significantly above zero. This is due to the fact that the 4 is often drawn with two stems, like in Figure 1.1. Therefore, it often starts with two connected components yielding this plot. The same feature can also be seen in the corresponding Laplacian landscape, however some additional information is added after it dies.

Looking at the mean persistent Laplacian landscapes of the 6 and the 9, we can see that they are almost flipped horizontally. This makes intuitive sense as scanning the digits from top to bottom visually is almost exactly inverted. Using the interpretation of the previous chapter, it can also be explained. The multiplicity operator is non-zero whenever a new path between vertices appears. So whenever the cycle in the 6 or the 9 closes, the 0-simplices that appeared at the top of the cycle have a new path which travels along the boundary of the cycle. For the 9, this happens with simplices that appear early and get connected again around halfway. For the 6 the first simplices of the loop appear halfway and get connected at the end. This explains the most prominent features of these persistent Laplacian landscapes.

To make a prediction of the digit in a new image, the distance d_{land} between the corresponding landscape and each of the mean landscapes is computed, see Section 2.4.3. The smallest distance is then considered the predicted digit. This can be done for both the persistence landscape and the Laplacian landscape. Finally a combination of the two is analyzed, where the distances between the persistence landscapes and the Laplacian landscapes are normalized to be between 0 and 1 and then added together. Thereby creating a new distance for each digit, where the smallest is chosen as the predicted number.

This is done for each of the images in the test set and the accuracies are denoted in Table 4.1. Note that the same experiment is done with different sizes ε of the balls in Ball Mapper. Furthermore, the experiment is also repeated by instead giving weights based on the average x value of points in a cluster, i.e. scanning the image from left to right instead of from top to bottom.

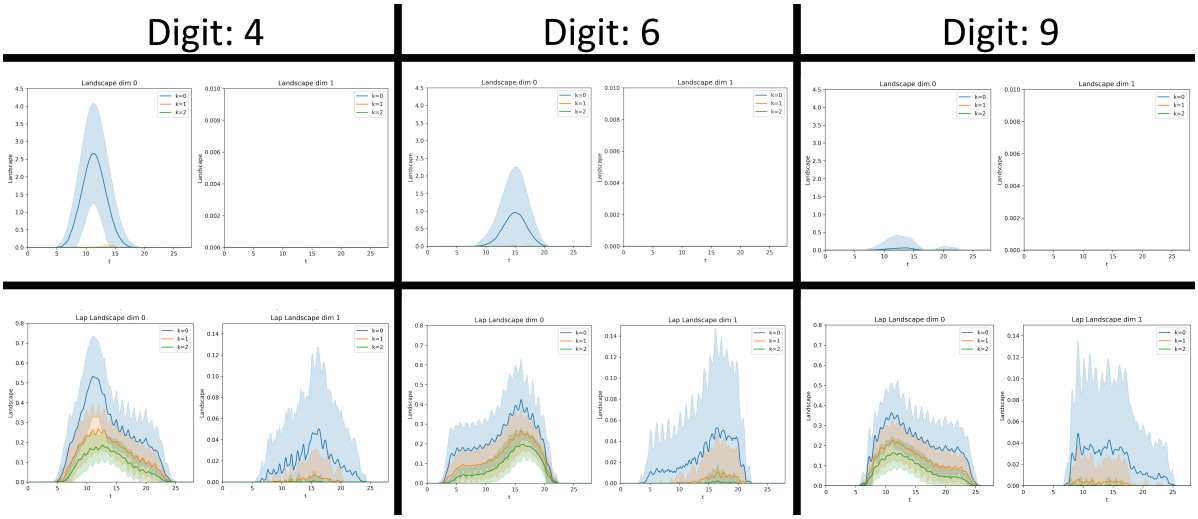


Figure 4.4: Mean persistence landscapes in the top row and persistent Laplacian landscapes in the bottom row with sample standard deviation around the mean. Landscapes are computed for the digits 4, 6 and 9 based on 5600 samples of the MNIST dataset. $\varepsilon = 2$ is used for the Ball Mapper parameter and the average y-value of the points in a cluster is used as the weight of the corresponding vertex in the graph.

Table 4.1: Classification performance obtained by applying the landscape methods to the first 7000 images in the MNIST dataset. 5400 are used to train the classifier and 1600 are used to obtain the noted accuracy. Persistent landscapes from persistent homology (PH) are compared to Laplacian landscapes from the trace of the multiplicity operator $[M_q^{S,t}]$ (MULT). Finally a combination of the two classifiers is used (COMBI). Different sizes of the Ball Mapper parameter ε are tested, together with different weight function for the VBCL. Here, x stands for taking the average x-value of points within a cluster and y for taking the average y-value of points within a cluster.

ε	1.5		2		2.5		3	
Weight function	x	y	x	y	x	y	x	y
PH	29.6%	33.2%	28.2%	32.0%	27.6%	30.2%	25.4%	28.1%
MULT	38.5%	53.2%	42.7%	58.2%	36.4%	50.5%	36.4%	45.9%
COMBI	**45.5%**	**56.4%**	**46.8%**	**61.2%**	**43.1%**	**55.8%**	**41.3%**	**50.9%**

Table 4.2: Classification performance of each digit applying the landscape methods to the first 7000 images in the MNIST dataset. 5400 are used to train the classifier and 1600 are used to obtain the noted accuracy. $\varepsilon = 2$ is used as size of the Ball Mapper parameter. The weight function for the VBCL is taken as the average y-value of points within a cluster. Persistent landscapes from persistent homology (PH) are compared to Laplacian landscapes from the trace of the multiplicity operator $[M_q^{S,t}]$ (MULT). Finally a combination of the two classifiers is used (COMBI).

Type\Digit	0	1	2	3	4	5	6	7	8	9
PH	0.0%	**98.8%**	12.1%	34.1%	79.3%	17.0%	30.3%	17.0%	8.2%	2.3%
MULT	**78.2%**	48.8%	44.7%	39.1%	64.3%	51.9%	**67.6%**	61.2%	67.2%	60.8%
COMBI	73.9%	53.5%	**47.7%**	**41.3%**	**80.0%**	**52.6%**	63.4%	**62.6%**	**70.5%**	**68.5%**

From Table 4.1, it becomes clear that the multiplicity operator outperforms standard persistent homology, achieving the best classification performance and it was better in every configuration tested. This can be explained by the fact that PH often failed to find any features that appear in the persistence landscape. Which could happen for many of the numbers and every time it happened, using this method, the prediction would be the same. Therefore, many numbers are misclassified this way. It therefore may not be completely fair to compare the two methods, however it does show the additional information that can be captured by the persistent Laplacian.

It is however interesting that using a combination of the two methods yielded the best accuracy, also across every tested configuration. This would suggest that not all the topological information is encoded in the Laplacian landscape and adding that information can be of benefit. Note that only one function was tested to combine the two methods while numerous options exist. One could for example add a weight to prefer one method over the other, which was not done here.

Furthermore, it can be seen that $\varepsilon = 2$ seems to be the optimal parameter for the ball size for the Laplacian landscapes. Increasing and decreasing the parameter value seemed to all decrease the total classification performance. However, for persistent landscapes, the accuracy seemed to increase the lower the value of ε. This is theorized to be because at lower values of the parameter, more geometric information is captured as the filtration is more sensitive to smaller changes.

Finally, it is interesting that the weight function which took the average x value of the points in each cluster performed considerably worse. Apparently scanning the images from left to right does not give many interpretable features independent of using persistence landscapes or Laplacian landscapes. Intuitively this seems to make sense as most numbers are drawn top to bottom.

Fixing $\varepsilon = 2$ and scanning the images from top to bottom, we can obtain the per digit accuracies of each of the models, see Table 4.2. Here it becomes clear that the persistent homology based model mainly predicts a 1, probably whenever no features are found. Therefore, its classification performance on the 1 is very high, but most of the other accuracies are low. Only the 4 it can predict well, which is due to the previously discussed two stems commonly used to draw the digit.

The Laplacian landscape and combination models both perform reasonably consistent across the different numbers. However, the digits without a cycle still appear to be more difficult. Here the 3 attains the lowest classification performance, probably due to its similarity to the 5. When half the image has been *scanned* the numbers are the same up to horizontal symmetry. This method is invariant to these symmetries, so the landscapes should look very similar at the start. The number 4 yielded the highest classification performance in the combination model, which can mainly be explained by the persistence landscape. Finally, the 0 obtained the highest classification performance for the Laplacian model, probably because the gap in the 0 is often relatively big and should always be visible in the Ball Mapper graph.

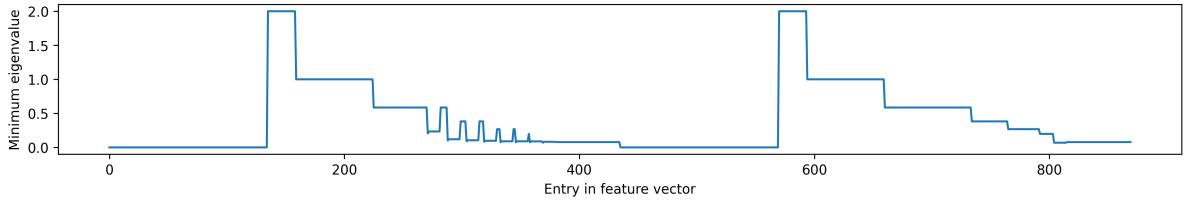
4.1.2. Comparing multiplicity to the persistent Laplacian

Using the persistent Laplacian without the multiplicity operator works differently. As it cannot be used in the same way and while technically a landscape could still be made, it would have a very high number of triangular functions, making any operations very slow. Therefore, to compare the multiplicity operator to the persistent Laplacian, a new classifier needs to be defined.

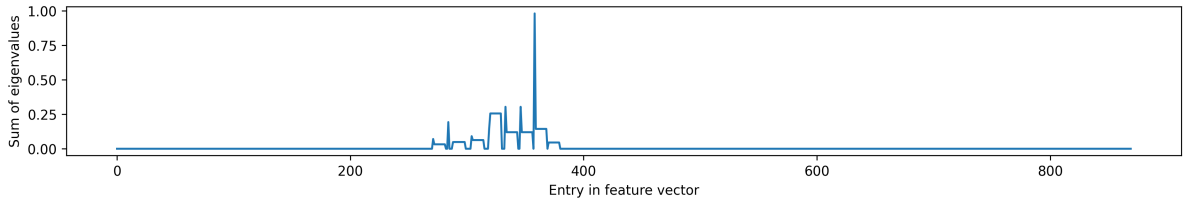
In [16], the authors extract a feature vector from the persistent Laplacians, which is then used to train a simple linear classifier. In order to create the feature vector, they select a set of filtration parameters and compute for each combination the persistent Laplacian. They concatenate the eigenvalues of all the Laplacians and use that as their feature vector. However, because some Laplacians have more eigenvalues, they sometimes need to append zeros or remove some of them to keep all feature vectors the same size. In order to simplify the process, in this thesis the minimum non-zero eigenvalue of each persistent Laplacian is used instead.

The set of filtration parameters used is $S = \{0, 1, \dots, 28\}$, so all possible pixel locations. The start time s is iterated from 0 to 28 and then the end time t from s to 28, afterwards the process is repeated for dimension 1. Therefore, the first entry in the vector corresponds to the minimum eigenvalue of $[\Delta_0^{0,0}]$, while the second entry corresponds to the minimum eigenvalue of $[\Delta_0^{0,1}]$. The 30-th entry in the vector corresponds to the minimum eigenvalue of $[\Delta_0^{1,1}]$ and the 436-th ($\frac{29 \cdot 30}{2} + 1$) entry corresponds to the minimum eigenvalue of $[\Delta_1^{0,0}]$.

For each image, the persistent Laplacian and the multiplicity operator are only calculated between



(a) Minimum eigenvalue of the persistent Laplacian.



(b) Sum of eigenvalues of the multiplicity operator.

Figure 4.5: Feature vectors using the filtration of Figure 4.1. First half of each of the vectors corresponds to dimension 0, the second half to dimension 1.

values of the filtration parameter where a simplex is added. Most of the time this is not on exactly the values in S. Therefore, for $s, t \in S$ where $s \leq t$, the used persistent Laplacian is the one that was computed with a start time closest to s and end time closest to t. Furthermore, note that most digits in the dataset do not use the first and last few pixels, therefore the persistent Laplacian is often 0.

The two vectors can be analyzed by plotting the entries, see Figure 4.5. Note that the multiplicity operator is zero for the whole of dimension 1, while the landscape in Figure 4.3 indicates that two features do exists. However, because only the evaluations at the points in S are captured in the vector, it may be possible that some non-integer timesteps are skipped. If between two integer timesteps three or more evaluations exist, the middle one does not appear in the feature vector.

To understand the effect of the multiplicity operator, we still want to compare the two vectors. The idea of the multiplicity operator is to only be non-zero on the points where the persistent Laplacian changes. However, in the previous chapter, it was already noted that information from the down Laplacian is not encoded. Comparing the two vectors like this is therefore hard as in the persistent Laplacian vector, it is unclear which information came from the down Laplacian. However, it is clear that representing the multiplicity operator this way yields an almost zero vector.

Besides using persistent Laplacian and the multiplicity operator, also a vector is obtained using the combinatorial Laplacian. Where the persistent Laplacian requires a start and end time, the combinatorial Laplacian only requires one time. Computing the minimum eigenvalue of the combinatorial Laplacian on every step in S therefore yields a smaller vector. Nevertheless, this may still provide useful information.

Computing the feature vectors for each image in the dataset, we obtain a dataset of feature vectors with corresponding labels. To now make predictions any classification Machine Learning method can be used. The focus of this thesis is on the feature vector and not on the ML method, therefore a simple Logistic Regression (LR) model is chosen. The dataset is again split into a training set of 5400 images and a test set of 1600 images. The LR model is trained on the feature vectors of the training set and then tested on the feature vectors of the test set. The resulting accuracies can be seen in Table 4.3.

It can be seen that the persistent Laplacian method, as well as the combinatorial Laplacian both outperform the multiplicity operator in this way. It is theorized that this is because of the effect seen in Figure 4.5b. There, no features can be seen in dimension 1 because of the chosen set S. This does not effect the persistent Laplacian as much as at every timestep it contains the effect of all the previous steps. However, the multiplicity operator is designed to only contain information of the actual step, therefore using it in this way, most of the information is lost.

Instead, it should be used in the same way as persistent homology. Therefore, a second feature vector is made using the Laplacian landscapes. In order to keep the vector roughly the same size as the persistent Laplacian vectors, the landscape is summed over all values of k, $\Sigma\lambda^l_{LD}(t) = \sum_k \lambda^l_{LD}(k,t)$.

Table 4.3: Accuracies on the test set for the models that extract a feature vector based on a selection of filtration parameters and use Logistic Regression (LR) to do predictions. Feature vectors are extracted using the sum of eigenvalues of the multiplicity operator $[M_q^{s,t}]$ (MULT), using the minimum non-zero eigenvalue of the persistent Laplacian $\Delta_q^{s,t}$ (PL) and the minimum non-zero eigenvalue of the combinatorial Laplacian Δ_q^s (CL). Finally, feature vectors are obtained from the summed Laplacian landscape on 333 locations in each dimension (LAPLAND). Training of the LR model is done on 5400 images and testing on 1600 images. Different sizes of the Ball Mapper parameter ε are tested, together with different weight function for the VBCL. Here, x stands for taking the average x-value of points within a cluster and y for taking the average y-value of points within a cluster.

ε	1.5		2		2.5		3	
Weight function	x	y	x	y	x	y	x	y
PL	65.1%	77.4%	64.5%	75.2%	62.0%	75.4%	57.8%	**75.6%**
CL	52.9%	67.5%	51.4%	68.2%	46.5%	66.0%	43.9%	66.1%
MULT	42.9%	66.4%	42.3%	60.9%	36.4%	58.0%	40.7%	58.0%
LAPLAND	56.8%	75.9%	53.1%	74.3%	45.0%	61.8%	42.5%	59.8%
LAPLAND+CL	**72.4%**	**81.9%**	**67.6%**	**81.1%**	**64.4%**	**76.9%**	**59.1%**	75.3%

Afterwards the interval $[0, 28]$ is discretized to obtain 333 values of t, where this summed landscape is evaluated. Doing this for both dimension 0 as well as dimension 1, a feature vector is obtained of length 666. This vector is used in the same way as the other feature vectors to make predictions with a LR model. The results can also be found in Table 4.3.

We can see that this greatly increases the accuracy of the multiplicity based models and it now closely matches the accuracies obtained from the persistent Laplacian. Still it remains a bit less accurate, especially at higher values of ε. This is thought to be because the information of the combinatorial Laplacian is not encoded into the multiplicity operator and therefore also not in the Laplacian landscape. For high values of ε, the step in each subsequent complex in the filtration contains more information as each simplex corresponds to more pixels. In the combinatorial Laplacian, the information of each complex is encoded. The multiplicity operator does not encode any information of the combinatorial Laplacian, so of $\Delta_q^{s,t}$, where $s = t$ and instead only describes when it changes in a subsequent time step $t > s$. Therefore it misses this information.

To test the validity of the previous statement, a final feature vector is used, which is the concatenation of the vector obtained from the landscape and the vector from the combinatorial Laplacian. The resulting vector has a size of 724, which is still less than the vector from the persistent Laplacian. In Table 4.3 accuracies of this vector are also noted. It can be seen that this final vector gave the best results in almost every configuration tested and therefore supports the previous argument. While for a linear model this concatenation is easily done, for more complex models, it would be better to include the information directly in the multiplicity operator.

4.2. Identifying crystalline structures

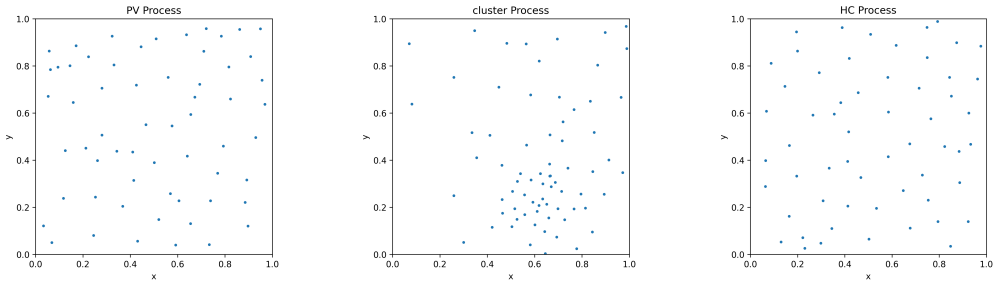
We now turn to the problem of classifying crystalline structures, see Figure 1.2 for an example. These images are made by slicing an alloy in half and taking a picture using a microscope. The focus of this application is on the red lines, which represents Kernel Average Misorientation (KAM). The pattern of these lines is thought to contain information about the type of alloy and its properties.

While the methods are meant to be used on real images of cross sections of alloys, data for this can be hard to acquire. Therefore, certain point processes are used to artificially generate the location of the centers of the cells, see Section 2.3. The goal of this application is to detect the underlying generation process for the artificial cross sections. The images are made using either the standard Poisson-Voronoi diagram (PV), cluster method or the Hard-Core method (HC).

4.2.1. Alpha filtration

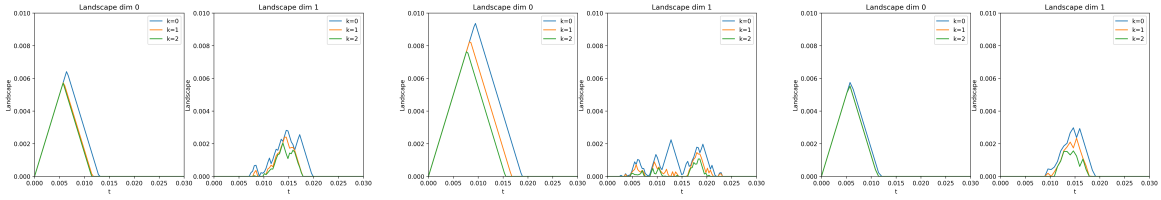
In order to use TDA methods, the authors of [43] describe these types of images by an alpha filtration, see Definition 2.4.8. An alpha filtration requires a point cloud, which is obtained by taking the center of each of the cells in the cross section. Because the structure of the cells is thought to be similar to a Voronoi diagram, the alpha complex may be relevant for describing this structure as it also relies on a Voronoi diagram.

Taking the center of each of the clusters in the artificially generated images of Figure 2.3, we obtain the point clouds visualized in Figure 4.6. These point clouds can be used to make an alpha filtration.



(a) Poisson Voronoi centers. (b) Cluster centers. (c) Hard core centers.

Figure 4.6: Centers of the cells in the cross sections of three dimensional Voronoi diagrams. Each is sampled using a different point cloud generation method while the amount of sampled points is kept to be between 225 and 275 over a unit cube.



(a) Persistence landscape of PV. (b) Persistence landscape of cluster. (c) Persistence landscape of HC.

Figure 4.7: Persistence landscapes of the point clouds visualized in Figure 4.6 using an alpha filtration.

However, visualizing this filtration was too complex as there are too many simplicial complexes which all contain too many simplices.

Instead, we can look at the corresponding persistence landscapes of the alpha filtration of the point clouds, see Figure 4.7. Here, we can already see that the landscapes corresponding to the PV and HC point processes are quite similar, however the persistence landscape of the cluster point process does seem to be different as in both dimensions it is non-zero for longer. This is because in the cluster process there are some sections which have very few cells and therefore cell centers. These points only get connected at higher values of r and therefore still generate features at higher values of r.

Instead of looking at one landscape, we instead focus on finding a mean landscape again. For each point process method 100 images are made and for each one a persistence landscape and Laplacian landscape are computed. The mean together with standard deviation can be seen in Figure 4.8. Interesting here is that the dimension 1 persistence landscape is very similar in shape to the Laplacian landscape. This shows that not many additional features are found using the multiplicity operator.

For dimension 0 a difference can be seen. Note that points can only be connected in the alpha complex if their corresponding Voronoi cells are adjacent. If a cell extends far in a certain direction, this connection is made at a late state, after all of the points are already connected and therefore do not correspond to a topological feature. Geometrically, these connections all correspond to a new immediate path between two 0-simplices, therefore in the multiplicity operator a high weight is attributed. This would explain features up to a value of around $t = 0.5$ as the probability that these cells are bigger than that is very low. Nevertheless, some features can still be found for higher values of t with the multiplicity operator. This is because the boundary of the figure is not taken into consideration, therefore the cells extend beyond the unit square, which can make very big cells. A good solution to prevent this was not found, the only thing done is to only consider values of t up to 1 as this is a theoretical max bound for cells within the square. Note that this causes the dimension 0 Laplacian landscapes of all the different generation methods to be very similar, which is an issue later on.

In order to see the difference between the multiplicity operator $[M_q^{s,t}]$ and the adjusted multiplicity operator $\tilde{M}_q^{s,t}$, the same mean Laplacian landscape is computed for the trace of $\tilde{M}$. Because the landscapes are very similar, only the difference is plotted in Figure 4.9. A positive value in this figure corresponds to the trace of $\tilde{M}$ to be bigger, while a negative value would correspond to the trace of $[M]$ to be bigger, however the latter was never found. The domain and range of the plot are kept the same as the plot in Figure 4.8. One can note that the landscapes are nearly identical, therefore mod-

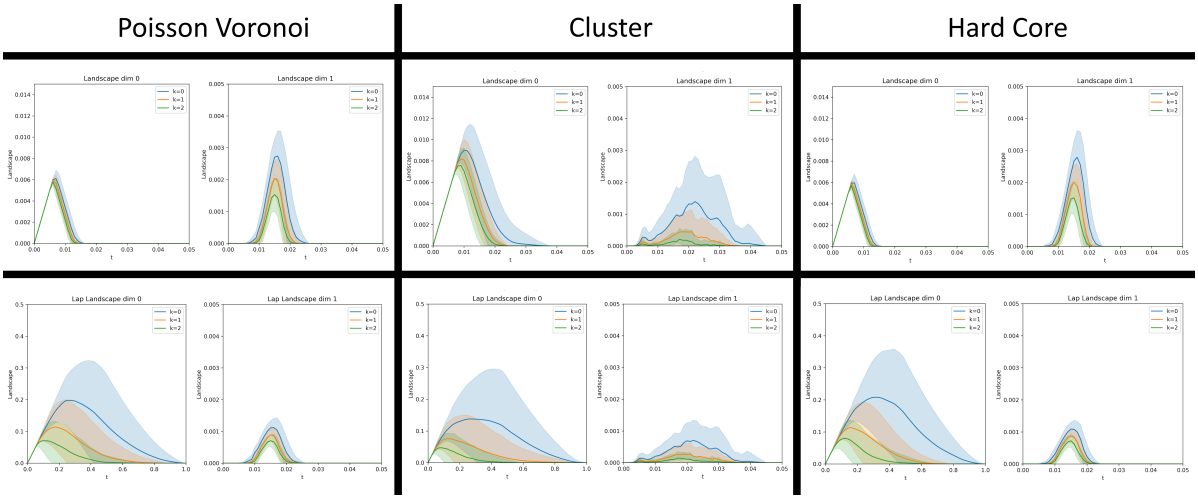


Figure 4.8: Mean persistence landscapes in the top row and mean Laplacian landscapes on the bottom row with sample standard deviation. Landscapes are computed for cross sections of Poisson Voronoi, Cluster and Hard Core point processes, using an alpha filtration. Mean and standard deviation are taken over 100 samples per plot.

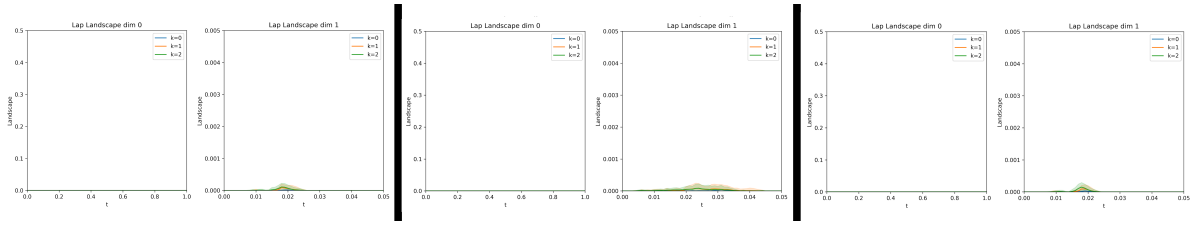


Figure 4.9: Difference between the mean Laplacian landscape based on the trace of $\tilde{M}_q^{s,t}$ and the trace of $[M_q^{s,t}]$, together with standard deviation of the difference. Positive values correspond to the trace of $\tilde{M}_q^{s,t}$ to be higher and negative values to the trace of $[M_q^{s,t}]$ to be higher, however the latter is never obtained and therefore only positive values are shown. The image follows the same structure as Figure 4.8 and is also taken over 100 samples per landscape.

els based on either trace should give very similar results. This is true at least when using this alpha filtration approach.

Instead of making a classifier like was done for the MNIST dataset, a few test statistics are proposed, similar to [43]. For each generation method m, L_0^m and L_1^m are based either on the found 0 dimensional features or the 1 dimensional features, respectively.

$$L_0^m := ||\hat{\lambda}_0 - \bar{\lambda}_{0,m}||_2 = \left[\sum_{k=1}^{\infty} \int_0^T (\hat{\lambda}_0(k,t) - \bar{\lambda}_{0,m}(k,t))^2 dt\right]^{\frac{1}{2}}, \tag{4.3}$$

$$L_1^m := ||\hat{\lambda}_1 - \bar{\lambda}_{1,m}||_2 = \left[\sum_{k=1}^{\infty} \int_0^T (\hat{\lambda}_1(k,t) - \bar{\lambda}_{1,m}(k,t))^2 dt\right]^{\frac{1}{2}}, \tag{4.4}$$

where $\hat{\lambda}_q$ is the q-dimensional landscape of a new cross section and $\bar{\lambda}_{q,m}$ the mean landscape of generation method m. In the same way, the test statistics $L_0^{l,m}$ and $L_1^{l,m}$ can be defined by using a Laplacian landscapes based on the trace of the persistent multiplicity operator instead of the persistence landscape. Finally, $L_0^{a,m}$ and $L_1^{a,m}$ are defined using Laplacian landscapes based on the trace of the adjusted multiplicity operator.

Let the null hypothesis be that $\hat{\lambda}$ corresponds to generation method m. We would like to find the critical region of the test statistic such that the hypothesis can be rejected. An approximation of this critical region can be found using the generated cross sections by finding realizations of the test statistic under the null hypothesis. Like in [43], a 'leave one out' procedure is used. Therefore realizations are

Table 4.4: Fraction of rejections for different tests based on different null hypotheses and test statistics. For each test, the critical value is chosen with confidence level $\alpha = 0.95$ over 100 samples. Rejections are counted over 100 samples each. Type and q refer to the statistic used. Here PH refers to L_q^m, MULT to $L_q^{l,m}$ and ADJ to $L_q^{a,m}$.

(a) Null hypothesis: PV

Type	q	cluster	HC
PH	0	**1.00**	0.04
MULT	0	0.02	0.05
ADJ	0	0.02	0.05
PH	1	0.96	**0.07**
MULT	1	0.97	0.05
ADJ	1.	0.97	0.06

(b) Null hypothesis: cluster

Type	q	PV	HC
PH	0	**0.60**	**0.76**
MULT	0	0.07	0.05
ADJ	0	0.07	0.05
PH	1	0.14	0.16
MULT	1	0.09	0.07
ADJ	1	0.14	0.16

(c) Null hypothesis: HC

Type	q	PV	cluster
PH	0	**0.13**	**1.00**
MULT	0	0.05	0.02
ADJ	0	0.05	0.02
PH	1	0.06	0.94
MULT	1	0.10	**1.00**
ADJ	1	0.06	0.96

obtained as follows:

$$l_{0(i)}^m := \left[\sum_{k=1}^{\infty} \int_0^T (\hat{\lambda}_{0(i)}(k,t) - \bar{\lambda}_{0,M(-i)}(k,t))^2 dt \right]^{\frac{1}{2}},$$

$$l_{1(i)}^m := \left[\sum_{k=1}^{\infty} \int_0^T (\hat{\lambda}_{1(i)}(k,t) - \bar{\lambda}_{1,M(-i)}(k,t))^2 dt \right]^{\frac{1}{2}}.$$

Here, $\hat{\lambda}_{0(i)}(k,t)$ and $\hat{\lambda}_{1(i)}(k,t)$ correspond to cross section i of method m, $\bar{\lambda}_{0,M(-i)}(k,t)$ and $\bar{\lambda}_{1,M(-i)}(k,t)$ are the mean landscapes, computed using all sections leaving out the i-th.

Selecting a significance level α, a critical value for the test statistic is found by computing the corresponding quantile of the realizations under the null hypothesis. As the test statistic corresponds to a distance, small values should correspond to not rejecting the null-hypothesis. Therefore, a one sided test is used. To estimate the power of the resulting test, the statistic is computed for samples of the other generation methods and the fraction of rejections is noted in Table 4.4.

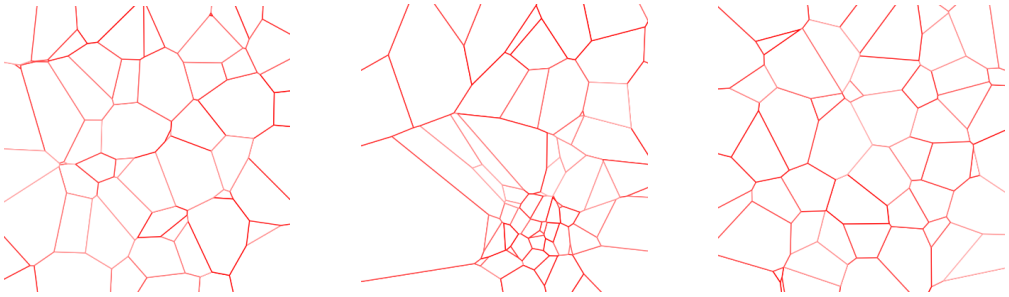
In this table it can be seen that the tests based on standard persistent homology almost always outperform the Laplacian based tests. Nevertheless, in testing under the null hypothesis of PV or HC, the resulting tests have comparable power. Here cluster point processes are easily rejected, but differentiating between PV and HC using these landscapes is near impossible as hypothesized. It can also be noted that no significant difference between the standard persistent multiplicity operator and the adjusted version is found. While their results are not exactly equal, they are not different enough to draw any conclusions.

Furthermore, it can be seen that the tests based on the multiplicity operators in dimension zero perform very poorly, which is due to the previously discussed similarity of the Laplacian landscapes in this dimension. The tests based on the cluster null hypothesis seem to need dimension 0 features to be of any power, which means that the Laplacian based tests do not function well. This is theorized to be because detecting the larger cells in the cluster cross sections is done when the filtration parameter is large enough to connect it another cell, creating a feature in dimension 0. Dimension 1 geometric features do not capture it as the length of edge is not encoded into the simplicial complex.

4.2.2. Using Ball Mapper

Because the alpha filtration did not seem to add many geometric features, the filtration used in the handwritten number recognition is also tested. Interpreting the images of Figure 2.3 as grey-scale and taking the pixels that have a value greater than some threshold as points, one can use Ball Mapper to create a graph. To create a filtration from this graph the VBCL is used. However, instead of using the average x or y value of the points in a cluster as the weight function, here we can use additional information normally found in real cross sections, see Figure 1.2. Not every edge in this structure has the same intensity of KAM, therefore the average KAM intensity could also be used to create a filtration, thereby encoding more information of the cross section into the filtration.

Translating this to the artificially generated images of Figure 4.6 means that an intensity for each edge can be chosen. Two options are considered for this. We can apply a gradient coloring, thereby



(a) Poisson Voronoi cross section. (b) Cluster cross section. (c) Hard core cross section.

Figure 4.10: Cross sections of three dimensional Voronoi diagrams where the edges are given a random intensity. Each is sampled using a different point cloud generation method while the amount of sampled points is kept to be between 225 and 275 over a unit cube.

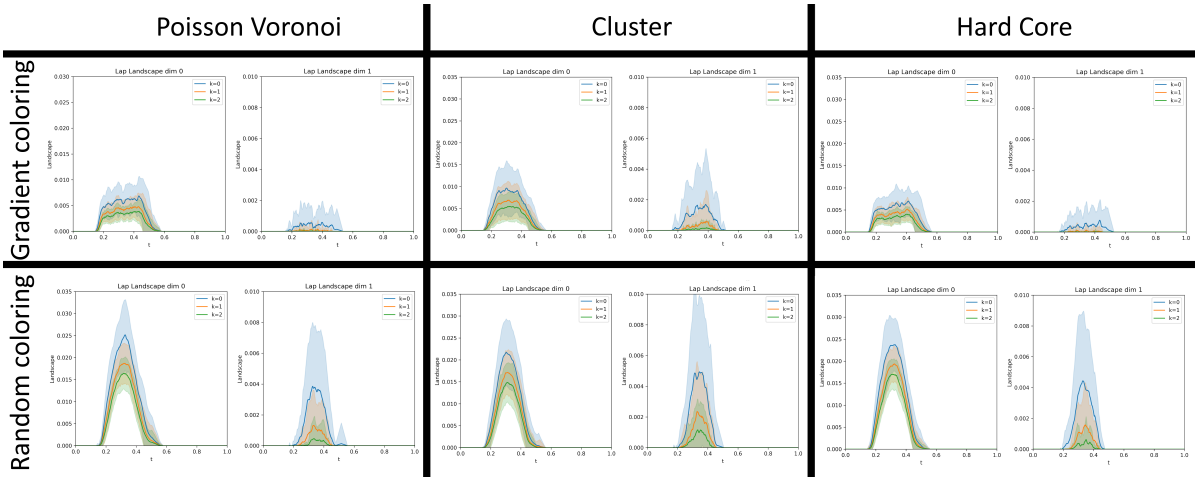


Figure 4.11: Laplacian Landscapes based on the trace of $[M_q^{s,t}]$ for the filtrations generated by different colorings of the edges of generated cross sections based on different generation methods. Mean landscape is taken over 100 samples per configuration, each is generated with a Ball Mapper parameter $\varepsilon = 15$.

effectively doing the same thing as taking the average x-value. This is done in the previously discussed figure. On the other hand, we could also randomly assign a weight to each edge based on a uniform distribution, see Figure 4.10. Both of these coloring methods are analyzed to see if there are any difference. Note furthermore that this gives an added benefit over the alpha complex method as now more information can be encoded.

To compare the differences between the gradient and random coloring, the mean Laplacian landscapes are computed and are shown in Figure 4.11. Here it can be seen that a clear difference exists as the gradient landscape seems to contain more noise, while the random coloring is more smooth. This is thought to be because the gradient landscape encodes more information about the actual structure, which can vary greatly from image to image. For example, in the cluster process, it would matter where the clusters are in the image. On the other hand, the random coloring more evenly distributes the edge weights through the image, therefore it is less dependent on where certain sections of close cells appear and it only encodes if they appear.

Finally, it can be seen that, fixing a color, the landscapes corresponding to different generation processes all look very similar, indicating that it is probably hard for a model to differentiate between them. Nevertheless, the same statistical tests as before can be made and analyzed for each different coloring method. Additionally, different values for the Ball Mapper parameter ε are used to compare the results. The results can be found in Tables 4.5 and 4.6, which correspond to gradient and random coloring of the edges respectively.

It is worth noting that the maximum obtained power of these tests is in most cases not as high as using the alpha complex method. Furthermore, note that the standard error of the estimation of the power is quite high as the amount of tested samples is relatively low. It can be computed by

$SE = \sqrt{\frac{p(1-p)}{100}}$, which is maximal for $p = 0.5$, returning a standard error of $SE = 0.05$.

However, still some results can be obtained. The models based on the trace of $[M_q^{s,t}]$ and $\tilde{M}_q^{s,t}$ seem to be very similar and are often within standard error. Therefore, the added inner product in the trace, found in Equation (3.33) seems to not affect the results in a significant way. This supports the usage of the adjusted multiplicity operator as even if not all eigenvalues are separately used, it still performs as well as $M_q^{s,t}$. However, both the persistent Laplacian based method seem to not significantly outperform standard persistent homology.

Furthermore, the tests created under the null hypothesis of the underlying generation process being of the cluster type, were mostly not able to reject PV or HC processes when the gradient coloring is used. This would support the hypothesis that the gradient coloring encodes the locations of the clusters. Therefore, landscapes corresponding to these generated cross sections could vary greatly from the mean landscape. In practice, this could be seen as an advantage, as it may be relevant to encode where certain high intensity KAM groups appear first and analyze the cells corresponding to these groups separately.

Finally, it seems to be difficult to set an optimal value of the Ball Mapper parameter ε. The Persistent Homology based tests, in general, seem to perform best on lower values of this parameter. However, the multiplicity based tests often achieve higher power in higher values. One possible reasoning could be that the geometric features contain more noise in lower values of ε as more simplices appear that can change these features. On the other hand PH only looks at topological features, which are less sensitive to the amount of simplices and instead perform better with additional information.

Table 4.5: Fraction of rejections for different statistical tests based on different null hypotheses and test statistics, coloring of the edges is done using a GRADIENT. For each test, the critical value is chosen with confidence level $\alpha = 0.95$ over 100 samples. Rejections are counted over 100 samples each. Type and q refer to the statistic used. Here PH refers to L_q^m, MULT to $L_q^{l,m}$ and ADJ to $L_q^{a,m}$. Furthermore, different values of Ball Mapper parameter ε are tested.

Null hypothesis			PV		cluster		HC	
Type	ε	q	cluster	HC	PV	HC	PV	cluster
PH	15	0	0.21	0.02	0.01	0.00	**0.15**	0.32
MULT	15	0	0.33	0.03	0.01	0.01	0.08	0.40
ADJ	15	0	0.25	0.02	**0.04**	**0.02**	0.11	0.42
PH	20	0	0.17	0.01	0.01	0.01	0.07	0.23
MULT	20	0	0.44	0.03	0.01	0.00	0.11	0.60
ADJ	20	0	0.43	0.03	0.03	0.00	**0.15**	0.60
PH	25	0	0.33	0.09	0.00	0.00	0.03	0.26
MULT	25	0	**0.57**	0.06	0.00	0.01	0.02	0.51
ADJ	25	0	0.53	0.04	0.00	**0.02**	0.07	**0.61**
PH	15	1	0.18	0.06	0.01	0.00	0.03	0.18
MULT	15	1	0.40	0.06	0.01	0.01	0.06	0.40
ADJ	15	1	0.40	0.06	0.01	0.01	0.04	0.39
PH	20	1	0.16	0.01	0.01	0.01	0.08	0.24
MULT	20	1	0.38	0.06	0.00	0.00	0.02	0.36
ADJ	20	1	0.38	0.06	0.00	0.00	0.02	0.36
PH	25	1	0.13	0.06	0.01	0.00	0.05	0.13
MULT	25	1	0.50	**0.11**	0.00	0.00	0.02	0.42
ADJ	25	1	0.50	**0.11**	0.00	0.00	0.02	0.42

Table 4.6: Fraction of rejections for different statistical tests based on different null hypotheses and test statistics, coloring of the edges is done RANDOMLY. For each test, the critical value is chosen with confidence level $\alpha = 0.95$ over 100 samples. Rejections are counted over 100 samples each. Type and q refer to the statistic used. Here PH refers to L_q^m, MULT to $L_q^{l,m}$ and ADJ to $L_q^{a,m}$. Furthermore, different values of Ball Mapper parameter ε are tested.

Null hypothesis			PV		cluster		HC	
Type	ε	q	cluster	HC	PV	HC	PV	cluster
PH	15	0	0.45	0.07	**0.30**	**0.31**	0.04	0.43
MULT	15	0	0.36	0.04	0.02	0.01	0.08	0.37
ADJ	15	0	0.36	0.03	0.01	0.04	0.07	0.45
PH	20	0	0.37	0.06	0.16	0.12	0.04	0.30
MULT	20	0	0.47	0.04	0.07	0.05	0.06	0.49
ADJ	20	0	**0.49**	0.04	0.11	0.11	0.09	0.56
PH	25	0	0.26	0.06	0.15	0.15	0.05	0.26
MULT	25	0	0.37	0.03	0.18	0.15	**0.16**	**0.59**
ADJ	25	0	0.42	0.04	0.24	0.20	0.12	0.56
PH	15	1	0.24	0.04	0.00	0.00	0.02	0.24
MULT	15	1	0.21	0.07	0.01	0.01	0.02	0.14
ADJ	15	1	0.21	0.07	0.01	0.01	0.02	0.14
PH	20	1	0.26	0.06	0.01	0.01	0.04	0.23
MULT	20	1	0.17	0.03	0.00	0.00	0.11	0.22
ADJ	20	1	0.17	0.03	0.00	0.00	0.11	0.22
PH	25	1	0.22	0.04	0.00	0.00	0.07	0.30
MULT	25	1	0.23	**0.08**	0.01	0.01	0.03	0.21
ADJ	25	1	0.23	**0.08**	0.01	0.01	0.04	0.21

5

Conclusion and Discussion

The field of Topological Data Analysis has seen a lot of attention in recent years. Especially Persistent Homology has been extensively studied and shown its capabilities in many applications. Nevertheless, its focus on only topological features, which are not affected by stretching, twisting or bending, may not be enough for every application. It was already shown that for handwritten number recognition, using the persistent Laplacian, which introduces geometric features, an improvement could be found [16].

However, using the persistent Laplacian is still an open problem requiring new methods to extract its information. Many papers look into making a feature vector from the operator [13, 16], nevertheless this requires some non-trivial choices being made. Because the field of Persistent Homology (PH) has seen a lot of attention, many different ways of using it have been proposed. Combining the two fields therefore shows a lot of promise as then the information of the persistent Laplacian can be combined with the ease of use of persistent homology.

For this, a slight change in calculation is proposed in terms of the multiplicity operator $M_q^{s,t}$, see Section 3.3.3. It has been shown that the trace of this operator has an interpretable meaning in terms of the features it captures. The operator describes geometry through the connections of $(q+1)$-simplices between q-cycles. This also gives an interpretation of the non-zero spectra of the persistent Laplacian operator as $M_q^{s,t}$ describes when it changes.

Nevertheless, the new operator is no longer positive semi-definite, therefore some of its eigenvalues can become negative. This was shown to happen at locations of start times s and end times t, where new features exist at (s,t), $(s-1,t)$ and $(s,t-1)$. At these locations the eigenvalues are not individually interpretable and only for the trace an interpretation could be found.

As a solution, the adjusted multiplicity operator $\tilde{M}_q^{s,t}$ is proposed, see Section 3.3.3. It has been shown that this operator is zero on the same locations as $M_q^{s,t}$, thereby also describing the locations where the persistent Laplacian changes. Furthermore, it was shown that the eigenvalues of this operator could not be negative and the amount of non-zero eigenvalues could not be higher than the amount of simplices appearing at a certain time. This means that every non-zero eigenvalue corresponds to at least one changing simplex. Finally, the trace of $\tilde{M}_q^{s,t}$ was analyzed and shown to be slightly adjusted from the trace of $[M_q^{s,t}]$, however the consequences could not easily be described.

Using the standard multiplicity operator $M_q^{s,t}$ on the MNIST dataset, containing images of handwritten numbers, outperformed classical persistent homology and could closely match the performance of the persistent Laplacian, see Section 4.1. It also showed the ease of use of the operator as a PH based method, persistent landscapes, could easily be altered to also work on the trace of the new operator. Nevertheless, the tests showed that adding the topological information of PH or the information of the combinatorial Laplacian to the multiplicity operator could yield even better results. This indicates that some topological information as well as the information from the combinatorial Laplacian is still missing in the operator.

Finally in Section 4.2 the operators were applied to images of artificially generated cross sections of crystalline structures often found in alloys. For the alpha filtration in Section 4.2.1, both of the new operators seemed to not provide any additional information and would decrease the accuracy due to the noise in dimension 0. Similarly using Ball Mapper to create a graph and analyzing this graph did not provide an improvement over using standard persistent homology. However, from this data it did

become clear that using the trace of the adjusted multiplicity operator $\tilde{M}_q^{s,t}$ is not significantly different from using the trace of $[M_q^{s,t}]$, therefore the adjusted operator can be used even in situations where the eigenvalues are not interpreted separately.

To conclude, it is worth noting that any method based on the persistent Laplacian brings an additional computational cost. Therefore for the usage of the multiplicity operator to make sense in practice, it has to outperform persistent homology based methods. In this thesis it is shown that this can be the case, however it requires choosing a filtration that contains geometric features. If this can be found, the additional time complexity can make sense if not much data is available or if more information about the existing data needs to be extracted for a specific task.

Future research

Because it was shown that adding information of the combinatorial Laplacian to the multiplicity operator can yield to better performing models, it could be useful to encode this information directly into the operator. In Section 3.3.2 it was noted that the down-Laplacian is not used because it is not needed for the multiplicity equation. However, for locations (s,t), where $s = t$, $(s,t-1)$ does not exist and therefore the down-Laplacian would not disappear. Instead in the multiplicity equation the down-Laplacian $\Delta_{q,-}^s$ would be left. This would give a natural way of including the information, however now there are some points on the diagonal. Standard PH methods such as the persistence landscapes would not be able to use these points as all points in a standard persistence diagram are above the diagonal. To circumvent this, all points could be moved up one spot to represent intervals where the persistent Laplacian is constant.

Using TDA on the MNIST dataset has been attempted often in literature, however often *cubical complexes* are used. Simplices in simplicial complexes all consist of triangles, whereas in cubical complexes they would consist of cubes. It has been shown that this works well for describing the digits of the MNIST dataset as pixels are more easily described by cubes than by triangles [21]. Therefore, future research could attempt to use these complexes instead. Furthermore, the same cubical complexes could also be used on the analysis of crystalline structures as the process is very similar.

Some results exist on the stability of the eigenvalues of the persistent Laplacian [33], however they are not as strong as the results for regular persistent homology. It would therefore be interesting to see if anything can be said about the stability of the multiplicity operator and its eigenvalues. This might give a new interpretation of the meaning of the eigenvalues and could be a strong argument for using the method.

Finally it is noted that the distance used between landscapes d_{land}, introduced in Section 2.4.3, sums over all dimensions, but landscapes of different dimensions could have greatly varying sizes. Therefore more emphasis is put on dimensions with large landscapes, while these may not always contain the most important features, as can be seen in Section 4.2.1. Therefore, the distance of each dimension should probably first be normalized before summing.

Bibliography

[1] Mehmet E. Aktas, Esra Akbas, and Ahmed El Fatmaoui. “Persistence homology of networks: methods and applications”. In: *Applied Network Science* 4.1 (Aug. 2019), p. 61. ISSN: 2364-8228. DOI: 10.1007/s41109-019-0179-3. URL: https://doi.org/10.1007/s41109-019-0179-3.

[2] E. Alpaydin and C. Kaynak. *Optical Recognition of Handwritten Digits*. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C50P49. 1998.

[3] D. Vijay Anand and Moo K. Chung. “Hodge Laplacian of Brain Networks”. In: *IEEE Transactions on Medical Imaging* 42.5 (May 2023), pp. 1563–1573. ISSN: 1558-254X. DOI: 10.1109/TMI.2022.3233876. URL: https://ieeexplore.ieee.org/document/10005115.

[4] D. Vijay Anand et al. “Weighted persistent homology for osmolyte molecular aggregation and hydrogen-bonding network analysis”. en. In: *Scientific Reports* 10.1 (June 2020). Publisher: Nature Publishing Group, p. 9685. ISSN: 2045-2322. DOI: 10.1038/s41598-020-66710-6. URL: https://www.nature.com/articles/s41598-020-66710-6.

[5] Ulrich Bauer and Michael Lesnick. “Induced Matchings of Barcodes and the Algebraic Stability of Persistence”. In: *Proceedings of the Thirtieth Annual Symposium on Computational Geometry*. SOCG’14. Kyoto, Japan: Association for Computing Machinery, 2014, pp. 355–364. ISBN: 9781450325943. DOI: 10.1145/2582112.2582168. URL: https://doi.org/10.1145/2582112.2582168.

[6] Michael Bleher et al. “Topology identifies emerging adaptive mutations in SARS-CoV-2”. In: *arXiv:2106.07292 [cs, q-bio]* (June 2021). arXiv: 2106.07292. URL: http://arxiv.org/abs/2106.07292.

[7] Peter Bubenik. *Statistical Topological Data Analysis using Persistence Landscapes*. 2015. URL: http://jmlr.org/papers/v16/bubenik15a.html.

[8] Anuraag Bukkuri, Noemi Andor, and Isabel K. Darcy. “Applications of Topological Data Analysis in Oncology”. English. In: *Frontiers in Artificial Intelligence* 4 (Apr. 2021). Publisher: Frontiers. ISSN: 2624-8212. DOI: 10.3389/frai.2021.659037. URL: https://www.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2021.659037/full.

[9] David Carlson, Emile Haynsworth, and Thomas Markham. “A Generalization of the Schur Complement by Means of the Moore-Penrose Inverse”. In: *SIAM Journal on Applied Mathematics* 26.1 (1974). Publisher: Society for Industrial and Applied Mathematics, pp. 169–175. ISSN: 0036-1399. URL: https://www.jstor.org/stable/2099662.

[10] Gunnar Carlsson and Vin de Silva. “Zigzag Persistence”. In: *Foundations of Computational Mathematics* 10.4 (Aug. 2010), pp. 367–405. ISSN: 1615-3383. DOI: 10.1007/s10208-010-9066-0. URL: https://doi.org/10.1007/s10208-010-9066-0.

[11] Gunnar Carlsson et al. “Persistence barcodes for shapes”. In: *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*. SGP ’04. Nice, France: Association for Computing Machinery, 2004, pp. 124–135. ISBN: 3905673134. DOI: 10.1145/1057432.1057449. URL: https://doi.org/10.1145/1057432.1057449.

[12] Frédéric Chazal et al. “Proximity of persistence modules and their diagrams”. In: *Proceedings of the twenty-fifth annual symposium on Computational geometry*. SCG ’09. New York, NY, USA: Association for Computing Machinery, June 2009, pp. 237–246. ISBN: 978-1-60558-501-7. DOI: 10.1145/1542362.1542407. URL: https://doi.org/10.1145/1542362.1542407.

[13] Jiahui Chen et al. “Persistent Laplacian projected Omicron BA.4 and BA.5 to become new dominating variants”. In: *Computers in Biology and Medicine* 151 (2022), p. 106262. ISSN: 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2022.106262.

[14] Randall E. Cline. “Representations for the Generalized Inverse of a Partitioned Matrix”. In: *Journal of the Society for Industrial and Applied Mathematics* 12.3 (1964). Publisher: Society for Industrial and Applied Mathematics, pp. 588–600. ISSN: 0368-4245. URL: https://www.jstor.org/stable/2946332.

[15] Asir Antony Gnana Singh Danasingh, Appavu alias Balamurugan Subramanian, and Jebamalar Leavline Epiphany. “Identifying redundant features using unsupervised learning for high-dimensional data”. en. In: *SN Applied Sciences* 2.8 (July 2020), p. 1367. ISSN: 2523-3971. DOI: 10.1007/s42452-020-3157-6. URL: https://doi.org/10.1007/s42452-020-3157-6.

[16] Thomas Davies, Zhengchao Wan, and Ruben J. Sanchez-Garcia. “The Persistent Laplacian for Data Science: Evaluating Higher-Order Persistent Spectral Representations of Data”. en. In: *Proceedings of the 40th International Conference on Machine Learning*. ISSN: 2640-3498. PMLR, July 2023, pp. 7249–7263. URL: https://proceedings.mlr.press/v202/davies23c.html.

[17] Paweł Dłotko. *Ball mapper: a shape summary for topological data analysis*. arXiv:1901.07410 [math]. Jan. 2019. DOI: 10.48550/arXiv.1901.07410. URL: http://arxiv.org/abs/1901.07410.

[18] Beno Eckmann. “Harmonische Funktionen und Randwertaufgaben in einem Komplex”. de. In: *Commentarii Mathematici Helvetici* 17.1 (Dec. 1944), pp. 240–255. ISSN: 1420-8946. DOI: 10.1007/BF02566245. URL: https://doi.org/10.1007/BF02566245.

[19] Herbert Edelsbrunner. “Herbert Edelsbrunner”. en. In: *Wiadomości Matematyczne* 48.2 (June 2012), p. 47. ISSN: 2543-991X, 2080-5519. DOI: 10.14708/wm.v48i2.316. URL: http://wydawnictwa.ptm.org.pl/index.php/wiadomosci-matematyczne/article/view/316.

[20] Florian Frantzen, Jean-Baptiste Seby, and Michael T. Schaub. “Outlier Detection for Trajectories via Flow-embeddings”. In: *2021 55th Asilomar Conference on Signals, Systems, and Computers*. ISSN: 2576-2303. Oct. 2021, pp. 1568–1572. DOI: 10.1109/IEEECONF53345.2021.9723128. URL: https://ieeexplore.ieee.org/document/9723128.

[21] Adélie Garin and Guillaume Tauzin. *A Topological ”Reading” Lesson: Classification of MNIST using TDA*. 2019. DOI: 10.1109/ICMLA.2019.00256.

[22] Shafie Gholizadeh et al. “Topological Data Analysis in Text Classification: Extracting Features with Additive Information”. In: *arXiv:2003.13138 [cs, math, stat]* (Mar. 2020). arXiv: 2003.13138. URL: http://arxiv.org/abs/2003.13138.

[23] Barbara Giunti and Janis Lazovskis. *TDA-Applications (an online database of papers on applications of TDA outside math)*. 2025. URL: https://www.zotero.org/groups/2425412/tda-applications (visited on 06/13/2025).

[24] Timothy E Goldberg. “Combinatorial Laplacians of Simplicial Complexes”. en. PhD thesis. Bard College, 2002. URL: https://pi.math.cornell.edu/~goldberg/Papers/CombinatorialLaplaci
pdf.

[25] G. Golub and W. Kahan. “Calculating the Singular Values and Pseudo-Inverse of a Matrix”. In: *Journal of the Society for Industrial and Applied Mathematics Series B Numerical Analysis* 2.2 (Jan. 1965). Publisher: Society for Industrial and Applied Mathematics, pp. 205–224. ISSN: 0887-459X. DOI: 10.1137/0702016. URL: https://epubs.siam.org/doi/abs/10.1137/0702016.

[26] Ching-Hsiang Hung and Thomas L. Markham. “The Moore-Penrose inverse of a partitioned matrix $M=\begin{pmatrix} A & 0 \\ B & C \end{pmatrix}$”. eng. In: *Czechoslovak Mathematical Journal* 25.3 (1975). Publisher: Institute of Mathematics, Academy of Sciences of the Czech Republic, pp. 354–361. ISSN: 0011-4642 (print). URL: https://dml.cz/handle/10338.dmlcz/101330.

[27] Takashi Ichinomiya, Ippei Obayashi, and Yasuaki Hiraoka. “Protein-folding analysis using features obtained by persistent homology”. In: *Biophysical Journal* 118.12 (2020), pp. 2926–2937.

[28] Sohail Iqbal et al. “Classification of COVID-19 via Homology of CT-SCAN”. In: *arXiv:2102.10593 [cs, eess, math]* (Feb. 2021). arXiv: 2102.10593. URL: http://arxiv.org/abs/2102.10593.

[29] Chi-Kwong Li and Fuzhen Zhang. “Eigenvalue continuity and Geršgorin’s theorem”. In: *Electronic Journal of Linear Algebra* 35.1 (Dec. 2019). arXiv:1912.05001 [math], pp. 619–625. ISSN: 1081-3810. DOI: 10.13001/1081-3810.4123. URL: http://arxiv.org/abs/1912.05001 (visited on 08/12/2025).

[30] Zechao Li et al. “Unsupervised Feature Selection Using Nonnegative Spectral Analysis”. en. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 26.1 (Sept. 2021), pp. 1026–1032. ISSN: 2374-3468, 2159-5399. DOI: 10.1609/aaai.v26i1.8289. URL: https://ojs.aaai.org/index.php/AAAI/article/view/8289.

[31] Huawen Liu, Xindong Wu, and Shichao Zhang. “Feature selection using hierarchical feature clustering”. In: *Proceedings of the 20th ACM international conference on Information and knowledge management*. CIKM ’11. New York, NY, USA: Association for Computing Machinery, Oct. 2011, pp. 979–984. ISBN: 978-1-4503-0717-8. DOI: 10.1145/2063576.2063716. URL: https://doi.org/10.1145/2063576.2063716.

[32] Vine Nwabuisi Madukpe, Bright Chukwuma Ugoala, and Nur Fariha Syaqina Zulkepli. *A Comprehensive Review of the Mapper Algorithm, a Topological Data Analysis Technique, and Its Applications Across Various Fields (2007-2025)*. arXiv:2504.09042 [math]. Apr. 2025. DOI: 10.48550/arXiv.2504.09042. URL: http://arxiv.org/abs/2504.09042.

[33] Facundo Mémoli, Zhengchao Wan, and Yusu Wang. “Persistent Laplacians: Properties, Algorithms and Implications”. In: *SIAM Journal on Mathematics of Data Science* 4.2 (2022), pp. 858–884. DOI: 10.1137/21M1435471. eprint: https://doi.org/10.1137/21M1435471. URL: https://doi.org/10.1137/21M1435471.

[34] Zhenyu Meng and Kelin Xia. “Persistent spectral–based machine learning (PerSpect ML) for protein-ligand binding affinity prediction”. In: *Science Advances* 7.19 (May 2021). Publisher: American Association for the Advancement of Science, eabc5329. DOI: 10.1126/sciadv.abc5329. URL: https://www.science.org/doi/10.1126/sciadv.abc5329.

[35] Elizabeth Munch. “A User’s Guide to Topological Data Analysis”. en. In: *Journal of Learning Analytics* 4.2 (July 2017). Number: 2, pp. 47–61. ISSN: 1929-7750. DOI: 10.18608/jla.2017.42.6. URL: https://learning-analytics.info/index.php/JLA/article/view/5196.

[36] R. Piziak, P.L. Odell, and R. Hahn. “Constructing projections on sums and intersections”. en. In: *Computers & Mathematics with Applications* 37.1 (Jan. 1999), pp. 67–74. ISSN: 08981221. DOI: 10.1016/S0898-1221(98)00242-9. URL: https://linkinghub.elsevier.com/retrieve/pii/S0898122198002429.

[37] Pratyush Pranav et al. “Unexpected topology of the temperature fluctuations in the cosmic microwave background”. In: *Astronomy & Astrophysics* (July 2019). URL: https://www.aanda.org/articles/aa/full_html/2019/07/aa34916-18/aa34916-18.html.

[38] Ysanne Pritchard et al. “Persistent homology analysis distinguishes pathological bone microstructure in non-linear microscopy images”. en. In: *Scientific Reports* 13.1 (Feb. 2023). Publisher: Nature Publishing Group, p. 2522. ISSN: 2045-2322. DOI: 10.1038/s41598-023-28985-3. URL: https://www.nature.com/articles/s41598-023-28985-3.

[39] Yuchi Qiu and Guo-Wei Wei. “Persistent spectral theory-guided protein engineering”. In: *Nature computational science* 3.2 (Feb. 2023), pp. 149–163. ISSN: 2662-8457. DOI: 10.1038/s43588-022-00394-y. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10456983/.

[40] Shiquan Ren, Chengyuan Wu, and Jie Wu. “Weighted Persistent Homology”. In: *The Rocky Mountain Journal of Mathematics* 48.8 (2018), pp. 2661–2687. ISSN: 00357596, 19453795. URL: https://www.jstor.org/stable/26579729.

[41] M. Saadatfar et al. “Pore configuration landscape of granular crystallization”. en. In: *Nature Communications* 8.1 (May 2017). Publisher: Nature Publishing Group, p. 15082. ISSN: 2041-1723. DOI: 10.1038/ncomms15082. URL: https://www.nature.com/articles/ncomms15082.

[42] Gurjeet Singh, Facundo Memoli, and Gunnar Carlsson. *Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object Recognition*. en. ISSN: 1811-7813. The Eurographics Association, 2007. ISBN: 978-3-905673-51-7. URL: https://doi.org/10.2312/SPBG/SPBG07/091-100.

[43] Martina Vittorietti et al. “General framework for testing Poisson-Voronoi assumption for real microstructures”. In: *Applied Stochastic Models in Business and Industry* 36.4 (2020), pp. 604–627. DOI: https://doi.org/10.1002/asmb.2517. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/asmb.2517. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/asmb.2517.

[44] Rui Wang, Duc Duy Nguyen, and Guo-Wei Wei. “Persistent spectral graph”. In: *International Journal for Numerical Methods in Biomedical Engineering* 36.9 (2020), e3376. DOI: https://doi.org/10.1002/cnm.3376. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cnm.3376. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/cnm.3376.

[45] JunJie Wee and Kelin Xia. “Persistent spectral based ensemble learning (PerSpect-EL) for protein–protein binding affinity prediction”. In: *Briefings in Bioinformatics* 23.2 (Mar. 2022), bbac024. ISSN: 1477-4054. DOI: 10.1093/bib/bbac024. URL: https://doi.org/10.1093/bib/bbac024.

[46] Xiaoqi Wei and Guo-Wei Wei. *Persistent Topological Laplacians – a Survey*. arXiv:2312.07563 [math]. Dec. 2024. DOI: 10.48550/arXiv.2312.07563. URL: http://arxiv.org/abs/2312.07563.

[47] Kelin Xia and Guo-Wei Wei. “Persistent homology analysis of protein structure, flexibility and folding”. In: *International journal for numerical methods in biomedical engineering* 30.8 (Aug. 2014), pp. 814–844. ISSN: 2040-7939. DOI: 10.1002/cnm.2655. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4131872/.

[48] Jingjie Yang et al. *Topological classification of tumour-immune interactions and dynamics*. arXiv:2308.05294 [q-bio]. Aug. 2023. DOI: 10.48550/arXiv.2308.05294. URL: http://arxiv.org/abs/2308.05294.

[49] Afra Zomorodian and Gunnar Carlsson. “Computing Persistent Homology”. en. In: *Discrete & Computational Geometry* 33.2 (Feb. 2005), pp. 249–274. ISSN: 1432-0444. DOI: 10.1007/s00454-004-1146-y. URL: https://doi.org/10.1007/s00454-004-1146-y.