Applications of Topology to Data Analysis

A Thesis

submitted to
Indian Institute of Science Education and Research Pune
in partial fulfillment of the requirements for the
BS-MS Dual Degree Programme

by

Shambhavi S.

Indian Institute of Science Education and Research Pune
Dr. Homi Bhabha Road,
Pashan, Pune 411008, INDIA.

April, 2021

Supervisor: Dr. Priyavrat Deshpande

Certificate

This is to certify that this dissertation entitled Applications of Topology to Data Analysis towards the partial fulfilment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research, Pune represents study/work carried out by Shambhavi S. at Indian Institute of Science Education and Research under the supervision of Dr. Priyavrat Deshpande, Assistant Professor, Chennai Mathematical Institute, Department of Mathematics , during the academic year 2020-2021.

Dr. Priyavrat Deshpande

Committee:

Dr. Priyavrat Deshpande

Dr. Anindya Goswami

This thesis is dedicated to my family

Declaration

I hereby declare that the matter embodied in the report entitled Applications of Topology to Data Analysis are the results of the work carried out by me at the Department of Mathematics, Indian Institute of Science Education and Research, Pune, under the supervision of Dr. Priyavrat Deshpande and the same has not been submitted elsewhere for any other degree.

Shambhavi S

Shambhavi S.

Acknowledgments

I wish to express my deepest gratitude to my supervisor, Dr. Priyavrat Deshpande, for all his support and encouragement through the course of this project. His inputs and feedback were indispensable in shaping this thesis. I thank my expert member, Dr. Anindya Goswami, for his helpful comments and suggestions. I want to take this opportunity to thank all my Professors at IISER for helping me grow as a student. I am very grateful to Dr. Rama Mishra for all the support and guidance she has extended to me. I also wish to thank my friends for always being there for me. Finally, I would like to thank my ever-supportive family for putting up with me this past year and for always nudging me in the right direction.

Abstract

This thesis aims to serve as an introduction to Topological Data Analysis (TDA), a collection of methods that seek to quantify the topological and geometric features of data using algebraic topology. The theory behind persistent homology, a stable multi-scale approach for characterizing the structure of data, is presented here. Further, an algorithm to compute persistence diagrams, a standard representation of persistent homology, is also discussed. An overview of some stable vectorized representations of persistent homology that are better suited for statistical and machine learning tasks is also given. The remainder of the thesis addresses how these techniques can help analyze images and financial time series data. Subsequently, a topological pipeline for image classification is put forth. Application of TDA to biological images and financial time series data is also presented to motivate the broad scope of these techniques.

Contents

Introduction

With the data produced becoming increasingly complex, high-dimensional, and noisy, it often poses many new challenges. Topological data analysis (TDA) proves effective in overcoming some of these hurdles by providing a framework to study the 'shape of data' using tools from algebraic topology.

The structure of the data often dictates the effectiveness of a particular mode of analysis. For example, linear methods fail when the data has a non-linear structure. TDA provides us qualitative insight into the organization of the data at a global level. It also enables us to analyse data in a manner independent of the choice of embedding and coordinates.

TDA provides a variety of tools that extract complementary information from data that can be used in conjunction with standard analytical and statistical techniques. For instance, combining this topological information with neural networks has offered promising results for image analysis [22, 16].

This thesis provides an exposition of some of the significant themes in TDA supported by some applications to biological and financial data. A topological pipeline for image analysis and classification is presented, and a general framework for analyzing time series data using TDA is also discussed. The thesis is organized as follows:

Chapter 1 establishes some definitions and results from homology theory required for the thesis. The computation of simplicial homology modules is also discussed here. Following this, Chapter 2 looks at the reconstruction theorem [8] for point cloud data, which provides mathematical backing for inferring topological properties from structures built on the data.

The focus of Chapter 3 is persistent homology, which describes how the topological features of the data evolve with respect to a changing parameter. We look at its description

as a graded module and also study an algorithm for its computation as put forth in [3]. The barcode and diagram representation of this module is described, and some results on their stability are also discussed.

The structure of persistence diagrams proves insufficient for further statistical analysis or for integration with machine learning algorithms. To aid these tasks, topological summaries that map persistence diagrams into a vector space are used. An overview of a few such commonly used summaries - persistent entropy [12], persistence images [14] and persistence landscapes [11] is presented in Chapter 4.

Chapter 5 focuses on the application of TDA to the problem of image classification. A pipeline that extracts topological features from images is constructed using the giotto-tda library in Python [25]. The stability of the pipeline features under rotation and transformation of the images is analysed, and a method to extract features that are more robust to such transformations is also presented. The topological pipeline was also used to analyse a few image datasets. The results of this have also been discussed in this chapter.

In the final chapter, we look at the application of TDA to time series data. TDA has proven successful in identifying periods of critical transitions in climate and financial data [9, 17]. Some prerequisite theory - including Taken's theorem and sliding window embeddings, as outlined in [21] are first discussed. These ideas are then applied to financial market data with the aim of identifying market crashes.

Original Contributions

While most steps in the pipeline presented in Chapter 5 were implemented using predefined functions from the giotto-tda library [25], the 'line filtration' function, a generalisation of the height filtration function available in the library, was separately defined and used.

The classification of Fundus [10], Flower [6] and Fashion MNIST [18] images using the pipeline features presented in Chapter 5, and Section 5.5 which outlines a method for extracting features that are robust under some transformations of the images represent original work done in the thesis.

The same is true of the analysis of financial time series data presented in Chapter 6.

Chapter 1

Preliminaries

This chapter seeks to establish some definitions and results required for the thesis. Sections 1.1 and 1.2 discuss simplicial and singular homology theory based on [1] and [2]. The contents of this chapter also include the homotopy invariance property of homology and the computation of simplicial homology modules using linear algebra.

1.1 Simplicial Homology

A set of points $\{a_0, a_1, \ldots, a_n\}$ in $\mathbb{R}^N$ is said to be *geometrically independent* iff the vectors $a_1 - a_0, \ldots, a_n - a_0$ are linearly independent.

Definition 1.1.1 (n-simplex)**.** The *n-simplex* σ spanned by a geometrically independent set of points $\{a_0, a_1, \ldots, a_n\}$ from $\mathbb{R}^N$, is the set of all points

$$\sigma = \{\sum_{i=0}^{n} t_i a_i \mid \sum_{i=0}^{n} t_i = 1 \ \& \ t_i \geq 0 \ \forall \ i \in \{0, \ldots, n\}\}.$$

The set of points $\{a_0, a_1, \ldots, a_n\}$ are called the vertices of σ and n is the *dimension* of the n-simplex. The simplex spanned by a subset of the vertices is called a *face* of σ and the union of all faces of σ is called the *boundary* of σ. The standard n-simplex Δ^n, is the simplex spanned by the standard basis vectors in $\{e_i\}_{i=1}^{n+1}$ in $\mathbb{R}^{n+1}$.

Definition 1.1.2 (Orientation)**.** An *orientation* of an n-simplex is the equivalence class of an ordering of the vertex set under the equivalence relation that identifies orderings that differ by an even permutation.

An oriented n-simplex spanned by the ordered set of vertices $v_0, v_1, \ldots, v_n$ is denoted by $[v_0, v_1, \ldots v_n]$

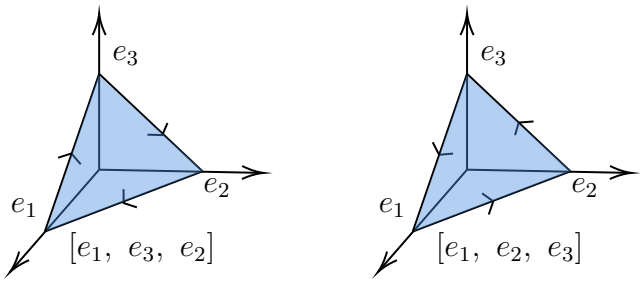


Figure 1.1: The standard Δ^2 simplex with both possible orientations.

Definition 1.1.3 (Geometric Simplicial Complex)**.** A *geometric simplicial complex* K in $\mathbb{R}^N$ is a collection of simplices in $\mathbb{R}^N$ such that,

1. every face of a simplex in K also belongs to K, and

2. the intersection of two distinct simplices in K is a face of both them.

Definition 1.1.4 (Abstract Simplicial Complex)**.** A collection $\mathcal{S}$, of finite non-empty sets is said to be an abstract simplicial complex if every non-empty subset of an element of $\mathcal{S}$ also belongs to $\mathcal{S}$. In this case, every element A of $\mathcal{S}$ is called a *simplex* of dimension $|A| - 1$.

The *vertex set* V, of an abstract simplicial complex $\mathcal{S}$ is the union of all singletons in $\mathcal{S}$. Given a simplicial complex K, the collection of vertices of all the constituent simplices forms an abstract simplicial complex called the *vertex scheme* of K.

A bijective map between the vertex sets of two abstract simplicial complexes is an *isomorphism* if it maps every simplex in one complex to a simplex in the other. Every abstract simplicial complex can be shown to be isomorphic to the vertex scheme of some simplicial complex which is called its *geometric realisation*. As a result, every abstract simplicial complex can be associated with a topological space determined by this geometric realisation.

A related concept is that of a *triangulation*. A geometric simplicial complex K is said to be a triangulation of a topological space X, if there exists a homeomorphism $\gamma : K \to X$. A space that accepts a triangulation is said to be triangulable.

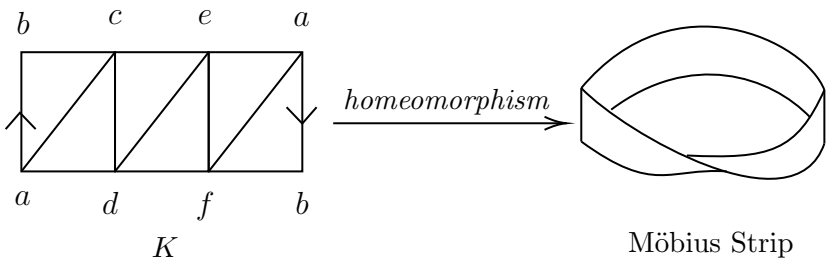


Figure 1.2

Example. The geometric realization K, of the abstract simplicial complex with the vertex set $\{\{a\},\{b\},\{c\},\{d\},\{e\},\{f\}\}$ whose simplices are given by $\{a,b,c\}$, $\{c,a,d\}$, $\{c,e,d\}$, $\{e,f,d\}$, $\{a,e,f\}$, $\{a,b,f\}$ and their non-empty subsets is depicted in Figure 1.2. K is homeomorphic to a Möbious strip and thus defines a triangulation for this space.

Definition 1.1.5 (p-chains)**.** Given a simplicial complex K, the free abelian group generated by all the oriented p-simplices in K is called the group of p-chains in K and is denoted by $C_p(K)$.

Definition 1.1.6 (Boundary Map)**.** The p^{th} boundary map is the linear homomorphism $\partial_p : C_p(K) \longrightarrow C_{p-1}(K)$ that maps each oriented p-simplex $[v_0, v_1, \dots v_p]$ as follows,

$$\partial_p \ : \ [v_0, v_1, \dots, v_p] \mapsto \sum_{i=0}^{n} (-1)^i [v_0, \dots, \hat{v_i}, \dots, v_p].$$

It can easily be shown that $\partial_p \circ \partial_{p+1} = 0$. As a result, one can associate the following *chain map* with every simplicial complex K.

$$\dots \longrightarrow C_{p+1}(K) \xrightarrow{\partial_{p+1}} C_p(K) \xrightarrow{\partial_p} C_{p-1}(K) \longrightarrow \dots C_0 \longrightarrow 0.$$

The kernel of the map ∂_p is called the group of p-cylces and is denoted by $Z_p(K)$ while the image of ∂_{p+1} is called the group of p-boundaries and is denoted by $B_p(K)$. As $\partial_p \circ \partial_{p+1} = 0$,

$$Im(\partial_{p+1}) = B_p(K) \subseteq Ker(\partial_p) = Z_p(K) \subseteq C_p(k).$$

Definition 1.1.7 (Homology)**.** The p^{th} homology group of a simplicial complex K, denoted by $H_p(K)$ is given by

$$H_p(K) \ = \ \frac{Z_p(K)}{B_p(K)}.$$

The rank of $H_p(K)$ is called the p^{th} Betti number of K and is denoted by $B_p(K)$.

1.2 Singular Homology

Definition 1.2.1 (n-singular simplex)**.** Given a space X, a singular n-simplex of X is a continuous map $\sigma \ : \ \Delta^n \longrightarrow X$.

One can then define singular homology groups of X by constructing chain groups and boundary maps in a similar manner as outlined previously using singular n-simplices.

Remark 1.2.1. Similar to how simplicial homology was built by considering simplices as building blocks, a cubical homology theory can also be developed by using n-cubes given by $I^n = \underbrace{[0,1] \times \ldots \times [0,1]}_{n- \text{ times}}$.

Remark 1.2.2. While chain groups for simplicial and singular homology groups were defined using $\mathbb{Z}$-linear combinations, the same can be done with any base ring R. Using this, one can define homology modules corresponding to any choice of a ground ring.

1.3 Homotopy Invariance

Let X and Y be topological spaces and $f \ : \ X \longrightarrow Y$ be a continuous map. For a given singular n-simplex σ of X, $f \circ \sigma$ is a singular n-simplex of Y. Thus, for each n we can define a map $f_{\#} : C_n(X) \longrightarrow C_n(Y)$ by linearly extending the map that assigns $\sigma \mapsto f \circ \sigma$.

Proposition 1.3.1. The maps $f_{\#} : C_n(X) \longrightarrow C_n(Y)$, define a chain map $f_{\#}$ between the singular chain complexes of X and Y.

$$\begin{array}{ccccccccc} \ldots & \longrightarrow & C_{n+1}(X) & \xrightarrow{\partial_{n+1}} & C_n(X) & \xrightarrow{\partial_n} & C_{n-1}(X) & \longrightarrow & \ldots \\ & & \downarrow f_\# & & \downarrow f_\# & & \downarrow f_\# & & \\ \ldots & \longrightarrow & C_{n+1}(Y) & \xrightarrow{\partial_{n+1}} & C_n(Y) & \xrightarrow{\partial_n} & C_{n-1}(Y) & \longrightarrow & \ldots \end{array}$$

Proof. As the maps $f_\# : C_n(X) \longrightarrow C_n(Y)$ are linear homomorphisms by construction, all that is left to be shown is that each square in the diagram commutes. Given an n-simplex σ in X,

$$f_\# \circ \partial_n(\sigma) = f_\#(\sum_{i=0}^{n}(-1)^i \sigma\big|_{\left[e_1,\ldots,\hat{e_{i+1}}\ldots e_{n+1}\right]}) = \sum_{i=0}^{n}(-1)^i f(\sigma\big|_{\left[e_1,\ldots,\hat{e_{i+1}}\ldots e_{n+1}\right]})$$

$$= \sum_{i=0}^{n}(-1)^i (f\sigma)\big|_{\left[e_1,\ldots,\hat{e_{i+1}}\ldots e_{n+1}\right]} = \partial_n \circ f_\#(\sigma).$$

The proposition thus holds as a result of linearity of ∂_n and $f_\#$. □

Proposition 1.3.1 implies that $f_\#(Z_n(X)) \subseteq Z_n(Y)$ and $f_\#(B_n(X)) \subseteq B_n(Y)$. Hence, the chain map $f_\#$ induces homomorphisms f_* at the homology level.

Proposition 1.3.2. Given topological spaces $X,\ Y$ & Z and maps f, g such that $X \xrightarrow{f} Y \xrightarrow{g} Z$, the following hold:

$$(g \circ f)_\# = g_\# \circ f_\#,$$

$$(g \circ f)_* = g_* \circ f_*.$$

Theorem 1.3.1. Homotopic maps $f, g : X \longrightarrow Y$ induce the same homomorphism at the homology level, $f_* = g_* : H_n(X) \longrightarrow H_n(Y)$.

$$\begin{array}{ccccccccc} \ldots & \xrightarrow{\partial} & C_{n+1}(X) & \xrightarrow{\partial} & C_n(X) & \xrightarrow{\partial} & C_{n-1}(X) & \xrightarrow{\partial} & \ldots \\ & & & \swarrow^{P} & \downarrow{\scriptstyle f_\#}_{\scriptstyle g_\#} & \swarrow^{P} & & & \\ \ldots & \xrightarrow{\partial} & C_{n+1}(Y) & \xrightarrow{\partial} & C_n(Y) & \xrightarrow{\partial} & C_{n-1}(Y) & \xrightarrow{\partial} & \ldots \end{array}$$

Proof. Given the space $\Delta^n \times I$, let $\Delta^n \times \{0\} = [e_0, e_1, \ldots e_n]$ and $\Delta^n \times \{1\} = [v_0, v_1 \ldots v_n]$ such that e_i maps to v_i under the projection $\Delta^n \times I \to \Delta^n$. The space $\Delta^n \times I$ can then be expressed as a union of $(n+1)-$simplices $\{[e_0, e_1, \ldots e_i, v_i, \ldots v_n]\}_{i=0}^{n}$.

Let $F : X \times I \longrightarrow Y$ represent the homotopy from f to g. Given any simplex σ in $C_n(X)$, we have the map

$$F \circ (\sigma \times \mathbb{1}) : \Delta^n \times I \longrightarrow Y.$$

We can thus define a family of maps $P : C_n(X) \longrightarrow C_{n+1}(Y)$ such that for any singular n-simplex σ in X,

$$P(\sigma) = \sum_{i=0}^{n} (-1)^i (F(\sigma \times \mathbb{1}))\Big|_{[e_0, e_1, \ldots e_i, \tilde{e_i}, \ldots \tilde{e_n}]}.$$

Consider the maps ∂P and $P\partial : C_n(X) \longrightarrow C_n(Y)$,

$$\begin{aligned} P\partial(\sigma) = &\sum_{j<i} (-1)^{i+j} (F(\sigma \times \mathbb{1}))\Big|_{[e_0, \ldots e_j, v_j, \ldots \hat{v_i}, \ldots v_n]} \\ &+ \sum_{j>i} (-1)^{i+j+1} (F(\sigma \times \mathbb{1}))\Big|_{[e_0, \ldots, \hat{e_i}, \ldots e_j, v_j, \ldots v_n]}. \end{aligned}$$

$$\begin{aligned} \partial P(\sigma) = &\sum_{j\leq i} (-1)^{i+j} (F(\sigma \times \mathbb{1}))\Big|_{[e_0, \ldots, \hat{e_j}, \ldots e_i, v_i, \ldots v_n]} \\ &+ \sum_{j\geq i} (-1)^{i+j+1} (F(\sigma \times \mathbb{1}))\Big|_{[e_0, \ldots e_i, v_i, \ldots \hat{v_j}, \ldots v_n]}. \end{aligned}$$

$$P\partial(\sigma) + \partial P(\sigma) = F(\sigma \times \mathbb{1})\Big|_{[v_0, v_1, \ldots v_n]} - F(\sigma \times \mathbb{1})\Big|_{[e_0, e_1, \ldots e_n]} = g_\#(\sigma) - f_\#(\sigma).$$

Given $\sigma \in Z_n(X)$,

$$g_\#(\sigma) - f_\#(\sigma) = \partial P(\sigma) + P\partial(\sigma) = \partial P(\sigma) \in B_n(Y).$$

From the previous statement, one can see that $f_\#$ & $g_\#$ map a cycle in X to homologous cycles in Y. Thus, the map induced at the homology level by f and g are the same. □

Proposition 1.3.3. Homotopically equivalent spaces X and Y, have isomorphic homology groups $H_n(X)$ and $H_n(Y)$ for all $n \in \mathbb{N} \cup 0$.

Proof. Let $f : X \longrightarrow Y$ be a homotopy equivalence and $g : Y \longrightarrow X$ be its homotopy inverse. It then follows from Proposition 1.3.2 and Theorem 1.3.1 that $g_* f_* = \mathbb{1}_{H_n(X)}$ and $f_* g_* = \mathbb{1}_{H_n(Y)}$. Thus, $H_n(X) \cong H_n(Y)$. □

Remark 1.3.1. The singular homology groups of a simplicial complex are the same as its simplicial homology groups. Using the homotopy invariance property of singular homology, we can also conclude that the singular homology groups of a triangulable space coincides with the simplicial homology of its triangulation.

1.4 Required Algebra

Definition 1.4.1 (Graded Ring)**.** A ring R is called *graded* if there exists a family of subgroups $\{R_i\}_{i\in\mathbb{Z}}$ of R such that

1. $R = \bigoplus_{i\in\mathbb{Z}} R_i$ as abelian groups and,
2. $R_n R_m \subseteq R_{n+m}$ for all $n, m \in \mathbb{Z}$.

A graded ring R is said to be *non-negatively graded* if $R_n = 0$ for all $n < 0$. For a given $i \in \mathbb{Z}$, any element in R_i is called a *homogeneous element* of *degree n*.

Definition 1.4.2 (Graded Modules)**.** A module M over a graded ring R is said to be a *graded* R-module if there exists a family of subgroups $\{M_i\}_{i\in\mathbb{Z}}$ of M such that

1. $M = \bigoplus_{i\in\mathbb{Z}} M_i$ as abelian groups and,
2. $R_n M_m \subseteq M_{n+m}$ for all $n, m \in \mathbb{Z}$.

A graded R-module is said to be *non-negatively graded* if $M_n = 0$ for all $n < 0$.

Definition 1.4.3 (α-shift)**.** Given a graded ring R and $\alpha \in \mathbb{Z}$, one can define a new ring denoted by $\sum^{\alpha} R$ as $\oplus_{i\in\mathbb{Z}} R_{\alpha+i}$ by shifting the gradation on R by α.

Theorem 1.4.1. Given a finitely generated module M over a PID D, there exist unique non-zero elements $d_1, \ldots, d_m \in D$, where $d_1 | d_2 \ldots | d_m$ and $\beta \in \mathbb{N} \cup \{0\}$ such that M is isomorphic to a direct sum of cyclic D-modules as follows:

$$M \cong D^{\beta} \oplus (\bigoplus_{i=1}^{m} D/d_j D). \tag{1.1}$$

A structure theorem for graded modules over a PID can be defined in a similar fashion as described in Theorem 1.4.1.

Theorem 1.4.2. Given a graded module M over a graded PID D, there exist unique homogeneous elements $d_1, \ldots, d_m \in D$ such that $d_1 | d_2 \ldots | d_m$ and $\alpha_1, \ldots, \alpha_n$ & $\gamma_1, \ldots, \gamma_m \in \mathbb{Z}$ such that

$$M \cong (\bigoplus_{i=1}^{n} \sum {}^{\alpha_i} D) \oplus (\bigoplus_{j=1}^{m} \sum {}^{\gamma_j} D/d_j D). \tag{1.2}$$

1.5 Computing Simplicial Homology over a PID

Simplicial homology over a PID, D can be easily computed using results from linear algebra. For most practical applications, the ground ring $\mathbb{Z}/2\mathbb{Z}$ is preferred.

$$\left[\begin{array}{cccc|c} d_1 & 0 & \ldots & & \\ 0 & d_2 & 0 & \ldots & \vdots \\ \vdots & 0 & \ddots & & 0 \\ & \vdots & & d_{k_n} & \\ \hline & & 0 & & 0 \\ & & & & \end{array}\right]$$

Figure 1.3: Smith-normal form of matrix M_n

A matrix M_n associated with boundary map $\partial_n : C_n(K) \to C_{n-1}(K)$ with the cycles as basis is constructed. This matrix can be reduced to the normal form as shown in 1.3 where $d_1 | d_2 \ldots | d_{m_n}$. Using the structure theorem 1.4.1, we can make the following observations.

- The elements greater than 1 in the set $\{d_1, \ldots d_{k_n}\}$ correspond to the torsion coefficients in the decomposition of $H_{n-1}(X)$.
- The number of zero columns in M_n, denoted by α_n, represents rank of the $Z_n(X)$.
- The $(n-1)^{th}$ Betti number of K, is given by $\beta_{n-1} = \alpha_{n-1} - k_n$.

Thus, we can determine the homology modules of K over D for all dimensions.

Similarly, the structure theorem over graded PIDs can be used to determine the graded homology modules.

Chapter 2

Geometric Reconstruction of Point Cloud Data

Before delving into topics in TDA, it is necessary to gain an understanding of the topological information of interest present in the data and the steps involved in estimating the same. This chapter explores these ideas by considering the specific example of point cloud data.

Here, a point cloud $\mathbb{X}$ is assumed to be a finite collection of points $\{x_1, \ldots, x_n\}$ in $\mathbb{R}^d$ sampled i.i.d using a probability measure μ with a compact support M. It is the topology of this underlying space M that one wishes to capture by using TDA.

The first step towards estimating the topological features of M using $\mathbb{X}$ is constructing structures on the point cloud which capture the required topological information. In section 2.1, we shall look at some complexes that can be built on the point cloud, and in section 2.2 define conditions under which they can be considered good representations of the underlying space M.

2.1 Complexes from Point Cloud

A space $\mathbb{X}_\epsilon$ can be naturally constructed from a given a point cloud $\mathbb{X} = \{x_1, \ldots, x_n\}$ by considering the union of closed balls of radius ϵ centred at $\{x_1, \ldots, x_n\}$. That is,

$$\mathbb{X}_\epsilon = \bigcup_{i=1}^{n} \overline{B_\epsilon(x_i)}.$$

$\mathbb{X}_\epsilon$ is known as the ϵ-*offset* or ϵ-*thickening* of the space $\mathbb{X}$.

Definition 2.1.1 (Nerve of a cover)**.** Given a cover $U = \{U_\alpha\}_{\alpha \in \mathcal{A}}$ of a space Y, the nerve of U is the abstract simplicial complex, $\mathcal{N}(U)$ whose k-simplices are determined by $k+1$ elements of U that have a non-empty intersection. That is,

$$[U_{i_0}, \ldots, U_{i_k}] \in \mathcal{N}(U) \iff \bigcap_{n=0}^{k} U_{i_n} \neq \emptyset.$$

Theorem 2.1.1 (Nerve Theorem)**.** Let $U = \{U_\alpha\}_{\alpha \in \mathcal{A}}$ be a cover of the space Y such that for any $\mathcal{A}' \subseteq \mathcal{A}$, the intersection $\bigcap_{\alpha \in \mathcal{A}'} U_\alpha$ is either contractible or empty. Then, the space Y is homotopically equivalent to $\mathcal{N}(U)$.

The Ĉech complex of the point cloud $\mathbb{X}$ for a given $\epsilon > 0$, denoted by $C_\epsilon(\mathbb{X})$, is the nerve of the covering $\{\overline{B_\epsilon(x_i)}\}_{i=1}^n$ of $\mathbb{X}_\epsilon$. As the intersection of closed balls in $\mathbb{R}^d$ is convex and hence contractible, it follows from Theorem 2.1.1 that the Ĉech complex is homotopically equivalent to $\epsilon-$thickening of $\mathbb{X}$.

Definition 2.1.2 (Ĉech Complex)**.** For a given point cloud $\mathbb{X} = \{x_1, \ldots, x_n\}$ and $\epsilon > 0$, the Ĉech complex, $C_\epsilon(X)$, is the abstract simplicial complex whose k-simplices are given by all $k+1$ points $\{x_{i_1}, \ldots, x_{i_{k+1}}\}$ such that $\bigcap_{j=0}^{k} \overline{B_\epsilon(x_{i_j})} \neq \emptyset$.

Definition 2.1.3 (Rips Complex)**.** For a given point cloud $\mathbb{X} = \{x_1, \ldots, x_n\}$ and $\epsilon > 0$, the *Rips complex*, $R_\epsilon(\mathbb{X})$, is the abstract simplicial complex whose k-simplices are determined by $k+1$ points $\{x_{i_1}, \ldots, x_{i_{k+1}}\}$ that are pairwise less than ϵ apart.

The Rips complex is an instance of a *flag complex* and is completely determined by the ϵ-connectivity graph given by its 1-simplices. It also should be noted that the Rips complex may lie in a Euclidean space of dimension greater than that in which the point cloud is embedded.

Proposition 2.1.1. For any given point-cloud $\mathbb{X}$ and $\epsilon > 0$,

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

Both these inclusions follow from the definitions of the Ĉech & Rips complex and the triangle inequality property of the Euclidean metric.

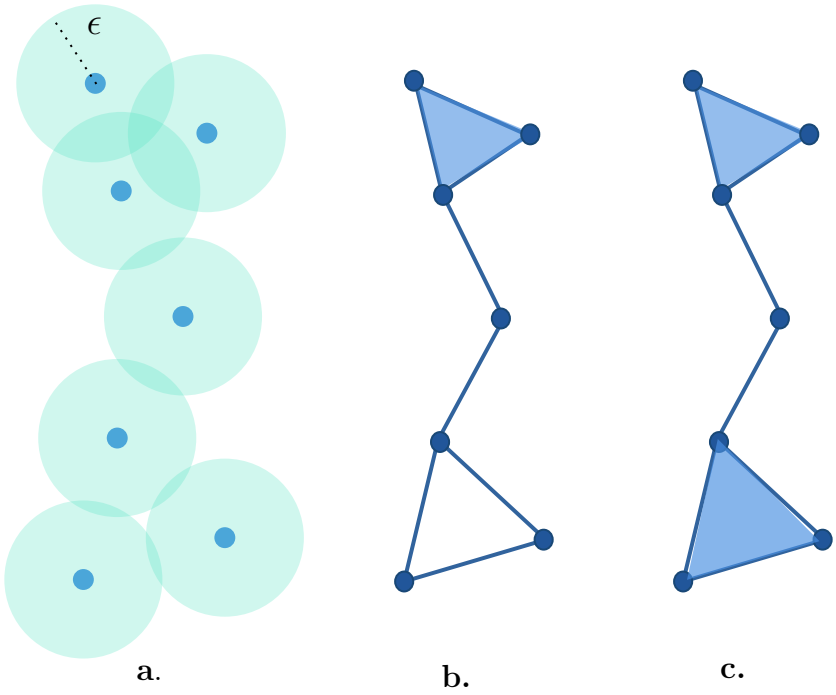


Figure 2.1: a. ϵ offset of a point cloud $\mathbb{X}$; b. Ĉech complex $C_\epsilon(\mathbb{X})$; c. Rips Complex $V_{2\epsilon}(\mathbb{X})$.

Remark 2.1.1. The number of intersections that need to be computed to determine the Ĉech complex of a point cloud combined with the size of the resultant complex often pose some practical difficulties. On the other hand, the ϵ-connectivity graph is more tractable to compute and store, making the Rips complex a more computationally viable option.

In cases where the point cloud is very large, even the construction of the Rips complex might prove difficult. In such scenarios, other complexes like the *weak* and *strong witness complexes* are preferred. These complexes are constructed on a smaller subset of the point cloud called the set of *landmark points*.

2.2 Reconstructing the Manifold

Definition 2.2.1 (Hausdorff Distance)**.** Given two subspaces K, K' of a metric space (X, d), the Hausdorff distance d_H : between them is given by

$$d_H(K, K') = \sup_{w \in X} | \inf_{x \in K} d(w, x) - \inf_{y \in K'} d(w, y)|.$$

The Hausdorff distance d_H defines a metric on the set of compact subspaces of a metric space (X, d). This can be used to define a distance function between two compact metric spaces as follows. This distance function can be viewed as a measure of how close the compact spaces are to being isometric.

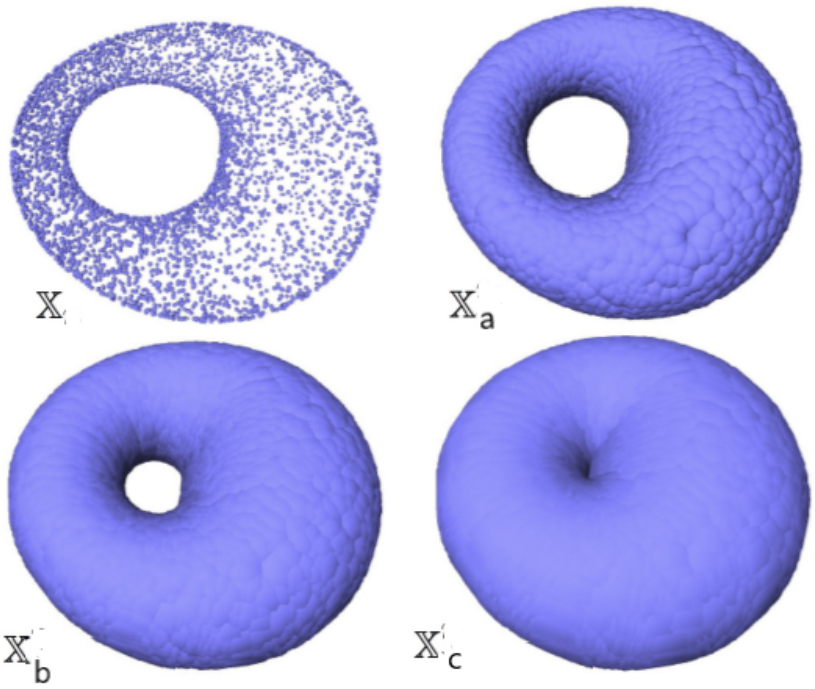


Figure 2.2: Point cloud $\mathbb{X}$ sampled from a torus along with the thickened spaces corresponding to values $a < b < c$. The thickened spaces $\mathbb{X}_a$ and $\mathbb{X}_b$ are homotopically equivalent to the torus. As the offset value increases, the 'central hole' gets filled in and the thickened spaces are no longer homotopically equivalent to the torus as evidenced by $\mathbb{X}_c$. (**Source:** [15])

Definition 2.2.2 (Gromov-Hausdorff Distance)**.** Let (K_1, d_1) and (K_2, d_2) be two compact metric spaces. The Gromov-Hausdorff distance $d_{GH}(K_1, K_2)$ is the infimum over all $r \geq 0$ such that there exists a metric space (X, d) and compact subsets $C_1, C_2 \subseteq X$ which are

isomteric to K_1 and K_2 respectively such that $d_H(C_1, C_2) \leq r$.

Consider the point cloud $\mathbb{X} \subseteq \mathbb{R}^d$ sampled using the probability measure μ with compact support M. As both $\mathbb{X}$ and K are compact subsets of $\mathbb{R}^d$, we can compute the Haussdorf distance between them. Additionally, we can define a map $d_M : \mathbb{R}^d \to \mathbb{R}^+$ that gives the distance of each point in $\mathbb{R}^d$ from the subspace M,

$$d_M \ : \ y \mapsto \inf_{m \in M} ||m - y||_2.$$

Under this setting, the ϵ-offsets of M correspond to the sublevel sets $d_M^{-1}([0, \epsilon])$ of the distance function d_M.

Proposition 2.2.1. The distance function d_M as defined, satisfies the following properties:

1. d_M is 1-Lipschitz, i.e. $|d_M(y) - d_M(z)| \leq ||y - z||$
2. ${d_M}^2$ is semiconcave, i.e. the map $y \mapsto ||y||^2 - {d_M}^2(y)$, is convex.

As a result of these properties, one can define the gradient of the distance function, $\nabla d_M \ : \ \mathbb{R}^d \longrightarrow \mathbb{R}^d$. In this case, a point $y \in \mathbb{R}^d$ is said to be α-critical if $||\nabla_x d_M|| \leq \alpha$. For an $\alpha \in (0, 1)$, the α-reach of d_M is the maximum value of R such that $d_M^{-1}((0, r])$ does not contain an α- critical point [8].

Theorem 2.2.1 (The Reconstruction Theorem)**.** For $\mathbb{X}$ and M in $\mathbb{R}^d$ as defined before, let $d_H(\mathbb{X}, M) < \epsilon$ and $\text{reach}_\alpha(d_M) \leq R$ for some $\alpha > 0$. Then, for any $\beta \in [4\epsilon/\alpha^2, R - 3\epsilon]$ and $\gamma \in (0, R)$, X_β the β-offset of $\mathbb{X}$ is homotopically equivalent to M_γ, the γ-offset of M, when

$$\epsilon \leq \frac{R}{5 + \frac{4}{\alpha^2}}.$$

The Reconstruction theorem establishes that under certain regularity conditions on the manifold, small offsets of the underlying manifold are homotopically equivalent to the β offset of $\mathbb{X}$ for suitably chosen values of β. We also know from using the nerve theorem, that this β-offset of $\mathbb{X}$ is homotopically equivalent to the Ĉech complex $C_\beta(\mathbb{X})$ constructed on the point cloud. When the underlying manifold is a compact Riemannian manifold, we also have the following result.

Theorem 2.2.2. For a compact Riemannian manifold M, there exists $\eta > 0$ such that for all $\epsilon < \eta$, the $\epsilon-$thickening of the manifold M_ϵ is homotopically equivalent to M.

These results provide a justification for using Ĉech complexes built on the point cloud to infer the topology of the underlying manifold.

From a practical perspective, there are a few challenges that need to be addressed before we can utilise these results to analyse data. The validity of the Reconstruction theorem rests on certain assumptions on the nature of the underlying manifold. Even if we were to work under the assumption that these satisfied for our data set, we still need a procedure to determine a suitable value of ϵ for building the Ĉech complex $C_\epsilon(\mathbb{X})$. Following this, we also need appropriate homotopically invariant objects or descriptors to quantify the topology of this complex.

The Betti numbers associated with the simplicial homology groups of the complexes are a good solution to the second issue due to their relative ease of computation. The other problems, while being more tricky to overcome, can be conveniently circumvented by taking into consideration all possible values of ϵ instead of choosing just one. These ideas form the basis for *persistent homology*, a central technique in TDA, that is discussed in the next chapter.

Chapter 3

Persistence Homology

Persistent homology is a framework to study the topological features of data through the lens of linear algebra. As indicated in the previous chapter, this is achieved by adopting homology theory to analyse a paremetrized family of topological spaces, generally simplicial complexes, built on data. This multi-scale approach helps us decipher complex relationships between features occurring at different scale levels.

The first section of this chapter will motivate the notion of *persistence* and the construction of persistence modules using examples. In the subsequent sections, we shall look at the algebraic structure of the persistence modules and how this can be utilised to define an algorithm for computing this object as described in [3]. The subject of the final section will be *persistence barcodes*, a parametrization of the persistence modules.

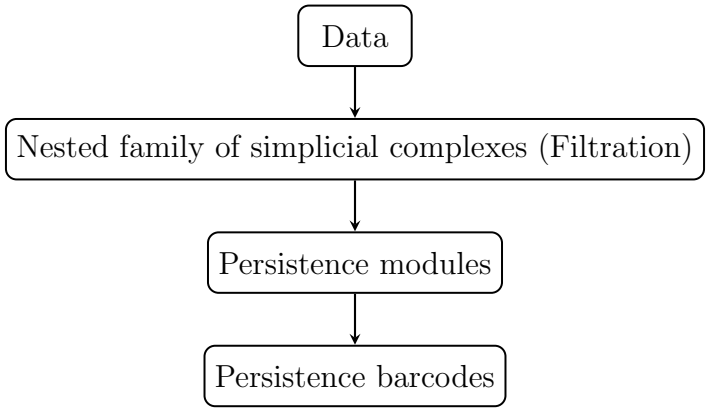


Figure 3.1: TDA pipeline

With this, we would have established most key components required to construct a standard TDA pipeline. In Figure 3.1 we have a flow chart outlining the steps involved in the same.

3.1 Persistence

Consider the Ĉech complexes constructed on a point cloud $\mathbb{X}$ with respect to the values ϵ_1 and ϵ_2, where $\epsilon_1 < \epsilon_2$, as depicted in Figure 3.2. The structure of these complexes suggests that a majority of the data points are concentrated around a circular loop.

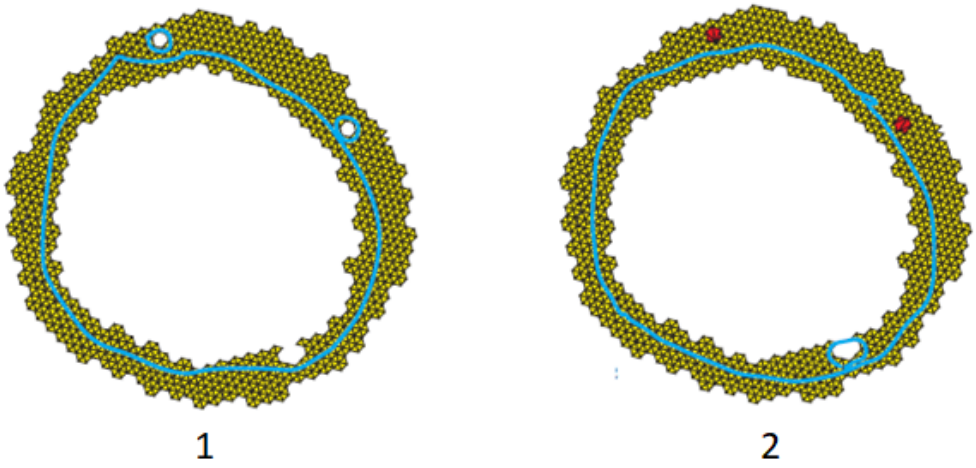


Figure 3.2: Ĉech complexes on point cloud $\mathbb{X}$, C_{ϵ_1} (Left) and C_{ϵ_2} (Right). (**Source:** [7])

The Betti number associated with the complex $C_{\epsilon_1}(\mathbb{X})$ is 3. Here, the larger loop indicates a more prominent feature of the data, while the two smaller loops are likely a result of noise or errors in sampling.

For a slightly larger value ϵ_2, it can be seen that the smaller loops present at ϵ_1 get filled in, while the large central loop remains. However, another loop has also been formed now. While neither ϵ_1 or ϵ_2 are the right choices, the inclusion $H_n(i) : H_n(C_{\epsilon_1}) \to H_n(C_{\epsilon_2})$ sheds more light as the image of this map is just the homology class of the large central loop.

From this exercise, one can see that more pertinent information on the topological features of the data can be extracted by considering all parameter values rather than just one. The

theory of *persistence* which is formalised in the following section builds on this very notion.

3.2 Persistence Module

Definition 3.2.1 (Persistence Complex)**.** A *persistence complex* $\mathcal{C}$ consists of a family of chain complexes $\{C^i\}_{i\in\mathbb{N}\cup\{0\}}$ over a ground ring R along with chain maps $\{f_i : C^i \longrightarrow C^{i+1}\}_{i\in\mathbb{N}\cup\{0\}}$.

$$\begin{array}{ccccccccc}
\ldots & \longrightarrow & C_{n+1}^{i-1} & \xrightarrow{f_{i-1}} & C_{n+1}^{i} & \xrightarrow{f_i} & C_{n+1}^{i+1} & \longrightarrow & \ldots \\
 & & \downarrow\partial & & \downarrow\partial & & \downarrow\partial & & \\
\ldots & \longrightarrow & C_{n}^{i-1} & \xrightarrow{f_{i-1}} & C_{n}^{i} & \xrightarrow{f_i} & C_{n}^{i+1} & \longrightarrow & \ldots \\
 & & \downarrow\partial & & \downarrow\partial & & \downarrow\partial & & \\
\ldots & \longrightarrow & C_{n-1}^{i-1} & \xrightarrow{f_{i-1}} & C_{n-1}^{i} & \xrightarrow{f_i} & C_{n-1}^{i+1} & \longrightarrow & \ldots
\end{array}$$

Figure 3.3: Persistence Complex

Definition 3.2.2 (Persistence Module)**.** A *persistence module* $\mathcal{M}$ is a family of R-modules $\{M_i\}_{i\in\mathbb{N}\cup\{0\}}$ along with R-linear maps $\{\phi_i : M_i \longrightarrow M_{i+1}\}_{i\in\mathbb{N}\cup\{0\}}$.

Definition 3.2.3. A persistence complex $\mathcal{M}$ is said to be of *finite type* if each M_i is finitely generated and if there exists an n such that for all $i \geq n$, ϕ_i is an isomorphism. One can similarly define a persistence module of finite type.

Example. A filtered simplicial complex $K^0 \subseteq K^1 \subseteq \ldots K^m$ is an example of a persistence complex. The corresponding homology modules along with the maps induced by inclusion, constitute a persistence module. If the simplicial complex is finite, then both the persistence complex and the persistence module are of finite type.

Example. For a point cloud $\mathbb{X}$, given any increasing sequence of real numbers $\{\epsilon_i\}_{i=0}^n$, the corresponding Ĉech complexes $\{C_{\epsilon_i}(\mathbb{X})\}_{i=0}^n$ determine a simplicial filtration. As the objects of interest in persistence are the inclusion maps at the homology level, and since Rips complexes can be squeezed between Ĉech complexes, it is sufficient for our purpose to construct Rips complexes over the point cloud.

Given a persistence module over $\mathcal{M}$, we can define an $R[t]$-module $\alpha(\mathcal{M})$ as

$$\alpha(\mathcal{M}) \;=\; \bigoplus_{i=0}^{\infty} M_i,$$

where given an element $(m_0, m_1, \ldots) \in \alpha(\mathcal{M})$, the action of t is given by

$$t.(m_0, m_1, \ldots) \;=\; (0, \phi_0(m_0), \phi_1(m_1), \ldots).$$

It follows directly from the definition of $\alpha(\mathcal{M})$, that it is in fact a graded module over $R[t]$.

Remark 3.2.1. The map α defines an equivalence between the category of persistence modules of finite type and the category of finitely generated non-negatively graded $R[t]-$modules.

With respect to the example of a filtered simplicial complex, this construction enables us to view the persistence homology modules of the filtration as a single graded module. Here, the order of appearance of the simplices is reflected through the gradation of the module.

The structure theorem 1.4.2 for graded modules over graded PIDs, motivates the choice of base ring of the persistence modules and complex to be a field. In such a case where the base ring is a field F, we have the following decomposition of the graded module $\alpha(\mathcal{M})$

$$\alpha(M) \;\simeq\; \bigoplus_{i=1}^{n} \Sigma^{\alpha_i} F[t] \;\oplus\; \bigoplus_{j=1}^{m} \Sigma^{\gamma_j} F[t]/t^{\nu_j}. \tag{3.1}$$

Based on this decomposition, $\alpha(\mathcal{M})$ can be further parametrized by a multiset of ordered tuples (i, j) of $\mathbb{R} \cup \{\infty\}$, where $i < j$, called a $\mathcal{P}-$interval . This is achieved using a map Q as follows.

Given $(i, j) \in \mathcal{S}$,

$$Q(i, j) = \Sigma^i F[t]/t^{j-i}, \text{ when } i < j < \infty,$$

$$Q(i,j) = \Sigma^i F[t], \text{ when } i < j = \infty, \text{ and}$$

$$Q(\mathcal{S}) := \bigoplus_{(i,j)\in(S)} Q(i,j).$$

The map Q defines a bijection between finite multisets of $\mathcal{P}-$intervals and the isomorphism classes of persistence modules of finite type over F.

3.3 Persistent Homology of a Simplicial Filtration

Consider a filtered simplicial complex K over a field F. The n-th homology module of any constituent simplex K^i is a vector space that is completely determined by its Betti number. In the previous section, it has been established that this associated family of homology vector spaces can be viewed as a graded module over $F[t]$ which accepts a decomposition as described by Theorem 1.4.2.

This decomposition guarantees the existence of a homogeneous basis for the $F[t]$-graded persistence module. This in turn defines a suitable basis for the constituent homology vector spaces that can be used to devise an algorithm to compute the persistent homology.

In this setting, the basis element of the module $Q(i,j) = \Sigma^i F[t]/t^{j-i}$, $j < \infty$ corresponds to an n-cycle that first appears in the complex K^i of the filtration which becomes an n-boundary only for the simplices K^p where $p \geq n$. When $j = \infty$ the basis of the module represents an $n-$cycle born at i that remains unbounded throughout the filtration.

3.3.1 Computing Persistent Homology

This section describes a standard algorithm for computing the persistent homology of a filtered simplicial complex over a field $\mathbb{F}$ as presented in [3]. The filtered simplicial complex pictured in Figure 3.4 will be used as an example to demonstrate steps in the algorithm over the field $\mathbb{Z}_2$.

Consider a simplicial complex K equipped with the filtration $K^0 \subseteq K^1 \ldots \subseteq K^m$. The p-th persistence homology module of this filtration is as follows:

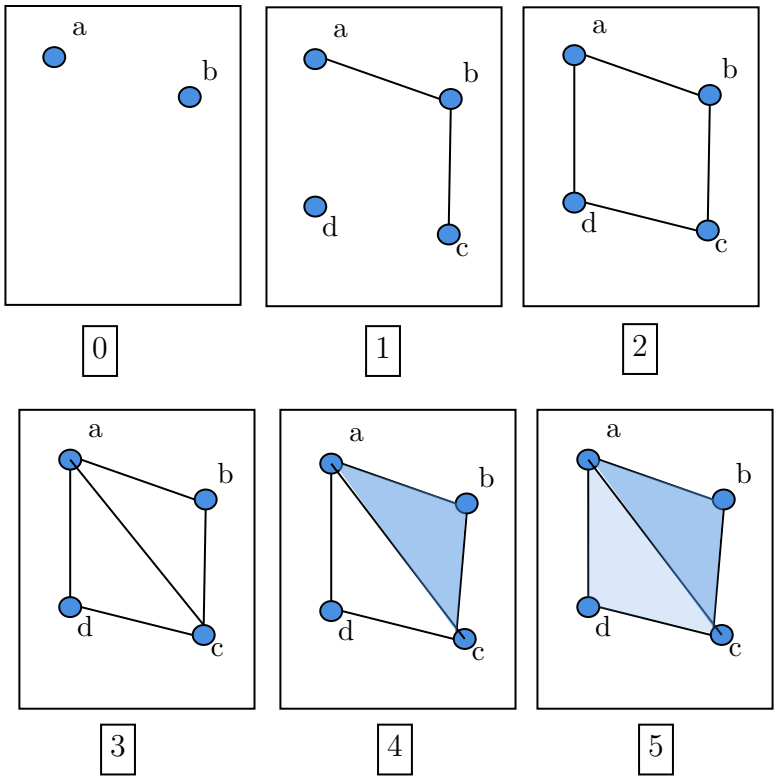


Figure 3.4: Example: Filtered simplicial complex

$$\bigoplus_{i=0}^{m} H_p(K^i) \;=\; \bigoplus_{i=0}^{m} \frac{Z_p(K^i)}{B_p(K^i)} \;=\; \frac{\bigoplus_{i=0}^{m} Z_p(K^i)}{\bigoplus_{i=0}^{m} B_p(K^i)}.$$

As a result, the $p-$th persistent homology module of the filtered complex K is the $p-th$ homology module corresponding to the chain complex whose p-chains $C_p(K)$, are the graded $\mathbb{Z}_2[t]$ module $\bigoplus_{i=1}^{m} C_p(K^i)$. The ordering of simplices in the filtration is translated as the gradation in the $p-$chains.

Similar to algorithm for computing the homology of a simplicial complex describe in Section 1.5, the first step here is to define the matrix corresponding to the boundary maps $\partial_p : C_p(K) \longrightarrow C_{p-1}(K)$ of the filtered simplicial complex.

Let $\{e_j\}$ and $\{\tilde{e}_i\}$ be the standard homogeneous basis for $C_p(K)$ and $C_{p-1}(K)$ respectively and M_p denote the matrix of ∂_p relative to these bases. The following relationship between the degrees of the basis and matrix elements can be established,

$$\deg(M_p(i,j)) = \deg(e_j) - \deg(\tilde{e_i}). \tag{3.2}$$

For the example in consideration, we have the following matrix representation M_1 of ∂_1

	ab	bc	cd	ad	ac
d	0	0	t	t	0
c	0	1	t	0	t
b	t	t	0	0	0
a	t	0	0	t^2	t^3

Figure 3.5: Matrix representation M_1 of ∂_1

Given a matrix representation of ∂_p in terms of a homogeneous basis for $C_p(K)$ and $Z_{p-1}(K)$, the homology module $H_{p-1}(K)$ can be determined using the Smith normal form. Since, $Z_0 = C_0$, the matrix M_1 is in the desired format. A homogeneous basis for $Z_p(K)$ and the matrix representation of ∂_{p+1} with respect to this can be determined inductively as sketched out below.

Assume that the representation M_p of ∂_p is in terms of these desired bases. Furthermore, order the basis $\{\tilde{e_j}\}$ in decreasing order of degree and $\{e_i\}$ by increasing degree. The matrix corresponding to this ordering of the bases is then be obtained by performing suitable row and column swaps. The matrix M_1 given above is already in this form.

The next step is to obtain the column echleon $\tilde{M_p}$ of this matrix. This is obtained by first performing column operations representing the basis change $e_j \to e_j + q * e_i$, to eliminate the non - zero entries in the pivot rows and by subsequently swapping columns.

	ab	bc	cd	z_1	z_2
d	0	0	$\boxed{t}$	0	0
c	0	$\boxed{1}$	t	0	0
b	$\boxed{t}$	t	0	0	0
a	t	0	0	0	0

Figure 3.6: Matrix representation $\tilde{M_1}$ of ∂_1 with the pivot elements in boxes, $z_1 = ad + cd + t.bc + t.ab$ and $z_2 = ac + t^2.bc + t^2.ab$.

The basis elements corresponding to the the non-pivot columns of $\tilde{M_p}$ forms a basis for $Z_p(K)$. Also,

$$\#\text{ pivots of } \tilde{M_p} \;=\; \text{rank}(\tilde{M_k}) \;= \text{rank}(B_{p-1}(K)).$$

Proposition 3.3.1. The pivot elements of the column echleon matrix $\tilde{M_p}$ is the same as the diagonal entries of the Smith normal form of this matrix.

Proof. Given any column j in M_p, $\deg(M_p(i,j)) = \deg(e_j) - \deg(\tilde{e_i})$. As the rows are already sorted by decreasing degree, the column element with the highest degree is the pivot. As a result, all other entries in this column can be eliminated by using row operations. All the pivot elements remain unchanged during this and the matrix can then be put in the Smith normal form by suitable row and column swaps. □

As a result, the $p-$th persistent homology module can be determined from $\tilde{M_p}$ using the decomposition described in Theorem 1.4.2.

Corollary 3.3.1. Given $\tilde{M_p}$, a matrix of ∂_p in column echelon form with respect to the basis $\{e_j\}$ for $C_p(K)$ and $\{\tilde{e_i}\}$ for $Z_{p-1}(K)$,

1. A pivot element $M_p(i,j)$ of degree n contributes to the module $\Sigma^{\deg(\tilde{e_i})}F[t]/t^{n+\deg(\tilde{e_i})}$ in the decomposition of $(p-1)$-th persistent module. This corresponds to the element $(\deg(\tilde{e_i}), \deg(\tilde{e_i}+n)$ in the $\mathcal{P}-$interval.
2. A zero row corresponding to the basis element e_i gives rise to the module $\Sigma^{\deg(\tilde{e_i})}F[t]$ and in turn the $\mathcal{P}$-interval element $(\deg(\tilde{e_i}), \infty)$.

To represent M_{p+1} in terms of a homogeneous basis for $Z_p(K)$, the following relation is used,

$$\partial_p \circ \partial_{p+1} \;=\; 0 \implies M_p \circ M_{p+1} \;=\; 0.$$

To effect the same basis change, $e_j \to q.e_i + e_j$, achieved by a column operation in M_p, a row operation where $\text{row}(i) \to (-q).\text{row}(j) + \text{row}(i)$ needs to performed on M_{p+1}. The above relation between these matrices remains unchanged under such operations.

Proposition 3.3.2. The matrix representation of ∂_{p+1} with respect to a basis for $C_{p+1}(K)$ and $Z_p(K)$ can be obtained from M_{p+1} by eliminating rows that correspond to pivot columns in M_p.

Proof. As already seen, reducing the matrix M_p to column echleon forms involves using column operations representing basis change $e_j \rightarrow q.e_i + e_j$, where the pivot element in column i eliminates a non-zero element in non-pivot column j and by subsequently utilising column swaps.

While the corresponding row swaps doesn't change the elements of $M_{p+1}(K)$, the row operations $\text{row}(i) \rightarrow (-q).\text{row}(j) + \text{row}(i)$ changes only the elements in rows that represent pivot columns in M_p. Eventually, these rows representing pivot columns in M_p become zero as a result of these operations. □

	abc	acd
ac	t	t^2
ad	0	t^3
cd	0	t^3
bc	t^3	0
ab	t^3	0

Figure 3.7: Matrix representation M_2 of ∂_2

	abc	acd
z_2	t	t^2
z_1	0	t^3

Figure 3.8: Matrix representation of ∂_2 with respect to basis for $C_2(K)$ and $Z_1(K)$ obtained by deleting rows in M_2 corresponding to pivot columns in M_1. $z_1 = ad + cd + t.bc + t.ab$ and $z_2 = ac + t^2.bc + t^2.ab$

The results discussed in this section form the basis for the algorithm described for computing $\mathcal{P}-$intervals of a filtered simplicial complex over all dimensions.

3.3.2 The Algorithm

As seen previously, the persistent homology modules of a filtered simplicial complex can be determined solely using column operations on the boundary matrices. In fact, it is sufficient for our purpose to just consider the set of boundary chains corresponding to each columns in

the boundary matrix. Additionally, equation 3.2 enables us to simulate the process over F instead of $F[t]$. Libraries that offer an implementation of this algorithm include the Dionysus and Gudhi libraries in C++.

The input to the algorithm is an ordering of all simplices in the complex K in terms of increasing degree where ties are broken arbitrarily. A data structure T with slots for each simplex which also allows for the marking of a simplex is defined.

Algorithm 1 Computing $\mathcal{P}$-intervals

```
procedure ComputeIntervals(K)
    Initialize:
    L_k ← ∅,  k = 1, ..., dim(K)
    T[k] ← ∅,  k = 1, ..., m
    for j = 1 to m do
        d = RemovePivotRows(σ_j);
        if d == ∅ then
            Mark σ_j;
        else
            i = maxindex(d);
            k = dim(σ_i);
            T[i] = d, j ;
            L_k = L_k ∪ {deg(σ_i), deg(σ_j)};
    for j = 1 to m do
        if σ_j is marked and T[j] ≠ ∅ then
            k = dim(σ_j);
            L_k = L_k ∪ {deg(σ_j), ∞}
```

The procedure ComputeIntervals() iteratively analyses boundary chains of each simplex in the filtration and computes the $\mathcal{P}-$intervals of all homology dimensions. Using the function RemovePivotRows(), one can determine whether the boundary chain of a given simplex σ_j represents a pivot or non-pivot column in the respective boundary matrix. If the column includes a pivot in row i, this corresponds to a $\mathcal{P}-$interval $(\deg(\sigma_i), \deg(\sigma_j))$ and both the boundary chain and the value j are stored in $T[i]$. If not, the simplex is marked as the corresponding row should not be deleted in the next dimension.

Following this, we loop through T to identify empty slots that also correspond to marked simplices. These represent intervals that persist till ∞ in persistent homology module.

The function RemovePivotRows() computes the column corresponding to a given

Algorithm 2 Returns column of boundary matrix in echleon form

```
function RemovePivotRows(σ)
    d = ∂(σ);
    Remove rows corresponding to unmarked simplices in d;
    while d ≠ ∅ do
        i = maxindex(d)
        if T[i] == ∅ then
            break;
        else
            q = coefficient of σ_i in T[i];
            d = d − q^{-1} T[i];
    return d;
```

simplex in the column echleon form of the boundary matrix. Starting with the boundary chain, one looks at maximum index as this is potentially a pivot location. If the slot at T for this index is filled, this implies that it is not a pivot and the element at this location is then eliminated using a column operation. This process is repeated till a pivot is found or the set becomes empty. For the example in consideration, the algorithm gives the following results: $L_0 = \{(0, \infty), (0, 1), (1, 2), (1, 1)\}$ and $L_1 = \{(3, 4), (2, 5)\}$.

3.4 Representing Persistent Homology

While the discussions so far focused on simplicial filtrations, these ideas can easily be generalised in terms of *tame* functions. A function $f : X \to \mathbb{R}$ is said to be *tame* if the homology module corresponding to each sublevel set $f^{-1}((-\infty, t])$ is of finite rank, and if there are finitely many values $t_1 < t_2 < \ldots < t_m \in \mathbb{R}$ across which the homology maps are not isomorphic.

In this case, one selects $s_0, s_1, \ldots, s_m$ such that $s_i - 1 < t_i \leq s_i$ and uses the persistence complex given by the sublevel sets $\{f^{-1}((-\infty, s_i])\}_{i=0}^{m}$ to computes the persistent homology. Here, intervals of the form (t_i, t_j) corresponding to $\mathcal{P}$-intervals (i, j) are used.

The intervals corresponding to the persistent homology modules can be depicted as a "barcode" as shown below in 3.9. Here, a bar joining the points s and t, represents the interval (s, t). At any point $u \in \mathbb{R}$, the number of bars of dimension k containing the u is

equal to the $k^{th}-$Betti number of the space $(f^{-1}((-\infty, u]))$.

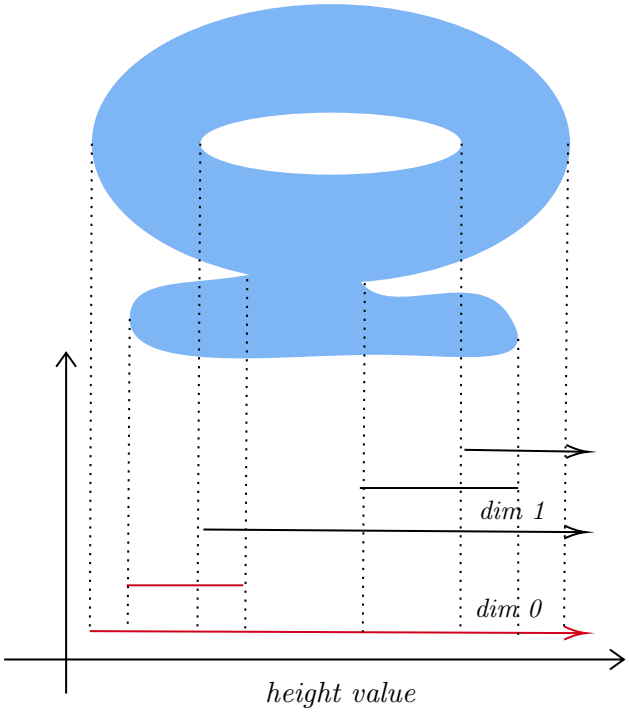


Figure 3.9: Persistence barcodes for dimensions 1 and 2, corresponding to the x-height function on this surface in $\mathbb{R}^3$.

Definition 3.4.1 (Persistence Diagram)**.** Another oft used representation is the *persistence diagram*. For a fixed homology dimension, the persistence diagram consists of the union of the set of tuples $\{(t_i, t_j)\}$ representing the intervals and the set of all points along the diagonal, $\Delta = \{(x, x \mid x \in \mathbb{R}\}$ considered with infinite multiplicity.

This can be represented in $\mathbb{R}^2$ as shown in Figure 3.10. For features persisting till ∞, it is often desirable to use a cutoff value instead.

In this setup, the *persistence* of a feature corresponds to the distance of the point representing it from the diagonal. Therefore, features caused due to noise or sampling error are represented by points closer to the diagonal in the persistence diagram.

The set of all persistence diagrams can be viewed as a metric space under the Wasserstein metric. This can be used to establish results on the stability of persistent homology.

Definition 3.4.2 (Wasserstein Distance)**.** For $p, q \in [1, \infty]$, and persistence diagrams D_1 and D_2, the $p^{th}-$*Wasserstein Distance* between them using the L_q metric on $\mathbb{R}^2$ is as geven

by,

$$d_p(D_1, D_2) = \inf_{\phi: D_1 \to D_2} \Big(\sum_{a \in D_1} ||a - \phi(a)||_q^p \Big)^{\frac{1}{p}},$$

where infimum is over all bijection $\phi : D_1 \to D_2$. When $p = q = \infty$, this is known as the *Bottleneck distance* and is denoted by d_B.

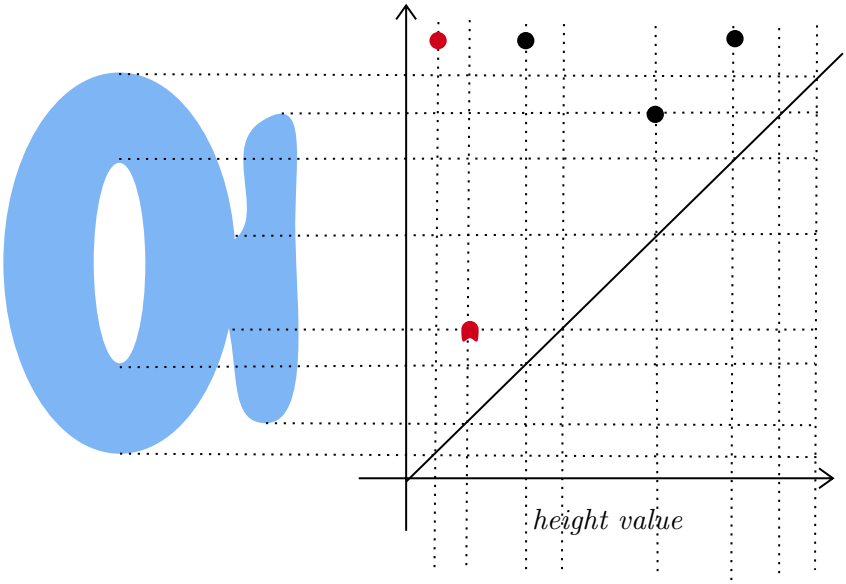


Figure 3.10: Persistence diagram corresponding to the height function on space. Red represents homology dimension 0 and black, dimension 1.

Proposition 3.4.1 (Stability)**.** Given tame functions $f, g : X \to \mathbb{R}$, let $D(f)$ and $D(g)$ represent the respective persistence diagrams for some homology dimension. Then,

$$d_B(D(f), D(g)) \leq ||f - g||_\infty.$$

Corollary 3.4.1. Given point clouds $\mathbb{X}$ and $\mathbb{Y}$, let $Filt(\mathbb{X})$ and $Filt(\mathbb{Y})$ denote the filtration corresponding to the Ĉech or Rips construction for some homology dimension. Then,

$$d_B\Big(D(Filt(\mathbb{X})), D(Filt(\mathbb{Y}))\Big) \leq 2 d_{GH}(\mathbb{X}, \mathbb{Y}),$$

where d_{GH} denotes the Gromov-Hausdorff distance.

Chapter 4

Vectorized Representations of Persistent Homology

The representations of persistent homology - *barcodes* and *diagrams*, discussed in the previous chapter are not good candidates for performing statistical analysis or for serving as input to machine learning tasks. The is in part due to their restrictive structure which poses difficulties in defining algebraic operations on them. This problem can be tackled by "vectorising" persistence diagram using summaries that map them to elements of a vector space. In this chapter, we shall review few such summaries.

4.1 Real-valued Summaries

Summaries that map persistence diagrams to real numbers are useful for conducting statistical analysis on small samples and can also easily be combined with machine learning techniques. One such summary, is the $p^{th}-$*Wasserstein amplitude*, which maps each persistence diagram to its Wasserstein distance from the empty diagram which contains only the diagonal points. Other commonly used summaries include total persistence, maximum persistence and persistent entropy [12].

Consider a persistence diagram $D = \{(b_i, d_i)\}_{i=1}^{m}$. Let the persistence of each feature, given by the length of bars be denoted by $\{l_i\}_{i=1}^{m}$ where $l_i = d_i - b_i$.

Definition 4.1.1. For the persistence diagram D, The *total persistence* is given by $\sum_{i=1}^{m} l_i$ and the *maximum persistence* is equal to $\max_i\{l_i\}$.

Definition 4.1.2 (Persistent Entropy)**.** Persistence entropy E_D of a persistence diagram D is the Shannon entropy of the distribution of its bar lengths. That is,

$$E_D = -\sum_{i=1}^{m} \frac{l_i}{L}\, log(\frac{l_i}{L}),$$

where $L = \sum_{i=1}^{i=m} l_i$. Persistent entropy is a scale invariant summary function which is stable under small perturbations.

4.2 Persistence Images

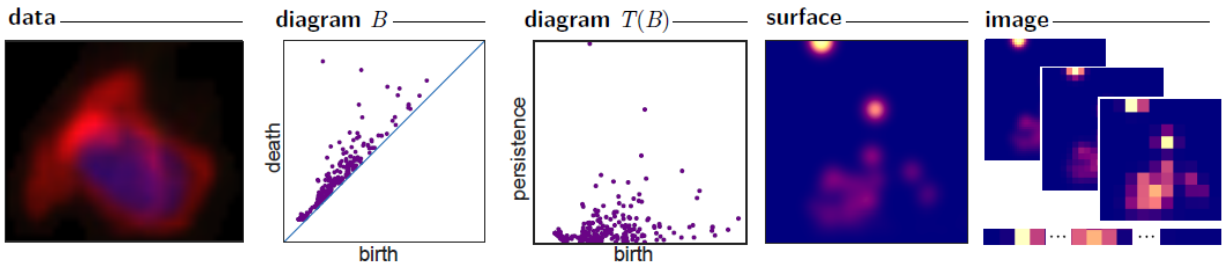


Figure 4.1: Sequence for obtaining persistence images from the given data.(**Source**: [14])

Persistence images are a stable (with respect to the $1^{st}-$Wasserstein distance) and interpretable finite dimensional vector representation of persistence diagrams [14]. Given a persistence diagram D, its persistence image is obtained as follows.

1. A transformation $T : \mathbb{R}^2 \to \mathbb{R}^2$, where $T(x, y) = (x, y - x)$ is applied to the diagram D. The transformed diagram $T(D)$, is now in birth-persistence coordinates.

2. $\Phi_u : \mathbb{R}^2 \to \mathbb{R}$, a differentiable probability distribution with mean $u \in \mathbb{R}^2$ and $f : \mathbb{R}^2 \to \mathbb{R}$, a continuous, piece-wise differentiable weight function which is zero on points on the $x-$axis is chosen. For instance, Φ_u can be a normalised Gaussian distribution with mean u, and f a weighting function that depends only on the y-persistence coordinate.

3. The persistence surface $P_D : \mathbb{R}^2 \to \mathbb{R}$, is given by

$$P_D(z) = \sum_{u \in T(D)} f(u)\Phi(z).$$

The weights corresponding to points along the diagonal in D is zero as T maps them onto the x-axis and these do not contribute to the above sum.

4. The persistence image $I(P_D)$ is obtained from the persistence surface by discretising the relevant region using a grid and integrating over each box/pixel. That is for any pixel p in the persistence image, the intensity is given by $I(P_D)_p = \iint_p P_D \; dydx$.

The persistence images offer a lot of flexibility through user choices for distribution and weight function. They can also be integrated with ML algorithms easily.

4.3 Betti Curves

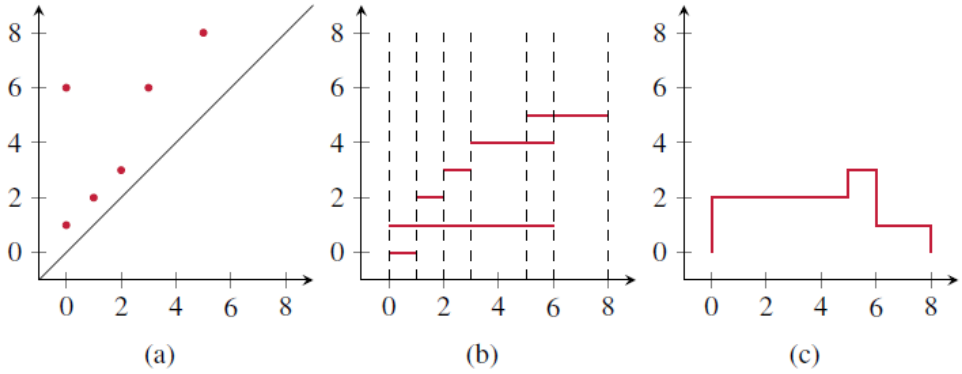


Figure 4.2: (a) A persistence diagram, (b) Its persistence barcode and (c) The corresponding Betti curve. (**Source:** [24])

Given a persistence diagram D, the associated Betti curve or persistence indicator function given by $\beta_D : \mathbb{R} \to \mathbb{N}$ is defined as follows

$$\beta_D : t \mapsto \left|\{i \mid t \in [b_i, d_i]\}\right| = \sum_{i=1}^{m} \mathbb{1}_{[b_i, d_i]}(t).$$

The Betti curve is a piecewise linear function on $\mathbb{R}$, and can hence be viewed as an element of the function space $L^p(\mathbb{R})$ for $p \geq 1$. Statistical analysis can be performed by considering the norm of the Betti curves as random variables [24].

4.4 Entropy Summary Function

Since persistence diagrams lie in an infinite dimensional space, mapping them to $\mathbb{R}$ using persistent entropy may result in the loss of some relevant information. This can be overcome by using the *entropy summary function*, a piece-wise linear function which combines persistent entropy with the persistence indicator function.

For a persistence diagram D, the entropy summary function is given by $ES_D : \mathbb{R} \to \mathbb{R}$ is given by

$$ES_D \ : \ t \mapsto -\sum_{i=0}^{m} \mathbb{1}_{[b_i,d_i]}(t) \ \frac{l_i}{L} \ log(\frac{l_i}{L}).$$

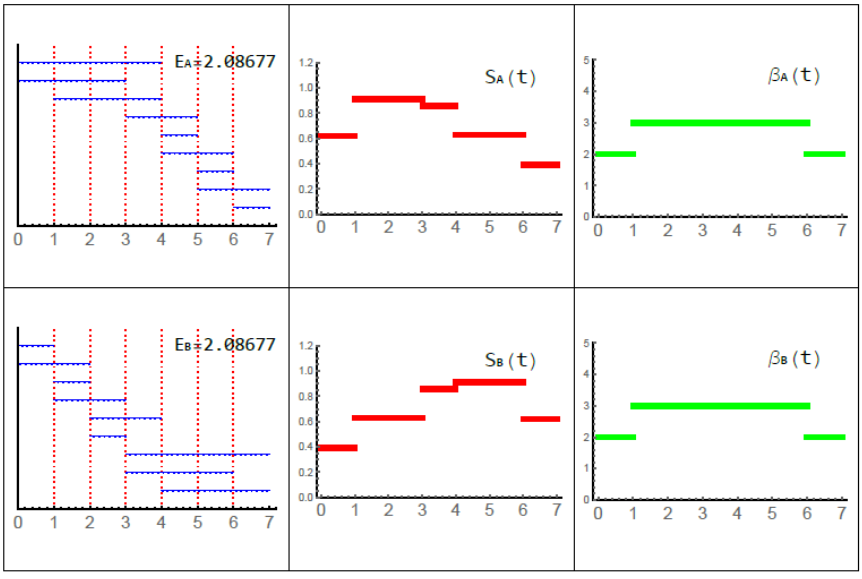


Figure 4.3: Two persistent diagrams with the same persistent entropy value and Betti curve but different entropy summary functions (**Source:** [23])

.

4.5 Persistence Landscapes

Persistence landscape is a functional summary which associates each persistence diagram with a sequence of piece-wise linear real valued functions. This structure makes these objects easy to compute and also lends itself for statistical tasks.

Given a persistence diagram D, the associated persistence landscape $\lambda = \{\lambda_k : \mathbb{R} \to \mathbb{R}\}_{k\in\mathbb{N}}$ is obtained by rotating the diagram clockwise by 45°, constructing isoceles triangles as shown in 4.4 and by tracing out the outermost layer inductively. That is, the $k-$th landscape function λ_k is given by,

$$\lambda_k(t) = \underset{i\in\{1,\ldots m\}}{\text{kmax}} \{\min(t - b_i, d_i - t)_+\},$$

where kmax gives the k^{th} largest value in the set and $c_+ = \max(c, 0)$.

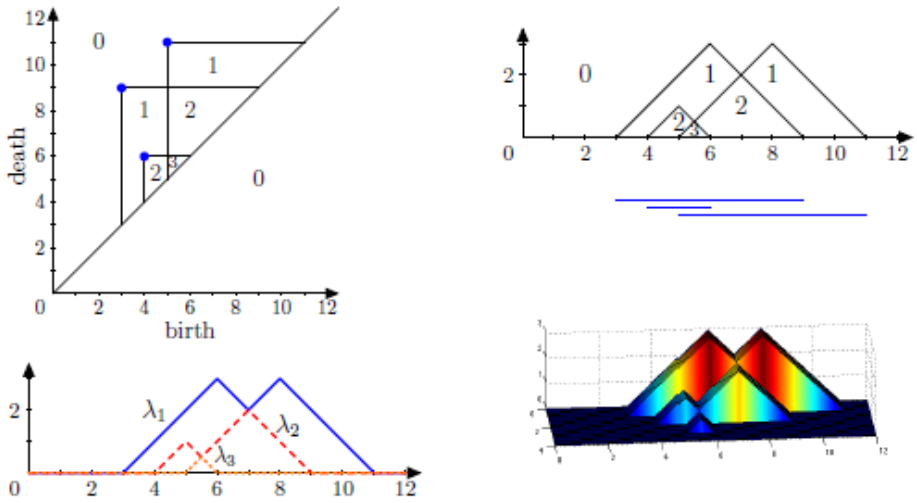


Figure 4.4: The persistence landscape functions of the given persistence diagram.(**Source:** [11])

$\lambda \in L^p(\mathbb{N}\times\mathbb{R})$ for $1 \le p \le \infty$ as

$$||\lambda||_p^p = \sum_{i\in\mathbb{N}} ||\lambda_i||_p^p < \infty.$$

Therefore, a persistence landscape can be viewed as an element in a separable Banach space. The Banach structure can be used to prove a central limit theorem for these objects [11].

For two persistence diagrams D, D' and their respective landscapes λ, λ', the $p^{th}-$landscape distance between D and D' is $\Lambda_p(D, D') = ||\lambda - \lambda'||_p$.

Proposition 4.5.1 (Stability)**.** Let D, D' be persistence diagrams and $f, g : X \to \mathbb{R}$ represent tame functions over X. We have the following results that establish the stability of persistence landscapes

- $\Lambda_\infty(D, D') \leq W_\infty(D, D')$, and
- $\Lambda_\infty(D(f), D(g)) \leq ||f - g||_\infty$.

4.6 Persistence Silhouette

Given a persistence diagram D, a univariate functional summary called persistence silhouette, $PS_D : \mathbb{R} \to \mathbb{R}$ can be obtained from the persistence landscape as shown below.

$$PS_D(t) \;=\; \frac{\sum_{i=1}^{m} w(d_i - b_i)\lambda_i(t)}{\sum_{i=1}^{m} w_i(d_i - b_i)}.$$

Here w is a persistence based weight function.

Chapter 5

Topological Pipeline for Image Analysis

In problems of image classification, one can often identify distinct shape features that characterize images in each class. In this project, we developed a 'topological pipeline' which uses persistent homology to extract topological descriptors from each image based on which classification can then be performed.

In 5.3, we shall expand on various aspects of the pipeline using the MNIST dataset of hand written digit images. This dataset consists of 70,000 grayscale images of dimension 28×28 pixels.

Subsequently, we used the pipeline for image classification of the Fashion MNIST [18], High Resolution Fundus [10] and Flower images [6] datasets. The results of these experiments have been presented in 5.4.

5.1 Topological Pipeline: Filtrations

The natural structure of grayscale images as a pixel intensity map on a rectangular grid, lends itself for the construction of cubical complexes. Given any function on a grid, a filtered cubical complex can be constructed corresponding to the sub-level or super-level sets of the

given function. In the Figure 5.1 below, examples of sublevel and superlevel sets of the grayscale intensity map is shown. Persistence modules of these filtered cubical complexes can be obtained using cubical homology in a similar manner to that of simplicial filtrations.

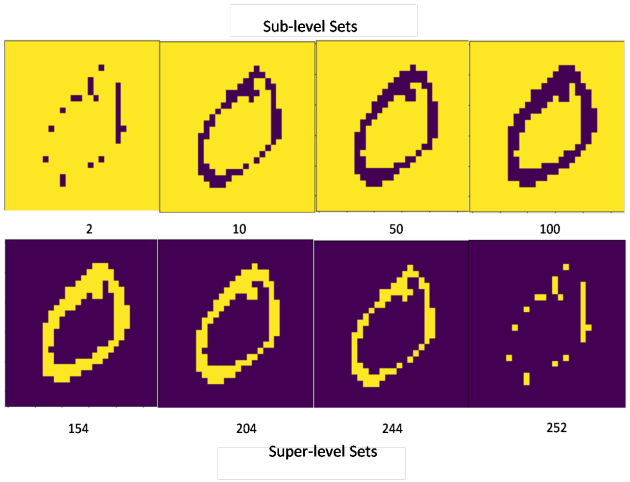


Figure 5.1: Cubical complexes (in purple) obtained using the grayscale function g. (Top) Sub-level sets $g^{-1}((-\infty, t])$ for $t = 2,\ 10,\ 50,\ 100$. (Bottom) Super-level sets $g^{-1}([t, \infty))$ for $t =\ 154,\ 204,\ 244,\ 252$.

As the grayscale filtration does not help in distinguishing between digits in the same homotopy class, other functions on the grid that take into account different aspects of the image's shape were also considered in the pipeline. These include the height, density and radial filtrations for different parameter values which were generated as described in [25] using the giotto-tda library in Python.

For obtaining these filtrations, the grayscale image is first binarised at a suitable threshold such that the digit corresponds to the 1-pixels. The following filtration functions are then defined on the grid, and the cubical complexes determined by its sub-level sets are constructed.

- Height : Each 1-pixel point on the grid is assigned the value of its distance from the hyperplane determined by a direction vector.
- Radial: Each 1-pixel point on the grid is assigned the value of its distance from a fixed grid point.
- Density: Each grid point is given the value of the number of 1-pixel points in a neighborhood of fixed radius around it. The radius values 5, 8 & 11 were used.

- Line : This function is a generalization of the height function present in the giotto-tda library. For any fixed line in $\mathbb{R}^2$, each 1-pixel grid point is given the value of its distance from the line.

- Grayscale and Inverted-grayscale : The pixel intensity map of the image and its photographic negative respectively.

A simplicial filtration of Vietoris-Rips complexes is also constructed by considering the 1-pixel points in the binarized image as a point cloud.

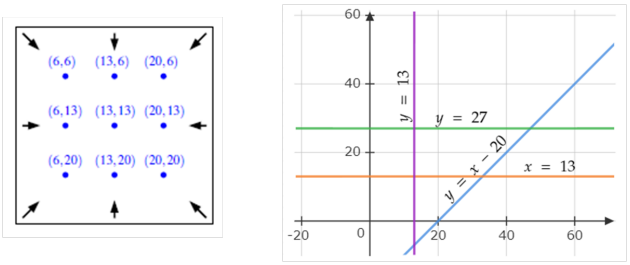


Figure 5.2: (Left) Directions and fixed grid points used in the pipeline for height and radial filtrations (Source: [25]). (Right) Lines used for the line filtrations in the pipeline.

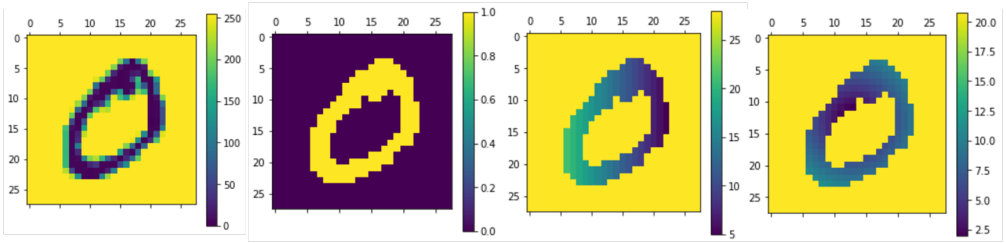


Figure 5.3: (Left-Right) Image; Binarized image; Height filtration with direction $(-1, 0)$; Radial filtration with center $(13, 13)$.

5.2 Topological Pipeline: Vectorisation

The topological information present in the persistence diagrams of these filtrations are encoded into feature vectors which then serve as input to machine learning algorithms. The vectorisations used on the diagrams are - persistent entropy, 2-norm of the persistence landscape, 2-norm of the Betti curve and 2^{nd}-Wasserstein amplitude. With this, all elements of the pipeline are now in place.

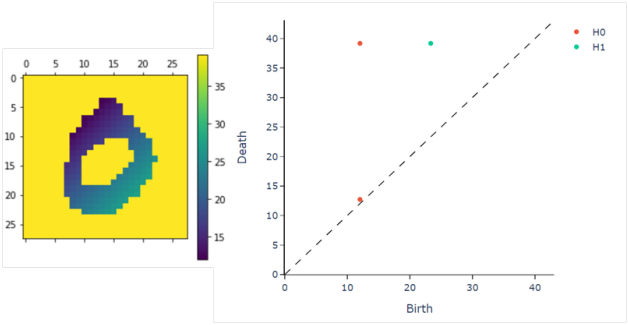


Figure 5.4: (Left-Right) Height filtration on image with direction $(-1, 1)$; Its persistence diagram.

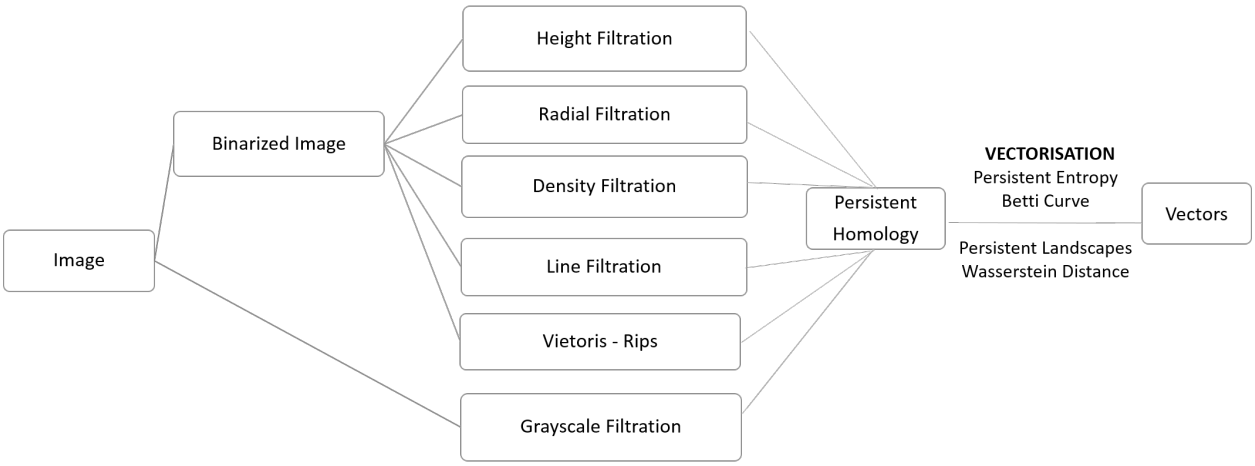


Figure 5.5: Schematic of the topological pipeline

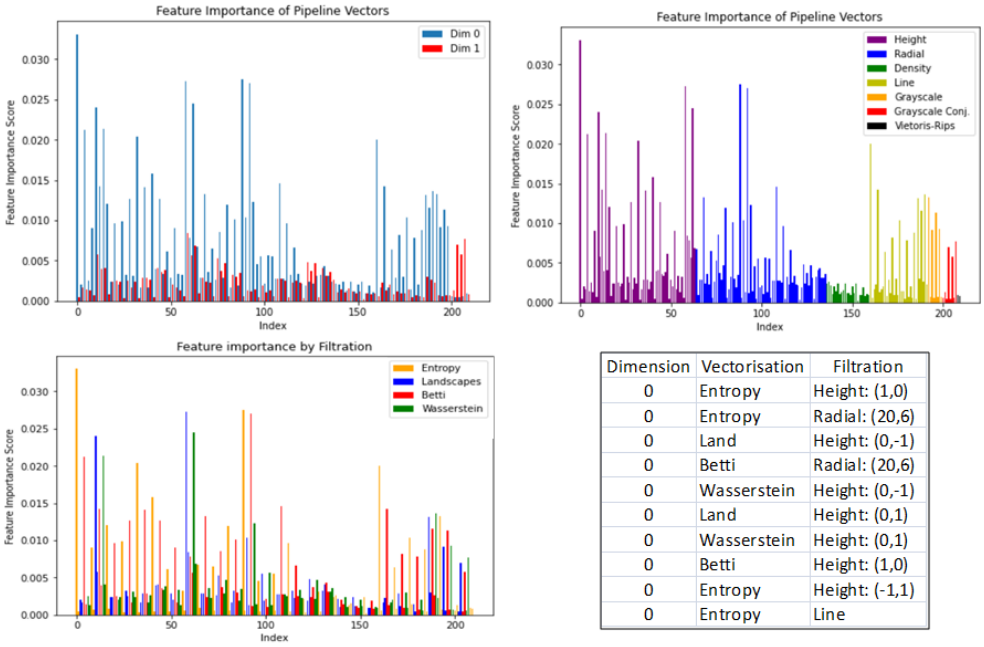
5.3 Analysing MNIST

The pipeline generated 210 features for each image. A random forest classifier with 1000 trees was used for both determining feature importance and for classification.

5.3.1 Feature Importance

The feature importance scores give a measure of how important a feature is in distinguishing between classes. The prominent filtrations by feature importance were height and radial

while by vectorisation it was persistence entropy. While dimension 1 features are useful in identifying the presence and position of the loops in the digits, the dimension 0 vectors cue in on how the digits are built. As evidenced by their feature importance scores, the dimension 0 vectors are quite effective in distinguishing between digits, especially ones in the same homotopy class.



Dimension	Vectorisation	Filtration
0	Entropy	Height: (1,0)
0	Entropy	Radial: (20,6)
0	Land	Height: (0,-1)
0	Betti	Radial: (20,6)
0	Wasserstein	Height: (0,-1)
0	Land	Height: (0,1)
0	Wasserstein	Height: (0,1)
0	Betti	Height: (1,0)
0	Entropy	Height: (-1,1)
0	Entropy	Line

Figure 5.6: Feature importance (using random forest classifier) of the pipeline vectors by dimension; filtration and vectorisation. (Bottom Left) Description of the 10 vectors with the highest feature importance scores.

5.3.2 Visualising MNIST Feature Vectors

Low dimensional projections of the feature vectors from the pipeline can be used to gain a better understanding of the topological properties of the images being captured by the it. UMAP [20] is used to generate the 2-dimensional projections of the features corresponding to the persistence entropy and betti curve vectorisation depicted in Figure 5.8. There are a few observations that can be made from these plots:

- In both these plots, points corresponding to digits of different homotopy types are farther apart while the clusters of digits of the same type are closer to each other.

Figure 5.7: Instances of varying styles and thickness of digits

- The clusters in the projection of entropy feature vectors are generally well separated and quite dense. There are also multiple clusters corresponding to the same digit. These correspond to different ways that a digit is written: for instance, 2 with or without a loop.
- In the projection of feature vectors obtained from betti curves, the clusters start out narrowly at one end and then later fan out. A gradation based on thickness of the digit and also the inclination angle can be observed within many digit clusters.

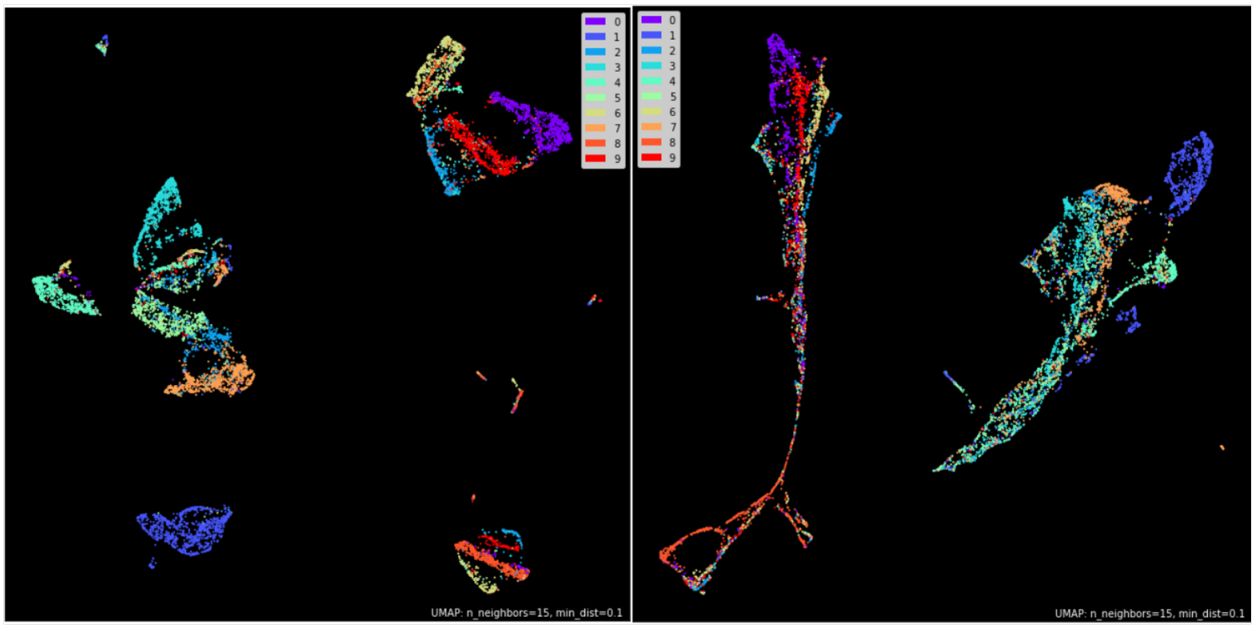


Figure 5.8: UMAP projection of (Left) entropy and (Right) Betti Curve feature vectors

5.3.3 Classification

Classification was performed using random forest classifier with 1000 trees. For a reference performance, all the pixel values except those that are 0 for all images were considered as a

vector. The dimension of the feature vector space can be reduced by dropping features with Pearson correlation coefficient more than 0.95. The accuracy for these is documented in the table 5.1 below.

Dimension	Description	Accuracy
703	Reference	96.3
210	Pipeline feature vectors	97.18
95	Uncorrelated pipeline feature vectors	97.05

Table 5.1: Classification Accuracy : MNIST

Figure 5.9.b is a plot of the accuracy for a varying number of features that are ranked in decreasing order of feature importance. An accuracy of over 96%, can be achieved just by considering the 50 most important features and this can be further increased by others.

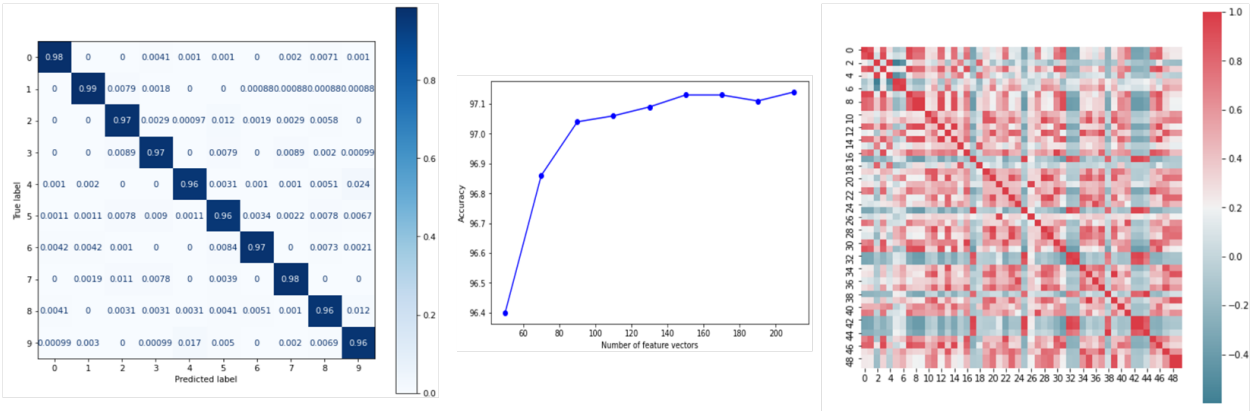


Figure 5.9: a. Normalized confusion matrix for pipeline vector classification; b. Classification accuracy plot for varying number of features; c. Correlation matrix for 50 most important features

In the previous sections, we have described a pipeline for extracting topological features from images and also looked at how they can be used for classification. It should be noted that this just offers a general framework and that the choice of filtration and vectorisation should be tailored to the dataset in question.

Dimension	Vectorisation	Accuracy
54	Entropy	96.02
52	Landscapes	94.94
52	Betti	95.84
52	Wasserstein	95.05

(a) Accuracy by Vectorisation

Dimension	Filtration	Accuracy
64	Height	96.33
72	Radial	95.45
24	Density	73.42
32	Line	91.6
8	Grayscale	36.19
8	Inverse Grayscale	35.69
2	Vietoris-Rips	33.19

(b) Accuracy by Filtration

Table 5.2: Accuracy of the MNIST pipleine features by vectorisation and filtration

5.4 Applications

5.4.1 Fashion MNIST

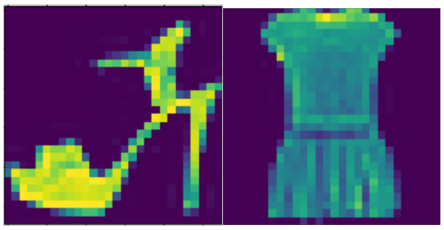


Figure 5.10: Sample images

The Fashion MNIST data set consists of 70,000 (60,000 training + 10,000 test) images of 10 classes of fashion apparel. The topological pipeline (excluding the Rips simplicial filtration) was used to generate 200 features which then served as input to a random forest classifier with 1000 trees. A classification accuracy of **85.02**% was achieved..

5.4.2 Flower Dataset

A subset of 7 categories of the 102 - Category Flowers dataset as shown in [6] was considered for classification. All 3 of the RGB filters were used to generate cubical complexes. Apart from this, the height and radial filtrations along with all four vectorisations were used to obtain 160 features. By dropping those features that had a Pearson correlation coefficient over 0.95, this number was reduced to 48. The classifi-

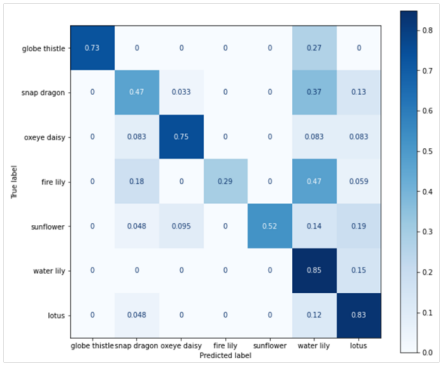


Figure 5.11: Correlation matrix

cation performed on a random 33-67 test-train split using a random forest classifier with 1000 trees gave anaccuracy of **67.75%**.

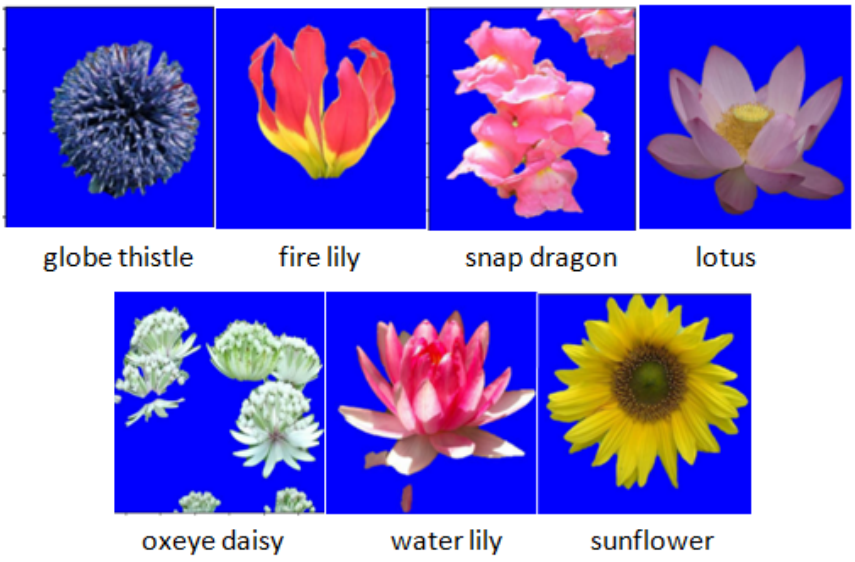


Figure 5.12: Flower categories considered for classification

5.4.3 Fundus Images

The High Resoultion Fundus image dataset contains 15 images each corresponding to healthy patients and also patients diagnosed with diabetic retinopathy. The images are converted to grayscale by considering a weighted sum of the RGB channels. Filtrations corrresponding to the grayscale and inverse grayscale maps along with persistence entropy and Wasserstein amplitude vectorisations were then used to generate 8 features. An estimate of the linear SVM classification accuracy: **86.67 ± 19.43%**, was then obtained using 5-fold cross validation.

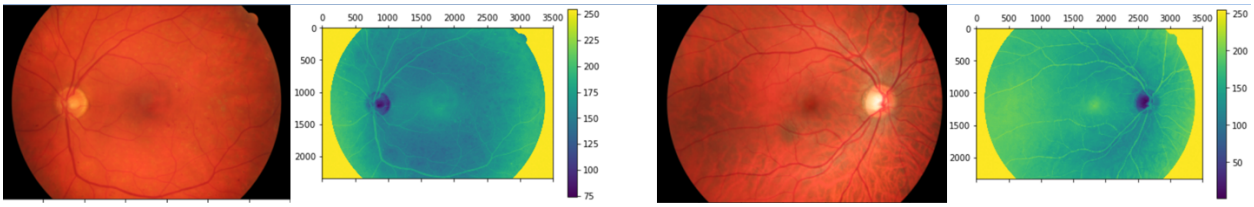


Figure 5.13: Fundus images and their corresponding grayscales (Left) Diabetic retinopathy (Right) Healthy

From these applications, it is clear that the features extracted from the pipeline capture sufficiently rich topological and shape properties of the image to facilitate classification. In fact, these topological features offer complementary information to those obtained from traditional machine learning algorithms and can be combined with the latter to boost accuracy. Work done in [16] and [22] in combining topological features with neural networks stands to support this claim.

In the next section, we shall analyse the robustness of the pipeline features under rotations and translations of the images and also look at a method for generating features which exhibit invariance to such transformations.

5.5 Rotational and Translational Invariance

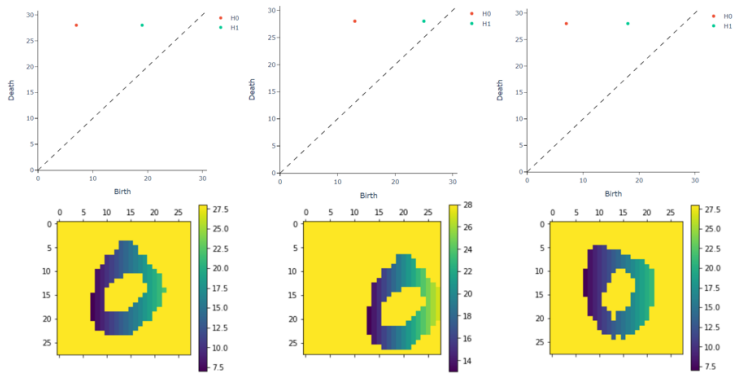


Figure 5.14: The persistence diagram corresponding to the height filtration (1,0) of an image under rotation and translation. (Left-Right) Unchanged, Translated by (6,3) and Rotated by 30° .

While the grayscale features are invariant under rotation and translation, the features obtained from other filtrations are not. This is because these filtrations in the pipeline capture how certain pixels, in the image are distributed relative to either the boundary, a line or fixed point. As a result, when an image is translated or rotated, the persistence diagrams corresponding to these filtrations change considerably. This is then reflected in a drop in accuracy when classification is performed on test data which has been rotated or translated.

For this analysis, we come back to the MNIST dataset. Of the 10000 MNIST test images,

3000 were modified using OpenCV to generate the new test data while the training data was left unchanged. Classification was performed on the pipeline features using random forest classifier with 1000 trees.

S.no	Description	Accuracy
1	Reference : Unchanged test	96.7
2	Translated by (-6,3)	60.17
3	Translated by 30° anticlockwise	80.94

Table 5.3: Classification accuracy using random forest on 3000 transformed MNIST test images

In the next section, we will look at a method to generate feature vectors that might be more robust to such transformations.

5.5.1 Simplicial Filtration generated from Thinned Images

Given an image, a 1-pixel wide skeleton was obtained by thinning. This was then used to generate a simplicial filtration with the skeleton points as vertices. This filtration was built by starting from a fixed point and iteratively adjoining the vertices neighbouring the ones added in the previous step along with an edge connecting both vertices. Figure 5.16 contains an example presenting the steps in this process.

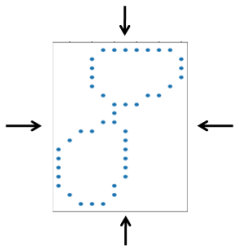


Figure 5.15: Directions

Persistent homology followed by vectorisation using persistent entropy, landscapes, Betti curves and Wasserstein amplitude were then considered for generating features. To address the question of choosing the initial fixed point, two approaches are considered:

1. The first vertex encountered along each of the 4 directions shown in Figure 5.15 is used to build the simplex streams and the features subsequently obtained are appended. This generates 40 features.

2. Computing an average of all the feature vectors obtained by considering all vertices in the graph as a fixed point for building the simplex stream. A total of 10 features are generated by this.

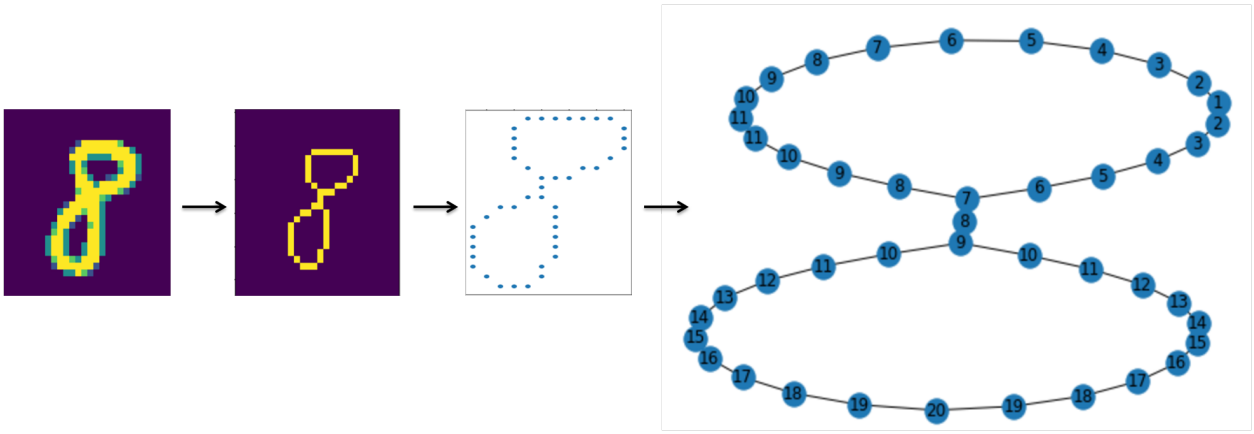


Figure 5.16: Simplicial filtratation obtained from using the 1-pixel skeleton of the image. The labelling of the vertices corresponds to the order of its appearance in the filtration.

Description	4-Dir Features	Avg Features	PL	PL+4-Dir	PL+Avg
Reference	62.03	55.4	96.4	96.27	96.47
Translation:(-6,3)	60.03	52.83	49.3	61.9	57.57
Translation:(6,3)	59.97	52.9	67.3	73.83	69.53
Rotation:$-30°$	49.47	47.43	80.6	81.5	80.83

Table 5.4: Classification accuracy under translation and rotation of test images. (PL: Uncorrelated pipeline features)

The features in the first approach capture the positional information in the image which might be lost while taking the average. On the other hand, as the initial vertex is chosen along fixed directions, the first method might be less robust to rotational transformations than the second.

The classification accuracy of the simplicial stream features with transformed test and unchanged training obtained using random forest (1000 trees) is documented below in Table 5.4. The robustness to transformations these features impart when combined with uncorrelated features (Pearson correlation coefficient cutoff of 0.95) from the topological pipeline is also studied.

- While the accuracy offered by these features are not very high, it is still greater than that of grayscale features. Also, the accuracies of the transformed images lie within a few percentages of the reference indicating some level of invariance.
- When appended to the feature vectors from the pipeline, these features can be seen

to increase the classification accuracy significantly for translation and marginally for rotation.

- Considering the extreme situation in use here, of unchanged training sets and significantly transformed test images, these improvements in accuracy suggest that these features are helpful in contributing to robustness under rotation and translation.

Chapter 6

Analysing Time Series Data

TDA has proven to be an effective tool for analyzing time series data. Methods from TDA have been successfully used to detect critical transitions in real-world dynamic systems, specifically climate models [9] and financial markets [17]. There is also evidence to suggest that it can be used to detect early warning signals that precede a market crash from financial time series data [19].

In this chapter, we shall explore how tools from dynamical systems theory can be combined with TDA to analyse time series data as discussed in [21].

Subsequently, these ideas were applied to analyse financial time series data, the results of which are presented in 6.2.

6.1 Dynamical Systems: Taken's Embedding Theorem

A dynamical system is a mathematical model of a time dependant process. The system is described completely by a state space and a set of rules which dictate how the states evolve with time.

Definition 6.1.1 (Dynamical System)**.** A *global continuous dynamical system* is given by the pair (M, Φ), where M is a topological space and $\Phi : \mathbb{R} \times M \to \mathbb{R}$ is a continuous function such that $\Phi(0, p) = p$ and $\Phi(t, \Phi(s, p)) = \Phi(t + s, p)$ for all $p \in M$ and $t, s \in \mathbb{R}$.

An attractor, A, of a dynamical system (M, Φ) is a subset of M to which most states evolve to over time. Mathematically, $A \subseteq M$, is said to be an attractor of (M, Φ) if it is compact, invariant under Φ and if it has an open basis of attraction.

Example (Lorenz System)**.** The Lorenz system is a dynamical system in $\mathbb{R}^3$ whose dynamics is given by the solution to the following system of differential equations.

Given $\sigma, \rho, \beta \in \mathbb{R}$

$$x'(t) = \sigma(y - x) \quad ; \quad y'(t) = x.(\rho - z) - y \quad ; \quad z'(t) = xy - \beta z.$$

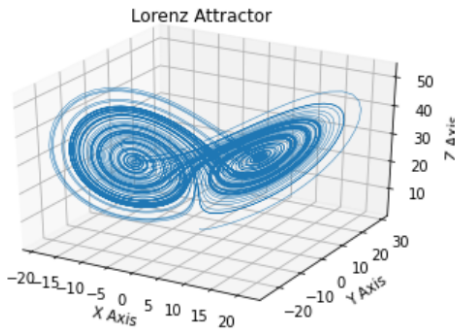


Figure 6.1: The attractor of the Lorenz system with $\sigma = 10$, $\beta = \frac{8}{3}$ and $\rho = 28$.

In practice, it is not often possible to determine the dynamical system completely, and all we have access to are observations of certain quantities for each state in the system. For instance, temperature and humidity in climate models or market indices in financial systems.

Let the map $F : M \to \mathbb{R}$ denote these observations. We can then define, for each state $p \in M$, a time series $\phi_p : \mathbb{R} \to \mathbb{R}$ where $\phi_p : t \mapsto F \circ \Phi(t, p)$.

Theorem 6.1.1 (Taken's Embedding)**.** Let M be a smooth compact Riemannian manifold. Let $\tau > 0$ and let $d \geq 2\dim(M)$ be an integer. If $\Phi \in C^2(\mathbb{R} \times M, M)$ and $F \in C^2(M, \mathbb{R})$ are generic, then the *delay map* $\phi : M \to \mathbb{R}^{d+1}$, where

$$\phi : p \mapsto (\phi_p(0), \phi_p(\tau), \ldots \phi_p(d\tau),$$

is an embedding. The function ϕ_p is as defined previously.

This theorem forms the basis for defining the sliding window embedding, which associates a point cloud with any time series.

Definition 6.1.2 (Sliding Window Embedding)**.** Given a function $f : \mathbb{R} \to \mathbb{R}$, a real number $\tau > 0$ and an integer $d > 0$, the sliding window embedding map $\mathbb{SW}_{d,\tau} f : \mathbb{R} \to \mathbb{R}^{d+1}$ is given by

$$\mathbb{SW}_{d,\tau} f : t \mapsto \big(f(t), f(t+\tau), \ldots, f(t+d\tau)\big).$$

Here, τ is the *time delay*, $d\tau$ the *window size* and $d+1$, the *embedding dimension*. For a subset $T \subseteq R$, the set

$$\{\mathbb{SW}_{d,\tau} f(t) \mid t \in T\}$$

is the the associated *sliding window point cloud*.

Given a time series representing observations from an abstract dynamical system, Taken's embedding theorem implies (under suitable conditions and parameter choices) that the sliding window embedding is a reconstruction of the system's attractor. TDA applied to this sliding window point cloud can hence be used to understand the topological properties of the dynamical system.

The following section will focus on applying these ideas to financial time series data for identifying market crashes.

6.2 Analysing Financial Data

Three stock market indices: S&P 500, Nasdaq Composite and Russell 2000, were considered for this exercise. The log returns of the adjusted stock prices of these indices from 12/1990 to 2/21 were used as the time series for constructing the sliding window point cloud.

6.2.1 Choosing Embedding Dimension & Time Delay

While the embedding theorem 6.1.1 guarantees the reconstruction with any choice of $\tau > 0$, this fails in practice as the time series data is often noisy or short. To determine an appropriate time delay, the autocorrelation function, which provides a measure of how similar

a time series is with delayed copies of itself, is used. The time delay was chosen as the delay value at which the autocorrelation function decays to 1/e.

The false nearest neighbours (FNN) method was used to determine the embedding dimension. This technique is based on the idea that points that are neighbours at a good choice of embedding dimension continue to remain so when the dimension increases. False neighbours refer to points that fail to be close to each other at higher values of embedding dimension. Increasing values of embedding dimension are considered, and the value m that ensures a sufficiently small number of false neighbours for the next dimension, $m + 1$, is chosen.

For all three market indices, the time delay and embedding dimension were determined to be 1 and 11 respectively, using the *NonlinearTseries* package in R.

6.2.2 Persistent Homology of Point Clouds

For every 50^{th} day in the considered time period, a sliding window point cloud was generated by considering T to be the next 75 days. In all, a sequence of 152 point clouds with 75 points each were built corresponding to each index. Persistent homology was then used to obtain the persistence diagrams of these point clouds.

Since we wish to understand how the topology of the system changes with time, for each point cloud in the sequence, the $2^{nd}-$Wasserstein distance of its persistence diagram from that of the point cloud preceding it is computed. The plots of the Wasserstein distances for all indices are shown below in Figure 6.2.

6.2.3 Analysis

The following observations can be made from Figure 6.2.

- The peaks in the dimension 0 plots of the Wasserstein distance appear to correspond to time periods around market crashes. The descriptions of the point clouds corresponding to the peaks is given in Table 6.1 and details of the relevant market crashes in this period in Table 6.2. No similar trends could be identified for the dimension 1 plots.

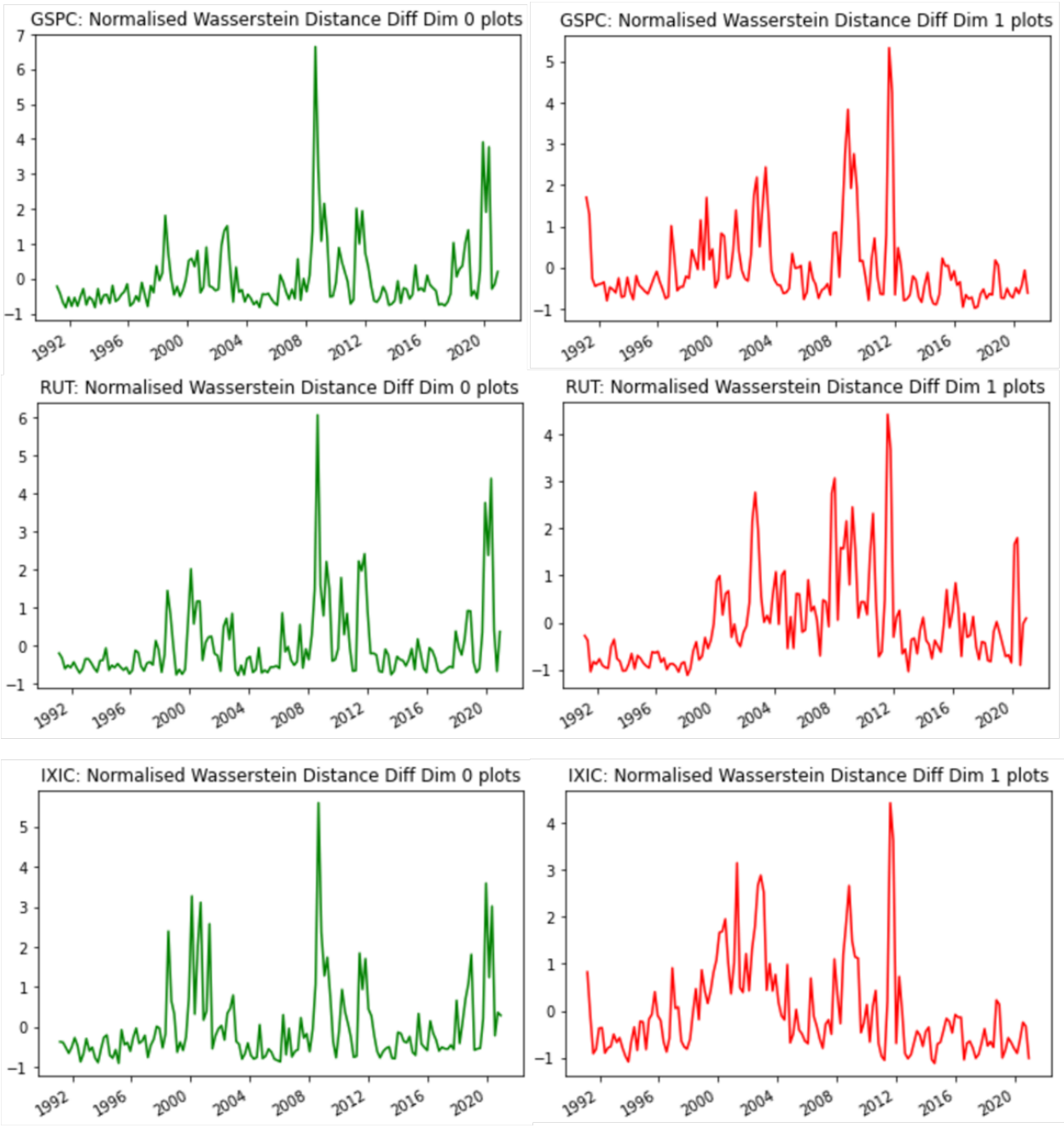


Figure 6.2: Wasserstein Distance Plots for Dimension 0 (Left) and 1 (Right) corresponding to (Top - Bottom) S&P 500 (GSPC) ; Nasdaq (IXIC) and Russell 2000 (RUT) indices.

- The first five peaks for all three market indices include the point clouds for periods around the 2007-2008 financial crisis and the 2020 market crash. The plot corresponding to the Russell 2000 index also has a peak corresponding to the Dotcom bubble crash.

- The time frames 14-01-2009 to 04-05-2009 and 26-06-2000 to 11-10-2000 which present as peaks, immediately follow the 2008 market crash and the Dotcom bubble respectively. The peaks at these periods can be attributed to changes in the system's topology when recovering from a crash.

- On the other hand, the time frame 07-10-2019 to 24-01-2020, which corresponds to a period before the 2020 stock market crash, is a peak in all three indices. This can potentially be due to some early topological changes in the system predating a crash as described in [19].

S.no	S & P 500	Nasdaq	Russell 2000
1	11-06-2008 to 26-09-2008	11-06-2008 to 26-09-2008	11-06-2008 to 26-09-2008
2	07-10-2019 to 24-01-2020	02-03-2020 to 17-06-2020	07-10-2019 to 24-01-2020
3	02-03-2020 to 17-06-2020	07-10-2019 to 24-01-2020	19-11-1999 to 09-03-2000
4	21-08-2008 to 08-12-2008	12-08-2011 to 29-11-2011	26-06-2000 to 11-10-2000
5	14-01-2009 to 04-05-2009	17-12-2019 to 06-04-2020	02-03-2020 to 17-06-2020

Table 6.1: Time frame of point clouds corresponding to first 5 peaks in the dimension 0 Wasserstein distance plots.

Date	Description
24-02-2020 to 7-04-2020	2020 Stock market crash
16-09-2008	2007-08 financial crisis
10-03-2000	Dot-com bubble
1-09-2011	August '11 stock market fall

Table 6.2: Relevant market crashes between 1990 and 2021

The purpose of this exercise was to showcase the potential of TDA as a tool for time series analysis. Here, we had segmented the time series to obtain 152 point clouds with 75 points each. By varying these numbers or the vectorisation method used, one can potentially gain more insight from the data.

Chapter 7

Conclusion

In conclusion, we have looked at some theoretical aspects of TDA and studied its application to image classification and time series analysis.

The topological pipeline for MNIST presented in Chapter 6 uncovered finer information from the images, based on the writing styles, than required for classification. We have also looked at how TDA can be used to extract features from images that are robust to translation and rotation. In Chapter 7, we saw that persistent homology proved successful in identifying periods of market crashes from stock market index data. The range of these applications speaks to the versatility of these methods.

TDA is a relatively new field with active research being undertaken in different areas. Hopefully, this thesis has provided the reader with enough justification that it offers promising tools for data analysis with broad scopes of application.

Bibliography

[1] James R. Munkres. *Elements of Algebraic Topology*. Addison Wesley Publishing Company, 1984. ISBN: 0201045869.

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

[3] Afra Zomorodian and Gunnar Carlsson. "Computing Persistent Homology". In: *Discrete Comput. Geom.* 33.2 (Feb. 2005), pp. 249–274. ISSN: 0179-5376.

[4] Herbert Edelsbrunner and John Harer. "Persistent homology-a survey". In: (2008).

[5] Robert Ghrist. "Barcodes: The persistent topology of data". In: *BULLETIN (New Series) OF THE AMERICAN MATHEMATICAL SOCIETY* 45 (Feb. 2008). DOI: 10.1090/S0273-0979-07-01191-3.

[6] Maria-Elena Nilsback and Andrew Zisserman. "Automated Flower Classification over a Large Number of Classes". In: *Indian Conference on Computer Vision, Graphics and Image Processing*. Dec. 2008.

[7] Gunnar Carlsson. "Topology and Data". In: *Bulletin of The American Mathematical Society - BULL AMER MATH SOC* 46 (Apr. 2009), pp. 255–308. DOI: 10.1090/S0273-0979-09-01249-X.

[8] Frédéric Chazal, David Cohen-Steiner, and Quentin Mérigot. "Geometric Inference for Probability Measures". In: *Foundations of Computational Mathematics* 11 (Dec. 2011), pp. 733–751. DOI: 10.1007/s10208-011-9098-0.

[9] Jesse Berwald, Marian Gidea, and Mikael Vejdemo-Johansson. "Automatic recognition and tagging of topologically different regimes in dynamical systems". In: *CoRR* abs/1312.2482 (2013). arXiv: 1312.2482. URL: http://arxiv.org/abs/1312.2482.

[10] Jan Odstrcilik et al. "Retinal vessel segmentation by improved matched filtering: Evaluation on a new high-resolution fundus image database". In: *Image Processing, IET* 7 (June 2013), pp. 373–383. DOI: 10.1049/iet-ipr.2012.0455.

[11] Peter Bubenik. "Statistical Topological Data Analysis using Persistence Landscapes". In: *Journal of Machine Learning Research* 16.3 (2015), pp. 77–102. URL: http://jmlr.org/papers/v16/bubenik15a.html.

[12] Harish Chintakunta et al. "An entropy-based persistence barcode". In: *Pattern Recognition* 48.2 (2015), pp. 391–401. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2014.06.023. URL: https://www.sciencedirect.com/science/article/pii/S0031320314002453.

[13] Anna Krakovská, Kristína Mezeiová, and Hana Budáčová. "Use of False Nearest Neighbours for Selecting Variables and Embedding Parameters for State Space Reconstruction". In: *Journal of Complex Systems* 2015 (Mar. 2015), p. 12. DOI: 10.1155/2015/932750.

[14] Henry Adams et al. "Persistence images: A stable vector representation of persistent homology". In: *Journal of Machine Learning Research* 18 (2017).

[15] Frédéric Chazal and Bertrand Michel. "An introduction to topological data analysis: fundamental and practical aspects for data scientists". In: *arXiv preprint arXiv:1710.04019* (2017).

[16] Tamal Krishna Dey, Sayan Mandal, and William Varcho. "Improved Image Classification using Topological Persistence". In: *22nd International Symposium on Vision, Modeling, and Visualization, VMV 2017, Bonn, Germany, September 25-27, 2017*. Eurographics Association, 2017, pp. 161–168. DOI: 10.2312/vmv.20171272. URL: https://doi.org/10.2312/vmv.20171272.

[17] Marian Gidea. "Topological data analysis of critical transitions in financial networks". In: *International Conference and School on Network Science*. Springer. 2017, pp. 47–59.

[18] Han Xiao, Kashif Rasul, and Roland Vollgraf. *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*. 2017. arXiv: 1708.07747 [cs.LG].

[19] Marian Gidea and Yuri Katz. "Topological data analysis of financial time series: Landscapes of crashes". In: *Physica A: Statistical Mechanics and its Applications* 491 (2018), pp. 820–834. ISSN: 0378-4371. DOI: https://doi.org/10.1016/j.physa.2017.09.028. URL: https://www.sciencedirect.com/science/article/pii/S0378437117309202.

[20] Leland McInnes, John Healy, and James Melville. "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction". In: *arXiv e-prints*, arXiv:1802.03426 (Feb. 2018), arXiv:1802.03426. arXiv: 1802.03426 [stat.ML].

[21] Jose A. Perea. "Topological Time Series Analysis". In: *arXiv e-prints*, arXiv:1812.05143 (Nov. 2018), arXiv:1812.05143. arXiv: 1812.05143 [math.AT].

[22] Yu-Min Chung et al. "TopoResNet: A hybrid deep learning architecture and its application to skin lesion classification". In: *CoRR* abs/1905.08607 (2019). arXiv: 1905.08607. URL: http://arxiv.org/abs/1905.08607.

[23] Nieves Atienza, Rocio Gonzalez-Díaz, and Manuel Soriano-Trigueros. "On the stability of persistent entropy and new summary functions for topological data analysis". In: *Pattern Recognition* 107 (2020), p. 107509. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2020.107509. URL: https://www.sciencedirect.com/science/article/pii/S0031320320303125.

[24] Bastian Rieck, Filip Sadlo, and Heike Leitte. "Topological Machine Learning with Persistence Indicator Functions". In: *Topological Methods in Data Analysis and Visualization V* (2020), pp. 87–101. ISSN: 2197-666X. DOI: 10.1007/978-3-030-43036-8_6. URL: http://dx.doi.org/10.1007/978-3-030-43036-8_6.

[25] Guillaume Tauzin et al. "giotto-tda: A Topological Data Analysis Toolkit for Machine Learning and Data Exploration". In: *CoRR* abs/2004.02551 (2020). arXiv: 2004.02551. URL: https://arxiv.org/abs/2004.02551.