UNIVERSITAT DE BARCELONA

ADVANCED MATHEMATICS
MASTER'S FINAL PROJECT

Time-Dependent Topological Analysis for Cardiovascular Disease Diagnosis using Magnetic Resonance

---

Author:
Juan David Prada Malagon

Supervisor:
Carles Casacuberta

Supervisor:
Polyxeni Gkontra

Facultat de Matemàtiques i Informàtica

September, 2023

Juan David Prada Malagon
pradamalagonjd@gmail.com
+(34) 691 147 108
+(57) 312 358 93 40

Abstract

The present research project aims to study the topology of time varying Cardiovascular Magnetic Resonance images (CMR) for disease diagnosis. CMR is a non-invasive technique that involves the acquisition of multiple 3D images at different cardiac phases throughout the cardiac cycle. Nonetheless, conventional assessment of CMR images typically involves the quantification of parameters related to the volumes, and more recently to the shape and texture by means of radiomics (Raisi-Estabragh, 2020), of the cardiac chambers at only two static time-point points: the end-systole and the end-diastole. Therefore, potentially rich information regarding the cardiac function and structure from other phases of the cardiac cycle might be lost.

To overcome this limitation, we propose to leverage Topological Data Analysis (TDA) to optimally exploit information from the entire cardiac cycle, by measuring the variation of persistence descriptors. This approach seems promising since a time series might not exhibit relevant geometrical features in its respective point cloud embedding, but it may rather display topological cyclic patterns and their respective variations that can be captured with the proposed machinery. Subsequently, the novel TDA-based CMR descriptors encompassing the entire cardiac cycle are used to feed supervised machine learning classifiers for cardiovascular disease diagnosis.

A full framework from data gathering, to image processing, mathematical modelling and classifier implementation is presented for this purpose. The performance of the proposed approach based on TDA features and ML is limited. Nonetheless, the approach could be easily adapted to other diseases and scenario where the integration of ML and TDA could be more beneficial.

Contents

Chapter I

Introduction

I.1 Motivation

According to the World Health Organisation (WHO), cardiovascular disease (CVD) remains the major cause of death. CVD is an overarching term for diseases of the heart (heart disease) and vascular diseases (concerning blood vessels). There are a number of factors which increase the risk of CVD, from lifestyle habits such as smoking or physical inactivity to genetics.

I.2 Research Outline

The present research work aims to study the topology of time varying Cardiovascular Magnetic Resonance images (CMR) for disease diagnosis. CMR is a non-invasive technique that involves the acquisition of multiple 3D images at different cardiac phases throughout the cardiac cycle. Nonetheless, conventional assessment of CMR images typically involves the quantification of parameters related to the volumes, and more recently to the shape and texture by means of radiomics [15], of the cardiac chambers at only two static time-point points: the end-systole and the end-diastole. Therefore, potentially rich information regarding the cardiac function and structure from other phases of the cardiac cycle might be lost. To overcome this limitation, we propose to leverage Topological Data Analysis to optimally exploit information from the entire cardiac cycle, by measuring the variation of persistence descriptors. This approach seems promising since a time series might not exhibit relevant geometrical features in its respective point cloud embedding, it may rather display topological cyclic patterns and their respective variations that can be captured with the proposed machinery. For this same purpose it will be determined if TDA enhances a classifier based on radiomics features. The ultimate goal of this work is to develop and validate a novel approach for the diagnosis of cardiac diseases from CMR.

I.2.1 General Objectives

- Evaluate whether TDA descriptors considering all time points of the cardiac cycle outperform TDA descriptors extracted only from the end-systole and end-diastole as predictors in ML models of CVD diagnosis.
- Measure the variation of persistence descriptors of CMR images.
- Design and implement a machine learning-based pipeline using TDA features as predictors that can adequately characterise the texture of CMR images.

I.2.2 Specific Objectives

- Model the cardiac cycle using persistence vineyards with appropriate time-series descriptors.
- Determine if TDA enhances the performance of a classifier based on radiomics features.

I.3 Context

I.4 Topology & Data Science

Today data is being generated in an unprecedented rate in a wide spectrum of formats and in all the areas of mankind wisdom. A couple of problems in the field are worth to be mentioned: it is a frequent scenario to have high-dimensional, noisy data or to have a bigger proportion of missing data or even a worse problem is in the fact that the ability for analyzing data is a step back from the pace of data production.

For the latter problems described, geometry and topology are good candidates to approach a feasible solution since we deal with the study of distance functions on finite data sets, in short point clouds in the context of topology.

The methodologies based on topology are powerful since they deal with the extraction of qualitative geometric information by means of computing topological characteristics such as connected components, loops or cavities in space as a tool of qualitative study.

On the other hand, it does not depend on the choice of metric, but it rather generalizes the concept of distance between points. Not being sufficient with the latter arguments, the interpretation tools from topology are delivered by summaries rather than from parameter choices and provide all the information from invariants at once without loss of generality nor loss of information. In this manner, a big portion of the information from

topological spaces can be obtained by means of discrete diagram sets using simplicial approximation.

The cornerstone of this theory is functorality, the common point of most of the mathematical applications and the basis of algebraic topology as well, since it allows one to compute the homological invariants from local information. Note that this idea relies on the intention of understanding the relationships between geometric objects constructed from data: in this particular case of topological spaces we are talking about continous maps, while in the case of the category of groups they are homomorphisms. In this line of reasoning, it can be stated at this stage that topological data analysis aims to define functorial geometric constructions for analyzing behavior on maps to get insights about point clouds.

Chapter II

Elements of Topological Data Analysis

II.1 Simplicial Complexes

To get a first taste of TDA one should start by thinking of an arbitrary data set from which a point cloud can be constructed and a filtering function can be defined. A filtered simplicial complex is constructed and by means of applying persistent homology to it we get topological descriptors.

Definition II.1.1 (Abstract Simplicial Complex)**.** An abstract simplicial complex with vertex set $V = \{v_i\}_{i \in I}$ is a collection K of nonempty finite subsets $\{v_0, \ldots, v_n\} \subseteq V$ such that:

I $\{v\} \in K \quad \forall v \in V$

II If $S \in K$ and $S' \subseteq S$ with $S' \neq \phi$, then $S' \in K$.

If the index set I of V is equipped with a total order, then K is called ordered.

Definition II.1.2 (n-faces)**.** The elements of K are called faces. For $n \geqslant 0$, a face $\{v_0, \ldots, v_n\}$ of cardinality $n+1$ is an n-face. The set of 0 -faces is in bijective correspondence with V and its elements are called vertices. The 1-faces are called edges.

Definition II.1.3 (d-skeleton)**.** The collection of all n-faces of K for $0 \leq n \leq d$ is an abstract simplicial complex for every $d \geqslant 0$, called the d-skeleton of k. The 1 -skeleton of K is an undirected graph.

Note that every abstract simplicial complex is determined by its maximal faces, that is, faces not contained in any larger one.

Definition II.1.4 (n-simplex)**.** An n-simplex of $\mathbb{R}^N$ is the convex hull of $n+1$ affinely independent points $p_0, \ldots, p_n$ in $\mathbb{R}^N$:

$$\Delta\left(p_0, \ldots, p_n\right) = \left\{x_0 p_0 + \cdots + x_n p_n \in \mathbb{R}^N \mid x_0 + \cdots + x_n = 1, x_i \geqslant 0 \ \ \forall i\right\}. \tag{II.1}$$

Intuitively speaking, every subset $\{p_{i_0}, \ldots, p_{i_k}\} \subseteq \{p_0, \ldots, p_n\}$ spans a k-simplex $\Delta\left(p_{i_0}, \ldots, p_{i_k}\right)$, which is called a k-face of $\Delta\left(p_0, \ldots, p_n\right)$.

Definition II.1.5 (Standard n-simplex)**.** The standard n-simplex Δ^n is the convex hull of the coordinate unit points in $\mathbb{R}^{n+1}$:

$$\Delta^n := \Delta(e_1, ..., e_{n+1}) \ , \ e_i = (0, ..., \overset{i}{1}, ..., 0). \tag{II.2}$$

Definition II.1.6 (Geometrical Simplicial Complex)**.** A geometric simplicial complex is a set X of simplices $\sigma \subset \mathbb{R}^N$ for some $N \in \mathbb{N}$ such that

I Every face of X is in X.

II Any two simplices in X are either disjoint or intersect in one common face.

From the last definition one can guess that for a finite dimensional geometrical simplicial complex, its dimension is the maximum of the dimensions of its simplices.

Note that this geometrical notion of a simplical complex has on its backstage a topological space $|X| = \bigcup_{\sigma \in X} \sigma$ endowed with the Euclidean topology and therefore it is a polyhedron.

Definition II.1.7 (Geometric Realization)**.** Let K be a finite ordered abstract simplicial complex with vertex set $V = \{v_1, \ldots, v_n\}$. The geometric realization of K is the geometric simplicial complex X_K with a k-face

$$\Delta\left(e_{i_0}, \ldots, e_{i_k}\right) \in \mathbb{R}^n \ \ \forall\left(i_0 \ldots i_k\right) \ \ \textit{a k-face of k.} \tag{II.3}$$

For ease of notation, denote the topological spaces of the geometric realization of K by $|X_K| = |K|$.

Definition II.1.8 (Triangulation)**.** If $|K| \cong Y$ then K is called a triangulation of a topological space Y.

One can think of simplicial complexes as space with a triangulation.

In this line of reasoning every abstract simplicial complex K_X determines a geometric simplicial complex X and the converse also holds. Note that any geometric simplicial complex X determines the abstract simplicial complex K_X whose set of vertices are the 0-faces of X and with elements $\{v_{i0}, ..., v_{in}\}$ for each n-simplex $\Delta(v_{i0}, ..., v_{in})$ of X. The order of K_X is inherited by X.

There are face-preserving bijective correspondences

$$X_{K_X} \cong X \text{ and } \quad K_{X_K} \cong K \tag{II.4}$$

which induce homeomorphisms $|X_{K_X}| \cong |X|$ and $|K_{X_k}| \cong |K|$. They can be polyhedra in different ambient spaces.

Definition II.1.9 (Graph (TDA))**.** In the context of TDA a graph is a 1-dimensional abstract simplicial complex. It is called a clique if it is fully connected, that is, if every pair of vertices is joined by an edge. And a weighted graph is a graph with a real number $w \geqslant 0$ attached to each of its edges.

Definition II.1.10 (Point Cloud)**.** A point cloud is a finite set of points $X = \{x_i\}_{i \in I}$ in $\mathbb{R}^N$ with $N \geqslant 2$.

Just for the sake of illustration let define two popular simplicial complexes.

Definition II.1.11 (Czech Complexes)**.** Let X be a point cloud and fix $\varepsilon \geqslant 0$. Denote the abstract simplicial complex $C_\varepsilon(X)$ named the Czech complex such that it has vertices in the set X and a k-face for each collection $x_{i_0}, \ldots, x_{i_k}$ such that

$$\bar{B}_{\varepsilon/2}(x_{i_0}) \cap \ldots \cap \bar{B}_{\varepsilon/2}(x_{i_k}) \neq \varnothing. \tag{II.5}$$

Definition II.1.12 (Vietoris-Rips Complexes)**.** Denote the Vietoris-Rips abstract simplicial complex by $R_\varepsilon(X)$ with vertex set X and a k-face for each collection $x_{i_0}, \ldots, x_{i_k}$ such that $\|x_{i_r} - x_{is}\| \leqslant \varepsilon \quad \forall r, s \in \{0, \ldots, k\}$:

$$R_\varepsilon(X) \textit{ has a k-face } \{x_{i_0}, \ldots, x_{ik}\} \iff \operatorname{diam}(\{x_{i0}, \ldots, x_{i_k}\}) \leq \varepsilon \tag{II.6}$$

where $\operatorname{diam}(A) = \sup\{\|a - b\| : a, b \in A\}$.

II.2 Cubical Complexes

Cubical complexes deserve a section since later in the methodology it will be clear usefulness for image processing and mathematical modelling to obtain topological features of the images.

Consider a grid with cells of all dimensions. The cubical complex aims to represent this mathematical structure and hence it is a key tool for computational mathematics. We start with some definitions.

Definition II.2.1 (Elementary Intervals)**.** A non-degenerate elementary interval is of the form $[n, n+1]$ and a degenerate interval is of the form $[n, n]$ $\forall n \in \mathbb{Z}$.

Definition II.2.2 (Boundary of Elementary Interval)**.** A boundary of an elementary interval is $\partial[n, n] = 0$ in the degenerate case, and in the non-degenerate case it is a chain of the form

$$\partial[n, n+1] = [n+1, n+1] - [n, n]. \tag{II.7}$$

Definition II.2.3 (Elementary Cube)**.** An elementary cube is a product of elementary intervals

$$C = I_1 \times \cdots \times I_n \tag{II.8}$$

The embedding dimension of an elementary cube n is associated to the number of elementary intervals without discriminating on degenerate or not. The dimension of a cube is the number of non degenerate elementary intervals in the product.

Definition II.2.4 (Boundary of a Cube)**.** A boundary of an elementary cube follows the formula,

$$\partial C := (\partial I_1 \times \cdots \times I_n) + (I_1 \times \partial I_2 \times \cdots \times I_n) + \cdots + (I_1 \times I_2 \times \cdots \times \partial I_n) \tag{II.9}$$

Definition II.2.5 (Cubical Complex)**.** A cubical complex $\mathcal{K}$ is a collection of cubes closed under the operation of taking boundaries.

Definition II.2.6 (Maximal Cube)**.** A cube C in a cubical complex $\mathcal{K}$ is maximal if it is not in a boundary of any other cube in $\mathcal{K}$.

Definition II.2.7 (Filtered Cubical Complex)**.** Filtered cubical complexes are cubical complexes whose cubes are equipped with filtration values.

The latter case of filtered cubical complexes is the one that concerns us. Images are modeled as cubical complexes since the cubical structure is inherited canonically from the pixel structure and their filtration values are the pixels intensities associated.

Definition II.2.8. A support of a cube C is the set in $\mathbb{R}^n$ occupied by C, where n is the embedding dimension of C.

At this stage one can think about the applications to image analysis as follows. Consider a rectangular region in $\mathbb{R}^n$. The cubical gives a representation on the embedding space. With this machinery we can store cubical complexes in bitmap form. Let $\mathcal{K}$ be a cubical complex and let $R = [b_1, d_1] \times \cdots \times [b_n, d_n]$ where $b_1, ..., b_n, d_1, ..., d_n \in \mathbb{Z}$ and $b_i \leqslant d_i \ \forall i$ be a rectangular region that is the support of $\mathcal{K}$. The ordering described is the base of the bitmap-based implementation where the cubical complex is stored as vectors of the filtrations, and moreover describes all cubes in $\mathcal{K}$.

II.3 Simplicial Homology

Let K be a finite ordered abstract simplicial complex with set of vertices $V = \{v_1, \ldots, v_N\}$. In this section the notation will have an slight change: write $(i_0 \ldots i_n)$ instead of $\{v_{i_0}, \ldots, v_{i_n}\}$ such that the order is strict among the indexes $i_0 < \ldots < i_n$. If it is the case that the indexes $i_0, \ldots, i_n$ are not in order denote the even and odd permutations by

$$(i_0 \cdots i_n) = \varepsilon(\sigma) \left(i_{\sigma(0)} \cdots i_{\sigma(n)}\right) \text{ where } i_{\sigma(0)} < \ldots < i_{\sigma(n)} \tag{II.10}$$

and $\varepsilon(\sigma) = 1$ if σ is an even permutation while $\varepsilon(\sigma) = -1$ if σ is odd.

Let $C_n(K)$ for $n \geqslant 0$ be the free abelian group on the set of n-faces of K. Therefore the elements of $C_n(K)$ are called n-chains in K. For instance $C_n(k) = 0$ if $n < 0$ or if the set of n-faces of K is empty, since the free abelian group on $\emptyset$ is the trivial group.

Let R be a field, so that an R-module is a vector space over R. The choice of the field can be anyone of $\mathbb{Q}, \mathbb{R}, \mathbb{C}$, $\mathbb{F}_2$, ..., but in this context think particularly in $R = \mathbb{Z}$ with a further generalization to the real field $\mathbb{R}$.

Definition II.3.1 (Boundary Operator)**.** The n-th boundary is the group homomorphism

$$\partial_n : C_n(K) \longrightarrow C_{n-1}(K) \tag{II.11}$$

which is defined on the space of generators as

$$\partial_n \left(i_0 \ldots i_n\right) = \sum_{k=0}^{n} (-1)^k \left(i_0 \ldots \hat{i}_k \ldots i_n\right) \tag{II.12}$$

where $\hat{i}_k$ means that the kth entry is deleted.

Theorem 1 (Fundamental Property of Boundary Operators)**.**

$$\partial_n \circ \partial_{n+1} = 0 \quad \forall n. \tag{II.13}$$

Consider the following remark.

Proposition 1.

$$\partial_n \circ \partial_{n+1} = 0 \Rightarrow \operatorname{Im} \partial_{n+1} \subseteq \operatorname{Ker} \partial_n. \tag{II.14}$$

The latter remark means that $\forall n \in \mathbb{N}$

$$C_{n+1}(K) \underbrace{\xrightarrow{\partial_{n+1}} C_n(K) \xrightarrow{\partial_n}}_{0} C_{n-1}(K) \tag{II.15}$$

Definition II.3.2 (n-th Homology of K)**.** The n-th homology of K is defined by

$$H_n(K) = \frac{Z_n(K)}{B_n(K)} \tag{II.16}$$

where $Z_n(K) = \operatorname{Ker} \partial_n$ and $B_n(K) = \operatorname{Im} \partial_{n+1}$

Note that $B_n(K) \subseteq Z_n(K)$ $\forall n$, makes sense since the elements of $Z_n(K)$ and $B_n(K)$ are n-cycles and n-boundaries, respectively. Therefore the definition of the n-th homology of K takes the space of n-cycles and makes a quotient by their n-boundaries so that two n-cycles are equivalent modulo a boundary.

It is important to mention at this stage some subtleties about the structure of $H_n(K, R)$. Note that if R is a ring $Z_n(K, R)$ is an R-submodule of $C_n(K, R)$ and $B_n(K, R)$ is an R-submodule of $Z_n(K, R)$. Therefore $H_n(K, R)$ inherits an R-module structure. In this line of reasoning, if R is a field then $H_n(K, R)$ is an R-vector space.

In particular, if $R = \mathbb{Z}$ then $Z_n(K)$ and $B_n(K)$ are free abelian groups, since $C_n(K)$ is free and every subgroup of a free abelian group is free.

If K is finite, then $Z_n(K)$ is finitely generated and so is $H_n(K)$ and therefore it can be decomposed as

$$H_n(K) \cong \mathbb{Z}^r \oplus \mathbb{Z}/p_1^{\alpha_1} \oplus \ldots \oplus \mathbb{Z}/p_m^{\alpha_m} \tag{II.17}$$

for some primes $p_1, \ldots, p_m$ and $\alpha_i \geqslant 1$ and where $r = \operatorname{rank} H_n(K) = \dim_{\mathbb{Q}} H_n(K; \mathbb{Q})$.

More generally, if R is a principal ideal domain then $Z_n(K)$ is finitely generated and free. According to the structure theorem for finitely generated modules over a principal ideal domain we have the following isomomorphism:

$$H_n(K;R) \cong R/(d_1) \oplus \ldots \oplus R/(d_m) \tag{II.18}$$

for ideals (d_i) of R with $(d_{i+1}) \leq (d_i)$ for all i, where indices with $d_i = 0$ correspond to the free part, so if R is a field, then $d_i = 0$ for all i, since all R-modules are free. In general, the ideals (d_i) are uniquely determined by $H_n(K;R)$, but the isomorphism is not unique.

By computing by hand the homology groups with coefficients in $\mathbb{Z}$. Furthermore, some insight can be gained through connecting the computations of homology groups with Betti numbers and the Euler characteristic.

Definition II.3.3 (Betti Numbers)**.** Let K be a finite ordered abstract simplicial complex and let $\mathbb{F}$ be a field. Define the Betti numbers of K by

$$\beta_n(\mathbb{F}) = \operatorname{rank} H_n(\mathbb{F}) = \dim_{\mathbb{F}} H_n(K;\mathbb{F}) \textit{ for } n \geqslant 0. \tag{II.19}$$

Definition II.3.4 (Euler Characteristic)**.** Let K be a finite ordered abstract simplicial complex and let $\mathbb{F}$ be a field. The Euler characteristic is defined by

$$\chi(K) = \sum_{n=0}^{\infty} (-1)^n \beta_n(K;\mathbb{F}) \tag{II.20}$$

It can be proved that the Euler characteristic does not depend on the field $\mathbb{F}$ but will not be done here.

Instead what concerns us at this point is to note that for an abstract simplicial complex K when computing its Euler characteristic for each of their n-skeleta we get that the homology groups counts specific topological features in K, being more specific:

- $H_0(K)$ counts connected components of K;
- $H_1(K)$ counts 1-dimensional cycles in K;
- $H_2(K)$ counts 2-dimensional cavities in K.

II.4 Persistent Homology

At this stage the discussion continues towards mappings between simplicial complexes, that is the functiorality of homology.

Definition II.4.1 (Simplicial Map)**.** A function $f : K \to L$ between abstract simplicial complexes is a simplicial map if it sends vertices of K to vertices of L and $f(v_{i_0}), \ldots, f(v_{i_n})$ form a face in L whenever $\{v_{i_0}, \ldots, v_{i_n}\}$ is a face of K.

Every simplicial map $f : K \to L$ between finite ordered abstract simplicial complexes induces a group homomorphism $f_n : C_n(K) \longrightarrow C_n(L)$, in general an R-module homomorphism if the coefficients are elements of a ring for all $n \geqslant 0$. From the later homomorphism one can deduce the following property for commuting diagrams between finite ordered simplicial complexes.

Proposition 2.

$$f_{n-1} \circ \partial_n^K = \partial_n^L \circ f_n \ \ \forall n \geqslant 0. \tag{II.21}$$

As a Corollary f induces the homomorphisms,

$$f_* : H_n(K) \to H_n(L) \text{ for all } n \geqslant 0 \tag{II.22}$$

defined as $f_*([z]) = [f_n(z)]$ for each n-cycle $z \in Z_n(k)$, which is well-defined and satisfies the relations $(g \circ f)_* = g_* \circ f_*$ and $id_* = id$.

Definition II.4.2 (Finite Filtration)**.** A finite filtration of an abstract simplicial complex K is a nested family of subcomplexes

$$K_0 \subseteq K_1 \subseteq \ldots \subseteq K_{m-1} \subseteq K_m = K. \tag{II.23}$$

A basic example of a finite filtration is the family of skeleta of a finite complex.

Fix the field $\mathbb{F}$. Given a finite filtration of a finite ordered abstract simplicial complex K such that

$$K_0 \subseteq K_1 \subseteq \ldots \subseteq K_{m-1} \subseteq K_m = K \tag{II.24}$$

for all $i \leq j$ and every $n \geqslant 0$, consider the homomorphism

$$\varphi_n^{i,j} : H_n(K_i) \longrightarrow H_n(K_j) \tag{II.25}$$

which is an $\mathbb{F}$-linear map and is induced by the inclusion $K_i \longrightarrow K_j$.

Definition II.4.3 (Persistent Homology Groups)**.**

- A nonzero homology class $\alpha \in H_n(K_j)$ is **born** at K_j if $\alpha \notin \operatorname{Im} \varphi_n^{i,j}$ for any $i < j$.
- A nonzero homology class $\alpha \in H_n(K_i)$ **dies** or vanishes at K_j for $j > i$ if $\varphi_n^{i,j}(\alpha) = 0$ but $\varphi_n^{i,j-1}(\alpha) \neq 0$.

- If α is born at K_i and dies at K_j with $j > i$, then $j - i$ is called the life or **persistence** of α.
- If α survives until $K_m = K$, then α is called essential or **permanent**.

  Define the persistent homology groups of K with respect to the filtration $\{K_i\}_{0 \leq i \leq m}$, in fact, $\mathbb{F}$-vector spaces, denoted by

$$H_n^{i,j}(K) = \operatorname{Im}\left(\varphi_n^{i,j} : H_n\left(K_i\right) \to H_n\left(K_j\right)\right) \tag{II.26}$$

  and denote its associated persistence Betti numbers by

$$\beta_n^{i,j}(K) = \dim_{\mathbb{F}} H_n^{i,j}(K) \tag{II.27}$$

II.5 Persistence Modules

Fix an arbitrary field $\mathbb{F}$.

Definition II.5.1 (Persistence Module)**.** A persistence module over $\mathbb{F}$ is a pair (V, π) where $V = \{V_t\}$, $t \in \mathbb{R}$, is a collection of $\mathbb{F}$-vector spaces indexed by the real numbers and π is a collection of $\mathbb{F}$-linear maps

$$\pi_{s,t} : v_s \longrightarrow v_t \text{ for } s \leq t \tag{II.28}$$

such that $\pi_{s,t} \circ \pi_{r,s} = \pi_{r,t}$ if $r \leq s \leq t$ and $\pi_{t,t} = \mathrm{id}$ for all t.

It can be thought (V, π) as a functor from $\mathbb{R}$ viewed as an ordered set to the category of $\mathbb{F}$-vector spaces.

Definition II.5.2 (Persistence Module of Finite Type)**.** A persistence module is of finite type or tame if:

- a) $\dim_{\mathbb{F}} v_t$ is finite for all t.
- b) There is a finite set $A = a_0, \ldots, a_n \subset \mathbb{R}$ such that:
  - For every $x \in \mathbb{R} \backslash A$ there is a $\delta > 0$ such that $\pi_{s,t}$ is an isomorphism for $x - \delta < s \leq t < x + \delta$.
  - For every $a \in A$ there is an $\varepsilon > 0$ such that if $a \leq t < a + \varepsilon$ then $\pi_{a,t}$ is an isomorphism while if $a - \varepsilon < s < a$ then $\pi_{s,a}$ is not an isomorphism.
  - $V_t = \{0\}$ if $t < a_0$, assuming that $a_0 < a_1 < \ldots < a_n$.

The set $A = \{\partial_0, \ldots, a_n\}$ is called the spectrum of (V, π)

Note from the previous definition that $\pi_{s,t}$ is an isomorphism whenever $a_n \leq s \leq t$.

Definition II.5.3 (Morphism)**.** A morphism $f : (V, \pi) \to (V', \pi')$ of persistence modules is an isomorphism if there is a morphism $g : (V', \pi') \to (V, \pi)$ such that $g \circ f = id_V$ and $f \circ g = id_V$.

It follows that f is an isomorphism if and only if f_t is an isomorphism of vector spaces for all t.

Definition II.5.4 (Interval Modules)**.** For $I = [a, b)$ or $I = [a, \infty)$, define a persistence module $\mathbb{F}(I)$ as

$$F(I)_t = \begin{cases} \mathbb{F} & \text{if } t \in I \\ 0 & \text{otherwise} \end{cases} \tag{II.29}$$

with $\pi_{s,t} = \text{id}$ if $s, t \in I$ or $\pi_{s,t} = 0$ otherwise.

These are persistence modules of finite type. The spectrum of $\mathbb{F}(I)$ is $\{a, b\}$ if $I = [a, b)$ or $\{a\}$ if $I = [a, \infty)$.

Definition II.5.5 (Direct Sum)**.** If (V, π) and (V', π') are persistence modules, their direct sum is the persistence module $(V \oplus V', \pi \oplus \pi'\ )$ with

$$\begin{aligned} (V \oplus V')_t &= V_t \oplus V'_t \\ (\pi \oplus \pi')_{s,t} &= \pi_{s,t} \oplus \pi'_{s,t}. \end{aligned} \tag{II.30}$$

If V and V' are of finite type, then $V \oplus V'$ is also of finite type. If A is the spectrum of V and A' is the spectrum of V' then the spectrum of $V \oplus V'$ is $A \cup A'$.

Theorem 2 (Normal Form Theorem)**.** For every persistence module V of finite type there is a finite collection of intervals $\{I_1, \ldots, I_N\}$ with $I_i = [a_i, b_i)$ or $I_i = [a_i, \infty)$ for every i such that $I_i \neq I_j$ if $i \neq j$ and there is an isomorphism of persistence modules

$$V \cong \mathbb{F}\,(I_1)^{m_1} \oplus \ldots \oplus \mathbb{F}\,(I_N)^{m_N} \tag{II.31}$$

with $m_i > 0$ for all i. Moreover, the set $\{I_1, \ldots, I_N\}$ and the integers $m_1, \ldots, m_N$ are unique. As a consequence of this fact, every persistence module of finite type yields a unique barcode (up to permutation of bars):

In general, if (V, π) is a persistence module of finite type, a nonzero vector $v \in V_t$ is born at t if $v \notin \operatorname{Im} \pi_{s,t}$ for any $s < t$. A nonzero vector $v \in V_s$ dies or vanishes at $t > s$ if $\pi_{s,t}(v) = 0$ and $\pi_{s,r}(v) \neq 0$ for $s \leq r < t$.

If v is born at $t = b$ and dies at $t = d$, then $d - b$ is its life or persistence. If $\pi_{b,t}(v) \neq 0$ for all $t > b$ then v is permanent.

Definition II.5.6 (Shift Action)**.** Let (V, π) be a persistence module of finite type with spectrum $A = \{a_0, \ldots, a_n\}$.

Consider a graded vector space $V_* = \bigoplus_{k=0}^{\infty} V(k)$, where

$$V(k) = \begin{cases} V_{a_k} & \text{for } 0 \leq k \leq n \\ V_{a_n} & \text{if } k > n. \end{cases} \tag{II.32}$$

We turn V_* into a graded module over the graded polynomial ring $\mathbb{F}[t]$ by defining, for $v \in V(k)$,

$$t \cdot v = \begin{cases} \pi_{a_k, a_{k+1}}(v) & \text{if } 0 \leq k < n \\ v & \text{if } k \geqslant n \end{cases} \tag{II.33}$$

Theorem 3 (Structure Theorem)**.** Let M be a finitely generated graded module over $\mathbb{F}[t]$, where $\mathbb{F}$ is a field. Then

$$M \cong \bigoplus_{i=1}^{m} \Sigma^{p_i} \mathbb{F}[t] \oplus \left(\bigoplus_{j=1}^{n} \Sigma^{q_j} \mathbb{F}[t] / (t_j) \right) \tag{II.34}$$

for some collection of integers $p_i \geqslant 0, q_j \geqslant 0, r_j \geqslant 1$. Moreover, this decomposition is unique up to a permutation of summands.

The Structure Theorem implies the Normal Form Theorem for persistence modules of finite type using the shift action.

For a persistence module (V, π) of finite type with spectrum $A = a_0, \ldots, a_n$, let V_* be the corresponding graded $\mathbb{F}[t]$-module.

Then V_* is finitely generated as an $\mathbb{F}[t]$-module and hence

$$V_* \cong \bigoplus_{i=1}^{m} \Sigma^{p_i} \mathbb{F}[t] \oplus \left(\bigoplus_{j=1}^{n} \Sigma^{q_j} \mathbb{F}[t] / (t^{r_j}) \right). \tag{II.35}$$

implies

$$V = \bigoplus_{i=1}^{m} \mathbb{F}\left[\partial p_i, \infty\right) \oplus \left(\bigoplus_{j=1}^{n} \mathbb{F}\left[\partial_{q_j}, \partial_{q_j + r_j}\right) \right) \tag{II.36}$$

II.6 Persistence Diagrams

Let (V, π) be a persistence module of finite type over a field $\mathbb{F}$, and let

$$\bigoplus_{i=1}^{n} \mathbb{F}\left[b_i, d_i\right) \oplus \bigoplus_{j=1}^{m} \mathbb{F}\left[c_j, \infty\right) \tag{II.37}$$

be its normal form.

The persistence diagram D of (v, π) has a point (b_i, d_i) for each $i \in \{1, \ldots, n\}$ and a point (c_j, d_∞) for each $j \in \{1, \ldots, m\}$, where d_∞ is an

arbitrary but fixed real number bigger than all values in the spectrum of V. Multiplicities are depicted with labels on the points of D.

II.7 Persistence Descriptors

The persistence diagrams can be summarized in a wide variety of forms, from a scalar summary such as total persistence or persistence entropy to a 2-dimensional summary such as a Betti curve, a landscape or a silhouette, or even in a 3-dimensional form such as a persistence surface or persistence image, where the role of kernels come into play with the aim of "smoothing" the persistence diagrams.

II.7.1 Numerical Summaries

Average Life

Definition II.7.1 (Average Life)**.**

$$L := \frac{1}{n} \sum_{i=1}^{n} (d_i - b_i) \,. \tag{II.38}$$

Average Midlife

Definition II.7.2 (Average Midlife)**.**

$$\bar{L} := \frac{1}{n} \sum_{i=1}^{n} \frac{b_i + d_i}{2}. \tag{II.39}$$

p-norms

For the following, let $\mathcal{D}$ be a persistence diagram. We are interested in computing summary statistics descriptors of the form $\mathcal{S} : \mathcal{D} \to \mathbb{R}$.

Definition II.7.3 (p-norm)**.**

$$||\mathcal{D}||_p := \sqrt[p]{\sum_{x,y \in \mathcal{D}} (b_i - d_i)^p} \tag{II.40}$$

The particular case when $p = 1$ is the respective total persistence measure that will be used as a topological feature descriptor in our modelling phase.

Persistence Entropy

Definition II.7.4.

$$E := \sum_{i=1}^{n} \frac{d_i - b_i}{L} \log_2 \left(\frac{d_i - b_i}{L} \right), \text{ where } L = \sum_{i=1}^{n} (d_i - b_i). \tag{II.41}$$

By looking at the latter equation it is clear that the persistence entropy definition is motivated by the usual entropy defined in physics by Shannon. In this context the persistence entropy of a random variable is the avarege level of uncertainty inherent to its outcomes.

II.7.2 Vectorized Summaries

Persistence Landscapes

For real numbers $b < d$, consider the tent function

$$\Lambda_{(b,d)}(t) = \sup \{0, \min\{t - b, d - t\}\}. \tag{II.42}$$

The landscape of a persistence module (V, π) of finite type is a sequence of piece wise linear functions $\lambda_k : \mathbb{R} \to \mathbb{R}, k = 0, 1, 2, \ldots$ defined as follows:

$$\lambda_k(t) = \text{kmax}_{i \in I} \left\{ \Lambda_{(b_i, d_i)}(t) \right\}. \tag{II.43}$$

If $\{(b_i, d_i)\}_{i \in I}$ is the multiset of points in the persistence diagram of (V, π) and kmax returns the kth largest value, or zero if there is no such.

Silhouettes

A silhouette of a persistence diagram $D = \{(b_i, d_i)\}_{i \in I}$ is a weighted average of tent functions from D:

$$\phi(t) = \frac{\sum_i w_i \Lambda_{(b_i, d_i)}(t)}{\sum_i w_i} \tag{II.44}$$

where $w_i \geqslant 0$ for all i.

A default choice is $\omega_i = (d_i - b_i)^p$ where p is a parameter. Choosing p small enhances low -persistence features while choosing p large enhances highly persistent features.

Betti Curves

Definition II.7.5. For each $k \geq 0$, let $\beta_k : \mathbb{R} \to \mathbb{R}$ be defined as

$$\beta_k(t) = \#\{(b, d) \mid b \leq t \leq d\} \tag{II.45}$$

where (b, d) ranges over the points in a given persistence diagram for homological dimension k.

Persistence Surfaces and Persistence Images

Let $\mathcal{D}$ be a persistence diagram and consider a function

$$\Phi(s,t) = \sum_{i=1}^{n} w_i G_i(s,t) \tag{II.46}$$

for (s,t) in a square, where each w_i is a weight and G_i is a 2-dimensional distribution function centered at (b_i, d_i). Note that the distribution function can be either a Gaussian, a Laplacian or a heat kernel, among other possibilities.

Definition II.7.6 (Persistence Surface)**.** The above equation gives the corresponding persistence surface associated to the underlying persistence diagram.

The persistence surface is the smoothed version of the persistence diagram via the kernel function defined by $K : X \times X \to \mathbb{R}$ if and only if there exist a Hilbert space H and a map $\Phi : X \to H$ such that $K(x,y) = \langle \Phi(x), \Phi(y) \rangle \; \forall x, y$, where the Hilbert space H is called feature space and the map Φ is called feature map.

Definition II.7.7 (Persistence Image)**.** It is the discretization of the map (II.46) on a 2-dimensional grid.

II.8 Distances

Definition II.8.1 (Matchings)**.** A matching between D and D' is a bijective function $\varphi : D \to D'$ such that for every $(x,x) \in \Delta$, either $\varphi(x,x) = (x,x)$ or $\varphi(x,x) = (b,d)$ with $b \neq d$, where Δ denotes the diagonal $b = d$.

Definition II.8.2 (l_∞-distance)**.** For each matching $\varphi : D \to D'$, define

$$\|\varphi\| = \sup \left\{ d_\infty((x,y), \varphi(x,y)) \mid (x,y) \in D \right\}. \tag{II.47}$$

where d_∞ is the l_∞-distance on $\mathbb{R}^2$, namely

$$d_\infty\left((x,y), \left(x', y'\right)\right) = \sup \left\{ \left|x - x'\right|, \left|y - y'\right| \right\} \tag{II.48}$$

Definition II.8.3 (Bottleneck Distance)**.** The bottleneck distance between two persistence diagrams is defined as

$$W_\infty\left(D, D'\right) = \min \left\{ \|\varphi\| \mid \varphi : D \to D' \text{ matching } \right\}. \tag{II.49}$$

Hence $W_\infty(D, D')$ is the smallest $\varepsilon \geqslant 0$ for which there exists a matching $\varphi : D \to D'$ for which $d_\infty((x,y), \varphi(x,y)) \leq \varepsilon$ for $\alpha ll(x,y) \in D$.

Definition II.8.4 (Wasserstein Distance)**.** The Wasserstein distances are defined for $p, q \geqslant 1$ as

$$W_p[q]\left(D, D'\right) = \min_{\varphi: D \to D'} \left(\sum_{(x,y) \in D} d_q((x,y), \varphi(x,y))^p \right)^{1/p} \tag{II.50}$$

where $d_q\left((x,y),(x',y')\right) = \left(|x-x'|^q + |y-y'|^q\right)^{1/q}$.

II.9 Sublevel Sets

Definition II.9.1. Let $f : [a,b] \to \mathbb{R}$ be a continous function. For each $t \in \mathbb{R}$, define its sublevel sets by

$$L_t(f) = \{x \in [a,b] \mid f(x) \leq t\}. \tag{II.51}$$

Note that if $s \leq t$ then $L_s(f) \subseteq L_t(f)$. From the definition $L_t(f) = \phi$ if $t < \inf(f)$ and $L_t(f) = [a,b]$ if $t \geqslant \sup(f)$.

The point $x_0 \in [a,b]$ is named a critical point if it is a local maximum or a local minimum, including $x_0 = a$ and $x_0 = b$ with its associated critical value $f\left(x_0\right)$.

Assume that f has finitely many critical points, therefore each critical point is isolated, so that we can associate to f a persistence module $V_t(f) = H_0\left(L_t(f)\right)$ and we let $\pi_{s,t} : V_s(f) \to V_t(f)$ be induced by the inclusion $V_s(f) \subseteq V_t(f)$ if $s \leq t$. In this order of ideas, the spectrum of (V, π) is contained in the set of critical values of f.

Theorem 4 (Stability Theorem)**.**

$$W_\infty(V(f), V(g)) \leqslant \|f-g\|_\infty \tag{II.52}$$

where $\|f-g\|_\infty = \sup\{|f(x)-g(x)| : a \leq x \leq b\}$.

Theorem 5 (Hausdorff Theorem)**.** Let M be a metric space. The diameter of a subset $X \leq M$ is $\operatorname{diam}(X) = \sin\{d(p,q) \mid p, q \in X\}$, recall that that X is bounded if diam (X) is finite.

Suppose given two subsets X, Y of M. Define

$$\begin{aligned} d(p,y) &= \inf\{d(p,y) \mid y \in Y\}, \\ d(x,y) &= \sup\{d(p,y) \mid p \in X4. \end{aligned} \tag{II.53}$$

Definition II.9.2 (Hausdorff Distance)**.** Let X and Y be compact sets. The Hausdorff distance is defined by

$$d_H(X,Y) = \sup\{d(X,Y), d(Y,X)\} \tag{II.54}$$

The Hausdorff distance is indeed a distance on the set of compact metric spaces.

Note that for the Hausdorff distance $d(X,Y) \neq d(Y,X)$ in general, and note also that $Y \leq X \Rightarrow d(Y,X) = 0$

If X is compact, then $d(y,x) = 0 \Leftrightarrow y \leq X$.

Definition II.9.3 (Gromov-Hausdorff Distance)**.** Let X and Y be compact metric spaces. The Gromov-Hausdorff distance between X and Y is defined as

$$d_{GH}(X,Y) = \acute{\inf}\left\{d_H^M(f(X), g(Y)) \mid f : X \hookrightarrow M, g : Y \hookrightarrow M \text{ isometrically }\right\} \tag{II.55}$$

where the infimum is taken over all isometric embeddings of X and Y into metric spaces, and d_H^M denotes Hausdorff distance in M.

Note that $d_{GH}(X,Y) = 0$ if and only if X and Y are isometric.

Definition II.9.4 (Correspondence)**.** A correspondence between X and Y is a surjective multivalued function from X to Y. That is, a subset $C \subseteq X \times Y$ such that for all $x_0 \in X$ there is some $(x_0, y) \in C$ and for all $y_0 \in Y$ there is some $(x, y_0) \in C$.

From the above definition we have that if C is a correspondence, then $C^{-1} = \{(Y,X) \in Y \times X \mid (X,Y) \in C\}$. Just to have in mind, the distortion of a correspondence $C \subseteq X \times Y$ is defined as $\text{dis}(c) = \sup\left\{\left|d^X(x,x') - d^Y(y,y')\right| : (x,y),(x',y')\right.$

Theorem 6 (Kalton-Ostrovskii)**.**

$$d_{GH}(X,Y) = \frac{1}{2}\inf\{\text{dis}(C) \mid C \subseteq X \times Y \tag{II.56}$$

Theorem 7 (Stability Theorem)**.** Let X and Y be point clouds. If $V_t(X) = H_*(R_t(X))$ then

$$W_\infty(V(X), V(Y)) \leq 2d_{GH}(X,Y) \tag{II.57}$$

Chapter III

Time-Dependent Topological Data Analysis

III.1 Persistence Diagram Bundles

Following the discussion on persistent diagrams, since a persistent diagram summarizes the persistent homology of a filtration one can consider in much more generality the notion of a persistence diagram bundle, which is the space of the persistence diagrams associated to a fibered filtration function. The underlying motivation relies in the fact that one would like to understand the evolution in the topology of a point cloud over a multi parameter set $p_0, p_1, p_2, ..., p_n$ (i.e. time t or filtration parameter r) in the case of taking a finite set of parameters one can compute a filtered complex of the form,

$$\mathcal{K}_{p_1}^{p_0} \subset \mathcal{K}_{p_2}^{p_0} \subset ... \subset \mathcal{K}_{p_n}^{p_0}. \tag{III.1}$$

Formally we define the notions of persistent diagram bundles contributed by [11].

Definition III.1.1. A fibered filtration function is a set

$$\{f_t : \mathcal{K}^t \to \mathbb{R}\}_{t \in \mathcal{T}} \tag{III.2}$$

where $\mathcal{T}$ is a topological space and $\{K^t\}_{t \in \mathcal{T}}$ is the set of simplicial complexes parameterized by $\mathcal{T}$ and f_t is a filtration function on $\mathcal{K}^t$.

Definition III.1.2 (Persistence Diagram Bundle)**.** Let $\{f_t : \mathcal{K}^t \to \mathbb{R}\}_{t \in \mathcal{T}}$ be a fibered filtration function. the base of the bundle is $\mathcal{T}$. The p-th total space of the bundle is defined by

$$E := \{(t, z) \mid t \in \mathcal{T}, z \in PD_p(f_t)\} \tag{III.3}$$

with the subspace topology inherited from the inclusion $E \hookrightarrow \mathcal{T} \times \bar{\mathbb{R}}^2$.

The p-th persistent diagram bundle is the triple $(E, \mathcal{T}, \pi)$, where $\pi : E \to \mathcal{T}$ is the projection $(t, z) \mapsto t$.

In the particular case when $\mathcal{T} \equiv \mathbb{R}$ and $\mathcal{K}^t \equiv \mathcal{K}$ the persistence diagram reduces to a 1-parameter set of persistence diagrams for a 1-parameter set of filtrations of $\mathcal{K}$ that we call a "vineyard" which will be treated in the next sections.

Some particular cases of this persistence diagram bundles including the vineyards will be the main concern of the project modelling, among others such as the persistent homology transform and fibered barcodes of multi-parameter persistence modules. The more general cases regarding multi-parameter sets that are not subsets of $\mathbb{R}$ will be left for a future research project.

III.2 The space $(\mathcal{D}_p, W_p)$

In order to construct the proper framework to treat the time-dependency of persistence one prerequisite is to define a suitable space of persistence diagrams equipped with a distance. For this purpose, the persistence diagrams will be defined in an abstract manner.

Definition III.2.1 (Abstract Persistence Diagram)**.** An abstract persistence diagram is a countable multi set of points along with the diagonal Δ, where the diagonal points are considered to have countable infinite multiplicity.

The latter space will be equiped with the pth Wasserstein distance.

Definition III.2.2 (p^{th} Wasserstein Distance)**.** The pth Wasserstein distance between two persistence diagrams X and Y is given by

$$W_p[\sigma](X, Y) := \inf_{\varphi: X \to Y} \left[\sum_{x \in X} \sigma(x, \varphi(x))^p \right]^{1/p}. \tag{III.4}$$

where $1 \leq p \leq \infty, \sigma$ is a metric on the plane, and φ ranges over bijections between X and Y.

In particular, the scenario treated is with $\sigma = L_q$. Note that for $p = \infty$ we have,

$$W_\infty [L_q] (X, Y) := \acute{\inf}_{\varphi: X \to Y} \sup_{x \in X} \|x - \varphi(x)\|_q \tag{III.5}$$

Note also that $W_\infty [L_\infty]$ is nothing else than the bottleneck distance.

Definition III.2.3 (The Space of Persistence Diagrams $\mathcal{D}_p$)**.** The space of persistence diagrams $\mathcal{D}_p$ consists of abstract persistence diagrams with finite distance to the empty diagram $D_\emptyset$, which is the diagram consisting of only the points on the diagonal:

$$\mathcal{D}_p = \{X \mid W_p\left(X, D_\emptyset\right) < \infty\} \tag{III.6}$$

where, $W_p = W_p[\sigma]$ is the pth-Wasserstein metric.

It is important to highlight the fact that $(\mathcal{D}_p, W_p\left[L_\infty\right])$ is a complete and separable (Polish) space.

III.3 Vineyards

The notion of time-dependent TDA comes from the concept of time-varying persistence diagrams which we refer to by vineyards. In such a case we have a timespan and for each time we have a persistent diagram which is going to be stacked to each other in the order of time, and each off-diagonal point of the diagram will vary continuously in time so that it forms a "vine". The space of abstract vineyards is going to be the space of paths in the persistent diagram space.

Definition III.3.1 (The Space of Abstract Vineyards)**.** The space of abstract vineyards is defined by

$$\mathcal{V}_2 := \{v : [0,1] \to \mathcal{D}_2\} \tag{III.7}$$

where v is W_2-continuous and $\mathcal{D}_2$ the space of persistence diagrams associated to the metric in question.

But in terms of persistence bundles we can get an alternative and maybe more useful definition for the vineyard structure as a particular case of bundle when considering a filtered complex varying with time and some additional parameter that gives place to a filtered complex as (III.1).

Definition III.3.2 (Vineyards Set)**.** Let the vineyards represent a 1-parameter set of filtrations obtained from a time-varying point-cloud

$$\{\mathcal{K}^t_{p_1} \subset \mathcal{K}^t_{p_2} \subset ... \subset \mathcal{K}^t_{p_n}\}_{t\in\mathbb{R}}. \tag{III.8}$$

For a fixed $t \in \mathbb{R}$ one can compute the persistent homology of the corrsponding filtration (III.1) replacing the hyper parameters $p_0 \longleftrightarrow t$ and obtain the correspoding persistence diagram $PD(t)$. Therefore it might seem more familiar to understand the vineyards as follows:

Definition III.3.3 (Persistence Vineyard)**.** A persistence vineyard is a continuously-varying stack of persistence diagrams

$$\{PD(t)\}_{t\in\mathbb{R}} \subset \mathbb{R}^3 \tag{III.9}$$

where each vine corresponds to a homology class and the persistence vineyard shows its evolution with time in the particular case of a 1-parameter set.

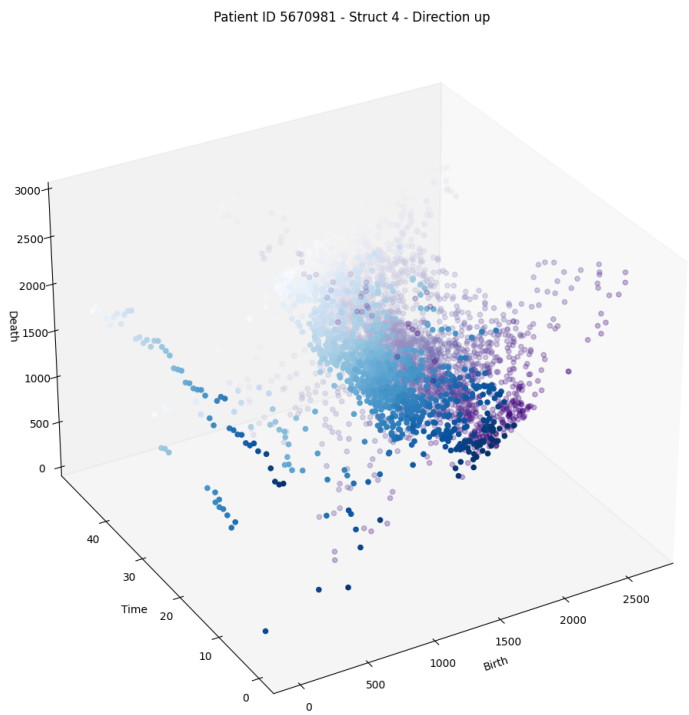


Figure III.1: Persistence vineyard Plot for one patient

Chapter IV

About Wavelets

IV.1 Brief Introduction

Wavelets are motivated by the problem of non-localization in the Fourier transform. One can think about the lack of sensitivity for these mappings to capture abrupt changes on signals. To solve this, the approach was to represent functions in $L^2(\mathbb{R})$ with a new class of well localised functions. In short, wavelets are zero mean, rapidly decaying, wave-like functions which by translation and dilation can represent any $f \in L^2(\mathbb{R})$.

To define properly this mathematical construction the discussion should be directed towards multi-resolution analysis. Let $f \in L^2(\mathbb{R})$ and let $j \in \mathbb{Z}$. Denote $(D_j f)(t) = 2^{j/2} f\left(2^j t\right)$, $t \in \mathbb{R}$.

Definition IV.1.1 (Multi-Resolution Analysis)**.** A multi-resolution analysis (MRA) is an increasing sequence $\cdots V_n \subset V_{n+1} \subset \cdots$ of closed subspaces of $L^2(\mathbb{R})$ such that:

- There exists $\varphi \in V_0$ such that the translates $\varphi_{0,k}(t) = \varphi(t-k), k \in \mathbb{Z}$, form an orthonormal base of V_0. The function φ is the scaling function of the MRA.
- $V_{n+1} = D_1(V_n)$ for all $n \in \mathbb{Z}$. Equivalently $f(t) \in V_n$ if and only if $f(2t) \in V_{n+1}$.
- $\overline{\bigcup_{n\in\mathbb{Z}} V_n} = L^2(\mathbb{R})$.
- $\bigcap_{n\in\mathbb{Z}} V_n = \{0\}$.

From the latter definition, the scaling function plays the role of determining the MRA.

Consider an MRA $\{V_n\}_{n\in\mathbb{Z}}$ as above, take the orthogonal complement of V_n in V_{n+1} and call it W_n; hence $V_{n+1} = V_n \oplus W_n$. So at this stage we have $V_{n+1} = \bigoplus_{j\leq n} W_j$ and $L^2(\mathbb{R}) = \bigoplus_{j\in\mathbb{Z}} W_j$.

It is important to mention that associated to the MRA we have two types of orthogonal projections:

$$P_n f = \sum_{k\in\mathbb{Z}} \langle f, \varphi_{n,k}\rangle \, \varphi_{n,k} \tag{IV.1}$$

which represents the resolution n for a given function and

$$Q_n f = P_{n+1} f - P_n f \tag{IV.2}$$

which can be interpreted as the detail that has to be added to an orthogonal projection to get the next level of detail.

Two important theorems will be mentioned without proof to have in mind for the development of this theory.

Theorem 8. If ψ is a wavelet of the $MRA\,\{V_n\}_{n\in\mathbb{Z}}$, then the system $\{\psi_{n,k}\}_{n,k\in\mathbb{Z}}$, where $\psi_{n,k} = D_n\left(\psi_{0,k}\right)$, is an orthonormal basis of $L^2(\mathbb{R})$ (called the wavelet basis of the MRA).

Theorem 9 (Mallat's Theorem)**.** Let $\{V_n\}_{n\in\mathbb{Z}}$ be a MRA with scaling function φ. Then

- $\sum_{k\in\mathbb{Z}} \left|\langle\varphi, \varphi_{1,k}\rangle\right|^2 = 1$ and $\sum_{k\in\mathbb{Z}} \langle\varphi, \varphi_{1,k}\rangle \, \overline{\langle\varphi, \varphi_{1,k-2l}\rangle} = 0$ for all $l \in \mathbb{Z}\backslash\{0\}$
- $\psi = \sum_{k\in\mathbb{Z}} (-1)^k \overline{\langle\varphi, \varphi_{1,1-k}\rangle} \varphi_{1,k}$ is a wavelet for the MRA.

In this line of reasoning, take into account a sketch about the construction of an MRA:

- Determine a $\varphi \in L^2(\mathbb{R})$ (scaling function) such that $\{\varphi_{0,k}\}_{k\in\mathbb{Z}}$ is an orthonormal system and define $V_0 = \overline{\langle\varphi_{0,k}\rangle}_{k\in\mathbb{Z}}$.
- Check that $V_n := D_n\left(V_0\right)$ is an increasing sequence of closed subspaces in $L^2(\mathbb{R})$ and that $\overline{\bigcup_{n\in\mathbb{Z}} V_n} = L^2(\mathbb{R})$.
- Find, using Mallat's theorem, the associated wavelet ψ, so that $\{\psi_{0,k}\}_{k\in\mathbb{Z}}$ is an orthonormal basis of $W_0 = V_1 \oplus V_0$.

The construction of a scaling function φ of an MRA can be made in terms of its Fourier transform $\widehat{\varphi}$. The following affirmation deals with the possibility to construct the associated wavelet ψ using Fourier analysis. Developing the scaling function in terms of the basis $\{\varphi_{1,k}\}_k$ of V_1 one sees that there exists a 1-periodic function $H(\xi)$ such that $\widehat{\varphi}(\xi) = H(\xi/2)\widehat{\varphi}(\xi/2)$ named the refinement mask or low pass filter of the MRA.

Theorem 10 (Main Lemma)**.** A function $f \in L^2(\mathbb{R})$ is in the detail space W_0 if and only if there exists a 1-periodic function v such that

$$\hat{f}(\xi) = e^{i\pi\xi} v(\xi) \overline{H\left(\xi + \frac{1}{2}\right)} \widehat{\varphi}(\xi/2). \tag{IV.3}$$

To close this technical introduction it is important to highlight the fact that Mallatâs theorem provides thus an algorithm for constructing the wavelet from the MRA and the scaling function via the Fourier coefficients of the 1-periodic refinement mask. This can be implemented numerically in the so-called cascade algorithm, that has the following steps: scaling function, low pass, high pass and finally Mallat's wavelet.

IV.2 Continuous Wavelet Transform

The Wavelet transform can be thought of as a generalization of the Fourier transform. In fact the idea is similar to that of the Short-Time Fourier Transform, but instead of using a sliding window we rather allow translations and dilations.

Let $\psi \in L^2(\mathbb{R})$ (mother wavelet) and assume it has some regularity, at least continuous or continuously differentiable such that,

- $\|\psi\|_2 = 1$,
- ψ is compactly supported, or very rapidly decaying,
- $\int \psi(t)dt = 0$.

There exist a whole spectrum of wavelets, despite all of the ones that one can think of.

Now proceed with the definition of the continuous wavelet transform in a similar way that we define the continuous Fourier transform.

Definition IV.2.1 (Inverse Wavelet Transform)**.** The continuous wavelet transform of $f \in L^2(\mathbb{R})$ associated to a wavelet ψ is

$$Wf(a,b) = \langle f, \psi_{a,b} \rangle = \frac{1}{\sqrt{a}} \int_{\mathbb{R}} f(t) \overline{\psi\left(\frac{t-b}{a}\right)} dt. \tag{IV.4}$$

The wavelet transform in fact contains enough information to reconstruct the underlying function f. For completeness it remains to mention the inverse wavelet transfom.

Definition IV.2.2. If $\psi \in L^1(\mathbb{R}) \cap L^2(\mathbb{R})$ is a real valued function satisfying the admissibility condition

$$c_\psi := \int_0^\infty \frac{|\widehat{\psi}(\xi)|^2}{|\xi|} d\xi < \infty \tag{IV.5}$$

then, for $f \in L^2(\mathbb{R})$,

$$f(t) = \frac{1}{c_\psi} \int_{\mathbb{R}} \int_0^\infty Wf(a,b)\psi_{a,b}(t) \frac{da}{a^2} db \tag{IV.6}$$

IV.3 Scaleograms

This is the key part for the application of the wavelets into the research project. Since wavelets capture frequencies over time but wihtout compromising on the precision of the time information or on the detected range of frequencies, they can be visualized with a scaleogram. The procedure relies on multiplying the original signal with a wavelet. It is important to have in mind that small-scale wavelets allow us to capture high frequencies within precise time intervals, and large-scale wavelets allow us to capture low frequencies across longer time intervals and wavelets achieve this since they can be translated in time and scaled in width.

To illustrate this process we may have in mind the operations that are happening at the backstage:

- Scale the wavelet according to the parameter and center it at time.
- We multiply the wavelet and signal together, to get a filtered signal. We compute the integral (or the sum, in the discrete case) of the filtered signal to output a scalar.

Regarding the scaleogram representation, the computations are performed for n translations and n scales with a scalar output, so that it can be represented as an $n \times n$ matrix where the entries are the scalars mentioned above and hence it can be represented as a 2D image. Here the vertical axis corresponds to wavelet's scales and the horizontal axis to time, in contrast with spectrogram representations associated to the Gabor transform where they correspond to frequencies and time, respectively. Intuitively the wavelet scale can be interpreted on an inverse relationship to frequency. In fact scaleograms are better to detect low frequencies and to localize high frequencies than spectrograms since they have widths that vary with scale.

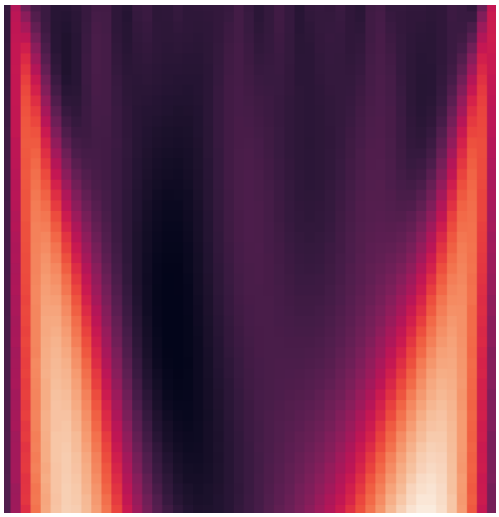


Figure IV.1: Scaleogram

Chapter V

A word on Artificial Intelligence

V.1 Classical Machine Learning

V.1.1 Support Vector Machines

As mentioned before, classification is one of the main purposes of mathematical modelling, and Support Vector Machines (SVM) is one of the most popular algorithms to achieve this task. The description of this procedure will begin by focusing in the linear case and afterwards extending to the non-linear one.

For the simplest case consider a databasis $\mathcal{S}$ whose datapoint records are $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subset \mathbb{R}^d$ and the target is the set of labels $\mathcal{Y} = \{-1, 1\}$, assume it is linearly separable. By assumption, there exists an affine hyperplane (V.1)

$$h(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b \textit{ such that } h(\boldsymbol{x}_i) > 0 \textit{ if } y_i = 1 \textit{ and } h(\boldsymbol{x}_j) < 0 \textit{ if } y_j = -1. \tag{V.1}$$

Finding such a hyperplane can be approached as a linear programming problem (V.2) where we have to search for the weights $\boldsymbol{w}$ and bias b:

$$y_i \left(\boldsymbol{w}^\top \boldsymbol{x}_i + b \right) - 1 \geq 0. \tag{V.2}$$

Note that the hyperplane in question is not unique. In fact it is equivalent modulo the selected margin for classification which turns to be an additional degree of freedom decided by computing the distance separating the hyperplane to the closest positive and closest negative points denoted by δ_+ and δ_-, respectively. Now define the quantity $\delta = \delta_+ + \delta_-$ which is nothing more than the margin of the classifier which one would like to maximize. Just for a matter of notation, the closest point to the hyperplane, that is when we have equality in (V.2) is called the support vector.

Note that the margin for the separating hyperplane must satisfy $\delta = 2/\|\boldsymbol{w}\|_2$ by geometrical arguments. With this in mind we only need to find the hyperplane with the largest margin, stated as a quadratic optimization problem (V.3):

$$\begin{aligned} &\underset{\boldsymbol{w},b}{\text{minimize}} \frac{1}{2}\|\boldsymbol{w}\|^2 \\ &\text{subject to } y_i \left(\boldsymbol{w}^\top \boldsymbol{x}_i + b \right) - 1 \geq 0, \quad 1 \leq i \leq n. \end{aligned} \tag{V.3}$$

This can be approached with the Lagrangian multipliers method. For a matter of completeness on this section it is mentioned the outline of the resolution to the problem with this methodology. The Lagrangian expression (V.4),

$$\begin{aligned}\mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\lambda}) &= \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{m} \lambda_i y_i \boldsymbol{w}^\top \boldsymbol{x}_i - \lambda_i y_i b + \lambda_i \\ &= \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} - \boldsymbol{\lambda}^\top \boldsymbol{X}\boldsymbol{w} - b\boldsymbol{\lambda}^\top \boldsymbol{y} + \sum_{i=1}^{m} \lambda_i\end{aligned} \tag{V.4}$$

where $\boldsymbol{X} \equiv y_i \boldsymbol{x}_i^\top$ and it is subject to the conditions,

$$\begin{aligned}\nabla_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\lambda}) &= \boldsymbol{w} - \boldsymbol{X}^\top \boldsymbol{\lambda} = \boldsymbol{0} \\ \frac{\partial \mathcal{L}}{\partial b}(\boldsymbol{w}, b, \boldsymbol{\lambda}) &= \boldsymbol{y}^\top \boldsymbol{\lambda} = 0.\end{aligned} \tag{V.5}$$

and finally we get the expression for the dual optimization problem (V.6):

$$\text{minimize } \frac{1}{2}\boldsymbol{\lambda}^\top \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{\lambda} - \boldsymbol{\lambda}^\top \boldsymbol{e} \quad \text{subject to} \quad \boldsymbol{\lambda} \geq \boldsymbol{0}. \tag{V.6}$$

The optimal value of the separating hyperplane can be found by solving the dual problem (V.6), since there is a one to one correspondence between dual variables λ_i and data points $\boldsymbol{x}_i$ the weights can be found by computing $\boldsymbol{w} = \boldsymbol{X}^\top \boldsymbol{\lambda}$ and the bias by the Karush Kuhn Tucker conditions for this particular problem. This can be re written by combining the restrictions of the initial and the gradient of the Lagrangian as follows:

$$\begin{aligned}\boldsymbol{X}\boldsymbol{w} + b\boldsymbol{y} - \boldsymbol{e} &\geq \boldsymbol{0} \\ \boldsymbol{\lambda} &\geq \boldsymbol{0} \\ \lambda_i \left(1 - y_i \left(\boldsymbol{w}^\top \boldsymbol{x}_i + b\right)\right) &= 0 \text{ for } 1 \leq i \leq n \\ \boldsymbol{w} - \boldsymbol{X}^\top \boldsymbol{\lambda} &= \boldsymbol{0} \\ \boldsymbol{y}^\top \boldsymbol{\lambda} &= 0.\end{aligned} \tag{V.7}$$

Now regarding real world applications the latter description only holds for particular cases when both classes are binary, with the points well separated by an affine hyperplane. But in practice the situation described is ideal. In this section, some details about not-exact separation and non-linear separation will be discussed, since the multiple classes case can be thought as an extension of several binary classifiers.

For non-exact separation the constraints mentioned before are not satisfied, therefore there is non-feasible solution of the optimization problem. So we proceed by adding a set of additional slack variables $s_1, \ldots, s_n$ and re write the constraints as

$$\boldsymbol{w}^\top \boldsymbol{x}_i + b \geq 1 - s_i, \text{ for } y_i = 1, \quad \boldsymbol{w}^\top \boldsymbol{x}_j + b \leq -1 + s_j, \text{ for } y_j = -1, \quad s_i \geq 0 \tag{V.8}$$

and allowing the possibility for the number of errors to not be bounded by $\sum_{i=1}^{n} s_i$ and there is a second problem that arises, namely minimizing the upper bound of the number of errors, stated as

$$\begin{aligned} \text{minimize} \frac{1}{2}\|\boldsymbol{w}\|^2 + \mu \sum_{j=1}^{n} s_j \\ \text{subject to } y_i \left(\boldsymbol{w}^\top \boldsymbol{x}_i + b\right) - 1 + s_i \geq 0, \quad 1 \leq i \leq n \\ s_i \geq 0, \quad 1 \leq i \leq n. \end{aligned} \tag{V.9}$$

On the other hand, for the non-linear case the key is that the dual formulation of the optimization problem is dependent on the dot products of the form $\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$. We proceed by mapping the data base into a higher dimensional space

$$\varphi : \mathbb{R}^p \to \mathcal{H} \tag{V.10}$$

now we can apply SVM to the representation in the higher dimensional space depending only on the dot products of the form,

$$K\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = \left\langle \varphi\left(\boldsymbol{x}_i\right), \varphi\left(\boldsymbol{x}_j\right) \right\rangle \tag{V.11}$$

where K is the kernel function. There is a full spectrum of choices of kernels (radial, Gaussian, etc).

It is important to highlight the fact that we are not concerned with the form of the function φ since in the equations for the hyperplane $\boldsymbol{w}^\top \boldsymbol{x}$ terms are to be replaced with $K(\boldsymbol{w}, \boldsymbol{x})$ so that we get a non-linear decision boundary as expected.

V.1.2 K Nearest Neighbors

The present classifier is characterized because it memorizes the training dataset rather than learning a discrminative function. Take a look to the algorithm which is very intuitive. It is based on the assumption that similar inputs have similar outputs based on the following classification rule; for a test input x, assign the most common label amongst its k most similar training inputs.

1. Choose a number k and a distance metric
2. For each sample, find the k-nearest neighbors
3. Assign the class label by majority vote

We define formally this method (V.1.1).

Definition V.1.1 (KNN Method)**.** Let $\mathbf{x}$ be a test point and let D be a domain.

Define the set of the k nearest neighbors of $\mathbf{x}$ as $S_{\mathbf{x}}$. Formally $S_{\mathbf{x}}$ is defined as $S_{\mathbf{x}} \subseteq D$ such that $|S_{\mathbf{x}}| = k$ and define

$$\operatorname{dist}\left(\mathbf{x}, \mathbf{x}'\right) \geq \sup_{(\mathbf{x}'', y'') \in S_{\mathbf{x}}} \operatorname{dist}\left(\mathbf{x}, \mathbf{x}''\right) \ , \forall \left(\mathbf{x}', y'\right) \in D \backslash S_{\mathbf{x}} \tag{V.12}$$

Define the KNN classifier by $KNN(\cdot)$,

$$\mathrm{KNN}(\mathbf{x}) = \operatorname{mode}\left(\left\{y'' : \left(\mathbf{x}'', y''\right) \in S_{\mathbf{x}}\right\}\right) \tag{V.13}$$

where mode $(\cdot)$ means to select the label of the highest occurrence.

Note that the classifier's performance relies deeply in the distance metric used which can be selected from a variety of distance functions such as: Manhattan, Minkowski, Euclinean and so on.

It is flexible under the choice of the number k of clusters and the choice of distance function, for the former one can tune the number k over a range to evaluate performance.

One important aspect to choose the present algorithm relies on the fact that it adapts to new training data. On the other hand, one has to take into account that the computational complexity for classifying new samples grows linearly with the number of samples on the training data set, but a bigger challenge occurs with the storage capacity for large data sets.

V.2 Deep Learning

V.2.1 Neural Networks

Deep neural networks have been gaining popularity lately since they have been achieving breakthroughs in several fields. Despite the latter they still are not well understood, it is still really challenging to understand the reasoning behind these objects. It has to be taken into account that the degree of complexity grows directly proportional with the dimensionality of the network, in fact we can build some intuition over low dimensional neural networks by using topological arguments. Additionally, the other mystery that arises is in understanding how does data go from one layer to another.

If one considers the simplest case of a neural network with just an input, one hidden layer and an output one would obtain something similar to the linear case. If hidden layers continue to be added the data will be transformed from one another by creating each time a new representation that is not linear in general. It has to be mentioned the variety of layers used for this purpose, they take weight matrices W which is point-wise applied to elements of the domain $x \in D$ and a translation vector is added b. For a concrete example consider the tanh layer $\tanh(Wx + b)$.

Note that each of the layers preserves topological properties since it deforms the space of the corresponding domain by stretching or squishing but never breaks, cuts or folds the space. For instance, connected sets are invariant to the application of NNs layers. In other words, the layers play the role of homeomorphisms.

Theorem 11. Layers with N inputs and N outputs are homeomorphisms, if the weight matrix, W, is non-singular. Moreover, the result remains valid for an arbitrary number of layers.

The result will not be proven here, but the main idea will be mentioned. Multiplying by an invertible matrix W is a homeomorphism, and so do the translations and applying continous functions such as tanh, sigmoid or softplu (but for instance not ReLU) is still an homeomorphism.

Continuing the discussion from the point that neural networks achieve non-linear performance, it is important to take into account the manifold hypothesis: data forms lower-dimensional manifolds in its embedding space, which makes a lot of sense from the theoretical and experimental perspectives. Therefore, the real job of the NNs is to separate tangled manifolds. In short, low dimensional networks fail on separating manifolds regardless of their depth which is resolved by considering wider layers.

V.2.2 Recurrent Neural Networks

Our human reasoning does not produce thoughts from scratch. The understanding of everything that surround is been understood by our empirical knowledge, in other words the thoughts have persistence. In terms of neural networks the traditional architectures do not follow the persistence when processing information, but recurrent neural networks aim to address such a problem with architectures that have internal loops that allow information to persist. The way they work is by receiving some input and producing an output after passing through the loop that if we think of it as the unrolled version of the network it would look like a linear network with several copies of the same network passing information from one another.

The discussion will now follow for a particular case of recurrent neural networks named the long-short term memory type. These are based on the idea to connect information in the past to draw conclusions in the present. There is a full spectrum of examples to take into account in different areas such as speech generation in natural language processing, time-series prediction, among other successful applications of LSTMs.

V.2.3 Long-Short Term Memory

Digging deeper on the LSTMs architecture, in contrast to RNNs which have the form of a chain repeating modules of a naural network with a simple structure of possibly s single layer (i.e. tanh layer), LSTMs have this same chain structure form of repeating modules but instead of having a single layer they have four of them that are constanly interacting.

The important detail behind these mysterious network modules relies on a cell state which crosses the complete module side to side with some linear interactions that have the ability to remove or add information that has been transported in the cell state regulated by gates, which are structures that are composed by sigmoid functions as well as point wise operations to let information in and out. Their output is a measure between 0 and 1 that can be interpreted as the relevance of the information to be let in, where 0 means completely get rid of the information and 1 means to keep it all. The LSTMs have three of these structures to have in control the information that goes to the cell state.

At this stage we have an idea of the internal structures of the network module. Now we will follow the reasoning process made by the module. The process starts with a forget layer to control which information will be thrown away from the cell state by looking at the input x_t, the previous time step output h_{t-1}, and mapping the sigmoid function, it looks like (V.14).

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right). \tag{V.14}$$

In the following step the module decides which information is going to be added into the cell state. For this job we have an input layer (V.15) and a new candidate layer (V.16), which play the role on deciding which values to update and create a vector with new candidates values, respectively:

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \tag{V.15}$$

$$\bar{C}_t = \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right). \tag{V.16}$$

With this setup we now need to transform the cell state from C_{t-1} to C_t. We do so by applying the forget layer to the previous cell state and adding the new candidate values weighted by the degree of need to update each state value, that is (V.19).

$$C_t = f_t * C_{t-1} + i_t * \bar{C}_t. \tag{V.17}$$

Finally the output is computed by filtering a last time the cell state. It consists of creating an output layer (V.20) and multiplying by the cell state mapped by a tanh layer for re scaling purposes.

$$o_t = \sigma\left(W_o \cdot [h_{t-1}, x_t] + b_o\right) \tag{V.18}$$

$$h_t = o_t * \tanh C_t. \tag{V.19}$$

Finally the output is computed by filtering a last time the cell state. It consists of creating an output layer (V.20) and multiplying by the cell state mapped by a tanh layer for re scaling purposes.

$$o_t = \sigma\left(W_o \cdot [h_{t-1}, x_t] + b_o\right) \tag{V.20}$$

$$h_t = o_t * \tanh C_t. \tag{V.21}$$

Despite the success of the LSTMs in a wide range of applications there is a further step of improvementthat is to consider attention. By attention it is meant to pick information at each step of the RNN from a pool of information. There exist a whole range of variations of this type of architectures

V.2.4 Convolutional Neural Networks

The convolutional neural networks (CNN) can be thought as a type of neural network that has many identical copies of the same neuron. This allows it to have a great number of them and in consequence process computationally large models with a small error cost since each of the neurons runs their processes in parallel.

Their architecture is composed by an input layer that passes information into an identical group of neurons A which compute convolutional features, it is composed by several neurons connected in parallel and pass them straight to a fully-connected layer F. The later was just the simplest case, but it has to be taken into account that in fact one can connect several convolutional layers $A, B, C, ...$ inputing the next layer the output from the former, this may have an advantage since the network would be able to detect more abstract and complex features.

It is a common practice to intertwine pooling layers between convolutional layers, in particular "max-pooling" layers are oftenly used. They behave as a "zoom-out" perspective which associates a small region of the data to the surroundings it belongs.

In particular, the CNNs are very useful for image analysis. Therefore the context where they are more useful is for a 2-dimensional CNN. Their architecture is quite similar to the 1-dimensional ones with the difference that they look into patches instead of segments. Another possible higher-dimensional version are the 3-dimensional CNNs that work well for analyzing volumetric data such as videos or medical scans, but they are used just for those particular cases, they are not as succesful as the 2-dimensional CNNs and they are difficult to visualize.

To formalize these class of networks, let x be the inputs and y the outputs, therefore one express their relation as

$$y = A(x) \ , \ A(x) = \sigma(Wx + b) \tag{V.22}$$

where W is the weights matrix, b the bias and σ some activation function.

The former argument is neat but is not how we would like to formalize the convolutional layers. Since we are concerned with image analysis, the 2-dimensional case will be treated. Define the 2-dimensional convolutional layers in the context of image analysis as follows:

Definition V.2.1 (Convolutional Layers 2-dim)**.** Let $x = \{x_{i,j}\}_{1 \leqslant i,j \leqslant n}$ be the pixels associated to an image and let $g : \{1, 2, ..., n\}^2 \to \mathbb{R}$ an arbitrary function. Define the convolution between x and g as

$$(x * g)_{i,j} := \sum_{t \leqslant N} \sum_{t \leqslant N} \left(x_{i-s,j-t} g(s,t) \right) . \tag{V.23}$$

The matrix $(x * g)_{i,j}$ will be named the feature map and the function g the kernel.

Some of the relevant properties of this operation for the underlying context rely on the commutativity and associativity of the image and the kernel. In addition, if one thinks about the convolution in the context of the Fourier

transform this operation inherits the property of invariance under translation, that is, the outputs are stable under input translation.

It is important to mention that one can interpret such a convolution as a regularization, because by applying the underlying operation the number of parameters being reduced.

V.3 Evaluation Metrics

In the context of classification problems there is a very wide range of metrics that can be used to evaluate performance they can be partitioned into the following classes:

- Threshold metrics

  Quantify the classification prediction errors.

- Ranking metrics

  Evaluate classifiers on how effective they are on separating classes.

- Probability metrics

  Quantify the uncertainty in a classifier's predictions.

For the purpose of this project we will be dealing with the first two classes of metrics.

V.3.1 Threshold Metrics

In general this class of metrics summarize the between prediction and validation sets.

Definition V.3.1 (Accuracy)**.**

$$Accuracy := \frac{CP}{TP} \tag{V.24}$$

where, CP denotes the correct predictions and TP the total number of predictions.

The error can be thought of as the complement to the accuracy, see the following definition.

Definition V.3.2 (Error)**.**

$$Error := \frac{IP}{TP} \tag{V.25}$$

where, IP denotes the incorrect predictions and TP the total number of predictions.

At this stage it can be seen the need of new metrics. Both of the latter metrics are used very often in the field of applied machine learning, but they are inappropriate for imbalanced classification problems, since a high accuracy or low error are achieved by a no skill model that only predicts the majority class.

For the following metrics it is appropriate to define the confusion matrix which provides a greater insight on the possible errors performed.

Definition V.3.3 (Confusion Matrix)**.** Let $C \in M_{N\times N}(\mathbb{R})$ be a matrix with entries in the real field. The rows correspond to the real classes and the columns to the predicted classes for an N-class classification problem. Hence its entries $C_{i,j}$ are equal to the number of observations to be of the class i and predicted class j for all $i \in 1, 2, ..., N$ and for all $j \in 1, 2, ..., N$.

In the case of a binary classification problem the confusion matrix looks like the following one:

		Predicted		
		Negative	Positive	Total
Real	Negative	TN	FN	$TN + FN$
	Positive	FP	TP	$FP + TP$
	Total	$TN + FP$	$FN + TP$	N

Where TN stands for true negatives, TP for true positives, FN for false negatives and FP for false positives. With this machinery we can proceed to define the rest of metrics that will be helpful for us.

The true positive rate which gives insights of how well is the positive class been predicted.

Definition V.3.4 (Sensitivity)**.**

$$Sensitivity := \frac{TP}{TP + FN} \tag{V.26}$$

with its respective complement, that is the measure of how well is the negative class been predicted.

Definition V.3.5 (Specificity)**.**

$$Specificity := \frac{TN}{FP + TN} \tag{V.27}$$

In an analogous we have the measure of the positive samples that have been correctly classified into the positive class.

Definition V.3.6 (Precision)**.**

$$Precision := \frac{TP}{TP + FP} \tag{V.28}$$

and the recall which is the same as sensitivity,

Definition V.3.7 (Recall)**.**

$$Recall := \frac{TP}{TP + FN} \tag{V.29}$$

One can create an additional measure that summarizes both precision and recall,

Definition V.3.8 (F1-Score)**.**

$$F1 := 2\frac{PR}{P + R} \tag{V.30}$$

where P and R denote precision and recall, respectively.

The last measure is often used for imbalanced classification purposes, and we can generalize it for gauging the trade-off between precision and recall.

Definition V.3.9 (Fβ-Score)**.**

$$F\beta := (1 + \beta^2)\frac{PR}{\beta^2 P + R} \tag{V.31}$$

where P and R denote precision and recall, respectively.

Note that the Fβ measure represents the precision to recall importance. For instance if $\beta > 1$ it gives more weight to recall, while $\beta < 1$ favors precision and $\beta = 1$ gets back the F1 measure.

Another useful metric to use for the imbalanced classification case is known as the balanced accuracy.

Definition V.3.10 (Balanced Accuracy)**.**

$$BalancedAccuracy := \frac{1}{2}(Sensitivty + Specificity). \tag{V.32}$$

The latter score identifies the positives predicted by the classifier even if they are true or false and is more reliable than accuracy since the later gives the same weight to both classes.

With the machinery of the threshold metrics it is important to mention its side backs. There is a big limitation for these class of metrics since they assume same class distributions in training, validation and prediction sets.

V.3.2 Ranking Measures

The advantage of this class of distributions relies on the fact that they do not make assumptions about class distribution. They are designed for a classifier that predicts a score or a probability of an element belonging to a class, from the underlying score some thresholds can be studied to test its effectiveness. By ranging the thresholds through the spectrum of thresholds maintaining a good score will perform better in class separation and hence will rank higher.

The receiving operating characteristic ROC curve gives the necessary and sufficient information for evaluating the ability to discriminate classes. To dig into it we will define two important quantities.

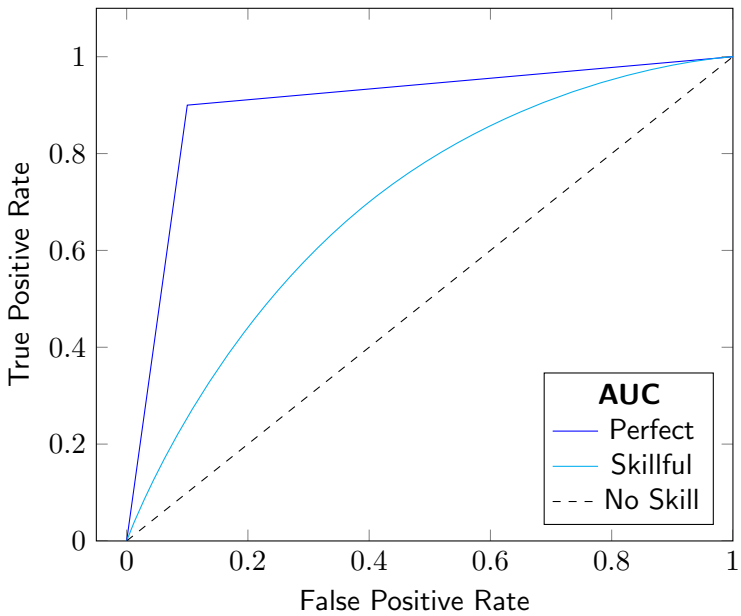
Definition V.3.11 (True Positive Rate)**.**

$$TruePositiveRate := \frac{TP}{TP + FN} \tag{V.33}$$

Definition V.3.12 (False Positive Rate)**.**

$$FalsePositiveRate := \frac{FP}{FP + TN} \tag{V.34}$$

Each point on the curve has an associated threshold. The diagonal dashed line represents a no skill classifier below that line it is classified to perform worse than no skill, the light blue curve represents a skillful classifier and the dark clue one a perfect classifier.



A useful metric for imbalanced classification is the area under the ROC curve, named by ROC AUC. It is the probability for a classifier to assign a random sample to be of the positive class more likely than for the negative class which can be very optmistic for imbalanced classification problems.

In a similar way a precision-recall curve can be constructed which focuses particularly on the minority class, hence more useful for imbalanced classification problems and in a similar way that before one can summarize the curve in a quantity named the PR AUC (precision-recall area under the curve).

Chapter VI

Database

VI.1 UK Biobank

The UK Biobank is a large-scale biomedical database and research resource that contains in-depth genetic and health information from half a million UK participants, which is periodically augmented. Moreover it is open-access for researchers in the field and in fact it has had a positive impact in scientific discoveries and understanding of the most threatening diseases, consequently the advance of modern medicine by enabling better predictive models for early diagnosis and the development of therapeutic interventions.

UK Biobank provides an open-access resource for global scientists to further our understanding of the most common and life-threatening diseases including cardiovascular disease. The imaging, health, genetic and lifestyle data generously donated by our $500,000$ participants is enabling better predictive models of disease for earlier diagnosis, and developing therapeutic interventions.

Figure VI.1: UKBB Patient Images

The images of the UK Biobank database are stored in .nifti files format which was initially an initiative from the Neuroimaging Informatics Technology which overcomes the downfalls from the precedent file formats used for storing medical images. In the UK Biobank there were around $\sim$ 40k patients with images available for research.

On the other hand, the data points are labeled with the International Classification of Diseases Tenth Revision (ICD-10) system, which is used

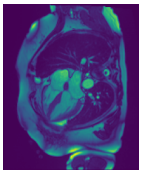


Figure VI.2: Heart Image

by physicians to classify and code all diagnoses, symptoms and procedures for claims processing. It was designed by the World Health Organization in company with the National Center for Health Statistics to have a better alignment. For the cardiovascular study purposes the following diseases are of interest:

- Ischaemic heart disease
- Valvular heart disease
- Non-ischaemic cardiomyopathies
- Heart failure
- Cardiac arrhythmia

From all the patients of the UK Bio bank data base there were $\sim$ 90k properly labeled.

Chapter VII

Methodology

In the present section details are given on the project methodology, from the database construction phase, through modelling, data preparation, topological feature extraction and finally data processing to the selected classifiers.

VII.1 Data Bases

The project database was constructed in parallel by gathering the images from the UK Biobank by means of the VPN direct access of the UB, from the list of all the patient IDs, where 10231 of them where chosen at random to have a sample of the real data base. Once the patients were selected, the process continued with image processing, data preparation and further with the topological feature extraction to have the topological summary of all the images. In parallel, the ID labels were obtained by means of a tool designed for this purpose on an external TFG. Part of the data preparation phase was to match the patients IDs from both the labels and topological feature data frames.

VII.1.1 Unbalance Degree

For the Ischemic Heart Disease study, the data base is composed of 10231 patients, where 1202 of them were tagged with the disease and 9029 where healthy. That means an imbalance degree of 11.75% and 88.25%, respectively. Which gives place to an imbalance ratio of 7.51. This measure indicates a significant unbalance of the data set. Therefore the situation from now on will be different and hence with a higher degree of complexity.

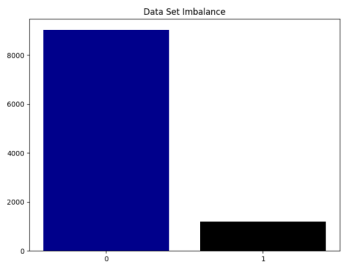


Figure VII.1: Wrong Segmentation

VII.1.2 Train & Test Split

The train/test split was performed initially before proceeding with the data preparation phases for the different experiments in order to guarantee comparability among them. From a bag of the whole list of patients IDs. An

stratified train/test split was performed in a random state with a test size of 0.3 and respective train size of 0.7. We are talking about a train size of 7161 patients distributed as 6320 healthy and 841 with the disease and a test size of 3070 patients distributed as 2709 healthy and 361 with the disease. Both lists of train and test ids where saved into a .pkl file for ease of partitioning the subsequent data sets used in a further data preparation stage.

VII.1.3 Baseline Database

The frames for the end systole and end diastole define a filter over the time steps of the time-topological database to get two different data sets tagged with "ES" and "ED", respectively. With the purpose of having baseline results with the classical machine learning algorithms with out taking into account the time dependence and also for being able to reproduce the state of art results.

VII.1.4 Control Database

Dynamic Collapse

Following the reasoning of the baseline database, the control database is also considered to in some sense compare the time dependent classification to a static case, this time by collapsing the time series corresponding to each patient to a single point by applying the mean map over all the time steps of each patient. This database is tagged as "dynamic collapse" and it is labeled in results with "DC".

Dynamic Frames

On the other hand, we also have the dynamic frames database that works as a control by giving a time dependence to the end-systole and end-diastole frames by considering it as a two-time series. It is labeled by DF in the results.

VII.2 Data Preparation

VII.2.1 Image Processing

In the UK Bio bank database one could find a folder named with the respective patients id with two files "$la_4ch.nii.gz$" and "$la_4ch_a seg.nii.gz$" which are the real long axis image and its associated segmentation, respectively.

Quality Filter

From the full list of ids a first quality filter was applied to discard patients with at least more than one incomplete images. The patients who had a

single incomplete image where let into the study but this time point information was droped out of the registers. In all of the cases the time point $t = 6$ had a null image in the segmentation, therefore attributed to some error in the segmentation processing phase.

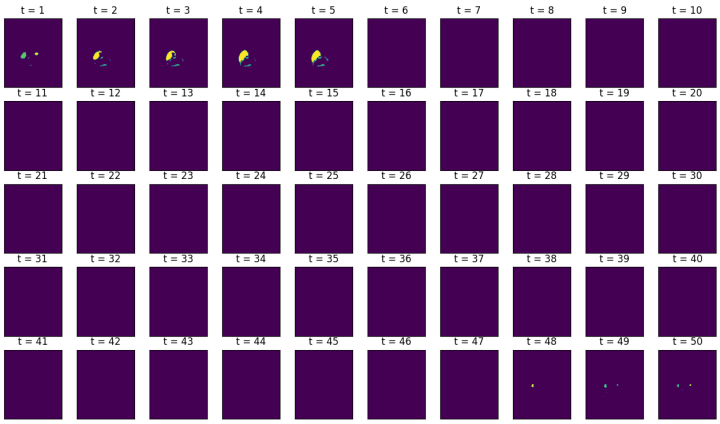


Figure VII.2: Incomplete Segmentation

Zoom-In

Despite the fact that persistence is invariant respect on the zoom to an image, that is, the persistence features are the same for any chosen zoom, for image visualization purposes both the real image and the segmentations where applied a zoom-in mapping for visualisation purposes (see Figure VII.4).

Split Structures

Since the aim is to study the five different cardiac structures, the segmented images are really useful but used independently therefore the split structures mapping returns an specific segmented structure provided the segmentation image and the identifier (color number) of the desired structure (Figure VII.4).

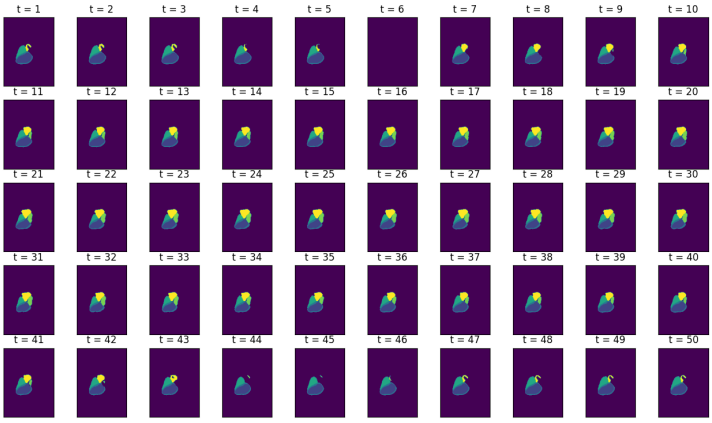


Figure VII.3: Wrong Segmentation

Hadamard Mask

With the five different structures isolated one proceeds to compute the real image with a fixed segmentation mask in front and the application of a pixel-wise Hadamard product (VII.2.1) between both, to provide the detailed version of each of the five structures (Figure VII.4).

Definition VII.2.1 (Hadamard Product)**.** Define the Hadamard product as an element-wise bilinear operation over the space of matrices $M_{N\times d}$ over an arbitrary field $\mathbb{K}$. Let $A, B \in M_{N\times d}$ be two matrices of the same dimension with entries in an arbitrary field. Then their Hadamard product is given by

$$A \odot B := (a \cdot b)_{ij} = a_{ij} \cdot b_{ij} \quad \forall i \in \{1, 2, ..., N\} \forall j \in \{1, 2, ..., d\} \tag{VII.1}$$

VII.2.2 Time-Database

To conclude the image processing phase, the images were stored in arrays for each structure and each time point in an individual patients data frame of size 48×7, where the two additional dimensions correspond to the id identifier and the time point see (Table VII.7).

To conclude the image processing phase, the image information of each of the patients was stored in a data frame with the columns corresponding to each of the cardiac structures plus an id identifier and an ordered time label variable, where each entry has the detailed image of its corresponding cardiac

structure in a determined time stored in an array. That is a 48×7 data frame with the name of the patients id and stored in a .pkl format. Latter all of the individual data frames in individual .pkl files where concatenated to a big data frame of all images.

VII.3 Modelling

At this stage we have achieved to transform the images .nifti file format to an structured pandas data frame were instead of having a folder of two .nifti files with the real and segmented images we transformed it to a big data frame were each sub data frame associated to an individual patient consists of the time point records and the variable associated to its id identifier, its time point labels and the column variables corresponding to each of the five structures under the study, where each entry of the matrix corresponds to the image stored in an array. As a crucial point in the research process the modelling is presented.

VII.3.1 2-Dimensional Time-Persistence

For a single patient of the study, as mentioned previously we have a sequence of time ordered images for each of the five cardiac structures. The explanation will proceed in general so let s be the arbitrary identifier of a cardiac structure fix a time $t \in [0, 48] \subset \mathbb{R}$ and then we have the image of a patient's cardiac structure s at some time t denoted by $Img(s, t) \subset \mathbb{R}^2$ which is a 2-dimensional array where each matrix entry corresponds to a voxel, which therefore creates the sublevel sets of a function determined by the color code in the heat map. Formally, we are dealing with some activation function of the form $f : \mathcal{V} \times \mathcal{T} \to \mathbb{R}$ over a bounded voxel area $\mathcal{V} \subset \mathbb{R}^2$ and a set of ordered time steps $\mathcal{T}$.

The pixels of the 2-dimensional array are the just 2-dimensional voxels which have the natural structure of a bi dimensional cubical complex in the case of image processing since each voxel corresponds precisely to one cubical simplex, otherwise using a simplicial complex would involve interpolation schemes [16]. Therefore, proceed by computing the cubical complex of the image.

Having computed the cubical complexes it remains to perform the filtration calculation, that is an ordering of the elements in the cubical complexes which it is going to be done for each of the time steps. We will use the sublevel set filtration and superlevel set filtration, in the following the former will be described and the latter it is equivalent modulo the inverse direction. Let $f_i : \mathcal{V}_i \times \mathcal{T} \to \mathbb{R}$ be the activation function in the direction of the time steps. Now use the canonical way of assigning values to $f_i(\cdot, t)$, fix t_j and let C_i be the cubical complex associated to a voxel $\mathcal{V}_i$, each vertex voxel of

C_i receives the activation value at time t_j and for the higher dimensional element of the complex σ it is assigned the value $f_i(\sigma, t_j) := \sup_{v \in |(\sigma)} f_i(c, t_j)$ and sort the cubical complexes C_i in ascending order, where in case of a tie we will take the ascending order in the homology dimension resulting in a nested sequence of simplicial complexes.

Once we have a sequence of cubical complexes properly filtered, the persistent homology of the underlying sequence can be computed. It results in a set of time-varying persistence diagrams ordered by time for each of the structures. In technical terms we obtain a persistence vineyard for each of the cardiac structures.

To summarize the topological phenomenon numerical persistence summaries will be used such as total persistence and persistence entropy computed over every single persistent diagram, thus providing each of both measures for each time and each structure for all of the patients.

See the image processing details for the 2-persistence model in the following diagram (Figure VII.4) in the next page.

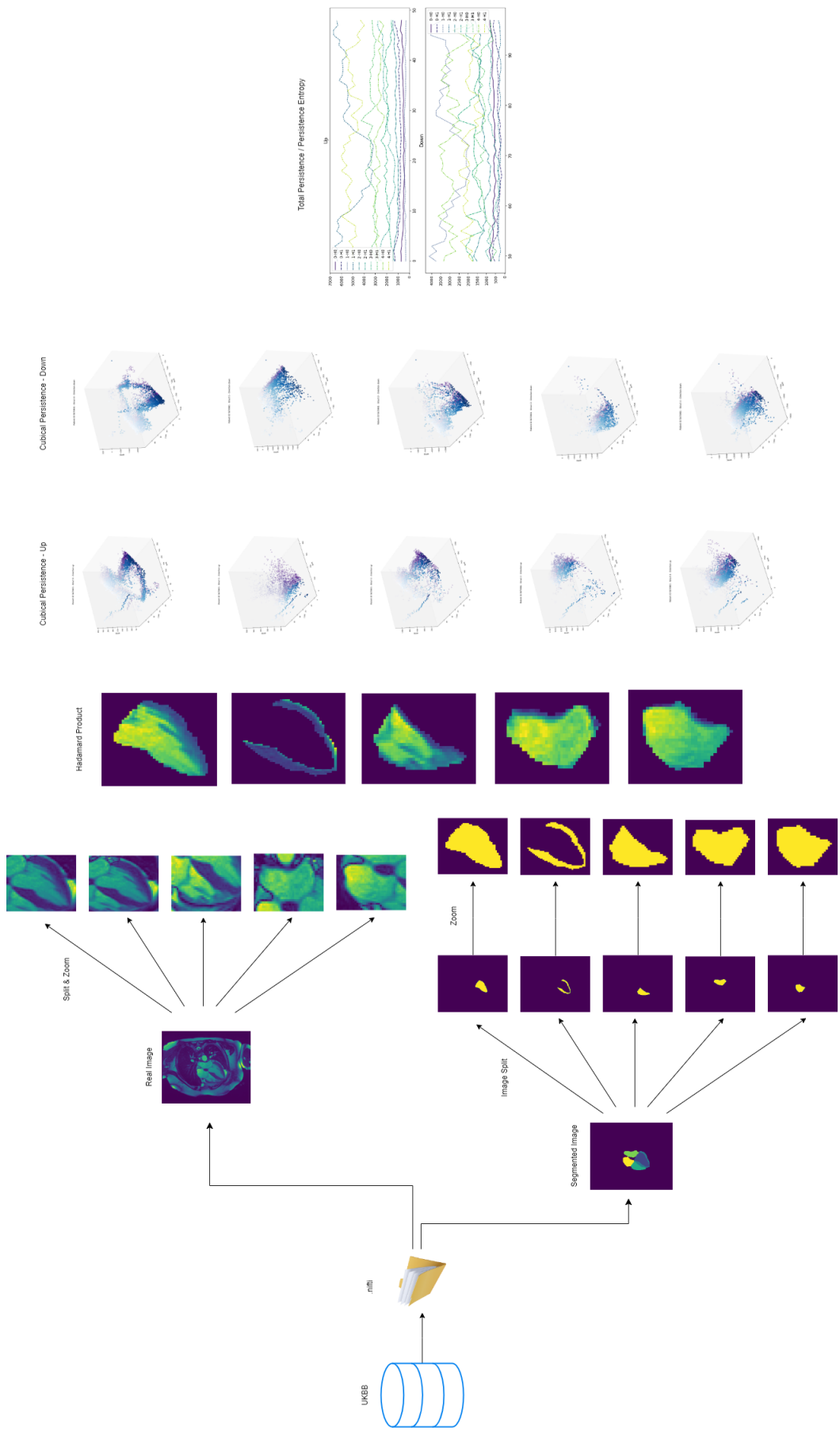


Figure VII.4: Persistence vineyard Plot for one patient

VII.3.2 $(2+1)$-Dimensional Time-Persistence

Recall the structure of the images for each of the patients:we have a stack of 2-dimensional images of cardiac long axis along the time dimension. Therefore we do not have a 3-dimensional image but rather we have a third dimension that determines the time evolution of the bi dimensional image. Therefore we can think of the stack of images as a $(2+1)$-dimensional image (the notation is the same used in general relativity to distinguish between the space and time dimensions). In this line of reasoning each patient has a $(2+1)$-dimensional image.

Since in this case the neighboring pixels form voxels that can be canonically associated to bi dimensional cubes we follow a similar reasoning from the previous section, namely we proceed with computing the cubical complex of each of the $(2+1)$-dimensional images.

In this case we are dealing with activation functions of the form $f : \mathcal{V} \times \mathcal{T} \to \mathbb{R}$ over a bounded voxel area $\mathcal{V} \subset \mathbb{R}^2$ and a set of ordered time steps $\mathcal{T}$ but in this case it differs with the previous approach in the subtlety of the assignment of the activation function values, each $f_i(\cdot,\cdot)$ will be assigned the activation function values corresponding to each cubical complex C_i while the higher dimensional element will be assigned $f_i(\sigma,\cdot) := \sup_{v \in |(\sigma)} f_i(c,\cdot)$ for the sublevel sets filtration and recall that the super level set filtration is equivalent modulo reversing the filtration order respect to the activation function.

Note that with the present approach we will still get connected paths and loops from the images and additionally we will be getting cavities that are generated by the dimension of time. Therefore we are now dealing with three homology groups. In that line of reasoning we get a single persistence diagram for each structure and each direction.

See the following diagram (Figure VII.5) to make clear the obtention of the persistence diagrams from the stack of bi dimensional images via considering cubical persistence and including time as an additional variable.

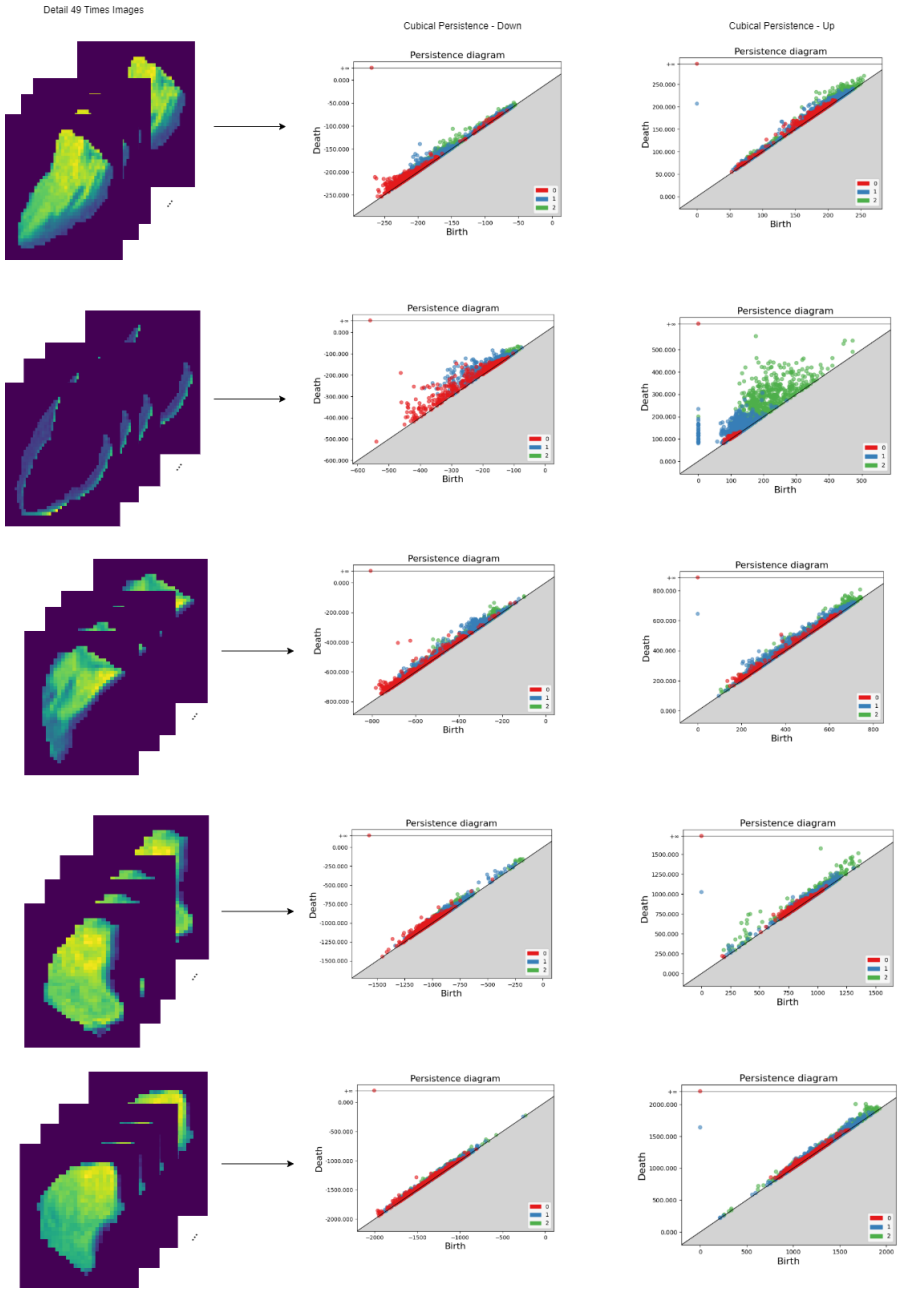


Figure VII.5: Persistence vineyard Plot for one patient

VII.3.3 Time Series Wavelets

The data base used for the time series wavelets has the same structure as the one used for the 3-Persistence model (Figure tab: 3-persistence) but with the slight difference that instead of having scalars on its entries, it has $N \times N$ arrays corresponding to the scaleograms.

The results from the 2-persistence model were nothing else than time series of topological features. Taking into account the underlying use case, one would be interested in having detailed information about the frequencies and the time where they occurred in between the cardiac cycle. As mentioned before the continuous wavelet transform provides us with this information by performing a projection into a chosen multi-resolution analysis basis, in some sense as a generalization of the Fourier transform. for this case the Mexican hat wavelet was used.

Therefore by fixing a patient of the study one obtains a single scaleogram for each structure and each direction which has associated scaleogram (Figure VII.6).

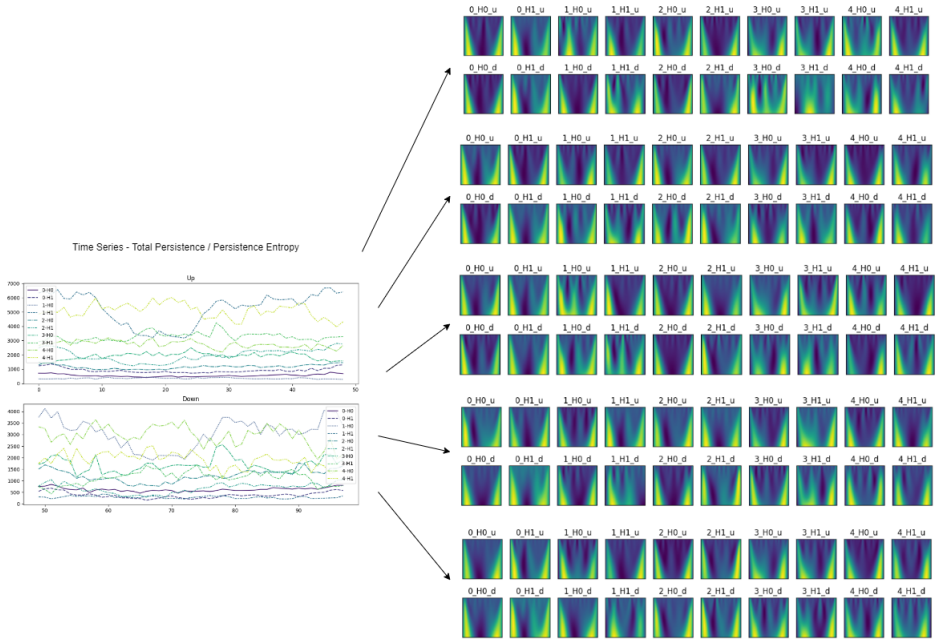


Figure VII.6: Persistence Time Series to Scaleograms

VII.3.4 Time-Topological Database

For each patient the five cardiac structures were considered. In the case of considering 2-dimensional homology groups the individual patients data frame was composed from 48 time points for each filtration direction and for each cardiac structures two homology groups, in addition to the time label and the id identifier, that is for an individual patient a data frame of size 96×24 (Table VII.7). Where the column variables are named with an encoding of the type $(structure) - H(hdimension) - (fdirection)$, where $structure$ is the integer associated to each of the cardiac structures in the range $\{0, 1, 2, 3, 4\}$, $hdimension$ is the homology group dimension in this case in $\{0, 1\}$ and $fdirection$ is the filtration direction that could either be up or $down$.

	time	id	0_H0_u	0_H1_u	1_H0_u	1_H1_u	2_H0_u	2_H1_u	3_H0_u	3_H1_u	...	0_H1_d	1_H0_d	1_H1_d	2_H0_d	2_H1_d	3_H0_d	3_H1_d	4_H0_d	4_H1_d	target
0	0	3013415	641.0	1157.0	330.0	3996.0	1330.0	2520.0	1728.0	3984.0	...	628.0	2374.0	284.0	1486.0	1227.0	1916.0	1668.0	2190.0	1555.0	0
1	1	3013415	657.0	1215.0	320.0	4130.0	1315.0	2433.0	2112.0	4496.0	...	752.0	2236.0	254.0	1504.0	1161.0	2732.0	2080.0	2350.0	1820.0	0
2	2	3013415	664.0	1197.0	346.0	4286.0	1429.0	2307.0	2024.0	4136.0	...	716.0	2280.0	324.0	1462.0	1272.0	2476.0	2164.0	2535.0	1845.0	0
3	3	3013415	589.0	1206.0	338.0	4938.0	1240.0	2292.0	1716.0	4908.0	...	570.0	2736.0	234.0	1381.0	1143.0	2744.0	1740.0	2655.0	1850.0	0
4	4	3013415	542.0	1039.0	362.0	3792.0	1180.0	2073.0	1840.0	4712.0	...	564.0	1940.0	364.0	1291.0	975.0	2732.0	2308.0	2850.0	1680.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
917275	44	1676631	803.0	1370.0	268.0	6284.0	2050.0	3117.0	1948.0	3200.0	...	650.0	2700.0	186.0	1615.0	1911.0	2064.0	1048.0	2115.0	1305.0	0
917276	45	1676631	882.0	1437.0	274.0	6212.0	2014.0	3054.0	1804.0	2848.0	...	799.0	2504.0	104.0	1678.0	1635.0	1212.0	724.0	2505.0	1855.0	0
917277	46	1676631	931.0	1581.0	262.0	6702.0	2023.0	2961.0	1832.0	3208.0	...	895.0	2540.0	186.0	1399.0	1863.0	2084.0	656.0	2840.0	1360.0	0
917278	47	1676631	878.0	1557.0	254.0	6666.0	1726.0	3261.0	1736.0	3028.0	...	851.0	2756.0	196.0	1495.0	1788.0	2160.0	828.0	2755.0	1570.0	0
917279	48	1676631	876.0	1710.0	310.0	6620.0	1663.0	3048.0	1640.0	3624.0	...	853.0	2696.0	204.0	1552.0	1527.0	1268.0	848.0	2230.0	1595.0	0

501319 rows × 23 columns

Figure VII.7: Table 2-Persistence

On the other hand, when considering the $(2+1)$-dimensional persistence, for an individual patient there was a single time record, but 3 homology groups for each of the structures and also two directions of the filtration. Therefore for an individual patient the data corresponds to 1×32 (Table VII.8). In this case the variable columns maintain the same convention, but considering three homology group dimensions in the range $\{0, 1, 2\}$. This time the advantage of considering three homology groups relies on the fact that additionally to connected paths and loops the topological features capture cavities.

In the case of the wavelet model data base we got the same structure from the $(2 + 1)$-dimensional persistence database but instead of having scalars as entries of the matrix we had $N \times N$ arrays associated to the scaleogram images.

	id	0_H0_u	0_H1_u	0_H2_u	1_H0_u	1_H1_u	1_H2_u	2_H0_u	2_H1_u	2_H2_u	...	2_H0_d	2_H1_d	2_H2_d	3_H0_d	3_H1_d	3_H2_d	4_H0_d	4_H1_d	4_H2_d	target
0	1511623	1796.0	10322.0	9446.0	566.0	33306.0	59526.0	2205.0	17937.0	20070.0	...	5610.0	15606.0	9750.0	10104.0	28820.0	16968.0	14885.0	44140.0	35130.0	0
1	1930752	1312.0	11696.0	11070.0	502.0	28616.0	50998.0	2205.0	21663.0	21843.0	...	6489.0	18780.0	10944.0	13128.0	34708.0	24516.0	12415.0	40690.0	29915.0	0
2	3284043	1110.0	10257.0	9559.0	122.0	41792.0	52222.0	2136.0	22923.0	23604.0	...	6435.0	19059.0	10668.0	8800.0	22856.0	16416.0	14410.0	36935.0	21110.0	0
3	2354916	1062.0	12389.0	11964.0	380.0	28094.0	56330.0	2145.0	30846.0	30546.0	...	7674.0	23352.0	14592.0	8156.0	21448.0	14276.0	16200.0	58045.0	43800.0	0
4	3263760	1425.0	9609.0	8990.0	1054.0	23076.0	38132.0	2916.0	24687.0	23511.0	...	7956.0	19473.0	11589.0	18224.0	54560.0	37924.0	15105.0	41600.0	28305.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
10229	2484009	868.0	6680.0	6052.0	338.0	48668.0	46834.0	1251.0	15381.0	15216.0	...	4305.0	11337.0	6828.0	5036.0	10356.0	6720.0	12475.0	38160.0	27430.0	0
10230	1827789	1954.0	12413.0	12410.0	752.0	35184.0	60654.0	2172.0	25239.0	22728.0	...	4626.0	17865.0	11076.0	9768.0	30384.0	21480.0	12805.0	34920.0	25730.0	1
10231	1654684	2489.0	14632.0	14108.0	1136.0	34646.0	57078.0	1500.0	15411.0	16749.0	...	4935.0	12360.0	7401.0	21792.0	75380.0	57616.0	15020.0	47055.0	35500.0	0
10232	3932203	1699.0	14990.0	13298.0	602.0	53586.0	89534.0	3525.0	33363.0	38511.0	...	8619.0	28893.0	20442.0	8128.0	22924.0	15388.0	11465.0	37245.0	23145.0	0
10233	4076586	2753.0	13663.0	12193.0	498.0	41450.0	44898.0	3792.0	30213.0	28872.0	...	10380.0	27342.0	14517.0	16276.0	45460.0	30932.0	21330.0	66410.0	46440.0	0

10234 rows × 32 columns

Figure VII.8: Table 3-Persistence

It is important to highlight the existence of the two independent databases for each persistence metric considered. That is the database with the total persistence measure was treated by separate from the database with the persistence entropy. Although their structure is the same, to avoid confusions the measure label was always present in the .pkl file name.

VII.4 Classifiers

This section should begin by mentioning the following quote "a classifier is only as good as the metric used to evaluate it". With this said by choosing the wrong evaluation metric the selected model will not be the correct one and the performance will not be the expected. Therefore this is a crucial and challenging decision for applied machine learning and the situation is even worse in the case of an imbalanced database basically because most of the metrics assume a balanced class distribution and in general not all prediction errors are equal for imbalanced classification.

VII.4.1 Classical Machine Learning

Classical Machine Learning algorithms do not take into account the time dependency these class of models are applied for the control case of starting baseline results for the classification of end systole and end diastole data and also serve as a control of the time dependent data respect to classical methods by taking the average of the persistence measures of all the time points for each of the patients. The models considered for this study where KNN and SVM.

For each data set and each model a grid search was performed by using a pipeline on a grid search with cross validation with at least 3-folds, because of computational capacity and refit parameter. This procedure was applied to the train data set, for a following evaluation on the test dataset.

The evaluation metrics selected were the confusion matrix entries, in addition to the balanced accuracy and F1 score for the imbalanced situation we have and also the Fβ score with $\beta = 0.5, 2$ which is a measure widely used for imbalanced classification and may give us some trade-off of precision-recall.

VII.4.2 LSTM Neural Networks

The LSTM fits canonically to the time dependent data structure we are passing to the model an array of the form (n samples, n timesteps, n features). The LSTM architecture considered is one of the simplest cases of this class of models:

- LSTM(20, input shape=(n timesteps, n features))
- Dropout(0.2)
- Dense(1, activation='sigmoid')

The compiler was built with an adam optimizer and binary cross entropy loss as it is done in the case of binary classification problems.

Taking into account the underlying situation of the imbalanced data set the evaluation metrics used are the confusion matrix entries in addition to the F1 score, balanced accuracy and the AUC indicator. The latter is reported in the literature as over optimistic on imbalanced data sets.

VII.4.3 Convolutional Neural Networks

The CNN architecture selected is a commonly used architecture for binary image classification as a standing point for future modifications, it consists of four blocks as follows:

- Block One
  - Conv 2D (filters=32, kernel size=3, activation='relu', padding='same', input shape=[49, 49, 20])
  - MaxPool2D
- Block Two
  - Conv2D(filters=64, kernel size=3, activation='relu', padding='same')
  - MaxPool2D
- Block Three
  - Conv2D(filters=128, kernel size=3, activation='relu', padding='same')

  - Conv2D(filters=128, kernel size=3, activation='relu', padding='same')
  - MaxPool2D

- Head
  - Flatten
  - Dense(6, activation='relu')
  - Dropout(0.2)
  - Dense(1, activation='sigmoid')

With the same compiler used for the LSTM case and considered the same evaluation metrics. It is important to mention that each of the neural network models was optimized on an individual basis of the data sets to gauge the parameters of the architecture such as epochs, batches, learning rate and drop rate, the space of hyper-parameters could not be as big because of computational capacity.

Chapter VIII

Results & Discussion

VIII.1 Classical ML

Recall that the data sets used for classification with classical machine learning algorithms KNN and SVM where the static framed data sets such as end systole and end-diastole ("ES", "ED") and the control cases regarding dynamic collapse by taking the mean of the 2-persistence time series "DC" for the topological features total persistence and persistence entropy. In addition, the 3-persistence also enters into these class of single point data sets that can be applied to ML algorithms.

Model	Data	balanced_accuracy	f1	tn	fp	fn	tp	precision	recall	sensitivity	specificity
KNN	2PE(ES)	0.53	0.20	1256	629	152	95	0.13	0.38	0.38	0.67
KNN	2PE(ED)	0.52	0.20	1786	923	221	140	0.13	0.39	0.39	0.66
KNN	2TP(ES)	0.53	0.20	1256	629	152	95	0.13	0.38	0.38	0.67
KNN	2TP(ED)	0.52	0.20	1786	923	221	140	0.13	0.39	0.39	0.66
KNN	2PE(DC)	0.54	0.21	1880	829	222	139	0.14	0.39	0.39	0.70
KNN	2TP(DC)	0.54	0.21	1880	829	222	139	0.14	0.39	0.39	0.70
KNN	3TP	0.58	0.25	1880	829	194	167	0.17	0.46	0.46	0.70

Table VIII.1: KNN results

model	data	balanced accuracy	f1	tn	fp	fn	tp	precision	recall	sensitivity	specificity
SVC	2PE(ES)	0.53	0.17	1628	257	200	47	0.15	0.19	0.19	0.86
SVC	2PE(ED)	0.52	0.15	2443	266	310	51	0.16	0.14	0.14	0.90
SVC	2TP(ES)	0.53	0.17	1628	257	200	47	0.15	0.19	0.19	0.86
SVC	2TP(ED)	0.52	0.15	2443	266	310	51	0.16	0.14	0.14	0.90
SVC	2PE(DC)	0.53	0.17	2394	315	300	61	0.16	0.17	0.17	0.88
SVC	2TP(DC)	0.53	0.17	2394	315	300	61	0.16	0.17	0.17	0.88
SVC	3PE	0.58	0.25	1992	717	210	151	0.17	0.42	0.43	0.74
SVC	3TP	0.62	0.30	2099	610	190	171	0.22	0.47	0.47	0.77

Table VIII.2: SVC results

Whereas the control case of the dynamic collapse is consistent with the static frames results. Moreover, (2+1)-persistence shows an advantage upon the static frames.

VIII.2 TDA & LSTM

In this application of the model the time dimension is already been taken into account.

Data	tp	fp	tn	fn	accuracy	precision	recall	auc	prc	spec	balanced accuracy
2TP	31	55	2654	330	0.87	0.36	0.09	0.73	0.27	0.97	0.53
2PE	23	29	2680	338	0.88	0.44	0.06	0.73	0.26	0.98	0.52
2PE(DF)	5	11	1874	242	0.88	0.31	0.02	0.65	0.21	0.99	0.51
2TP(DF)	0	0	1885	247	0.88	0	0.00	0.62	0.20	1	0.5

Table VIII.3: TDA & LSTM results

Unfortunately, it follows from the evaluation metrics that the performance of the classifier is very low. Despite its performance by taking into account the AUC metric, which as reported in the literature is a good indicator for imbalanced problems, there is an advantage for considering the time dimension in the persistent homology framework and morevoer it shows that taking the full heart beat cycle has better results than considering only a two-point time series including the end-systole and end-diastole.

VIII.3 TDA, Wavelets & Convolutional Neural Networks

As a last attempt to achieve better results for the classification the wavelets processing phase was motivated by the applied harmonic analysis course. Since it generates scaleograms it is a suitable input for a 2-dim CNN model.

Data	tp	fp	tn	fn	accuracy	precision	recall	auc	sensitivity	specificity	balanced accuracy	f1 score
2PE	294.0	1380.0	1329.0	67.0	0.53	0.18	0.81	0.70	0.81	0.49	0.65	0.29
2TP	294.0	1557.0	1152.0	67.0	0.47	0.16	0.81	0.68	0.81	0.43	0.62	0.27

Table VIII.4: TDA, Wavelets & CNN results

This time also has to be mentioned the lack of expected results for the classifiers. But taking into account the optimistic case of the AUC metric this classifier also has comparable results to the LSTM case that takes the full heart beat cycle. Therefore it is worth to continue exploring this model.

Chapter IX

Conclusions & Outlook

To conclude the research project, a framework for measuring the variation of the descriptors of CMR images was designed from scratch with an initial strong computational component for the data preparation phase, followed by a robust mathematical modelling phase based on algebraic topology and harmonic analysis to be the input of a machine learning classifier. The underlying framework was able to achieve the modeling of a cardiac cycle using the approach of persistence vineyards. Moreover, the constructed framework achieved to evaluate the potential improvement offered by the TDA descriptors that consider all the time points of a cardiac cycle against considering a few of them.

In particular, there are two promising approaches that worked better in the use of all the time points: the LSTM for time series classification, which is a traditional tool for this purpose, and on the other hand the wavelet processing and CNN classification with a balanced accuracy over 0.6. In the future, there is potential for improvement by integrating with other types of features such as demographic variables (i.e. age, sex and BMI). For a future research project it is worth to explore the mentioned approaches taking care on every detail in the data gathering and processing phases which could be a source of potential errors.

Bibliography

[1] https://gudhi.inria.fr/python/latest/cubical_complex_user.html.

[2] Clasificacion sobre datos desequilibrados tensor flow core — tensorflow.org. https://www.tensorflow.org/tutorials/structured_data/imbalanced_data?hl=es-419#understanding_useful_metrics. [Accessed 02-09-2023].

[3] Colahs blog — colah.github.io. https://colah.github.io/. [Accessed 02-09-2023].

[4] The NIFTI file format — brainder.org. https://brainder.org/2012/09/23/the-nifti-file-format/. [Accessed 02-09-2023].

[5] UK Biobank - UK Biobank — ukbiobank.ac.uk. https://www.ukbiobank.ac.uk/. [Accessed 02-09-2023].

[6] Peter L Bartlett, Cristina Butucea, and Johannes Schmidt-Hieber. Mathematical foundations of machine learning. *Oberwolfach Reports*, 18(1):853–894, 2022.

[7] Julius Berner, Philipp Grohs, Gitta Kutyniok, and Philipp Petersen. The modern mathematics of deep learning. *arXiv preprint arXiv:2105.04026*, pages 86–114, 2021.

[8] Jason Brownlee. *Imbalanced classification with Python: better metrics, balance skewed classes, cost-sensitive learning.* Machine Learning Mastery, 2020.

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

[10] Carles Casacuberta. Topological data analaysis course. *Master of Advanced Mathematics, Universitat de Barcelona*, 2022.

[11] Abigail Hickok. Persistence diagram bundles: A multidimensional generalization of vineyards. *arXiv preprint arXiv:2210.05124*, 2022.

[12] MARTIN Lotz. Mathematics of machine learning. 2018.

[13] Xavier Massaneda. Applied harmonic analysis course. *Master of Advanced Mathematics, Universitat de Barcelona*, 2022.

[14] Elizabeth Munch, Katharine Turner, Paul Bendich, Sayan Mukherjee, Jonathan Mattingly, and John Harer. Probabilistic fréchet means for time varying persistence diagrams. 2015.

[15] Zahra Raisi-Estabragh, Cristian Izquierdo, Victor M Campello, Carlos Martin-Isla, Akshay Jaggi, Nicholas C Harvey, Karim Lekadir, and Steffen E Petersen. Cardiac magnetic resonance radiomics: basic principles and clinical perspectives. *European Heart Journal-Cardiovascular Imaging*, 21(4):349–356, 2020.

[16] Bastian Rieck, Tristan Yates, Christian Bock, Karsten Borgwardt, Guy Wolf, Nicholas Turk-Browne, and Smita Krishnaswamy. Uncovering the topology of time-varying fmri data using cubical persistence. *Advances in neural information processing systems*, 33:6900–6912, 2020.