UNIVERSITÀ DEGLI STUDI DI PADOVA
Dipartimento di Matematica "Tullio Levi-Civita"

Corso di Laurea Magistrale in Matematica

Kernel-based methods for persistent homology and their applications to Alzheimer's Disease

Relatore:
Prof. Stefano De Marchi
Correlatore:
Dott. Francesco Marchetti

Candidato:
Federico Lot
Numero di matricola:
1211259

25 giugno 2021 - Anno Accademico 2020/2021

All that is gold does not glitter,
Not all those who wander are lost
- ***J.R.R. Tolkien***

Acknowledgements

During these months of hard work, I received a lot of support from many people around me.

This thesis would not have arrived up to this point, without all that support, patience and trust.

I sincerely thank all professors I meet over the past three years, who, sharing their passion of unending research and understanding, lead me to truly enjoy the days passed to write and understand the magic behind the following pages.

A special thanks goes to Stefano De Marchi and Francesco Marchetti, who gave me the incredible opportunity to work where different field of mathematics blend, showing that our true limit is imagination. I also thank them for guiding me through this journey, helping me to accept failures and enjoy successes.

Another special thank goes to Davide Poggiali, who, despite its commitments and research, has found precious moments to help me in my first steps in neuroscience with patience and availability.

I thank my parents and my family who gave me all they had[1], even in these difficult times.

Lastly I thanks my friends, who share every day the failures and the successes of my path, always there for me. Always.

[1] especially my brother, who sacrificed his computer to my algorithms.

Abstract

Kernel-based methods are powerful tools that are widely applied in many applications and fields of research. In recent years, methods from computational topology have emerged for characterizing the intrinsic geometry of data. Persistence homology is a central tool in topological data analysis, which allows to capture the evolution of topological features of the data. Persistence diagrams represent a natural way to summarize these features, but they can not be directly used in machine learning algorithms. To deal with them, we first analyse various kernel-based methods of recent development, then we propose and apply Variable Scaled Kernels (VSKs) to the persistence diagrams framework. We therefore discuss the application of these kernels in medical imaging in the context of Alzheimer's Disease classification. Taking into account the cortical thickness measures on the cortical surface, we build the persistence diagrams upon different MRI subjects and we perform some classification tests using the support vector machines classifier.

Introduction

Kernel methods are well-established tools in a variety of research and applied fields, including engineering, machine learning, pattern recognition and in general in many fields of computational mathematics.

In this thesis, we first introduce kernels, their properties and the related Hilbert structure. We recall some notions concerning kernel-based approximation and we present some examples of well-known kernels.
Then, we focus on machine learning and pattern analysis. After an overview about the statistical learning theory, we recall some important qualities of kernels, such as the so-called *kernel trick*, which guarantee the effectiveness of kernel-based methods in representing complex patterns in the data with controlled computational costs. We then introduce the Support Vector Machines (SVMs) classification scheme, which is widely used in applications. This algorithm approaches the learning problem from a geometrical point of view, finding the hyperplanes which better separate the data with respect to the output classes.

In the second part of this work, we give an introduction to Topological Data Analysis, which is an emerging trend in data science to design descriptors for complex data. The core of applied topology is based on persistent homology. After a theoretical presentation, we present some examples in order to visualize the descriptive power of persistent homology by using persistence diagrams, which are effective and stable tools, and focusing on point cloud data. After that, we show how dedicated kernels can be associated to persistence diagrams. These kernels are constructed in order to deal with this peculiar framework and can characterize meaningful similarities from a topological perspective.

In the last part of this manuscript, we apply the introduced kernels in concrete classification tasks and we test their usage in the framework of the so-called Variably Scaled Kernels (VSKs), originally introduced in the field of functions approximation. The application we consider is a dataset of MRI images and we analyse them by means of persistence diagrams built upon the cortical thickness estimates information. Then, considering the presented kernels in the the Support Vector Machines (SVMs) scheme, we show how the constructed classifiers are effective in discriminating between healthy and Alzheimer's Disease (AD) patients.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction to Kernels

In this introductory chapter, we present kernels in the context of approximation theory and we mainly refer to the books [15, 16].

The starting point is the *scattered data fitting problem*. Let $\mathcal{X}$ be the set of the data sites, $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\} \subset \Omega \subset \mathbb{R}^d$ where $n, d \in \mathbb{N}$, and let f be the unknown function to be recovered, which is associated to the known values at the data $y_i = f(\mathbf{x}_i)\ \forall i = 1, ..., n$ with $y_i \in \mathbb{R}$. Our aim is to find a function $s : \Omega \longrightarrow \mathbb{R}$ such that $s(\mathbf{x}_i) = y_i\ \forall i = 1, ..., n$.

To solve this problem, we assume that the function can be expressed as a linear combination of particular functions $B_j : \Omega \to \mathbb{R}\ \ j = 1, ..., n$ called *basis functions*, so that

$$s(\mathbf{x}) = \sum_{j=1}^{n} c_j B_j(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^d, \text{ for some } c_j \in \mathbb{R}\ \ j = 1, ..., n. \tag{1.1}$$

The coefficients c_j are determined by imposing the interpolation conditions

$$s(\mathbf{x}_i) = \ y_i = \sum_{j=1}^{n} B_j(\mathbf{x}_i) c_j.$$

Looking at the well-posedness of the problem, we need to introduce the definition of *Haar system* and we present some fundamental properties useful to understand their role in the context of interpolation.

Definition 1.1 (Haar System). Let the finite-dimensional linear function space $\mathcal{B} \subseteq C(\Omega)$ have a basis $\{B_1, ..., B_N\}$. Then $\mathcal{B}$ is a Haar Space on Ω if $\det B \neq 0$ for any set of distinct $\mathbf{x}_1, ..., \mathbf{x}_n$ in Ω, where B is the matrix with entries $(B)_{i,j} = B_j(\mathbf{x}_i)$. The set $\{B_1, ..., B_N\}$ is then called a Haar system.

Then, we have the following results.

Theorem 1.2. *A set $\{B_1, ..., B_n\}$ of continuous functions on [a,b] forms a Haar system if and only if any non-trivial linear combination of $B_1, ..., B_n$ has at most $n - 1$ zeros in (a, b).*

Theorem 1.3 (Haar-Mairhuber-Curtis). *If $\Omega \subset \mathbb{R}^d$, $d \geq 2$, contains an interior point, then there exist no Haar spaces of continuous functions except for the trivial ones, i.e. spaces spanned by a single function.*

Haar systems are powerful spaces that guarantee the uniqueness of the interpolant, but in the multivariate setting we do not have the powerful result

of Theorem 1.2, which provide the uniqueness of the interpolant without assumptions on the data sites. As we will see in this chapter, the choice of the data sites is important for interpolation purposes.

1.1 Kernels and Native Spaces

In this introduction, we will confine on kernels associated to positive definite matrices. In the following we consider kernels: $\kappa : \Omega \times \Omega \longrightarrow \mathbb{R}, \quad \kappa : (\mathbf{x}, \mathbf{y}) \mapsto \kappa(\mathbf{x}, \mathbf{y})$. Moreover, unless otherwise stated, we consider $\Omega \subset \mathbb{R}^d$ and κ as a real-valued function.

Kernel-based methods are born with the idea of building a point-dependent basis. In the case of the scattered data fitting problem, we can then take the set $\mathcal{B} = \{\kappa(\cdot, \mathbf{x}_1), ..., \kappa(\cdot, \mathbf{x}_n)\}$ with $\mathbf{k}(\mathbf{x}) = (\kappa(\mathbf{x}, \mathbf{x}_1), ..., \kappa(\mathbf{x}, \mathbf{x}_n)) \in \mathbb{R}^d$. Considering Equation (1.1) and $\mathbf{c} = (c_1, ..., c_n)$, we write

$$s(x) = \sum_{j=1}^{n} c_j \kappa(\mathbf{x}, \mathbf{x}_j) = \mathbf{k}(\mathbf{x})^T \mathbf{c}, \quad \mathbf{x} \in \mathbb{R}^d. \tag{1.2}$$

If we denote as K the matrix $K_{i,j} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$ and $\mathbf{y} = (y_1, ..., y_n)$, then can we find the vector of coefficients $\mathbf{c}$ by solving the linear system

$$K\mathbf{c} = \mathbf{y}.$$

It is well-known that the solution of this linear system is unique as long as the matrix is nonsingular.

Definition 1.4 (Gram Matrix). Given a function $\kappa : \Omega \times \Omega \longrightarrow \mathbb{R}$, and data sites $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n \in \mathcal{X} \subset \Omega$, the $n \times n$ matrix K with elements $K_{ij} := \kappa(\mathbf{x}_i, \mathbf{x}_j)$ is called the *Gram matrix* of κ with respect to the point set $\mathbf{x}_1, ..., \mathbf{x}_n$.

Definition 1.5 (Positive (Negative) definite matrix). A real symmetric $n \times n$ matrix K is called positive (negative) definite if its associated quadratic form is positive (negative) for any nonzero coefficient vector $\mathbf{c} = (c_1, ...c_n)^T \in \mathbb{R}^n$, i.e.:

$$\mathbf{c}^T K \mathbf{c} > 0 \, (< 0).$$

The matrix is called positive (negative) semi-definite if the quadratic form is allowed to be nonnegative.

Definition 1.6 (Strictly positive definite kernel). A symmetric kernel κ is called strictly positive definite on $\Omega \times \Omega$ if its associated kernel matrix K with entries $(K)_{i,j} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$, $i, j = 1, ..., n$, is positive definite for any $n \in \mathbb{N}$ and for any set $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\} \in \Omega$ of distinct points.

Definition 1.7 (Positive (negative) definite kernel). A symmetric kernel κ is called positive definite on $\Omega \times \Omega$ if its associated kernel matrix K with entries $(K)_{i,j} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$, $i, j = 1, ..., n$, is positive semi-definite for any $n \in \mathbb{N}$ and for any set $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\} \in \Omega$ of distinct points.

Remark 1.8. Concerning the notation, in the scientific literature, we can find either the notation of *positive semi-definite kernels* or *positive definite kernels* respectively for kernels with associated positive semi-definite or positive definite matrix. In this thesis we use the notation of [15].

The definitions above can also be stated for strictly negative and negative definite kernels considering respectively negative definite and negative semi-definite matrices.

There are many interesting and important relations between positive and negative definite kernels [4].

Lemma 1.9. *Let $\mathcal{X}$ be a non-empty set, $\mathbf{x}_0 \in \mathcal{X}$, and let $\psi : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a symmetric kernel. Put $\kappa(\mathbf{x}, \mathbf{y}) := \psi(\mathbf{x}, \mathbf{x}_0) + \psi(\mathbf{y}, \mathbf{x}_0) - \psi(\mathbf{x}, \mathbf{y}) - \psi(\mathbf{x}_0, \mathbf{x}_0)$. Then κ is positive definite if and only if ψ is negative definite. If $\psi(\mathbf{x}_0, \mathbf{x}_0) \geq 0$ and $\kappa_0(\mathbf{x}, \mathbf{y}) := \psi(\mathbf{x}, \mathbf{x}_0) + \psi(\mathbf{y}, \mathbf{x}_0) - \psi(\mathbf{x}, \mathbf{y})$, then κ_0 is positive definite if and only if ψ is negative definite.*

Theorem 1.10. *Let $\mathcal{X}$ be a nonempty set, and let $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a kernel. Then κ is negative definite if and only if $\exp(-t\,\kappa)$ is positive definite for all $t > 0$.*

Of course, it is trivially true that $-\kappa$ is negative definite whenever κ is positive definite.

Another meaningful definition is the infinitely divisibility of positive definite kernels which is particularly useful in applications, as we will see later in Section 4.1.

Definition 1.11 (Infinitely divisible kernel). A positive definite kernel κ is called *infinitely divisible* if for each $n \in \mathbb{N}$ there exists a positive definite kernel κ_n such that $\kappa = (\kappa_n)^n$.

Proposition 1.12. *Let $\mathcal{N}_\infty(\mathcal{X})$ denote the closure of all real valued negative definite kernels on $\mathcal{X} \times \mathcal{X}$ in the space $]-\infty, \infty]^{\mathcal{X}\times\mathcal{X}}$. Then for a positive definite kernel $\kappa \geq 0$ on $\mathcal{X} \times \mathcal{X}$ the following conditions are equivalent:*

(i) *κ is infinitely indivisible.*

(ii) $-\log(\kappa) \in \mathcal{N}_\infty(\mathcal{X})$.

(iii) *κ^t is positive definite for all $t > 0$.*

We denote the well-known $\mathscr{L}_p$ space as

$$\mathscr{L}_p(\Omega) = \left\{ f : \Omega \longrightarrow \mathbb{R} \,\middle|\, \int_\Omega |f(\mathbf{x})|^p \, dx < \infty \right\}$$

with the associated norm: $\|f\|_{\mathscr{L}_p(\Omega)} = \left(\int_\Omega |f(\mathbf{x})|^p \, dx\right)^{\frac{1}{p}}$.

An alternative definition for positive definite kernels is the following.

Definition 1.13 (Integrally positive definite kernel). A symmetric kernel K is called integrally positive definite on $\Omega \times \Omega$ if $\forall f \in \mathscr{L}_2(\Omega)$

$$\int_\Omega \int_\Omega K(\mathbf{x}, \mathbf{y}) f(\mathbf{x}) f(\mathbf{y}) \, dx \, dy \geq 0.$$

To highlight the analogy between this definition and the other, we recall the definition of self-adjoint operator, positive self-adjoint operator.

Definition 1.14 (Self-adjoint operator). A linear operator $\mathcal{K}$ defined on a linear everywhere-dense set $\mathscr{D}(\mathcal{K})$ in a Hilbert space $\mathcal{H}$ is called self-adjoint if it coincides with its adjoint operator $\mathcal{K}^*$, that is, such that $\mathscr{D}(\mathcal{K}) = \mathscr{D}(\mathcal{K}^*)$ and $\langle \mathcal{K}f, g\rangle = \langle f, \mathcal{K}g\rangle \; \forall f, g \in \mathscr{D}(\mathcal{K})$

Definition 1.15 (Positive self-adjoint operator). A self-adjoint operator $\mathcal{K}$ action on a Hilbert space $\mathcal{H}$ is called positive if $\langle \mathcal{K}f, f\rangle_{\mathcal{H}} \geq 0$, $\forall f \in \mathcal{H}$

Then, considering the operator $(\mathcal{K}f)(\mathbf{x}) = \int_\Omega K(\mathbf{x}, \mathbf{y}) f(\mathbf{y})\, dy$, with $f \in \mathscr{L}_2(\Omega)$, and using the standard $\mathscr{L}_2$ inner product i.e. $\langle f, g\rangle_{\mathscr{L}_2(\Omega)} = \int_\Omega f(\mathbf{x}) g(\mathbf{x})\, dx$, we find that an integrally positive definite kernel is the kernel of a positive integral operator.

Now we introduce series expansion of positive definite kernels [16, Section 2.2].

Definition 1.16 (Hilbert-Schmidt operators). Let $\mathcal{H}$ be a Hilbert space and $\mathcal{T} : \mathcal{H} \longrightarrow \mathcal{H}$ a bounded linear operator. The operator $\mathcal{T}$ is called a Hilbert-Schmidt operator if there exists an orthonormal basis $\{e_n\}_{n=1,\ldots}$ of $\mathcal{H}$ such that

$$\sum_{n=1}^{\infty} \|\mathcal{T} e_n\|_{\mathcal{H}}^2 < \infty.$$

Where $\|\cdot\|_{\mathcal{H}}$ denotes the norm in $\mathcal{H}$ induced by its inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. Usually, $\sum_{n=1}^{\infty} \|\mathcal{T} e_n\|_{\mathcal{H}}^2 = \|\mathcal{T}\|_{HS}^2$ is called Hilbert-Schmidt norm.

Theorem 1.17 (Hilbert-Schmidt integral operator). *Let $\mathscr{L}_2(\Omega, \mu)$ be a Hilbert space on a locally compact Hausdorff space Ω with positive Borel measure μ. Further let the kernel $\kappa : (\mathbf{x}, \mathbf{y}) \mapsto \kappa(\mathbf{x}, \mathbf{y})$ be in $\mathscr{L}_2(\Omega \times \Omega, \mu \times \mu)$, i.e. assume that*

$$\int_\Omega \int_\Omega |\kappa(\mathbf{x}, \mathbf{y})|^2 \, d\mu(\mathbf{x})\, d\mu(\mathbf{y}) < \infty.$$

Then the operator $\mathcal{K}$ defined by

$$(\mathcal{K}f)(\mathbf{x}) = \int_\Omega \kappa(\mathbf{x}, \mathbf{y}) f(\mathbf{y})\, d\mu(\mathbf{y}), \quad f \in \mathscr{L}_2(\Omega, \mu) \tag{1.3}$$

is a Hilbert-Schmidt operator. Conversely, every Hilbert-Schmidt operator on $\mathscr{L}_2(\Omega, \mu)$ is on the form 1.3 for some unique kernel $\kappa : (\mathbf{x}, \mathbf{y}) \mapsto \kappa(\mathbf{x}, \mathbf{y})$ in $\mathscr{L}_2(\Omega \times \Omega, \mu \times \mu)$

Theorem 1.18 (Mercer's theorem). *Let (Ω, μ) be a locally compact Hausdorff space with positive Borel measure μ and $\kappa \in \mathscr{L}_2(\Omega \times \Omega, \mu \times \mu)$ be a kernel such that the Hilbert-Schmidt operator $\mathcal{K} : \mathscr{L}_2(\Omega, \mu) \longrightarrow \mathscr{L}_2(\Omega, \mu)$*

$$(\mathcal{K}f)(\mathbf{x}) = \int_\Omega \kappa(\mathbf{x}, \mathbf{y}) f(\mathbf{y})\, d\mu(\mathbf{y}) \tag{1.4}$$

is positive. Let $\varphi_n \in \mathscr{L}_2(\Omega, \mu)$, $n = 1, 2, \ldots$ be the $\mathscr{L}_2(\Omega, \mu)$ orthonormal eigenfunctions of $\mathcal{K}$ associated to the eigenvalues $\lambda_n > 0$. Then the following are true:

(i) *The kernel has a Mercer expansion*

$$\kappa(\mathbf{x}, \mathbf{y}) = \sum_{n=1}^{\infty} \lambda_n \varphi_n(\mathbf{x}) \varphi_n(\mathbf{y}). \tag{1.5}$$

which holds μ^2 almost everywhere, and converges absolutely and uniformly μ^2 almost everywhere.

(ii) *The eigenvalues $\{\lambda_n\}_{n=1}^{\infty}$ are absolutely summable, and so $\mathcal{K}$ has finite trace.*

Mercer's Theorem grants not only the convergence but the uniform convergence, slightly differentiating by Schmidt's version.

Reproducing Kernel Hilbert Spaces

Following these ideas, we want to further analyse the properties of the interpolation space. We start by presenting the definition of reproducing kernel in the context of Hilbert spaces [15, Chapter 13.1].

Definition 1.19 (Reproducing Kernel). Let $\mathcal{H}$ be a Hilbert space of real-valued functions f defined on $\Omega \subset \mathbb{R}^d$ with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. A kernel $\kappa : \Omega \times \Omega \longrightarrow \mathbb{R}$ is called reproducing kernel for $\mathcal{H}$ if:

(i) $\kappa(\cdot, \mathbf{x}) \subset \mathcal{H} \, \forall \mathbf{x} \in \Omega$.

(ii) $\langle f, \kappa(\cdot, \mathbf{x}) \rangle_{\mathcal{H}} = f(\mathbf{x}) \quad \forall f \in \mathcal{H}, \, \forall \mathbf{x} \in \Omega$.

The name *reproducing kernel* is motivated by the reproducing property (ii), which shows how evaluating the function f on $\mathbf{x}$ is equivalent to considering the inner product between f with $\kappa(\cdot, \mathbf{x})$. Recalling the Riesz representation theorem [36], since for any $\mathbf{x} \in \Omega$ we have a function $\kappa(\cdot, \mathbf{x})$ such that the reproducing property holds, it follows that $\kappa(\cdot, \mathbf{x})$ is the Riesz representative of function evaluation at $\mathbf{x}$. The reproducing kernel of a Hilbert space is unique.

Definition 1.20. Given a kernel function $\kappa : \Omega \times \Omega \to \mathbb{R}$, we let $\mathcal{H}_\kappa(\Omega)$ denote the unique reproducing kernel Hilbert space (RKHS) with reproducing kernel κ.

We remark some useful properties of reproducing kernels on Hilbert spaces.

(i) A reproducing kernel is symmetric, i.e. $\forall \mathbf{x}, \mathbf{y} \in \Omega$

$$\langle \kappa(\cdot, \mathbf{x}), \kappa(\cdot, \mathbf{y}) \rangle_{\mathcal{H}_\kappa(\Omega)} = \kappa(\mathbf{x}, \mathbf{y}) = \kappa(\mathbf{y}, \mathbf{x}).$$

This follows directly by reproducing property and the symmetry of inner product.

(ii) Point evaluation in a RKHS is bounded i.e., $\forall f \in \mathcal{H}_\kappa(\Omega)$, $\mathbf{x} \in \Omega$

$$|f(\mathbf{x})| \leq \sqrt{\kappa(\mathbf{x}, \mathbf{x})} \|f\|_{\mathcal{H}_\kappa(\Omega)}.$$

This follows from the reproducing property and Cauchy-Schwarz inequality to the inner product, added to the fact that $\|\kappa(\cdot, \mathbf{x})\|^2_{\mathcal{H}_\kappa(\Omega)} \geq 0$.

(iii) Convergence in Hilbert space norm implies pointwise convergence i.e. if we have $\|f - f_n\|_{\mathcal{H}_\kappa(\Omega)} \to 0$ for $n \to \infty$, then $|f(\mathbf{x}) - f_n(\mathbf{x})| \to 0 \, \forall \mathbf{x} \in \Omega$.

(iv) If κ is the reproducing kernel of the RKHS $\mathcal{H}$ and $\mathcal{V}$ is a closed subspace of $\mathcal{H}$, then $\mathcal{V}$ is also a RKHS whose reproducing kernel is given in terms of the orthogonal projection $P_{\mathcal{V}}$ onto $\mathcal{V}$, i.e.:

$$\kappa_{\mathcal{V}}(\mathbf{x}, \mathbf{y}) = P_{\mathcal{V}}(\kappa(\cdot, \mathbf{y}))(\mathbf{x}).$$

Finally, we can link reproducing kernel Hilbert spaces to positive definite kernels.

Theorem 1.21. *Suppose $\mathcal{H}_\kappa(\Omega)$ is a reproducing kernel Hilbert function space with reproducing kernel $\kappa : \Omega \times \Omega \longrightarrow \mathbb{R}$. Then κ is positive definite. Moreover, κ is strictly positive definite if and only if the point evaluation functionals δ_x are linearly independent in the dual space $\mathcal{H}_\kappa(\Omega)^*$.*

Now we show that every strictly positive definite kernel can indeed be associated to a reproducing kernel Hilbert space, its *native space* [15, Section 13.2].

The space

$$\mathcal{M}_\kappa(\Omega) = span\{\kappa(\cdot, \mathbf{y}) : \mathbf{y} \in \Omega\} \tag{1.6}$$

equipped with the bilinear form $\langle \cdot, \cdot \rangle_\kappa$,

$$\langle \sum_{i=1}^{N_\kappa} c_i \kappa(\cdot, \mathbf{x}_i), \sum_{j=1}^{N_\kappa} d_j \kappa(\cdot, \mathbf{y}_j) \rangle = \sum_{i=1}^{N_\kappa} \sum_{j=1}^{N_\kappa} c_i d_j \kappa(\mathbf{x}_i, \mathbf{y}_j), \quad N_\kappa \in \mathbb{N} \cup \{+\infty\}. \tag{1.7}$$

for some $c_i, d_i \in \mathbb{R} \, \forall i = 1, ..., N_k$, is a pre-Hilbert space if κ is a symmetric strictly positive definite kernel. Therefore we define the native space $\mathcal{N}_\kappa(\Omega)$ of κ, by completing $\mathcal{M}_\kappa(\Omega)$ with respect to the κ-norm $\| \cdot \|_{\mathcal{M}_\kappa(\Omega)}$, so that $\|f\|_{\mathcal{M}_\kappa(\Omega)} = \|f\|_{\mathcal{N}_\kappa(\Omega)}$ for all $f \in \mathcal{M}_\kappa(\Omega)$.

Here Mercer Theorem 1.18 allows us to build a reproducing kernel Hilbert space $\mathcal{H}_\kappa(\Omega)$ as infinite linear combination of eigenfunctions.

$$\mathcal{H}_\kappa(\Omega) = \{f : f = \sum_{k=1}^{\infty} c_k \, \varphi_k\},$$

where φ_k are the eigenfunctions of $\mathcal{K}$ defined in (1.4). We notice that the kernel κ itself is in $\mathcal{H}_\kappa(\Omega)$ since from Mercer theorem it has the eigenfunction expansion. Lastly, the interpolation space, as subset of the native space, is a reproducing kernel Hilbert space.

Native Space Error Estimates

Now we provide some error estimates for scattered data interpolation with strictly positive definite functions. We want our estimates to depend on some measure of the data distribution [15]. In approximation theory, it is widely used the the *fill distance*,

$$h = h_{\mathcal{X},\Omega} = \sup_{\mathbf{x} \in \Omega} \min_{\mathbf{x}_j \in \mathcal{X}} \|\mathbf{x} - \mathbf{x}_j\|, \tag{1.8}$$

that denotes the radius of the largest empty ball that can be placed among the data sites, as displayed in Figure 1.1. Another measure that we consider is the *separation distance*,

$$q_{\mathcal{X}} = \frac{1}{2} \min_{i \neq j} \|\mathbf{x}_i - \mathbf{x}_j\|, \tag{1.9}$$

that denotes the maximum radius of the balls centered in the data points, such that each ball is disjoint by the others.

We introduce the cardinal functions.

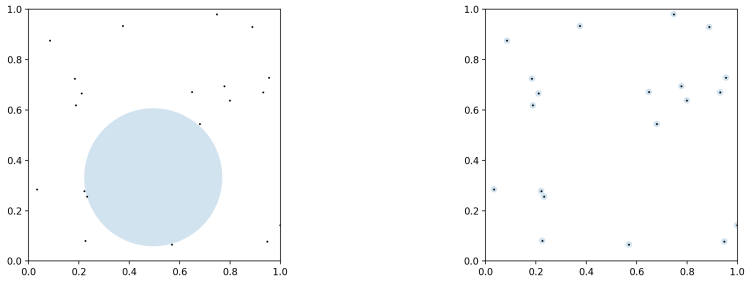

Figure 1.1: The fill distance of a random set of points in $[0,1]^2$ (left) and their separation distance (right)

Theorem 1.22. *Suppose κ is a strictly positive definite kernel on $\mathbb{R}^d$, then for any distinct points $\mathbf{x}_1, ..., \mathbf{x}_n$, $n \in \mathbb{N}$, there exist functions $u_j^* \in span\{\kappa(\cdot, \mathbf{x}_j),\ j = 1, ..., n\}$ such that $u_j^*(\mathbf{x}_i) = \delta_{ij}$ for every $j = 1, ..., n$. Where δ_{ij} is the Kronecker delta.*

Therefore, we can write the interpolant $\mathcal{P}_f$ at $\mathbf{x}_1, ..., \mathbf{x}_n$ in the cardinal form

$$\mathcal{P}_f(\mathbf{x}) = \sum_{j=1}^{N} f(\mathbf{x}_j) u_j^*(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^d. \tag{1.10}$$

Then, we define the power function [15, Section 14.3].

Definition 1.23 (Power function). Suppose $\Omega \subset \mathbb{R}^d$ and $\kappa \in \mathscr{C}(\Omega \times \Omega)$ is strictly positive definite on $\mathbb{R}^d$. For any distinct points $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\} \subset \Omega$ the *power function* is defined by

$$[P_{\kappa,\mathcal{X}}(\mathbf{x})]^2 = Q(\mathbf{u}^*(\mathbf{x})), \tag{1.11}$$

where $\mathbf{u}^*$ is the vector of cardinal function from Theorem 1.22, and $Q(\mathbf{u})$ is the following quadratic form:

$$Q(\mathbf{u}) = \kappa(\mathbf{x}, \mathbf{x}) - 2\sum_{j=1}^{N} u_j \kappa(\mathbf{x}, \mathbf{x}_j) + \sum_{i=1}^{N}\sum_{j=1}^{N} u_i u_j \kappa(\mathbf{x}_i, \mathbf{x}_j). \tag{1.12}$$

The power function has an alternative representation.

$$P_{\kappa,\mathcal{X}}(\mathbf{x}) = \sqrt{\kappa(\mathbf{x}, \mathbf{x}) - (\mathbf{b}(\mathbf{x}))^T A^{-1} \mathbf{b}(\mathbf{x})}, \tag{1.13}$$

where $A_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$ for $i, j = 1, ..., n$, and $\mathbf{b} = [\kappa((\cdot, \mathbf{x}_1), ..., \kappa(\cdot, \mathbf{x}_n)]^T$. Since A is a positive definite matrix whenever κ is strictly positive, we have a bound for the power function:

$$0 \leq P_{\kappa,\mathcal{X}}(\mathbf{x}) \leq \sqrt{\kappa(\mathbf{x}, \mathbf{x})}.$$

Therefore, we can give the following error estimates in terms of the power function.

Theorem 1.24. *Let $\Omega \subset \mathbb{R}^d$, $\kappa \in \mathscr{C}(\Omega \times \Omega)$ be strictly positive definite on $\mathbb{R}^d$, and suppose that the points $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ are distinct. Denote the interpolant of $f \in \mathcal{N}_\kappa(\Omega)$ on $\mathcal{X}$ by $\mathcal{P}_f$. Then for every $\mathbf{x} \in \Omega$*

$$|f(\mathbf{x}) - \mathcal{P}_f(\mathbf{x})| \leq \|f\|_{\mathcal{N}_\kappa(\Omega)} P_{\kappa,\mathcal{X}}(\mathbf{x}). \tag{1.14}$$

The strength of this theorem it that we are now able to estimate the interpolation error by considering two independent phenomena:

1. the smoothness of the data, measured in terms of the native space norm of f (which is independent of the data locations);
2. the contribution of the specific kernel κ and the distribution of the data, measured in term of the power function (which is independent of data values).

Theorem 1.25. *Let $\Omega \subset \mathbb{R}^d$, and suppose $\kappa \in \mathscr{C}(\Omega \times \Omega)$ is strictly positive definite on $\mathbb{R}^d$. Let $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ a set of distinct points in Ω, and define the quadratic form $Q(\mathbf{u})$ as in 1.12. The minimum of $Q(\mathbf{u})$ is given for the vector $\mathbf{u} = \mathbf{u}^*(x)$ from theorem 1.22 i.e.,*

$$Q(\mathbf{u}^*(\mathbf{x})) \leq Q(\mathbf{u}), \qquad \text{for all } \mathbf{u} \in \mathbb{R}^n.$$

We introduce the following notation: for $\beta = (\beta_1, ..., \beta_d) \in \mathbb{N}_0^d$ with $|\beta| = \sum_{i=1}^s \beta_i$ we define the differential operator D^β as:

$$D^\beta = \frac{\partial^{|\beta|}}{(\partial \mathbf{x}_1)^{\beta_1} \dots (\partial \mathbf{x}_d)^{\beta_d}}.$$

Moreover, we define what follows.

Definition 1.26. A region $\Omega \subset \mathbb{R}^d$ satisfies an *interior cone condition* if there exists an angle $\theta \in (0, \frac{\pi}{2})$ and a radius $r > 0$ such that for every $\mathbf{x} \in \Omega$ there exists a unit vector $\chi(\mathbf{x})$ such that the cone

$$C = \{\mathbf{x} + \lambda \mathbf{y} : \mathbf{y} \in \mathbb{R}^d, \|\mathbf{y}\|_2 = 1, \mathbf{y}^T \chi(\mathbf{x}) \geq cos\theta, \lambda \in [0, r]\}$$

is contained in Ω.

Thus, we can finally include the fill distance in Theorem 1.24, highlighting the contribute of the data sites on the bound.

Theorem 1.27. *Suppose $\Omega \subset \mathbb{R}^d$ is bounded and satisfying an interior cone condition and suppose $\kappa \in \mathscr{C}^{2k}(\Omega \times \Omega)$ is symmetric and strictly positive definite. Denote the interpolant to $f \in \mathcal{N}_\kappa(\Omega)$ on the set $\mathcal{X}$ by $\mathcal{P}_f$. Then, there exist positive constants h_0 and C (independent of $\mathbf{x}$, f and κ) such that*

$$|f(\mathbf{x}) - \mathcal{P}_f(\mathbf{x})| \leq C h_{\mathcal{X},\Omega}^k \sqrt{C_\kappa(\mathbf{x})} \|f\|_{\mathcal{N}_\kappa(\Omega)} \tag{1.15}$$

provided $h_{\mathcal{X},\Omega} \leq h_0$. Here

$$C_\kappa(\mathbf{x}) = \max_{|\beta|=2k} \max_{\mathbf{w},\mathbf{z} \in \Omega \cap B} |D_2^\beta \kappa(\mathbf{w}, \mathbf{z})|,$$

where $B = B(\mathbf{x}, c_2 h_{\mathcal{X},\Omega})$ denote the ball of radius $c_2 h_{\mathcal{X},\Omega}$ for some $c_2 \in \mathbb{R}$, centered in $\mathbf{x}$.

Moreover if we fix $\alpha \in \mathbb{N}_0^d$, with $|\alpha| \leq k$, we have that

$$|D^\alpha f(\mathbf{x}) - D^\alpha \mathcal{P}_f(\mathbf{x})| \leq C h_{\mathcal{X},\Omega}^{k-|\alpha|} \sqrt{C_\kappa(\mathbf{x})} \|f\|_{\mathcal{N}_\kappa(\Omega)},$$

with

$$C_\kappa(\mathbf{x}) = \max_{\substack{\beta,\gamma \in \mathbb{N}_0^d \\ |\beta|+|\gamma|=2k}} \max_{\mathbf{w},\mathbf{z} \in \Omega \cap B} |D_1^\beta D_2^\gamma \kappa(\mathbf{w},\mathbf{z})|.$$

Here $B = B(\mathbf{x}, c_2 h_{\mathcal{X},\Omega})$ denote the ball of radius $c_2 h_{\mathcal{X},\Omega}$ for some $c_2 \in \mathbb{R}$, centered in $\mathbf{x}$. Estimates built by using the separation distance $q_{\mathcal{X}}$ are used mainly for error bounds for functions outside the native space [29].

In approximation theory, for measuring the stability of a method, we need to look at the condition number of the interpolation matrix.

Definition 1.28. Let be A a matrix. its ℓ_2-condition number (usually referred to simply as the matrix condition number) is given by:

$$\text{cond}(A) = \|A\|_2 \|A^{-1}\|_2 = \frac{\sigma_{\max}}{\sigma_{\min}}.$$

Here $\sigma_{\min}$ and $\sigma_{\max}$ are respectively the smallest and the largest singular value of A. If A is a positive or negative definite matrix, then we can compute the condition number as the ration between the highest and the lowest eigenvalues.

$$\text{cond}(A) = \frac{\lambda_{\max}}{\lambda_{\min}}$$

1.2 Radial Basis Functions

In this section, we show some of the most used radial kernels, that later on will be important for the construction of more specific kernels. A radial kernel is of the form $\kappa(\mathbf{x},\mathbf{y}) = \phi(\|\mathbf{x}-\mathbf{y}\|)$ with $\phi : [0,\infty) \longrightarrow \mathbb{R}$, i.e. it is invariant for for both translations and rotations, the function ϕ is called *Radial Basis Function* (RBF). In the following we use the notation $r = \|\mathbf{x}-\mathbf{y}\|$, taking the Euclidean norm in the following examples, and we consider a shape parameter ε.

We recall an important theorem [15, Section 15] that along with Theorem 1.27 gives us the tools to derive more accurate error bounds.

Theorem 1.29. *Let be Ω a cube in $\mathbb{R}^d$. Suppose that $\kappa = \phi(\|\cdot\|)$ is a strictly conditionally positive definite radial function such that $\psi = \phi(\sqrt{\cdot})$ satisfies $|\psi^{(l)}(r)| \leq l! M^l$ for all integers $l \geq l_0$ and all $r \geq 0$, where M is a fixed positive constant. Then there exists a constant c such that for any $f \in \mathcal{N}_\kappa(\Omega)$:*

$$\|f - \mathcal{P}_f\|_{\mathscr{L}_\infty(\Omega)} \leq \exp\left(\frac{-c}{h_{\mathcal{X},\Omega}}\right) \|f\|_{\mathcal{N}_\kappa(\Omega)}, \tag{1.16}$$

for all data sites $\mathcal{X}$ with sufficiently small fill distance $h_{\mathcal{X},\Omega}$.

Moreover, if ψ satisfies even $|\psi^{(l)}(r)| \leq M^l$, then, provided $h_{\mathcal{X},\Omega}$ is sufficiently small, we have

$$\|f - \mathcal{P}_f\|_{\mathscr{L}_\infty(\Omega)} \leq \exp\left(\frac{-c|\log h_{\mathcal{X},\Omega}|}{h_{\mathcal{X},\Omega}}\right) \|f\|_{\mathcal{N}_\kappa(\Omega)}. \tag{1.17}$$

Concerning the numerical stability, recalling the Gershgorin theorem [44], we know that

$$|\lambda_{\max} - K_{ii}| \leq \sum_{\substack{i=j \\ i \neq j}} |K_{ij}|,$$

for some $i \in \{1, ..., n\}$. Therefore, considering $K_{ij} = \phi(x_i - x_j)$:

$$\lambda_{\max} \leq n \cdot \max_{i,j=1,...,n} |K_{ij}| = n \cdot \max_{x_i, x_j \in \mathcal{X}} |\phi(x_i - x_j)| \leq n \cdot \phi(0).$$

To find a lower bound for $\lambda_{\min}$, we consider the *Rayleigh quotient*

$$\lambda_{\min} = \min_{\mathbf{c} \in \mathbb{R}^n \setminus \{\mathbf{0}\}} \frac{\mathbf{c}^T K \mathbf{c}}{\mathbf{c}^T \mathbf{c}}.$$

Some near optimal bounds can be found [15, Section 15] For example, for the Gaussian $\phi(r) = \exp(-\varepsilon^2 r^2)$,

$$\lambda_{\min} \geq C_d (\sqrt{2}\varepsilon)^{-d} \exp\big(-\frac{40.71 d^2}{q_{\mathcal{X}} \varepsilon}\big) q_{\mathcal{X}}^{-s},$$

with C_d constant. We see that, for a fixed shape parameter ε, the lower bound for $\lambda_{\min}$ goes exponentially to zero as the separation distance $q_{\mathcal{X}}$ decreases. Since we observed that the condition number of the matrix K depends on the ratio of its largest and smallest eigenvalues and the growth of $\lambda_{\max}$ is of order n, we see that the condition number grows exponentially with the decrease of the separation distance. This shows that, if one adds more interpolation points in order to improve the accuracy of the interpolant, then the problem becomes ill-conditioned.

This observations lead us to the so called *uncertainty principle*. This principle states that if we use the standard approach to the RBF interpolation problem a conflict between theoretically achievable accuracy and numerical stability occurs. For well-distributed data a decrease in the fill distance also implies a decrease of the separation distance. But now the condition estimated above imply that the condition number of K grows exponentially. This leads to numerical instabilities which make virtually impossible to obtain the accurate results promised by the theoretical error bounds.

Moreover, Schaback [38] looked at the power function $P_{\phi,\mathcal{X}}$ and showed that it can always be bounded from above by a function F_ϕ depending on the fill distance. On the other hand, he showed that the Rayleigh quotient can always be bounded from below by a function G_ϕ depending on the separation distance. Furthermore, he showed that

$$G_\phi(q_{\mathcal{X}}) \leq F_\phi(h_{\mathcal{X},\Omega})$$

and therefore, for well-distributed data (with $q_{\mathcal{X}} \approx h_{\mathcal{X},\Omega}$), a small error bound (i.e., small $F_\phi(h_{\mathcal{X},\Omega})$) will necessarily result in a small lower bound (i.e., small $G_\phi(q_{\mathcal{X}})$) for the Rayleigh quotient, and therefore for the smallest eigenvalue. This however implies a large condition number.

Gaussian kernel

We consider firstly the Gaussian RBF:

$$\phi(r) = e^{-\varepsilon^2 r^2}, \tag{1.18}$$

which is clearly $\mathscr{C}^\infty(\mathbb{R})$ smooth. In Figure 1.2 we plot the Gaussian function for different values of the shape parameter in $[-1,1]^2$.

In the last section we focused on the analysis of the strictly positive definite kernels, since the resulting linear system is well-posed. Now, in the contest of radial functions, we recall an useful theorem about the characterization of strictly positive definite functions.

Definition 1.30 ((Strictly) Positive definite function). A real valued continuous function $\phi : \mathbb{R}^d \to \mathbb{R}$ is called *positive definite* on $\mathbb{R}^d$ if

$$\sum_{j=1}^{n}\sum_{i=1}^{n} c_j c_i \phi(\mathbf{x}_j - \mathbf{x}_i) \geq 0 \tag{1.19}$$

for any n pairwise different points $\mathbf{x}_1, ..., \mathbf{x}_n \in \mathbb{R}^d$, and $\mathbf{c} = (c_1, ..., c_n)^T \in \mathbb{R}^n$. The function ϕ is called *strictly positive definite* on $\mathbb{R}^d$ if the quadratic form 1.19 is zero only for $\mathbf{c} \equiv \mathbf{0}$.

Theorem 1.31. *A continuous function $\phi : [0,\infty) \to \mathbb{R}$ such that $r \mapsto r^{d-1}\phi(r) \in \mathscr{L}_1[0,\infty)$ is strictly positive definite and radial on $\mathbb{R}^d$ if and only if the d-dimensional Fourier transform*

$$\mathcal{F}_d\phi(r) = \frac{1}{\sqrt{r^{d-2}}}\int_0^\infty \phi(t) t^{\frac{d}{2}} J_{\frac{d-2}{2}}(rt)\, dt.$$

Here $J_{\frac{d-2}{2}}$ is the classical Bessel function of the first kind of order $\frac{d-2}{2}$, is non-negative and not identically equal to zero.

Thanks to this result, we can show that the Gaussian is strictly positive definite on $\mathbb{R}^d$ for all d. In fact, the Fourier transform of a Gaussian is a Gaussian:

$$\hat{\phi}(\omega) = \frac{1}{(\varepsilon\sqrt{2})^d} e^{-\frac{\|\omega\|^2}{4\varepsilon^2}}$$

Furthermore we can apply Theorem 1.29 and the error bound (1.16) to the Gaussian $\phi(x) = e^{-\varepsilon^2\|x\|^2}$, with $\varepsilon > 0$ fixed. Since $\psi(r) = e^{-\varepsilon^2 r}$, then $\psi^{(l)}(r) = (-1)^l \varepsilon^{2l} e^{-\varepsilon^2 r}$ for $l \geq l_0 = 0$, so lastly $M = \varepsilon^2$.

Matérn kernel

Another example of strictly positive definite radial function, is given by the class of *Matérn functions*:

$$\phi(r) = \frac{(\varepsilon r)^{\beta-\frac{d}{2}} K_{\beta-\frac{d}{2}}(\varepsilon r)}{2^{\beta-1}\Gamma(\beta)}, \quad \beta > \frac{d}{2} \quad \varepsilon > 0. \tag{1.20}$$

Here K_ν is the *modified Bessel function of the second kind of order ν*. The Fourier transform of the Matérn functions is given by the *Bessel kernels*

$$\hat{\phi}(\omega) = (1 + \|\omega\|^2)^{-\beta} > 0.$$

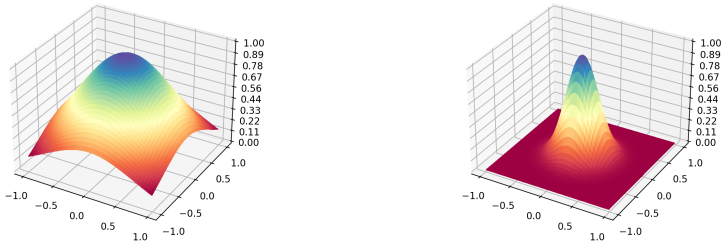

Figure 1.2: Gaussian kernel, with shape parameter $\varepsilon = 1$ (left) and $\varepsilon = 3$ (right)

Thus the Matérn kernels are strictly positive on $\mathbb{R}^d$ for $d < 2\beta$ and their smoothness changes depending on the parameter β as we can see in Table 1.1:

β	RBF	smoothness
$\frac{d+1}{2}$	$e^{-\varepsilon r}$	$\mathscr{C}^0$
$\frac{d+3}{2}$	$e^{-\varepsilon r}(1+\varepsilon r)$	$\mathscr{C}^2$
$\frac{d+5}{2}$	$e^{-\varepsilon r}(3+3(\varepsilon r)+(\varepsilon r)^2)$	$\mathscr{C}^4$

Table 1.1: Matérn functions for various choices of β, and their smoothness

These functions are also called *Sobolev splines*, because their native space is the Sobolev space $\mathscr{W}_2^\beta(\mathbb{R}^d)$, where

$$\mathscr{W}_q^p(\Omega) = \{f \in \mathscr{L}_q(\Omega) \cap \mathscr{C}(\Omega) : D^\alpha f \in \mathscr{L}_q(\Omega) \text{ for all } |\alpha| \leq p\}. \tag{1.21}$$

Moreover we have the following error estimate [15, Section 15]

$$|D^\alpha f(\mathbf{x}) - D^\alpha \mathcal{P}_f(\mathbf{x})| \leq C h_{\mathcal{X},\Omega}^{\beta - \frac{d}{2} - |\alpha|} \|f\|_{\mathcal{N}_\kappa(\Omega)}. \tag{1.22}$$

Provided that $|\alpha| \leq \beta - \lceil \frac{d+1}{2} \rceil$, $h_{\mathcal{X},\Omega}$ is sufficiently small, and $f \in \mathcal{N}_\kappa(\Omega)$.

Two examples are displayed in Figure 1.3: the functions are centered in the origin of $\mathbb{R}$ and evaluated in $[-1,1]^2$. The shape parameter is set to $\varepsilon = 1$ for both functions. Concerning β, we change the values to highlight the difference in terms of smoothness.

Inverse Multiquadric kernel

Since in last section both the functions ϕ and $\hat{\phi}$ were positive and radial, we can use Hankel inversion theorem to switch their roles. We find the so-called *generalized inverse multiquadrics*

$$\phi(r) = (1+\varepsilon^2 r^2)^{-\beta}, \quad \beta > \frac{d}{2}, \quad \varepsilon > 0. \tag{1.23}$$

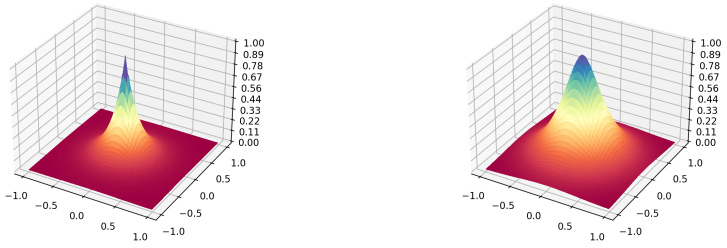

Figure 1.3: Matérn kernel, with $\beta = \frac{3}{2}$ (left) and $\beta = \frac{5}{2}$ (right)

Then, the function is strictly positive definite on $\mathbb{R}^d$ for $d < 2\beta$. It can be shown that we need only $\beta > 0$ for the generalized inverse multiquadrics to be strictly positive on $\mathbb{R}^d$ for any d. Moreover these functions are infinitely differentiable.

As in the previous examples, we can apply the error estimate (1.17) [15, Section 15]. In fact, for the generalized multiquadric $\psi(r) = (1 + r)^\beta$, we can show that $|\psi^l(r)| \leq l! M^l$ whenever $l \geq \lceil \beta \rceil$. Here $M = 1 + |\beta + 1|$.

Figure 1.4 display the well-known inverse multiquadric and inverse quadratic, respectively for β equal to $\frac{1}{2}$ and 1. In these examples we have fixed the shape parameter ε equal to 5.

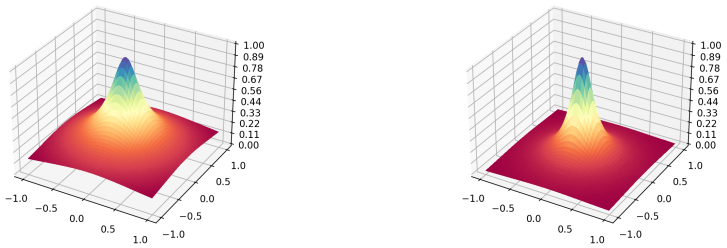

Figure 1.4: Generalized inverse multiquadric kernel, with $\beta = \frac{1}{2}$ (left, also called inverse multiquadric) and $\beta = 1$ (right, also called inverse quadratic)

Variably Scaled Kernel

In kernel-based approximation, the tuning of the *shape parameter* might be a problem. In [5], the authors manage to overcome it through the introduction of Variably Scaled Kernels (VSKs). The idea is to define a scale function c to

transform the interpolation problem on the domain $\Omega \subset \mathbb{R}^d$ at data $\mathbf{x}_i$ to data $(\mathbf{x}_j, c(\mathbf{x}_j))$ on $\mathbb{R}^{d+1}$.

Since the aim is the scale function, for simplify the notation the shape parameter is fixed $\varepsilon = 1$.

Definition 1.32. Let κ be a strictly positive definite kernel on $\mathbb{R}^{d+1}$. If a *scale function* $c : \mathbb{R}^d \longrightarrow (0, \infty)$ is given, we can define a VSK on $\mathbb{R}^d$ by:

$$\kappa_c(\mathbf{x}, \mathbf{y}) := \kappa((\mathbf{x}, c(\mathbf{x})), (\mathbf{y}, c(\mathbf{y}))), \qquad \text{for all } \mathbf{x}, \mathbf{y} \in \mathbb{R}^d. \tag{1.24}$$

The following holds true.

Theorem 1.33. *If κ is positive definite, then κ_c is positive definite, moreover if κ is strictly positive definite, then κ_c is strictly positive definite.*

Now, considering κ a strictly positive definite kernel, interpolation of values $f(\mathbf{x}_1), ..., f(\mathbf{x}_n)$ on data sites $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$, we proceed as usual, solving the linear system:

$$A_{c,\mathcal{X}}\, \mathbf{a} = f \in \mathbb{R}^d$$

with $A_{c,\mathcal{X}} := (\kappa_c(\mathbf{x}_i, \mathbf{x}_j))_{1 \le i,j \le n}$, the kernel matrix which is positive definite and a coefficient vector $\mathbf{a}$. Then we can express the interpolant

$$s_{c,\mathcal{X},f}(\mathbf{x}) := \sum_{j=1}^{N} a_j \kappa_c(\mathbf{x}, \mathbf{x}_j) = \sum_{j=1}^{N} a_j \kappa((\mathbf{x}, c(\mathbf{x})), (\mathbf{x}_j, c(\mathbf{x}_j))). \tag{1.25}$$

In the case of radial kernel i.e. $\kappa(\mathbf{x}, \mathbf{y}) = \phi(\|\mathbf{x} - \mathbf{y}\|)$,

we find $\kappa_c(\mathbf{x}, \mathbf{y}) = \phi(\sqrt{\|\mathbf{x} - \mathbf{y}\|^2 + |c(\mathbf{x}) - c(\mathbf{y})|^2})$ and

$$s_{c,\mathcal{X},f}(\mathbf{x}) = \sum_{j=1}^{N} a_j \phi(\|\mathbf{x} - \mathbf{x}_j\|_2^2 + (c(\mathbf{x}) - c(\mathbf{x}_j))^2).$$

In particular, if we consider κ as the Gaussian kernel, then

$$\kappa_c(\mathbf{x}, \mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|_2^2} e^{-(c(\mathbf{x})-c(\mathbf{y}))^2}$$

is the scaled kernel, and the interpolant is in the form

$$s_{c,\mathcal{X},f}(\mathbf{x}) = \sum_{j=1}^{N} a_j e^{-\|\mathbf{x}-\mathbf{x}_j\|_2^2} e^{-(c(\mathbf{x})-c(\mathbf{x}_j))^2}.$$

Considering the map

$$C : \mathbf{x} \mapsto (\mathbf{x}, c(\mathbf{x})) \tag{1.26}$$

we can show, being L the Lipschitz constant of c, that

$$\|\mathbf{x} - \mathbf{y}\|_2^2 \le \|C(\mathbf{x}) - C(\mathbf{y})\|_2^2 \le \|\mathbf{x} - \mathbf{y}\|_2^2 (1 + L)^2.$$

then if we apply varying scale technique (VSK) on points $\mathbf{x}_i$ and $\mathbf{x}_j$ such that

$$\|\mathbf{x}_i - \mathbf{x}_j\|_2 = q_{\mathcal{X}} \ll h_{\mathcal{X},\Omega},$$

The scales should then vary like

$$|c(\mathbf{x}_i) - c(\mathbf{x}_j)| \approx h_{\mathcal{X},\Omega} \gg q_{\mathcal{X}}.$$

This should be done for all close–by pairs of centers until one roughly gets that

$$q_{C(\mathcal{X})} \approx h_{\mathcal{X},\Omega} \approx h_{C(\mathcal{X}),C(\Omega)}.$$

holds. In particular we have that the transformed centers are approximately uniformly distributed and:

$$q_{C(\mathcal{X})} \gg q_{\mathcal{X}}$$

This result suggest us to use VSK to improve the stability of the interpolation process and to work on points that may bring stability issues.

Lastly in [5], it is proved the following theorem:

Theorem 1.34. *Given a kernel* $\kappa : \Omega \times \Omega \longrightarrow \mathbb{R}$ *and a bijective map* $C : \Omega \mapsto C(\Omega)$*, the kernel*

$$\kappa_C(C(\mathbf{x}), C(\mathbf{y})) := \kappa(\mathbf{x}, \mathbf{y}) \quad \textit{for all } \mathbf{x}, \mathbf{y} \in \Omega$$

now acts on $C(\Omega)$ *and inherits the definiteness properties of* κ*. Moreover the native space for* κ *and* κ_C *are isometric.*

CHAPTER 2

Learning with Kernels

Pattern analysis deals with the automatic detection of patterns in data, and plays a central role in many modern artificial intelligence and computer science problems. By patterns we understand any relations, regularities or structures inherent in some source of data. By detecting significant patterns in the available data, a system can expect to make predictions about new data that come from the same source. In this sense, the system has acquired generalisation power by *'learning'* something about the source generating the data [41].

For the purposes of this thesis, we present the supervised approach to pattern analysis, in particular focusing on the binary case and mainly referring to [39, 41].

2.1 Elements of statistical learning

Let $\Omega \subset \mathbb{R}^d$, and $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\} \subset \Omega$ be a set of input data, $d, n \in \mathbb{N}$. Each data $\mathbf{x}_i$ is associated to a label $y_i \in \mathcal{Y}$, with $\mathcal{Y} = \{-1, +1\}$. Thus we can consider the labeled set of the couples $(\mathbf{x}_i, y_i)$. The binary supervised learning task consists in finding a non-trivial function $f : \Omega \to \mathcal{Y}$ such that it models the input-output relation in the available data and in the unseen labeled data $\xi_i \in \Omega \setminus \mathcal{X}$.

As expected, the problem is not in satisfying the relation $f(\mathbf{x}_i) = y_i$ but in being able to predict in a satisfactory way the others data. To better explain what we mean with 'satisfactory' we consider the so-called *loss function*.

Definition 2.1 (Loss function). Denote by $(\mathbf{x}, y, f) \in \Omega \times \mathcal{Y} \times \Omega^*$ the triplet consisting of a pattern $\mathbf{x}$, an observation y and a prediction function f. Then the map $L : \Omega \times \mathcal{Y} \times \Omega^* \to [0, \infty)$, such that $L(\mathbf{x}, y, f) = 0$ for all couples of labeled data $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$, is called loss function.

In the binary classification case, we consider e.g.

$$L(\mathbf{x}, y, f) = \frac{1}{2}|f(\mathbf{x}) - y|. \tag{2.1}$$

In the following we consider two different approaches to the problem. The first goes in the direction of the Vapnik-Chervonenkis dimension [39], the second one involves the Rademacher complexity [41].

Let us suppose that there exists a probability distribution $P(\mathbf{x}, y)$ on $\mathcal{X} \times \mathcal{Y}$ which governs the data generation, and training and test data are both drawn independently and identically distributed (iid).

Definition 2.2 (Expected risk). If we have no knowledge about the test patterns (or decide to ignore them) we should minimize the expected error over all possible training patterns. Hence we have to minimize the expected loss with respect to P and L

$$R[f] = \mathbb{E}[L(\mathbf{x}, y, f)] = \int_{\mathcal{X}\times\mathcal{Y}} L(\mathbf{x}, y, f)\, dP(\mathbf{x}, y). \tag{2.2}$$

$R[f]$ is called *expected risk*.

Since we do not know $P(\mathbf{x}, y)$ explicitly and all we have is the training data, we consider instead the *empirical risk*:

$$R_{\text{emp}}[f] := \frac{1}{m} \sum_{i=1}^{m} L(\mathbf{x}_i, y_i, f) \tag{2.3}$$

Furthermore, being m the cardinality of the training data, it can be proved that

$$P\big(|R_{\text{emp}}[f] - R[f]\big) \geq \varepsilon] \leq 2\exp(-2m\varepsilon^2).$$

For any fixed function the training error thus provides an estimate of the test error. Moreover, the convergence in probability $R_{\text{emp}}[f] \to R[f]$ as $m \to \infty$ is exponentially fast in the number of training examples. Unfortunately this relation is useful only when the sample size m is sufficiently large.

What we need for empirical risk minimization to work is *consistency*. It amounts to say that, as the number of examples m tends to infinity, we want the function f that minimizes R to provide a test error which converges to the lowest achievable value. It turns out that without restricting the set of admissible functions, empirical risk minimization is not consistent [39].

Moreover it can be proved the following theorem, which underlines the dependency of the result on the class of functions $\mathcal{F}$.

Theorem 2.3. *One-sided uniform convergence in probability,*

$$\lim_{m\to\infty} P\big(\sup_{f\in\mathcal{F}}(R[f] - R_{emp}[f]) > \varepsilon\big) = 0 \tag{2.4}$$

for all $\varepsilon > 0$, is a necessary and sufficient condition for nontrivial consistency of empirical risk minimization.

Thus the next step is to find a suitable class of functions $\mathcal{F} \ni f$ such that we can minimize $R_{\text{emp}}[f]$ with respect to $\mathcal{F}$. Unfortunately, determining $\mathcal{F}$ could be difficult and the minimization of the error can lead to ill-posedness problems. For example, one possible source of problems is the condition number of the matrix $F := (f_i(\mathbf{x}_j))_{i,j}$.

A subsequent direction in this analysis, instead of finding a suitable class of functions, is to restrict the class of admissible solutions, taking into account specific properties. Some possible strategies might be to narrow down to a

compact set, or add a regularization term to the original object function $R_{\text{emp}}[f]$. A connection between support vector machines and regularization operators can be established, which can provide some insight on why support vector machine and other kernel algorithms have been found to exhibit high generalization capability [39].

Since Theorem 2.4 gives us conditions for the consistency of empirical risk minimization, our aim is to find some simpler equivalent formulations to deal with. Some concepts are needed to further analyse these possibilities.

If $\mathcal{F} = \{f_1, ..., f_n\}$, letting

$$C_\varepsilon^i := \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_m, y_m) \,|\, (R[f_i] - R_{\text{emp}}[f_i] > \varepsilon\},$$

then we have that

$$P\big(\sup_{f\in\mathcal{F}}(R[f] - R_{\text{emp}}[f]) > \varepsilon\big) = P\big(C_\varepsilon^1 \cup ... \cup C_\varepsilon^n\big) \leq \sum_{i=1}^{n} P(C_\varepsilon^i). \tag{2.5}$$

This inequality is called *union bound*, where the equality holds only if all events $P(C_\varepsilon^i)$ are disjoint. Thus, if we have a finite function class we can bound the left hand side of 2.5, using law of large numbers to have a constant on the right hand side of the bound.

Lemma 2.4 (VC Symmetrization). *For $m\varepsilon^2 \geq 2$, we have*

$$P\big(\sup_{f\in\mathcal{F}}(R[f] - R_{emp}[f]) > \varepsilon\big) \leq 2P\big(\sup_{f\in\mathcal{F}}(R_{emp} - R'_{emp}[f]) > \frac{\varepsilon}{2}\big), \tag{2.6}$$

where the first P refers to the distribution of iid samples of size m, while the second one refers to iid samples of size 2m. In the latter case R_{emp} measures the loss on the first half of the sample, and R'_{emp} on the second half.

The Lemma 2.4 implies that the function class $\mathcal{F}$ is finite: restricted to the $2m$ points of the right hand side of 2.6, it has at most 2^{2m} elements, since the possible outputs are ± 1 for every pattern with $2m$ elements.

Let $Z_{2m} = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_{2m}, y_{2m})\}$ be the given set of input-output couples of the pattern. Denote with $\mathcal{N}(\mathcal{F}, Z_{2m})$ the cardinality of $\mathcal{F}$ when restricted to $\{\mathbf{x}_1, ..., \mathbf{x}_{2m}\}$, and with $\mathcal{N}(\mathcal{F}, 2m)$ the maximum number of function that can be distinguished on it. $\mathcal{N}(\mathcal{F}, m)$ is also known in this framework as the *shatter coefficient*. It is particularly relevant since it measures the number of ways that the functions in $\mathcal{F}$ can separate the patterns in the two classes.

Lastly using a famous inequality due to Chernoff,

$$P\left\{\left|\frac{1}{m}\sum_{i=1}^{m}\xi_i - \mathbb{E}(\xi)\right| \geq \varepsilon\right\} \leq 2\exp(-2m\varepsilon^2) \tag{2.7}$$

we finally obtain the inequality of Vapnik-Chervonenkis

$$P\big(\sup_{f\in\mathcal{F}}(R[f] - R_{\text{emp}}[f]) > \varepsilon\big) \leq 4\exp\left(\ln \mathbb{E}[\mathcal{N}(\mathcal{F}, Z_{2m})] - \frac{m\varepsilon^2}{8}\right). \tag{2.8}$$

We can rewrite Equation 2.8 in a more explicit way, in particular with a probability at least $1 - \delta$ we have that

$$R[f] \leq R_{\text{emp}}[f] + \sqrt{\frac{8}{m}\left(\ln \mathbb{E}[\mathcal{N}(\mathcal{F}, Z_{2m})] + \ln \frac{4}{\delta}\right)}. \tag{2.9}$$

Note that this bound holds independently on the function f. This is a strength, because learning algorithms do not truly minimize the empirical risk, and this bound grant at least an acceptable performance, but also a weakness, since using more information related on the function we are interested in, we could have better bounds.

The critical factor that controls how much our choice may have compromised the stability of the resulting pattern is the *capacity* of the function class $\mathcal{F}$. The capacity is the capability of a function class to fit different data. Clearly the higher the capacity of the class the greater the risk of overfitting the particular training data and identifying a spurious pattern. On the other hand if we choose a class with poor capacity we may have an underfitting problem related with a possibly high empirical risk. The capacity will be related to tunable parameters of the algorithms for pattern analysis, hence making it possible to directly control the risk of overfitting the data.

In Equation (2.9) the so called capacity-term is $\sqrt{\frac{8}{m}\left(\ln \mathbb{E}[\mathcal{N}(\mathcal{F}, Z_{2m})] + \ln \frac{4}{\delta}\right)}$

A further step can be done, by trying to minimize not only the empirical risk but rather the right hand side of Equation (2.8), since this leads not only to a small risk but also to choosing a function from a class with smaller capacity. This intuition leads to the so-called *structural risk minimization*, where the main idea is to define a structure on $\mathcal{F}$ and minimize over the choice of the structure.

As capacity measure, we have used so far the so-called *annealed entropy*

$$H_{\mathcal{F}}^{\text{ann}}(m) = \ln \mathbb{E}\left[\mathcal{N}(\mathcal{F}, (x_1, y_1), ..., (x_m, y_m))\right], \tag{2.10}$$

but there are more function that we can consider, like the *VC Entropy*:

$$H_{\mathcal{F}}(m) = \mathbb{E}\left[\ln \mathcal{N}(\mathcal{F}, (x_1, y_1), ..., (x_m, y_m))\right], \tag{2.11}$$

or the *Growth function*:

$$G_{\mathcal{F}}(m) = \max_{(x_1,y_1),...,(x_m,y_m)\in\mathcal{F}\times\{\pm 1\}} \ln \mathcal{N}\left(\mathcal{F}, (x_1, y_1), ..., (x_m, y_m)\right) \tag{2.12}$$

and lastly the *VC dimension*.

Definition 2.5 (VC dimension). Let be $\mathcal{F}$ a class of classifying functions. the VC dimension of $\mathcal{F}$ is the maximum number of points that can be shattered by functions in $\mathcal{F}$. If such a number does not exist, then the VC dimension is infinity.

For example, take three not collinear points x_1, x_2, x_3 in $\mathbb{R}^2$. Independently on their labels, we can always find an hyperplane which divides the points in the two classes. This implies that the VC dimension of the set of hyperplanes on $\mathbb{R}^2$ is $h \geq 3$. Moreover if we consider four points this ceases to be true, thus

we can never shatter four points in $\mathbb{R}^2$ with a set of hyperplanes. Therefore the VC dimension in this case is $h = 3$. It can be shown that the VC dimension in the case of hyperplanes on $\mathbb{R}^d$ is $d + 1$.

It is denoted by h and it is possible to prove that, for $m > h$ it holds:

$$G_{\mathcal{F}}(m) \leq h\left(\ln \frac{m}{h} + 1\right)$$

furthermore we can consider the other capacity measures and have

$$H_{\mathcal{F}}(m) \leq H_{\mathcal{F}}^{\text{ann}}(m) \leq G_{\mathcal{F}}(m) \leq h\left(\ln \frac{m}{h} + 1\right) \tag{2.13}$$

from which we can rewrite Equation 2.9 accordingly of our needs.

Another relevant capacity measure is the so called Rademacher complexity, which is based on the capability of a class to fit random data.

Definition 2.6 (Rademacher complexity). For a sample $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ generated by a distribution $p(\mathbf{x})$ on a set Ω and a real-valued function class $\mathcal{F}$ with domain Ω, the empirical Rademacher complexity of $\mathcal{F}$ is the random variable

$$\hat{R}_n(\mathcal{F}) = \mathbb{E}_\sigma\left[\sup_{f\in\mathcal{F}}\left|\frac{2}{n}\sum_{i=1}^{n}\sigma_i f(\mathbf{x}_i)\right| \,\middle|\, \mathbf{x}_1, ..., \mathbf{x}_n\right], \tag{2.14}$$

where $\sigma = \{\sigma_1, ..., \sigma_n\}$ are independent uniform $\{\pm 1\}$-valued (Rademacher) random variables. The Rademacher complexity of $\mathcal{F}$ is

$$R_n(\mathcal{F}) = \mathbb{E}_{\mathcal{X}}\left[\hat{R}_n(\mathcal{F})\right] = \mathbb{E}_{\mathcal{X}\sigma}\left[\sup_{f\in\mathcal{F}}\left|\frac{2}{n}\sum_{i=1}^{n}\sigma_i f(\mathbf{x}_i)\right|\right]. \tag{2.15}$$

Here the labeled set Z_n are not taken into account, instead it is assumed that the data are described by a probability distribution and the labels are randomly chosen in $\mathcal{Y} = \{\pm 1\}$ with equal probability (the so called Rademacher variables). The Rademacher complexity uses precisely the ability of the class to fit noise as its measure of capacity, since pattern detection is a probabilistic process, there is always a chance of detecting a pattern in noise.

Now we show some useful propertiesies of empirical Rademacher complexity and thus Rademacher complexity.

Proposition 2.7. *Let $\mathcal{G}$, $\mathcal{F}$, $\mathcal{F}_1$, ..., $\mathcal{F}_m$ class of real functions. Then:*

(i) *If $\mathcal{F} \subset \mathcal{G}$, then $\hat{R}_n(\mathcal{F}) \leq \hat{R}_n(\mathcal{G})$.*

(ii) *$\hat{R}_n(\mathcal{F}) = \hat{R}_n(conv\mathcal{F})$.*

(iii) *For every $c \in \mathbb{R}$, $\hat{R}_n(c\mathcal{F}) = |c|\hat{R}_n(\mathcal{F})$.*

(iv) *$\hat{R}_n(\sum_{i=1}^m \mathcal{F}_i) \leq \sum_{i=1}^m \hat{R}_n(\mathcal{F}_i)$.*

We point out that property (i) assures the stability of Rademacher complexity as capacity measure.

Like with VC dimension, is possible to bound the expected risk using a Rademacher complexity dependent capacity term:

Theorem 2.8. *Let $\mathcal{F}$ be a class of classifying functions and let Z_n be a set of n labeled data drawn independently according with a probability distribution p(x,y). Then with probability at least $1-\sigma$, for all $f \in \mathcal{F}$ and $\sigma > 0$:*

$$R[f] \leq R_{emp}[f] + R_n(\mathcal{F}) + \sqrt{\frac{\ln(2/\sigma)}{2n}} \tag{2.16}$$

$$\leq R_{emp}[f] + \hat{R}_n(\mathcal{F}) + 3\sqrt{\frac{\ln(2/\sigma)}{2n}}. \tag{2.17}$$

Moreover, in [25], is shown that using the Rademacher penalty $R_n(\mathcal{F})$ or its conditional expectation $\mathbb{E}[R_n(\mathcal{F})|Z_n]$ directly instead of VC dimension bounds is advantageous because the Rademacher bound is able to take both the exact size and structure of $\mathcal{F}|Z_n$ into account. Using a VC bound means essentially neglecting the information inherent in Z_n and also the properties of $\mathcal{F}$ not captured by VC dimension. $R_n(\mathcal{F})$ automatically captures both these properties neglected by the VC dimension bound.

2.2 Kernels in Machine Learning

In machine learning and support vector machines we are interested in separating/classifying the given data in the input space. Ideally, we want to do this with a hyperplane. However, using a linear separation severely limits the effectiveness and applicability of such an approach. Therefore, one looks to use nonlinear separation in the input space, and the setting of reproducing kernel Hilbert spaces provides (Section 1.1) a perfect framework to accomplish this while still applying linear techniques.

Recalling the problem at the beginning of this chapter, we have a set of labeled data $(\mathbf{x}_i, y_i)_{i=1,\dots,n}$ with $\mathbf{x}_i \in \mathcal{X} \subset \Omega$, and $y_i \in \mathcal{Y} = \{-1, +1\}$. Here we want to generalize this pattern, in other words if we consider a data $\mathbf{x} \in \Omega \setminus \mathcal{X}$, we want to choose y such that $(\mathbf{x}, y)$ is similar in some sense to the starting labeled dataset. To achieve this, we consider a symmetric similarity measure also called kernel:

$$\kappa : \Omega \times \Omega \to \mathbb{R}, \qquad (\mathbf{x}, \mathbf{x}') \mapsto \kappa(\mathbf{x}, \mathbf{x}')$$

Here we have the same definition of Chapter 1 and the same theoretical background, but we gave it a different interpretation. Since $\Omega \subset \mathbb{R}^d$ a possibility is to consider the Euclidean inner product to compute similarity measures, i.e. $\kappa(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T\mathbf{y}$, but this structure may not always be up to the problem.

In order to overcome this limitation we introduce the concept of *feature map*.

Definition 2.9 (Feature map). Assume that κ is a real valued positive definite kernel, and Ω a non-empty set. if $\mathbb{R}^\Omega := \{f : \Omega \to \mathbb{R}\}$, a feature map is a function such that

$$\begin{aligned} \Phi : \Omega &\to \mathbb{R}^\Omega. \\ \mathbf{x} &\mapsto \kappa(\mathbf{x}, \cdot) \end{aligned}$$

Φ maps patterns into function of $\mathbb{R}^{\Omega}$. This allow us to embed the data into a vector space called *feature space*

$$\mathscr{F} = \left\{ \sum_{i=1}^{n} \alpha_i \kappa(\mathbf{x}_i, \cdot) \,\middle|\, n \in \mathbb{N},\, \mathbf{x}_i \in \Omega,\, \alpha_i \in \mathbb{R},\, i = 1, ..., n \right\}.$$

Using these concept we can build a pre-Hilbert space (or dot product space). Let be $f,\, g \in \mathscr{F}$ with coefficients respectively α_i and $\beta_j \in \mathbb{R}$, and associated with patterns $\mathbf{x}_i$ and $\mathbf{x}'_j \in \Omega$ $i = 1, ..., n$, $j = 1, ..., n'$.

$$f = \sum_{i=1}^{n} \alpha_i \kappa(\mathbf{x}_i, \cdot) \qquad g = \sum_{j=1}^{n'} \beta_j \kappa(\mathbf{x}'_j, \cdot)$$

Thus, we define the inner product as

$$\langle f, g \rangle := \sum_{j=1}^{n'} \sum_{i=1}^{n} \alpha_i \beta_j \kappa(\mathbf{x}_i, \mathbf{x}'_j). \tag{2.18}$$

Note that using the properties of kernels we have that

$$\langle f, g \rangle = \sum_{j=1}^{n'} \beta_j f(\mathbf{x}'_j) = \sum_{i=1}^{n} \alpha_i g(\mathbf{x}_i). \tag{2.19}$$

Theorem 2.10. *A function κ defined on $\Omega \times \Omega$ is a reproducing kernel if and only if there exist a Hilbert space $\mathcal{H}$ and a mapping $\Phi : \Omega \to \mathcal{H}$, such that for all $\mathbf{x},\, \mathbf{x}' \in \Omega$*

$$\kappa(\mathbf{x}, \mathbf{x}') = \langle \Phi_{\mathbf{x}}, \Phi_{\mathbf{x}'} \rangle_{\mathcal{H}}.$$

Specifically, in terms of the native space inner product, we had that

$$\kappa(x, x') = \langle \kappa(\cdot, x), \kappa(\cdot, x') \rangle_{\mathcal{H}_\kappa(\Omega)}.$$

Additionally $\langle f, g \rangle = \langle g, f \rangle$ and

$$\langle f, f \rangle = \sum_{i,j=1}^{n} \alpha_i \alpha_j \kappa(\mathbf{x}_i, \mathbf{x}_j) \geq 0,$$

$$\sum_{i,j=1}^{n} c_i c_j \langle f_i, f_j \rangle = \left\langle \sum_{i=1}^{n} c_i f_i, \sum_{j=1}^{n} c_j f_j \right\rangle \geq 0.$$

This implies that $\langle \cdot, \cdot \rangle$ is actually itself a positive definite kernel on the feature space.

Lastly, $\langle f, f \rangle = 0$ directly implies $f = 0$ and

$$|f(x)|^2 = |\langle \kappa(\mathbf{x}, \cdot), f \rangle|^2 \leq \kappa(\mathbf{x}, \mathbf{x}) \cdot \langle f, f \rangle.$$

Thus, $\langle \cdot, \cdot \rangle$ is a well defined dot product.

Recalling the reproducing property of positive definite kernels we can see that for all functions in $\mathscr{F}$ we have

$$\langle \kappa(\mathbf{x}, \cdot), f \rangle = f(\mathbf{x}),$$

and in particular

$$\langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle = \kappa(\mathbf{x}, \mathbf{x}'). \tag{2.20}$$

So far we have shown that any positive definite kernel can be viewed as a dot product in another space. Therefore, the dot product space constructed in this way is one possible choice of the feature space associated with a a kernel.

Furthermore we can build a kernel starting from a feature map Φ using the equation above and

$$\sum_{i,j=1}^{n} c_i c_j \kappa(\mathbf{x}_i, \mathbf{x}_j) = \left\langle \sum_{i=1}^{n} c_i \Phi(\mathbf{x}_i), \sum_{j=1}^{n} c_j \Phi(\mathbf{x}_j) \right\rangle = \left\| \sum_{i=1}^{n} c_i \Phi(\mathbf{x}_i) \right\|^2 \geq 0$$

The Equation 2.20 states the equivalence between a kernel evaluation and a dot product of feature maps, it is often referred to as the *kernel trick* in the machine learning literature. It is particularly interesting by a computational point of view, since $\kappa(\mathbf{x}, \mathbf{x}')$ gives us the scalar product without mapping Φ, also allow us to work with a more suitable space and associated scalar product. In fact we can build different kernels based on the problems and data we are working with, and the RKHS will always be equipped with the proper inner product.

Remark 2.11 (Kernel Trick). Given an algorithm which is formulated in terms of a positive definite kernel κ, one can construct an alternative algorithm by replacing κ by another positive definite kernel $\tilde{\kappa}$.

Another useful insight that characterizes the continuity of the feature map Φ is the following:

Proposition 2.12 (Continuity of the feature map). *If $\mathcal{X}$ is a topological space and κ is a continuous positive definite kernel on $\mathcal{X} \times \mathcal{X}$, then there exists a Hilbert space $\mathcal{H}$ and a continuous map $\Phi : \mathcal{X} \to \mathcal{H}$ such that for all $\mathbf{x}$, $\mathbf{x}' \in \mathcal{X}$, we have $\kappa(x\, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle$.*

On the other hand, recalling Theorem 1.18 we can also interpret the Mercer series representation of κ in terms of an inner product, now in the sequence space ℓ_2,

$$\kappa(\mathbf{x}, \mathbf{x}') = \sum_{n=1}^{\infty} \lambda_n \varphi_n(\mathbf{x}) \varphi_n(\mathbf{x}') = \langle \Phi_{\mathbf{x}}, \Phi_{\mathbf{x}'} \rangle_{\ell_2} \tag{2.21}$$

where λ_n are the eigenvalues of the eigenfunctions φ_n, and

$$\begin{aligned} \Phi : \Omega &\to \ell_2^n \\ \mathbf{x} &\mapsto \left(\sqrt{\lambda_1} \varphi_1(\mathbf{x}), \sqrt{\lambda_2} \varphi_2(\mathbf{x}), ... \right) \end{aligned}$$

is the so called Mercer feature vector. Even if we use Φ for both the construction of the feature maps, although the target spaces are different. However we have shown different methods for constructing Hilbert spaces where the kernel corresponds to a dot product meaningful for us, and as long as we

are interested only in dot products, if we consider two different feature spaces $\mathcal{H}_1$ and $\mathcal{H}_2$ with their maps Φ_1 and Φ_2, we have that

$$\langle \Phi_1(\mathbf{x}), \Phi_1(\mathbf{x}') \rangle_{\mathcal{H}_1} = \langle \Phi_2(\mathbf{x}), \Phi_2(\mathbf{x}') \rangle_{\mathcal{H}_2}.$$

We conclude this section with some properties of positive definite kernels in order to construct more complicated kernels from simpler ones.

Proposition 2.13. *Let κ_1 and κ_2 be kernels over $\Omega \times \Omega$, $\Omega \subset \mathbb{R}^d$, $a \in \mathbb{R}$, $\sigma \in \mathbb{R}^+$, $f(\cdot)$ a real value function on Ω, $\psi : \Omega \to \mathbb{R}^m$ with κ_3 a kernel over $\mathbb{R}^m \times \mathbb{R}^m$, B a symmetric positive definite $n \times n$ matrix, $\mathbf{x}, \mathbf{x}' \in \Omega$ and $p(\mathbf{x})$ is a polynomial with positive coefficients. Then the following functions are kernels:*

(i) $\kappa(\mathbf{x}, \mathbf{x}') = \kappa_1(\mathbf{x}, \mathbf{x}') + \kappa_2(\mathbf{x}, \mathbf{x}')$.

(ii) $\kappa(\mathbf{x}, \mathbf{x}') = a\kappa_1(\mathbf{x}, \mathbf{x}')$.

(iii) $\kappa(\mathbf{x}, \mathbf{x}') = \kappa_1(\mathbf{x}, \mathbf{x}')\, \kappa_2(\mathbf{x}, \mathbf{x}')$.

(iv) $\kappa(\mathbf{x}, \mathbf{x}') = f(\mathbf{x}) f(\mathbf{x}')$.

(v) $\kappa(\mathbf{x}, \mathbf{x}') = \kappa_3(\psi(\mathbf{x}), \psi(\mathbf{x}'))$.

(vi) $\kappa(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T B\, \mathbf{x}'$.

(vii) $\kappa(\mathbf{x}, \mathbf{x}') = p(\kappa_1(\mathbf{x}, \mathbf{x}'))$.

(viii) $\kappa(\mathbf{x}, \mathbf{x}') = \exp(\kappa_1(\mathbf{x}, \mathbf{x}'))$.

(ix) $\kappa(\mathbf{x}, \mathbf{x}') = \exp(-\|\mathbf{x} - \mathbf{x}'\|/(2\sigma^2))$.

2.3 Support Vector Machines

In this section we deal with the problem introduced in the first pages of this chapter with the well-known *Support Vector Machines* (SVMs). For a complete discussion about this topic see [21, 39, 40, 41].

We start considering the so called *hard margin* setting, where linear separability between training data is assumed.

Suppose we are given a pre-Hilbert space $\mathcal{X}$, and a set of pattern vectors $\mathbf{x}_1, ..., \mathbf{x}_n \in \mathcal{X}$. Any hyperplane in $\mathcal{X}$ is of the form

$$\{\mathbf{x} \in \mathcal{X} | \, \langle \mathbf{w}, \mathbf{x} \rangle + b = 0\}, \qquad \mathbf{w} \in \mathcal{X},\, b \in \mathbb{R}. \tag{2.22}$$

Here $\mathbf{w}$ is a vector orthogonal to the hyperplane: If $\mathbf{w}$ has unit length, then $\langle \mathbf{w}, \mathbf{x} \rangle$ is the length of $\mathbf{x}$ along the direction of $\mathbf{w}$. In any case the set 2.22 describes vectors that have the same length along $\mathbf{w}$. Since if we multiply b and $\mathbf{w}$ by the same non-zero constant we have the same set, in order to have the uniqueness of the hyperplane we define:

Definition 2.14 (Canonical hyperplane). The pair $(\mathbf{w}, b) \in \mathcal{X} \times \mathbb{R}$ is called canonical form of the hyperplane 2.23 with respect to $\mathbf{x}_1, ..., \mathbf{x}_n \in \mathcal{X}$, if it is scaled such that

$$\min_{i=1,...,n} |\langle \mathbf{w}, \mathbf{x}_i \rangle + b| = 1, \tag{2.23}$$

which amounts to say that the point closest to the hyperplane has a distance of $\frac{1}{\|\mathbf{w}\|}$.

We can further notice that the two different pairs $(\mathbf{w}, b)$ and $(-\mathbf{w}, -b)$ satisfy the condition 2.23. For the purpose of classification, these hyperplanes are different since they correspond to two decision function

$$\begin{aligned} f_{\mathbf{w},b} :&\mathcal{X} \to \{\pm 1\}. \\ &\mathbf{x} \mapsto f_{\mathbf{w},b}(\mathbf{x}) = \operatorname{sgn}(\langle \mathbf{w}, \mathbf{x} \rangle + b) \end{aligned}$$

In pattern recognition we attempt to find a function $f_{\mathbf{w},b}$ which correctly classifies $f_{\mathbf{w},b}(\mathbf{x}_i) = y_i$, or at least for a large fraction of the patterns $\mathbf{x}_i$.

Now we introduce the concept of margin, which play a key role on the design of SVMs.

Definition 2.15 (Geometrical margin). For a hyperplane $\{\mathbf{x} \in \mathcal{X} | \, \langle \mathbf{w}, \mathbf{x} \rangle + b = 0\}$ we call

$$\rho_{(\mathbf{w},b)}(\mathbf{x}, y) := \frac{y(\langle \mathbf{w}, \mathbf{x} \rangle + b)}{\|\mathbf{w}\|} \tag{2.24}$$

the geometrical margin (usually simply called margin) of the point $(\mathbf{x}, y) \in \mathcal{X} \times \{\pm 1\}$. The minimum value

$$\rho_{(\mathbf{w},b)} := \min_{i=1,\ldots,n} \rho_{(\mathbf{w},b)}(\mathbf{x}_i, y_i),$$

shall be called the geometrical margin of $(\mathbf{x}_1, y_1)$, ..., $(\mathbf{x}_n, y_n)$. If the latter is omitted, it is understood that the training set is meant.

If $(\mathbf{x}, y)$ is a well classified point the margin is the distance from $\mathbf{x}$ to the hyperplane. Indeed, notice that that the margin is zero on the hyperplane. The multiplication by y ensures that that the margin is positive for all points that are correctly classified, and negative for the misclassified ones. Moreover notice that for canonical hyperplanes the margin is $1/\|\mathbf{w}\|$.

It turns out that the margin of a separating hyperplane, and thus the length of the weight vector $\mathbf{w}$, plays a fundamental role in SVMs. In fact if we manage to separate the training data with a large margin, we can assumed that the classification of the test set goes well.

The simplest justification is the following: since we assumed that training and test data are generated by the same underlying dependence, it seems reasonable to assume that most of the test patterns are close to at least one of the training patterns. For simplicity, we suppose that all test points are generated with random noise added to training patterns, i.e. given training patterns $(\mathbf{x}_i, y_i)$, the test patterns are of the form $(\mathbf{x} + \Delta\mathbf{x}, y)$, with $\Delta\mathbf{x}$ bounded by some $r > 0$. Clearly if we separate the points with a margin $\rho > r$, we correctly classify all test points. As r grows up to ρ, the resulting hyperplane should better approximate the maximum margin solution.

A similar robustness approach can be made for the dependence of the hyperplane parameters $(\mathbf{w}, b)$. It can be shown that small perturbations to the hyperplane parameters will not change the classification of the training data [42].

For a more formal proof we consider hyperplanes that have offset $b = 0$, leaving $f(\mathbf{x}) = \operatorname{sgn}\langle \mathbf{w}, \mathbf{x} \rangle$

Theorem 2.16 (Margin error bound). *Consider the set of decision function $f(\mathbf{x}) = sgn\langle \mathbf{w}, \mathbf{x} \rangle$ with $\|\mathbf{w}\| \leq \Lambda$ and $\|\mathbf{x}\| \leq R$, for some $\Lambda, R > 0$. Moreover, let $\rho > 0$, and ν denote the fraction of training examples with margin smaller that $\frac{\rho}{\|\mathbf{w}\|}$, referred to as the margin error.*

For all distributions P generating the data, with probability at least $1 - \delta$ over the drawing of the n training patterns, and for any $\rho > 0$ and $\delta \in (0, 1)$, the probability that a test pattern drawn from P will be misclassified is bounded from above, by

$$\nu + \sqrt{\frac{c}{n}\left(\frac{R^2\Lambda^2}{\rho^2}\ln^2 n + \ln\frac{1}{\sigma}\right)}. \tag{2.25}$$

Here c is a universal constant

The test error is bound by a margin error ν and a capacity term that reminds us the VC bounds shown in Section 2.1. The latter of the capacity term tends to zero as the number of examples tends to infinity, assuming that R and Λ are bounded, the main influence is given by ρ. As we can see from Theorem 2.16, a large ρ leads to a small capacity term, but leads also to a larger margin error ν. On the other hand, a small ρ leads to a smaller margin error but capacity terms will increase correspondingly.

Notice that maximizing ρ is the same as minimizing the length of $\mathbf{w}$. Thus we might fix ρ (usually fixed $\rho = 1$) and search for hyperplanes which has small $\|\mathbf{w}\|$ and few points with a margin smaller than $\frac{\rho}{\|\mathbf{w}\|}$.

The favored approach in literature is the following: keep the margin training error small, and the margin large, in order to achieve high generalization ability. In other words, hyperplane decision functions should be constructed such that they maximize the margin, and at the same time separate the training data with few exceptions as possible.

Our aim is now to find the optimal margin hyperplane. Suppose we have a training set $(\mathbf{x}_i, y_i)$ with $i = 1, ..., n$. We want to find a decision function $f_{\mathbf{w},b}(\mathbf{x}) = \operatorname{sgn}(\langle \mathbf{w}, \mathbf{x} \rangle + b)$ satisfying

$$f_{\mathbf{w},b}(\mathbf{x}_i) = y_i,$$

if such function exists, the canonicality (2.23), implies that

$$y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1.$$

Thus following the previous results, a separating hyperplane which generalizes well can be constructed by solving

$$\begin{array}{ll} \underset{\mathbf{w} \in \mathcal{X},\, b \in \mathbb{R}}{\text{minimize}} & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{subject to} & y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 \quad \text{for all } i = 1, ..., n \end{array} \tag{2.26}$$

This is called the *primal optimization problem.*

Methods or techniques to solve this kind of problems are beyond the scope of this thesis. Furthermore we derive the *dual problem*, since it can be shown that it has the same solution as 2.26. As we will see, in this case is more convenient to deal with the dual.

To derive it we introduce the Lagrangian,

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i (y_i(\langle \mathbf{x}_i, \mathbf{w}\rangle + b) - 1), \tag{2.27}$$

with Lagrange multipliers $\alpha_i \geq 0$. Since the Lagrangian must be maximized with respect to α_i, and minimized with respect to $\mathbf{w}$ and b, thanks to Karush-Kuhn-Tucker conditions [39]. Moreover, at this saddle point, the derivatives of L with respect the primal variables must vanish.

$$\frac{\partial}{\partial b} L(\mathbf{w}, b, \alpha) = 0, \qquad \frac{\partial}{\partial \mathbf{w}} L(\mathbf{w}, b, \alpha) = 0.$$

Thus we have the dual form of the problem

$$\begin{aligned} \underset{\alpha \in \mathbb{R}^n}{\text{maximize}} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \\ \text{subject to} \quad & \alpha_i \geq 0 \quad \text{for all } i = 1, ..., n, \\ & \sum_{i=1}^{n} \alpha_i y_i = 0 \end{aligned} \tag{2.28}$$

and that

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i,$$

where we can see an expansion of the solution vector in term of the training examples. Furthermore the solution $\mathbf{w}$ is unique.

It can be proven that only the Lagrangian multipliers α_i that are non zero at the saddle point, correspond to constraints 2.26 which are precisely met. The patterns x_i for which $\alpha_i > 0$ are called *support vector*. Its easy to see that they lay exactly on the margin. All remaining patterns in training set are irrelevant, for which $\alpha_i = 0$, since their constraints are satisfied automatically and they not appear in the expansion of the solution vector.

This lead directly to an upper bound on the generalization ability of optimal margin hyperplanes.

Therefore the optimal classifier is

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{n} \alpha_i y_i \langle \mathbf{x}, \mathbf{x}_i \rangle + b \right). \tag{2.29}$$

So far we have followed a linear approach to data, to have a more general approach now we introduce kernels to nonlinearly transform the input data into a high dimensional feature space, using the feature map $\Phi : \mathbf{x}_i \mapsto \Phi(\mathbf{x}_i)$. Then we do a linear separation there.

Thanks to Cover's Theorem [13] we have a characterization of the number of possible linear separations of n points in general position in a d-dimensional

space. If $n \leq d+1$, then all 2^n separations are possible, if $n > d+1$ then the Cover's Theorem states that the number of linear separation is

$$2\sum_{i=0}^{d}\binom{n-1}{i}.$$

This Theorem formalize the idea that the number of possible separations increase as the dimension increase.

On the practical level, no particular changes is needed, since in the framework defined so far we do not make restrictive assumptions. $\mathcal{X}$ needs only to be equipped with a dot product and the patterns $\mathbf{x}_i$ do not have to coincide with the input patterns, they can be the results of mapping the original patterns into a high dimensional space. We simply need to take into account that whenever we consider $\mathbf{x}$ before we meant $\Phi(\mathbf{x})$. Furthermore the computation of $\langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$ is needed to maximize the target function 2.28 and evaluate the optimal classifier 2.29. These expensive calculations are reduced significantly by using a positive definite kernel and use the kernel trick 2.11.

Then the form of the decision function is

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{n} \alpha_i y_i \kappa(\mathbf{x}, \mathbf{x}_i) + b\right). \tag{2.30}$$

To find it, we solve the following problem

$$\begin{aligned} \underset{\alpha}{\text{maximize}} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \kappa(\mathbf{x}_i, \mathbf{x}_j) \\ \text{subject to} \quad & \alpha_i \geq 0 \quad \text{for all } i = 1, ..., n, \\ & \sum_{i=1}^{n} \alpha_i y_i = 0 \end{aligned} \tag{2.31}$$

If κ is positive definite, $(y_i y_j \kappa(\mathbf{x}_i, \mathbf{x}_j))_{ij}$ is a positive definite matrix, which provides us with a convex problem that can be solved efficiently. In fact

$$\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \kappa(\mathbf{x}_i, \mathbf{x}_j) = \left\langle \sum_{i=1}^{n} \alpha_i y_i \Phi(\mathbf{x}_i), \sum_{i=j}^{n} \alpha_j y_j \Phi(\mathbf{x}_j) \right\rangle \geq 0.$$

To compute the threshold b, we take into account that due to the Karush-Kuhn-Tucker conditions, $\alpha > 0$ implies

$$\sum_{i=1}^{n} \alpha_i y_i \kappa(\mathbf{x}_i, \mathbf{x}_j) + b = y_i.$$

Thus the threshold can be obtained, for instance, by averaging

$$b = y_i - \sum_{i=1}^{n} \alpha_i y_i \kappa(x_i, x_j). \tag{2.32}$$

Sometimes is also useful to do not use the optimal threshold b, but change it in order to balance the number of false positives and false negatives.

Among all the SVM optimization problems, the one in 2.31 is actually a general framework that includes all the others. Indeed, the linear case can be recovered by simply considering the linear kernel $\kappa(\mathbf{x}, y) = \mathbf{x} \cdot y$.

So far we have discussed freely about the separating hyperplane. But there is no guarantee that such hyperplane exists, i.e. that the data are linearly separable, and even if it does, it is not always the best solution to the classification problem. After all in practical applications we can have mislabelled patterns that can sensitively can affect the hyperplane. We would rather have algorithms which can take into account the presence of some outliers. This setting is the so called *soft margin* formulation. If in the following we take the stance that wherever we will write $\mathbf{x}$, we actually meant $\Phi(\mathbf{x})$, we have the resulting non linear soft margin formulation.

A natural idea might be to have an algorithm that return the hyperplane which lead to the minimal number of training errors. Unfortunately this approach is hard to approximate: it can be shown that finding such hyperplane is it NP-hard. Moreover it can be also shown that by disregarding points that are within some fixed positive margin of the hyperplane, then the problem has polynomial complexity.

We can let some elements violating the constraint 2.26, using the so called *slack variables*,

$$\xi_i \geq 0, \qquad \text{with } i = 1, ..., n$$

and use them to relax the constraint

$$y_i(\kappa(\mathbf{x}_i, \mathbf{w}) + b) \geq 1 - \xi_i, \qquad \text{with } i = 1, ..., n \tag{2.33}$$

Clearly increasing ξ_i the constraint 2.33 is always satisfied. In order to not obtain the trivial solution with all large values of ξ_i, we have to penalize them in the objective function. Thus, we add the term $\sum_i \xi_i$ on the objective function.

Then, for some $C > 0$, we introduce the primal form of our soft margin problem, approached with the so called $C-$SV classifier

$$\begin{array}{ll} \underset{\mathbf{w} \in \mathcal{X},\, \xi \in \mathbb{R}^n}{\text{minimize}} & \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{n}\sum_{i=1}^{n} \xi_i \\ \text{subject to} & y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i \quad \text{for all } i = 1, ..., n, \\ & \xi_i \geq 0 \qquad\qquad\quad \text{for all } i = 1, ..., n \end{array} \tag{2.34}$$

It is interesting to compare this with Theorem 2.16, considering the case $\rho = 1$. Whenever the constraint 2.33 is met with $\xi_i = 0$, the corresponding point will not be a margin error. On the other hand all $\xi_i > 0$ corresponds to margin errors. Hence, the the fractions of margin error of the Theorem increase along with $\sum_i \xi_i$, while the capacity term increase with $\|\mathbf{w}\|$. Therefore, for a suitable constant C, this approach minimizes the right hand side of the bound.

However if the term $\sum_i \xi_i$ is too large, in particular larger than the fraction of margin errors, there is no guarantee that the hyperplane generalizes well.

As in separable case the solution can be shown to have an expansion

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i.$$

Again we can find the dual formulation and and find the coefficient α_i

$$\begin{aligned} \underset{\alpha \in \mathbb{R}^n}{\text{maximize}} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \kappa(\mathbf{x}_i, \mathbf{x}_j) \\ \text{subject to} \quad & 0 \leq \alpha_i \leq \frac{C}{n} \quad \text{for all } i = 1, ..., n, \\ & \sum_{i=1}^{n} \alpha_i y_i = 0 \end{aligned} \tag{2.35}$$

to compute the threshold b, we take into account that for 2.33, for support vectors $\mathbf{x}_i$ associated with $\xi_i = 0$ we have an analogous situation to the separability case and thus find the optimal threshold averaging 2.32 over all support vectors with $\alpha_i < C$.

In the above formulation C is the trade-off between the minimization of the error and the maximization of the margin. Unluckily the parameter C is not intuitive, and we do not have a priori ways to select it.

Therefore, a sightly different method is proposed in [40] which replace C by a parameter ν.

As a primal form of this problem, referred as $\nu-$SV classifier, we have

$$\begin{aligned} \underset{\mathbf{w} \in \mathcal{X},\, \xi \in \mathbb{R}^n \rho, b \in \mathbb{R}}{\text{minimize}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 - \nu\rho + \frac{1}{n}\sum_{i=1}^{n} \xi_i \\ \text{subject to} \quad & y_i(\langle \mathbf{w}, \mathbf{x} \rangle + b) \geq \rho - \xi_i \quad \text{for all } i = 1, ..., n, \\ & \xi_i \geq 0 \qquad\quad \text{for all } i = 1, ..., n, \\ & \rho \geq 0 \end{aligned} \tag{2.36}$$

Instead of C we have the parameter ν and a new variable ρ to optimize. To understand the meaning of ρ, note that for $\xi_i = 0$ the first constraint of 2.36 states that the separation of the two classes is done by margin $\frac{2\rho}{\|\mathbf{w}\|}$. On the other hand for underline the role of ν we state some of its property

Proposition 2.17. *Suppose we run the $\nu-$SV classifier with κ on some data with the result that $\rho > 0$. Then*

(i) *ν is an upper bound on the fraction of margin errors.*

(ii) *ν is a lower bound on the fraction of support vectors.*

(iii) *suppose the data $(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)$ were generated iid from a distribution $P(\mathbf{x}, y) = P(\mathbf{x})P(y|\mathbf{x})$, such that neither $P(\mathbf{x}, y = 1)$ nor $P(\mathbf{x}, y = -1)$ contains any discrete component. Suppose, moreover, that the kernel used is analytic and non constant. With probability 1, asymptotically, ν equals both the fraction of support vectors and the fraction of errors.*

The derivation of the dual is similar to the other cases. Thus we consider the Lagrangian

$$L(\mathbf{w},\xi,b,\rho,\alpha,\beta,\delta) = \frac{1}{2}\|\mathbf{w}\|^2 - \nu\rho + \frac{1}{m}\sum_{i=1}^{n}\xi_i \\ - \sum_{i=1}^{n}(\alpha_i(y_i(\kappa(\mathbf{x}_i,\mathbf{w})+b)-\rho+\xi_i)+\beta_i\xi_i) - \delta\rho,$$

thus we obtain the following conditions

$$\mathbf{w} = \sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i, \tag{2.37}$$

$$\alpha_i + \beta_i = \frac{1}{m}, \tag{2.38}$$

$$\sum_{i=1}^{n}\alpha_i y_i = 0, \tag{2.39}$$

$$\sum_{i=1}^{n}\alpha_i - \delta = \nu. \tag{2.40}$$

From which we find the dual formulation

$$\begin{aligned} \underset{\alpha\in\mathbb{R}^n}{\text{maximize}} \quad & -\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j \kappa(\mathbf{x}_i,\mathbf{x}_j) \\ \text{subject to} \quad & 0 \le \alpha_i \le \frac{1}{n} \quad \text{for all } i = 1,...,n, \\ & \sum_{i=1}^{n}\alpha_i y_i = 0, \\ & \sum_{i=1}^{n}\alpha_i \ge \nu \quad \text{for all } i = 1,...,n \end{aligned} \tag{2.41}$$

As above, the resulting decision function can be shown to take the form

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{n}\alpha_i y_i \kappa(\mathbf{x},\mathbf{x}_i) + b\right). \tag{2.42}$$

Furthermore it can be shown that optimal values for b and ρ are respectively

$$b = -\frac{1}{2s}\sum_{\mathbf{x}\in S_+\cup S_-}\sum_{j=1}^{n}\alpha_j y_j \kappa(\mathbf{x},\mathbf{x}_j), \tag{2.43}$$

$$\rho = \frac{1}{2s}\left(\sum_{\mathbf{x}\in S_+}\sum_{j=1}^{n}\alpha_j y_j \kappa(\mathbf{x},\mathbf{x}_j) - \sum_{\mathbf{x}\in S_-}\sum_{j=1}^{n}\alpha_j y_j \kappa(\mathbf{x},\mathbf{x}_j)\right). \tag{2.44}$$

Here $S_\pm$ are sets of identical size $s > 0$, containing support vectors with $0 < \alpha_i < 1$ and $y_i = \pm 1$ respectively.

Proposition 2.18 (Connection between $\nu-$SV and $C-$SV classifiers). *If $\nu-SV$ classification leads to $\rho > 0$, then $C-SV$ classification, with C set a priori to $\frac{1}{\rho}$, leads to the same decision function.*

Lastly we consider robustness of soft margin classification. When we introduced the slack variables, we did not justify the choice of the penalizer $\sum_i \xi_i$, so now we do it thanks to the following proposition introduced and proven in [40].

Proposition 2.19 (Resistance of support vector classification). *Suppose* $\mathbf{w}$ *can be expressed in terms of of the support vectors which are not bound*

$$\mathbf{w} = \sum_{i=1}^{n} \gamma_i(\mathbf{x}_i), \tag{2.45}$$

with $\gamma_i \neq 0$ only if the coefficient of dual solution $\alpha_i \in (0, \frac{1}{n})$. Then local movements of any margin errors x_n parallel to $\mathbf{w}$ *do not change the hyperplane.*

Note that the assumption 2.45 is not restrictive as it seems. Even though the support vector expansion of the solution, $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$, often contains many multipliers α_i which are bound, it still possible that we can obtain an expansion, as the one in the proposition, in terms of a subset of support vectors.

CHAPTER 3

Topological Persistence and its application

3.1 Introduction to Simplicial Complex and Simplicial Homology Group

Before getting into the details of the topological persistence and its properties we need to define some topological basic concepts that underlie it. For this introductory section we refer to [19, Chapter 1] while for the section dealing with *persistent homology* we refer to [9, 20].

Definition 3.1 (n-dimensional simplex). Consider $n+1$ linearly independent points $A_0, ...A_n$ in the Euclidean space $\mathbb{R}^{n+1}$, so there are $n+1$ distinct vectors going from the origin into these points. A *n-dimensional simplex* Δ^n is a convex linear hull of the points $A_0, ..., A_n$. The points $A_0, ..., A_n$ are called *vertices* of the simplex.

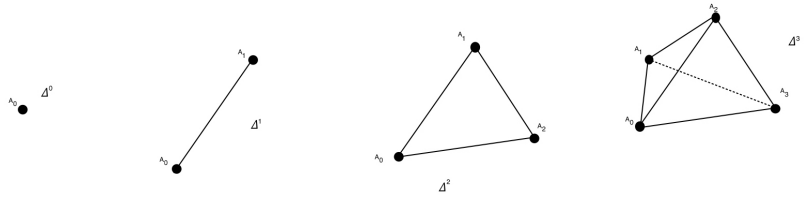

Figure 3.1: n-dimensional simplex from left to right with $n = 0, 1, 2, 3$

Associating every point A_i with the non-negative mass m_i, $i = 0, ..., n$ and requiring that $m_0 + ... + m_n = 1$, let us the opportunity to define the *center of gravity* of the simplex: $A = m_0 OA_0 + ... + m_n OA_n$. The numbers m_i, $i = 0, ..., n$, are called *barycentric coordinates* of the point A.

Now we consider a set of points of a n-dimensional simplex, such that $\exists i \in \{1, ..., n\}$, $m_i = 0$. From the definition of simplex we know that this set is a (n-1)-dimensional simplex, that is embedded in the original simplex Δ^n and is opposite to the vertex A_i. This simplex (which we denote with Δ_i^{n-1}) is

called the *i-th (n-1)-dimensional face* of the simplex Δ^n. Thus, a n-dimensional simplex has n+1 faces (of dimension n-1).

Generalizing the idea, given a n-dimensional simplex Δ^n, a set of points of the simplex Δ^n, for which some n-k barycentric coordinates are equal to zero, is called *k-dimensional face* of the simplex. The remaining k+1 coordinates change so that the corresponding masses are non-negative and their sum is the unity.

The 1-dimensional faces are called *edges*.

The geometric *boundary* of a n-dimensional simplex Δ^n is the union of its $n-1$ faces.

We say that a simplex is *oriented* if a finite order of vertices is given. We assume that two orders of vertices of a simplex differing by an even permutation determine the same orientation, otherwise, if the order of the vertices differ by an odd permutation, the two orders determine opposite orientation. Usually given a simplex with an orientation we denote it as $+\Delta^n$, while the same simplex with the opposite orientation is denoted as $-\Delta^n$.

In what follows we consider oriented simplex, unless otherwise indicated.

Considering a rectilinear (Euclidean) simplex, as the ones in figure 3.1, a *curvilinear or topological simplex* is an image of the simplex under an homeomorphism.

We say that curvilinear simplexes form a finite *simplicial subdivision* of the set $\mathcal{X}$ of points of a topological space if the two following conditions are met.

(i) There is a finite number of simplexes and each point of the set $\mathcal{X}$ gets into a certain simplex.

(ii) Either two simplexes do not intersect at all, i.e. do not have common points, or one of them is a face of the other, or they have a common face which is the intersection of these simplexes.

Definition 3.2 (Polyhedron). If a set of points of a topological space is divided into simplexes such that the conditions (i) and (ii) hold, this set is called topological polyhedron. The set of points in a Euclidean space, which is the sum of a finite or countable number of rectilinear simplexes satisfying the conditions, thus forms a topological polyhedron.

There exists various ways to subdivide one and the same set of points into the union of simplexes satisfying the conditions (i) and (ii).

A *simplicial complex* is a set of simplexes of a fixed subdivision of a polyhedron.

The language of polyhedra and simplicial homology groups is rather obvious and convenient for the first acquaintance with the important geometric concepts as one can find in [19, Section 1.2].

Let $\mathcal{X}$ be a polyhedron with fixed arbitrary simplicial division. For simplicity we denote with the same letter $\mathcal{X}$ the corresponding simplicial complex. We shall further observe the set of all k-dimensional simplexes of the polyhedron $\mathcal{X}$, numbering them and associate them with some orientation, both in an arbitrary ways, denoting with Δ_i^k the i-th simplex of dimension k. By the definition of polyhedron, the index i runs from one to infinity. Finally, we assume to fix the numbers and the orientations of simplexes.

Definition 3.3 (Simplicial chain). Let G be an abelian group, $\mathcal{X}$ a simplicial complex (polyhedron). The linear combinations of the form:

$$c = \sum_i g_i \Delta_i^k \tag{3.1}$$

are called *k-simplicial chains* or simply *k-chains*,
where g_i belong to the abelian group G with only a finite number of non-zero elements and Δ_i^k are k-dimensional simplexes of $\mathcal{X}$.

Chains can be summed up as ordinary linear forms, that is $c_1 = \sum_i g_i \Delta_i^k$ and $c_2 = \sum_i h_i \Delta_i^k$, the chain $c_1 + c_2$ is equal to $\sum_i (g_i + h_i)\Delta_i^k$. Consequently the set of all k-dimensional chains forms an abelian group which we denote by $C_k(\mathcal{X}, G)$.

Definition 3.4 (Group of simplicial chains). Let G be an abelian group, $\mathcal{X}$ a polyhedron. The group $C_k(\mathcal{X}, G)$ is called the *group of k-dimensional simplicial chains* of the polyhedron $\mathcal{X}$ (of the simplicial complex $\mathcal{X}$).

In the following with $C_k(\mathcal{X})$ we consider $G = \mathbb{Z}$. The associated simplicial chains are called *integer-valued chains*. The simplest chains are those on the form $1 \cdot \Delta_i^k$ and $(-1) \cdot \Delta_i^k$, similarly $C_k(\mathcal{X})$ is the group of integer-valued chains.

Before talking about *boundary of a chain*, we introduce the concept of *orientation induced* on the i-th face Δ_i^{k-1} of the simplex Δ^k. We recall that the orientation of a simplex is supposed to be fixed and given, and as the simplex is given by the set of its vertices, we can obtain the (k-1)-dimensional faces of the simplex by elimination of successive vertices, i.e. if $\Delta^k = (A_0, A_1, ..., A_k)$ we have that $\Delta_i^{k-1} = (A_0, A_1, ..., A_{i-1}, A_{i+1}, ..., A_k)$. The orientation induced on the i-th face Δ_i^{k-1} by the orientation of the simplex Δ^k is determined by the sign $(-1)^i$.

Definition 3.5 (Boundary of an oriented simplex). The *boundary* $\partial\Delta^k$ of an oriented simplex Δ^k is the sum of all its (k-1)-dimensional faces taken with induced orientation. We shall write the simplex boundary operator in algebraic language. We obtain:

$$\partial\Delta^k = \sum_{i=0}^{k} (-1)^i \Delta_i^{k-1} = \Delta_0^{k-1} - \Delta_1^{k-1} + \Delta_2^{k-1} - ...\Delta_k^{k-1} \tag{3.2}$$

In a simple case where $\mathcal{X} = \{A_0, A_1, A_2\} \subset \mathbb{R}^2$, we have:

$$\begin{aligned} \partial\Delta^2 &= \partial(A_0, A_1, A_2) \\ &= (A_0, A_1) + (A_1, A_2) - (A_0, A_2) \\ &= (A_0, A_1) + (A_1, A_2) + (A_2, A_0) \end{aligned}$$

We can generalize this definition and define the boundary of a simplex chain.

Definition 3.6 (Boundary of a simplicial chain). Let $c = \sum_i g_i \Delta_i^k$ be as in equation 3.1. The boundary of a k-dimensional chain c is a (k-1)-dimensional simplicial chain ∂c given by the following:

$$\partial c = \sum_i g_i \partial\Delta_i^k \tag{3.3}$$

The ∂ operator is called *boundary operator* recalling its early definition in the case of an oriented simplex, we can extend it "by linearity" to arbitrary linear combinations of elementary chains thus obtaining a well defined operator on the abelian group $C_k(\mathcal{X})$.

Now before analysing the properties of the boundary operator we have to recall some definitions:

Definition 3.7. A chain z is called a *cycle* if its boundary is equal to zero, i.e. $\partial z = 0$. A chain b is called a *boundary* if $b = \partial h$, for some h simplicial chain whose dimension is greater by unity.

Here we present some simple but useful properties of the boundary operator [19, Section 1.2]

Proposition 3.8. *Let c be a k-chain, and G an abelian group with $g \in G$.*

(i) *The operator ∂ is linear.*

(ii) *The square of the boundary operator ∂ is identically zero.*

(iii) *If $\partial(gc) = 0$ and the coefficient $g \neq 0$, then $\partial c = 0$.*

(iv) *The set of cycles forms an abelian subgroup in a chain group. This subgroup is denoted as $Z_k(\mathcal{X}, G)$ or $Z_k(\mathcal{X})$ in the integer-valued case.*

(v) *the set of boundaries forms an abelian subgroup in a chain group. This subgroup is denoted as $B_k(\mathcal{X}, G)$ or $B_k(\mathcal{X})$ in the integer-valued case.*

(vi) *Each boundary is a cycle, so $B_k(\mathcal{X}) \subset Z_k(\mathcal{X})$, and clearly $Z_k(\mathcal{X}) \subset C_k(\mathcal{X})$. Contrariwise a cycle should not necessary be a boundary.*

Let z be a k-dimensional cycle, we shall say that it is *homologic* to zero if it is the boundary for a certain (k+1)-dimensional chain h, i.e. $z = \partial h$. Notice that the chain h such that $z = \partial h$ is not unique. In fact $z = \partial(h + l)$ where l is an arbitrary (k+1)-dimensional cycle, i.e. $\partial l = 0$.

Two k-dimensional cycles z_1 and z_2 will be called *homologic* if their difference is homologic to zero. Then the homology is sometimes written as follows: $z_1 \sim z_2$, notice that if a cycle z_1 is homologic to a cycle z_2 they differ by a boundary, i.e. for some h (k+1)-dimensional chain holds that $z_1 \sim z_2 + \partial h$.

Similarly two chains c_1 and c_2 are called homologic if they differ by a boundary, i.e. $c_1 = c_2 + \partial h$ or $c_1 \sim c_2$.

Supposing $\mathcal{X}$ is an arbitrary polyhedron, $Z_k(\mathcal{X})$ and $B_k(\mathcal{X})$ are well defined and are both abelian groups. Then we can define:

Definition 3.9 (Simplicial homology group). Given an arbitrary polyhedron $\mathcal{X}$, the group $H_k(\mathcal{X}, G) := Z_k(\mathcal{X}, G)/B_k(\mathcal{X}, G)$ is called the *k-dimensional simplicial homology group* of the polyhedron $\mathcal{X}$. Furthermore it is an abelian group. likewise $H_k(\mathcal{X}) := Z_k(\mathcal{X})/B_k(\mathcal{X})$ is the *k-dimensional integer-valued simplicial homology group*.

We can notice that "Homologies are indivisible": there may exist non-zero elements $\{z\}$ from the group $H_k(\mathcal{X})$, such that their integer multiple is equal to zero, i.e. $m\{z\} \sim 0$ while $z \nsim 0$ for some $m \in \mathbb{Z}$. This implies that $H_k(\mathcal{X})$ are not generally free abelian groups.

Each group H_k can be represented as the direct sum of its two subgroups A_k and B_k, where A_k is a free abelian group (the direct sum of a certain number of copies of the group $\mathbb{Z}$) and B_k is a finite abelian group.

The group A_k is uniquely characterized by the number β_k of constituent copies of the group $\mathbb{Z}$, or in other words, β_k is the rank of the group A_k, i.e. the minimal number of generator of the group. Clearly β_k is also the rank of the entire group H_k.

Definition 3.10 (Betti number). The number β_k is called a *k-dimensional Betti number* of a polyhedron $\mathcal{X}$.

Remark 3.11. The 0-dimensional Betti number is the number of connected components, while the $i-th$ Betti number is the number of $i-$dimensional holes.

Summarizing what we have defined so far we can consider a n-dimensional polyhedron $\mathcal{X}$, and associate the homology groups $H_i(\mathcal{X})$ with $i = 0, 1, ..., n$. Clearly $H_k = 0$ for $k > n$.

These groups depend on the choice of the simplicial complex of the polyhedron. Supposing we fixed the simplicial polyhedron partition we can build the following sequence:

$$C_n(\mathcal{X}) \xrightarrow{\partial_n} C_{n-1}(\mathcal{X}) \xrightarrow{\partial_{n-1}} .... \xrightarrow{\partial_{k+1}} C_k(\mathcal{X}) \xrightarrow{\partial_k} ... \xrightarrow{\partial_1} C_0(\mathcal{X}) \xrightarrow{\partial_0} 0 \qquad (3.4)$$

This sequence is called *chain complex* (of a simplicial complex).

A polyhedron can be represented by different simplicial complex, i.e. different triangulations can be created. This lack of uniqueness might therefore affect the homology groups, making them rely on the choice of the polyhedron triangulation. But this is not the case. The theorem 3.12 grants us that the simplicial homology groups is determined only by the polyhedron itself and not by its triangulation.

Theorem 3.12. *The simplicial homology groups of the polyhedron do not depend on the way the polyhedron is represented as a simplicial complex.*

3.2 Persistent Homology

When using tools from simplicial homology to study a dataset $\mathcal{X} = \{x_i\}_{i=1}^m \subset \mathbb{R}^n$ we face the problem of not having a simplicial complex structure [20]. Assuming that $\mathcal{X}$ is sampled from a manifold, e.g. $\mathcal{X} \subset \mathcal{M}$, usually we hope to gain homological information of $\mathcal{M}$ using only the dataset $\mathcal{X}$. Attempting to construct a simplicial complex structure from $\mathcal{X}$ can be a difficult problem. A simple, but effective, idea would be to consider the homology of the spaces

$$\mathbb{X}_\varepsilon = \bigcup_{i=1}^{m} B(x_i, \varepsilon)$$

where a ball $B(x_i, \varepsilon)$ of radius ε is centered around each point of $\mathcal{X}$. A first strategy would be to try to find an optimal parameter ε_0 such that the homology of $\mathbb{X}_{\varepsilon_0}$ corresponds to the homology of $\mathcal{M}$, but this approach is highly unstable. Here comes the motivation under the introduction of persistent homology: we

get topological information from all the positive values of ε simultaneously, not from just one single value. We firstly describe how to build a suitable structure on $\mathbb{X}_\varepsilon$, then how persistent homology works in details.

There are two principal ways to construct a simplicial complex from a point cloud $\mathcal{X}$ with given parameter $\varepsilon > 0$ [9]:

The first and more intuitive one is the so-called *Čech complex*.

Definition 3.13 (Nerve). Let $\mathcal{X}$ be a topological space and let $\mathcal{U}$ any covering of $\mathcal{X}$. The *nerve* of, denoted by $\mathrm{N}(\mathcal{U})$, will be the the abstract simplicial complex with vertex set A, and where a family $\{\alpha_0, ..., \alpha_k\}$ spans a k-simplex if and only if $U_{\alpha_0} \cap ... \cap U_{\alpha_k} \neq \emptyset$.

Supposing that our dataset is defined in a metric space, for some $\varepsilon > 0$ we can consider the nerve of the set of balls $\mathbb{X}_\varepsilon$ defined above. The Čech complex is usually denoted with $\check{\mathrm{C}}(\mathcal{X}, \varepsilon)$. An important result for this construction is the *nerve lemma* [9, Theorem 2.3], which provide criteria for the homotopy equivalence between $\mathrm{N}(\mathcal{U})$ and $\mathbb{X}_\varepsilon$. However the computation of Čech complex is particularly expensive, since it needs the storage of simplicial complexes of various dimensions.

That is why in [8] was introduced the so called Vietoris-Rips complex. A simplicial complex which can be recovered solely from the edge information.

Definition 3.14 (Vietoris-Rips complex). Let $\mathcal{X}$ denote a metric space, with metric d. Then the *Vietoris-Rips* complex for $\mathcal{X}$, attached to the parameter ε, denoted by $VR(\mathcal{X}, \varepsilon)$, will be the simplicial complex whose vertex set is $\mathcal{X}$, and where $\{x_0, x_1, ..., x_k\}$ spans a k-simplex if and only if $d(x_i, x_j) \leq \varepsilon$ for all $0 \leq i, j \leq k$.

This construction, as wished, depends only on the vertex set and the the pairwise distance between vertexes. This considerably lightens the computation.

Note that both construction have the same vertex sets, so they can be viewed as subcomplexes of the complete simplex on the initial dataset, $\mathcal{X}$ in our case. We have the following relation between the two constructions:

$$\check{\mathrm{C}}(\mathcal{X}, \varepsilon) \subset VR(\mathcal{X}, 2\varepsilon) \subset \check{\mathrm{C}}(\mathcal{X}, 2\varepsilon)$$

In figure 3.2 we can confront the construction of the two kind of simplicial complexes. In the construction of the Vietoris-Rips complex we mark 1-simplicial complex with black, 2-simplicial complex with blue and 3-simplicial complex with red, we can even mark $n-$simplicial complex with $n > 3$ because the V-R complex only depends on the relations between the edges, instead in the construction of the Čech complex we only marked 1-simplicial complex and 2-simplicial complex, because this construction depends on the embedding on $\mathcal{X}$, and with this representation on $\mathbb{R}^2$ we cannot highlight k-simplicial complex with $k > 2$.

For a given ε_k the Vietoris-Rips complex $VR(\mathcal{X}, \varepsilon_k)$ provides an element of the filtration $K_1 \subset K_2 \subset ... \subset K_r$ with $K_i = VR(\mathcal{X}, \varepsilon_i)$. In conclusion, there is only a finite set of positive values $\{\varepsilon_i\}_{i=1}^r$ that describe homological characteristics of $\mathcal{X}$, each of which generate a Vietoris-Rips complex $\{K_i\}_{i=1}^m$

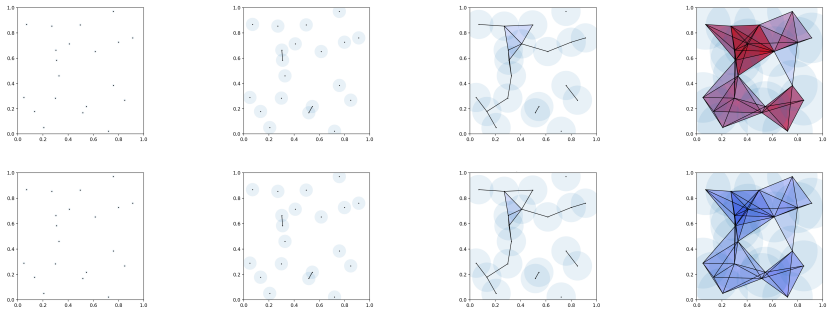

Figure 3.2: Given a random set of point, the construction of the Vietoris-Rips complex (top) and the Čech complex (bottom) for $\varepsilon = 0.02, 0.1, 0.22, 0.4$

representing the topological features of the family $\{\mathbb{X}_\varepsilon, \varepsilon > 0\}$. Therefore, the topological analysis of a point cloud data $\mathcal{X}$ boils down to the analysis of a filtration $K_1 \subset K_2 \subset ... \subset K_r$ which is the main object of study in persistent homology [20, Section 3.2].

Persistent homology was firstly introduced in [14], where the authors wish to simplify a complex through the removal of its topological attributes. They describe a measure that ranks attributes by their life-time in a filtration: their persistence in being a feature in the face of growth.

Denoting a chain complex, as in 3.4, with C_*, we recall that we can define the k-cycle groups and the k-boundary groups respectively as $Z_k = ker\partial_k$, $B_k = im\partial_{k+1}$. A *simple complex* is defined as a family of chain complexes $\{C_*^i\}_{i\geq 0}$ over a commutative ring R, together with the maps:

$$f^i : C_*^i \to C_*^{i+1}$$

or more explicitly:

$$\begin{array}{ccccccc}
\vdots & & \vdots & & \vdots & & \\
\downarrow{\scriptstyle \partial_3} & & \downarrow{\scriptstyle \partial_3} & & \downarrow{\scriptstyle \partial_3} & & \\
C_2^0 & \xrightarrow{f^0} & C_2^1 & \xrightarrow{f^1} & C_2^2 & \xrightarrow{f^2} & \dots \\
\downarrow{\scriptstyle \partial_2} & & \downarrow{\scriptstyle \partial_2} & & \downarrow{\scriptstyle \partial_2} & & \\
C_1^0 & \xrightarrow{f^0} & C_1^1 & \xrightarrow{f^1} & C_1^2 & \xrightarrow{f^2} & \dots \\
\downarrow{\scriptstyle \partial_1} & & \downarrow{\scriptstyle \partial_1} & & \downarrow{\scriptstyle \partial_1} & & \\
C_0^0 & \xrightarrow{f^0} & C_0^1 & \xrightarrow{f^1} & C_0^2 & \xrightarrow{f^2} & \dots \\
\downarrow{\scriptstyle \partial_0} & & \downarrow{\scriptstyle \partial_0} & & \downarrow{\scriptstyle \partial_0} & & \\
0 & \xrightarrow{f^0} & 0 & \xrightarrow{f^1} & 0 & \xrightarrow{f^2} & \dots
\end{array}$$

we will assume that chain complexes are trivial in negative dimension, because of their applications to our problems. Given a filtration of a simplicial complex K, a basic example of a persistent complex is given by considering the functions f^i as the inclusion maps between each simplicial complex in the nested sequence $\emptyset = K_0 \subset K_1 \subset \ldots \subset K_r = K$.

We define Z_k^l, B_k^l respectively to be the k-cycle group and the k-boundary group of the l-th complex K^l in a filtration. To capture persistent cycles in K^l, we factor its k-th cycle group by the k-boundary group of K^{l+p}, p complexes later in the filtration.

Formally the **p-persistence k-homology group of K^l** is:

$$H_k^{l,p} = Z_k^l / (B_k^{l+p} \cap Z_k^l) \tag{3.5}$$

which is well defined because $B_k^{l+p} \cap Z_k^l$ is the intersection of two subgroups of C_k^{l+p} and thus a group itself. In particular we are interested in the p-persistent k-th Betti number number $\beta_k^{l,p}$ of K^l that is the rank of $H_k^{l.p}$.

These persistent homology groups contain homology classes that are stable in the interval i to $i+p$: they are born before the index i and are still alive at index $i+p$. Persistent homology classes alive for large values of p are stable topological features of $\mathcal{X}$, while classes alive only for small values of p are unstable or noise-like topological components.

The output of the persistence homology algorithm are representations of evolution, with respect to the parameter $\varepsilon > 0$, of the topological features of $\mathcal{X}$. These representation are depicted with persistence diagrams indicating, for each homology level k, the amount and the stability of the different k-dimensional holes (Betti numbers) of the dataset $\mathcal{X}$.

We define a *persistence module* as a family of R-modules M^i and homomorphisms $\phi^i : M^i \to M^{i+1}$. we also say that the persistence module is of finite type if each M^i is finitely generated, and the maps ϕ^i are isomorphisms for $i \geq k$ and some integer k. Now we try to describe the analysis of persistent homology groups by capturing their properties in a single algebraic entity represented by a finitely generated module.

Recall that a main objective of persistent homology is to construct a summary of the evolution (with respect to ε) of the topological features of $\mathcal{X}$. This property is analysed when constructing, with the homology groups of the complexes K_i, a graded module over the polynomial ring $R = \mathbb{F}[t]$, with a field $\mathbb{F}$. We then consider a persistent module $M = \{M^i, \phi_i\}_{i \geq 0}$ and construct the graded module $\alpha(M) = \bigoplus_{i \geq 0} M^i$ over the graded polynomial ring $\mathbb{F}[t]$ defined with the action of t given by the shift $t \cdot (m^0, m^1, \ldots) = (0, \phi^0 m^0, \phi^1 m^1, \ldots)$. The crucial property of this construction is that α is a functor which defines an equivalence of categories between the category of persistence modules of finite type over $\mathbb{F}$, and the category of finitely generated non-negatively graded modules over $\mathbb{F}[t]$ [20]. In our case, considering the filtration of complexes K_i, this characterization of persistence modules provides the finitely generated $\mathbb{F}[t]$ module:

$$\alpha(M) = H_p(K_0) \oplus H_p(K_1) \oplus \cdots \oplus H_p(K_r).$$

These modules are now used in a crucial step that defines and characterizes the output of persistent homology. The main tool is the well-known structure

theorem characterizing finitely generated modules over principle ideal domains (this is why we need $\mathbb{F}$ to be a field). This property considers a finitely generated non-negatively graded module $\mathfrak{M}$ and ensures that there are $i_1, ..., i_m, j_1, ..., j_n, l_1, ..., l_n \in \mathbb{Z}$ and the following isomorphism:

$$\mathfrak{M} \cong \bigoplus_{s=1}^{m} \mathbb{F}[t](i_s) \oplus \bigoplus_{r=1}^{n} (\mathbb{F}[t]/t^{l_r})(j_r)$$

This decomposition is unique up to permutation of factors, and the notation $\mathbb{F}[t](i_s)$ denotes an i_s shift upward in grading. The relation with persistent homology is given by the fact that when a persistent homology class τ is born at K_i and dies at K_j it generates a torsion module of the form $\mathbb{F}[t]\tau/t^{j-i}(\tau)$. When a class τ is born at K_i but does not die, it generates a free module of the form $\mathbb{F}[t]\tau$.

We can now explain the concept of persistence diagrams using an additional characterization. Firstly we define a *persistence interval* as an ordered pair (i, j), where $0 \leq i < j$ for $i, j \in \mathbb{Z} \cup \infty$. Now we construct the function Q mapping a persistence interval as $Q(i, j) = (\mathbb{F}[t]/t^{j-i})(i)$, $Q(i, \infty) = (\mathbb{F}[t])(i)$, and for a set of persistence intervals $S = \{(i_1, j_1), (i_2, j_2), ..., (i_n, j_n)\}$, we have the $\mathbb{F}[t]$-module

$$Q(S) = \bigoplus_{h=1}^{n} Q(i_h, j_h).$$

The map Q turns out to be a bijective map between the sets of finite families of persistence intervals and the set of finitely generated graded modules over $\mathbb{F}[t]$.

Now, we can recap all these results by noticing that the concept of persistent diagrams can be described as the corresponding set of persistence intervals associated with the finitely generated graded module over $\mathbb{F}[t]$, constructed with the functor α, from a given filtration $\emptyset = K_0 \subset K_1 \subset K_2 \subset ... \subset K_r = K$.

Definition 3.15. Let $D = \{(b_i, d_i) \in \mathbb{R}^2 | i \in I, b_i < d_i\}$ be a persistence diagram, every point $(b_i, d_i) \in D$ is called generator of the persistent homology, represents a topological property which appears at $\mathcal{X}_{b_i}$ and disappears at $\mathcal{X}_{d_i}$. The difference $d_i - b_i$ is called *persistence* of the generator, represent its lifespan and shows the robustness of the topological property.

Examples

There are many ways to represent persistent diagrams, but one of the best known is the persistence barcodes. The persistence images are also well-known but, since we talk about them in details in the next section, we postpone their introduction.

Persistence barcodes represent in a graph of lines the behaviour of a persistence diagrams emphasizing the "life" of the topological features. In the graph the horizontal axes is associated to the evolution parameter ε, while in the vertical axes there are the homology generator in an arbitrary order.

His strength is the visual impact of the representation: in fact it is easy discriminate the relevant topological features with the ones caused by noise.

In this introduction and later on we use Ripser [3, 45] and Persim [37], respectively for the computation of Vietoris-Rips simplicial complex and for the visualization of the persistence diagrams.

Sphere

We start considering the $2-$sphere.

The Betti numbers for the $n-$sphere are [14]:

$$\beta_0,\, \beta_n = 1 \qquad \beta_i = 0 \quad \forall i = 1, ..., n-1$$

Thus in our case: $\beta_0 = 1$, $\beta_1 = 0$ and $\beta_2 = 1$.

In Figure 3.3 we consider an approximation of the sphere with 100 points, to which we added negligible noise. As we can see from the persistence barcode, there is only one line in the third graph and his lifespan is not short as the ones in the second graph. In the same way in the first graph we can see that there is only one line with high persistent and a lot of shorter ones. So we can suppose that our cloud of point has no $1-$dimensional holes ($\beta_1 = 0$) along with the hypothesis that the short lines in persistence barcode are expression of topological noise as the ones in the first graph. Instead, looking for $2-$dimensional homology, we can suppose that it has only one generator.

Usually the $0-$dimensional persistence barcodes is useful to have an idea of the position of the data, as in this case there is only one meaningful component, while in different cases, multiples lines with a high persistence, can reveal the presence of different clusters.

In Figure 3.4 we show the construction of the simplicial complex, with the Vietoris-Rips scheme, frozen on $\varepsilon = 0.2$, and the persistence barcodes where we highlight the state of evolution of the system at the time of the displayed simplicial complex.

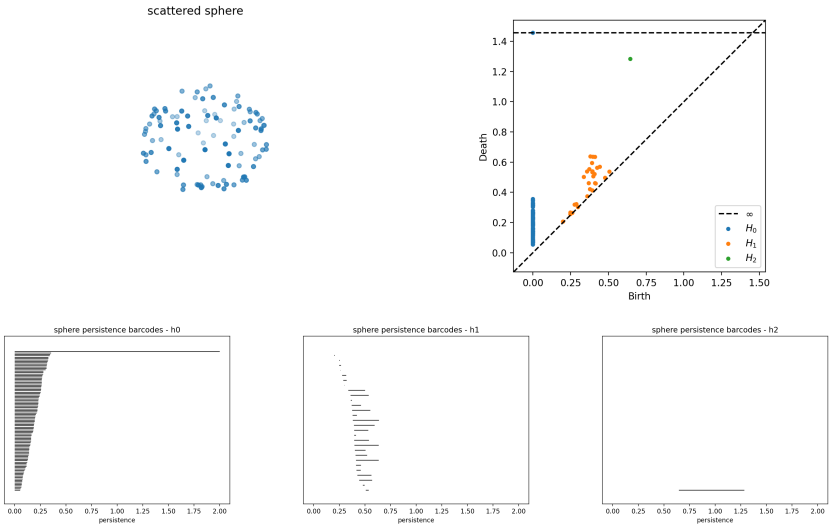

Figure 3.3: Approximation of a sphere in $\mathbb{R}^3$ with a cloud of 100 points and the persistence barcodes for (from left to right) the $0-$, $1-$ and $2-$ dimensional homology

As we can see at the value $\varepsilon = 0.2$ in the simplicial complex there is a hole, but from the persistence barcodes we know that it does not last long. We can also see that the points are not yet all connected, and there is no $2-$dimensional hole in the complex.

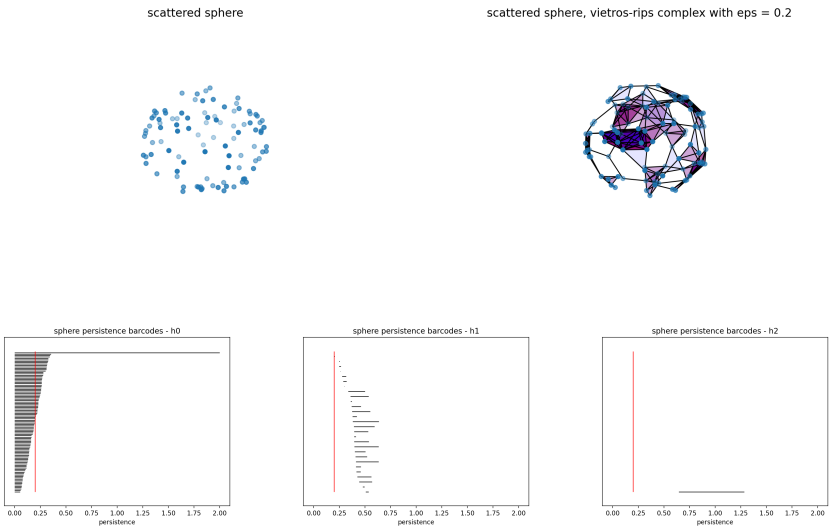

Figure 3.4: The Vietoris-Rips complex of a cloud of points approximating a sphere for $\varepsilon = 0.2$, and some correlation with the persistence barcodes

At the end, we consider a more dense cloud of points that approximates the sphere. We see that some assumptions find further confirmation. In Figure 3.5 we see that all the lines in the H_1 persistence barcode are still too short, and similarly in the persistence barcodes of H_0 and H_2, there are no major changes with respect to the barcodes obtained before.

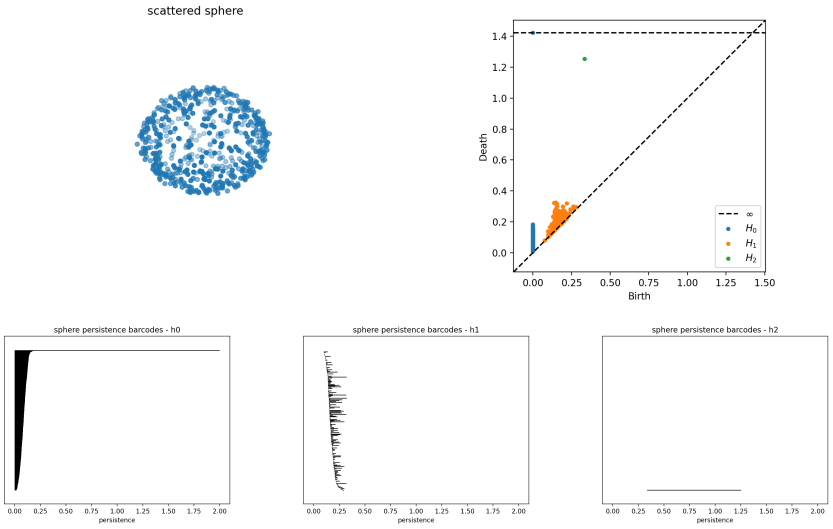

Figure 3.5: Approximation of a sphere in $\mathbb{R}^3$ with a cloud of 600 points and the persistence barcodes for (from left to right) the $0-$, $1-$ and $2-$ dimensional homology

Torus

Another interesting example is the torus.
The Betti numbers for the 1−torus are [14]:

$$\beta_0 = 1, \quad \beta_1 = 2, \quad \beta_2 = 1$$

Even in this case, in Figure 3.6 we first consider an approximation of the torus with 100 points, to which we added negligible noise. This time the results are sightly different of what we expect. In fact from the persistence barcodes we do not see the generator of the H_2 homology, and only one out of two is noticeable in the second graph. This happens, for two main reasons: the humble number of points used for the approximation, and the geometry of the torus itself. Looking at the simplicial complex at a certain time of his filtration may help to understand this results.

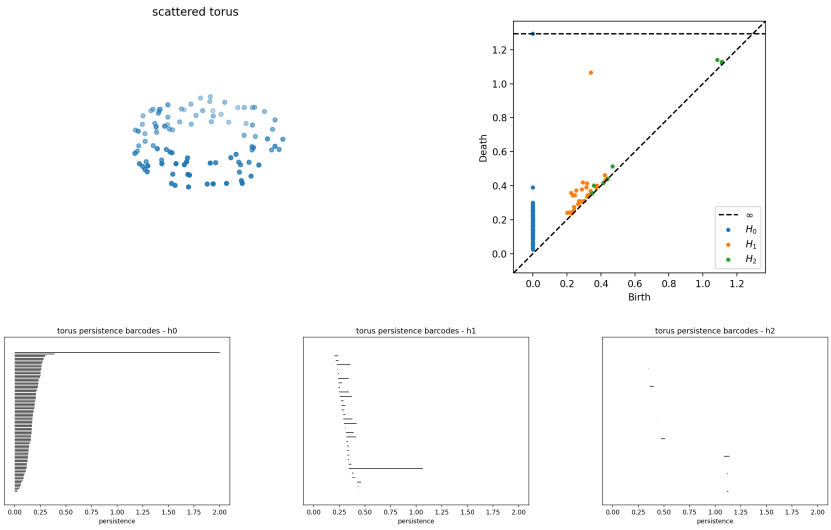

Figure 3.6: Approximation of a torus with a cloud of 100 points and its persistence barcodes for (from left to right) the 0−, 1− and 2− dimensional homology

In Figure 3.7 we show the construction of the simplicial complex, with the Vietoris-Rips scheme, at two different points of the filtration with $\varepsilon = 0.21, 0.5$. This time we do not display the H_0 and H_2 persistence barcodes, but focus our attention in the H_1 persistence barcode, in the two moment of the construction of the simplicial complex.

As we can see, with $\varepsilon > 0.5$ there are no more births and only one generator remains, and looking at the simplexes, due to random nature of points and the tight radius of the tube, we can suppose that the holes longitudinal to the body of the torus are missing, or better that all that kind of holes have a low persistence. On the other hand the generator with the greatest persistence is one of the last to be born. This support our hypothesis since the hole traced at a certain latitude needs a circular connection of the points, not achieved in the left Figure 3.7.

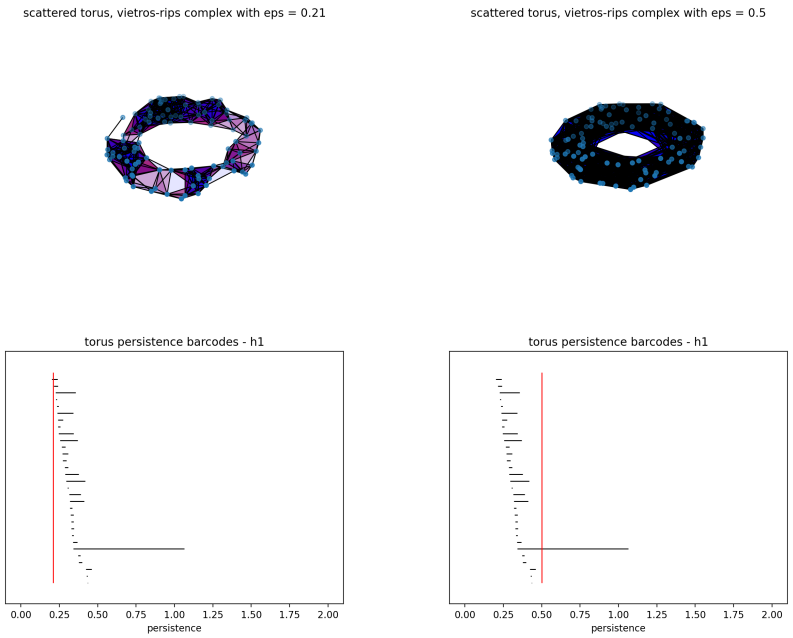

Figure 3.7: The Vietoris-Rips complex of a cloud of points approximating a torus for $\varepsilon = 0.2, 0.5$, and some correlation with the persistence barcodes

Furthermore we are considering a more deep cloud of points to approximate the torus, in addition we consider a torus with a greater internal radius, to bound complications due the thickness of the "ring". In Figure 3.8 we see that in the H_1 persistence barcode we have two enough persistent lines, and even in the H_2 persistence barcode, there is a sign of the expected generator.

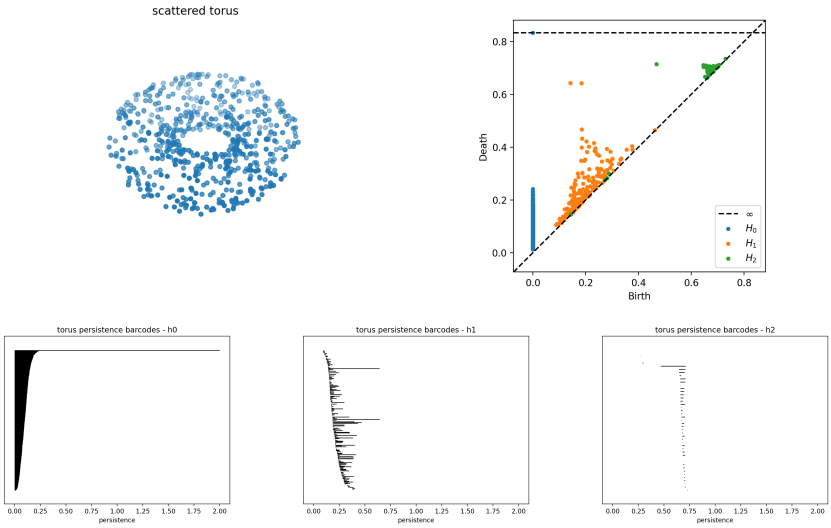

Figure 3.8: Approximation of a torus with a cloud of 100 points and its persistence barcodes for (from left to right) the 0, 1 and 2 -dimensional homology

Stability Proprieties of Persistence Diagrams

A crucial property in persistent homology is the concept of stability of persistent diagrams. We recall that for a topological space $\mathcal{X}$, and a map $h : \mathcal{X} \to \mathbb{R}$, we say that h is *tame* if the homology properties of $\{\mathcal{X}_\varepsilon, \varepsilon > 0\}$, for $\mathcal{X}_\varepsilon = h^{-1}([-\infty, \varepsilon])$, can be completely described with a finite family of sets $\mathcal{X}_{a_0} \subset \mathcal{X}_{a_1} \subset ... \subset \mathcal{X}_{a_r}$, where the positive values $\{a_i\}_{i=0}^r$ are *homology critical points*. If we denote the *persistent diagram* for $\mathcal{X}$ and $h : \mathcal{X} \to \mathbb{R}$, as $D_n(h)$, we have a summary of the stable and unstable holes generated by the filtration[20]:

$$\mathcal{X}_{a_0} \subset \mathcal{X}_{a_1} \subset ... \subset \mathcal{X}_{a_r}$$

With these concepts, the stability of persistent diagrams is a property indicating that small changes in the persistent diagram $D_n(h)$ can be controlled with small changes in the tame function $h : \mathcal{X} \to \mathbb{R}$ [11]. In order to investigate the theoretical stability of the persistent homology features, we now introduce some definitions.

Definition 3.16 (Homological critical value). Let $\mathcal{X}$ be a topological space, and $\alpha : \mathcal{X} \to \mathbb{R}$ a continuous function. A homological critical value (HCV) is a number $a \in \mathbb{R}$ for which the map induced by α:

$$H_n(\alpha^{-1}(]-\infty, a-\varepsilon[)) \to H_n(\alpha^{-1}(]-\infty, a+\varepsilon[))$$

is not an isomorphism for all $\varepsilon > 0$ and for some integer $n \geq 0$. Recall that each $\alpha^{-1}(]-\infty, a[)$ is a *sublevel* set of α.

In other words, the HCVs are the levels where the homology of the sublevel sets changes.

Definition 3.17 (Tame function). A function $f : \mathcal{X} \to \mathbb{R}$ is *tame* if it has a finite number of HCVs and the homology groups $H_k(f^{-1}(-\infty, a])$ are finite dimensional for all $k \in \mathbb{Z}$ and $a \in \mathbb{R}$.

Definition 3.18. Let $\mathcal{X}$ be a closed subset of a metric space M, and $d^x : M \to \mathbb{R}$ the function that map each point $p \in M$ to its distance from $\mathcal{X}$. The *homological feature size* of $\mathcal{X}$ denoted by *hfs*($\mathcal{X}$) is the smallest positive HCV of d^x.

An interesting result is the Homology Interference Theorem [11]. Suppose we estimate the homology of $\mathcal{X}$, closed subset of a metric space M, from another closed subset P approximating $\mathcal{X}$, which may be a finite set of points. For any two numbers $x < y$, let $\mathcal{X}_x^y$ and P_x^y be the persistent k-th homology groups of $d^{\mathcal{X}}$ and d^P associated with x and y.

Theorem 3.19 (Homology Interference Theorem). *Let $\mathcal{X}^{+\delta}$ be the parallel body consisting of all points in M at distance less than δ from $\mathcal{X}$. For all real numbers ε with $d_H(\mathcal{X}, P) < \varepsilon < hfs(\mathcal{X}/4)$ and all sufficiently small $\delta > 0$, the dimensions of the homology group of $\mathcal{X}^{+\delta}$ and $P_\varepsilon^{3\varepsilon}$ are either both infinite or both finite and equal.*

Thinking in terms of persistence diagrams this theorem states that when we have a sufficiently dense finite sample P from a space $\mathcal{X}$, the points in the persistence diagram with sufficiently small birth times can estimate the homology groups of the space.

Definition 3.20. For a tame function $\alpha : \mathcal{X} \to \mathbb{R}$, we define its persistent diagram, $D(\alpha)$ as the persistent diagram of the filtration $K_1 \subset K_2 \subset ... \subset K_r = \mathcal{X}$ where we let $K_i = f^{-1}(-\infty, a_i)$, and $a_1 < a_2 < ... < a_r$ are critical values of α.

Definition 3.21. For two nonempty multisets $\mathcal{X}, \mathcal{Y} \subset \mathbb{R}^2$ with the same cardinality the *Hausdorff distance* and *bottleneck distances* [1] are defined as:

$$d_H(\mathcal{X}, \mathcal{Y}) = \max\{\sup_{x\in\mathcal{X}} \inf_{y\in\mathcal{Y}} ||x - y||_\infty, \sup_{y\in\mathcal{Y}} \inf_{x\in\mathcal{X}} ||x - y||_\infty\} \tag{3.6}$$

$$d_B(\mathcal{X}, \mathcal{Y}) = \inf_{\gamma} \sup_{x\in\mathcal{X}} ||x - \gamma(x)||_\infty \tag{3.7}$$

where we consider all possibly bijection of multisets $\gamma : \mathcal{X} \to \mathcal{Y}$. Here, we use:

$$||p - q||_\infty = \max\{|p_1 - q_1|, |p_2 - q_2|\} \qquad \text{for } q,\, p \in \mathbb{R}$$

[1] By convention, all points on the diagonal are taken with infinite multiplicity. This facilitates the definitions of the p-Wasserstein and bottleneck distances, discussed in 4

We also have that[11]:

$$d_H(\mathcal{X}, \mathcal{Y}) \leq d_B(\mathcal{X}, \mathcal{Y}) \tag{3.8}$$

And another stability property for bottleneck distance [22].

Proposition 3.22. *Let $\mathcal{X}$ and $\mathcal{Y}$ be finite subset in a metric space (M, d_M). Then the persistence diagrams satisfy*

$$d_B(D_q(\mathcal{X}), D_q(\mathcal{Y})) \leq d_H(\mathcal{X}, \mathcal{Y})$$

This proposition provides a geometrically intuitive idea of stability: given a dataset $\mathcal{X}$ of a finite number of points, we consider as $\mathcal{Y}$ the data with some noise $\varepsilon < d_H(\mathcal{X}, \mathcal{Y})$. If we consider a point $(b, d) \in D_q(\mathcal{X})$, then we can find at least one generator $(\tilde{b}, \tilde{d}) \in \mathcal{Y}$ such that $\tilde{b} \in (b-\varepsilon, b+\varepsilon)$ and $\tilde{d} \in (d-\varepsilon, d+\varepsilon)$. Thus the similarity of the persistence diagrams is guaranteed by this stability result.

Lastly the main theorem of [11]:

Theorem 3.23. *Let $\mathcal{X}$ be a topological space with tame functions $\alpha, \beta : \mathcal{X} \to \mathbb{R}$. Then, the following stability property holds:*

$$d_B(D(\alpha), D(\beta)) \leq ||\alpha - \beta||_\infty \tag{3.9}$$

CHAPTER 4

Kernels and Persistent Homology

The construction of kernels for comparing persistent diagrams has recently become an important topic due to the fact that measures like the bottleneck distance are inefficient to compute in practice [20]. A better conceptual and numerical strategy is to use reproducing kernels and their ability to translate unstructured data in a more convenient setting of linear algebra. Several methods have been proposed in the last years, some of which we will introduce later in the chapter, mainly focusing on classification problems.

Usually persistence diagrams are produced from two different types of input data.

(i) If the data are a point cloud, then we produce persistence diagrams using the Vietoris–Rips filtration.

(ii) If the data are from a real-valued function, then we produce persistence diagrams using the sublevel set filtration.

We have seen in Section 3.2 how to handle the first case. Now we focus on the second case. Let $\mathscr{X}$ be a topological space and let $f : \mathscr{X} \to \mathbb{R}$ be real-value function. The sublevel sets of the function f are defined as $R(y) = f^{-1}((-\infty, y])$. In other words, a sublevel set is the set of all points x for which $f(x) \leq y$. One can study the function f, through its sublevel set $f^{-1}((-\infty\varepsilon])$, with $\varepsilon \in \mathbb{R}$. In fact one can study map f using the persistent homology of the resulting filtration of topological spaces, known as the sublevel set filtration:

Given $\varepsilon_1 < \varepsilon_2 < ... < \varepsilon_n$ the sublevel set filtration is:

$$f^{-1}((-\infty, \varepsilon_1]) \subset f^{-1}((-\infty, \varepsilon_2]) \subset \ ... \, f^{-1}((-\infty, \varepsilon_n])$$

As we increase y, the critical points of the function are visited. The homology of $f(x) \leq y$ changes (with respect to y) if a homology class is born (or dies) by visiting f(x).

In both settings, the output of the persistent homology computation is a collection of persistence diagrams encoding homological features of the data across a range of scales. Denoting with $\mathfrak{D}$ the set of all persistence diagrams. The space $\mathfrak{D}$ can be equipped by metrics as studied by [11].

The bottleneck distance, that was introduced in the previous chapter, can be embedded in *p-Wasserstein distances*, which are a more general type of distances defined for any positive real number p:

$$d_{W,p}(D,E) = (\inf_{\gamma} \sum_{x \in D} ||x - \gamma(x)||_{\infty}^{p})^{\frac{1}{p}} \tag{4.1}$$

Here, as before, γ ranges over all bijections from the elements of D to the elements of E. Notice that as $p \to \infty$, we have that $d_{W,\infty} = d_B$. The Wasserstein distance is induced by the Wasserstein metric which is the most used one in the context of persistence diagrams and optimal transport problems.

We have the following result bounding the p-Wasserstein distance in terms of the $\mathscr{L}_\infty$ distance [12]:

Theorem 4.1. *Assume that $\mathscr{X}$ is a compact triangulable metric space such that for every l-Lipschitz function f on $\mathscr{X}$ and for $k \geq 1$, the degree k total persistence $\sum_{(b,d)\in D_f}(d-b)^k$ is bounded above above by some constant C. Let f, g be two L-Lipschitz piecewise linear functions on $\mathscr{X}$. Then for all $p \geq k$ we have:*

$$d_{W,p}(D_f, D_g) \leq (LC)^{\frac{1}{p}} ||f-g||_{\infty}^{1-\frac{k}{p}} \tag{4.2}$$

Some studies consider the vectorization of a persistence diagram [7]. In these methods, a vector representing a persistence diagram is typically expressed in a Euclidean space $\mathbb{R}^k$ or a function space $\mathscr{L}_p$. The aim of these methods is to make topological data analysis with principal component analysis or support vector machines classification using statistical information gained from vector representation of the persistence diagram.

For the purpose of this thesis we are more interested in methods focused on building kernels for persistence diagrams. Before introducing the kernels recently developed we briefly discuss other vectorization techniques used in machine learning, and often compared with the works we will subsequently present.

Persistence Landscapes

Persistence Landscape [7] is a well-known vectorization of persistence diagrams approach in topological data analysis. Let D be a persistence

diagram, then

$$\lambda_D(k,t) = \text{k-th largest value of } \min\{t-b_i, d_i-t\}_+$$

is the persistence landscape of D, where $c_+ = \max\{c, 0\}$. λ_D is a vector on $\mathscr{L}^2(\mathbb{N}\times\mathbb{R})$. Persistence landscape can be used as a linear positive definite kernel on $\mathscr{L}^2(\mathbb{N}\times\mathbb{R})$:

$$\kappa_{PL}(D,E) := \langle \lambda_D, \lambda_E \rangle_{\mathscr{L}^2(\mathbb{N}\times\mathbb{R})} = \int_{\mathbb{R}} \sum_{k=1} \lambda_D(k,t)\lambda_E(k,t)\, dt \qquad (4.3)$$

Since a persistence landscape does not have any parameters, we do not need to consider the parameter tuning. However, the integral computation must be done and it requires much computational time. Let $\mathfrak{D}$ be a collection of persistence diagrams with at most m points. Since $\lambda_{D_i}(k,t) \equiv 0$ for any $k > m$, $t \in \mathbb{R}$, $i = 1, ..., n$, evaluating λ_{D_i} have time complexity $\mathcal{O}(m\log(m))$

Persistence Images

As a finite dimensional vector representation of a persistence diagram, persistence images are proposed in [1]. The idea is to produce a persistence surface from a persistence diagram by taking a weighted sum of Gaussians centered at each point. We create vectors, or *persistence image*s, by integrating our surfaces over a grid, allowing machine learning techniques for finite-dimensional vector spaces to be applied to persistence diagrams. Persistence images are stable, and distinct homology dimensions may be concatenated together into a single vector to be analyzed simultaneously.

Firstly, let D be a persistence diagram in birth-death coordinates. Let $T : \mathbb{R}^2 \to \mathbb{R}^2$ be the linear transformation $(b,d) \mapsto T(b,d) = (b, d-b)$ and let $T(D)$ be the transformed multiset in birth-persistence coordinates. Let $\phi_u : \mathbb{R}^2 \to \mathbb{R}$ be a differentiable probability distribution with mean $u = (u_x, u_y) \in \mathbb{R}^2$. Authors used normalized symmetric Gaussian in applications:

$$g_u(x,y) := \frac{1}{2\pi\sigma^2} \exp\big(-\frac{(x-u_x)^2 + (y-u_y)^2}{2\sigma^2}\big)$$

Definition 4.2. Let $f : \mathbb{R}^2 \to \mathbb{R}$ be a non-negative weight function that is zero along the horizontal axis, continuous and piecewise differentiable. For a persistence diagram D, the corresponding *persistence surface* $\rho_D : \mathbb{R}^2 \to \mathbb{R}$ is the function

$$\rho_D(z) = \sum_{u\in T(D)} f(u)\phi_u(z) \qquad (4.4)$$

The weight function f is fundamental to ensure the stability of the transformation of persistence diagrams in persistence images, which is proven in [1, Section 5]. In their paper-related works authors use as weight function $f(x, y) = w_b(y)$ with:

$$w_b(t) = \begin{cases} 0 & \text{if } t \leq 0 \\ \frac{t}{b} & \text{if } 0 < t < b \\ 1 & \text{if } t \geq b \end{cases}$$

where $b > 0$ is the persistence value of the most persistent feature between the trials of the experiment.

Definition 4.3. For a persistence diagram D, its *persistence image* is the collection of the pixels $I(\rho_D)_p = \iint_p \rho_D \, dy \, dx$.

Persistence images provide a convenient way to combine persistent diagrams of different homological dimensions into a single vector. In fact one can concatenate the persistence images vectors for $H_0, H_1, ..., H_k$ into a single vector representing all homological dimensions simultaneously, and then use this concatenated vector as input into machine learning algorithms.

The choices of parameters in persistence images as resolution, distribution and weight function grant them high flexibility. On the other hand these choices are non-canonical

In [1] authors have shown persistence images improved classification accuracy over persistence landscapes and persistence diagrams on sampled data of common topological spaces at multiple noise levels. Additionally the classification accuracy is robust to the choice of parameters for building persistence images, providing evidence that it is not necessary to perform large-scale parameter searches to achieve reasonable classification accuracy. This indicates the utility of persistence images even when there is no prior knowledge of the underlying data. The flexibility of persistence images allows for customization tailored to a wide variety of real-world data sets.

In the following sections we focus our analysis on kernel-based methods for persistence diagrams. In the last decade four kernels have been proposed to deal with classification problems. The original contributes of the next section is contained in papers which are published on internationals journals [10, 22, 27, 35].

4.1 Persistence Kernels

Persistence Scale-Space Kernel

One of the first kernels defined for persistence diagrams in the *Persistence Scale-Space Kernel*.

Considering a set $\mathfrak{D}$. The kernel will be defined via a feature map $\Phi_\sigma : \mathfrak{D} \to \mathscr{L}_2(\Omega_{ad})$ with $\Omega_{ad} \subset \mathbb{R}^2$ denoting the closed half plane above the diagonal.

To motivate the definition of Φ_σ, the authors point out that the set of persistence diagrams, i.e., multisets of points in $\mathbb{R}^2$, does not possess a Hilbert space structure per se. However, a persistence diagram D can be uniquely represented as a sum of Dirac delta distributions, one for each point in D. Since Dirac deltas are functionals in the Hilbert space $H^{-2}(\mathbb{R}^2)$, we obtain a canonical Hilbert space structure for persistence diagrams by adopting this point of view.

Unfortunately, the induced metric on $\mathfrak{D}$ does not take into account the distance of the points in the diagrams or to the diagonal, and therefore cannot be robust against perturbations of the diagrams. This issue is solved by using the sum of Dirac deltas as an initial condition for a heat diffusion problem with a Dirichlet boundary condition on the diagonal. The solution of this partial differential equation is a $\mathscr{L}_2(\Omega_{ad})$ function for any chosen scale parameter $\sigma > 0$.

Definition 4.4. Let $\Omega_{ad} = \{\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2 : x_2 \geq x_1\}$ denote the space above the diagonal, and let $\delta_\mathbf{x}$ denoted a Dirac delta centered at the point $\mathbf{x}$. For a given persistence diagram D, we now consider the solution $u : \Omega_{ad} \times \mathbb{R}_{\geq 0} \to \mathbb{R}$, such that $(\mathbf{x}, t) \to u(\mathbf{x}, t)$ of the following partial differential equation:

$$\Delta_\mathbf{x} u = \partial_t u \qquad in \quad \Omega_{ad} \times \mathbb{R}_{\geq 0} \tag{4.5}$$

$$u = 0 \qquad on \quad \partial\Omega_{ad} \times \mathbb{R}_{\geq 0} \tag{4.6}$$

$$u = \sum_{\mathbf{y} \in D} \delta_\mathbf{y} \qquad on \quad \Omega_{ad} \times 0 \tag{4.7}$$

The feature map $\Psi_\sigma : \mathfrak{D} \to \mathscr{L}_2(\Omega_{ad})$ at scale $\sigma > 0$ of a persistence diagram D is now defined as $\Psi_\sigma(D) = u|_{t=\sigma}$. This map yields the persistence scale space kernel κ_σ on $\mathfrak{D}$ as:

$$\kappa_\sigma(D, E) = \langle \Psi_\sigma(D), \Psi_\sigma(E) \rangle_{\mathscr{L}_2(\Omega_{ad})}. \tag{4.8}$$

Note that $\Psi_\sigma(D) = 0$ for some $\sigma > 0$ implies that $u = 0$ on $\Omega_{ad} \times \{0\}$, which means that D has to be the empty diagram. From linearity of the solution operator it now follows that Ψ_σ is an injective map.

The solution of the partial differential equation can be obtained by extending the domain from Ω to $\mathbb{R}^2$ and replacing 4.7 with

$$u = \sum_{\mathbf{y} \in D} \delta_{\mathbf{y}} - \delta_{\bar{\mathbf{y}}} \qquad \text{on } \mathbb{R}^2 \times \{0\}. \tag{4.9}$$

Here $\bar{\mathbf{y}} = (b, a)$ is $\mathbf{y} = (a, b)$ mirrored at the diagonal. It can be shown that restricting the solution of this extended problem to Ω_{ad} yields a solution for the original equation. It is given by convolving the initial condition 4.9 with a Gaussian kernel:

$$u(t, \mathbf{x}) = \frac{1}{4\pi t} \sum_{\mathbf{y} \in D} \exp(-\frac{||\mathbf{x} - \mathbf{y}||^2}{4t}) - \exp(-\frac{||\mathbf{x} - \bar{\mathbf{y}}||^2}{4t}). \tag{4.10}$$

Using this closed form solution of u, we can derive a simple expression for evaluating the kernel explicitly [35]

$$\kappa_\sigma(D, E) = \frac{1}{8\pi\sigma} \sum_{\substack{\mathbf{y} \in D \\ \mathbf{z} \in E}} \exp(-\frac{||\mathbf{y} - \mathbf{z}||^2}{8\sigma}) - \exp(-\frac{||\mathbf{y} - \bar{\mathbf{z}}||^2}{8\sigma}). \tag{4.11}$$

Note that the kernel can be computed in $O(|F| \cdot |G|)$ time, where $|F|$ and $|G|$ denote the cardinality of the multisets F and G respectively.

Theorem 4.5. *The kernel κ_σ is 1-Wasserstein stable. More precisely*

$$||\Phi_\sigma(D) - \Phi_\sigma(E)||_{\mathscr{L}_2(\Omega_{ad})} \leq \frac{1}{2\sigma\sqrt{\pi}} d_{W,1}(D, E). \tag{4.12}$$

The left-hand side of 4.12 is called persistence scale space distance d_{κ_σ}. Note that the right-hand side of 4.12 decreases as σ increases. Adjusting σ accordingly allows us to counteract the influence of noise in the input data, which causes an increase in $d_{W,1}(D, E)$.

A natural question is whether our stability result extends to p-Wasserstein distances for $p > 1$. Firstly note that the kernel is additive, i.e. $\kappa(E \cup F, D) = \kappa(E, D) + \kappa(F, D)$ for all $F, E, D \in \mathfrak{D}$. By choosing $D = \emptyset$, we see that if κ is additive then $\kappa(\emptyset, E) = 0$ for all $E \in \mathfrak{D}$. We can further say that a kernel κ is trivial if $\kappa(D, E) = 0$ for all $D, E \in \mathfrak{D}$.

Then the following theorem can be proven:

Theorem 4.6. *A non-trivial additive kernel κ on persistence diagrams is not stable with respect to $d_{W,p}$ for any $1 < p \leq \infty$.*

Lastly the authors propose a comparison between the Persistence scale-space kernel and persistence landscape, introduced in [7], for this purpose they consider the persistence landscape kernel κ^L, associated with the feature map $\Phi^L : \mathfrak{D} \to \mathscr{L}_2(\mathbb{R}^2)$.

The results obtained show the need to have different methods for different type of analysis. In fact persistence landscape, if considering its kernel version, can be used for their remarkable topological features but the induced distance tends to over-emphasize points of high persistence. Considering classes of persistence diagrams that differ only in their points with low persistence, the distance associated with persistence landscape kernel will be dominated by the by variations in the points of high persistence. Hence instances of these classes would be inseparable. Persistence space-scale kernel instead allow to distinguish classes that differ only in their points with low persistence, because they do not over-emphasize points with high persistence.

In [35] persistence scale-space kernels are applied in the context of shape classification/retrieval and texture classification with good results. Moreover the tuning of the scale parameter σ brings numerous practical benefits.

Persistence Weighted Gaussian Kernel

In [22, 23] is introduced a stable kernel following the idea of the kernel mean embedding of distributions [43], the Persistence Weighted Gaussian Kernel.

Let Ω be a locally compact Hausdorff space, let $M_b(\Omega)$ be the space of all finite signed Radon measures on Ω, and let κ be a bounded measurable kernel on Ω. Then we define a mapping from $M_b(\Omega)$ to $\mathcal{H}_\kappa$ by

$$E_\kappa : M_b(\Omega) \to \mathcal{H}_\kappa, \quad \mu \mapsto \int \kappa(\cdot, \mathbf{x})\, d\mu(\mathbf{x}), \tag{4.13}$$

which is well defined since $\int ||\kappa(\cdot, \mathbf{x})||_{\mathcal{H}_\kappa}\, d\mu(\mathbf{x})$ is finite. Let denote

$$\mathscr{C}_0(\Omega) = \{f : \Omega \to \mathbb{R} \,|\, f \in \mathscr{C}(\Omega), \forall \varepsilon > 0 \\ \exists K \text{ compact } \in \Omega \text{ such that } \sup_{\mathbf{x} \in K^c} |f(\mathbf{x})| \leq \varepsilon\}.$$

A kernel κ on Ω is said to be $\mathscr{C}_0-$kernel if $\kappa(\mathbf{x}, \mathbf{x})$ is $\mathscr{C}_0(\Omega)$ as a function of $\mathbf{x}$. Moreover, if κ is a $\mathscr{C}_0$-kernel the associated Reproducing Kernel Hilbert Space $\mathcal{H}_\kappa$ is a subspace of $\mathscr{C}_0(\Omega)$. A $\mathscr{C}_0$-kernel is called $\mathscr{C}_0$-universal if $\mathcal{H}_\kappa$ is dense in $\mathscr{C}_0(\Omega)$. When κ is $\mathscr{C}_0$-universal, the vector $E_\kappa(\mu)$ defined in 4.13 uniquely determines the measure μ in the RKHS. Furthermore if κ is $\mathscr{C}_0$-universal then the map E_κ is injective. Thus $d_\kappa(\mu, \nu) := ||E_\kappa(\mu) - E_\kappa(\nu)||_{\mathcal{H}_\kappa}$ define a distance on $M_b(\Omega)$ [43].

In vectorizing persistence diagrams, it is useful to mitigate the importance of generators located near the diagonal, since they tend to be caused by noise. The authors proposed two different ways of embeddings, which turn out to introduce the same inner products for two persistence diagrams.

The first method involves the introduction of a weighted measure for a persistence diagram D

$$\mu_D^w := \sum_{\mathbf{x}\in D} w(\mathbf{x})\delta_{\mathbf{x}},$$

where $w(\mathbf{x}) > 0$ is a weight for every $\mathbf{x} \in D$ and $\delta_{\mathbf{x}}$ is the Dirac delta in $\mathbf{x}$. A proper choice for the weight function $w(\mathbf{x})$ will be discussed later. As discussed before given a $\mathscr{C}_0$-universal kernel κ on $\Omega_{ad} = \{(b,d) \in \mathbb{R}^2 | b < d\}$, the measure μ_D^w can be embedded as an element of the RKHS via

$$\mu_D^w \mapsto E_\kappa(\mu_D^w) := \sum_{\mathbf{x}\in D} w(\mathbf{x})\, \kappa(\cdot, \mathbf{x}), \tag{4.14}$$

then we can use $E_\kappa(\mu_D^w) \in \mathcal{H}_\kappa$ as a representation of the persistence diagram.

The second construction involves the following weighted kernel

$$\kappa^w(\mathbf{x}, \mathbf{y}) := w(\mathbf{x})w(\mathbf{y})\kappa(\mathbf{x}, \mathbf{y}), \tag{4.15}$$

where w is the same weight function as above. Then the mapping

$$E_{\kappa^w} : \mu_D \mapsto \sum_{\mathbf{x}\in D} w(\mathbf{x})w(\cdot)\kappa(\cdot, \mathbf{x}) \in \mathcal{H}_\kappa \tag{4.16}$$

also define a vectorization of persistence diagram.

Notice that the inner products introduced by two RKHS vectors 4.14 and 4.16 are the same:

$$\langle E_\kappa(\mu_D^w), E_\kappa(\mu_D^w)\rangle_{\mathcal{H}_\kappa} = \langle E_\kappa(\mu_D), E_\kappa(\mu_D)\rangle_{\mathcal{H}_{\kappa^w}}.$$

Moreover we can prove the following proposition [23].

Proposition 4.7. *Let κ be a $\mathscr{C}_0$-universal on Ω_{ad} and w be a positive function on Ω_{ad}. Then the mapping*

$$\mathcal{H}_\kappa \to \mathcal{H}_{\kappa^w}, \quad f \mapsto w\, f$$

defines an isomorphism between the RKHSs. Under this isomorphism, $E_\kappa(\mu_D^w)$ and $E_{\kappa^w}(\mu_D)$ are identified.

For practical applications, the map $E_\kappa\left(\mu^w_{D_q(\mathcal{X})}\right)$ from a dataset $\mathcal{X}$ to the vectorization of the persistence diagram $D_q(\mathcal{X})$ should be stable with respect to perturbations of the data.

The following proposition can be proven [22]:

Proposition 4.8. *Let D and E be persistence diagrams, a $\mathscr{C}_0$-universal kernel κ satisfy:*

(K) *$\exists$ constants B_κ, $L_\kappa > 0$ such that*

$$||\kappa(\cdot,\mathbf{x})||_{\mathcal{H}_\kappa} \leq B_\kappa, \quad ||\kappa(\cdot,\mathbf{x}) - \kappa(\cdot,\mathbf{y})||_{\mathcal{H}_\kappa} \leq L_\kappa \, ||\mathbf{x}-\mathbf{y}||_\infty \quad (\mathbf{x},\mathbf{y} \in \mathbb{R}^2) \tag{4.17}$$

and a weight function w satisfy:

(W1) *For any persistence diagrams D and E, and any multi-bijection $\gamma : D \cup \Delta \to E \cup \Delta$, there exist B_1, $L_1 > 0$ such that*

$$\sum_{\mathbf{x}\in D} |w(\mathbf{x})| \leq B_1, \qquad \sum_{\mathbf{x}\in D\cup\Delta} |w(\mathbf{x}) - w(\gamma(\mathbf{x}))| \leq L_1 \sup_{\mathbf{x}\in D\cup\Delta} ||\mathbf{x}-\gamma(\mathbf{x})||_\infty \tag{4.18}$$

Then,

$$||E_\kappa(\mu^w_D) - E_\kappa(\mu^w_E)||_{\mathcal{H}_\kappa} \leq (L_\kappa B_1 + B_\kappa L_1) d_{W_B}(D,E). \tag{4.19}$$

Authors choose as weight function

$$w_{arc}(\mathbf{x}) = \arctan(C \cdot \mathrm{pers}(\mathbf{x})^p), \quad (C > 0,\, p \in \mathbb{Z}_{>0}),$$

where C, p are parameters used to control the effect of the persistence and $\mathrm{pers}(\mathbf{x})$ is the persistence of $\mathbf{x}$. w_{arc} is a bounded and increasing function of $\mathrm{pers}(\mathbf{x})$ and, by restricting a class of persistence diagrams to that of a Čech complex filtration, it satisfies (W1) [22].

With this weight function the authors define a positive definite kernel, the *Persistence Weighted Gaussian Kernel* (PWGK):

$$\kappa_{PWGK}(\mathbf{x},\mathbf{y}) := w_{arc}(\mathbf{x}) w_{arc}(\mathbf{y}) \exp\left(-\frac{||\mathbf{x}-\mathbf{y}||^2}{2\sigma^2}\right). \tag{4.20}$$

Therefore, considering a Čech complex filtration, we have the bottleneck stability for PWGK:

Theorem 4.9. *Let M be a triangulable compact subspace in $\mathbb{R}^d$, $\mathcal{X}$, $\mathcal{Y} \in M$ be finite subsets, $p > d+1$, and a $\mathscr{C}_0$-universal kernel κ satisfy (K). Then,*

$$||E_\kappa(\mu^{w_{arc}}_{D_q(\mathcal{X})}) - E_\kappa(\mu^{w_{arc}}_{D_q(\mathcal{Y})})||_{\mathcal{H}_\kappa} \leq L_{\kappa,arc} d_{W_B}(D_q(\mathcal{X}), D_q(\mathcal{Y}))$$

where $L_{\kappa,arc}$ is a constant independent of $\mathcal{X}$ and $\mathcal{Y}$.

If we denote with k_G the Gaussian kernel, since the constant $L_{k_G,arc}$ is independent of $\mathcal{X}$ and $\mathcal{Y}$, recalling Proposition 3.22, we have that

$$\mathcal{P}_{finite}(M) \to \mathcal{H}_{k_G}, \quad \mathcal{X} \mapsto E_{k_G}(\mu^{w_{arc}}_{D_q(\mathcal{X})})$$

is Lipschitz continuous, where $\mathcal{P}_{finite}(M)$ is the set of finite subsets in a triangulable compact subspace $M \subset \mathbb{R}^d$.

Moreover results on the stability with respect to 1-Wasserstein distance for PWGK can be proven too.

Proposition 4.10. *Let D and E be persistence diagrams, a $\mathscr{C}_0$-universal kernel κ satisfy (K), and a weight function w satisfy*

(W2) *For any $\mathbf{x}$, $\mathbf{y} \in \mathbb{R}^2$, there exist constants $B_2, L_2 > 0$ such that*

$$|w(\mathbf{x})| \leq B_2, \quad |w(\mathbf{x}) - w(\mathbf{y})| \leq L_2 ||\mathbf{x} - \mathbf{y}||_\infty.$$

Then,

$$||E_\kappa(\mu_D^w) - E_\kappa(\mu_E^w)||_{\mathcal{H}_\kappa} \leq (L_\kappa B_2 + B_\kappa L_2) d_{W_1}(D, E). \tag{4.21}$$

Remark 4.11. Note that the assumption (W2) is weaker than (W1).

Some weight function that satisfy (W2) are:

$$w_{\text{pers}}(\mathbf{x}) := \begin{cases} 0 & \text{if pers}(\mathbf{x}) < 0 \\ \frac{1}{L} & \text{if } 0 \leq \text{pers}(\mathbf{x}) \leq L \\ 1 & \text{if pers}(\mathbf{x}) > L \end{cases} \tag{4.22}$$

where $L > 0$ is a parameter. Even a simpler function like $w_{one}(\mathbf{x}) \equiv 1$ works, so these functions are 1-Wasserstein stable. But none of them satisfies (W1), thus is still unknown if bottleneck stability holds for this weight functions.

On the other hand w_{arc} satisfies (W2) choosing $p = 1$ by $B_2 = \frac{\pi}{2}$ and $L_2 = 2C$ without any assumption on persistence diagrams structure.

More formally:

Corollary 4.12. *Let D and E be persistence diagrams and a $\mathcal{C}_0$-universal kernel κ satisfy (K). Then,*

$$||E_\kappa(\mu_D^{w_{pers}}) - E_\kappa(\mu_E^{w_{pers}})||_{\mathcal{H}_\kappa} \leq (L_\kappa + \frac{2B_\kappa}{L}) d_{W_1}(D, E),$$
$$||E_\kappa(\mu_D^{w_{one}}) - E_\kappa(\mu_E^{w_{one}})||_{\mathcal{H}_\kappa} \leq L_\kappa d_{W_1}(D, E),$$
$$||E_\kappa(\mu_D^{w_{arc}}) - E_\kappa(\mu_E^{w_{arc}})||_{\mathcal{H}_\kappa} \leq (\frac{\pi L_\kappa}{2} + 2B_\kappa C) d_{W_1}(D, E) \quad \textit{with } p = 1 \textit{ in } w_{arc}.$$

Furthermore for $p > 1$ in general w_{arc} does not satisfy (W2) since Ct^p is not Lipschitz continuous with respect $t \in \mathbb{R}$. Similarly to Theorem 4.9, restricting to the class of persistence diagrams 1-Wasserstein stability can be proven:

Corollary 4.13. *Let M be a triangulable compact subset in $\mathbb{R}^d$, $\mathcal{X}, \mathcal{Y} \subset M$ be finite subset, $p > d+1$, and a $\mathscr{C}_0$-universal kernel κ satisfy (K). Then,*

$$||E_\kappa(\mu^{w_{arc}}_{D_q(X)}) - E_\kappa(\mu^{w_{arc}}_{D_q(Y)})||_{\mathcal{H}_\kappa} \leq (\frac{\pi L_\kappa}{2} + B_\kappa C \frac{4p(p-1)}{p-1-d} C_M \, diam(M)^{p-1-d}) d_{W_1}(D_q(X), D_q(Y)) \tag{4.23}$$

for some constant $C_M > 0$.

Once persistence diagrams are represented as RKHS vectors, any kernel methods can be applied to those vectors by defining a kernel over the vector representation. In a similar way to standard vectors, two simple choices can be done: considering the inner product as a linear kernel or as a non-linear kernel on the RKHS.

$$\kappa_L(D, E; \kappa, w) := \langle E_\kappa(\mu^w_D), E_\kappa(\mu^w_E) \rangle_{\mathcal{H}_\kappa} = \sum_{\mathbf{x} \in D} \sum_{\mathbf{y} \in E} w(\mathbf{x}) w(\mathbf{y}) \kappa(\mathbf{x}, \mathbf{y}) \tag{4.24}$$

In the first case we have a *(κ, w)-linear kernel*.

$$\kappa_G(D, E; \kappa, w) := \exp(-\frac{1}{2\tau^2} ||E_\kappa(\mu^w_D) - E_\kappa(\mu^w_E)||) \quad \tau > 0 \tag{4.25}$$

Or the second one a *(κ,w)-Gaussian kernel*. This is also the most used since better performance has been observed with non-linear kernels in the context of complex tasks [28].

Concerning the computational cost of the methods, if the persistence diagrams contains at most m points, each element of the Gram matrix $(\kappa_G(D_i, D_j; k_G, w)_{i,j})$ involves $\mathcal{O}(m^2)$ evaluations of $e^{-\frac{||\mathbf{x}-\mathbf{y}||^2}{2\tau^2}}$. If we consider in our application a collection $\mathfrak{D}$ of n persistence diagrams, the total complexity will be $\mathcal{O}(n^2 M^2)$.

Authors solved this computational issue with random Fourier features [33], reducing the computational complexity of the approximated Gram matrix to $\mathcal{O}(mnM + n^2 M)$, where M is a constant, which is linear to m. This technique could be particularly useful since usually $m \gg n$ and $m > M$, but can be sensitive to the choice of τ.

The main advantage of Persistence Weighted Gaussian Kernel over previous work, as Persistence Images and Persistence Scale-Space Kernel, is the stability with respect to the bottleneck distance, even if is satisfied

only for specific filtrations. Another strength of the PWGK is the flexibility with which it can control the effect of persistence independently of the Gaussian parameter σ.

Lastly we consider a comparison between the performance of the algorithms introduced so far applied to a simple SVM classification problem [22]. In the experiment the authors design data sets so that important generators close to the diagonal must be taken into account to solve the classification task.

The results show that the PWGK and the Gaussian kernel on the persistence image with w_{arc} and large mesh size have higher classification rates (85% accuracy), than the other methods (Persistence scale-space kernel : 50%, Persistence Landscapes kernel : 50%, and Persistence Images kernel : 55%). These unfavorable results must be caused by the difficulty in handling the local and global locations of generators simultaneously.

Furthermore it is observed that the classification accuracies are not sensitive to p. Thus, authors set $p = 5$ because the assumption $p > d+1$ in Theorem 4.9 ensures the continuity in the kernel embedding of persistence diagrams and all data points used in others experiments are obtained from $\mathbb{R}^3$.

Sliced Wasserstein Kernel

Although the above defined persistence scale-space kernel and persistence weighted Gaussian kernel are both stable kernels, in the sense that the metric they induce in their respective RKHS is bounded above by the distance between persistence diagrams, there is no evidence that their induced RKHS distances are discriminative and therefore follow the geometry of the diagram distances.

One of the reasons why the derivation of kernels for persistence diagrams is not straightforward is that the natural metrics between persistence diagrams are not trivial, since the diagram distances are not negative semi-definite. Moreover is experimentally observed in appendix A of [34] that $d_{W,1}$ is not conditionally definite. However, a relaxation of this metric called the Sliced Wasserstein distance has been shown to be negative semi-definite [32].

That is the idea behind the introduction of the *Sliced Wasserstein Kernel* in [10] which is both stable and discriminative.

The Wasserstein distance [46], is a distance between probability measures. For the purpose of their work authors will focus on the 1-Wasserstein distance for nonnegative, not necessary normalized, measures on the real line.

Let μ and ν be two nonnegative measures on the real line such that $|\mu| = \mu(\mathbb{R})$ and $|\nu| = \nu(\mathbb{R})$, both equal to the same number r, then consider:

$$\mathcal{W}(\mu,\nu) = \inf_{P\in\Pi(\mu,\nu)} \iint_{\mathbb{R}\times\mathbb{R}} |\mathbf{x}-\mathbf{y}| P(dx,dy), \tag{4.26}$$

$$\mathcal{Q}_r(\mu,\nu) = r\int_{\mathbb{R}} |M^{-1}(\mathbf{x}) - N^{-1}(\mathbf{x})|\, dx, \tag{4.27}$$

$$\mathcal{L}(\mu,\nu) = \inf_{f\in 1-Lipschitz} \int_{\mathbb{R}} f(\mathbf{x})[\mu(dx)-\nu(dx)]. \tag{4.28}$$

where $\Pi(\mu,\nu)$ is the set of measures on $\mathbb{R}^2$ with marginals μ and ν, and M^{-1} and N^{-1} the generalized quantile functions of the probability measures $\frac{\mu}{r}$ and $\frac{\nu}{r}$ respectively.

Proposition 4.14. *We have $\mathcal{W} = \mathcal{Q}_r = \mathcal{L}$. Additionally (i) $\mathcal{Q}_r$ is conditionally negative definite on the space of measures of mass r; (ii) for any three positive measures μ, ν, γ such that $|\mu| = |\nu|$, we have $\mathcal{L}(\mu+\gamma,\nu+\gamma) = \mathcal{L}(\mu,\nu)$.*

The idea underlying this metric is to slice the plane with lines passing through the origin, to project the measures onto these lines where $\mathcal{W}$ is computed, and to integrate those distances over all possible lines. Formally:

Definition 4.15. Given $\theta \in \mathbb{R}^2$ with $||\theta||_2 = 1$, Let $L(\theta)$ denote the line $\{\lambda\theta \,|\, \lambda\in\mathbb{R}\}$, and let $\pi_\theta : \mathbb{R}^2 \to L(\theta)$ be the orthogonal projection onto $L(\theta)$. Let D and E be two persistence diagrams, and let $\mu_1^\theta = \sum_{p\in D} \delta_{\pi_\theta(p)}$ and $\mu_{1\Delta}^\theta = \sum_{p\in D} \delta_{\pi_\theta\circ\pi_\Delta(p)}$, and similarly for μ_1^θ, where π_Δ is the orthogonal projection onto the diagonal. Then, the Sliced Wasserstein distance is defined as:

$$SW(D,E) := \frac{1}{2\pi}\int_{\mathbb{S}_1} \mathcal{W}(\mu_1^\theta + \mu_{2\Delta}^\theta, \mu_2^\theta + \mu_{1\Delta}^\theta)\, d\theta. \tag{4.29}$$

Recalling Proposition 4.14, the conditionally positive definiteness of the Sliced Wasserstein distance can be proven in the space of finite and bounded persistence diagrams $\mathcal{D}_f^b$:

Lemma 4.16. *the Sliced Wasserstein distance is conditionally definite on $\mathcal{D}_f^b$*

therefore we can define the *Sliced Wasserstein Kernel*:

$$\kappa_{SW}(D,E) := \exp\left(-\frac{SW(D,E)}{2\sigma^2}\right). \tag{4.30}$$

The peculiarity of the Kernel 4.30, compared with others kernels defined so far, is that the metric with which his RKHS is equipped is

induced by the sliced Wasserstein distance, which is strongly equivalent to 1-Wasserstein distance $d_{W,1}$.

Theorem 4.17. *Sliced Wasserstein distance is stable with respect to $d_{W,1}$ on $\mathcal{D}_f^b$. For any D, E $\in \mathcal{D}_f^b$, one has*

$$SW(D, E) \leq 2\sqrt{2} d_{W,1}(D, E).$$

For the discriminative property, stronger assumptions have to be done on persistence diagrams, namely their cardinalities have not only to be finite, but also bounded by some $N \in \mathbb{N}^*$.

Theorem 4.18. *Sliced Wasserstein distance is discriminative with respect to $d_{W,1}$ on $\mathcal{D}_N^b$. For any D, E $\in \mathcal{D}$, one has*

$$\frac{1}{2M} d_{W,1}(D, E) \leq SW(D, E),$$

where $M = 1 + 2N(2N - 1)$.

In particular, Theorems 4.17 and 4.18 allow us to show that d_{SW}, the distance induced by κ_{SW} in its RKHS, is also equivalent to $d_{W,1}$ in a wide sense: there exist continuous, positive and monotone functions g, h such that $g(0) = h(0) = 0$ and $h \circ d_{W,1} \leq d_{SW} \leq g \circ d_{W,1}$.

Moreover the condition on cardinalities can be relaxed. In fact it can be proven that the feature map Φ_{SW} induced by κ_{SW} is injective when persistence diagrams are in $\mathcal{D}_f^b$. In particular κ_{SW} can become an universal kernel considering $\exp(\kappa_{SW})$ [24].

Proposition 4.19. *The feature map Φ_{SW} is continuous and injective with respect to $d_{W,1}$ on $\mathcal{D}_f^b$.*

Speaking of computational complexity, the *exact* computation of the kernel for persistence diagrams in general position can be done in $\mathcal{O}(N^2 \log(N)$ time, where N is the cardinality of the persistence diagrams. A persistence diagram is said to be in *general position* if it has no triplet of aligned points. In practice, given two persistence diagrams D and E, the kernel is evaluated with $\tilde{D}$ and $\tilde{E}$, which is the modified diagrams with infinitesimally small random perturbations.

Since the time complexity in the exact computation case could be significant in applications, an approximating computation is also proposed. The approximation time complexity is $\mathcal{O}(MN \log(N))$, where M is the approximating factor (with $M \ll N$). This approximation is still a negative semi-definite kernel. This approximation of κ_{SW} is useful since, as shown in [10], authors have observed empirically that $M \sim 10$ is sufficient to get good classification accuracies.

Time complexity is the major issue of the proposed kernel, since other analysed kernels have linear complexity with respect to the cardinality of the persistence diagrams.

The parameter σ of Equation 4.30 is the only one of this kernel and is, in practice, easier to tune than the parameters of the other kernels analysed so far when using grid search. Indeed, as is the case for all infinitely divisible kernels, the Gram matrix does not need to be recomputed for each choice of σ, since it only suffices to compute all the Sliced Wasserstein distances between persistence diagrams in the training set once. As we will see in the last chapter, this property is particularly appealing, since it strongly speeds up the tuning of parameters through cross validation.

Sliced Wasserstein Kernel has tested on several benchmark application compared with PWGK and PSSK. Authors show on several datasets substantial improvements in accuracy and training times (when tuning parameters is done with grid search) over competing kernels.

Persistence Fisher Kernel

The last kernel method for persistence diagrams we propose is the Persistence Fisher Kernel introduced in [27]. In their work authors explore an alternative Riemannian geometry, namely the Fisher information metric [2], for persistence diagrams.

Persistence diagrams can be considered as a discrete measure $\mu_D = \sum_{\mathbf{u}\in D} \delta_{\mathbf{u}}$, where $\delta_{\mathbf{u}}$ is the Dirac delta on $\mathbf{u}$. Therefore, the Wasserstein geometry is a popular choice in kernel-based methods for persistence diagrams content, together with related metrics to compute distances, on the set of persistence diagrams with bounded cardinalities.

Firstly we briefly introduce the Fisher information geometry:

Given a bandwidth $\sigma > 0$, for a set Θ, one can smooth and normalize μ_D as follows

$$\rho_D := \left[\frac{1}{Z} \sum_{\mathbf{u}\in D} N(\mathbf{x}; \mathbf{u}, \sigma I) \right]_{\mathbf{x}\in\Theta} \tag{4.31}$$

where $Z = \int_\Theta \sum_{\mathbf{u}\in D} N(\mathbf{x}; \mathbf{u}, \sigma I)\, dx$, N is a Gaussian function and I is an identity matrix. Therefore, each persistence diagram can be regarded as a point in a probability simplex $\mathbb{P} := \{\rho \,|\, \int \rho(\mathbf{x})\, dx = 1, \rho \geq 0\}$. In case, one chooses Θ as an entire Euclidean space, each persistence diagram turns into a probability distribution as in [1].

Fisher Information Metric is a well-known Riemannian geometry on the probability simplex $\mathbb{P}$, especially in information geometry. Given two points ρ_D and ρ_E in $\mathbb{P}$, the Fisher information metric is defined as:

$$d_{\mathcal{P}}(\rho_D, \rho_E) = \arccos\Big(\int \sqrt{\rho_D(\mathbf{x})\rho_E(\mathbf{x})}\, dx\Big) \tag{4.32}$$

In the following we denote $D_\Delta := \{\Pi_\Delta(\mathbf{u}) \,|\, \mathbf{u} \in D\}$, where $\Pi_\Delta(\mathbf{u})$ is the projection of a point $\mathbf{u}$ on the diagonal set Δ.

Definition 4.20. Let D , E be two finite and bounded persistence diagrams. The Fisher information metric between D and E is defined as follows,

$$d_{\mathrm{FIM}}(D, E) := d_{\mathcal{P}}\Big(\rho_{(D\cup E_\Delta)}, \rho_{(E\cup D_\Delta)}\Big) \tag{4.33}$$

Lemma 4.21. *Let $\mathcal{D}_f^b$ be the set of finite and bounded persistence diagrams. Then, $(d_{FIM} - \tau)$ is negative definite on $\mathcal{D}_f^b$ for all $\tau \geq \frac{\pi}{2}$.*

Motivated by the lemma above and Theorem 1.10, we introduce a positive definite kernel: the *Persistence Fisher Kernel*

$$\kappa_{PF}(D, E) := \exp\big(-td_{\mathrm{FIM}}(D, E)\big) \tag{4.34}$$

where t is a positive scalar since we can rewrite the kernel as

$$\kappa_{PF}(D, E) = \alpha \exp\big(-t(d_{\mathrm{FIM}}(D, E) - \tau)\big) \tag{4.35}$$

where $\tau \geq \frac{\pi}{2}$ and $\alpha = \exp(-t\tau) > 0$.

Moreover the κ_{PF} is positive definite without approximation.

The square distance induced by the Persistence Fisher kernel, denoted as $d^2_{\kappa_{PF}}$, can be computed by the Hilbert norm of the difference between two corresponding mappings. Given two persistence diagrams D and E, we have:

$$d^2_{\kappa_{PF}}(D, E) = \kappa_{PF}(D, D) + \kappa_{PF}(E, E) - 2\kappa_{PF}(D, E)$$

Since κ_{PF} is based on the Fisher information geometry, it is interesting to bound the kernel induced distance $d^2_{\kappa_{PF}}$ with respect to the corresponding Fisher information metric d_{FIM} between persistence diagrams.

Lemma 4.22. *Let D, $E \in \mathcal{D}_f^b$ be a persistence diagrams. Then,*

$$d^2_{\kappa_{PF}}(D, E) \leq 2td_{FIM}(D, E)$$

where t is a parameter of κ_{PF}.

The Lemma above implies that the Persistence Fisher kernel is stable on Riemannian geometry in a similar sense as work of [23] and [35] on Wasserstein geometry.

Another important result is the infinite divisibility of the kernel

Lemma 4.23. *The Persistence Fisher kernel κ_{PF} is infinite divisible.*

Thus, the Gram matrix of the kernel does not need to be recomputed for each choice of t (Equation 4.34), since it suffices to compute the Fisher information metric between persistence diagrams in training set only once. This property is shared with the Sliced Wasserstein kernel. However, neither Persistence Scale Space kernel nor Persistence Weighted Gaussian kernel has this property.

Given two finite persistence diagrams D and E with cardinalities bounded by N, for the computation of the kernel, is considered the finite set $\Theta := D \cup E_\Delta \cup E \cup D_\Delta$ without multiplicity in $\mathbb{R}^2$ for smoothed and normalized measures $\rho_{(\cdot)}$ 4.31. Then let m be the cardinality of Θ, we have $m \leq 4N$. Consequently the time complexity of $\rho_{(\cdot)}$ is $\mathcal{O}(mN)$. For acceleration, we propose to apply the Fast Gauss Transform to approximate the sum of Gaussian functions in $\rho_{(\cdot)}$ with a bounded error. The time complexity of $\rho_{(\cdot)}$ is reduced to $\mathcal{O}(m + N)$. Due to the low dimension of points in persistence diagrams ($\mathbb{R}^2$), this approximation by the Fast Gauss Transform is very efficient in practice. Additionally, $d\mathcal{P}$ (Equation 4.32) is evaluated between two points in the m-dimensional probability simplex $\mathbb{P}_{m-1} = \{\mathbf{x} \,|\, \mathbf{x} \in \mathbb{R}^m, \mathbf{x} \geq 0, ||\mathbf{x}||_1 = 1\}$. So, the time complexity of the Persistence Fisher kernel κ_{PF} between two smoothed and normalized measures is $\mathcal{O}(m)$.

Hence, the time complexity of κ_{PF} between D and E is $\mathcal{O}(N^2)$, or $\mathcal{O}(N)$ for the acceleration version with Fast Gauss Transform.

Lastly in [27] authors evaluated the Persistence Fisher kernel with SVM on many benchmark dataset. The performance of the kernel was compared with the other kernels for persistence diagrams discussed so far: Persistence Scale-Space kernel (κ_{PSS}), Persistence Weighted Gaussian kernel (κ_{PWG}) and Sliced Wasserstein kernel (κ_{SW}). For the parameter tuning the guidelines from the original papers were followed.

For the Persistence Fisher kernel, there are 2 hyper-parameters: t (Equation 4.34) and σ for smoothing measures (Equation 4.31). The choice of them is done through cross validation.

The accuracy results shows that κ_{PF} outperforms other kernels followed by κ_{SW}. Moreover, looking at computational time with approximations, the κ_{PF} is faster than κ_{PSS}, but slower than κ_{PWG} and even more so than κ_{SW}.

Variably Scaled Persistence Kernels

In section 1.2 we introduced the *variably scaled kernels*, proposed as alternative kernels that shift the problem of the parameter tuning into the problem of chose a suitable scaling function.

Can a similar approach be introduced in the context of kernels for persistence diagrams?

The main difference between standard kernels and kernels for persistence diagrams is that the underlying space is different. The persistent diagrams are a collection of topological features, in particular of birth-death couples in $\mathbb{R}^2_{ad}$. There is not a meaningful reason behind the introduction of a scale function, in its classical sense, applied to a persistence diagram, since outside $\mathbb{R}^2_{ad}$ we do not longer talk of persistence diagrams as long all the metrics discussed so far.

Thinking of the algebraic derivation of the persistent homology, another way to approach this idea is to add a a couple to the persistence diagram.

Following this intuition we can define a Variably Scaled Persistence Kernel.

Definition 4.24. Let $\kappa : \mathfrak{D} \times \mathfrak{D} \to \mathbb{R}$ be a kernel for persistence diagrams and let $\Psi : \mathfrak{D} \to \mathbb{R}^2_{ad}$ be a scale function. A variably scaled persistence kernel κ_Ψ on $\mathfrak{D} \times \mathfrak{D}$ is defined as

$$\kappa_\Psi(D, E) := \kappa(D \cup \Psi(D), E \cup \Psi(E)) \tag{4.36}$$

for $D, E \in \mathfrak{D}$

If we consider the kernel introduced so far, except for persistence scale space kernel, they are of the kind:

$$\kappa(D, E) = \exp\left(-\gamma d(D, E)\right)$$

where $d(\cdot, \cdot)$ is the distance induced by the metric introduced along with the kernel. Thus it is easy to prove that if κ is a (strictly) positive definite kernel, so is κ_ψ. The same result follows for (strictly) negative definite kernels.

More precisely, the kernel κ_Ψ inherits all properties of κ.

As scale function we propose a sort of center of mass:

$$\Psi(D) = \frac{1}{W} \sum_{x \in D} w(x)\, x \tag{4.37}$$

where $w(x) \in \mathbb{R}^+$ and $W = \sum_{x \in D} w(x)$.

Some common choices for w are $\frac{1}{|D|}$, where we denote as $|D|$ the number of generator of the persistence diagram D, or the persistence of x.

We denote as *center of uniform mass* the scale function

$$\Psi(D) = \frac{1}{|D|} \sum_{x \in D} x \tag{4.38}$$

and as *center of persistence* the scale function:

$$\Psi(D) = \frac{1}{\sum_{x \in D} \mathrm{pers}(x)} \sum_{x \in D} \mathrm{pers}(x)\, x. \tag{4.39}$$

Note that $\sum_{x \in D} \mathrm{pers}(x) = \mathrm{pers}\left(\sum_{x \in D} x\right)$.

For specific application it could be useful to give different weights accordingly to the needs of the situations. For example, we may have meaningful information on low persistence generators, in this case we can consider $w(x) = \frac{1}{p(x)}$.

CHAPTER 5

Application to Alzheimer's Disease Diagnosis

5.1 Introduction

Dementia is an umbrella term used to describe a range of neurological conditions affecting the brain that get worse over time. It is the loss of cognitive functioning (thinking, remembering, and reasoning) and behavioral abilities to such an extent that it interferes with the daily routines of a person. These functions include memory, language skills, visual perception, problem solving, self-management, and the ability to focus and pay attention. Some people with dementia cannot control their emotions, and their personalities may change. Dementia ranges in severity from the mildest stage, when it has just started to affect the functioning of a person, to the most severe stage, when the person must depend completely on others for basic activities of living.

In the past, dementia was sometimes referred to as "senility" and was thought to be a normal part of aging, likely because it is more common as people age. As many as half of all people age 85 or older may have dementia. But dementia is not a normal part of aging. Not everyone develops dementia as they get older, and, in rare cases, some people develop dementia in midlife.

Dementia is the result of changes in the brain that cause nerve cells, or neurons, to stop working properly and eventually die. Researchers have connected changes in the brain to certain forms of dementia, but in most cases the specific brain changes that cause dementia are unknown. Moreover the overlap in symptoms of various dementias can make it hard to get an accurate diagnosis. Currently, there are no cures for these types of disorders.

Although some people may be diagnosed with general dementia, to best tailor treatment, it is ideal to know the specific type. One of the most known type of dementia is the Alzheimer's disease (AD).

AD is an irreversible, progressive brain disorder that slowly destroys memory and cognitive functions. It slowly gets worse over time. People with this disease progress at different rates and in several stages. Symptoms may get worse and then improve, but until an effective treatment for the disease itself is found, the personal ability will continue to decline over the course of the disease. AD is currently ranked as the sixth leading cause of death in the United States, but recent estimates indicate that the disorder may rank third, just behind heart disease and cancer, as a cause of death for older people.

AD is named after Dr. Alois Alzheimer. In 1906, Dr. Alzheimer noticed changes in the brain tissue of a woman who had died of an unusual mental illness. Her symptoms included memory loss, language problems, and unpredictable behavior. After she died, he examined her brain and found many abnormal clumps (now called amyloid plaques) and tangled bundles of fibers (now called neurofibrillary, or tau, tangles). These plaques and tangles in the brain are still considered some of the main features of AD. Another feature is the loss of connections between nerve cells (neurons) in the brain. Neurons transmit messages between different parts of the brain, and from the brain to muscles and organs in the body. During the preclinical stage of AD, people seem without symptoms, but abnormal deposits of proteins form amyloid plaques and tau tangles throughout the brain. When healthy neurons stop functioning, because of the amyloid plagues and tau tangles, they lose connections with other neurons, and die. Although the damage initially takes place only near the hippocampus and the entorhinal cortex, as more neurons die, additional parts of the brain are affected and begin to shrink. By the final stage of Alzheimer's, damage is widespread, and brain tissue has shrunk significantly.

Early clinical diagnosis is challenging because AD-specific changes begin years before the patient becomes symptomatic: its cellular hallmarks can accumulate in the living brain up to 30 years before the characteristic symptoms of dementia can be identified. Moreover, brain changes in AD are difficult to compare from those in normal aging. Therefore, a major focus is lead to identify such changes at the earliest possible stage by leveraging neuroimaging data. One specific interest is to go beyond group analysis (i.e., between clinically different groups), rather to "learn" patterns characteristic of neurode-generation using machine learning (ML) methods [30].

5.2 Materials and Methods

Since AD leads to a progressive shrinkage of cerebral tissue, a key feature in diagnosis of the disease is the cortical thickness. Using medical software Freesurfer [17] we can, from MRI images, estimate the cortical thickness

in various points of the brain. Then, by exploiting the high complexity and the importance of the geometry structure, we can build a simplex on these points as in Section 3.2.

Statistical machine learning methods provide means for learning a hypothesis from a set of training examples. In the context of a specific disorder, the training data is given as a set comprised of both diseased and control subjects, and our objective is to learn a pattern in such examples to help predicting the target variables for new test cases.

In our study we aim to test the effectiveness of the different methods proposed in Chapter 4 to build a discriminative kernel in the context of SVM classification of cortical thickness persistence diagrams.

We evaluated the accuracy of our method by performing experiments on data collected as part of the OASIS Brains Datasets.

The Open Access Series of Imaging Studies (OASIS) is a project aimed at making neuroimaging data sets of the brain freely available for the scientific community.

In particular, OASIS-3 is a compilation of MRI and PET imaging and related clinical data for 1098 participants who were collected across several ongoing studies in the Washington University Knight Alzheimer Disease Research Center over the course of 15 years. Participants include 605 cognitively normal adults and 493 individuals at various stages of cognitive decline ranging in age from 42 to 95 years. The OASIS-3 dataset contains over 2000 MR sessions, including multiple structural and functional sequences. PET metabolic and amyloid imaging includes over 1500 raw imaging scans and the accompanying post-processed files from the PET Unified Pipeline are also available in OASIS-3. OASIS-3 also contains post-processed imaging data such as volumetric segmentations and PET analyses. Imaging data is accompanied by dementia and APOE status and longitudinal clinical and cognitive outcomes [26].

In the selection of a proper study group, we consider the needs of our methods. Not all the subject in the Oasis-3 dataset have both an estimate of the cortical thickness and a clear diagnosis, so we do not take them into account. Moreover, after this initial selection, we further reduce the set of subjects in order to have a balanced dataset.

A summary of demographic and neuropsychological details of the subjects considered in our study is presented in Table 5.1[1].

For each subject, we build the persistence diagrams using the estimation of cortical thickness on 34 points in both right and left hemisphere of the brain, for a total of 64 values. For simplicity in the study we consider the same coordinates of the above mentioned points

[1]MMSE in Table 5.1 is for Mini-Mental State Examination [18]

	AD (mean)	AD (st.dev.)	Control (mean)	Control (st.dev.)
No. of subjects	225	-	248	-
Gender (F/M)	114/111	-	126/122	-
Hand preference (A/L/R)	5/23/197	-	6/26/216	-
Age at entry	74.41	7.60	65.21	9.62
Education (years)	14.77	3.08	16.04	2.51
MMSE	20.33	6.38	29.27	1.30

Table 5.1: Demographic details and baseline cognitive status measures of the study population.

for all subjects. The coordinates are computed with the scipy toolbox [47]. From this coordinates we build the Vietoris-Rips complex and then compute the persistence diagrams of the subjects using persim and ripser [37, 45]. We extract 0, 1 and 2-dimensional topological features, but for the experimental setup we consider the generators asssociated with H_1 and H_2.

In Figure 5.1 we show two examples of persistence diagrams built in this way.

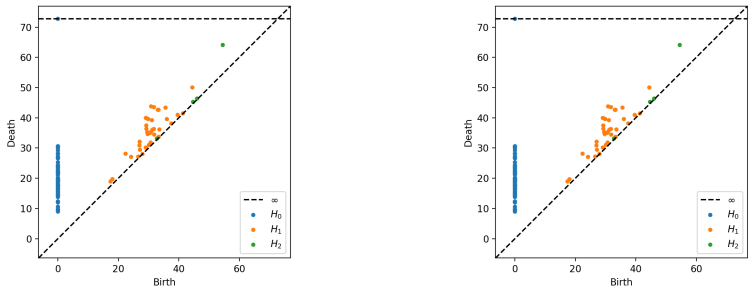

Figure 5.1: Persistence diagram of an AD subject with a MMSE of 7 (right) and the persistence diagram of a control subject with a MMSE of 30 (left)

We compare the performance, in terms of classification accuracy and computational cost, of the kernel introduced in Chapter 4, namely: the Persistence Scale-Space Kernel (PSSK), the Persistence Weighted Gaussian Kernel (PWGK), the Sliced Wasserstein Kernel (SWK) and the Persistence Fisher Kernel (PFK). This classification is based on 1-dimensional generators of persistence diagrams.

Furthermore, we employ the so-called variable scaled persistence kernel technique to all the defined above kernels, and compare them with their original version in a classification task based on 2-dimensional generators of persistence diagrams. As scale function we consider the *center of persistence* (Equation 4.39).

All kernels are handled using Python 3.8 and the modulus scikit-learn [31]. The results are averaged over 6 runs, with random 70%/30% splitting of the data for training and test, on a 2.6Ghz Dual-Core Intel Core i5. The cost factor C of SVM is cross validated in the following grid: $\{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$.

Free and open source PYTHON software concerning this application is available at

https://github.com/reevost/master_thesis

In the validation step we consider a 5-fold CV on the training set. The hyperparameters of the methods are chosen following the guidelines of the authors.

- **PSSK**.
  $\sigma \in \{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50, 100, 500, 1000\}$.
- **PWGK**.
  C_{arc}, σ and τ are chosen in $\{0.01, 0.1, 1, 10, 100\}$, while p is fixed[2] to 10.
- **SWK**.
  The range of possible values for the bandwidth σ in PSWK is obtained by computing the squareroot of the median, the first and the last deciles of the training matrix, then by multiplying these values by the following range of 5 factors: 0.01, 0.1, 1, 10 and 100.
- **PFK**.
  σ is chosen in $\{0.01, 0.1, 1, 10, 100\}$ and $\frac{1}{t}$ from $\{q_1, q_2, q_5, q_{10}, q_{20}, q_{50}\}$ where q_i is the $i\%$ quantile of a subset of Fisher information metric observed on the training set. After few experiments, we denote a poor results with these candidates hyperparameters, Then we decide to change the candidates for hyperparameter $\frac{1}{t}$. The new possible values are chosen from $\{q_1, q_{20}, q_{50}\}$ multiplied by the following factors: 0.001, 1, 1000.

For the validation we consider the f1-score.

$$\text{f}_1\text{-score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{5.1}$$

where:

[2]because the assumption $p > d + 1$ ensures the continuity of the kernel embedding (Theorem 4.9), and in our case $d = 4$

$$\text{precision} = \frac{\text{true positive}}{\text{true positive} + \text{false positive}} \tag{5.2}$$

$$\text{recall} = \frac{\text{true positive}}{\text{true positive} + \text{false negative}} \tag{5.3}$$

5.3 Results

We report the considered algorithms performance over the dataset introduced before. To have a meaningful comparison with results of [30], we firstly consider the classification result of the proposed kernel on the H_1 persistence diagrams of the subjects. In Table 5.2 we display the results obtained.

As second experiment we consider the same task but on H_2 persistence diagrams. Here we compare each kernel with its variable scaled variant. in Tables 5.3-5.6 we show some further details on the comparison between the considered kernels and their variably-scaled version (VS-kernel).

Lastly, we demonstrate that VS-kernels are discriminative showing, in Table 5.7, the results of a SVM classification with Gaussian kernel based on the set of center of persistence or center of mass and of the H_2 persistence diagrams.

	Accuracy	f_1-score	testing time	validation time
PSSK	0.78	0.77	24	8727
PWGK	0.74	0.72	54	239786
PFK	0.74	0.73	3871	162582
PSWK	0.76	0.72	70	220

Table 5.2: Results of SVM classification on H_1 persistence diagrams.

	PSSK	VS-PSSK
accuracy	0.78	0.81
precision (AD)	0.75	0.83
recall (AD)	0.80	0.77
f_1-score (AD)	0.77	0.80
precision (Control)	0.80	0.80
recall (Control)	0.76	0.85
f_1-score (Control)	0.78	0.82

Table 5.3: Comparison between PSSK and VS-PSSK on SVM classification of the H_2 persistence diagrams.

	PWGK	VS-PWGK
accuracy	0.76	0.80
precision (AD)	0.83	0.82
recall (AD)	0.64	0.77
f_1-score (AD)	0.72	0.79
precision (Control)	0.72	0.79
recall (Control)	0.88	0.84
f_1-score (Control)	0.79	0.84

Table 5.4: Comparison between PWGK and VS-PWGK on SVM classification of the H_2 persistence diagrams.

	PSWK	VS-PSWK
accuracy	0.76	0.82
precision (AD)	0.81	0.86
recall (AD)	0.67	0.74
f_1-score (AD)	0.73	0.80
precision (Control)	0.73	0.79
recall (Control)	0.85	0.89
f_1-score (Control)	0.79	0.84

Table 5.5: Comparison between PSWK and VS-PSWK on SVM classification of the H_2 persistence diagrams.

	PFK	VS-PFK
accuracy	0.73	0.67
precision (AD)	0.72	0.67
recall (AD)	0.72	0.64
f_1-score (AD)	0.72	0.65
precision (Control)	0.74	0.68
recall (Control)	0.74	0.70
f_1-score (Control)	0.74	0.69

Table 5.6: Comparison between PFK and VS-PFK on SVM classification of the H_2 persistence diagrams.

5.4 Discussion and Conclusion

Differently from the examples considered in the original articles of the kernels, the persistence diagrams used in this application had a very small number of generators. This has meant that the computational cost of the related kernels has not influenced results as in the examples reported in comparing articles. Instead, the number of parameters of each kernel was much more relevant, since the brief time needed to validate a combination of hyperparameters. In fact, while the testing times are quite similar to each other, the validation times are influenced

	center of uniform mass	center of persistence
accuracy	0.56	0.65
precision (AD)	0.55	0.60
recall (AD)	0.55	0.90
f_1-score (AD)	0.55	0.72
precision (Control)	0.58	0.82
recall (Control)	0.58	0.42
f_1-score (Control)	0.58	0.56

Table 5.7: Results of SVM classification task performed on the center of mass of the persistence diagrams. As scale function we chose the center of uniform mass (Equation 4.38 and the center of persistence (Equation 4.39)

by the number of parameters. The PWGK has a set of 125 possible parameters against the 10/15 of the other methods.

In this framework the infinitely divisible kernels has a great advantage. For this reason, despite his theoretical computational cost, PSWK is by far the fastest kernel, helped also by having only one hyperparameter to tune.

Concerning the accuracy results, we found that the persistence diagrams can be a discriminative representation of our brain. Moreover, all the proposed kernels reached a solid accuracy.

Comparing the results with a similar study made by Pachauri et al. in [30], where the classification is based only on cortical thickness estimates on MRI, we can say that our approach leads to slightly better results in some cases ($\approx 75\%$ in [30]).

Furthermore we presented an original approach to employ the VSKs in the persistence diagrams framework, showing their effectiveness in classification experiments in the context of Alzheimer's Disease diagnosis.

Appendices

APPENDIX A

Cross Validation

Cross-validation is one of the most widely used data resampling methods to estimate the true prediction error of models and to tune model parameters.

Cross-validation was originally employed to evaluate the predictive validity of linear regression equations used to forecast a performance criterion from scores on a battery of tests. It was found that the multiple correlation coefficient within the original sample used to assign values to regression weights gave an optimistic impression of the predictive effectiveness of the regression equation when applied to future observations. In order to investigate this phenomenon the cross-validation procedure was employed. Two samples from the same population were drawn. The first, the calibration sample, was used to calibrate the regression equation, that is, to assign values to the regression weights. The weighted linear composite of predictors, with previously calibrated weights, was then correlated with the criterion in a second sample, the validation sample. The cross- validation index so obtained yielded a realistic impression of the predictive effectiveness of the linear composite of tests. This two-sample form of cross-validation can be wasteful because only about half of the available observations are used for calibration purposes leading to less effective calibrations. The remaining observations are required for the validation sample [6].

More formally, let $\Omega \subset \mathbb{R}^d$, let $\mathcal{X} = \{x_1, ..., x_n\} \subset \Omega$ be a set of data and $\mathcal{Y} = \{y_1, ..., y_n\} \subset \{0, 1\}^n$ a set of labels associated to $\mathcal{X}$, with $d, n \in \mathbb{N} \setminus \{0\}$. In this case we considered the label for a binary classification problem, but it may be different if we consider other tasks. Then we can define the labeled dataset $\mathcal{Z} = \{(x_i, y_i), |x_i \in \mathcal{X}, y_i \in \mathcal{Y}\}$.

Now let suppose that our aim is to describe a process where the labelled dataset $\mathcal{Z}$ is involved. We also have a model m_γ, where $\gamma = (\gamma_i)_{i=1,...,m}$ are the vector of real parameters, for a some $m \in \mathbb{N}$.

The role of Cross Validation is to identify the set of parameter that better represent the data in the set $\mathcal{Z}$, along with unknown samples

from $\Omega \setminus \mathcal{X}$. The cross validation techniques are based on the following approach: splitting the initial data into two set, one of the sets is used to calibrate the hyperparameter (via CV), after that the remaining set is used to test the model.

There are various CV routines. One of the simplest, but not for this less effective, is the k-fold CV. Fixed Γ, the set of all possible hyperparameter vectors, the cross validation scheme is the following:

(i) we fix $\gamma \in \Gamma$.

(ii) Let $k \in \mathbb{N}, 1 < k \leq n$. We divide the set $\mathcal{Z}$ into k disjoint sets $\{\mathcal{Z}_i\}_{i=1,...,k}$ called *folds*.

(iii) We choose $k-1$ training folds, and we leave the last one as validation fold. We denote as $V_j = \mathcal{Z}_j$ the $j-$th validation fold and as $T_j = \bigcup_{i \in I} \mathcal{Z}_i$, with $I = \{1, ..., i-1, i+1, ..., k\}$. After that we train the model with the hyperparameter vector γ on T_j.

(iv) We take a cost function, for example the zero-one function 2.1, then we evaluate the model and compute the cost on V_j

(v) We repeat the training-validation process for every $j = 1, ..., k$. Finally we consider the average of all cost gained during the process and save it assigning the value to the hyperparameter vector γ.

(vi) repeat the whole process for all $\gamma \in \Gamma$. Lastly we choose the γ associated with the minimum averaged cost.

There are other type of cross validation as *stratified CV* or *nested CV*, with various kind of complexity and specific for certain type of problems.

Bibliography

[1] Adams, H. et al. 'Persistence images: A stable vector representation of persistent homology'. In: *Journal of Machine Learning Research* vol. 18 (2017).

[2] Amari, S.-i. and Nagaoka, H. *Methods of Information Geometry*. Vol. 191. Jan. 2000.

[3] Bauer, U. 'Ripser: efficient computation of vietoris-rips persistence barcodes'. In: *arXiv preprint arXiv:1908.02518* (2019).

[4] Berg, C., Christensen, J. P. R. and Ressel, P. *Harmonic analysis on semigroups: theory of positive definite and related functions*. Vol. 100. Springer, 1984.

[5] Bozzini, M. et al. 'Interpolation with variably scaled kernels'. In: *IMA Journal of Numerical Analysis* vol. 35, no. 1 (2015), pp. 199–219.

[6] Browne, M. W. 'Cross-validation methods'. In: *Journal of mathematical psychology* vol. 44, no. 1 (2000), pp. 108–132.

[7] Bubenik, P. 'Statistical topological data analysis using persistence landscapes.' In: *J. Mach. Learn. Res.* vol. 16, no. 1 (2015), pp. 77–102.

[8] Carlsson, E., Carlsson, G. and De Silva, V. 'An algebraic topological method for feature identification'. In: *International Journal of Computational Geometry & Applications* vol. 16, no. 04 (2006), pp. 291–314.

[9] Carlsson, G. 'Topology and data'. In: *Bulletin of the American Mathematical Society* vol. 46, no. 2 (2009), pp. 255–308.

[10] Carriere, M., Cuturi, M. and Oudot, S. 'Sliced wasserstein kernel for persistence diagrams'. In: *International Conference on Machine Learning*. PMLR. 2017, pp. 664–673.

[11] Cohen-Steiner, D., Edelsbrunner, H. and Harer, J. 'Stability of persistence diagrams'. In: *Discrete & computational geometry* vol. 37, no. 1 (2007), pp. 103–120.

[12] Cohen-Steiner, D. et al. 'Lipschitz functions have L p-stable persistence'. In: *Foundations of computational mathematics* vol. 10, no. 2 (2010), pp. 127–139.

[13] Cover, T. M. 'Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition'. In: *IEEE transactions on electronic computers*, no. 3 (1965), pp. 326–334.

[14] Edelsbrunner, H., Letscher, D. and Zomorodian, A. 'Topological persistence and simplification'. In: *Proceedings 41st Annual Symposium on Foundations of Computer Science* (2000), pp. 454–463.

[15] Fasshauer, G. E. *Meshfree approximation methods with MATLAB*. Vol. 6. World Scientific, 2007.

[16] Fasshauer, G. E. and McCourt, M. J. *Kernel-based approximation methods using Matlab*. Vol. 19. World Scientific Publishing Company, 2015.

[17] Fischl, B. 'FreeSurfer'. In: *Neuroimage* vol. 62, no. 2 (2012), pp. 774–781.

[18] Folstein, M. F., Folstein, S. E. and McHugh, P. R. '"Mini-mental state": a practical method for grading the cognitive state of patients for the clinician'. In: *Journal of psychiatric research* vol. 12, no. 3 (1975), pp. 189–198.

[19] Fomenko, A. T. *Visual geometry and topology*. Springer Science & Business Media, 2012.

[20] Guillemard, M. and Iske, A. 'Interactions between kernels, frames, and persistent homology'. In: *Recent Applications of Harmonic Analysis to Function Spaces, Differential Equations, and Data Science*. Springer, 2017, pp. 861–888.

[21] Hofmann, T., Schölkopf, B. and Smola, A. J. 'Kernel Methods in Machine Learning'. In: *The Annals of Statistics* vol. 36, no. 3 (2008), pp. 1171–1220.

[22] Kusano, G., Fukumizu, K. and Hiraoka, Y. 'Kernel method for persistence diagrams via kernel embedding and weight factor'. In: *The Journal of Machine Learning Research* vol. 18, no. 1 (2017), pp. 6947–6987.

[23] Kusano, G., Hiraoka, Y. and Fukumizu, K. 'Persistence weighted Gaussian kernel for topological data analysis'. In: *International Conference on Machine Learning*. PMLR. 2016, pp. 2004–2013.

[24] Kwitt, R. et al. 'Statistical topological data analysis-a kernel perspective'. In: *Advances in neural information processing systems*. 2015, pp. 3070–3078.

[25] Kääriäinen, M. *Relating the Rademacher and VC bounds*. Tech. rep. Citeseer, 2004.

[26] LaMontagne, P. J. et al. 'OASIS-3: longitudinal neuroimaging, clinical, and cognitive dataset for normal aging and Alzheimer disease'. In: *MedRxiv* (2019).

[27] Le, T. and Yamada, M. 'Persistence fisher kernel: A riemannian manifold kernel for persistence diagrams'. In: *arXiv preprint arXiv:1802.03569* (2018).

[28] Muandet, K. et al. 'Learning from distributions via support measure machines'. In: *arXiv preprint arXiv:1202.6504* (2012).

[29] Narcowich, F., Ward, J. and Wendland, H. 'Sobolev bounds on functions with scattered zeros, with applications to radial basis function surface fitting'. In: *Mathematics of Computation* vol. 74, no. 250 (2005), pp. 743–763.

[30] Pachauri, D. et al. 'Topology-Based Kernels With Application to Inference Problems in Alzheimer's Disease'. In: *IEEE Transactions on Medical Imaging* vol. 30, no. 10 (2011), pp. 1760–1770.

[31] Pedregosa, F. et al. 'Scikit-learn: Machine Learning in Python'. In: *Journal of Machine Learning Research* vol. 12 (2011), pp. 2825–2830.

[32] Rabin, J. et al. 'Wasserstein barycenter and its application to texture mixing'. In: *International Conference on Scale Space and Variational Methods in Computer Vision*. Springer. 2011, pp. 435–446.

[33] Rahimi, A., Recht, B. et al. 'Random Features for Large-Scale Kernel Machines.' In: *NIPS*. Vol. 3. 4. Citeseer. 2007, p. 5.

[34] Reininghaus, J. et al. 'A Stable Multi-Scale Kernel for Topological Machine Learning'. In: *stat* vol. 1050 (2014), p. 21.

[35] Reininghaus, J. et al. 'A stable multi-scale kernel for topological machine learning'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 4741–4748.

[36] Riesz, F. 'Sur le opérations fonctionnelles linéaires'. In: *C. R. Math. Acad. Sci. Paris* vol. 149, no. 1 (1909), pp. 974–977.

[37] Saul, N. and Tralie, C. *Scikit-TDA: Topological Data Analysis for Python*. 2019.

[38] Schaback, R. 'Error estimates and condition numbers for radial basis function interpolation'. In: *Advances in Computational Mathematics* vol. 3, no. 3 (1995), pp. 251–264.

[39] Schölkopf, B., Smola, A. J., Bach, F. et al. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[40] Schölkopf, B. et al. 'New support vector algorithms'. In: *Neural computation* vol. 12, no. 5 (2000), pp. 1207–1245.

[41] Shawe-Taylor, J., Cristianini, N. et al. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

[42] Smola, A. J. et al. *Advances in large margin classifiers*. MIT press, 2000.

[43] Sriperumbudur, B. K., Fukumizu, K. and Lanckriet, G. R. 'Universality, Characteristic Kernels and RKHS Embedding of Measures.' In: *Journal of Machine Learning Research* vol. 12, no. 7 (2011).

[44] Stewart, G. W. 'Gershgorin theory for the generalized eigenvalue problem Ax=λBx'. In: *Mathematics of Computation* (1975), pp. 600–606.

[45] Tralie, C., Saul, N. and Bar-On, R. 'Ripser.py: A Lean Persistent Homology Library for Python'. In: *The Journal of Open Source Software* vol. 3, no. 29 (Sept. 2018), p. 925.

[46] Villani, C. *Optimal transport: old and new*. Vol. 338. Springer Science & Business Media, 2008.

[47] Virtanen, P. et al. 'SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python'. In: *Nature Methods* vol. 17 (2020), pp. 261–272.