Persistent homology applications in data analysis *

Stefano De Marchi
University of Padova - Department of Medicine
email: stefano.demarchi@unipd.it

March 23, 2026

Abstract

Homology, and in particular *persistent homology*, is a tool in Topological Data Analysis (TDA) for extracting features of data, such as connected components, high-dimensional holes, and cavities. Simplicial complexes are used as a basis to construct the so-called Persistence Diagrams (PD). We introduce kernels defined on PD and their extension, which we call Variably Scaled Persistence Kernels. Examples of their use in SVM classification of popular data sets in data analysis are discussed. This paper extends the talks given in Reggio Calabria at the 2025 edition of the SMART Conference.

1 Introduction and notations

Topology is a branch of mathematics that studies the fundamental properties of shapes and spaces that persist under continuous transformations. Topological Data Analysis (TDA) [8, 19] is a discipline that has gained popularity in recent years and uses tools from algebraic and computational topology to infer the shape of data (see also [15, 8] through the persistent homology approach. Some fields of application include chemistry, medicine, biomedicine, and the neurosciences (see, e.g., [37, 10, 36, 26]). As an example, Molecular Topological fingerprints (MTFs) based on *Persistent homology* can be used for protein identification, classification, quantitative analysis of rigidity, and evolution of structural features during folding [21].

TDA primarily detects the *invariant features* of data, such as connected components, holes, and cavities. To this end, simplicial homology, which uses simplicial complexes, allows us to obtain these invariants via generalized triangulations that are filtered by particular maps, thereby providing information on persistent features through new structures such as the Persistent Diagrams (PD). The use of topological methods and analysis techniques to extract significant features and patterns from data is receiving increasing interest, as documented in the book by Carlsson and Vejdemo-Johansson [19] (see also [8, 15, 17]).

*Invited talk at the SMART2025 Conference, held in Reggio Calabria (Italy) from 28 September to 2 October 2025

Kernel methods are well-established tools for approximating functions and for data analysis, in particular in supervised machine learning and across various research and applied fields [33, 34, 35]. The flexibility provided by kernel-based schemes allows for the handling of different kinds of structured data, e.g., graphs and words, which are encoded in some dot product space where even complex patterns may be distinguished (cf. e.g., [5, 12, 24, 39]),

In what follows, our data will consist of *persistence diagrams*, which are outputs of *persistent homology*. We note that in recent years several kernels specifically designed to handle the peculiar structure of persistence diagrams have been proposed [9, 22, 25, 32]; therefore, the construction of suitable kernels is an active research direction in this context.

The interactions between kernel methods, frame analysis, and persistent homology in the framework of reproducing kernel Hilbert spaces have been explored in [18]. The authors showed that interactions among these novel methodologies create new opportunities for constructing numerical algorithms to analyze properties of data that have so far remained unexplored.

After an overview of the main ideas and tools of persistent homology in Section 2, we recall in Section 3 the main idea of positive-definite kernel approximation and, in particular, the *Variably Scaled Kernels (VSK)* [6]. Kernels for persistent diagrams are presented in Section 4 and then extended in Section 5 as *Variably Scaled Persistence Kernels (VSPKs)*, inspired by the VSK and employed as a feature augmentation strategy in [7, 27] for kernel-based learning. Indeed, we aim to bridge classical VSK and kernels defined in the context of persistent homology. After defining VSPKs, we design scaling functions and test the performance of the resulting kernels across two datasets. The results in Section 6 show that the variably scaled setting may yield better classification performance and efficiency than the standard setting, warranting further investigation and application in persistence diagram-based settings. Moreover, in Section 7, we discuss the use of persistent homology in the k-NN (k-Nearest Neighbors) classification problem. We give the conclusions and some future developments in Section 8.

2 Persistent homology

We begin by introducing basic concepts of persistent homology. This procedure associates a sequence of abelian groups (or modules) with a given mathematical object. For more details, especially concerning the algebraic aspects of the construction, we invite the readers to refer to these fundamental papers [15, 18].

2.1 Basics on persistent homology

Data, manifolds and simplicial complexes. Consider the set of data $X = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\} \subset \Omega$ as a set of vertices sampled from some manifold M immersed in a topological space, of which we want to find intrinsic homological properties such as the number of connected components, holes, and cavities present in the data set.

Definition 2.1 *Let $v_0, v_1, \dots, v_n$ be linearly independent points in $\mathbb{R}^m$, with $m \geq n$. We*

*define an n-**simplex** σ as the convex combination of the points $v_0, \ldots, v_n$:*

$$\sigma = [v_0, v_1, \ldots, v_n] = \left\{ \sum_{i=0}^{n} t_i v_i \;\middle|\; t_i \geq 0,\ \sum_{i=0}^{n} t_i = 1 \right\}.$$

*$v_0, v_1, \ldots, v_n$ are called **vertices** of the simplex, and each non-empty subset of vertices determines a **face** of σ.*

We say that a simplex has **dimension** p if it has $p+1$ vertices (i.e its cardinality). If needed, we denote by Σ_p the set of p-simplices. In the Figure 1 (taken from [40]), we show examples of simplices of dimensions 0, 1, 2, and 3, which are points, segments, triangles, and tetrahedra, respectively.

We can now define a structure that allows us to join these simplices, even when they have different dimensions.

Definition 2.2 *A **simplicial complex** K is a finite set of simplices of different dimensions such that:*

- *Each face of a simplex σ in K must belong to K*
- *The non-empty intersection of two simplices σ_1 and σ_2 is a face of both σ_1 and σ_2*

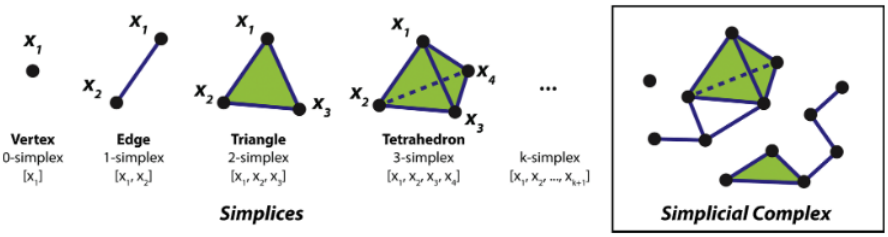


Figure 1: Left: simplices up to dimension 3. Right: a simplicial complex of dimension 3.

Simplices in K can be enumerated and denoted by σ_i^k, i.e., the i-th simplex of K of dimension k.

To study geometrical properties, we consider the manifold built on X, which we treat as an approximation of M. More precisely, given $\varepsilon > 0$, we study

$$M^{X,\varepsilon} = \bigcup_{i=1}^{n} B(\boldsymbol{x}_i, \varepsilon),$$

where $B(\boldsymbol{x}_i, \varepsilon)$ is the ball centered in $\boldsymbol{x}_i$ of radius ε. Then, we associate to $M^{X,\varepsilon}$ an abstract *simplicial complex*, which is a family of sets closed with respect to the *subset operator*.

Simplicial complexes can be defined by different construction rules that determine how higher-dimensional structures are built from lower-dimensional ones. In Figure 1, the complex of dimension 3 on the right is a *Vietoris-Rips complex*.

Specifically, we associate to $M^{X,\varepsilon}$ a *Vietoris-Rips simplicial complex*, say $K^{X,\varepsilon}$, where two distinct vertices $\boldsymbol{x}_i, \boldsymbol{x}_j$ are connected by an edge if and only if $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 \leq \varepsilon$, and we consider r-dimensional elements that are determined by $r+1$ connected $(r-1)$-dimensional faces as long as $r \leq v$. In Figure 2 (cf. [14]), we show a two-dimensional example ($v = 2$) where different numbers of edges or triangles are generated depending on the chosen ε.

Simplicial homology groups. On the Abelian group $\mathbb{G} = (\mathbb{Z}, +)$ we can consider linear combinations of simplices with coefficients in $\mathbb{G}$, getting *chains of simplices*. Integer combinations of r-dimensional simplices in $K^{X,\varepsilon}$ form the *r-chains* algebraic group $C_r^{X,\varepsilon}$.

- σ and $-\sigma$ are the simpliest chains.
- Let $\sigma_i, i = 1,2,3$ be 1-simplices (i.e. line segments), then $2\sigma_1 - \sigma_2 + 2\sigma_3$ is a 1-chain, that is an element of $C_1^{X,\varepsilon}$. The coefficients indicate how many times each simplex is included with its orientation (positive/negative).

Moreover, letting $[\boldsymbol{x}_{i_0}, \ldots, \boldsymbol{x}_{i_r}]$ be the face constructed upon distinct vertices $\boldsymbol{x}_{i_0}, \ldots, \boldsymbol{x}_{i_r}$ in X, $i_0, \ldots, i_r \in \{1, \ldots, n\}$ (which is an r-dimensional simplex), we consider the *boundary operator* $\partial_r : C_r^{X,\varepsilon} \longrightarrow C_{r-1}^{X,\varepsilon}$ as

$$\partial_r[\boldsymbol{x}_{i_0}, \ldots, \boldsymbol{x}_{i_r}] = \sum_{j=0}^{r} (-1)^r [\boldsymbol{x}_{i_0}, \ldots, \boldsymbol{x}_{i_{j-1}}, \boldsymbol{x}_{i_{j+1}}, \ldots, \boldsymbol{x}_{i_r}].$$

In practice, the boundary of a chain is the linear combination of the boundary of the simplices in the r-chain, which is a chain of order $r-1$. The operator ∂_r is linear and $\partial_r^2 = 0$, since the boundary of a simplex has no boundary. Moreover, if $\partial_r(ac) = 0$ with $a \neq 0$, then $\partial_r = 0$.

Example 2.1 *Consider the path from the point* $\boldsymbol{x}_1$ *to* $\boldsymbol{x}_4$. *Take the* 1*-simplices* $\sigma_1 = [\boldsymbol{x}_1, \boldsymbol{x}_2]$, $\sigma_2 = [\boldsymbol{x}_2, \boldsymbol{x}_3]$, $\sigma_3 = [\boldsymbol{x}_3, \boldsymbol{x}_4]$ *and the chain* $c = \sigma_1 + \sigma_2 + \sigma_3$. *Hence,*

$$\begin{aligned} \partial_1 c &= \partial_1(\sigma_1 + \sigma_2 + \sigma_3) = \partial_1(\sigma_1) + \partial_1(\sigma_2) + \partial_1(\sigma_3), \\ &= \partial_1([\boldsymbol{x}_1, \boldsymbol{x}_2]) + \partial_1([\boldsymbol{x}_2, \boldsymbol{x}_3]) + \partial_1([\boldsymbol{x}_3, \boldsymbol{x}_4]) \\ &= (\boldsymbol{x}_2 - \boldsymbol{x}_1) + (\boldsymbol{x}_3 - \boldsymbol{x}_2)(\boldsymbol{x}_4 - \boldsymbol{x}_3) = \boldsymbol{x}_4 - \boldsymbol{x}_3. \end{aligned}$$

Let c an r-chain. It is called an *r-cycle* if its boundary is zero, i.e. $\partial c = 0$. As an example, consider a closed polygonal curve. Boundaries are cycles (not the opposite!), so chains form a chain complex, whose homology groups (cycles modulo boundaries) are called *simplicial homology groups*.

More specifically, *r-cycles* and *r-boundaries* form Abelian groups that are denoted as $Z_r^{X,\varepsilon} = \ker\partial_r$ and $B_r^{X,\varepsilon} = \mathrm{im}\partial_{r+1}$, respectively. Furthermore, it can be proved that the rank of the *r-homology* group

$$H_r^{X,\varepsilon} = Z_r^{X,\varepsilon} / B_r^{X,\varepsilon}$$

encapsulates the concept of *r-dimensional holes* in the set of the complexes $K^{X,\varepsilon}$, thus relating M and X, for characterizing their geometrical features.

Filtration and Persistent Homology. Unfortunately, finding an optimal $\varepsilon^\star$ that represents the intrinsic geometric properties of M, is a tough and unstable process. This is evident in the example depicted in Figure 2, where different values of ε determine different simplicial structures.

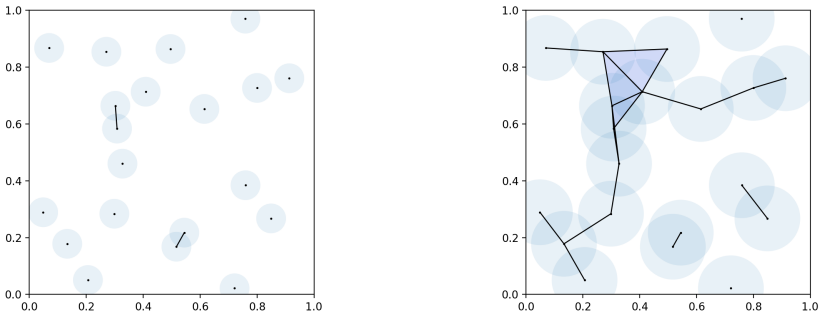


Figure 2: Different values of ε determine different simplicial structure

To solve this issue, one can drop the idea of selecting a good parameter and analyse the whole *filtration* $\{M^{X,\varepsilon} \mid \varepsilon > 0\}$.

Definition 2.3 *Given the simplicial complex K,a family $(K_t)_{t\in T}$ of subcomplexes, where T is a totally ordered set is a* ***filtration*** *if:*

- $\forall t_1, t_2 \in T$ *with* $t_1 < t_2$ *we get* $K_{t_1} \subset K_{t_2}$
- $\bigcup_{t\in T} K_t = K$

The simplicial complex K obtained with such a sequence of subcomplexes is termed *filtered simplicial complex*. As an example, in Figure 3 (taken from [20, Fig. 9 and 10]), we show two filtrations of the complex consisting of the union of two touching triangles. In Figure 4 (taken from [29, Fig. 5]), the evolving sequence of complexes is shown.

Given a filtration $\emptyset = K_0 \subseteq K_1 \subseteq K_2 \subseteq \cdots \subseteq K_l = K$ of a simplicial complex K, we have the following inclusion maps:

$$\imath_{i,j} : K_i \to K_j$$

for each $i, j = 1, \ldots, l$ with $i \leq j$.

Thanks to homology properties, these maps induce a linear map on the corresponding homology groups:

$$f_{i,j} : H_k(K_i) \to H_k(K_j)$$

for each $i, j = 1, \ldots, l$ with $i \leq j$.

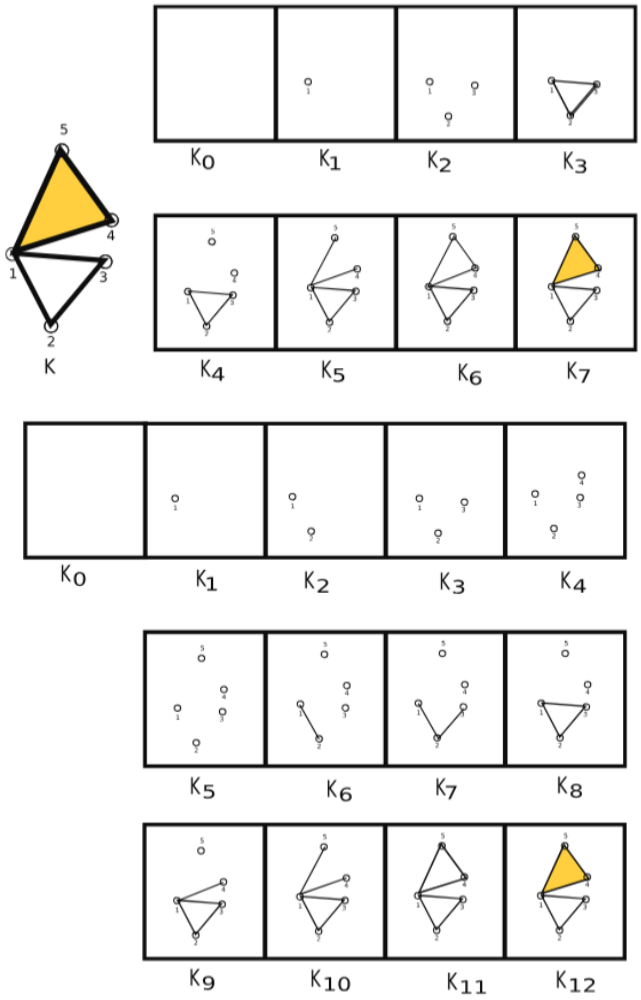


Figure 3: Two filtrations of the complex made up of two touching triangles.

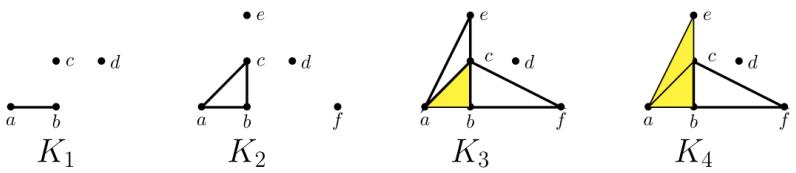


Figure 4: Example of a filtered complex.

Definition 2.4 *Let $\emptyset = K_0 \subseteq K_1 \subseteq \cdots \subseteq K_l = K$ be a filtered simplicial complex. Then the* ***p-th persistent homology*** *of K is the pair*

$$(\{H_p(K_i)\}_{1\leq i\leq l}, \{f_{i,j}\}_{1\leq i\leq j\leq l})$$

where the maps $f_{i,j} : H_p(K_i) \to H_p(K_j)$ are the maps induced by the corresponding inclusion maps.

For each i, j, we observe that the p-th persistent homology represents the set of p-dimensional homology classes that arose in K_i or earlier and are still present in K_j, meaning they have not yet closed or transformed into edges.

Example 2.2 *In this example, we have a filtered simplicial complex K consisting of the following four points: $A = (0,0); B = (1,0); C = (0.5, 0.8); D = (1.9, 0.6)$.*
The filtration is given by:

$$\emptyset = K_0 \subseteq K_1 \subseteq K_2 \subseteq K_3 = K$$

At step 1 we observe that a cycle A-B-C is created, thus obtaining a hole, an element of the 1-dimensional homology class.
At step 2, the segment B-D is added, but the cycle remains, thereby maintaining the cavity.
Finally, at step 3, we obtain a 2-simplex (the filled triangle A-B-C) which kills the cycle.

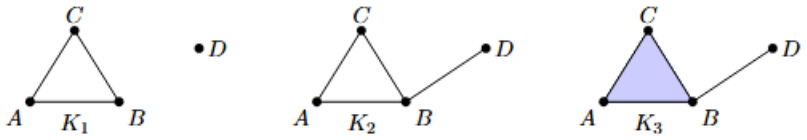


Figure 5: Birth (K_1), persistence (K_2), death (K_3) of a 1-dimensional cycle (A-B-C).

All these characteristics of the birth, death, and duration of existence of the various p-dimensional cavities can be represented by diagrams that make the homological properties of the spaces easier to understand. Each class can therefore be described as a pair of indices $[b, d)$, where b represents the birth index (*birth*) of the class, i.e., it is born in K_b, while d represents the death index (*death*), i.e., it dies in K_d. Note that $b < d$ always holds, and d could also be $+\infty$ if it never dies during filtering.

The difference $d - b$ is called the **persistence** of the homology class.

Persistent barcodes and diagrams One way to represent class persistence is through barcodes:

Definition 2.5 *The* ***barcode*** *is a graph consisting of a set of intervals in $\mathbb{R} \cup \{+\infty\}$ where each interval represents the life of a p-dimensional homology class. Each interval indicates the birth and death of the class.*

Another useful tool is the persistence diagram:

Definition 2.6 *The **persistence diagram** D_p is a plane representation of a set of points (b,d), where each point represents an element of the homology classes with the x coordinate representing birth and the y coordinate representing death.*

Remark 2.1 *In the persistence diagram, points on the diagonal (i.e., with $b = d$) correspond to classes with zero persistence. The points located away from the diagonal represent significant classes, i.e., classes with long persistence.*

More formally, each element of the persistent homology groups obtained by considering the whole filtration can be represented by a birth-death pair $(b,d) \in \mathbb{R}^2_+$, $b = \varepsilon_h$, $d = \varepsilon_k$ for some $h \in \{1,\dots,m\}$, $k \in \{1,\dots,m\} \cup \{\infty\}$, $h < k$. We say that $d-b$ is the *persistence* of (b,d), and that a birth-death pair is r-dimensional if it is related to r-dimensional homology groups.

Definition 2.7 *Letting $\boldsymbol{\varepsilon} = (\varepsilon_1,\dots,\varepsilon_u)$ a persistence diagram $D_r(X,\boldsymbol{\varepsilon})$ related to the filtration $K^{X,\varepsilon_1} \subseteq \cdots \subseteq K^{X,\varepsilon_m}$ is the multiset*

$$D_r(X,\boldsymbol{\varepsilon}) = \{(b,d) \,|\, (b,d) \in P_r(X,\boldsymbol{\varepsilon})\} \cup B, \quad r \geq 0, \tag{1}$$

subset of $\mathbb{R}^2$, where $P_r(X,\boldsymbol{\varepsilon})$ denotes the set of r-dimensional birth-death pairs obtained with the filtration and $B = \{(x,x) \,|\, x \geq 0\}$.

We remark that $D_r(X,\boldsymbol{\varepsilon})$ is a multiset, since a couple (b,d) might appear more than once, i.e., might have multiplicity greater than one. Furthermore, the bisector $B = \{(x,x), x \geq 0\}$ is composed of an infinite number of elements of infinite multiplicity, which are concerned with noise in the data. It is added to achieve *uniformity* across different persistence diagrams and to facilitate the formulation of proper metrics, as we present below. Each point in $(b,d) \in D_r(X,\boldsymbol{\varepsilon})$ is called *generator* and the difference $d-b$ its *persistence*.

Example 2.3 *Consider 500 points of a torus as a dataset in Figure 6. The corresponding barcode and the persistence diagram are displayed in the Figures 7.*

Example 2.4 *As another example, we consider 500+500 points of two intersecting tori in Figure 8, and we show the barcode and the associated persistence diagrams in the Figure 9.*

Persistence diagrams exhibit stability under perturbations of the underlying dataset, as discussed in [11]. To clarify this point, we recall two useful metrics for comparing datasets and persistence diagrams.

Definition 2.8 *Let $X,Y \subset \Omega$ be two non-empty datasets. The Hausdorff distance between X,Y is*

$$d_H(X,Y) = \max\left\{ \sup_{x\in X} \inf_{y\in Y} \|x-y\|_\infty, \sup_{y\in Y} \inf_{x\in X} \|x-y\|_\infty \right\}.$$

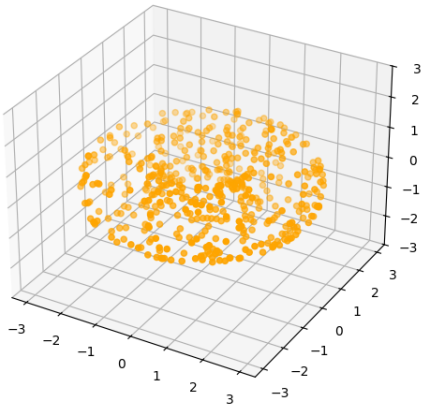


Figure 6: Dataset: points on a torus

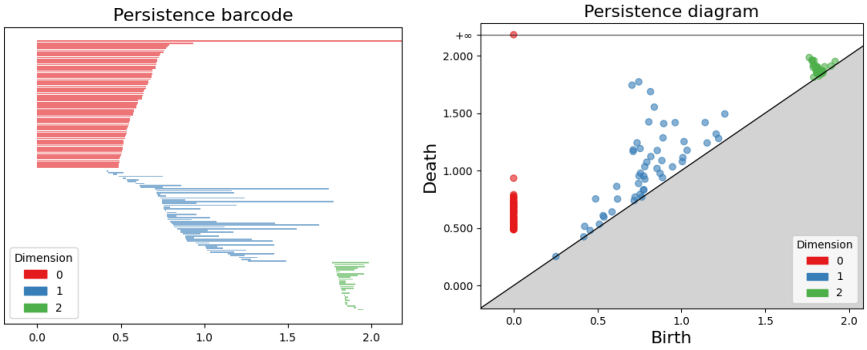


Figure 7: Barcode and the Persistence Diagram of the torus

Definition 2.9 *Let $D_r(X,\boldsymbol{\varepsilon})$ and $D_r(Y,\boldsymbol{\varepsilon})$ be persistence diagrams for some $r \geq 0$ and filtration radii vector $\boldsymbol{\varepsilon}$. The p-Wasserstein distance*

$$d_{W,p}(D_r(X,\boldsymbol{\varepsilon}),D_r(Y,\boldsymbol{\varepsilon})) = \left(\inf_{\gamma\in\Gamma}\sum_{x\in D_r(X,\boldsymbol{\varepsilon})}\|x-\gamma(x)\|_\infty^p\right)^{\frac{1}{p}},$$

where $\Gamma = \{\gamma : D_r(X,\boldsymbol{\varepsilon}) \longrightarrow D_r(Y,\boldsymbol{\varepsilon}) \mid \gamma \text{ is a bijection}\}$. In particular, letting $p \to \infty$, we obtain the bottleneck distance

$$d_{W,\infty}(D_r(X,\boldsymbol{\varepsilon}),D_r(Y,\boldsymbol{\varepsilon})) = d_B(D_r(X,\boldsymbol{\varepsilon}),D_r(Y,\boldsymbol{\varepsilon})) = \inf_{\gamma\in\Gamma}\sup_{x\in D_r(X,\boldsymbol{\varepsilon})}\|x-\gamma(x)\|_\infty.$$

We have the stability result [22]

$$d_B(D_r(X,\boldsymbol{\varepsilon}),D_r(Y,\boldsymbol{\varepsilon})) \leq d_H(X,Y),$$

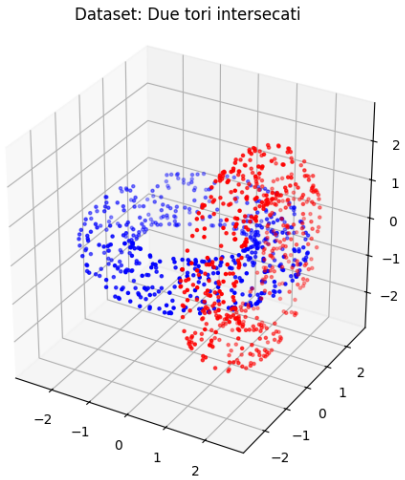


Figure 8: Dataset: points on a torus

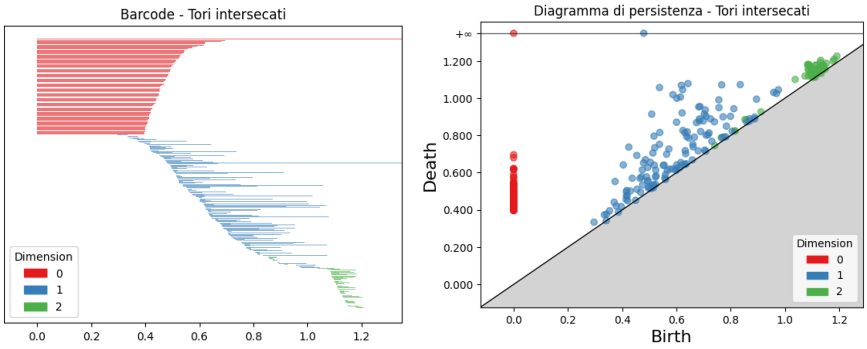


Figure 9: Barcode and the Persistence Diagram of the intersecting tori

i.e., the bottleneck distance between persistence diagrams is controlled, provided that the underlying datasets are close in the Hausdorff metric.

3 Variably scaled kernels

Let $\Phi : \Omega \times \Omega \longrightarrow \mathbb{R}$ be a kernel. Given a set of data $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subset \Omega$, the $n \times n$ matrix K with elements $K_{ij} = \Phi(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $i, j = 1, \ldots, n$, is the *Gram matrix* of the kernel Φ with respect to X. If Φ is positive definite (strictly positive definite) on $\Omega \times \Omega$, i.e., K is positive semi-definite (definite) for all possible datasets in Ω. It is possible to decompose the kernel according to *Mercer's theorem* [28], and to interpret such decomposition as an inner product in a *Reproducing Kernel Hilbert Space* (RKHS) $\mathscr{F}$.

Indeed, there exists a (non-unique) *feature map* $\Phi : \Omega \longrightarrow \mathscr{F}$ such that

$$\Phi(\boldsymbol{x},\boldsymbol{y}) = \langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{y}) \rangle_{\mathscr{F}} \qquad \boldsymbol{x},\boldsymbol{y} \in \Omega,$$

being $\langle \cdot,\cdot \rangle_{\mathscr{F}}$ the bilinear form related to the RKHS (e.g. $\Phi(\boldsymbol{x}) = \Phi(\cdot,\boldsymbol{x})$). Moreover, the kernel Φ induces a distance d_Φ on Ω

$$d_\Phi(\boldsymbol{x},\boldsymbol{y}) = \Phi(\boldsymbol{x},\boldsymbol{x}) + \Phi(\boldsymbol{y},\boldsymbol{y}) - 2\Phi(\boldsymbol{x},\boldsymbol{y}). \tag{2}$$

As already mentioned, Variably Scaled Kernels (VSK) were introduced in [6] in the context of kernel-based approximation to overcome instability issues. They have subsequently been extended to a more general setting in [7].

Definition 3.1 *Let $\Lambda \subseteq \mathbb{R}^\nu$, $\nu > 0 \in \mathbb{N}$ and let $\Phi : \tilde{\Omega} \times \tilde{\Omega} \longrightarrow \mathbb{R}$ be a continuous (strictly) positive definite kernel, where $\tilde{\Omega} = \Omega \times \Lambda \subseteq \mathbb{R}^{\nu+\nu}$. Given a scaling function $\psi : \Omega \longrightarrow \Lambda$, and by defining $\Psi(\boldsymbol{x}) = (\boldsymbol{x}, \psi(\boldsymbol{x}))$, a VSK $\Phi^\Psi : \Omega \times \Omega \longrightarrow \mathbb{R}$ is defined as*

$$\Phi^\Psi(\boldsymbol{x},\boldsymbol{y}) = \Phi((\boldsymbol{x},\psi(\boldsymbol{x})),(\boldsymbol{y},\psi(\boldsymbol{y})))$$

for $\boldsymbol{x},\boldsymbol{y} \in \Omega$.

The function Ψ can be interpreted as a *feature augmentation* map, which adds ν coordinates (features) to the original sample. In this view, the VSK setting has been analysed in [7] as a *stacking* technique that enhances the predictive performance of classical kernel-based classifiers, such as *Support Vector Machines (SVMs).*

We recall that a binary SVM classifier considers a set of points $\boldsymbol{x} = (x_1,\ldots,x_d)^\intercal \in \Omega$, an associated binary set of labels $T = \{-1,+1\}$ and the decision function

$$s(\boldsymbol{x}) = \operatorname{sign}(h(\boldsymbol{x})) = \operatorname{sign}(\langle \Phi(\boldsymbol{x}), \boldsymbol{w} \rangle_{\mathscr{F}} + b),$$

where

$$\boldsymbol{w} = \sum_{i=1}^{n} \alpha_i y_i \Phi(\boldsymbol{x}_i) \in \mathscr{F}.$$

The coefficients $\boldsymbol{\alpha} = (\alpha_1,\ldots,\alpha_n) \in \mathbb{R}^n$ can be obtained as the solution of the following *soft margin* problem (cf. e.g. [16, §18, p. 346–347])

$$\begin{cases} \min_{\boldsymbol{\alpha}\in\mathbb{R}^n} \dfrac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \Phi(\boldsymbol{x}_i,\boldsymbol{x}_j) - \sum_{i=1}^n \alpha_i, \\ \text{s.t. } \sum_{i=1}^n \alpha_i y_i = 0, \\ 0 \le \alpha_i \le \zeta, \quad i = 1,\ldots,n, \end{cases}$$

where $[0,\zeta]^n$ is the *bounding box*, with $\zeta \in [0,+\infty)$. The problem is also called the *supervised learning* problem in the machine learning literature. Moreover, a binary SVM classifier can be extended to the multiclass setting by considering a *one-vs-rest* approach.

Remark 3.1 *The strength of kernel-based classifiers, such as SVMs, largely depends on the flexibility of kernel construction. Indeed, any structured data may potentially be translated into a particular Hilbert space (e.g., the classical $\mathbb{R}^n$) by an appropriately defined feature map. This is the case of persistence diagrams, which are valuable tools in topological data analysis, characterized by a peculiar structure and useful for exploring the shape of data.*

4 Kernels for persistence diagrams

To better measure similarities between persistence diagrams, various positive-definite kernels for persistence diagrams have been introduced in recent literature. Besides, we will sometimes use the notations $D_1 = D_r(X, \boldsymbol{\varepsilon})$ and $D_2 = D_r(Y, \boldsymbol{\varepsilon})$ for simplicity.

- The **Persistence Scale Space (PSS) kernel** [32].

  The key idea is to compute the feature map as the solution of the *heat equation*. We consider $\Omega_{ad} = \{\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2 : x_2 \geqslant x_1\}$ and denote with $\delta_{\mathbf{x}}$ the Dirac delta centered at $\mathbf{x}$. For a given $D \in \mathfrak{D}_r$, we consider the solution $u : \Omega_{ad} \times \mathbb{R}_{\geqslant 0} \to \mathbb{R}$, $(\mathbf{x}, t) \mapsto u(\mathbf{x}, t)$ of the following PDE:

$$\begin{aligned} \Delta_{\mathbf{x}} u = \partial_t u \quad & \text{in } \Omega_{ad} \times \mathbb{R}_{\geqslant 0} \\ u = 0 \quad & \text{on } \partial\Omega_{ad} \times \mathbb{R}_{\geqslant 0} \\ u = \sum_{\mathbf{y} \in D} \delta_{\mathbf{y}} \quad & \text{on } \Omega_{ad} \times 0. \end{aligned}$$

  The feature map $\Phi_\sigma : \mathfrak{D}_r \to \mathrm{L}^2(\Omega_{ad})$ at scale $\sigma > 0$ at the persistence diagram D is defined as $\Phi_\sigma(D) = u\big|_{t=\sigma}$. This map yields the **Persistence Scale-Space Kernel** (**PSS kernel**) K_{PSS} on $\mathfrak{D}_r$ as:

$$\Phi_\sigma(D_1, D_2) = \langle \Phi_\sigma(D_1), \Phi_\sigma(D_2) \rangle_{\mathrm{L}^2(\Omega_{ad})},$$

  or explicitly as

$$\Phi_\sigma(D_1, D_2) = \frac{1}{8\pi\sigma} \sum_{\substack{\boldsymbol{y} \in D_1 \\ \boldsymbol{z} \in D_2}} \exp\left(-\frac{||\boldsymbol{y} - \boldsymbol{z}||^2}{8\sigma} \right) - \exp\left(-\frac{||\boldsymbol{y} - \overline{\boldsymbol{z}}||^2}{8\sigma} \right), \quad (3)$$

  where $\overline{\boldsymbol{z}} = (d, b)$ if $\boldsymbol{z} = (b, d)$.

  The kernel Φ_σ is 1-Wasserstein stable, i.e.:

$$\|\Phi_\sigma(D_1) - \Phi_\sigma(D_2)\|_{L_2(\Omega)} \leq \frac{1}{2\sqrt{\pi}\sigma} d_{W,1}(D_1, D_2),$$

- The **Persistence Weighted Gaussian kernel (PWG)** [22]

  Here, the idea is to replace each persistence diagram with a discrete measure. Starting with a strictly positive definite kernel, as the Gaussian one $\Phi_G(\mathbf{x}, \mathbf{y}) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}}$, $\sigma > 0$ we denote the corresponding Reproducing Kernel Hilbert Space H_{Φ_G}.

  If $\Omega \subset \mathbb{R}^d$, we denote with $M_b(\Omega)$ the space of finite signed Radon measures and

$$E_{\Phi_G} : M_b(\Omega) \to H_{\Phi_G}, \mu \mapsto \int_\Omega \Phi_G(\cdot, \mathbf{x}) d\mu(\mathbf{x}).$$

Then, the PWG kernel is defined as

$$\Phi_G(D_1,D_2;\Phi,\omega) = \exp\left(-\frac{1}{2\tau^2}\|E_\Phi(\mu^\omega_{D_1}) - E_\Phi(\mu^\omega_{D_2})\|^2_{\mathscr{F}}\right) \quad \tau > 0, \quad (4)$$

which is built upon a *standard* Gaussian kernel Φ and a weight function ω, where

$$E_\Phi(\mu^\omega_{D_1}) = \sum_{\boldsymbol{x}\in D_1} \omega(\boldsymbol{x})\,\Phi(\cdot,\boldsymbol{x}).$$

It is both 1-Wasserstein and bottleneck stable if we choose as a weight function

$$\omega_{\text{arc}}(\boldsymbol{x}) = \arctan(C(d-b)^\delta) \quad \boldsymbol{x} = (b,d), C > 0, \delta \in \mathbb{Z}_{>0},$$

where $d-b$ is the persistence. Indeed, there exist $\delta \in \mathbb{Z}_{>0}$ and $L > 0$ such that

$$\|E_\Phi(\mu^{\omega_{\text{arc}}}_{D_1}) - E_\Phi(\mu^{\omega_{\text{arc}}}_{D_2})\| \leq L d_B(D_1,D_2). \quad (5)$$

- The **Sliced Wasserstein (SW) kernel** [9]

  We consider μ and ν two nonnegative measures, not necessarily normalized, on $\mathbb{R}$ such that $\mu(\mathbb{R}) = r = |\mu|$ and $\nu(\mathbb{R}) = r = |\nu|$, we recall that the 1-Wasserstein distance for nonnegative measures

$$W(\mu,\nu) := \inf_{P\in\Pi(\mu,\nu)} \int\int_{\mathbb{R}\times\mathbb{R}} |x-y|\,dP(x,y)$$

where $\Pi(\mu,\nu)$ is the set of measures on $\mathbb{R}^2$ with marginals μ and ν.

Definition 4.1 *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 $D_1, D_2 \in \mathfrak{D}_r$ and let $\mu^\theta_{D_1} := \sum_{\boldsymbol{x}\in D} \delta_{\pi_\theta(\boldsymbol{x})}$ and $\mu^\theta_{D_1\Delta} := \sum_{\boldsymbol{x}\in D} \delta_{\pi_\theta\circ\pi_\Delta(\boldsymbol{x})}$ and similarly for $\mu^\theta_{D_2}$ and $\mu^\theta_{D_2\Delta}$ where π_Δ is the orthogonal projection onto the diagonal. Then, the **Sliced Wasserstein distance** is*

$$SW(D_1,D_2) = \frac{1}{2\pi}\int_{\mathbb{S}_1} W(\mu^\theta_{D_1} + \mu^\theta_{D_2\Delta}, \mu^\theta_{D_2} + \mu^\theta_{D_1\Delta})\,d\theta.$$

with S_1 the usual $1d$ sphere, i.e. the cicle.

Hence, the SW kernel is defined as follows

$$\Phi_{SW}(D_1,D_2) := \exp\left(-\frac{SW(D_1,D_2)}{2\sigma^2}\right). \quad (6)$$

Moreover, the Sliced Wasserstein distance, $SW(\cdot,\cdot)$, is equivalent to the 1-Wasserstein distance thanks to the following inequalities

$$\frac{1}{2M} d_{W,1}(D_1,D_2) \leq d_{SW}(D_1,D_2) \leq 2\sqrt{2}\, d_{W,1}(D_1,D_2),$$

for some positive constant M.

Remark 4.1 *We notice that the PWG and SW kernels are in the form*

$$\Phi(D_1, D_2) = \exp(-\beta d(D_1, D_2))$$

for some $\beta > 0$, where $d(\cdot,\cdot)$ is the distance induced by the underlying metric in the case of the SW, while for the PWG kernel the distance is induced by the kernel (see the definition of d_Φ in (2)*).*

Other persistence kernels have been studied in the recent literature. It is worth mentioning the *Persistence Fisher Kernel*, which is a Riemannian manifold kernel for persistence diagrams [25], and the *Persistence Image Kernel* , for homological image analysis [1]. These two kernels have been used, for instance, in SVM classification applications in [3].

5 Variably Scaled Persistence Kernels

The purpose is to reinterpret the idea underlying VSKs in the context of persistent homology. The main difference between standard kernels and kernels on persistence diagrams is the structure of the input data. Since persistence diagrams consist of collections of topological features, (birth, death) couples, that is, objects in $\mathbb{R}^2_{\geq B} = \{\boldsymbol{x} = (b,d) \in \mathbb{R}^2_+ \,|\, d \geq b\}$. Introducing a scaling function whose output lies outside $\mathbb{R}^2_{\geq B}$ would be meaningless. Hence, we make the following.

Definition 5.1 *Let $\mathfrak{D}_r(\boldsymbol{\varepsilon}) = \{D_r(X,\boldsymbol{\varepsilon}) \,|\, X \subset \Omega\}$ and $\Phi : \mathfrak{D}_r(\boldsymbol{\varepsilon}) \times \mathfrak{D}_r(\boldsymbol{\varepsilon}) \longrightarrow \mathbb{R}$ be a kernel for persistence diagrams. Consider $\Psi : \mathfrak{D}_r(\boldsymbol{\varepsilon}) \to \mathfrak{D}_r(\boldsymbol{\varepsilon})$. A variably scaled persistence kernel Φ^Ψ on $\mathfrak{D}_r(\boldsymbol{\varepsilon}) \times \mathfrak{D}_r(\boldsymbol{\varepsilon})$ is*

$$\Phi^\Psi(D_r(X,\boldsymbol{\varepsilon}), D_r(Y,\boldsymbol{\varepsilon})) := \Phi(\Psi(D_r(X,\boldsymbol{\varepsilon})), \Psi(D_r(Y,\boldsymbol{\varepsilon})))$$

for any couple of persistence diagrams $D_r(X,\boldsymbol{\varepsilon}), D_r(Y,\boldsymbol{\varepsilon}) \in \mathfrak{D}_r(\boldsymbol{\varepsilon})$.

As previously discussed, a proper function Ψ needs to be designed. We will construct it upon an auxiliary function $\psi : \mathfrak{D}_r(\boldsymbol{\varepsilon}) \longrightarrow \mathbb{R}^2_{\geq B}$, whose design is deepened below in Section 5.3. Instead, we present two possible functions Ψ, that we denote as Ψ_a and Ψ_ρ.

5.1 The feature augmentation map: Ψ_a

We define

$$\Psi_a(D_r(X,\boldsymbol{\varepsilon})) = D_r(X,\boldsymbol{\varepsilon}) \cup \psi(D_r(X,\boldsymbol{\varepsilon})).$$

The map Ψ_a plays the role of a *feature augmenting* map, since an additional generator is added in the persistence diagram. The following stability results hold for VSPKs with the map Ψ_a in terms of the classical setting, whose proofs can be found in [14].

Proposition 5.1 *Let Φ_σ be the PSS kernel defined in* (3) *and let D_1, D_2 be r-dimensional persistence diagrams for some $r \geq 0$. We have*

$$\|\Phi_\sigma^{\Psi_a}(D_1) - \Phi_\sigma^{\Psi_a}(D_2)\|_{L_2(\Omega)} \leq \frac{1}{2\sqrt{\pi\sigma}} \left(d_{W,1}(D_1, D_2) + \|\psi(D_1) - \psi(D_2)\|_\infty \right).$$

Proposition 5.2 *Let $\Phi_G(\cdot,\cdot,\Phi,\omega_{\text{arc}})$ be the PWG kernel defined in* (4)*, where $\Phi(\boldsymbol{x},\boldsymbol{y}) = \exp(-\|\boldsymbol{x}-\boldsymbol{y}\|^2/(2\eta^2))$, and let D_1, D_2 be r-dimensional persistence diagrams for some $r \geq 0$. By denoting as $E_\Phi^{\Psi_a}(\mu_{D_1}^{\omega_{arc}}) = E_\Phi(\mu_{D_1 \cup \psi(D_1)}^{\omega_{arc}})$, we have*

$$\|E_\Phi^{\Psi_a}(\mu_{D_1}^{\omega_{arc}}) - E_\Phi^{\Psi_a}(\mu_{D_2}^{\omega_{arc}})\| \leq L d_B(D_1,D_2) + L^\psi(D_1,D_2)$$

with

$$\begin{aligned} L^\psi(D_1,D_2) &= 2\omega_{arc}(\psi(D_1))\left(1-\exp\left(-\frac{\|\psi(D_1)-\psi(D_2)\|^2}{2\eta^2}\right)\right) + \\ &+ |\omega_{arc}(\psi(D_1)) - \omega_{arc}(\psi(D_2))| \end{aligned}$$

and L as in (5)*.*

Proposition 5.3 *Let Φ_{SW} be the SW kernel defined in* (6) *and let D_1, D_2 be r-dimensional persistence diagrams for some $r \geq 0$. We have*

$$d_{SW}^{\Psi_a}(D_1,D_2) \leq d_{SW}(D_1,D_2) + d_{SW}(\psi(D_1),\psi(D_2)).$$

Corollary 5.1 *In the hypotheses of Proposition 5.3, we have*

$$d_{SW}^{\Psi_a}(D_1,D_2) \leq 2\sqrt{2}(d_{W,1}(D_1,D_2) + \|\psi(D_1)-\psi(D_2)\|_\infty).$$

Indeed

$$\begin{aligned} d_{SW}^{\Psi_a}(D_1,D_2) &\leq d_{SW}(D_1,D_2) + d_{SW}(\psi(D_1),\psi(D_2)) \\ &\leq 2\sqrt{2}(d_{W,1}(D_1,D_2) + d_{W,1}(\psi(D_1),\psi(D_2))) \\ &\leq 2\sqrt{2}(d_{W,1}(D_1,D_2) + \|\psi(D_1)-\psi(D_2)\|_\infty). \end{aligned}$$

5.2 The feature extraction map: Ψ_ρ

Let $\rho \in \mathbb{N}$ and $\widetilde{D_r}(X,\boldsymbol{\varepsilon}) = D_r(X,\boldsymbol{\varepsilon}) \setminus B$. We define the set $\widetilde{D_r}(X,\boldsymbol{\varepsilon},\rho)$, which consists of the ρ most persistent elements in $\widetilde{D_r}(X,\boldsymbol{\varepsilon})$, and the function

$$\Psi_\rho(D_r(X,\boldsymbol{\varepsilon})) = \widetilde{D_r}(X,\boldsymbol{\varepsilon},\rho) \cup \psi\big(D_r(X,\boldsymbol{\varepsilon}) \setminus \widetilde{D_r}(X,\boldsymbol{\varepsilon},\rho)\big) \cup B.$$

We remark that $(D_r(X,\boldsymbol{\varepsilon}) \setminus \widetilde{D_r}(X,\boldsymbol{\varepsilon},\rho)) \in \mathfrak{D}_r(\boldsymbol{\varepsilon})$, therefore $\Psi_\rho(D_r(X,\boldsymbol{\varepsilon}))$ is well defined. The map Ψ_ρ performs a *feature extraction* procedure. Indeed, the resulting persistence diagram consists of the most ρ persistent elements, say B. In contrast, the remaining elements, which are possibly large in numbers, are *compressed* into a single generator. It can be expressed by using Ψ_a as

$$\Psi_\rho = \widetilde{D_r}(X,\boldsymbol{\varepsilon},\rho) \cup \Psi_a(D_r(X,\boldsymbol{\varepsilon}) \setminus \widetilde{D_r}(X,\boldsymbol{\varepsilon},\rho)) \setminus (D_r(X,\boldsymbol{\varepsilon}) \setminus \widetilde{D_r}(X,\boldsymbol{\varepsilon},\rho)).$$

As we experiment in Section 6, Ψ_ρ is particularly useful when many generators are associated with *noisy* structures in the data, yielding significant savings in computational cost.

5.3 The auxiliary function ψ

We define the auxiliary function $\psi : \mathfrak{D}_r(\boldsymbol{\varepsilon}) \to \mathbb{R}^2_{\geq B}$ as

$$\psi(D_r(X,\boldsymbol{\varepsilon})) = \frac{1}{W} \sum_{\boldsymbol{x} \in \widetilde{D_r}(X,\boldsymbol{\varepsilon})} w(\boldsymbol{x})\boldsymbol{x} \tag{7}$$

where $w : \mathbb{R}^2_+ \longrightarrow \mathbb{R}_+$ is a weight function and $W = \sum_{\boldsymbol{x} \in \widetilde{D_r}(X,\boldsymbol{\varepsilon})} w(\boldsymbol{x})$. We observe that $\widetilde{D_r}(X,\boldsymbol{\varepsilon})$ contains a finite number of elements (generators), therefore the sum in (7) is always defined (see also Remark 5.1).

We can consider the following weights.

1. Let $w_1(\boldsymbol{x}) = 1/M$, being $M = |\widetilde{D_r}(X,\boldsymbol{\varepsilon})|$, i.e., the cardinality of the multiset, in which each element is counted with its multiplicity. We denote as *centre of uniform mass* the resulting auxiliary function

$$\psi_1(D_r(X,\boldsymbol{\varepsilon})) = \frac{1}{M} \sum_{\boldsymbol{x} \in \widetilde{D_r}(X,\boldsymbol{\varepsilon})} \boldsymbol{x}.$$

2. Let $\boldsymbol{x} = (b,d) \in \widetilde{D_r}(X,\boldsymbol{\varepsilon})$, where $b,d \in \mathbb{R}_+$ are the birth-death *time* of the element $\boldsymbol{x}$ (see (1)), and let $w_2(x) = d - b$ be the persistence of $\boldsymbol{x}$. We denote as *centre of persistence* the auxiliary function. Letting $C = \sum_{\boldsymbol{x}=(b,d) \in \widetilde{D_r}(X,\boldsymbol{\varepsilon})} (d-b)$

$$\psi_2(D_r(X,\boldsymbol{\varepsilon})) = \frac{1}{C} \sum_{\boldsymbol{x}=(b,d) \in \widetilde{D_r}(X,\boldsymbol{\varepsilon})} (d-b)\boldsymbol{x}.$$

Both options aim to extract a *representative* element from the generators of the persistence diagram. While the centre of uniform mass is the *barycentre* of the elements of the multiset, the centre of persistence assigns different weights according to the persistence of the elements. This is a natural choice for analysis, since elements with low persistence are more likely to be related to *noise* structures that give rise to the filtration. In contrast, aspects of significant persistence are associated with more representative geometric features of the dataset (see the example in Figure 2). For these reasons, in Section 6 we mainly use the function ψ_2 in our experiments, but we also compare ψ_1 and ψ_2 in Subsection 6.1.

Remark 5.1 *If we take $\boldsymbol{x} \in D_r(X,\boldsymbol{\varepsilon})$ in* (7)*, additional conditions on the weight function w are needed to guarantee the convergence of the sum. However, such an infinite setting is not meaningful to analyse, since elements in the bisector carry no topological information about the dataset. As a further observation, the centre of persistence might be formally computed by summing over $D_r(X,\boldsymbol{\varepsilon})$, as in this case $w(\boldsymbol{x}) = 0$ for $\boldsymbol{x} \in B$.*

Remark 5.2 *Since $D_r(X,\boldsymbol{\varepsilon})$ contains an infinite number of elements for all $X \subset \Omega$ by definition, the VSPK Φ_Ψ is still well defined on $\mathfrak{D}_r(\boldsymbol{\varepsilon}) \times \mathfrak{D}_r(\boldsymbol{\varepsilon})$. Moreover, if Φ is (strictly) positive definite, so it is Φ_Ψ.*

Remark 5.3 *Referring to Remark 4.1, if Φ is a PWG or SW kernel, then Φ_Ψ can be directly expressed in terms of the distance $d_\Psi(\cdot,\cdot)$ induced in the variably scaled setting.*

6 Tests

We aim to show how the SVM classifier may benefit from the introduced VSPKs. In all experiments presented, the kernels are computed using Python 3.8 and the `scikit-learn` module [31]. Scikit-learn is a Python module that integrates different machine learning algorithms for medium-scale supervised and unsupervised problems. The package has been designed as a machine-learning tool for non-specialists and implemented in a general-purpose high-level language such as Python. To reproduce the experiments, as referred to in [14], the open-source Python code is available at `https://github.com/reevost/vspk_code`.

We validated the following hyperparameters:

- for the SVM classifier: $\zeta \in \{10^j \mid j = -3, \dots, 3\}$;
- for the PSS kernel: $\sigma \in \{10^j \mid j = -3, \dots, 3\} \cup \{5 \cdot 10^j \mid j = -3, \dots, 2\}$;
- for the PWG kernel: C and τ are chosen in $\{10^j \mid j = -2, \dots, 2\}$, while $\delta = 10$ (see [22, Theorem 3.2]). Moreover, as the underlying standard kernel, we use the Gaussian;
- for the SW kernel: σ is obtained following the procedure carried out in [9, §4].

To assess the performance of the binary classifiers, we considered the *Accuracy* and the f_1-*score*. Letting TP=true positives, TN=true negatives, FP=false positives and FN=false negatives, these scores are defined as follows.

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}},$$
$$\text{f}_1\text{-score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}},$$

where

$$\text{precision} = \frac{\text{TP}}{\text{TP+FP}}, \quad \text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}.$$

The f_1-score, a performance metric used in classification problems, especially when classes are imbalanced, is the harmonic mean of precision and recall and is therefore widely used in the literature for providing a useful trade-off between these two important quantities. Hence, f_1-score is a number in $[0,1]$ with the extremal values telling the worst or the best performance in the classification process, respectively.

6.1 Alzheimer's Disease diagnosis

The Open Access Series of Imaging Studies (OASIS) is a well-known, publicly available neuroimaging dataset, widely used for research in neuroscience, machine learning, and medical image analysis. In particular, OASIS-3 is a larger (compared to OASIS-1 and 2) compilation of MRI and PET imaging data, along with related clinical data, from 1098 participants collected over 15 years across several ongoing studies at the Washington University Knight Alzheimer's Disease (AD) Research Center. Imaging

data are accompanied by dementia status, APOE status, and longitudinal clinical and cognitive outcomes [23, 30].

We considered a subset of the study group to ensure balanced data. For each subject, we constructed persistence diagrams using estimates of cortical thickness at 34 points in both hemispheres, yielding 64 values. For simplicity, we used the exact coordinates (computed with the *scipy* toolbox [38]) of the above-mentioned points across all subjects. From the coordinates, we constructed the persistence diagrams and extracted 1- and 2-dimensional topological features, i.e., the generators of H_1 and H_2.

In Figure 10, we show two examples of persistence diagrams, and in Figure 11, we highlight the generator added as the centre of the persistence diagram.

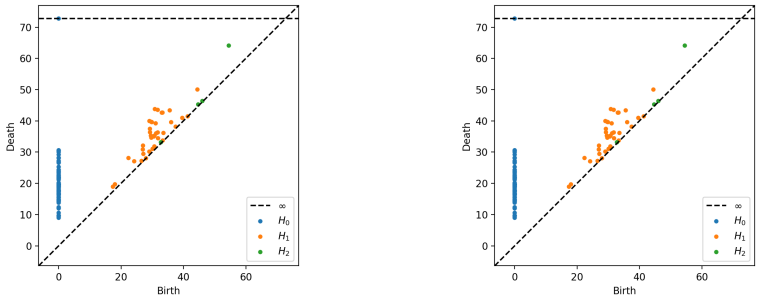


Figure 10: Persistence diagrams of a subject with AD with a MMSE of 30 (left) and the persistence diagram of a control subject with a MMSE of 7 (right).

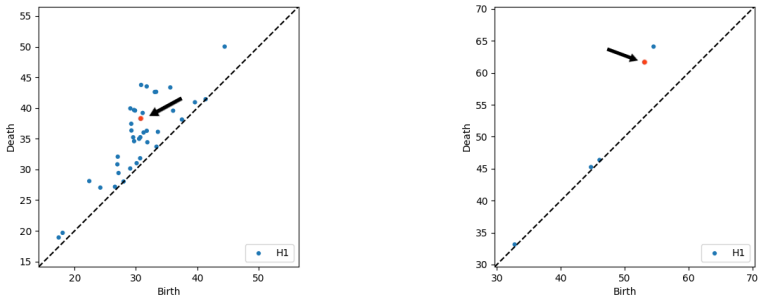


Figure 11: 1-dimensional (left) and 2-dimensional (right) persistence diagram of an AD subject. The red dot, pointed by a black arrow, is the added *center of persistence* via Ψ_a and ψ_2.

We evaluated the performance of an SVM classifier using the PSS, PWG, and SW kernels for both the classical and VSK settings. In each test, we performed a random 70%/30% split of the dataset for training and testing, and we used 5-fold cross-validation on the training set to tune the hyperparameters. The results displayed in

Tables 1 2 3 have been averaged over 10 runs of tests. The reported VT (Validation Time) values (in seconds) represent the time required to complete the entire cross-validation process.

In the VSK setting, we consider Ψ_a for both H_1 and H_2 diagrams, while Ψ_ρ, with $\rho = 10$, is employed with H_1 only, since H_2 diagrams are limited in the number of generators, and therefore compressing features is not meaningful.

Furthermore, we use ψ_2 as an auxiliary function. Indeed, as highlighted in Table 4, ψ_2 definitely outperforms ψ_1 in our setting. Moreover, we observe that the performance achieved by the auxiliary function alone, i.e., using the centers of mass or persistence in place of the persistence diagram, is definitely not competitive with the classical and variably scaled settings.

	Ψ	ψ	Acc.	f_1-score	VT (s)
SW (H_1)	-	-	0.741	0.716	220
VSP-SW (H_1)	Ψ_a	ψ_2	0.732	0.700	223
VSP-SW (H_1)	Ψ_ρ	ψ_2	0.731	0.693	184
SW (H_2)	-	-	0.741	0.712	183
VSP-SW (H_2)	Ψ_a	ψ_2	0.753	0.720	300

Table 1: OASIS-3 dataset. Results of SVM classification obtained using H_1 and H_2 persistence diagrams and the SW kernel.

	Ψ	ψ	Acc.	f_1-score	VT (s)
PWGK (H_1)	-	-	0.749	0.723	18165
VSP-PWG (H_1)	Ψ_a	ψ_2	0.750	0.726	18082
VSP-PWG (H_1)	Ψ_ρ	ψ_2	0.759	0.735	8731
PWG (H_2)	-	-	0.716	0.699	4422
VSP-PWG (H_2)	Ψ_a	ψ_2	0.709	0.683	4713

Table 2: OASIS-3 dataset. Results of SVM classification obtained using H_1 and H_2 persistence diagrams and the PWG kernel.

	Ψ	ψ	Acc.	f_1-score	VT (s)
PSS (H_1)	-	-	0.743	0.721	9238
VSP-PSS (H_1)	Ψ_a	ψ_2	0.752	0.728	9045
VSP-PSS (H_1)	Ψ_ρ	ψ_2	0.750	0.723	3330
PSS (H_2)	-	-	0.781	0.762	2825
VSP-PSS (H_2)	Ψ_a	ψ_2	0.775	0.755	3120

Table 3: OASIS-3 dataset. Results of SVM classification obtained using H_1 and H_2 persistence diagrams and the PSS kernel.

	Acc.	f_1-score
ψ_1	0.56	0.55
ψ_2	0.65	0.72

Table 4: OASIS-3 dataset. Results of SVMs classification obtained by using the center of mass and persistence alone in place of the persistence diagrams.

We can remark that VSPKs are competitive with the classical setting and, in some cases, outperform it. Conversely, using Ψ_ρ yields consistent savings in validation time.

6.2 Orbit detection

As a second experiment, we discuss the linked twisted map, which models fluid flows [1]. The linked twist map represents a Poincaré section of *eggbeater-type flow* in continuous dynamical systems. Letting

$$\mathbf{z}_n = \begin{pmatrix} x_n \\ y_n \end{pmatrix},$$

the corresponding orbits are computed via the discrete dynamical system

$$\mathbf{z}_{n+1} = F(\mathbf{z}_n) \qquad \mod 1,$$

where

$$F\left(\begin{pmatrix} x \\ y \end{pmatrix}\right) = \begin{pmatrix} x + ry(1-y) \\ y + r\big(x + ry(1-y)\big)\Big(1 - \big(x + ry(1-y)\big)\Big) \end{pmatrix}.$$

which can be rewritten as

$$\mathbf{z}_{n+1} = \begin{pmatrix} x_{n+1} \\ y_n + r x_{n+1}(1 - x_{n+1}) \end{pmatrix} \qquad \mod 1,$$

with $x_{n+1} = x_n + r y_n(1-y_n)$, which highlights that the second component depends on the updated value, making the map triangular (or sequentially coupled). We note that $\mathbf{z}_0 \in [0,1] \times [0,1]$ is the initial position and $r > 0$ is a real parameter that influences the orbit. The topological structure of the orbit changes with the initial position and r; when r is dense in the domain, voids form; otherwise, the orbit is dense. In Figures 12 and 13 we depict the first 1000 *iterations* $\{\mathbf{z}_n : n = 0, \dots, 1000\}$.

According to [1], we construct the persistence diagrams as follows.

- choose a set parameters r as classification labels (we took $r = \{2.5, 3.5, 4, 4.1, 4.3\}$;
- for each label, we compute the first 1000 points of 50 orbits, with a random starting point, giving a dataset consisting of 250 elements;
- compute the persistence diagram related to each orbit.

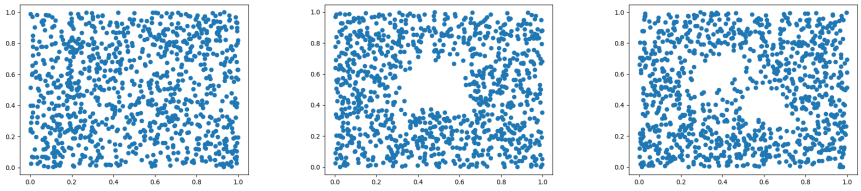


Figure 12: Fix $(x_0, y_0) \in [0,1]^2$, the orbits resulting from the linked twisted map taking the values $r = 2.5, 4.1, 4.3$, from left to right, respectively.

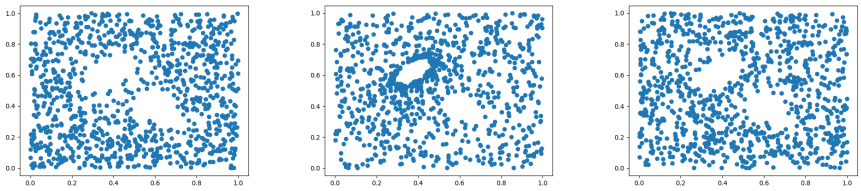


Figure 13: Fix $r = 4.3$. The resulting orbits, from the linked twisted map by taking different starting points $(x_0, y_0) \in [0,1]^2$.

Here, since each persistence diagram has a considerable number of generators ($\approx 10^5$), we restrict to the most $\rho = 10$ persistent elements in each diagram. Indeed, in our experiments, we observed that increasing the number of generators does not significantly improve classification performance but does increase computational cost. More precisely, here Ψ_a is computed with respect to such 10 elements, while in the case of Ψ_ρ the discarded less persistent generators are compressed in a unique element via ψ_2. Moreover, since we are dealing with 2-dimensional orbits, we compute only the H_1 homology group.

As in Section 6.1, we consider a 5-fold cross-validation on the training set, and the results displayed in Tables 5, 6, and 7 are averaged over 10 runs with a 70%/30% training-test split of the data.

	Ψ	ψ	Acc.	f_1-score
SW (H_1)	-	-	0.832	0.831
VSP-SW (H_1)	Ψ_a	ψ_2	0.814	0.812
VSP-SW (H_1)	Ψ_ρ	ψ_2	0.833	0.832

Table 5: Orbit Recognition. Results of SVMs classification on H_1 persistence diagrams using the SW kernel.

	Ψ	ψ	Acc.	f_1-score
PWG (H_1)	-	-	0.858	0.866
VSP-PWG (H_1)	Ψ_a	ψ_2	0.858	0.866
VSP-PWG (H_1)	Ψ_ρ	ψ_2	0.846	0.853

Table 6: Orbit Recognition. Results of SVMs classification on H_1 persistence diagrams using the PWG kernel.

	Ψ	ψ	Acc.	f_1-score
PSS (H_1)	-	-	0.806	0.803
VSP-PSS (H_1)	Ψ_a	ψ_2	0.824	0.821
VSP-PSS (H_1)	Ψ_ρ	ψ_2	0.826	0.823

Table 7: Orbit recognition. Results of SVMs classification on H_1 persistence diagrams using the PSS kernel.

7 Persistent homology and k-NN

We saw that, starting from a point cloud dataset, persistent homology yields graphs, such as the Barcode and Persistence Diagram. These diagrams collect information from a pair of $(birth, death)$ data, which describe the topological features that appear and disappear along a filtration. These objects, however, do not lie in a vector space or a normed space, so they cannot be used directly for classification. Despite this, we can use the Wasserstein and Bottleneck distances to compare persistence diagrams, since both distances measure how far the points move in the plane, with coordinates $(birth, death)$.

The *k-Nearest Neighbors (k-NN)* method is a simple, supervised, non-parametric, and lazy learning method used primarily for classification and regression, relying on feature similarity to predict outcomes, Generally, k-NN is used as a multi-class classifier [13]. A common task is to classify new data that has been added to an already classified dataset. The key notion is distance: the closer two objects are, the more likely they are to belong to the same class. Let's look at its use in the Figure 14

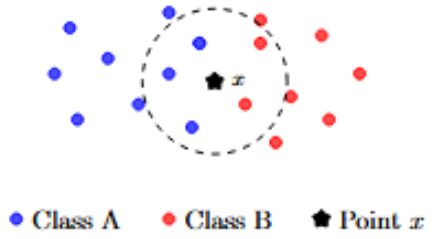


Figure 14: *k-NN* algorithm. The point x (star) finds the k nearest neighbors (in this case k=5), which are located inside the dashed circle. Based on the class to which the neighbors belong, the class of x (class A) is decided.

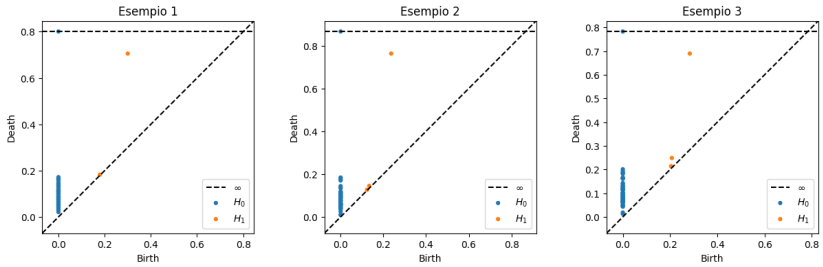
To perform classification on persistence diagrams using the k-NN algorithm, we can proceed in three steps.

1. Generate the persistence diagrams associated with the datasets.
2. Measure the distance between different persistence diagrams (with one of the two distances defined).
3. Classify with the k-NN method.

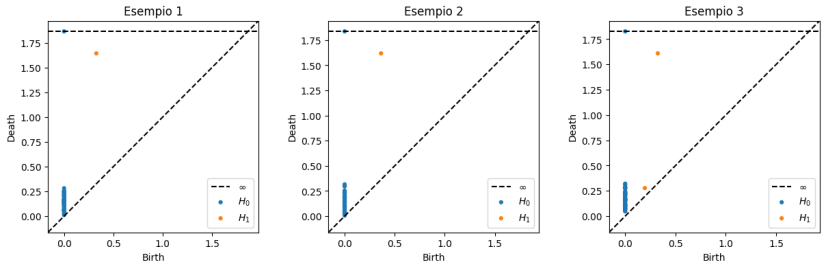
Thus, the distance between persistence diagrams is then used as the metric for k-NN classification. For readers interested in computing the bottleneck distance in Python, the `gudhi` module provides the function `gudhi.bottleneck_distance`, which operates on persistence diagrams. The Wasserstein distance between persistence diagrams can be computed using the function `wasserstein_distance_pd` available in the `scipy.stats` module.

Example 7.1 *We present an example illustrating how the k-NN algorithm works for classification on persistence diagrams. Suppose we have three classes of persistence diagrams and generate three examples for each class.*

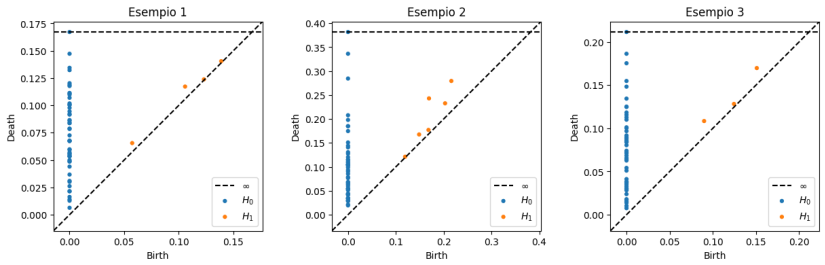
- ***Class 1:*** *persistence diagrams relative to a large circle.*



- ***Class 2:*** *persistence diagrams relative to a small circle.*



- ***Class 3:*** *persistence diagrams relative to a line.*



Now, let's give a new diagram, which is the one we need to classify.

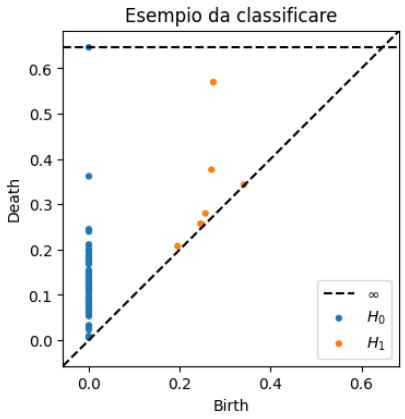


Figure 15: Persistence diagram to be classified

We calculate the bottleneck distances for each diagram. Using the k-NN algorithm with $k = 3$, we take the three closest diagrams, so we can determine the class to which the new persistence diagram belongs. We obtain that the three closest persistence diagrams are:

Class	Example	Distance
3	2	0.206
1	3	0.281
1	1	0.317

Table 8: Persistence diagrams closest to the one to be classified

We therefore observe that there are two persistence diagrams of class 0 and one of class 2, therefore the diagram to be classified will be assigned to ***class 0****.*

Classification with this algorithm is advantageous in some respects because it does not require vectorizing persistence diagrams, as is necessary, for example, with SVM. Furthermore, it is a simple, stable procedure, and is suitable for moderately large datasets.

Despite these limitations, it has some critical aspects: calculating distances can be expensive for very large datasets; its results depend on the value of k; and small variations in the persistence diagrams can significantly alter the distances, thereby impacting classification.

One solution to the lengthy computation time is to use an algorithm similar to k-NN: Approximate Nearest Neighbors (ANN).

This algorithm does not compute distances for all objects, as k-NN does, but instead approximates the desired neighbor, reducing computational time and considering only a smaller set. This method does not necessarily return the closest neighbor, but it is faster than k-NN. This algorithm is useful for large spaces, where dimensionality reduction is necessary prior to its application.

8 Conclusion and future developments

The classification of Persistence Diagrams using SVM enabled the introduction of a suitable definition of kernels acting on a pair of persistence diagrams. We analysed the behaviour of four *persistence kernels* (PKs) and their *scaled* version, making experiments for tuning the proper hyperparameters. This comparative analysis was largely absent in the literature but has recently been conducted in [14, 3], leading to the conclusion that SWK provides the best choice among the tested kernels, mainly for two reasons.

First, it predicts labels better than the other. It is computationally less expensive, since its `preGram` matrix depends on no parameters and can be computed only once at the beginning of the code. In [3], other datasets were considered, and it remains an open question why, in the ENZYMES dataset, all kernels perform poorly, resulting in lower accuracy than in other datasets.

Another interesting application of persistent homology worth mentioning is estimating the *Intrinsic Dimension* (ID) of a manifold or, more generally, a point cloud. Indeed, within the Machine Learning framework, the user typically works with large amounts of data and a considerable number of features. The number of the *essential features* is precisely what the Intrinsic Dimension is about. In the Ph.D. dissertation

by Cinzia Bandiziol [2, Chapter 5], the most popular methods for estimating ID today are discussed, including the *box-counting dimension* and the *correlation dimension*. We show how PH can be used to construct ID estimators, describing *i-Dimensional Persistent Homology Fractal Dimension*, *Persistent Homology Dimension*, and *Persistent Homology Complexity*. The numerical tests done in the dissertation reveal how *i-Dimensional Persistent Homology Fractal Dimension* ends up being "optimal" in the sense that its α generalization obtains its best value when $\alpha = 1$ [4]. Further analysis is definitely needed to better understand how the estimator poorly approximates the ID of some datasets, to improve, in the future, the methods themselves.

These final considerations show that Topological Data Analysis by Persistent Homology is an active area of research, combining pure mathematics with data science. Data analysis is the present and future of our society. We must make more bridges between mathematics and the applied sciences.

Acknowledgements. This work has been carried out within the *Approximation Theory and Applications* topical group of the Italian Mathematical Union, the RITA "Italian Network on Approximation", and the INdAM-GNCS group. This work has the support of PRIN 2023-2025, Computational mEthods for Medical Imaging (CEMI). We also thank the Padova Neuroscience Center for the opportunity to use neuroscience data. This project has been done as part of the Internationalization Exchange Program between the University of Padova and Selçuk Üniversitesi for 2025-2026.

References

[1] ADAMS, H., EMERSON, T., KIRBY, M., AND ET AL. Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research 18* (2017).

[2] BANDIZIOL, C. *Applications of Persistent Homology: Data Classification and Intrinsic Dimension of Manifold*. Phd thesis, University of Padova, Padova, Italy, March 2025. Available at `https://hdl.handle.net/11577/3557744`.

[3] BANDIZIOL, C., AND DE MARCHI, S. Persistence symmetric kernels for classification: A comparative study. *Symmetry 16*, 9 (2024). article 1236.

[4] BANDIZIOL, C., DE MARCHI, S., AND ALLEGRA, M. On intrinsic dimension of point clouds by a persistent homology approach: computational tips. *Modern Math. Models 3*, 3 (2025), 171–184.

[5] BORGWARDT, K. M., ONG, C. S., SCHÖNAUER, S., VISHWANATHAN, S. V., SMOLA, A. J., AND KRIEGEL, H. P. Protein function prediction via graph kernels. *Bioinformatics 21*, Suppl 1 (2005), i47–i56.

[6] BOZZINI, M., LENARDUZZI, L., ROSSINI, M., AND SCHABACK, R. Interpolation with variably scaled kernels. *IMA Journal of Numerical Analysis 35* (2015), 199–219.

[7] CAMPI, C., MARCHETTI, F., AND PERRACCHIONE, E. Learning via variably scaled kernels. *Advances in Computational Mathematics 47* (2021), 51.

[8] CARLSSON, G. Topology and data. *Bulletin of the American Mathematical Society (N.S.) 46*, 2 (2009), 255–308.

[9] CARRIÈRE, M., CUTURI, M., AND OUDOT, S. Sliced wasserstein kernel for persistence diagrams. In *Proceedings of the 34th International Conference on Machine Learning* (2017), vol. 70, JMLR.org, pp. 664–673.

[10] CHUL MOON, Q. L., AND XIAO, G. Using persistent homology topological features to characterize medical images: Case studies on lung and brain cancers, 2023.

[11] COHEN-STEINER, D., EDELSBRUNNER, H., AND HARER, J. Stability of persistence diagrams. *Discrete Comput Geom 37, 103–120 (2007) 37* (2007), 103–120.

[12] COLLINS, M., AND DUFFY, N. Convolution kernels for natural language. *In Proceedings of the 15th International Conference on Neural Information Processing Systems: Natural and Synthetic (NIPS'01)* (2001), 625–632.

[13] COVER, THOMAS M.; HART, P. E. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory 13*, 1 (1967), 21–27.

[14] DE MARCHI, S., LOT, F., MARCHETTI, F., AND POGGIALI, D. Variably scaled persistence kernels (vspks) for persistent homology applications. *Journal of Computational Mathematics and Data Science 4* (2022), 100050.

[15] EDELSBRUNNER, H., LETSCHER, D., AND ZOMORODIAN, A. Topological persistence and simplification. *Proceedings of the 41st Annual Symposium on Foundations of Computer Science* (2000), 454–463.

[16] FASSHAUER, G. E., AND MCCOURT, M. J. *Kernel-based Approximation Methods Using MATLAB*. World Scientific, Singapore, 2015.

[17] FOMENKO, A. *Visual Geometry and Topology*. Springer-Verlag, Berlin, 1994. translated from the Russian by Marianna V. Tsaplina.

[18] 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: Novel Methods in Harmonic Analysis*, vol. 2. Springer International Publishing, 2017, pp. 861–888.

[19] GUNNAR CARLSSON, M. V.-J. *Topological Data Analysis with Applications*. Cambridge University Press, 2022.

[20] HORAK, D., MALETIĆ, S., AND RAJKOVIĆ, M. Persistent homology of complex networks. *Journal of Statistical Mechanics* (2009), P03034.

[21] KELIN XIA, G.-W. W. Persistent homology analysis of protein structure, flexibility, and folding. *Numer. Meth. Biomed. Eng. 30*, 14 (2014).

[22] KUSANO, G., FUKUMIZU, K., AND HIRAOKA, Y. Kernel method for persistence diagrams via kernel embedding and weight factor. *Journal of Machine Learning Research 18* (2017), Paper No. 189.

[23] LAMONTAGNE, P. J., BENZINGER, T. L. S., MORRIS, J. C., KEEFE, S., HORNBECK, R., XIONG, C., GRANT, E., HASSENSTAB, J., MOULDER, K., VLASSENKO, A. G., RAICHLE, M. E., CRUCHAGA, C., AND MARCUS, D. Oasis-3: Longitudinal neuroimaging, clinical, and cognitive dataset for normal aging and alzheimer disease. *medRxiv* (2019).

[24] LAMPERT, C. H. Kernel methods in computer vision. In *Foundations and Trends in Computer Vision* (2009), vol. 4, pp. 193–285.

[25] LE, T., AND YAMADA, M. Persistence fisher kernel: A riemannian manifold kernel for persistence diagrams. In *NIPS'18* (2018), Curran Associates Inc., pp. 10028–10039.

[26] M., F. Persistent homology-based classification of chaotic multi-variate time series: Application to electroencephalograms. *SN COMPUT. SCI. 5*, 107 (2024).

[27] MARCHETTI, F., MARTINO, F. D., SHAMSEDDIN, M., DE MARCHI, S., AND BRISKEN, C. Variably scaled kernels improve classification of hormonally-treated patient-derived xenografts. In *2020 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)* (Bari, Italy, 2020), pp. 1–6.

[28] MERCER, J. Functions of positive and negative type and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society 209* (1909), 415–446.

[29] OTTER, N., PORTER, M. A., TILLMANN, U., ET AL. A roadmap for the computation of persistent homology. *EPJ Data Science 6* (2017), 17.

[30] PACHAURI, D., HINRICHS, C., CHUNG, M. K., JOHNSON, S. C., AND SINGH, V. Topology-based kernels with application to inference problems in Alzheimer's disease. *IEEE Transactions on Medical Imaging 30*, 10 (2011), 1760–1770.

[31] PEDEGROSA, F. ET AL. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research 12* (2011), 2825–2830.

[32] REININGHAUS, J., HUBER, S., BAUER, U., AND KWITT, R. A stable multi-scale kernel for topological machine learning. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), pp. 4741–4748.

[33] SCHABACK, R., AND WENDLAND, H. Kernel techniques: from machine learning to meshless methods. In *Acta Numerica*, vol. 15. 2006, pp. 543–639.

[34] SHAWE-TAYLOR, J., AND CRISTIANINI, N. Kernel methods for pattern analysis. In *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[35] SHAWE-TAYLOR, J., AND SUN, S. Kernel methods and support vector machines. In *Academic Press Library in Signal Processing, Volume 1*. Elsevier, 2014, pp. 857–881.

[36] SKAF, Y., AND LAUBENBACHER, R. Topological data analysis in biomedicine: A review. *Journal of Biomedical Informatics 130* (2022), 104082.

[37] TOWNSEND J., MICUCCI C.P., H. J. E. A. Representation of molecular structures with persistent homology for machine learning applications in chemistry. *Nature Commun. 11* (2020), 3230.

[38] VIRTANEN, P., GOMMERS, R., ET AL., O., AND SCIPY 1.0 CONTRIBUTORS. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods 17* (2020), 261–272.

[39] VISHWANATHAN, S. V. N., SCHRAUDOLPH, N. N., KONDOR, R., AND BORGWARDT, K. M. Graph kernels. In *Kernel Methods*, vol. 11. 2010, pp. 1201–1242.

[40] ZHANG, M., KALIES, W. D., KELSO, J. S., AND TOGNOLI, E. Topological portraits of multiscale coordination dynamics. *Journal of Neuroscience Methods 339* (2020), 108672.