

Learning topology: bridging computational topology and machine learning

Davide Moroni¹[0000-0002-5175-5126] and M. Antonietta
Pascali¹[0000-0001-7742-8126]

Institute of Information Science and Technologies,
National Research Council of Italy, Via Moruzzi, 1 - 56124-Pisa (IT)
{Name.Surname}@isti.cnr.it
www.isti.cnr.it

Abstract. Topology is a classical branch of mathematics, born essentially from Euler's studies in the XVII century, which deals with the abstract notion of shape and geometry. Last decades were characterised by a renewed interest in topology and topology-based tools, due to the birth of computational topology and Topological Data Analysis (TDA). A large and novel family of methods and algorithms computing topological features and descriptors (e.g. persistent homology) have proved to be effective tools for the analysis of graphs, 3d objects, 2D images, and even heterogeneous datasets. This survey is intended to be a concise but complete compendium that, offering the essential basic references, allows you to orient yourself among the recent advances in TDA and its applications, with an eye to those related to machine learning and deep learning.

Keywords: computational topology · persistent homology · machine learning · deep learning · image and shape analysis · data analysis.

1 Introduction

Topology is a branch of mathematics dealing with shape and geometry. Complexity and size of current collections of natural or synthetic dataset (2D, 3D, and multidimensional) is rapidly increasing. Hence, the ability to look at the shape of data and to discover patterns in any dimension is gaining great importance. Recently, successful applications of computational topology [27] to data analysis boosted a renewed interest in that field, and Topological Data Analysis (TDA) [10] has earned a prominent place in contemporary research, as a rich family of algorithms and methods from computational topology (e.g. Morse theory or persistent homology) to analyse and visualize data.

Focusing on image analysis, in low dimensions (typically 2 or 3), techniques from TDA are used to extract and classify geometric features, e.g. level sets or integral lines in [50]. For what regards Persistent Homology (PH), in [60] the authors describe how to define the Morse complex of a two or three-dimensional grayscale digital image, which is simpler than the cubical complex originally used

to represent the image and to compute persistent homology. Also, [46] show that *persistence diagrams* built from functions defined on objects are compact and informative descriptors and, for example, can be used for retrieval of images and shapes. Also, the topological representation of data could provide tools for hierarchical image segmentation, as in [68].

Looking at higher dimensions, e.g. to multidimensional datasets, techniques from TDA have been adapted to develop novel algorithms of data clustering: in 2008 Carlsson, with Singh and Sexton, contributed to founding Ayasdi (www.ayasdi.com), maybe the first machine intelligence platform with a TDA core able to compute groupings and similarity across large and high dimensional data sets, and to generate network maps visually supporting analysts in understanding data clusters (for example showing high dimensional patterns and trends) and which variables are relevant.

Several works have shown that TDA can be beneficial in a diverse range of problems, even very distant from each other, such as: studying the manifold of natural image patches [9]; analyzing activity patterns of the visual cortex [61]; in the classification of 3D surface meshes [57, 46]; complex networks [56, 41]; clustering [20, 55]; recognition of 2D object shapes [64]; protein folding [7, 67, 42]; viral evolution [18].

In this survey, we focus specifically on persistent homology (PH), because we found this technique the most interesting with respect to the interplay with machine and deep learning.

Outline The following section is devoted to providing the reader basic notions and a historical overview of the main results in the theory of persistence in computational topology, starting from early works by P. Frosini, V. Robins and H. Edelsbrunner, which established independently the very first definitions and theorems. Section 3 deals with the computability of the most used PH descriptors, together with a summary of the software developed to compute PH. Section 4 explains the versatility of such methods, applied in several domains: one of the main factors of the recent interest around PH. Section 5 gives the reader a tour in the most promising applications of TDA in machine and deep learning, showing the great potential of embedding topological tools in the learning pipeline along with implementations of *topological layers*. These recent efforts have given rise to the novel field of Topological Machine Learning. The last section concludes the paper, and it is devoted to emerging studies focused on the interplay of topological data analysis and deep learning theory: e.g. concerning how to exploit tools from topological analysis to enhance explainability and interpretability of artificial intelligence methods.

2 Persistence homology: history and basic notions

Algebraic topology is a branch of mathematics using tools from abstract algebra to study and characterise topological spaces (see [38] for an introductory textbook). The basic aim is to find algebraic invariant able to classify topological

spaces up to homeomorphism. Persistent Homology bridges algebraic topology with the Morse theory core idea: exploring topological attributes of an object in an evolutionary context. The concept of *persistence* was introduced independently in 1990 by P. Frosini, M. Ferri and collaborators in Bologna (Italy), by V. Robins in 2000 in her PhD thesis devoted to multi-scale topology applied to fractals and dynamics, and by the group of Edelsbrunner at Duke (North Carolina).

2.1 Basics

In order to understand the core idea of PH, it is necessary to be familiar with the basics of algebraic topology. A simplicial complex is the standard algebraic object used to represent shapes of any dimension; simplices are its building blocks.

Definition 1. A k -simplex is the k -dimensional convex hull of $k + 1$ vertices. The convex hull of any nonempty subset of the $k + 1$ vertices is called a face of the simplex

A simplicial complex K is a set built from 0-dimensional simplices (0-simplices or points), 1-dimensional simplices (line segments), 2-simplices (triangles), 3-simplices (tetrahedra) and so on. The dimension of K is defined as the largest dimension of any simplex in it. Actually, to be a simplicial complex, K should satisfy the following conditions.

Definition 2. A simplicial complex K is a set of simplices such that:

- every face of a simplex is also a simplex of K ;
- the intersection of any two simplices σ_1 and σ_2 in K is either a face of both σ_1 and σ_2 , or the empty set.

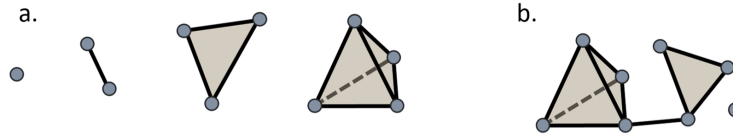


Fig. 1. a. An example of a simplex for each dimension from 0 to 3. b. An example of a 3-dimensional simplicial complex.

These conditions allow defining the *boundary* operator, which is fundamental to define the homology, an algebraic object computable (via linear algebra) for K that accounts for the number of connected components, holes, voids, etc. The *boundary* of a k -simplex $\partial(\sigma_j)$ is the formal sum of the $(k - 1)$ -dimensional faces of σ_j . For example, the boundary of a triangle $\{a, b, c\}$ of vertices a, b and c is given by the sum of the edges $\{b, c\} + \{a, c\} + \{a, b\}$. More formally,

$$\partial(\{v_0, \dots, v_k\}) = \sum_{i=0}^k (-1)^i \{v_0, \dots, \hat{v}_i, \dots, v_k\}$$

It's straightforward that a boundary has no boundary: $\partial \circ \partial = \partial^2 = 0$. Let K be a k -simplicial complex, and F a field; in PH the most used F is the two-element field $\mathbb{F}_2 = \mathbb{Z}/2\mathbb{Z}$. Let $\{\sigma_1, \dots, \sigma_n\}$ be the set of p -simplices of K , where $p \in \{0, 1, \dots, k\}$. $C_p(K)$ denotes the vector space generated over F by the p -dimensional simplices of K ; hence, $C_p(K)$ is made of all p -chains, which are the formal sums over the p -simplex $c = \sum_{j=1}^n a_j \sigma_j$ where $a_j \in F$ and σ_j is a p -simplex in K . Hence, the boundary operator defined above is a linear operator between chain vector spaces: $\partial_p : C_p(K) \rightarrow C_{p-1}(K)$; also, now we define p -cycles $Z_p(K)$ and k -boundaries $B_p(K)$

$$Z_p(K) := \text{Ker}(\partial : C_p \rightarrow C_{p-1}) \text{ and } B_p(K) := \text{Im}(\partial : C_{p+1} \rightarrow C_p).$$

And it yields that: $B_p(K) \subset Z_p(K) \subset C_p(K)$. Finally, the p -th homology group $H_p(K)$ is defined as the quotient space Z_p/B_p : two cycles c_1 and c_2 are *homologous* if they are in the same homology class: $\exists b \in B_p(K)$ such that $c_2 - c_1 = b$. In algebraic topology, the homology group of a complex is one of the most studied and used. Also, the homology group is linked to the Betti numbers β_p , very important topological invariants: $\beta_p(K) = \dim(H_p(K))$. The p -th Betti number, informally, counts the number of p -dimensional holes on a topological surface; for example, a two-dimensional torus has $\beta_0 = 1$ (it is connected), $\beta_1 = 2$ (it shows two independent loops on its surface), and $\beta_2 = 1$ (only one cavity). Back to the case of a simplicial complex K , the 0-dim Betti number is the number of connected components and 2-dim Betti number is the number of voids of K .

The homology group and the Betti numbers are able to encode the global topological properties of a shape, represented by a simplicial complex.

Persistence needs a core ingredient: filtrations. A filtration of a simplicial complex K is a sequence of nested sub-complexes:

$$\emptyset = K^0 \subset K^1 \subset \dots \subset K^m = K$$

In a few words, think of a filtration as a way to build the given complex iteratively by adding simplices, starting from vertices, step by step. Of course, given a complex, there are many ways to define a filtration. Depending on the data, different filtered simplicial complexes are considered, and the definition of most of them is based on the distance induced by the metric of the ambient space of data. At each filtration step t_j consider the simplicial complex K^j , sub-complex of K , and compute the rank of each p -th homology group (i.e. β_p) for each $p \in \{0, 1, \dots, k\}$: the variation of the β_0 (or β_1) will account for the birth or death of connected components (or loops). Increasing p , the variation of Betti numbers will account for p -dimensional *topological features*, whose evolution is able to encode precious information about the global structure of the growing complex looking at the lifespan (death - birth) of each topological feature.

2.2 History

Frosini, Ferri and collaborators [30, 32, 31, 66] in a family of papers published in between 1990 and 1993, introduced the size functions, which are equivalent to the

0-dimensional persistent homology. The size functions are defined as functions from the real plane to the natural numbers which describe the *shape of the objects* (seen as sub-manifolds of a Euclidean space). Also, different techniques of computation of size functions are provided, together with the definition of a deformation distance between manifolds measuring the *difference in shape* of two manifolds, and applications to shape analysis.

Edelsbrunner et al. in 2002 [28] formalize the notion of persistence within the framework of a filtration, which is the history of a growing complex. They introduced the classification of a topological event occurring during growth as either a *feature* or *noise*, depending on its lifespan within the filtration. The algorithm provided in this paper for computation yields only for sub-complexes of spheres and only with coefficients in \mathbb{F}_2 .

Zomorodian and Carlsson in 2005 [69] show that the persistent homology of a filtered d -dimensional simplicial complex is simply the standard homology of a particular graded module over arbitrary field coefficients. In the same paper, authors provide an algorithm for computing individual persistent homology groups over an arbitrary principal ideal domain in any dimension. They also introduced *barcodes*, a combinatorial invariant; a complete introduction to persistence homology and its application from the perspective of barcodes is provided by Ghrist in [35].

From now on, the research community becomes more and more interested not only in the theoretical advances in persistence, but also in how to implement the persistence algorithms, in order to exploit the existence of computable topological descriptors in shape analysis, and, more generally, in data analysis. In 2010 Plex, the first software for computing persistence is released (see also Section 3.1).

2.3 Persistence diagrams

The first descriptors derived from PH are persistence diagrams. Such descriptors provide a (visual) summary of births and deaths of topological events. E.g. they track when a loop appears and disappears while the complex is growing (i.e. while the filtration parameter increases). From another viewpoint, they can be seen as a parametrized version of the Betti numbers.

More in detail, a persistence diagram is a collection of points in R^2 : any topological feature has a birth b and a death d , and is represented in diagrams as a point of coordinates (b, d) . Two persistence diagrams may be compared efficiently using proper distances such as the p -Wasserstein distance, or the bottleneck distance which is the limit over p of the p -Wasserstein distance with p going to infinity. The Bottleneck (or matching) distance d_B between two subsets X and Y of a metric space (M, d) is:

$$d_B(X, Y) := \inf_{\phi} \sup_{x \in X} d(x, \phi(x))$$

where ϕ runs over all bijections between X and Y .

Bottleneck distance is used to compare persistence diagrams and to derive their stability. Indeed, despite the way these descriptors are built, they show robustness with respect to noise, as stated in the stability theorem [21, 6]: persistence diagrams are stable with respect to perturbations of the data, and such a stability yields also for multidimensional persistent homology, as shown in [16].

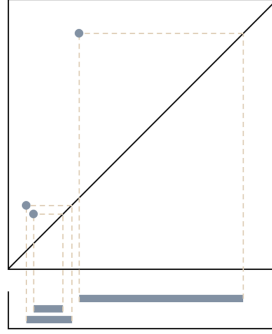


Fig. 2. Diagrams and barcodes are equivalent representations of topological persistence, and the correspondence is visually proved in this picture. (credits: Matthew L. Wright, 2014 <http://www.mrwright.org>)

An alternative way to represent topological persistence is given by *barcodes*, which are equivalent to diagrams, and their equivalence is shown in Figure 2. Topological features, in barcodes, are represented as line segments of length $(d - b)$ where b is its birth, and d is its death; hence, a barcode is a collection of horizontal bars in a plane: the horizontal axis corresponds to the filtration parameter growing the complex, while the vertical axis represents an (arbitrary) ordering of homology generators. Barcodes have been introduced in [69] along with an algorithm to compute them, via linear algebra on the boundary matrix. The runtime of that algorithm is $O(n^3)$, where n is the number of simplices.

In Figure 3 the first row shows a growing complex associated to a set of points sampled on a torus. The filtration is defined using the Euclidean metric (Rips complex). In the second row, there are its barcodes of dimension 0, 1, and 2. The length of each bar is the *lifespan* of the corresponding generator: long bars are interpreted as relevant features, while short bars as noise. Equivalently in diagrams, as can be seen from Figure 2, dots near the diagonal represent noise.

Unfortunately, persistence diagrams and barcodes exhibit a complex structure, and are difficult to integrate into today's machine learning workflows. This is why the community currently works to define novel topological descriptors derived from the topological ones, or defining novel differentiation rules applicable to known topological descriptors.

In the following section, a list of the most successful PH-based descriptors is provided and discussed, together with the current algorithms and software used (and developed) by the research community to compute them.

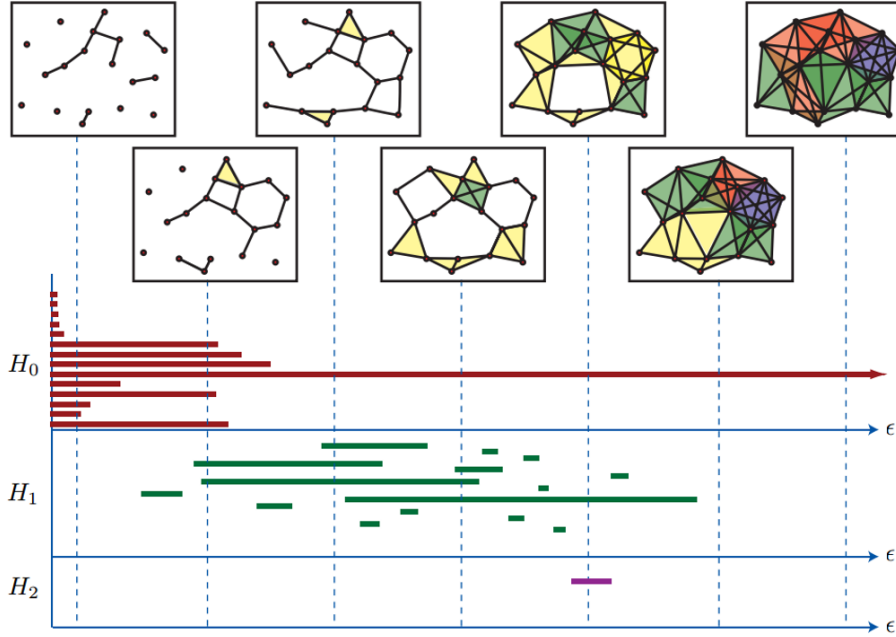


Fig. 3. The rank of the homology groups of dimension 0, 1, and 2 associated with a sequence of nested sub-complexes. Look at the longest feature in H_0 : it represents that along the sequence only one connected component survives. This figure is by R. Ghrist [35].

3 PH-based descriptors and implementations

After persistence diagrams, other PH-based descriptors were defined, implemented, and used: persistence silhouette, persistence landscape, and persistence images.

The birth of them was motivated by the need for stronger properties of stability and for easy and fast algorithms to compute and compare them, in order to increase their usage and efficacy in data analysis.

In 2015 Bubenik [8] developed the notion of a *persistence landscape*, a stable functional representation of a persistence diagram that lies in a Banach space (Hilbert, for $p = 2$), where statistical learning methods can be directly applied. The persistence landscape is a collection of continuous, piecewise linear functions $\lambda_p : \mathbb{N} \times \mathbb{R} \rightarrow \mathbb{N}$ that summarizes a persistence diagram. For $1 \leq p \leq \infty$ the p -landscape distance between two landscapes λ_1 and λ_2 is defined as $\|\lambda_1 - \lambda_2\|_p$; the ∞ -landscape distance is stable with respect to the bottleneck distance, and the p -landscape distance is continuous with respect to the p -Wasserstein distance on persistence diagrams. Statistical properties of landscapes and similar descriptors (average landscape, silhouette) are investigated in [19], resulting in establishing useful stability properties.

In [1] barcodes are mapped to the so-called persistence surfaces. This is done by computing a weighted sum of normalized (isotropic) Gaussians, evaluated at each point in the diagram. Upon discretization of this persistence surface, one obtains the persistence image, stable and computable, that can then be concatenated in a vector and fed to a support vector machine.

3.1 Implementations

In this subsection, we present the most relevant implementations of PH. The goal is to provide effective tools for the computation of barcodes as well as methods for their analysis and comparison. A reference paper for the comparative analysis is represented by the work [54], in which a benchmark of the selected open-source implementations is carried out on 12 reference public-available datasets. In addition, other software tools not described in the paper [54] are briefly surveyed here for the sake of completeness.

Javaplex [2] implements persistent homology and related techniques from computational and applied topology, in a library designed for ease of use, ease of access from Matlab and java-based applications. The Computational Topology workgroup at Stanford University has mainly developed JavaPlex, which is grounded on previous similar packages from the same group. Among them, **Plex** is the first known software providing computation of PH, as well as the first proposing the use of zigzag persistence [11].

Dionysus [51], and its new version **Dionysus 2**, are C++ libraries for the computation of persistent homology. Dionysus has been the first software package to implement the dual algorithm [23], but it is also known since it contains advanced tools for the construction of vineyards (i.e. continuous families of persistence diagrams) [22], for the determination of homology generators and for the computation of Wasserstein and bottleneck distances.

Another interesting C++ software package is **Perseus** [53] which computes the persistent homology leveraging Morse-theoretic reduction. Since the standard algorithm for computing persistence intervals relies on Smith normal form and is therefore of super-cubical complexity in the total number of cells, reducing the number of cells might result in relevant savings both in memory and time. Perseus achieves a drastic reduction of the number of cells in the original filtration in linear time via discrete Morse theory without altering its persistent homology [49]. In addition, being based on general discrete Morse theory, this preprocessing step does not rely on peculiarities of a particular type of complex structure, but it can be applied straightforwardly to simplicial complexes, cubic complexes and Vietoris-Rips complexes to name a few.

More recently, **PHAT** [5] (and its spin-off **DIPHA** [4] devoted to distributed calculus) proposed a C++ implementation focused on the efficient and fast computation of PH based on matrix reduction. The authors aimed at a generic design that decouples algorithms from data structures; several different reduction strategies as well as data types to store and manipulate the boundary matrix are provided. **GUDHI** [47] is a C++ library with a Python interface implementing an efficient data structure for general simplicial complexes (*simplex tree*)

as well as the possibility to compute simultaneously persistence diagrams with coefficients over multiple finite fields \mathbb{F}_p .

The **R package TDA** [29] provides an R interface for the efficient algorithms of the C++ libraries GUDHI, Dionysus and PHAT, including the PH of Rips filtrations and of sublevel sets of arbitrary functions evaluated over a grid of points.

SIMBA [24] implements a new algorithm, leveraging on a batch collapse strategy as well as a new sparse Rips-like filtration, that enables the approximation the persistent homology of Rips filtrations with quality guarantees. A software, developed in C++, is made available upon request and, in practical application, is an order of magnitude faster than existing methods.

The **Topology ToolKit (TTK)** [48] is an open-source library for TDA which implements, in a generic and efficient way, a substantial collection of reference algorithms, including those for the computation of persistence diagrams. The main merit of TTK is having made such algorithms accessible to a wider community by proving a library integrated with state-of-the-art libraries for scientific visualization (VTK) and image analysis (ITK) and with a graphical front-end, i.e. Paraview.

4 A plethora of applications

PH, as concern data analysis, is a versatile method: there is no restriction to apply to any particular kind of data (such as images, sensor measurements, time-series, graphs, etc.). When we want to analyse an image, a shape or a dataset, generally we choose a representation for the input data, e.g. the vertices of a triangulation for a shape or an n -dimensional point cloud for a dataset, along with a (natural) notion of distance, or similarity, between them. This distance is generally induced by the metric in the ambient space (e.g. the Euclidean metric) or may come as an intrinsic metric defined by a pairwise distance matrix. It is important to notice that the choice of the metric may be critical to reveal interesting topological and geometric features of the data. To exploit PH methods, depending on the data, different filtered simplicial complexes may be used, e.g. Vietoris-Rips complex, Čech complex, Alpha complex, Witness complex, Morse complex, cubical complex, clique (or flag) complex, or CW complex.

In general, the TDA pipeline consists of the following three steps: 1. Give to input data a multi-scale topological structure, i.e. a complex along with a filtration; 2. Compute multi-scale topological signature (as PH-based descriptors); 3. Take advantage of the signature to perform pattern analysis tasks, exploiting machine learning methods (i.e. looking at statistical aspects and representations of topological persistence).

In the following, there is a focus on applications of PH to image and signal analysis.

Image and shape analysis Results in [9] provide an example of how unexpected could be the findings of topological enquire of imaging: the authors

showed that the space of 3×3 high-contrast patches from digital images has the topology of a Klein bottle, and suggested to use this fact to implement a novel compression method of 2D images.

First applications of PH were in 2D and 3D shape analysis, and specifically in diverse tasks, i.e. classification, segmentation, retrieval, and many others. We list here just a few examples: in 2006 [17] the original size functions were used to generate 25 measuring functions for automatic retrieval of trademark images, outperforming existing whole-image matching techniques; In 2010 [62] a persistence based clustering and the Heat Kernel Signature function are combined to achieve a multi-scale isometry invariant segmentation of deformable shapes; in [64], Turner and colleagues demonstrated how PH may be used to represent shapes and execute operations such as computing distances between shapes or classifying and modeling shapes and surfaces. Persistence descriptors have been used also statistical shape analysis, as in [34].

Signal processing and time series analysis Even if PH originates in the context of image and shape analysis, due to its versatility it was successfully and largely applied in signal processing and analysis. Indeed PH provides efficient tools to denoise and analyse both homogeneous and heterogeneous time series, and many researchers exploited topological features.

Perea and Harer [55] used a sliding window approach to obtain a point cloud from a time series; the point cloud is then analysed looking at periodicity as the repetition of patterns, quantifying this recurrence as the degree of circularity/roundness in the generated point-cloud. This approach has been applied data from gene expression and physiology, astronomical data, and weather.

Y. Umeda, in [65], proposed a novel approach for the classification of volatile time series: TDA is used to extract the structure of attractors, resulting efficient for both chaotic and non-chaotic time series, achieving performances improved of 18.5% compared to conventional approaches.

In [3] the occupancy of specific areas or rooms in a smart building is monitored, using a method based on the analysis of a set of topological features extracted from the data acquired in a room for a week by three different low-cost sensors.

The set of signals to which TDA can be applied is today quite rich; it includes, for example, physiological signals such as EEG or ECG (as in [26]), and financial time series such as stock market indices. The analysis of market crashes in [36] is quite interesting, because was the first application of TDA to this kind of data, providing a new type of econometric analysis, which complements the standard statistical measures, to perform a reliable early detection of early warning signals of imminent market crashes.

5 New trends: PH into ML

The idea of allowing neural networks to learn topological information has been explored most frequently by feature engineering, looking at some predefined stan-

standard features conveying topological information. Only very recently, researchers devoted such an effort in building topological layers to be used in deep learning. Unfortunately, even if persistence diagrams and barcodes found a large number of applications, the space of persistence diagrams lacks structure, e.g. different persistence diagrams may have a different number of points, and several basic operations are not well-defined, such as addition and scalar multiplication: the (metric) space of persistence diagrams is not a Hilbert space. In addition, the cost of computing the bottleneck or Wasserstein distance grows quickly as the number of off-diagonal points in the diagrams increases [25].

To tackle this issue, a lot of effort has been devoted to vectorization and kernel methods. Vectorizations of persistence diagrams is based on the construction of either finite-dimensional embeddings [1, 14, 19], i.e., embeddings turning persistence diagrams into vectors in Euclidean space \mathbb{R}^d . We already met some of them in Section 3: landscapes and images are the most referred and used.

Persistence kernels are generalized scalar products that implicitly turn persistence diagrams into elements of infinite-dimensional Hilbert spaces. As for vectorization, the construction of kernels for persistence diagrams, preserving their stability properties has attracted some attention. Most kernels have been obtained by considering diagrams as discrete measures in \mathbb{R}^2 . Convolving a symmetrized (with respect to the diagonal) version of persistence diagrams with a 2D Gaussian distribution, Reininghaus et al. [58] introduce a multi-scale kernel to perform shape classification and texture recognition. Considering Wasserstein distance between projections of persistence diagrams on lines, Carriere and Oudot [13] build another kernel and test its performance on several benchmarks. Other kernels, still obtained by considering persistence diagrams as measures, have also been proposed by Kusano et al. [44] and Le et al. [45].

Refer to [40] for further details about vectorization and kernel approaches to represent barcodes.

Even though vectorization and kernel methods improved the use of persistence diagrams in machine learning tremendously, several issues remain. For instance, most of them only have a few trainable parameters; therefore, it may be very difficult to determine which vectorization is going to work best for a given task. On the contrary, kernel methods are generally efficient, but require large memory resources to compute and store the kernel evaluations (whose computations have at least linear complexity) for each pair of persistence diagrams. Hence, such methods are very costly with respect to memory usage and running time on large datasets.

In general, a framework using topological signatures in a neural network could suffer from some limitations: *(i)* it may rely on a particular filtration, *(ii)* it may lack stability results, and *(iii)* the differentiability of persistent homology generally is not guaranteed with respect to the layer's input. Hence, such a *topological* layer cannot be placed in the middle of a deep network. In the following a list of most used and successful topological layers, most of them published with code, is provided.

5.1 Topological layers

In these last five years, many research groups defined and implemented topological layers to exploit topological features in deep learning pipeline.

Hofer et al.[39] first developed a technique to input persistence diagrams into neural networks by introducing their own topological layer, able to learn a task-optimal representation during training.

In [33] the authors propose a differentiable **Topology Layer** that computes persistent homology, based on level set filtrations and edge-based filtrations. It is publicly available and its implementation is based on PyTorch. A note: this layer may be placed at the beginning of a deep network and, using the fact that our input layer is differentiable, it can be used to perform adversarial attacks (gradient attack), i.e. cause a trained neural network to misclassify input.

PersLay [15] is one of the first neural network layers, designed to handle persistence diagrams. It is based on a general framework for diagram vectorization: maybe the simplest way to generate a permutation-invariant and differentiable feature map is to turn each point of the persistence diagram into a vector, and then sum over all such vectors to eventually get a single vector. This is the core idea of Perslay: depending on the way the diagram points are turned into vectors and on the permutation-invariant operation that is being used, one can show that one can compute persistence images, persistence landscapes, persistence silhouettes as particular instances of Perslay.

Very recently, in [43] the authors propose **PLLAY**, a layer based on the weighted persistence landscapes. They show a tight stability bound that does not depend on the input complexity; therefore PLLAY is less prone to extreme topological distortions. Importantly, they provide guarantees of the differentiability of PLLAY with respect to the layer’s input: hence, such a layer may be placed anywhere in the network.

In [63] it is demonstrated how to fuse persistence image computation in supervised deep learning architectures: **PI-Net** is maybe the first framework using deep learning for computing topological features directly from data. Authors tested such framework on two applications: human activity recognition using tri-axial accelerometer sensor data and image classification. Also, the authors speeded up the extraction of persistence images form data of several orders of magnitude, paving the way to new real-time applications for TDA.

6 Conclusions

The present paper provides an overview of TDA and PH, enabling the reader to appreciate the remarkable steps to the spread and great popularity of such methods: *(i)* from persistence theory to computable topological descriptors, *(ii)* from algorithms to fast computation, enabling people to compute topological features and explore their efficacy (e.g. classification or clustering) in several application domains; *(iii)* from the computation of PH-descriptors to the development of topological layers for deep learning.

Moreover, PH gave rise to novel techniques for improving or even understanding CNN. For example, Guss and Salakhutdinov [37], Rieck et al. [59] proposed a complexity measure for neural network architectures based on topological data analysis; Carlsson and Gabrielsson [12] applied topological approaches to deep convolutional networks to understand and improve the computations of the network; and Naitzat et al [52] provide insights in how shallow and deep nets behave with respect to the topology of input data.

In this perspective, we have foreseen that the interplay of PH with machine learning and deep learning techniques will continue to be successful, for example, in machine learning explainability, using the topological lens to look into the black-box.

References

1. Adams, H., Emerson, T., Kirby, M., Neville, R., Peterson, C., Shipman, P., Chepushtanova, S., Hanson, E., Motta, F., Ziegelmeier, L.: Persistence images: A stable vector representation of persistent homology. *J. Mach. Learn. Res.* **18**(1), 218–252 (2017)
2. Adams, H., Tausz, A., Vejdemo-Johansson, M.: Javaplex: A research software package for persistent (co) homology. In: *International Congress on Mathematical Software*. pp. 129–136. Springer (2014)
3. Barsocchi, P., Cassarà, P., Giorgi, D., Moroni, D., Pascali, M.: Computational topology to monitor human occupancy. *Proceedings* **2**(99) (2018)
4. Bauer, U., Kerber, M., Reininghaus, J.: Dipha (a distributed persistent homology algorithm). Software available at <https://github.com/DIPHA/dipha> (2014)
5. Bauer, U., Kerber, M., Reininghaus, J., Wagner, H.: Phat–persistent homology algorithms toolbox. *Journal of symbolic computation* **78**, 76–90 (2017)
6. Biasotti, S., Cerri, A., Frosini, P., Giorgi, D., Landi, C.: Multidimensional size functions for shape comparison. *Journal of Mathematical Imaging and Vision* **32**(2) (2008)
7. Bowman, G., Huang, X., Yao, Y., Sun, J., Carlsson, G., Guibas, L., Pande, V.: Structural insight into rna hairpin folding intermediates. *J Am Chem Soc.* **130**(30), 9676–8 (2008)
8. Bubenik, P.: Statistical topological data analysis using persistence landscapes. *Journal of Machine Learning Research* **16**(3), 77–102 (2015)
9. Carlsson, G., Ishkhanov, T., de Silva, V., Zomorodian, A.: On the local behavior of spaces of natural images. *International Journal of Computer Vision* **76**, 1–12 (2008)
10. Carlsson, G.: Topology and data. *Bull. Amer. Math. Soc.* **46**, 255–308 (2009)
11. Carlsson, G., De Silva, V.: Zigzag persistence. *Foundations of computational mathematics* **10**(4), 367–405 (2010)
12. Carlsson, G., Gabrielsson, R.B.: Topological approaches to deep learning. In: Baas, N.A., Carlsson, G.E., Quick, G., Szymik, M., Thau, M. (eds.) *Topological Data Analysis*. pp. 119–146. Springer International Publishing, Cham (2020)
13. Carrière, M., Cuturi, M., Oudot, S.: Sliced wasserstein kernel for persistence diagrams. In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. p. 664–673. ICML’17, JMLR.org (2017)
14. Carrière, M., Oudot, S.Y., Ovsjanikov, M.: Stable topological signatures for points on 3d shapes. *Computer Graphics Forum* **34**(5), 1–12 (2015)

15. Carrière, M., Chazal, F., Ike, Y., Lacombe, T., Royer, M., Umeda, Y.: Perslay: A neural network layer for persistence diagrams and new graph topological signatures (2020)
16. Cerri, A., Fabio, B., Ferri, M., Frosini, P., Landi, C.: Betti numbers in multidimensional persistent homology are stable functions. *Mathematical Methods in the Applied Sciences* **36**, 1543–1557 (2013)
17. Cerri, A., Ferri, M., Giorgi, D.: Retrieval of trademark images by means of size functions. *Graphical Models* **68**(5), 451 – 471 (2006), special Issue on the Vision, Video and Graphics Conference 2005
18. Chan, J.M., Carlsson, G., Rabadan, R.: Topology of viral evolution. *Proceedings of the National Academy of Sciences* **110**(46), 18566–18571 (2013)
19. Chazal, F., Fasy, B.T., Lecci, F., Rinaldo, A., Wasserman, L.: Stochastic convergence of persistence landscapes and silhouettes. In: *Proceedings of the Thirtieth Annual Symposium on Computational Geometry*. p. 474–483. SOCG’14, Association for Computing Machinery, New York, NY, USA (2014)
20. Chazal, F., Guibas, L.J., Oudot, S.Y., Skraba, P.: Persistence-based clustering in riemannian manifolds. *J. ACM* **60**(6) (2013)
21. Cohen-Steiner, D., Edelsbrunner, H., Harer, J.: Stability of persistence diagrams. *Discrete Comput Geom* **37**, 103–120 (2007)
22. Cohen-Steiner, D., Edelsbrunner, H., Morozov, D.: Vines and vineyards by updating persistence in linear time. In: *Proceedings of the twenty-second annual symposium on Computational geometry*. pp. 119–126 (2006)
23. De Silva, V., Morozov, D., Vejdemo-Johansson, M.: Dualities in persistent (co) homology. *Inverse Problems* **27**(12), 124003 (2011)
24. Dey, T.K., Shi, D., Wang, Y.: SimBa: An efficient tool for approximating Rips-filtration persistence via simplicial batch collapse **24**(1) (2019)
25. Di Fabio, B., Ferri, M.: Comparing persistence diagrams through complex vectors. In: *International Conference on Image Analysis and Processing 2015 Part I*; Editors V. Murino, E. Puppo. vol. LNCS 9279, p. 294–305 (2015)
26. Dindin, M., Umeda, Y., Chazal, F.: Topological data analysis for arrhythmia detection through modular neural networks. In: *Advances in Artificial Intelligence. Canadian AI 2020. Lecture Notes in Computer Science*. vol. 12109. Springer, Cham (2020)
27. Edelsbrunner, H., Harer, J.: *Computational Topology: An Introduction*. American Mathematical Society (2010)
28. Edelsbrunner, H., Letscher, D., Zomorodian, A.: Topological persistence and simplification. *Discrete Comput Geom* **28**, 511–533 (2002)
29. Fasy, B.T., Kim, J., Lecci, F., Maria, C.: Introduction to the r package tda. arXiv preprint arXiv:1411.1830 (2014)
30. Frosini, P.: A distance for similarity classes of submanifolds of a euclidean space. *Bulletin of the Australian Mathematical Society* **42**(3), 407–416 (1990)
31. Frosini, P.: Discrete computation of size functions. *Journal of Combinatorics, Information & System Sciences* **17**(3-4), 232–250 (1992)
32. Frosini, P.: Measuring shapes by size functions. In: Casasent, D.P. (ed.) *Intelligent Robots and Computer Vision X: Algorithms and Techniques*. vol. 1607, pp. 122 – 133. International Society for Optics and Photonics, SPIE (1992)
33. Gabriellsson, R.B., Nelson, B.J., Dwaraknath, A., Skraba, P.: A topology layer for machine learning. In: Chiappa, S., Calandra, R. (eds.) *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research*, vol. 108, pp. 1553–1563. PMLR (2020)

34. Gamble, J., Heo, G.: Exploring uses of persistent homology for statistical analysis of landmark-based shape data. *Journal of Multivariate Analysis* **101**(9), 2184 – 2199 (2010)
35. Ghrist, R.: Barcodes: The persistent topology of data. *Bulletin of the American Mathematical Society* **45**(1), 61–75 (2008)
36. Gidea, M., Katz, Y.: Topological data analysis of financial time series: Landscapes of crashes. *Physica A: Statistical Mechanics and its Applications* **491**, 820 – 834 (2018)
37. Guss, W.H., Salakhutdinov, R.: On characterizing the capacity of neural networks using algebraic topology. *CoRR* **abs/1802.04443** (2018)
38. Hatcher, A.: *Algebraic topology*. Cambridge Univ. Press, Cambridge (2000)
39. Hofer, C., Kwitt, R., Niethammer, M., Uhl, A.: Deep learning with topological signatures. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. p. 1633–1643. NIPS’17, Curran Associates Inc., Red Hook, NY, USA (2017)
40. Hofer, C.D., Kwitt, R., Niethammer, M.: Learning representations of persistence barcodes. *Journal of Machine Learning Research* **20**(126), 1–45 (2019), <http://jmlr.org/papers/v20/18-358.html>
41. Horak, D., Maletić, S., Rajković, M.: Persistent homology of complex networks. *Journal of Statistical Mechanics: Theory and Experiment* **2009**(03), P03034 (mar 2009)
42. Ichinomiya, T., Obayashi, I., Hiraoka, Y.: Protein-folding analysis using features obtained by persistent homology. *Biophysical Journal* **118**(12), 2926 – 2937 (2020)
43. Kim, K., Kim, J., Zaheer, M., Kim, J.S., Chazal, F., Wasserman, L.: Pllay: Efficient topological layer based on persistence landscapes (2020)
44. Kusano, G., Fukumizu, K., Hiraoka, Y.: Persistence weighted gaussian kernel for topological data analysis. In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*. p. 2004–2013. ICML’16, JMLR.org (2016)
45. Le, T., Yamada, M.: Persistence fisher kernel: A riemannian manifold kernel for persistence diagrams. In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. p. 10028–10039. NIPS’18, Curran Associates Inc., Red Hook, NY, USA (2018)
46. Li, C., Ovsjanikov, M., Chazal, F.: Persistence-based structural recognition. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. pp. 2003–2010 (2014)
47. Maria, C., Boissonnat, J.D., Glisse, M., Yvinec, M.: The gudhi library: Simplicial complexes and persistent homology. In: *International Congress on Mathematical Software*. pp. 167–174. Springer (2014)
48. Masood, T.B., Budin, J., Falk, M., Favelier, G., Garth, C., Gueunet, C., Guilou, P., Hofmann, L., Hristov, P., Kamakshidasan, A., et al.: An overview of the topology toolkit. In: *TopoInVis 2019-Topological Methods in Data Analysis and Visualization* (2019)
49. Mischaikow, K., Nanda, V.: Morse theory for filtrations and efficient computation of persistent homology. *Discrete & Computational Geometry* **50**(2), 330–353 (2013)
50. Monasse, P., Guichard, F.: Fast computation of a contrast-invariant image representation. *IEEE Transactions on Image Processing* **9**(5), 860–872 (2000)
51. Morozov, D.: Dionysus, a c++ library for computing persistent homology. <https://www.mrzv.org/software/dionysus/> (2007)
52. Naitzat, G., Zhitnikov, A., Lim, L.H.: Topology of deep neural networks. *Journal of Machine Learning Research* **21**(184), 1–40 (2020)

53. Nanda, V.: Perseus: the persistent homology software. Software available at <http://www.sas.upenn.edu/~vnanda/perseus> (2012)
54. Otter, N., Porter, M.A., Tillmann, U., Grindrod, P., Harrington, H.A.: A roadmap for the computation of persistent homology. *EPJ Data Science* **6**(1), 17 (2017)
55. Pereira, C.M., de Mello, R.F.: Persistent homology for time series and spatial data clustering. *Expert Systems with Applications* **42**(15), 6026 – 6038 (2015)
56. Petri, G., Scalamiero, M., Donato, I., Vaccarino, F.: Topological strata of weighted complex networks. *PLOS ONE* **8**(6), 1–8 (06 2013)
57. Reininghaus, J., Huber, S., Bauer, U., Kwitt, R.: A stable multi-scale kernel for topological machine learning. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 4741–4748 (2015)
58. Reininghaus, J., Huber, S., Bauer, U., Kwitt, R.: A stable multi-scale kernel for topological machine learning. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 4741–4748 (2015)
59. Rieck, B., Togninalli, M., Bock, C., Moor, M., Horn, M., Gumbsch, T., Borgwardt, K.: Neural persistence: A complexity measure for deep neural networks using algebraic topology. In: *International Conference on Learning Representations (ICLR)* (2019)
60. Robins, V., Wood, P., Sheppard, A.: Theory and algorithms for constructing discrete morse complexes from grayscale digital images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **33**(8), 1646–1658 (2011)
61. Singh, G., Memoli, F., Ishkhanov, T., Sapiro, G., Carlsson, G., Ringach, D.: Topological analysis of population activity in visual cortex. *Journal of Vision* **8**(8), 1–18 (2008)
62. Skraba, P., Ovsjanikov, M., Chazal, F., Guibas, L.: Persistence-based segmentation of deformable shapes. In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*. pp. 45–52 (2010)
63. Som, A., Choi, H., Ramamurthy, K.N., Buman, M.P., Turaga, P.: Pi-net: A deep learning approach to extract topological persistence images. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (June 2020)
64. Turner, K., Mukherjee, S., Boyer, D.: Persistent homology transform for modeling shapes and surfaces. *Information and Inference: A Journal of the IMA* **3**(4), 310–344 (2014)
65. Umeda, Y.: Time series classification via topological data analysis. *Transactions of the Japanese Society for Artificial Intelligence* **32**(3), D-G72₁ – –12(2017)
66. Verri, A., Uras, C., Frosini, P., Ferri, M.: On the use of size functions for shape analysis. *Biological Cybernetics* **70**, 99–107 (1993)
67. Xia, K., Wei, G.: Persistent homology analysis of protein structure, flexibility, and folding. *Int J Numer Method Biomed Eng* **30**(8), 814–44 (2014)
68. Xu, Y., Carlinet, E., Géraud, T., Najman, L.: Hierarchical segmentation using tree-based shape spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **39**(3), 457–469 (2017)
69. Zomorodian, A., Carlsson, G.: Computing persistent homology. *Discrete Comput Geom* **3**, 249–274 (2005)