Introduction to the special session on Topological Data Analysis, ICASSP 2016

Harish Chintakunta
Coordinated Science Lab.,
University of Illinois, Urbana Champaign

Michael Robinson
Dept. of Mathematics,
American Univeristy

Hamid Krim
Dept. of Electrical and Computer Engineering,
North Carolina State University

***Abstract*—Topological Data Analysis (TDA) is a topic which has recently seen many applications. The goal of this special session is to highlight the bridge between signal processing, machine learning and techniques in topological data analysis. In this way, we hope to encourage more engineers to start exploring TDA and its applications. This paper briefly introduces the standard techniques used in this area, delineates the common theme connecting the works presented in this session, and concludes with a brief summary of each of the papers presented.**

***Index Terms*—Topology, algebraic topology, data analysis, machine learning, signal processing, network processing, graph theory.**

I. Introduction

The fundamental premise underlying Topological Data Analysis (TDA) is that data has a "shape," and that this shape sometimes conveys meaningful information about the data. In TDA, shape refers to those features which do not change under continuous deformations. For example, consider the objects shown in Figs. (1a), and (1b). If we quantify the shape of these objects by the number holes (the topological features) in them, then they differ in topology even though they may be geometrically similar. On the other hand, the surface of a sphere and that of a bunny as shown in Figs. (1c) and (1d) differ substantially in geometry, but have similar topology since they both are surfaces which enclose a void. Such features may appear too coarse to convey meaningful information, but they are extremely robust. Topological methods therefore trade resolution for an increase in robustness. There are many applications (as seen in the references and the papers presented in this session) where the increase in robustness is more important than loss of resolution, and so benefit from a topological approach.

TDA lies at the intersection of combinatorics, discrete geometry, linear algebra, and of course topology. The combination of these fields yields *algebraic topology*, which is a constructive way to obtain topological summaries. Topological summaries are algebraic objects

(a) (b) (c) (d)

Fig. 1: Figures in (a) and (b) have differ in topology, whereas those in (c) and (d) share the same topology.

such as groups and vector spaces to topological spaces in such a way to remain invariant under continuous changes. Of these, homology spaces in particular are very useful since they can be obtained through numerous computationally efficient algorithms. Indeed, computing homology has not been computationally feasible until the past decade or so. Given a topological space, homology assigns a sequence of spaces H_k, one for each dimension k, called the *homology spaces*, whose dimensions covey the measure of features described above. For example, the first homology spaces – describing non-contractible loops – of Figs. (1a) and (1b) are of dimensions 1 and 2 respectively. We will further discuss homology spaces in Section II.

In the prototypical TDA application, data comes in the form of either point clouds in some Euclidean space, as a set of points with some measure of distance between them, or in the form of a graph or hypergraph. The works presented in this session will cover TDA techniques to deal with all these different forms of input data.

The classical way of describing the shape of a point cloud is by clustering, which is a very mature topic with literally hundreds of papers describing various associated techniques. However, clusters need not be the only shape descriptors for point clouds – indeed, clustering only addresses H_0. One may similarly ask if the points are distributed around in a cyclical structure, or if they are distributed around the surface of a sphere. Performing these tasks is a central theme in TDA techniques. As

described in Section II, the first step is to construct combinatorial objects such as *simplicial complexes*, whose structure is then algebraically expressed. Perea *et al* will present one such application where the presence of harmonics in a time signal are inferred by first converting the signal into a point cloud and then looking for cyclical structures in the point cloud.

The shape of graphs in general is a much more difficult problem. Once again, classical approaches include graph clustering and community detection algorithms. As is in the case of point clouds [12], graph clustering is an ill-posed problem and many clustering algorithm rely on *a priori* knowledge about the nature of clusters or make additional assumptions. Gama *et al* will present a novel method of obtaining a multiscale summary of network clusters using topological signatures, and show its application in clustering handwritten digits from the MNIST database.

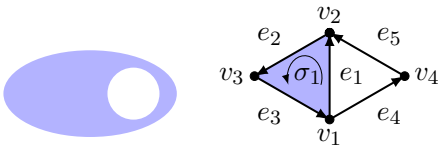
The same TDA techniques which are applied to combinatorial objects derived from point clouds and metric spaces can also be applied to graphs and certain hypergraphs in general, and the interpretation of the resulting topological signatures is specific to the context. Memoli *et al* will present one such application on using network signatures to compare different weighted directed networks, and Fasey *et al* will show how local topological signatures can be used to construct geographical maps by tracking mobile nodes in a region of interest.

In addition to the problem of studying the structure of a network itself, analyzing signals on a network has attracted much attention and is also a very mature topic. One very interesting question in this field is the dependence of signal processing techniques on the topology of the network. Barbarossa *et al* will present some signal processing techniques of simplicial complexes.

We end the introduction by citing recent applications of TDA. The purpose is to illustrate the diversity of topics where topological signatures can be useful, and this is by no means a comprehensive list of existing applications.

Persistent homology, as described in Section II, is one of the most commonly used tool in TDA. It may be viewed as a "higher order" analogue of hierarchical clustering. The reader is directed to two surveys in [10], [7].

As topological methods provide tools for analysis of various data shapes, there are several application avenues in computer vision and image processing. Some of the examples include graphical representation of gray-scale images [19], deformation invariant models for digital images [13], [9], shape segmentation [21] and motion analysis [24].



(a) (b)

Fig. 2: Figure (a) shows a topological space, and (b) shows its simplicial complex representation.

As a further illustration of the scope of this topic, we refer the readers to applications such as comparison of maps [1], graph comparison [4], localization [18], text mining [25] and distributed trees for high performance computing [14], sensor networks [3], [2], [5], and robotics [22].

Various books have recently been published in this area of research, including a great introduction by Edelsbrunner and Harer [6], a concise book by Zomorodian, [26] a more specific book about computational homology by Kaczynski, Mischaikow and Mrozek [11], and a more recent book with a more engineering flavor, by Robinson [17].

II. Background

In this section, we introduce the most commonly used notions in TDA, which also serves to build a context to introduce the papers presented in this session.

A. Simplicial complexes

Simplicial complexes are combinatorial representations of input data, and may be abstractly viewed as hypergraphs that are closed under the operation of taking subsets.

Given a set of vertices $V = \{v_0, v_1, \ldots\}$, a *k-simplex* $\sigma = [v_0, v_1, \ldots, v_k]$ is an ordered list of $k+1$ vertices. We refer to k as the *dimension* of a k-simplex. Any subset (without regard to order) of the $k+1$ vertices forming a simplex is called a *face* of the simplex. Clearly, each face is also a simplex itself once its vertices are given an ordering, which may or may not agree with the ordering from any other simplices. A *simplicial complex* K is a set of simplices such that any simplex in K also has all of its faces in K. (This implies that the intersection of any two simplices σ_1 and σ_2 in K is a face of both σ_1 and σ_2.) Note that if K only contains simplices of dimension $k \leq 1$, K is a graph in the classical sense. There are two main advantages of simplicial complexes, 1) they provide a discrete and

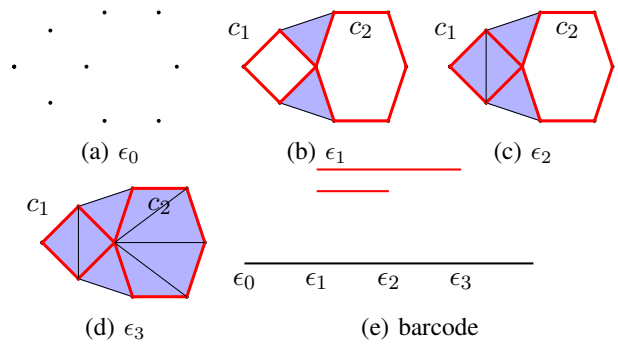


Fig. 3: Figure illustrates the persistent homology computation given a point cloud. (a) shows a point cloud with two cyclic structures, which are captured by the birth and death of the cycles c_1 and c_2. Both the cycles are born at (b), c_1 dies in (c) and c_2 dies in (d). The lifetime of each cycle indicated the size of the cyclic structure it corresponds to.

simple representation of continuous topological spaces (as in Fig. (2)), and 2) the combinatorial-topological structure is limited enough that it can be algebraically represented using linear operators.

B. Homology spaces

Given a simplicial complex K, let C_k be abstract vector space whose basis is the set of k-simplices. (Each k-simplex is therefore thought of as a basis vector.) Therefore C_0 is built using vertices (0-simplices), C_1 using edges (1-simplices) and so on. The *boundary operators* $\partial_k : C_k \to C_{k-1}$ are functions that algebraically extract boundaries of each simplex. Each ∂_k is given as $\partial_k \sigma = \sum_{i=0}^{k} (-1)^i [v_0, \ldots, \hat{v}_i, \ldots, v_k]$, where $\hat{v}_i$ means that vertex v_i is deleted from the list.

For example, in Fig. (2b), $\partial_1(e_1) = v_2 - v_1$, and $\partial_2(\sigma_1) = e_1 + e_2 + e_3$. Cycles which are also boundaries (eg. $e_1 + e_2 + e_3$) are considered to be "filled in". Therefore, the cycles which cannot be expressed as sums of boundaries (eg. $e_4 + e_5 - e_1$) will surround holes. This is the intuition behind homology spaces, which are defined as the quotient space $H_k(K) = \ker(\partial_k)/\text{image}(\partial_{k+1})$. In Fig. (2b), H_1 is generated by $e_4 + e_5 - e_1$, which is a one dimensional space, indicating the presence of a single hole.

C. Persistence

Given a point cloud and no specifically predefined scale, it is well known that the clustering problem is ill-posed [12] and unstable. It is therefore common practice under these circumstances to perform hierarchical clustering and provide a multi-scale summaries in the form of dendrograms. In hierarchical clustering, one first produces a sequence of spaces $K_0 \subseteq K_1 \subseteq \cdots \subseteq K_m = K$ and tracks how the connected components merge together.

Persistent homology may be viewed as analogous to hierarchical clustering for other topological features such as holes, voids etc. Since a formal introduction is beyond the scope of this introductory paper, we use the example point cloud shown in Fig. (3a) to provide a brief description of the process.

In the case of clustering, hierarchical clustering follows the merging of connected components in the graph as the scale is increased. Analogously, one can follow the "birth" and "death" of *cycles*.

As in the case of clustering, we produce a sequence of spaces by increasing the value of a parameter $\epsilon > 0$, and for each ϵ value, we add all the edges in the Delaunay triangulation of the point cloud whose length is less than or equal to ϵ to obtain K_ϵ. If a cycle c – a nontrivial element of H_1 – appears for the first time at ϵ_i, we say that c is *born* at ϵ_i. We say that c *dies* at ϵ_j if j is the first time when c can be expressed in terms of boundaries of triangle and is therefore trivial in H_1.

In Fig. (3), at ϵ_1, we see the birth of two cycles in red: c_1 and c_2. We see that c_1 dies at ϵ_2, and c_2 dies at ϵ_3, resulting in the barcode shown in Fig. (3e). It is straightforward to infer from the barcode that the point cloud exhibits two cyclic structures, one of which is larger than the other. For the sake of concreteness, we will now provide some algebraic background which gives rise to these bars.

For a triangulation K_ϵ, as in Fig. (3), the basis elements of $H_1(K\epsilon)$ are equivalence classes of cycles in K_ϵ. Each cycle is an element of the kernel of the boundary map ∂_1. For example, the equivalence class of $[c_1] \neq 0$ in K_{ϵ_1}, whereas $[c_1] = 0$ in K_{ϵ_2}. Also, since ϵ_1 is the smallest value for which $[c_1] = 0$, we say that $[c_1]$ *persists* in the interval $[\epsilon_1, \epsilon_2)$, and this is represented as bar from ϵ_1 to ϵ_2 in the output. More generally, the homology which persists in the interval $[a, b]$ is given by image $\left(H_1(K_a) \xrightarrow{i_*} H_1(K_b)\right)$, where i_* is the map induced by the inclusion $i : K_a \to K_b$. In other words, the number of bars in any interval $[a, b]$ in the barcode is equal to the dimension of image $\left(H_1(K_a) \xrightarrow{i_*} H_1(K_b)\right)$.

From a data analysis perspective, each bar of interval $[a, b]$ in the barcode corresponds to a cyclic structure which is present in the triangulations corresponding to thresholds in that interval.

III. Applications

In this section we give a brief description of the papers presented in this session, which fall into three broad categories, 1) signal processing, 2) unsupervised learning and 3) signatures for networks and metric spaces.

A. Signal processing

Many classical signal processing techniques rely on transforming the time domain signal into some other domain (fourier, wavelet, etc.). Recently, another perspective has come to light [20] where one converts a time signal into a point cloud using delay embedding, or more generally, sliding window embedding. Some properties of the signal such as periodicity or quasi-periodicity show up in the resulting point cloud as cyclic structure which can then be analyzed using persistent homology as described in Section II. The advantages of such a procedure is that it reduces the complexity for certain applications, and increases robustness [8], [16], [15].

In the paper "Persistent homology of toroidal sliding window embeddings", Perea introduces the topological analysis of signals exhibiting quasi-periodic behavior. He studies the persistent homology of sliding window embeddings for sums of harmonics with incommensurate frequencies, in which case the resulting sliding window point-clouds are (dense in) high-dimensional tori. He also proves theorems which guide the choice of window size and embedding dimension, and describe the associated persistent homology.

The paper " Uncertainty Principle and Sampling on a Graph of arbitrary Topology" looks at another fundamental question of signal processing, that of sampling principles for signals on arbitrary domains (as opposed to linear time domains in classical signal processing, or regular grids as in image processing). As described in Section II, simplicial complexes can be used to represent a very broad set of spaces. Building on their work on sampling theorems for graphs [23], Barbarossa *et al* develop sampling theorems for signals on simplicial complexes.

B. Unsupervised learning

In the paper "Overlapping clustering of networked data using cut metrics", Gama *et al* present a novel method to hierarchically cluster networked data, i.e. a set of nodes endowed with a pairwise dissimilarity function, allowing nodes to simultaneously belong to multiple clusters. Traditional clustering algorithms output a partition of the node set such that a node belongs to exactly one subset or cluster. However, in many situations there are nodes that are hard to classify in any given cluster, or that might legitimately belong to more than one category due to having strong similarities with multiple groups. They accommodate for these situations by proposing a method to obtain a nested collection of overlapping clusters. More specifically, given a network, their method outputs a cut metric on the underlying node set, which can be related to data coverings at different resolutions. The cut metric is obtained by averaging a set of ultrametrics, which are themselves the output of (non-overlapping) hierarchically clustering noisy versions of the original network of interest. The resulting algorithm is applied to three synthetic networks as well as to the problem of clustering handwritten digits from the MNIST database.

Manifold learning is another major part of unsupervised learning, where the primary assumption is that the data is sampled (with noise) from an underlying manifold. It is quite possible that this assumption is not satisfied in practice, and dealing with more complicated scenarios such as stratified manifolds is challenging. In "TBD", Fasy *et al* show how a variation of persistent homology, called the *local persistent homology* can be used to classify data points into different strata.

C. Signatures for networks and metric spaces

Networks which show the relationships within and between complex systems are key tools in a variety of current scientific areas. A central aim in network analysis is to find a suitable metric for network similarity and comparison. In "Distances between directed networks and applications", Memoli *et al* present their work on 1) defining notions of dissimilarity between directed weighted networks, 2) using this distance for studying the network reconstruction problem from partial measurements, 3) computing succinct summaries/invariants of the networks, like hierarchical clustering dendrograms and statistics over subnetworks, and 4) computing estimates of the proposed distance between two networks via the comparison of respective network signatures – a task that often leads to computationally simpler problems.

From an another perspective of comparing networks, Fasy *et al* use local persistent homology to define a local distance between road networks. One can then integrate this distance to get a global picture of the distance, or can plot the distances like an image in order to determine where the differences between the two graphs lie.

REFERENCES

[1] Mahmuda Ahmed, Brittany Terese Fasy, and Carola Wenk. Local persistent homology based distance between maps. *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 43–52, 2014.

[2] Harish Chintakunta and Hamid Krim. Distributed localization of coverage holes using topological persistence. *Signal Processing, IEEE Transactions on*, 62(10):2531–2541, May 2014.

[3] Vin De Silva and Robert Ghrist. Homological sensor networks. *Notices of the American Mathematical Society*, 54, 2007.

[4] Tamal Dey, Dayu Shi, and Yusu Wang. Comparing graphs via persistence distortion. *arXiv preprint arXiv:1503.07414*, 2015.

[5] Pawel Dłotko, Robert Ghrist, Mateusz Juda, and Marian Mrozek. Distributed computation of coverage in sensor networks by homological methods. *Applicable Algebra in Engineering, Communication and Computing*, 23(1-2):29–58, 2012.

[6] Herbert Edelsbrunner and John L. Harer. *Computational Topology*. American Mathematical Society, Providence, RI, 2010. An introduction.

[7] Herbert Edelsbrunner and Dmitriy Morozov. Persistent homology: Theory and practice. *Proceeding of the European Congress of Mathematics*, 2014.

[8] Saba Emrani, Harish Chintakunta, and Hamid Krim. Real time detection of harmonic structure: A case of topological signal analysis. *ICASSP, Signal processing techniques and methods*, May 2014.

[9] Jan Ernst, Maneesh Kumar Singh, and Visvanathan Ramesh. Discrete texture traces: Topological representation of geometric context. *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 422–429, 2012.

[10] Robert Ghrist. Barcodes: the persistent topology of data. *Bull. Amer. Math. Soc. (N.S.)*, 45(1):61–75, 2008.

[11] Tomasz Kaczynski, Konstantin Mischaikow, and Marian Mrozek. *Computational Homology*. Springer, 2004.

[12] Jon Kleinberg. An impossibility theorem for clustering. *Advances in neural information processing systems*, pages 463–470, 2003.

[13] Loïc Mazo, Nicolas Passat, Michel Couprie, and Christian Ronse. Digital imaging: A unified topological framework. *Journal of Mathematical Imaging and Vision*, 44(1):19–37, 2012.

[14] Dmitriy Morozov and Gunther Weber. Distributed merge trees. *ACM SIGPLAN Notices*, 48(8):93–102, 2013.

[15] Jose A Perea, Anastasia Deckard, Steve B Haase, and John Harer. Sw1pers: Sliding windows and 1-persistence scoring; discovering periodicity in gene expression time series data. *BMC bioinformatics*, 16(1):257, 2015.

[16] JoseA. Perea and John Harer. Sliding windows and persistence: An application of topological methods to signal analysis. *Foundations of Computational Mathematics*, 15(3):799–838, 2015.

[17] Michael Robinson. *Topological Signal Processing*. Mathematical Engineering. Springer, 2014.

[18] Michael Robinson and Robert Ghrist. Topological localization via signals of opportunity. *Signal Processing, IEEE Transactions on*, 60(5):2362–2373, 2012.

[19] Peter Saveliev. A graph, non-tree representation of the topology of a gray scale image. *IS&T/SPIE Electronic Imaging, International Society for Optics and Photonics*, 2011.

[20] Primoz Skraba, Vin de Silva, and Mikael Vejdemo-Johansson. Topological analysis of recurrent systems. In *NIPS 2012 Workshop on Algebraic Topology and Machine Learning, December 8th, Lake Tahoe, Nevada*, pages 1–5, 2012.

[21] Primoz Skraba, Maks Ovsjanikov, Frederic Chazal, and Leonidas Guibas. Persistence-based segmentation of deformable shapes. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 45–52, 2010.

[22] Alireza Tahbaz-Salehi and Ali Jadbabaie. Distributed coverage verification in sensor networks without location information. *Automatic Control, IEEE Transactions on*, 55(8):1837–1849, 2010.

[23] Mikhail Tsitsvero, Sergio Barbarossa, and Paolo Di Lorenzo. Signals on graphs: Uncertainty principle and sampling. *arXiv preprint arXiv:1507.08822*, 2015.

[24] Mikael Vejdemo-Johansson, Florian T. Pokorny, Primoz Skraba, and Danica Kragic. Cohomological learning of periodic motion. *Applicable Algebra in Engineering, Communication and Computing*, 26(1-2):5–26, 2015.

[25] Hubert Wagner, Paweł Dłotko, and Marian Mrozek. Computational topology in text mining. *Computational Topology in Image Context*, pages 68–78, 2012.

[26] Afra J. Zomorodian. *Topology for Computing*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2005.