University of Oxford

Topological Data Analysis of Temporal Networks

by

Dimitri Lozeve

St Catherine's College

A dissertation submitted in partial fulfilment of the degree of
Master of Science in Applied Statistics

Department of Statistics,
24–29 St Giles, Oxford, OX1 3LB

September 2018

Declaration of authorship

This my own work (except where otherwise indicated).

Date Signature

Abstract

Abstract here

Acknowledgements

Thank you!

Contents

List of Figures

1 Introduction

2 *Graphs and Temporal Networks*

2.1 Definition and basic properties

In this section, we introduce the notion of temporal networks (or temporal graphs). This is a complex notion, with many concurrent definitions and interpretations.

After clarifying the notations, we restate the standard definition of a non-temporal graph.

Notation.
- $\mathbb{N}$ is the set of non-negative natural numbers $0, 1, 2, \ldots$
- $\mathbb{N}^*$ is the set of positive integers $1, 2, \ldots$
- $\mathbb{R}$ is the set of real numbers. $\mathbb{R}_+ = \{x \in \mathbb{R} \mid x \geq 0\}$, and $\mathbb{R}_+^* = \{x \in \mathbb{R} \mid x > 0\}$.

Definition 2.1 (Graph)**.** A *graph* is a couple $G = (V, E)$, where V is a set of *nodes* (or *vertices*), and $E \subseteq V \times V$ is a set of *edges*. A *weighted graph* is defined by $G = (V, E, w)$, where $w : E \mapsto \mathbb{R}_+^*$ is called the *weight function*.

We also define some basic concepts that we will need later to build simplicial complexes on graphs.

Definition 2.2 (Clique)**.** A *clique* is a set of nodes where each pair is adjacent. That is, a clique C of a graph $G = (V, E)$ is a subset of V such that for all $i, j \in C, i \neq j \implies (i, j) \in E$. A clique is said to be *maximal* if it cannot be augmented by any node, such that the resulting set of nodes is itself a clique.

Temporal networks can be defined in the more general framework of *multilayer networks* [15]. However, this definition is much too general for our simple applications, and we restrict ourselves to edge-centric time-varying graphs [9]. In this model, the set of nodes is fixed, but edges can appear or disappear at different times.

In this study, we restrict ourselves to discrete time stamps. Each interaction is taken to be instantaneous.

Definition 2.3 (Temporal network)**.** A *temporal network* is a tuple $G = (V, E, \mathcal{T}, \rho)$, where:
- V is a set of nodes,
- $E \subseteq V \times V$ is a set of edges,
- $\mathbb{T}$ is the *temporal domain* (often taken as $\mathbb{N}$ or any other countable set), and $\mathcal{T} \subseteq \mathbb{T}$ is the *lifetime* of the network,
- $\rho : E \times \mathcal{T} \mapsto \{0, 1\}$ is the *presence function*, which determines whether an edge is present in the network at each time stamp.

The *available times* of an edge are the set $\mathcal{I}(e) = \{t \in \mathcal{T} : \rho(e, t) = 1\}$.

Temporal networks can also have weighted edges. In this case, it is possible to have constant weights (edges can only appear or disappear over time, and always have the same weight), or time-varying weights. In the latter case, we can set the domain of the presence function to be $\mathbb{R}_+$ instead of $\{0, 1\}$, where by convention a 0 weight corresponds to an absent edge.

Definition 2.4 (Additive and dismantling temporal networks)**.** A temporal network is said to be *additive* if for all $e \in E$ and $t \in \mathcal{T}$, if $\rho(e,t) = 1$, then for all $t' > t, \rho(e,t') = 1$. An additive network can only gain edges over time.

A temporal network is said to be *dismantling* if for all $e \in E$ and $t \in \mathcal{T}$, if $\rho(e,t) = 0$, then for all $t' > t, \rho(e,t') = 0$. An dismantling network can only lose edges over time.

2.2 Examples of applications

2.3 Network partitioning

Temporal networks are a very active research subject, leading to multiple interesting problems. The additional time dimension adds a significant layer of complexity that cannot be adequately treated by the common methods on static graphs.

Moreover, data collection can lead to large amount of noise in datasets. Combined with large dataset sized due to the huge number of data points for each node in the network, temporal graphs cannot be studied effectively in their raw form. Recent advances have been made to fit network models to rich but noisy data [21], generally using some variation on the expectation-maximization (EM) algorithm.

One solution that has been proposed to study such temporal data has been to *partition* the time scale of the network into a sequence of smaller, static graphs, representing all the interactions during a short interval of time. The approach consists in subdividing the lifetime of the network in *sliding windows* of a given length. We can then "flatten" the temporal network on each time interval, keeping all the edges that appear at least once (or adding their weights in the case of weighted networks).

This partitioning is sensitive to two parameters: the length of each time interval, and their overlap. Of those, the former is the most important: it will define the *resolution* of the study. If it is too small, too much noise will be taken into account; if it is too large, we will lose important information. There is a need to find a compromise, which will depend on the application and on the task performed on the network. In the case of a classification task to determine periodicity, it will be useful to adapt the resolution to the expected period: if we expect week-long periodicity, a resolution of one day seems reasonable.

Once the network is partitioned, we can apply any statistical learning task on the sequence of static graphs. In this study, we will focus on classification of time steps. This can be used to detect periodicity, outliers, or even maximise temporal communities.

3 Topological Data Analysis and Persistent Homology

3.1 Basic constructions

3.1.1 Homology

Our goal is to understand the topological structure of a metric space. For this, we can use *homology*, which consists of associating a vector space $H_i(X)$ to a metric space X and a dimension i. The dimension of $H_i(X)$ gives us the number of i-dimensional components in X: the dimension of $H_0(X)$ is the number of path-connected components in X, the dimension of $H_1(X)$ is the number of holes in X, and the dimension of $H_2(X)$ is the number of voids.

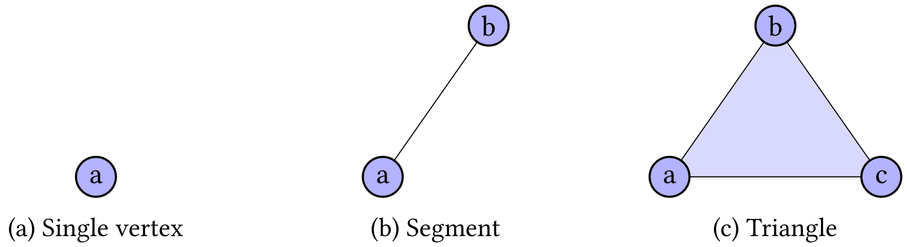
Crucially, these vector spaces are robust to continuous deformation of the underlying metric space (they are *homotopy invariant*). However, computing the homology of an arbitrary metric space can be extremely difficult. It is necessary to approximate it in a structure that would be both combinatorial and topological in nature.

3.1.2 Simplicial complexes

To understand the topological structure of a metric space, we need a way to decompose it in smaller pieces that, when assembled, conserve the overall organisation of the space. For this, we use a structure called a *simplicial complex*, which is a kind of higher-dimensional generalization of a graph.

The building blocks of this representation is the *simplex*, which is the convex hull of an arbitrary set of points. Examples of simplices include single points, segments, triangles, and tetrahedrons (in dimensions 0, 1,, 2, and 3 respectively).

Definition 3.1 (Simplex)**.** A *k-dimensional simplex* $\sigma = [x_0, \ldots, x_k]$ is the convex hull of the set $\{x_0, \ldots, x_k\} \in \mathbb{R}^d$, where $x_0, \ldots, x_k$ are affinely independent. $x_0, \ldots, x_k$ are called the *vertices* of σ, and the simplices defined by the subsets of $\{x_0, \ldots, x_k\}$ are called the *faces* of σ.



(a) Single vertex (b) Segment (c) Triangle

Figure 31: Examples of simplices

We then need a way to meaningfully combine these basic building blocks so that the resulting object can adequately reflect the topological structure of the metric space.

Definition 3.2 (Simplicial complex)**.** A *simplicial complex* is a collection K of simplices such that:

- any face of a simplex of K is a simplex of K
- the intersection of two simplices of K is either the empty set, or a common face, or both.

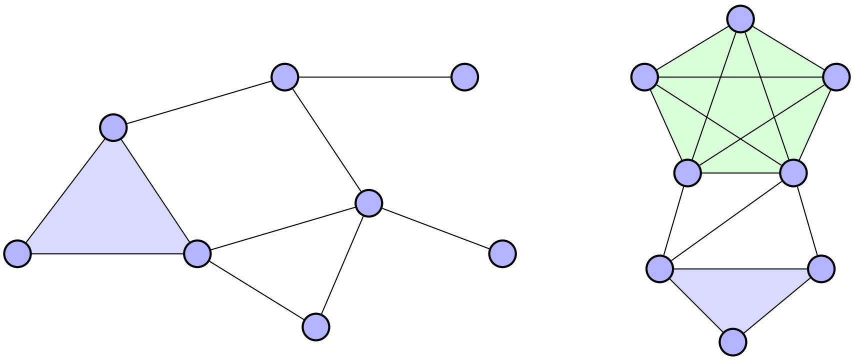


Figure 32: Example of a simplicial complex that has two connected components, two 3-simplices, and one 5-simplex.

The notion of simplicial complex is closely related to that of a hypergraph. One important distinction lies in the fact that a subset of a hyperedge is not necessarily a hyperedge itself.

Using these definitions, we can define homology on simplicial complexes.

3.1.3 Filtrations

If we consider that a simplicial complex is a kind of "discretization" of a subset of a metric space, we realise that there must be an issue of *scale*. For our analysis to be invariant under small perturbations in the data, we need a way to find the optimal scale parameter to capture the adequate topological structure, without taking into account some small perturbations, nor ignoring some important smaller features.

One possible solution to these problems is to consider all scales at once. This is the objective of *filtered simplicial complexes*.

Definition 3.3 (Filtration)**.** A *filtered simplicial complex*, or simply a *filtration*, K is a sequence $(K_i)_{i\in I}$ of simplicial complexes such that:

- for any $i, j \in I$, if $i < j$ then $K_i \subseteq K_j$,
- $\bigcup_{i\in I} K_i = K$.

3.2 Persistent Homology

We can now compute the homology for each step in a filtration. This leads to the notion of *persistent homology* [5, 30], which gives all the information necessary to establish the topological structure of a metric space at multiple scales.

Definition 3.4 (Persistent homology)**.** The *p-th persistent homology* of a simplicial complex $K = (K_i)_{i\in I}$ is the pair $(\{H_p(K_i)\}_{i\in I}, \{f_{i,j}\}_{i,j\in I, i\leq j})$, where for all $i \leq j$, $f_{i,j} : H_p(K_i) \mapsto H_p(K_j)$ is induced by the inclusion map $K_i \mapsto K_j$.

The functions $f_{i,j}$ allow one to link generators in each successive homology space in a filtration. Because each generator corresponds to a topological feature (connected component, hole, void, and so on, depending on the dimension p), we can determine whether it survives in the next step of the filtration. We can also determine when each feature is born and when it dies (if it dies at all). The couples of intervals (birth time, death time) depends on the choice of basis for each homology space $H_p(K_i)$. However, by the Fundamental Theorem of Persistent Homology [30], we can choose basis vectors in each homology space such that the collection of half-open intervals is well-defined and unique. This construction is called a *barcode* [5].

3.3 Topological summaries: barcodes and persistence diagrams

To interpret the results of the persistent-homology computation, we need to compare the output for a particular data set to a suitable null model. For this, we need some kind of similarity measure between barcodes and a way to evaluate the statistical significance of the results.

One possible approach is to define a space in which we can project barcodes and study their geometric properties. One such space is the space of *persistence diagrams* [13].

Definition 3.5 (Multiset)**.** A *multiset* M is the couple (A, m), where A is the *underlying set* of M, formed by its distinct elements, and $m : A \mapsto \mathbb{N}^*$ is the *multiplicity function* giving the number of occurrences of each element of A in M.

Definition 3.6 (Persistence diagrams)**.** A *persistence diagram* is the union of a finite multiset of points in $\overline{\mathbb{R}}^2$ with the diagonal $\Delta = \{(x, x) \mid x \in \mathbb{R}^2\}$, where every point of Δ has infinite multiplicity.

One adds the diagonal Δ for technical reasons. It is convenient to compare persistence diagrams by using bijections between them, so persistence diagrams must have the same cardinality.

In some cases, the diagonal in the persistence diagrams can also facilitate comparisons between diagrams, as points near the diagonal correspond to short-lived topological features, so they are likely to be caused by small perturbations in the data.

One can build a persistence diagram from a barcode by taking the union of the multiset of (birth, death) couples with the diagonal Δ. Figure 33 summarises the entire pipeline.

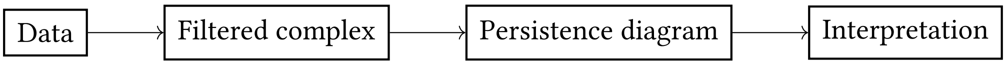


Figure 33: Persistent homology pipeline

One can define an operator dgm as the first two steps in the pipeline. It constructs a persistence diagram from a subset of a metric space, via persistent homology on a filtered complex.

We can now define several distances on the space of persistence diagrams.

Definition 3.7 (Wasserstein distance)**.** The *p-th Wasserstein distance* between two diagrams X and Y is

$$W_p[d](X, Y) = \inf_{\phi : X \mapsto Y} \left[\sum_{x \in X} d\left(x, \phi(x)\right)^p \right]$$

for $p \in [1, \infty)$, and:

$$W_\infty[d](X, Y) = \inf_{\phi : X \mapsto Y} \sup_{x \in X} d\left(x, \phi(x)\right)$$

for $p = \infty$, where d is a distance on $\mathbb{R}^2$ and ϕ ranges over all bijections from X to Y.

Definition 3.8 (Bottleneck distance)**.** The *bottleneck distance* is defined as the infinite Wasserstein distance where d is the uniform norm: $d_B = W_\infty[L_\infty]$.

The bottleneck distance is symmetric, non-negative, and satisfies the triangle inequality. However, it is not a true distance, as one can come up with two distinct diagrams with bottleneck distance 0, even on multisets that do not touch the diagonal Δ.

3.4 Stability

One of the most important aspects of topological data analysis is that it is *stable* with respect to small perturbations in the data. More precisely, the second step of the pipeline in Figure 33 is Lipschitz with respect to a suitable metric on filtered complexes and the bottleneck distance on persistence diagrams [10, 11]. First, we define a distance between subsets of a metric space [23].

Definition 3.9 (Hausdorff distance)**.** Let X and Y be subsets of a metric space (E, d). The *Hausdorff distance* is defined by

$$d_H(X, Y) = \max \left[\sup_{x \in X} \inf_{y \in Y} d(x, y), \sup_{y \in Y} \inf_{x \in X} d(x, y) \right].$$

We can now give an appropriate stability property [10, 11].

Proposition 3.1. *Let X and Y be subsets in a metric space. We have*

$$d_B(\mathrm{dgm}(X), \mathrm{dgm}(Y)) \leq d_H(X, Y).$$

3.5 Algorithms and implementations

[3, 18, 19, 25]

3.6 Discussion

4 Topological Data Analysis on Networks

4.1 Persistent homology for networks

We now consider the problem of applying persistent homology to network data. An undirected network is already a simplicial complex of dimension 1. However, this is not sufficient to capture enough topological information; we need to introduce higher-dimensional simplices. One method is to project the nodes of a network onto a metric space [22], thereby transforming the network data into a point-cloud data. For this, we need to compute the distance between each pair of nodes in the network (e.g. with the shortest-path distance). This also requires the network to be connected.

Another common method, for weighted networks, is called the *weight rank-clique filtration* (WRCF) [24], which filters a network based on weights. The procedure works as follows:

1. Consider the set of all nodes, without any edge, to be filtration step 0.
2. Rank all edge weights in decreasing order $\{w_1, \dots, w_n\}$.
3. At filtration step t, keep only the edges whose weights are larger than or equal to w_t, thereby creating an unweighted graph.
4. Define the maximal cliques of the resulting graph to be simplices.

At each step of the filtration, we construct a simplicial complex based on cliques; this is called a *clique complex* [29]. The result of the algorithm is itself a filtered simplicial complex (definition 3.3), because a subset of a clique is necessarily a clique itself, and the same is true for the intersection of two cliques.

This leads to a first possibility for applying persistent homology to temporal networks. It is possible to segment the lifetime of a network into sliding windows, creating a time-independent graph on each window by retaining only the edges available during the time interval. We can then apply WRCF on each graph in the sequence, obtaining a filtered complex for each window, to which we can then apply persistent homology.

This method can quickly become very computationally expensive, as finding all maximal cliques (e.g. using the Bron–Kerbosch algorithm) is a complicated problem, with an optimal computational complexity of $\mathcal{O}(3^{n/3})$ [27]. In practice, one often restrict the search to cliques of dimension less than or equl to a certain bound d_M. With this restriction, the new simplicial complex is homologically equivalent to the original: they have the same homology groups up to H_{d_M-1}.

This method is sensitive to the choice of sliding windows on the time scale. The width and the overlap of the windows can completely change the networks created and their topological features. Too small a window, and the network becomes too small to have any significant topological properties, too large, and we lose important information in the evolution of the network over time.

4.2 Zigzag persistence

The standard algorithm to compute persistent homology (see section 3.2) relies on the fact that filtrations (see definition 3.3) are nested sequences of simplicial complexes:

$$\cdots \subseteq K_{i-1} \subseteq K_i \subseteq K_{i+1} \subseteq \cdots$$

One can now create an independent filtration (e.g. with WRCF) for each time step. The issue is that the topological features will be orthogonal to the time dimension.

Another possibility is to create a filtration along the time dimension. The issue in this case is that the sequence is no longer nested (except for additive or dismantling temporal networks, see definition 2.4).

The solution to consider the time dimension is provided by *zigzag persistence* [6], which allows one to compute persistence on alternating nested sequences:

$$\cdots \supseteq K_{i-1} \subseteq K_i \supseteq K_{i+1} \subseteq \cdots$$

This sequence can in turn be computed from a temporal network by computing the union of each pair of consecutive time steps, constructing an alternating sequence.

Zigzag persistence is a special case of the more general concept of *multi-parameter persistence* [7, 12], where filtrations can encompass multiple parameters.

5 Persistent Homology for Machine-Learning Applications

The output of persistent homology is not directly usable by most statistical methods. For example, barcodes and persistence diagrams, which are multisets of points in $\overline{\mathbb{R}}^2$, are not elements of a metric space in which one can perform statistical computations.

The distances between persistence diagrams defined in section 3.3 allow one to compare different outputs. From a statistical perspective, it is possible to use a generative model of simplicial complexes and to use a distance between persistence diagrams to measure the similarity of our observations with this null model [2]. This would effectively define a metric space of persistence diagrams. It is even possible to define some statistical summaries (means, medians, confidence intervals) on these spaces [20, 28].

The issue with this approach is that metric spaces do not offer enough algebraic structure to be amenable to most machine-learning techniques. One of the most recent development in the study of topological summaries has been to find mappings between the space of persistence diagrams and Banach spaces.

5.1 Vectorization methods

5.1.1 Persistence landscapes

Persistence landscapes [4] give a way to project barcodes to a space where it is possible to add them meaningfully. It is then possible to define means of persistence diagrams, as well as other summary statistics.

The function mapping a persistence diagram to a persistence landscape is *injective*, but no explicit inverse exists to go back from a persistence landscape to the corresponding persistence diagram. Moreover, a mean of persistence landscapes does not necessarily have a corresponding persistence diagram.

Definition 5.1 (Persistence landscape)**.** The persistence landscape of a diagram $D = \{(b_i, d_i)\}_{i=1}^n$ is the set of functions $\lambda_k : \mathbb{R} \mapsto \mathbb{R}$, for $k \in \mathbb{N}$, such that

$$\lambda_k(x) = k\text{-th largest value of } \{f_{(b_i,d_i)}(x)\}_{i=1}^n,$$

(and $\lambda_k(x) = 0$ if the k-th largest value does not exist), where $f_{(b,d)}$ is a piecewise-linear function defined by:

$$f_{(b,d)} = \begin{cases} 0, & \text{if } x \notin (b, d), \\ x - b, & \text{if } x \in (b, \frac{b+d}{2}), \\ -x + d, & \text{if } x \in (\frac{b+d}{2}, d)\,. \end{cases}$$

Moreover, one can show that persistence landscapes are stable with respect to the L^p distance, and that the Wasserstein and bottleneck distances are bounded by the L^p distance [4]. We can thus view the landscapes as elements of a Banach space in which we can perform the statistical computations.

5.1.2 Persistence images

[1]

5.1.3 Tropical and arctic semirings

[14]

5.2 Kernel-based methods

5.2.1 Persistent scale-space kernel

[17, 26]

5.2.2 Persistence weighted-Gaussian kernel

[16]

5.2.3 Sliced Wasserstein kernel

[8]

5.3 Comparison

6 Conclusions

Bibliography

[1] Henry Adams et al. "Persistence Images: A Stable Vector Representation of Persistent Homology". In: *Journal of Machine Learning Research* 18.8 (2017), pp. 1–35 (cit. on p. 11).

[2] Robert J. Adler et al. *Persistent homology for random fields and complexes*. Institute of Mathematical Statistics, 2010. ISBN: 978-0-940600-79-9. DOI: 10.1214/10-IMSCOLL609 (cit. on p. 10).

[3] Ulrich Bauer. *ripser: Ripser: a lean C++ code for the computation of Vietoris–Rips persistence barcodes*. original-date: 2015-10-27T21:43:59Z. Apr. 3, 2018 (cit. on p. 7).

[4] Peter Bubenik. "Statistical Topological Data Analysis using Persistence Landscapes". In: *Journal of Machine Learning Research* 16 (2015), pp. 77–102 (cit. on p. 10).

[5] Gunnar Carlsson. "Topology and data". In: *Bulletin of the American Mathematical Society* 46.2 (Jan. 29, 2009), pp. 255–308. ISSN: 0273-0979. DOI: 10.1090/S0273-0979-09-01249-X (cit. on pp. 5, 6).

[6] Gunnar Carlsson, Vin de Silva, and Dmitriy Morozov. "Zigzag Persistent Homology and Real-valued Functions". In: *Proceedings of the Twenty-fifth Annual Symposium on Computational Geometry*. SCG '09. New York, NY, USA: ACM, 2009, pp. 247–256. ISBN: 978-1-60558-501-7. DOI: 10.1145/1542362.1542408 (cit. on p. 9).

[7] Gunnar Carlsson and Afra Zomorodian. "The Theory of Multidimensional Persistence". In: *Discrete & Computational Geometry* 42.1 (July 1, 2009), pp. 71–93. ISSN: 0179-5376, 1432-0444. DOI: 10.1007/s00454-009-9176-0 (cit. on p. 9).

[8] Mathieu Carrière, Marco Cuturi, and Steve Oudot. "Sliced Wasserstein Kernel for Persistence Diagrams". In: *International Conference on Machine Learning*. International Conference on Machine Learning. July 17, 2017, pp. 664–673 (cit. on p. 11).

[9] Arnaud Casteigts et al. "Time-varying graphs and dynamic networks". In: *International Journal of Parallel, Emergent and Distributed Systems* 27.5 (Oct. 1, 2012), pp. 387–408. ISSN: 1744-5760. DOI: 10.1080/17445760.2012.668546 (cit. on p. 2).

[10] Frédéric Chazal, Vin de Silva, and Steve Oudot. "Persistence stability for geometric complexes". In: *Geometriae Dedicata* 173.1 (Dec. 1, 2014), pp. 193–214. ISSN: 0046-5755, 1572-9168. DOI: 10.1007/s10711-013-9937-z (cit. on p. 7).

[11] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. "Stability of Persistence Diagrams". In: *Discrete & Computational Geometry* 37.1 (Jan. 1, 2007), pp. 103–120. ISSN: 0179-5376, 1432-0444. DOI: 10.1007/s00454-006-1276-5 (cit. on p. 7).

[12] Tamal K. Dey, Fengtao Fan, and Yusu Wang. "Computing Topological Persistence for Simplicial Maps". In: *Proceedings of the Thirtieth Annual Symposium on Computational Geometry*. SOCG'14. New York, NY, USA: ACM, 2014, 345:345–345:354. ISBN: 978-1-4503-2594-3. DOI: 10.1145/2582112.2582165 (cit. on p. 9).

[13] Herbert Edelsbrunner and J. Harer. *Computational topology: an introduction.* OCLC: ocn427757156. Providence, R.I: American Mathematical Society, 2010. 241 pp. ISBN: 978-0-8218-4925-5 (cit. on p. 6).

[14] Sara Kališnik. "Tropical Coordinates on the Space of Persistence Barcodes". In: *Foundations of Computational Mathematics* (Jan. 30, 2018), pp. 1–29. ISSN: 1615-3375, 1615-3383. DOI: 10.1007/s10208-018-9379-y (cit. on p. 11).

[15] Mikko Kivelä et al. "Multilayer Networks". In: *Journal of Complex Networks* 2.3 (Sept. 1, 2014), pp. 203–271. ISSN: 2051-1310, 2051-1329. DOI: 10.1093/comnet/cnu016. arXiv: 1309.7233 (cit. on p. 2).

[16] Genki Kusano, Kenji Fukumizu, and Yasuaki Hiraoka. "Kernel method for persistence diagrams via kernel embedding and weight factor". In: *arXiv:1706.03472 [physics, stat]* (June 12, 2017). arXiv: 1706.03472 (cit. on p. 11).

[17] Roland Kwitt et al. "Statistical Topological Data Analysis - A Kernel Perspective". In: *Advances in Neural Information Processing Systems 28.* Ed. by C. Cortes et al. Curran Associates, Inc., 2015, pp. 3070–3078 (cit. on p. 11).

[18] Clément Maria et al. "The Gudhi Library: Simplicial Complexes and Persistent Homology". In: *Mathematical Software – ICMS 2014.* International Congress on Mathematical Software. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, Aug. 5, 2014, pp. 167–174. ISBN: 978-3-662-44198-5. DOI: 10.1007/978-3-662-44199-2_28 (cit. on p. 7).

[19] Dimitriy Morozov. *dionysus: Library for computing persistent homology.* original-date: 2017-07-14T19:02:35Z. Apr. 11, 2018 (cit. on p. 7).

[20] Elizabeth Munch et al. "Probabilistic Fréchet means for time varying persistence diagrams". In: *Electronic Journal of Statistics* 9.1 (2015), pp. 1173–1204. ISSN: 1935-7524. DOI: 10.1214/15-EJS1030 (cit. on p. 10).

[21] M. E. J. Newman. "Network structure from rich but noisy data". In: *Nature Physics* 14.6 (June 2018), pp. 542–545. ISSN: 1745-2481. DOI: 10.1038/s41567-018-0076-1 (cit. on p. 3).

[22] Nina Otter et al. "A roadmap for the computation of persistent homology". In: *EPJ Data Science* 6.1 (Dec. 1, 2017), p. 17. ISSN: 2193-1127. DOI: 10.1140/epjds/s13688-017-0109-5 (cit. on p. 8).

[23] Steve Y. Oudot. *Persistence theory: from quiver representations to data analysis.* Mathematical surveys and monographs volume 209. Providence, Rhode Island: American Mathematical Society, 2015. 218 pp. ISBN: 978-1-4704-2545-6 (cit. on p. 7).

[24] Giovanni Petri et al. "Topological Strata of Weighted Complex Networks". In: *PLoS ONE* 8 (June 1, 2013), e66506. DOI: 10.1371/journal.pone.0066506 (cit. on p. 8).

[25] Jan Reininghaus. *DIPHA (A Distributed Persistent Homology Algorithm).* original-date: 2015-12-25T17:23:32Z. Apr. 3, 2018 (cit. on p. 7).

[26] Jan Reininghaus et al. "A stable multi-scale kernel for topological machine learning". In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2015, pp. 4741–4748 (cit. on p. 11).

[27] Etsuji Tomita, Akira Tanaka, and Haruhisa Takahashi. "The worst-case time complexity for generating all maximal cliques and computational experiments". In: *Theoretical Computer Science.* Computing and Combinatorics 363.1 (Oct. 25, 2006), pp. 28–42. ISSN: 0304-3975. DOI: 10.1016/j.tcs.2006.06.015 (cit. on p. 8).

[28] Katharine Turner et al. "Fréchet Means for Distributions of Persistence Diagrams". In: *Discrete & Computational Geometry* 52.1 (July 1, 2014), pp. 44–70. ISSN: 0179-5376, 1432-0444. DOI: 10.1007/s00454-014-9604-7 (cit. on p. 10).

[29] Afra Zomorodian. “The Tidy Set: A Minimal Simplicial Set for Computing Homology of Clique Complexes”. In: *Proceedings of the Twenty-sixth Annual Symposium on Computational Geometry*. SoCG ’10. New York, NY, USA: ACM, 2010, pp. 257–266. ISBN: 978-1-4503-0016-2. DOI: 10.1145/1810959.1811004 (cit. on p. 8).

[30] Afra Zomorodian and Gunnar Carlsson. “Computing Persistent Homology”. In: *Discrete & Computational Geometry* 33.2 (Feb. 1, 2005), pp. 249–274. ISSN: 0179-5376, 1432-0444. DOI: 10.1007/s00454-004-1146-y (cit. on pp. 5, 6).