Topology, Big Data and Optimization

Mikael Vejdemo-Johansson and Primoz Skraba

Abstract The idea of using geometry in learning and inference has a long history going back to canonical ideas such as Fisher information, Discriminant analysis, and Principal component analysis. The related area of Topological Data Analysis (TDA) has been developing in the last decade. The idea is to extract robust topological features from data and use these summaries for modeling the data. A topological summary generates a coordinate-free, deformation invariant and highly compressed description of the geometry of an arbitrary data set. Topological techniques are well-suited to extend our understanding of Big Data. These tools do not supplant existing techniques, but rather provide a complementary viewpoint to existing techniques. The qualitative nature of topological features do not give particular importance to individual samples, and the coordinate-free nature of topology generates algorithms and viewpoints well suited to highly complex datasets. With the introduction of persistence and other geometric-topological ideas we can find and quantify local-to-global properties as well as quantifying qualitative changes in data.

Keywords Applied topology · Persistent homology · Mapper · Euler characteristic curve · Topological Data Analysis

1 Introduction

All data is geometric.

Every data set is characterized by the way individual observations compare to each other. Statistics of data sets tend to describe location (mean, median, mode) or shape of the data. The shape is intrinsically encoded in the mutual distances between

M. Vejdemo-Johansson (✉)
Computer Vision and Active Perception Laboratory, KTH Royal Institute
of Technology, 100 44 Stockholm, Sweden
e-mail: mvj@kth.se

P. Skraba
AI Laboratory, Jozef Stefan Institute, Jamova 39, Ljubljana, Slovenia
e-mail: primoz.skraba@ijs.si

A. Emrouznejad (ed.), *Big Data Optimization: Recent Developments and Challenges*, Studies in Big Data 18, DOI 10.1007/978-3-319-30265-2_7

data points, and analyses of data sets extract geometric invariants: statistical descriptors that are stable with respect to similarities in data. A statistic that describes a data set needs to stay similar if applied to another data set describing the same entity.

In most mainstream big data, computationally lean statistics are computed for data sets that exceed the capacity of more traditional methods in volume, variety, velocity or complexity. Methods that update approximations of location measures, or of fits of simple geometric models—probability distributions or linear regressions—to tall data, with high volume or velocity, are commonplace in the field.

We will focus instead on a family of methods that pick up the geometric aspects of big data, and produce invariants that describe far more of the *complexity* in wide data: invariants that extract a far more detailed description of the data set that goes beyond the location and simplified shapes of linear or classical probabilistic models. For the more detailed geometric description, computational complexity increases. In particular worst case complexities tend to be far too high to scale to large data sets; but even for this, a linear complexity is often observed in practice.

Whereas geometric methods have emerged for big data, such as information geometry [7] and geometric data analysis [67], our focus is on topological methods. Topology focuses on an underlying concept of *closeness*, replacing *distance*. With this switch of focus, the influence of noise is dampened, and invariants emerge that are coordinate-free, invariant under deformation and produce compressed representations of the data. The *coordinate-free* nature of topological methods means, inter alia, that the ambient space for data—the width of the data set—is less relevant for computational complexities and analysis techniques than the intrinsic complexity of the data set itself. *Deformation invariance* is the aspect that produces stability and robustness for the invariants, and dampens out the effects of noise. Finally, *compressed representations* of data enables far quicker further analyses and easily visible features in visualizations.

One first fundamental example of a topological class of algorithms is *clustering*. We will develop homology, a higher-dimensional extension of clustering, with persistent homology taking over the role of hierarchical clustering for more complex shape features. From these topological tools then flow coordinatization techniques for dimensionality reduction, feature generation and localization, all with underlying stability results guaranteeing and quantifying the fidelity of invariants to the original data.

Once you are done with this book chapter, we recommend two further articles to boost your understanding of the emerging field of topological data analysis: Topology and Data by Carlsson [24] and Barcodes: the persistent topology of data by Ghrist [57].

We will start out laying down the fundamentals of topology in Sect. 2. After the classical field of topology, we introduce the adaptation from pure mathematics to data analysis tools in Sect. 3. An important technique that has taken off significantly in recent years is Mapper, producing an intrinsic network description of a data set. We describe and discuss Mapper in Sect. 4. In Sect. 5 we explore connections between topology and optimization: both how optimization tools play a large importance in our topological techniques, and how topological invariants and partitions of features help setup and constrain classes of optimization problems. Next in Sect. 6, we go through several classes of applications of the techniques seen earlier in the chapter. We investigate how topology provides the tools to glue local information into global descriptors, various approaches to nonlinear dimensionality reduction with topological tools, and emerging uses in visualization.

2 Topology

Topology can be viewed as geometry where *closeness* takes over the role of *size* from classical geometry. The fundamental notion is closeness expressed through connectedness. This focus on connections rather than sizes means that invariants focus on qualitative features rather than quantitative: features that do not change with the change of units of measurement, and that stay stable in the face of small perturbations or deformations. For an introductory primer, we recommend the highly accessible textbook by Hatcher [61]. Most of what follows are standard definitions and arguments, slightly adapted to our particular needs in this chapter.

In topological data analysis the focus is on compressed combinatorial representations of shapes. The fundamental building block is the *cell complex*, most often the special case of a *simplicial complex*—though for specific applications *cubical complex*es or more general constructions are relevant.

Definition 1 A *convex polytope* (or convex polyhedron) is the convex hull of some collection of points in $\mathbb{R}^d$. The dimension of the polytope P is the largest n such that the intersection of P with some n-dimensional linear subspace of $\mathbb{R}^d$ contains an n-dimensional open ball.

For an n-dimensional polytope, its boundary decomposes into a union of $n-1$-dimensional polytopes. These are called the *facets* of the polytope. Decomposing facets into their facets produces lower dimensional building blocks—this process continues all the way down to vertices. The set of facets of facets etc. are called the *faces* of the polytope. We write P_n for the set of n-dimensional faces of P.

A *cell complex* is a collection of convex polytopes where the intersection of any two polytopes is a face of each of the polytopes.

We illustrate these geometric conditions in Fig. 1.

From a cell complex, we can produce a *chain complex*. This is a collection of vector spaces with linear maps connecting them. C_nP is the vector space spanned by the n-dimensional faces of P: C_nP has one basis vector for each n-dimensional face. The connecting linear maps are called *boundary maps*: the boundary map $\partial_n : C_nP \to C_{n-1}P$ maps the basis vector v_σ corresponding to a face σ to a linear combination of the vectors that correspond to facets of σ. The coefficients of this linear combination depends on the precise way that the polytopes are connected—if

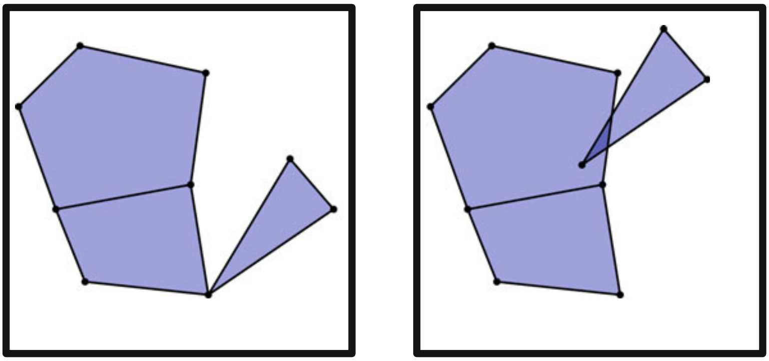


Fig. 1 *Left* a valid polyhedral complex in $\mathbb{R}^2$. *Right* an invalid polyhedral complex in $\mathbb{R}^2$. There are invalid intersections where the *triangle* and the *pentagon* overlap

we work with vector spaces over $\mathbb{Z}_2$ these coefficients reduce to 0 and 1 and the boundary of v_σ is

$$\partial v_\sigma = \sum_{\tau \text{ facet of } \sigma} v_\tau$$

These coefficients need to ensure that $\partial_n \partial_{n+1} = 0$.

The cell complex can be represented in full abstraction as the boundary maps, abstracting away the geometry of this definition completely.

Most commonly, we use simplicial complexes—complexes where the polyhedra are all simplices. Simplices are the same shapes that show up in the simplex method in optimization. Geometrically, an n-dimensional simplex is the convex hull of $n + 1$ points in general position: where no $k + 1$ points lie on the same k-dimensional plane. More interesting for our applications is the idea of an *abstract simplicial complex*.

Definition 2 An *abstract simplicial complex* Σ on a set of (totally ordered) vertices V is a collection of subsets of vertices (simplices) such that whenever some set $\{v_0, \ldots, v_k\}$ is in Σ, so is every subset of that set.

We usually represent a simplex as a sorted list of its constituent vertices.

The boundary map assigns to the facet $[v_0, \ldots, v_{i-1}, v_{i+1}, \ldots, v_k]$ the coefficient $(-1)^i$ so that the full expression of the boundary map is

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

where $\hat{v}$ means to leave v out of the simplex.

We illustrate this definition in Fig. 2.

Now consider a closed chain of edges, such as $a - b - c - d - a$ in Fig. 3. The boundary of the sum of these edges includes each vertex twice: once from each edge

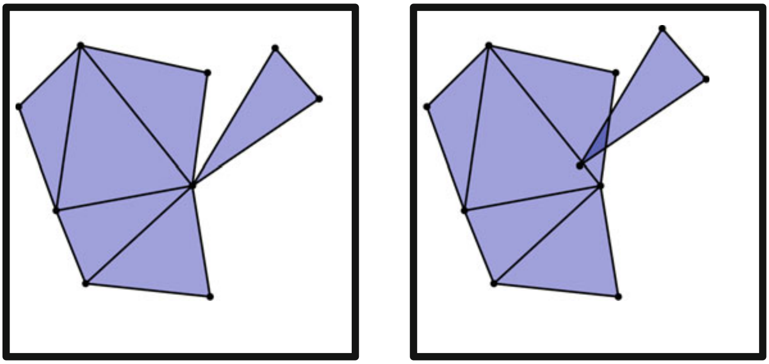
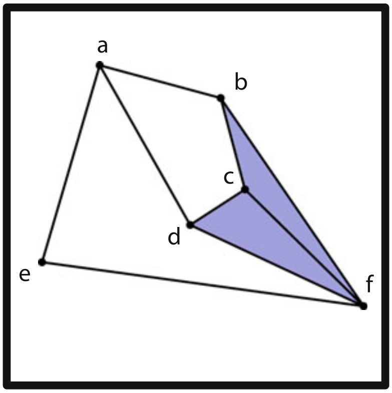


Fig. 2 *Left* a valid simplicial complex in $\mathbb{R}^2$. *Right* an invalid simplicial complex in $\mathbb{R}^2$. There are invalid intersections where two *triangles* overlap

Fig. 3 An illustration of cycles and boundaries. $a - b - c - d - a$ is an essential cycle, while $b - c - d - f - b$ is a non-essential cycle, filled in by higher-dimensional cells



that includes the vertex. The coefficients of these vertices cancel out to 0, so that the closed chain is in $\ker \partial_1$. This generalizes: an element of $\ker \partial_n$ is a collection of n-cells that enclose an $n + 1$-dimensional hypervolume of some sort, in the same way that a closed chain of edges can be seen as enclosing some surface.

Some of these closed chains in a given complex end up being filled in, such as the sequence $b - c - d - f - b$ in Fig. 3, while others have an empty void enclosed. The cells that fill in a closed cycle are part of $C_{n+1}\Sigma$, and the boundary map applied to those cells precisely hits the enclosing cell collection. Thus, $\operatorname{img} \partial_{n+1}$ is the collection of closed cycles that are filled in. This means that the vector space quotient $\ker \partial_n / \operatorname{img} \partial_{n+1}$ is precisely the *essential* enclosures: those that detect a void of some sort.

Definition 3 The n-dimensional *homology* $H_n(\Sigma)$ of a cell complex Σ is the vector space $\ker \partial_n / \operatorname{img} \partial_{n+1}$.

Later in the text we will also need the concept of *cohomology*. This is the homology of the vector space dual of the chain complex: we write $C^n\Sigma = C_n\Sigma$, and $\delta^n = \partial_n^T$; the transposed matrix. Elements of $C^n\Sigma$ correspond to $\mathbb{R}$-valued maps defined on the n-dimensional cells. The n-dimensional cohomology $H^n(\Sigma)$ is $\ker\delta^n/\operatorname{img}\delta^{n-1}$.

3 Persistence

The tools and definitions in Sect. 2 all are most relevant when we have a detailed description of the topological shape under study. In any data-driven situation, such as when facing big or complex data, the data accessible tends to take the shape of a discrete point cloud: observations with some similarity measure, but no intrinsic connection between them.

Persistence is the toolbox, introduced in [51] and developed as the foundation of topological data analysis that connects discrete data to topological tools acting on combinatorial or continuous shapes.

At the heart of persistence is the idea of sweeping a parameter across a range of values and studying the ways that a shape derived from data changes with the parameter change. For most applications, the shape is constructed by "decreasing focus": each data point is smeared out over a larger and larger part of the ambient space until the smears start intersecting. We can sometimes define these shapes using only dissimilarity between points, removing the role of an ambient space completely so that data studied can have arbitrary representations as long as a dissimilarity measure is available. These intersection patterns can be used to build cell complexes that then can be studied using homology, cohomology, and other topological tools.

The most commonly used construction for this smearing process is the *Vietoris-Rips complex*. For a data set $\mathbb{X}$, the vertex set is the set of data points. We introduce a simplex $[x_0, \ldots, x_k]$ to the complex $\mathrm{VR}_\varepsilon(\mathbb{X})$ precisely when all pairwise dissimilarities are small enough: $d(x_i, x_j) < \varepsilon$. An illustration can be found in Fig. 4.

At each parameter value ε, there is a simplicial complex $\mathrm{VR}_\varepsilon(\mathbb{X})$. As the parameter grows, no intersections vanish—so no existing simplices vanish with a growing parameter. By functoriality—a feature of the homology construction—there is a kind of continuity for topological features: the inclusion maps of simplicial complexes generate linear maps between the corresponding homology (or cohomology) vector spaces. For a growing sequence $\varepsilon_0 < \varepsilon_1 < \varepsilon_2 < \varepsilon_3$, there are maps

$$\begin{tikzcd}
\mathrm{VR}_{\varepsilon_0}(\mathbb{X}) \arrow[r, hook] & \mathrm{VR}_{\varepsilon_1}(\mathbb{X}) \arrow[r, hook] & \mathrm{VR}_{\varepsilon_2}(\mathbb{X}) \arrow[r, hook] & \mathrm{VR}_{\varepsilon_3}(\mathbb{X}) \\
H_k\mathrm{VR}_{\varepsilon_0}(\mathbb{X}) \arrow[r] & H_k\mathrm{VR}_{\varepsilon_1}(\mathbb{X}) \arrow[r] & H_k\mathrm{VR}_{\varepsilon_2}(\mathbb{X}) \arrow[r] & H_k\mathrm{VR}_{\varepsilon_3}(\mathbb{X}) \\
H^k\mathrm{VR}_{\varepsilon_0}(\mathbb{X}) & H^k\mathrm{VR}_{\varepsilon_1}(\mathbb{X}) \arrow[l] & H^k\mathrm{VR}_{\varepsilon_2}(\mathbb{X}) \arrow[l] & H^k\mathrm{VR}_{\varepsilon_3}(\mathbb{X}) \arrow[l]
\end{tikzcd}$$

For a diagram of vector spaces like these, there is a consistent basis choice across the entire diagram. This basis choice is, dependent on the precise argument made, either a direct consequence of the structure theorem for modules over a Principal Ideal Domain (result available in most commutative algebra textbooks, e.g. [53])

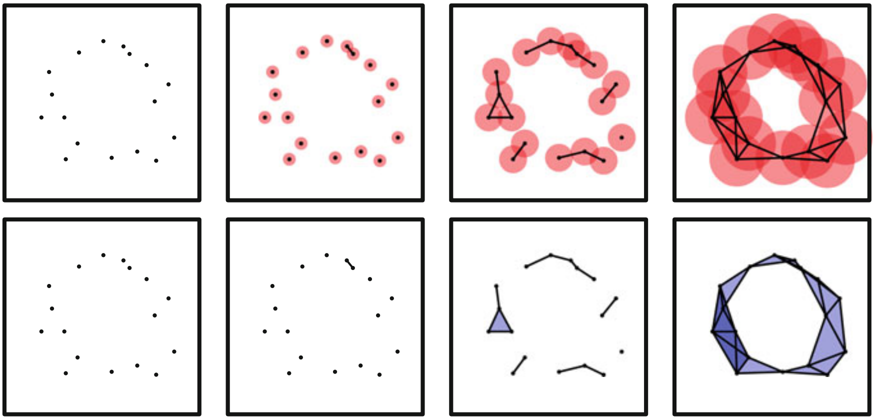


Fig. 4 The growth of a Vietoris-Rips complex as points are smeared out. *Top row* is the view of the data with each data point surrounded by a $\varepsilon/2$ radius ball, while the *bottom row* shows the corresponding abstract complex as it grows. At the very end, the *circle*-like nature of the point cloud can be detected in the Vietoris-Rips complex. This will stick around until ε is large enough that the hole in the *middle* of the *top right figure* is filled in

or a direct consequence of Gabriel's theorem [56] on decomposing modules over tame quivers. The whole diagram splits into components of one-dimensional vector spaces with a well defined start and endpoint along the diagram. These components correspond precisely to topological features, and tell us at what parameter value a particular feature shows up, and at what value it is filled in and vanishes. The components are often visualized as a *barcode*, as can be seen in Fig. 5.

Features that exist only along a very short range of parameter values can be considered noisy: probably the result of sampling errors or inherent noise in the production of the data. These show up along the diagonal of the persistence diagram. Features that exist along a longer range of parameter values are more likely to be

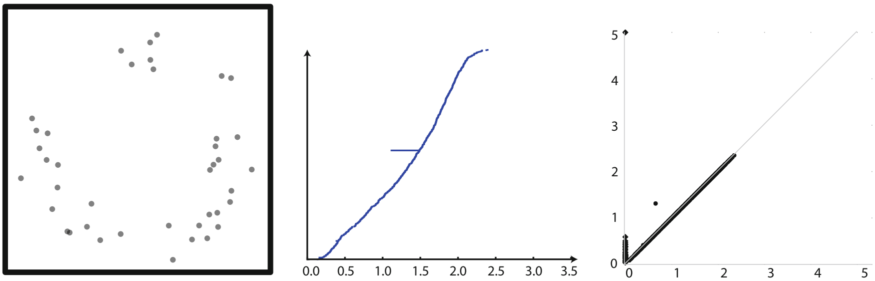


Fig. 5 To the *left*, a point cloud. In the *middle*, the corresponding persistence barcode for dimension 1 homology. To the *right*, the persistence diagram for dimension 0 (*diamonds* along the y-axis) and dimension 1. We see a large amount of very short intervals, and then one significantly larger interval corresponding to the *circle*-like shape of the data

features inherent in the source of the data—and the length of the corresponding bar is a measure of the size of the feature.

These barcode descriptors are stable in the sense that a bound on a perturbation of a data set produces a bound on the difference between barcodes. This stability goes further: data points can vanish or appear in addition to moving around, and there still are bounds on the difference of barcodes.

In particular, this means that with guarantees on sampling density and noise levels, large enough bars form a certificate for the existence of particular topological features in the source of a data set.

Additional expositions of these structures can be found in the survey articles by Carlsson [24] and by Ghrist [57]. For the algebraic and algorithmic focused aspects, there are good surveys available from the first author [113], and by Edelsbrunner and Harer [47, 48].

3.1 Persistence Diagrams as Features

Homology serves as a rough descriptor of a space. It is naturally invariant to many different different types of transformations and deformations. Unfortunately, homology groups of a single space (for example, data viewed at a single scale) are highly unstable and lose too much of the underlying geometry. This is where persistence enters the picture. Persistence captures information in a stable way through the filtration. For example, the Vietoris-Rips filtration encodes information about the underlying metric space.

Therefore, by choosing an appropriate filtration, we can encode information about the space. The first such instance was referred to as *topological inference*. The intensity levels of brain activity in fMRI scans was investigated using Euler characteristic curves [6].

These curves have a long history in the probabilistic literature [5, 116–118], are topological in nature and can be inferred from persistence diagrams. The Euler characteristic can be computed by taking the alternating sum of the ranks of homology groups (or equivalently Betti numbers),

$$\chi(X) = (-1)^k \mathrm{rk}(H_k(X))$$

If X is parameterized by t, we obtain an Euler characteristic curve. Surprisingly, the expectation of this quantity can be computed analytically in a wide range of settings. This makes it amenable for machine learning applications. Another notable application of this approach can be found in distinguishing stone tools from different archaeological sites [93].

These methods work best in the functional setting where the underlying space is fixed (usually some triangulated low dimensional manifold).

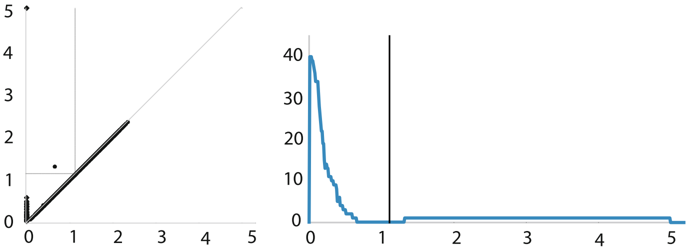


Fig. 6 The relation between an Euler characteristic curve and the corresponding persistence diagram. To the *left*, a persistence diagram, with the quadrant anchored at some (t, t) marked out. To the *right*, the Euler characteristic curve from the corresponding data set, with the corresponding t marked

The persistence diagram encodes more information—the Euler characteristic curve can easily be computed from a persistence diagram by taking the alternating sum over different dimensions for each quadrant anchored at (t, t) as in Fig. 6.

There are several different approaches to using persistence diagrams as features. Initially, it was observed that the space of persistence diagrams can be transformed into a metric space [112]. A natural metric for persistence diagrams is the *bottleneck matching distance*. Given two diagrams Dgm(F) and Dgm(G) (corresponding to two filtrations F and G), the bottleneck distance is defined as

$$d_B(\mathrm{Dgm}(F), \mathrm{Dgm}(G) = \inf_{\phi \in bijections} \sup_{p \in F} d_\infty(p, \phi(p))$$

This has the benefit of always being well-defined, but also has been shown to be not as informative as other distances—namely, Wasserstein distances.

The most commonly used Wasserstein distances used are:

1. 1-Wasserstein distance—W_1
2. 2-Wasserstein distance—W_2

Under some reasonable conditions, persistence diagrams satisfy stability under these metric as well [40] (albeit with a worse constant in front).

While first order moments exist in this space in the form of Frechet means, this space is generally quite complicated. For example, while means exist, there are no guarantees they are unique [112]. Below, we have an example of this phenomenon. This presents numerous algorithmic challenges both for computing the means themselves, as well as for interpretation. This can be made Hölder continuous by considering the *distribution* of persistence diagrams [81].

Ultimately, the problem with viewing the space of persistence diagrams as a metric space is that the space is insufficiently nice to allow for standard machine learning techniques. Furthermore, the standard algorithmic solution for computing bottleneck distance is the Hungarian algorithm for computing the maximum weight bipartite matching between the two diagrams. This computes an explicit matching between points and has at worst an $O(n^3)$ complexity where n is the number of points in the

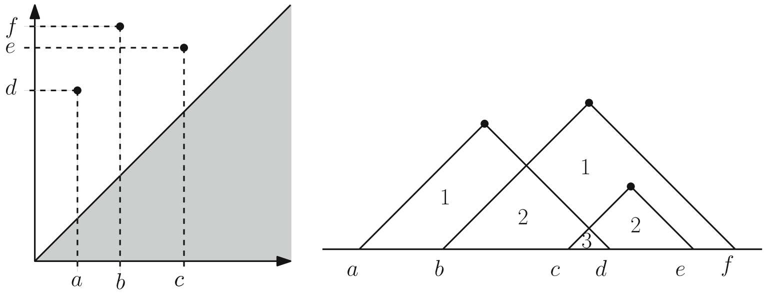


Fig. 7 The construction of a persistence landscape from a persistence diagram. We rotate the diagram by 45°. Each point in the persistence diagram creates a region in the persistent landscape by considering the 45° lines from the point. These correspond to *vertical* and *horizontal lines* from the point to diagonal. To each region in the landscape, we assign the number which corresponds to how many times it is covered. Distances between landscapes are computed by integrating the absolute difference point-wise

persistence diagrams. This is often expensive, which led to the development of the following algorithms.

The key insight came with the development of the *persistence landscape*. This is a functional on the space of the persistence diagrams in line with the kernel trick in machine learning. The main idea is to raise the diagram into a functional space (usually a Hilbert space), where the space behaves fundamentally like Euclidean space, making techniques like support vector machines feasible.

There have been several approaches to constructing functionals on persistence diagrams. The most developed is the *persistence landscape* [19]. This assigns to each point in the plane a support on how many points lie above it. We illustrate the process in Fig. 7, but it assigns to each point in the plane a number which corresponds to how many points in the persistence diagram cover it. In addition to being useful for machine learning algorithms, it is also much faster to compute that distances directly on persistence diagrams (which are based on bipartite matching problems).

The algebraic structure connecting persistence diagrams to functionals was partially addressed in [3]. In this work, Adcock et al. show that the algebraic structure in persistence diagram has a family of functionals which can be used to parameterize the family. This was used to train a SVM classifier on the MINST handwriting dataset. The performance of the classifier is near state-of-the-art, where it is important to mention this the case for generic features rather than the specially chosen ones in current state-of-the-art techniques for handwriting recognition. The same techniques were also used to classify hepatic lesions [4].

3.1.1 Applications

Here we recount some successful applications of the above techniques to real-world data.

The first is on a study of the effects of psilocybin (e.g. magic mushrooms) on the brain using fMRI [86]. In this case, persistence diagram based features are shown to clearly divide the brain activity under the effects of psilocybin from normal brain activity. The authors found that while normal brain activity is highly structured, brain activity under psilocybin is much more chaotic, connecting parts of the brain which are usually not connected.

The second application we highlight the use of topological features to distinguish stone tools coming from different archaeological sites [93]. In this work, the authors began with three dimensional models of the tools obtained from scans. Then they computed the Euler characteristic curves given by curvature, e.g. they used curvature as the filtering function. They found that training a classifier using these curves, they were able to obtain a high classification accuracy (∼80 %).

The final application we highlight is for detecting and classifying periodicity in gene expression time series [84]. Gene expressions are a product of the cell cycle and in this work, the authors recognize that in sufficiently high dimensional space, periodicity is characterized by closed one forms (i.e. circles). The work in the following section makes a similar observation, but parametrizes the circle rather than compute a feature. Circles are characterized by one-dimensional homology and so the authors use the 1-dimensional persistence diagram in order to compare the periodicity of different gene expressions. To obtain, the persistence diagram, the authors embed the time series in high dimension using a sliding window embedding (also known as a Takens' embedding or a delay embedding). The idea is, given a time series $x(1), x(2), \ldots$, take a sliding window over a time series and map each point to the vector of the window. For example, for a window size of three, a data point at time 1, $x(1)$ would be mapped to the vector $[x(1), x(2), x(3)]$ which is in $\mathbb{R}^3$. After some normalization, the authors computed the persistence diagram of the embedded time series which they used to compare different gene expressions.

3.2 Cohomology and Circular Coordinates

One particular derived technique from the persistent homology described here is using persistent cohomology to compute coordinate functions with values on the circle. We have already mentioned cohomology, and it plays a strong role in the development of fast algorithms. For the applications to coordinatization, we use results from homotopy theory—another and far less computable part of algebraic topology. This approach was developed by the first author joint with de Silva and Morozov [78, 101, 102].

An equivalence class element in $H^1(X, \mathbb{Z})$—an equivalence class of functions $X \to \mathbb{Z}$—corresponds to an equivalence class of functions $X \to S^1$ to the circle. The correspondence is algorithmic in nature, and efficient to compute. In particular, for any specific function $X \to \mathbb{Z}$ in a cohomology equivalence class The drawback at this stage is that applied to complexes like the Vietoris-Rips complex produces maps that send all data points to a single point on the circle.

We can work around this particular problem by changing the target domain of the function from $\mathbb{Z}$ to $\mathbb{R}$. As long as we started out with a $\mathbb{Z}$-valued function, and stay in the same equivalence class of functions, the translation to circle-valued coefficient maps remains valid. So we can optimize for as smooth as possible a circle-valued map. This turns out to be, essentially, a LSQR optimization problem: the circle valued function related to a cocycle ζ with the coboundary matrix B is

$$\arg\min_{z} \|\zeta - Bz\|_2$$

reduced modulo 1.0. The cocycle is a 1-dimensional cocycle, and B is the coboundary map from 0-cochains to 1-cochains. This makes the correspondingly computed z a 0-cochain—a circle-valued function on the vertices and hence on the data set itself.

4 Mapper

Mapper is a different approach to topological data analysis. Proposed in 2008, it is much faster than persistent homology, and produces an intrinsic shape of an arbitrary data set as a small simplicial complex. This complex can be used for visualization or for further analysis. Applications of this method have been widespread: from medical research through financial applications to politics and sports analyses. This section is based on several articles by Singh et al. [71, 104].

At the core of the Mapper algorithm is the idea that data can be viewed through "lenses"—coordinate functions displaying interesting characteristics of the data set. For any such lens, the data can be stratified according to values of that lens, and local summaries within each stratum can be related to each other to form a global picture of the data set. We see the process illustrated in Fig. 8.

To be precise, given a dataset $\mathbb{X}$ and some function $\ell : \mathbb{X} \to \mathbb{R}^k$ and a cover of $\mathbb{R}^k$ by overlapping open subsets U_i (for instance open balls or open axis-aligned hypercubes), we compute all inverse images $\ell^{-1}(U_i)$. Each such inverse image might contain data points separated from each other—using a clustering algorithm of the user's choice, each inverse image is broken down into its component clusters. Finally, since the sets U_i cover $\mathbb{R}^n$, some of them will overlap. These overlaps may contain data points: when they do, a data point contained in clusters from several inverse images $\ell^{-1}(U_{i_0}), \ell^{-1}(U_{i_1}), \dots, \ell^{-1}(U_{i_k})$ gives rise to a k-simplex spanned by the corresponding clusters. The collection of clusters from the various layers with these connecting simplices forms a simplicial complex describing the inherent shape of the data set. For any given data point, its corresponding location in the simplicial complex can easily be found.

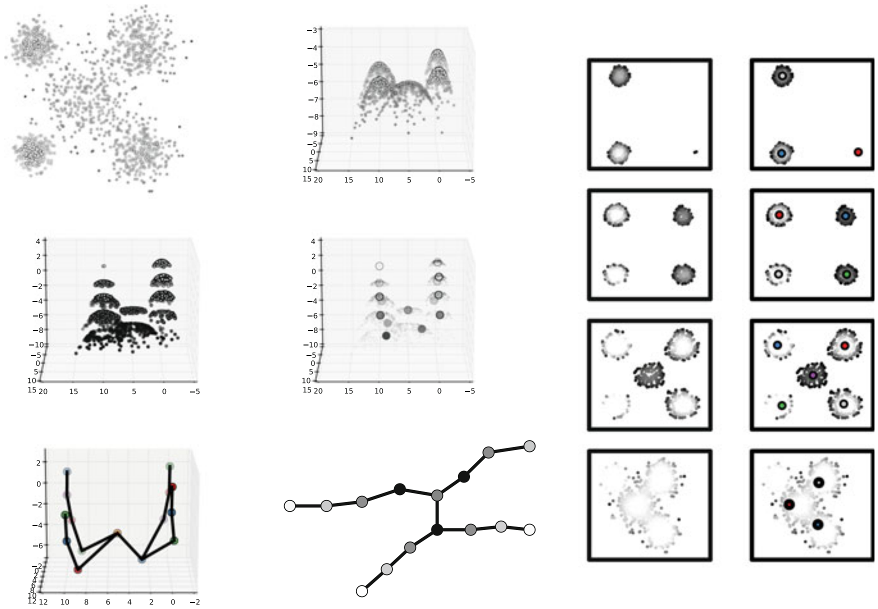


Fig. 8 The entire Mapper pipeline applied to random samples from a mixture of Gaussians, as viewed through the lens of Gaussian density estimation. In the *left two columns* are: the original data set; a graph of the density function on the data set; this same graph split up according to the Mapper method; computed clusters within each section of the split; the corresponding mapper graph in 3D; the resulting Mapper graph in 2D. In the *right two columns*, we see the split and cluster process in more detail: the *left* of these two has the points as split into sections while the *right* has these same points as well as their cluster centers

5 Optimization

Topological tools group objects by qualitative behaviors, in ways that can be deformed to each other within each group. Finding good representatives for qualitative features often turn out to be a case of searching within such a class for an optimal member.

Computing a representative circle-valued coordinate from a cohomology class $[\zeta]$ is a matter of computing $\arg\min_{\mathrm{x}} \phi(\zeta - \mathrm{Bx})$ for some penalty function ϕ defining the optimality of the coordinate function, where B is the coboundary matrix of the triangulation. In [102], the penalty function chosen was $\phi(w) = \|w\|_2$, whereas for other applications, other penalty functions can be used.

The work of computing optimal homology cycles has gotten a lot of attention in the field, using growing neighborhoods, total unimodularity or computational geometry and matroid theory [21, 34, 35, 43, 54]. Unimodularity in particular turns out to have a concrete geometric interpretation: simplifying the optimization significantly, it requires all subspaces to be torsion free. An interesting current direction of research is the identification of problems which become tractable when the equivalence class is fixed. There are many examples of fixed-parameter tractable algorithms—where there is an exponential dependence on a certain parameter (such as dimension). In such instances, it would be beneficial to identify the global structure of the data and optimize within each (co)homology class. This has been used indirectly in network and other shortest path routing [23, 62].

Another area where homology shows up as a tool for optimization is in evaluating coverage for sensor agents—such as ensembles of robots, or antenna configurations. Here, for a collection of agents with known coverage radii and a known set of boundary agents, degree 2 homology of the Vietoris-Rips complex of the agents relative to the boundary reveals whether there are holes in the coverage, and degree 1 homology of the Vietoris-Rips complex reveals where holes are located [59, 99, 100]. This has given rise to a wealth of applications, some of which can be found in [2, 42, 46, 80, 108, 115].

In other parts of topological data analysis, optimization formulations or criteria form the foundations of results or constructions—in ways that turn out unfeasible and require approximations or simplifications for practical use. The main example is for the various stability results that have shown up for persistent homology. The metrics we use for persistence diagrams, bottleneck and Wasserstein distances, take the form of optimization problems over spaces of bijections between potentially large finite sets [20, 27, 28, 31, 32, 38, 39, 41, 44, 69].

6 Applications

6.1 Local to Global

Topological techniques are designed to extract global structure from local information. This local information may be in the form of a metric or more generally a similarity function. Often a topological viewpoint can yield new insights into existing techniques. An example of this *persistence-based clustering* [30]. This work is closely related with *mode-seeking clustering* techniques [36]. This class of methods assumes the points are sampled from some underlying density function and defines the clusters as the modes of the density function (e.g. the basins of attraction of the peaks of the density function). There are generally two steps involved:

1. Estimation of the underlying density function
2. Estimation of the peaks

These techniques have the advantage that the number of clusters is not required as input. Rather the main problem is to determine which peaks are “real” versus which peaks are noise. For example, in *mean-shift clustering*, points are flowed to local maxima incrementally, but require a stopping parameter (as we never exactly hit the peak). There are many other criteria which have been proposed—however, it turns out that persistence provides an important insight. Due to the stability of the

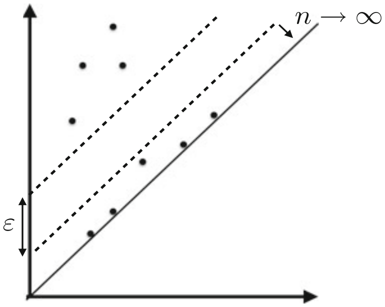


Fig. 9 A persistence diagram with a gap of ε. The topological noise and the "true" features are separated by an empty region of size ε. Note that under mild assumptions on the underlying sample, the noise goes to 0 as the number of points goes to infinity, therefore the gap increases with increasing amounts of data

persistence diagram, if we the first step (i.e. estimation of the density function) is done correctly, then the persistence diagram is provably close. Furthermore, if there is a separation of noise and the peaks (i.e. a gap as shown in Fig. 9), then we can estimate the number of clusters as well. It can also be shown that the noise goes to zero as the number of points increases, ensuring that the gap exists if we have sufficiently many points.

This approach also allows for the identification of stable and unstable parts of the clustering. The main idea is that since the persistence diagram is stable, the number of clusters is also stable. Furthermore, persistent clusters can be uniquely identified in the presence of resampling, added noise, etc. The idea is illustrated in Fig. 10. This can be useful when determining unstable regions for tasks such as segmentation [105]. Here unstable regions are themselves considered separate segments.

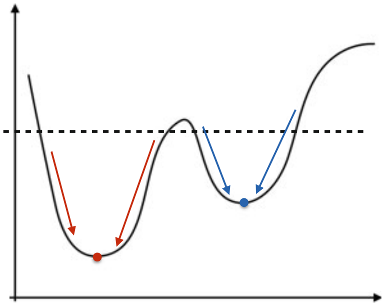


Fig. 10 Persistence based clustering is based on the idea that the basins of attraction of the peaks of a density function are clusters. Here we show the negative of a density function (so we look for valleys rather than peaks), with two clusters. For clustering, there also exists a spatial stability for persistent clusters, since if we consider a point before two clusters meet, they are disjoint—shown here by the *dashed line*

In addition to clustering based on density (or a metric—in which case we obtain single-linkage clustering), topological methods can find clusters which have similar local structure. Often we consider data as living in Euclidean space or a Riemannian manifold. While this may be *extrinsically* true, i.e. the data is embedded in such a space, the *intrinsic* structure of data is rarely this nice. A natural generalization of a manifold is the notion of a *stratified space*. This can be thought of as a mixture of manifolds (potentially of different dimensions) which are glued together in a nice way. Specifically, the intersection of two manifold pieces is itself be a manifold. The collection of manifolds of a given dimension is called a *stratum*. We omit the technical definition, but refer the reader to the excellent technical notes [74].

The problem of *stratified manifold learning* is to identify the manifold pieces directly from the data. The one dimensional version of this problem is the graph construction problem, which has been considered for reconstruction of road networks from GPS traces [1]. In this setting, zero dimensional strata are intersections, forks and merges (i.e. vertices in the graph) while one dimensional strata are the connecting roads (i.e. edges in the graph). Some examples are shown in Fig. 11.

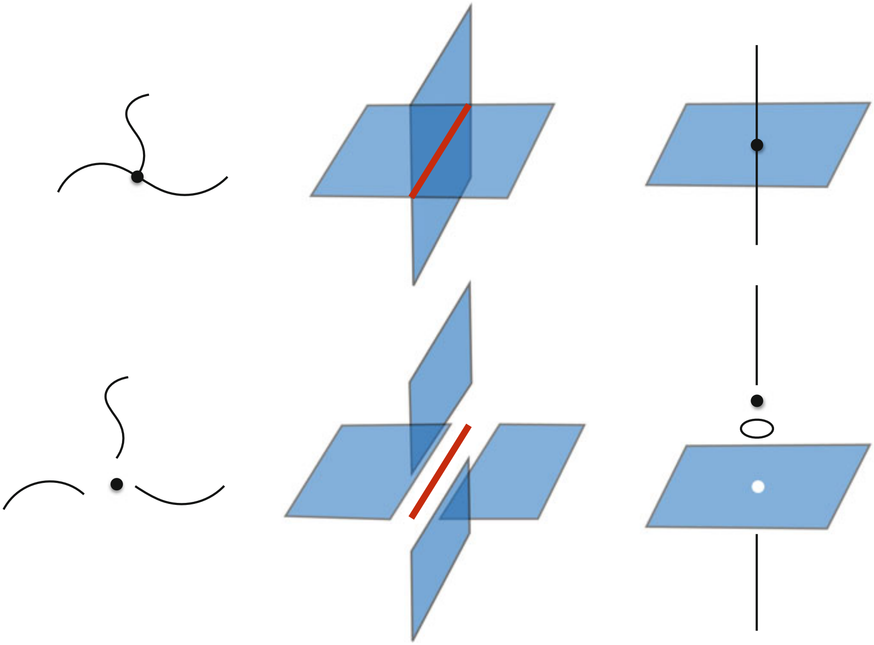


Fig. 11 Three examples of stratified spaces (*top row*) and their corresponding strata (*bottom row*). On the *left*, we have part of a graph with three edges (1-strata) coming from a vertex (0-strata). The intersection of two 1-strata, must be a 0-strata. In the *middle*, we have two planes intersecting along a *line*. This gives four 2 strata, which meet together at a 1-strata. On the *right* we have a line through a plane. The plane is a 2-strata, the *line* is divided into two pieces and the intersection is a 1 and 0-strata. The intersection is a point, however to ensure it is a valid stratification we must consider the loop around the point

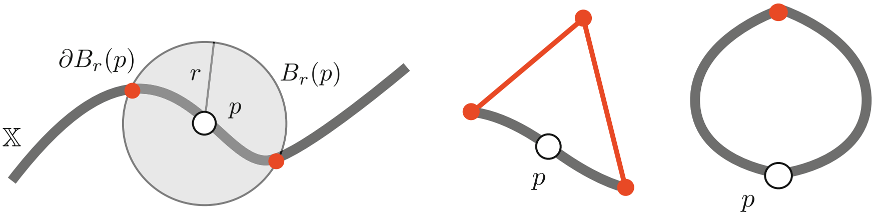


Fig. 12 The intuition behind local homology. We consider the local neighborhood around a point, in this case an intersection between the *ball* of radius r at point p and the space $\mathbb{X}$. The boundary is two points which are collapsed together to make a *circle*

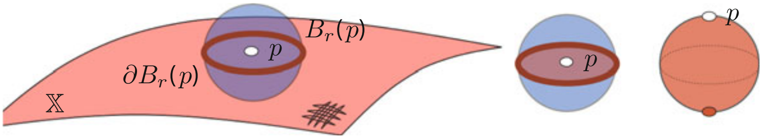


Fig. 13 A higher dimensional example for local homology. In this case an intersection between the *ball* of radius r at point p and the space $\mathbb{X}$ is a disc and the boundary is a *circle*. When we collapse the boundary (i.e. *circle*) to a point we obtain a sphere

The problem has been considered in more generality in [14]. Intuitively, the problem can be thought of as determining the local structure of a space and then clustering together points which share the same structure. The topological tool for determining the local structure is called *local homology*. The idea is to consider the structure of the local neighborhood. An important technical point is that we consider the quotient of the neighborhood modulo its boundary. By collapsing the boundary to a point (as shown in Figs. 12 and 13), we can distinguish different dimensions. In the case of a k-dimensional manifold, each point will have the same structure—that of a k-dimensional sphere. In the cases shown in Fig. 11, we obtain a different answer. On the left we obtain a wedge of two circles, in the middle a wedge of three spheres and on right we obtain a wedge of a sphere and two circles.

While preliminary results have been promising, this is currently an active area of research.

6.2 Nonlinear Dimensionality Reduction

Dimensionality reduction is well rooted in data analysis, as a way to reduce an unmanageably wide data set to a far more narrow and thus more easily analyzed derived data set. Classical techniques often work by linear projections, as is done by principal component analysis or by random projection methods. While some techniques for non-linear dimensionality reduction have been known since the 1960s [63–65], a more significant boost in the development of new reduction methods showed up in the late 1990s.

van der Maaten et al. [72] distinguish between three types of nonlinear dimensionality reduction techniques: those that try to preserve global features; those that try to preserve local features; and those that globally align mixtures of linear techniques.

Where some methods are focused on retaining most or all distances in the dataset—such as multidimensional scaling [63], many nonlinear techniques focus on retaining *closeness*.

Isomap [111] generates approximations to the geodesic distance of a dataset as computed on a neighborhood graph. As compared to MDS, it puts more emphasis on retaining closeness by first focusing the scaling on local connectivity before generating a coordinate set globally approximating these geodesic distances.

Local techniques fit a linear local dimensionality reduction to small neighborhoods in the data and then gluing the local coordinates to a global description. First out was Local Linear Embeddings (LLE): fitting a local tangent plane through each data points, and then minimizes the distortion of these local tangents [95]. This minimization reduces to an eigenvector computation.

Several improvements on LLE have been constructed as eigenvector computations of the Laplacian operator, or through enriched representations of the local tangent descriptions [13, 45, 70]. The Laplacian or Laplace operator is a classical operator in algebraic topology. The coboundary operator δ has a dual operator δ^*—represented by the same matrix as the boundary operator. The Laplacian is defined as the composition $\delta^*\delta$. The operator smooths out a function along the connectivity of the underlying space, and its eigenmaps form smooth—in the sense of keeping nearby points close together—and produces globally defined functions that retain closeness of data points.

Isomap and LLE both suffer from weaknesses when constructing coordinate functions on data sets with holes. One possible solution was offered by Lee and Verleysen [68], who give a graph algorithm approach to cutting the data set to remove the non-trivial topology. They give a complexity of $O(n \log_2 n)$ for their cutting procedure, based on using Dijkstra's algorithm for spanning trees. Such a cut can also be produced based on persistent cohomology, with a representative cocycle demonstrating a required cut to reduce topological complexity [9, 26, 58]. While the worst case complexity for this computation is matrix-multiplication time, for many data sets, linear complexity has been observed [12, 121].

Some shapes require more linear coordinates to represent accurately than the intrinsical dimension would indicate. A first example is the circle: while a one-dimensional curve, any one-dimensional projection will have to collapse distant points to similar representations. With the techniques we describe in Sect. 3.2, we can generate circle-valued coordinates for the data points. This has been used in finding cyclic structures [10, 37] and for analyzing quasiperiodic or noisily recurrent signals in arbitrary dimension [94, 114].

Mapper provides an approach to dimensionality reduction with intrinsic coordinate spaces: instead of providing features on a line or a circle, the Mapper output is a small, finite model space capturing the intrinsic shape of the original data set.

The often large reduction in representation size with a Mapper reduction enables speedups in large classes of problems. Classic dimensionality reduction such as done by MDS, Isomap or LLE can be done on the Mapper model, and with coordinate values pulled back and interpolated onto the original data points themselves, while optimization problems could be solved on the Mapper model to produce seed values or approximate solutions when pulled up to the original data points. As long as all functions involved are continuous, and the Mapper analysis sufficiently fine grained, each vertex of the Mapper model corresponds to a compact set of data points with trivial topology and each higher dimensional simplex corresponds to a connection between sets of data points.

6.3 Dynamics

Topological methods have a long history of use in simplifying, approximating and analyzing dynamical systems. For this approach, the Conley index—a homology group computed in a small neighborhood in a dynamical system—gives a measure of the local behavior of the dynamical system, stable and useful for nonlinear and multiparameter cases. This approach has found extensive use [22, 75, 79].

Computing persistence on point clouds from dynamical systems, and then using clustering to extract features from the resulting invariants has found some use. In [15], bifurcation detection for dynamical systems using persistent cohomology was explored, while in [66] clustered persistence diagrams helped classify gait patterns to detect whether and what people were carrying from video sequences.

The idea of using the Takens delay embedding [109] to create point clouds representative of dynamic behavior from timeseries data has emerged simultaneous from several groups of researchers in topological data analysis. Harer and Perea [85] used 1-dimensional persistent homology to pick out appropriate parameters for a delay embedding to improve accuracy for the embedded representation of the original dynamical system. The same idea of picking parameters for a delay embedding, but with different approaches for subsequent analyses were described by Skraba et al. [103], and later used as the conceptual basis for the analysis of the dynamics of motion capture generated gait traces by Vejdemo-Johansson et al. [114]. The work in [114] uses persistent cohomology to detect intrinsic phase angle coordinates, and then use these either to create an average gait cycle from a sequence of samples, or to generate gait cycle classifiers functions, indicating similarity of a new sequence of gait samples to the sequences already seen.

From the same group of researchers, persistent homology and cohomology has been used for motion planning in robotics. Moduli spaces for grasping procedures give geometric and topological ways of analyzing and optimizing potential grasp plans [87, 88, 90], and 1-dimensional persistent homology provides suggestions for grasp sites for arbitrary classes of objects with handles [92, 106, 107]. Topology also generates constraints for motion planning optimization schemes, and produces approaches for caging grasps of wide classes of objects [89, 91, 119, 120].

6.4 Visualization

Topological techniques are common in visualization, particularly so-called scientific visualization. Perhaps the most prominent of these applications are topological skeleta and the visualization and simplification of two or three dimensional scalar and vector fields.

6.4.1 Topological Skeleta

There has been a large amount of work on topological skeleta extraction. Here we highlight two types of constructions (without exhaustively describing all related work)

- Reeb graphs
- Contour trees

Though there are many variations the main idea behind both constructions is that given a space $\mathbb{X}$ and a real-valued function, i.e.

$$f : \mathbb{X} \to \mathbb{R}$$

a topological summary can be computed by taking every possible function value $a \in \mathbb{R}$, and considering its preimage, $f^{-1}(a) \in \mathbb{X}$. For each preimage, we can count the number of connected components.[1] If we consider very small intervals rather than just points, we see that we can connect these components if they overlap. Connecting these connected components together using this criteria, we obtain a graph (again under reasonable assumptions). The resulting graph is called a *Reeb graph*.

By only considering the connected components, a potentially high-dimensional structure can be visualized as a graph. However, the input need not be high-dimensional, as these constructions have are useful as shape descriptors for 2-dimensional shapes. In addition, they are a crucial part of understanding 3-dimensional data sets, where direct visualization is impossible.

When the underlying space is contractible, there is additional structure, which allows for more efficient computation of the structure, interestingly in any dimension [25]. This is mainly due to the observation that if the underlying space is contractible (such as on a convex subset of Euclidean space), then the Reeb graph has the structure of a *tree*, and is therefore called a *contour tree*.

Mapper can be thought of as a "fuzzy" Reeb graph, where connectivity is scale-dependent and is computed via clustering rather than as an intrinsic property of the space.

[1] Technically, these are path-connected components. However, this distinction is a mathematical formality, as the two are indistinguishable in any form of sampled data.

6.4.2 Features and Simplification

In addition to visualizing a scalar function, the presence of noise, or more generally multi-scale structure, makes the ability to perform simplification desirable. By simplifying a function, larger structures become clearer as they are no longer overwhelmed by large numbers of smaller features.

There has been a substantial amount of work done on simplifying functions on topological spaces. Initially, the work was done on two dimensional manifolds (surfaces) using the Morse-Smale complex [50] and has been extended to three dimensions [49, 60]. Morse theory connects the topological and geometric properties of a space in terms of the critical points of a function on that space. For example, consider the height function on a sphere. This has two critical points, the global minimum (the bottom of the sphere) and maximum (the top of the sphere). If we distort the function to add another maxima, we also add a saddle. Simplification in this setting proceeds by considering the reverse of this process. By combining minima and saddles (or maxima and saddles), it simplifies the underlying function. The order of simplification can be done in a number of different ways, such as distance based (i.e. distance between critical points). The persistence ordering is given if it is done by relative heights (e.g. small relative heights first) and the methods are closely tied to Morse theory [17].

The natural extension from scalar fields is to vector fields, that is each point is assigned a vector. These are often used to model flows in simulations of fluids or combustion. Simplifying these is much more difficult than simplifying vector fields. However, the notion of fixed point naturally generalizes critical points of a scalar function. These are studied in *topological dynamics*. We highlight two topological approaches which are based on Conley index theory and degree theory respectively. The Conley index [96] is a topological invariant based on homology, which is an extension of Morse theory. The main problem in this approach is the requirement to find a neighborhood which isolates the fixed points from the rest of the flow. This neighborhood (called an *isolating neighborhood*), must be nice with the respect to the flow, in particular, the flow must not be internally tangent to the boundary of the neighborhood.

The second approach is based on a variant of persistence called *robustness* [52]. A vector field in Euclidean space can be thought of as a map from $\mathbb{R}^n \to \mathbb{R}^n$ and we can compute its robustness diagram [33]. This carries similar information and and shares similar stability properties as the persistence diagram. Furthermore, this can be connected to degree theory, which is yet another invariant developed to describe maps. Famously, the existence of Nash equilibrium in game theory is the consequence of a fixed point theorem (i.e. Brouwer fixed point theorem). Using a classical result from differential topology, which states that if a part of the flow has has degree zero, then is can be deformed to a fixed point free vector field, a general vector field can be simplified using the robustness order in the same way as persistence order gives an ordering in the case of scalar fields.

This is only a brief glimpse at topology and visualization as this is a large field of research. The main motivation for using topological methods for large complex data sets is that they ensure consistency, whereas ad-hoc methods may introduce artifacts during simplification.

7 Software and Limitations

These methods are implemented in a range of software packages. There is a good survey of the current state of computation and computational timings written by Otter et al. [83]. We will walk through a selection of these packages and their strengths and weaknesses here.

The available software can be roughly divided by expected platform. Some packages are specifically adapted to work with the statistics and data analysis platform R, some for interacting well with Matlab, and some for standalone use or to work as libraries for software development.

In R, the two main packages are pHom and R-TDA. pHom is currently abandoned, but still can be found and included. R-TDA [55] is active, with a userbase and maintenance. Both are built specifically to be easy to use from R and integrate into an R data analysis workflow.

When working in Matlab, or in any other java-based computational platform—such as Maple or Mathematica—the main software choice is JavaPlex [110] or JPlex [97]. JPlex is the predecessor to JavaPlex, built specifically for maximal computational efficiency on a Java platform, while JavaPlex was built specifically to make extension of functionality easy. Both of them are also built to make the user experience as transparent and accessible as possible: requiring minimal knowledge in topological data analysis to be usable. While less efficient than many of the more specialized libraries, JavaPlex has one of the most accessible computational pipelines. The survey by Otter et al. [83] writes "However, for small data sets (less than a million simplices) the software Perseus and javaPlex are best suited because they are the easiest to use[...]".

Several other packages have been constructed that are not tied to any one host platform: either as completely standalone processing software packages, or as libraries with example applications that perform many significant topological data analysis tasks. Oldest among these is ChomP [76]. ChomP contains a C++ library and a couple of command line applications to compute persistent homology of cubical sets, and has been used in dynamical systems research [8]. Perseus [82] works on Vietoris-Rips complexes generated from data sets, as well as from cubical and simplicial complexes. Perseus needs its input data on a particular format, with meta data about the data points at the head of the input file, which means many use cases may need to adjust input data to fit with Perseus expectations. DIPHA [11] is the first topological data analysis program with built in support for distributed computing: building on the library PHAT [12], DIPHA works with MPI for parallelization or distribution of computation tasks. DIPHA takes in data, and produces a persistence diagram, both in their own file format—the software distribution includes Matlab functions to convert to and from the internal file format. Last among the standalone applications, Dionysus [77] is a library for topological data analysis algorithm development in C++ and Python. The package comes with example application for persistent homology and cohomology, construction of Vietoris-Rips complexes and a range of further techniques from computational topology.

Two more libraries are focused on use for software developers. Gudhi [73] is a library that focuses on the exploration of different data structures for efficient computations of persistent homology or with simplicial complexes. CTL[2] is a very recent library maintained by Ryan Lewis. The library is still under development, and currently supports persistent homology and complex construction, and has plans to support persistent cohomology, visualizations and bindings to other languages and platforms.

Complex construction the current computational bottleneck in topological data analysis. Simplicial complexes are built up dimension by dimension and in higher dimensions, a small number of points can result in a large number of simplices. For example in 2000 points in dimension 6 can easily yield overall billion simplicies. We do have the option of limiting our analysis to low dimension (e.g. clustering only requires the graph to be built), and there are techniques which yield an approximate filtration while maintaining a linear size [98]. Current research is finding further speedups as well as modifying this to a streaming model. The second problem is that although the volume of data is getting larger, the data itself does not cover the entire space uniformly and preforming a global analysis where we have insufficient data in some regions is impossible. One approach that is currently being explored is how to construct "likely" analysis to fill in regions where data is sparse (e.g. anomalies).

8 Conclusions

At the state of the field today, topological data analysis has proven itself to produce descriptors and invariants for topological and geometric features of data sets. These descriptors are

COORDINATE-FREE so that the descriptors are ultimately dependent only on a measure of similarity or dissimilarity between observations. Ambient space, even data representation and their features are not components of the analysis methods, leading to a set of tools with very general applicability.

STABLE UNDER PERTURBATIONS making the descriptors stable against noise. This stability forms the basis for a topological inference.

COMPRESSED so that even large data sets can be reduced to small representations while retaining topological and geometric features in the data.

[2] http://ctl.appliedtopology.org/.

Looking ahead, the adaptation and introduction of classical statistical and inferential techniques into topological data analysis is underway [16, 18, 29, 112].

The problem of efficient constructions of simplicial complexes encoding data geometry remains both under-explored and one of the most significant bottlenecks for topological data analysis.

Over the last few years, big data techniques have been developed which perform well for specific tasks: building classifiers, linear approaches or high speed computations of simple invariants of large volume data sets. As data and complexity grows, the need emerges for methods that support interpretation and transparency—where the data is made accessible and generalizable without getting held back by the simplicity of the chosen models. These more qualitative approaches need to include both visualization and structure discovery: nonlinear parametrization makes comparison and correlation with existing models easier. The problems we encounter both in non-standard optimization problems and in high complexity and large volume data analysis are often NP-hard in generality. Often, however, restricting the problem to a single equivalence class under some equivalence relation—often the kinds found in topological methods—transforms the problem to a tractable one: examples are maximizing a function over only one persistent cluster, or finding optimal cuts using cohomology classes to isolate qualitatively different potential cuts. The entire area around these directions is unexplored, wide open for research. We have begun to see duality, statistical approaches and geometric features of specific optimization problems show up, but there is a wealth of future directions for research.

As for data, the current state of software has problems both with handling streaming data sources and data of varying quality. The representations available are dependent on all seen data points, which means that in a streaming or online setting, the computational problem is constantly growing with the data stream. Data quality has a direct impact on the computational results. Like with many other techniques, topological data analysis cannot describe what is not present in the data but rather will produce a description of the density indicated by the data points themselves. If the data quality suffers from variations in the sampling density, the current software is not equipped to deal with the variations. There is research [30] into how to modify the Vietoris-Rips construction to handle well-described sampling density variations, but most of the major software packages have yet to include these modifications.

All in all, topological data analysis creates features and descriptors capturing topological and geometric aspects of complex and wide data.

References

1. Aanjaneya, M., Chazal, F., Chen, D., Glisse, M., Guibas, L., Morozov, D.: Metric graph reconstruction from noisy data. Int. J. Comput. Geom. Appl. **22**(04), 305–325 (2012)
2. Adams, H., Carlsson, G.: Evasion paths in mobile sensor networks. Int. J. Robot. Res. **34**(1), 90–104 (2015)
3. Adcock, A., Carlsson, E., Carlsson, G.: The ring of algebraic functions on persistence bar codes. http://comptop.stanford.edu/u/preprints/multitwo (2012)

4. Adcock, A., Rubin, D., Carlsson, G.: Classification of hepatic lesions using the matching metric. Comput. Vis. Image Underst. **121**, 36–42 (2014)
5. Adler, R.J.: The Geometry of Random Fields, vol. 62. Siam (1981)
6. Adler, R.J.: Some new random field tools for spatial analysis. Stochast. Environ. Res. Risk Assess. **22**(6), 809–822 (2008)
7. Amari, S.I., Nagaoka, H.: Methods of Information Geometry, vol. 191. American Mathematical Society (2007)
8. Arai, Z., Kalies, W., Kokubu, H., Mischaikow, K., Oka, H., Pilarczyk, P.: A database schema for the analysis of global dynamics of multiparameter systems. SIAM J. Appl. Dyn. Syst. **8**(3), 757–789 (2009)
9. Babson, E., Benjamini, I.: Cut sets and normed cohomology with applications to percolation. Proc. Am. Math. Soc. **127**(2), 589–597 (1999)
10. Bajardi, P., Delfino, M., Panisson, A., Petri, G., Tizzoni, M.: Unveiling patterns of international communities in a global city using mobile phone data. EPJ Data Sci. **4**(1), 1–17 (2015)
11. Bauer, U., Kerber, M., Reininghaus, J.: Distributed computation of persistent homology. In: ALENEX, pp. 31–38. SIAM (2014)
12. Bauer, U., Kerber, M., Reininghaus, J., Wagner, H.: PHAT-persistent homology algorithms toolbox. In: Mathematical Software-ICMS 2014, pp. 137–143. Springer (2014)
13. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Comput. **15**(6), 1373–1396 (2003)
14. Bendich, P., Wang, B., Mukherjee, S.: Local homology transfer and stratification learning. In: Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1355–1370. SIAM (2012)
15. Berwald, J., Gidea, M., Vejdemo-Johansson, M.: Automatic recognition and tagging of topologically different regimes in dynamical systems. Discontinuity Non-linearity Complex. **3**(4), 413–426 (2015)
16. Blumberg, A.J., Gal, I., Mandell, M.A., Pancia, M.: Robust statistics, hypothesis testing, and confidence intervals for persistent homology on metric measure spaces. Found. Comput. Math. **14**(4), 745–789 (2014)
17. Bremer, P.T., Edelsbrunner, H., Hamann, B., Pascucci, V.: A multi-resolution data structure for two-dimensional morse-smale functions. In: Proceedings of the 14th IEEE Visualization 2003 (VIS'03), p. 19. IEEE Computer Society (2003)
18. Bubenik, P.: Statistical Topology Using Persistence Landscapes (2012)
19. Bubenik, P.: Statistical topological data analysis using persistence landscapes. J. Mach. Learn. Res. **16**, 77–102 (2015)
20. Bubenik, P., Scott, J.A.: Categorification of persistent homology. arXiv:1205.3669 (2012)
21. Busaryev, O., Cabello, S., Chen, C., Dey, T.K., Wang, Y.: Annotating simplices with a homology basis and its applications. In: Algorithm Theory-SWAT 2012, pp. 189–200. Springer (2012)
22. Bush, J., Gameiro, M., Harker, S., Kokubu, H., Mischaikow, K., Obayashi, I., Pilarczyk, P.: Combinatorial-topological framework for the analysis of global dynamics. Chaos: Interdiscip. J. Nonlinear Sci. **22**(4), 047,508 (2012)
23. Cabello, S., Giannopoulos, P.: The complexity of separating points in the plane. In: Proceedings of the Twenty-Ninth Annual Symposium on Computational Geometry, pp. 379–386. ACM (2013)
24. Carlsson, G.: Topology and data. Am. Math. Soc. **46**(2), 255–308 (2009)
25. Carr, H., Snoeyink, J., Axen, U.: Computing contour trees in all dimensions. Comput. Geom. **24**(2), 75–94 (2003)
26. Chambers, E.W., Erickson, J., Nayyeri, A.: Homology flows, cohomology cuts. SIAM J. Comput. **41**(6), 1605–1634 (2012)
27. Chazal, F., Cohen-Steiner, D., Glisse, M., Guibas, L.J., Oudot, S.Y.: Proximity of persistence modules and their diagrams. In: Proceedings of the 25th Annual Symposium on Computational Geometry, SCG'09, pp. 237–246. ACM, New York, NY, USA (2009). doi:10.1145/1542362.1542407

28. Chazal, F., Cohen-Steiner, D., Guibas, L.J., Oudot, S.Y.: The Stability of Persistence Diagrams Revisited (2008)
29. 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. ACM (2014)
30. Chazal, F., Guibas, L.J., Oudot, S.Y., Skraba, P.: Persistence-based clustering in riemannian manifolds. J. ACM (JACM) **60**(6), 41 (2013)
31. Chazal, F., de Silva, V., Glisse, M., Oudot, S.: The structure and stability of persistence modules. arXiv:1207.3674 (2012)
32. Chazal, F., de Silva, V., Oudot, S.: Persistence stability for geometric complexes. arXiv:1207.3885 (2012)
33. Chazal, F., Skraba, P., Patel, A.: Computing well diagrams for vector fields on $\mathbb{R}^n$. Appl. Math. Lett. **25**(11), 1725–1728 (2012)
34. Chen, C., Freedman, D.: Quantifying homology classes. arXiv:0802.2865 (2008)
35. Chen, C., Freedman, D.: Hardness results for homology localization. Discrete Comput. Geom. **45**(3), 425–448 (2011)
36. Cheng, Y.: Mean shift, mode seeking, and clustering. IEEE Trans. Pattern Anal. Mach. Intell. **17**(8), 790–799 (1995)
37. Choudhury, A.I., Wang, B., Rosen, P., Pascucci, V.: Topological analysis and visualization of cyclical behavior in memory reference traces. In: Pacific Visualization Symposium (PacificVis), 2012 IEEE, pp. 9–16. IEEE (2012)
38. Cohen-Steiner, D., Edelsbrunner, H., Harer, J.: Stability of persistence diagrams. Discrete Comput. Geom. **37**(1), 103–120 (2007)
39. Cohen-Steiner, D., Edelsbrunner, H., Harer, J.: Extending persistence using Poinca and Lefschetz duality. Found. Comput. Math. **9**(1), 79–103 (2009). doi:10.1007/s10208-008-9027-z
40. Cohen-Steiner, D., Edelsbrunner, H., Harer, J., Mileyko, Y.: Lipschitz functions have L_p-stable persistence. Found. Comput. Math. **10**(2), 127–139 (2010)
41. 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, SCG'06, pp. 119–126. ACM, New York, NY, USA (2006). doi:10.1145/1137856.1137877
42. de Silva, V., Ghrist, R., Muhammad, A.: Blind swarms for coverage in 2-D. In: Robotics: Science and Systems, pp. 335–342 (2005)
43. Dey, T.K., Hirani, A.N., Krishnamoorthy, B.: Optimal homologous cycles, total unimodularity, and linear programming. SIAM J. Comput. **40**(4), 1026–1044 (2011)
44. Dey, T.K., Wenger, R.: Stability of critical points with interval persistence. Discrete Comput. Geom. **38**(3), 479–512 (2007)
45. Donoho, D.L., Grimes, C.: Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. Proc. Natl. Acad. Sci. **100**(10), 5591–5596 (2003)
46. Dłotko, P., Ghrist, R., Juda, M., Mrozek, M.: Distributed computation of coverage in sensor networks by homological methods. Appl. Algebra Eng. Commun. Comput. **23**(1), 29–58 (2012). doi:10.1007/s00200-012-0167-7
47. Edelsbrunner, H., Harer, J.: Persistent homology—a survey. In: Goodman, J.E., Pach, J., Pollack, R. (eds.) Surveys on Discrete and Computational Geometry: Twenty Years Later, Contemporary Mathematics, vol. 453, pp. 257–282. American Mathematical Society (2008)
48. Edelsbrunner, H., Harer, J.: Computational Topology: An Introduction. AMS Press (2009)
49. Edelsbrunner, H., Harer, J., Natarajan, V., Pascucci, V.: Morse-smale complexes for piecewise linear 3-manifolds. In: Proceedings of the Nineteenth Annual Symposium on Computational Geometry, pp. 361–370. ACM (2003)
50. Edelsbrunner, H., Harer, J., Zomorodian, A.: Hierarchical morse complexes for piecewise linear 2-manifolds. In: Proceedings of the Seventeenth Annual Symposium on Computational Geometry, pp. 70–79. ACM (2001)

51. Edelsbrunner, H., Letscher, D., Zomorodian, A.: Topological persistence and simplification. In: 41st Annual Symposium on Foundations of Computer Science, 2000. Proceedings, pp. 454–463 (2000)
52. Edelsbrunner, H., Morozov, D., Patel, A.: Quantifying transversality by measuring the robustness of intersections. Found. Comput. Math. **11**(3), 345–361 (2011)
53. Eisenbud, D.: Commutative Algebra with a View Toward Algebraic Geometry, vol. 150. Springer (1995)
54. Erickson, J., Whittlesey, K.: Greedy optimal homotopy and homology generators. In: Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1038–1046. Society for Industrial and Applied Mathematics (2005)
55. Fasy, B.T., Kim, J., Lecci, F., Maria, C.: Introduction to the R package TDA. arXiv:1411.1830 (2014)
56. Gabriel, P.: Unzerlegbare Darstellungen I. Manuscripta Mathematica **6**(1), 71–103 (1972). doi:10.1007/BF01298413
57. Ghrist, R.: Barcodes: the persistent topology of data. Bull. Am. Math. Soc. **45**(1), 61–75 (2008)
58. Ghrist, R., Krishnan, S.: A topological max-flow-min-cut theorem. In: Proceedings of Global Signal Inference (2013)
59. Ghrist, R., Muhammad, A.: Coverage and hole-detection in sensor networks via homology. In: Proceedings of the 4th International Symposium on Information Processing in Sensor Networks, p. 34. IEEE Press (2005)
60. Gyulassy, A., Natarajan, V., Pascucci, V., Hamann, B.: Efficient computation of morse-smale complexes for three-dimensional scalar functions. IEEE Trans. Vis. Comput. Graph. **13**(6), 1440–1447 (2007)
61. Hatcher, A.: Algebraic Topology. Cambridge University Press (2002)
62. Huang, K., Ni, C.C., Sarkar, R., Gao, J., Mitchell, J.S.: Bounded stretch geographic homotopic routing in sensor networks. In: INFOCOM, 2014 Proceedings IEEE, pp. 979–987. IEEE (2014)
63. Kruskal, J.B.: Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrika **29**(1), 1–27 (1964)
64. Kruskal, J.B.: Nonmetric multidimensional scaling: a numerical method. Psychometrika **29**(2), 115–129 (1964)
65. Kruskal, J.B., Wish, M.: Multidimensional Scaling, vol. 11. Sage (1978)
66. Lamar-Leon, J., Baryolo, R.A., Garcia-Reyes, E., Gonzalez-Diaz, R.: Gait-based carried object detection using persistent homology. In: Bayro-Corrochano, E., Hancock, E. (eds.) Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, no. 8827 in Lecture Notes in Computer Science, pp. 836–843. Springer International Publishing (2014)
67. Le Roux, B., Rouanet, H.: Geometric Data Analysis. Springer, Netherlands, Dordrecht (2005)
68. Lee, J.A., Verleysen, M.: Nonlinear dimensionality reduction of data manifolds with essential loops. Neurocomputing **67**, 29–53 (2005). doi:10.1016/j.neucom.2004.11.042
69. Lesnick, M.: The Optimality of the Interleaving Distance on Multidimensional Persistence Modules. arXiv:1106.5305 (2011)
70. Li, X., Lin, S., Yan, S., Xu, D.: Discriminant locally linear embedding with high-order tensor data. IEEE Trans. Syst. Man Cybern. Part B: Cybern. **38**(2), 342–352 (2008)
71. Lum, P.Y., Singh, G., Lehman, A., Ishkanov, T., Vejdemo-Johansson, M., Alagappan, M., Carlsson, J., Carlsson, G.: Extracting insights from the shape of complex data using topology. Sci. Rep. **3** (2013). doi:10.1038/srep01236
72. van der Maaten, L.J., Postma, E.O., van den Herik, H.J.: Dimensionality reduction: a comparative review. J. Mach. Learn. Res. **10**(1–41), 66–71 (2009)
73. Maria, C., Boissonnat, J.D., Glisse, M., Yvinec, M.: The Gudhi library: simplicial complexes and persistent homology. In: Mathematical Software-ICMS 2014, pp. 167–174. Springer (2014)
74. Mather, J.: Notes on Topological Stability. Harvard University Cambridge (1970)

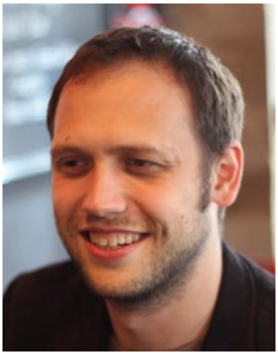
75. Mischaikow, K.: Databases for the global dynamics of multiparameter nonlinear systems. Technical report, DTIC Document (2014)
76. Mischaikow, K., Kokubu, H., Mrozek, M., Pilarczyk, P., Gedeon, T., Lessard, J.P., Gameiro, M.: Chomp: Computational homology project. http://chomp.rutgers.edu
77. Morozov, D.: Dionysus. http://www.mrzv.org/software/dionysus/ (2011)
78. Morozov, D., de Silva, V., Vejdemo-Johansson, M.: Persistent cohomology and circular coordinates. Discrete Comput. Geom. **45**(4), 737–759 (2011). doi:10.1007/s00454-011-9344-x
79. Mrozek, M.: Topological dynamics: rigorous numerics via cubical homology. In: Advances in Applied and Computational Topology: Proceedings Symposium, vol. 70, pp. 41–73. American Mathematical Society (2012)
80. Muhammad, A., Jadbabaie, A.: Decentralized computation of homology groups in networks by gossip. In: American Control Conference, ACC 2007, pp. 3438–3443. IEEE (2007)
81. Munch, E., Turner, K., Bendich, P., Mukherjee, S., Mattingly, J., Harer, J.: Probabilistic fréchet means for time varying persistence diagrams. Electron. J. Statist. **9**(1), 1173–1204 (2015). doi:10.1214/15-EJS1030. http://dx.doi.org/10.1214/15-EJS1030
82. Nanda, V.: Perseus: The Persistent Homology Software (2012)
83. Otter, N., Porter, M.A., Tillmann, U., Grindrod, P., Harrington, H.A.: A roadmap for the computation of persistent homology. arXiv:1506.08903 [physics, q-bio] (2015)
84. Perea, J.A., Deckard, A., Haase, S.B., Harer, J.: Sw1pers: Sliding windows and 1-persistence scoring; discovering periodicity in gene expression time series data. BMC Bioinf. (Accepted July 2015)
85. Perea, J.A., Harer, J.: Sliding windows and persistence: an application of topological methods to signal analysis. Found. Comput. Math. **15**(3), 799–838 (2013)
86. Petri, G., Expert, P., Turkheimer, F., Carhart-Harris, R., Nutt, D., Hellyer, P.J., Vaccarino, F.: Homological scaffolds of brain functional networks. J. R. Soc. Interface **11**(101) (2014). doi:10.1098/rsif.2014.0873
87. Pokorny, F.T., Bekiroglu, Y., Exner, J., Björkman, M.A., Kragic, D.: Grasp Moduli spaces, Gaussian processes, and multimodal sensor data. In: RSS 2014 Workshop: Information-based Grasp and Manipulation Planning (2014)
88. Pokorny, F.T., Bekiroglu, Y., Kragic, D.: Grasp moduli spaces and spherical harmonics. In: Robotics and Automation (ICRA), 2014 IEEE International Conference on, pp. 389–396. IEEE (2014)
89. Pokorny, F.T., Ek, C.H., Kjellström, H., Kragic, D.: Topological constraints and kernel-based density estimation. In: Advances in Neural Information Processing Systems 25, Workshop on Algebraic Topology and Machine Learning, 8 Dec, Nevada, USA (2012)
90. Pokorny, F.T., Hang, K., Kragic, D.: Grasp moduli spaces. In: Robotics: Science and Systems (2013)
91. Pokorny, F.T., Kjellström, H., Kragic, D., Ek, C.: Persistent homology for learning densities with bounded support. In: Advances in Neural Information Processing Systems, pp. 1817–1825 (2012)
92. Pokorny, F.T., Stork, J., Kragic, D., others: Grasping objects with holes: A topological approach. In: 2013 IEEE International Conference on Robotics and Automation (ICRA), pp. 1100–1107. IEEE (2013)
93. Richardson, E., Werman, M.: Efficient classification using the Euler characteristic. Pattern Recogn. Lett. **49**, 99–106 (2014)
94. Robinson, M.: Universal factorizations of quasiperiodic functions. arXiv:1501.06190 [math] (2015)
95. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**(5500), 2323–2326 (2000)
96. Salamon, D.: Morse theory, the conley index and floer homology. Bull. London Math. Soc **22**(2), 113–140 (1990)
97. Sexton, H., Vejdemo-Johansson, M.: jPlex. https://github.com/appliedtopology/jplex/ (2008)
98. Sheehy, D.R.: Linear-size approximations to the vietoris-rips filtration. Discrete Comput. Geom. **49**(4), 778–796 (2013)

99. de Silva, V., Ghrist, R.: Coordinate-free coverage in sensor networks with controlled boundaries via homology. Int. J. Robot. Res. **25**(12), 1205–1222 (2006). doi:10.1177/0278364906072252
100. de Silva, V., Ghrist, R.: Coverage in sensor networks via persistent homology. Algebraic Geom. Topol. **7**, 339–358 (2007)
101. de Silva, V., Morozov, D., Vejdemo-Johansson, M.: Dualities in persistent (co)homology. Inverse Prob. **27**(12), 124,003 (2011). doi:10.1088/0266-5611/27/12/124003
102. de Silva, V., Vejdemo-Johansson, M.: Persistent cohomology and circular coordinates. In: Hershberger, J., Fogel, E. (eds.) Proceedings of the 25th Annual Symposium on Computational Geometry, pp. 227–236. Aarhus (2009)
103. de Silva, V., Škraba, P., Vejdemo-Johansson, M.: Topological analysis of recurrent systems. In: NIPS 2012 Workshop on Algebraic Topology and Machine Learning, 8 Dec, Lake Tahoe, Nevada, pp. 1–5 (2012)
104. Singh, G., Mémoli, F., Carlsson, G.E.: Topological methods for the analysis of high dimensional data sets and 3D object recognition. In: SPBG, pp. 91–100 (2007)
105. 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 (CVPRW), pp. 45–52. IEEE (2010)
106. Stork, J., Pokorny, F.T., Kragic, D., others: Integrated motion and clasp planning with virtual linking. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3007–3014. IEEE (2013)
107. Stork, J., Pokorny, F.T., Kragic, D., others: A topology-based object representation for clasping, latching and hooking. In: 2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids), pp. 138–145. IEEE (2013)
108. Tahbaz-Salehi, A., Jadbabaie, A.: Distributed coverage verification in sensor networks without location information. IEEE Trans. Autom. Control **55**(8), 1837–1849 (2010)
109. Takens, F.: Detecting strange attractors in turbulence. Dyn. Syst. Turbul. Warwick **1980**, 366–381 (1981)
110. Tausz, A., Vejdemo-Johansson, M., Adams, H.: javaPlex: a research platform for persistent homology. In: Book of Abstracts Minisymposium on Publicly Available Geometric/Topological Software, p. 7 (2012)
111. Tenenbaum, J.B., De Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science **290**(5500), 2319–2323 (2000)
112. Turner, K., Mileyko, Y., Mukherjee, S., Harer, J.: Fréchet means for distributions of persistence diagrams. Discrete Comput. Geom. **52**(1), 44–70 (2014)
113. Vejdemo-Johansson, M.: Sketches of a platypus: persistent homology and its algebraic foundations. Algebraic Topol.: Appl. New Dir. **620**, 295–320 (2014)
114. Vejdemo-Johansson, M., Pokorny, F.T., Skraba, P., Kragic, D.: Cohomological learning of periodic motion. Appl. Algebra Eng. Commun. Comput. **26**(1–2), 5–26 (2015)
115. Vergne, A., Flint, I., Decreusefond, L., Martins, P.: Homology based algorithm for disaster recovery in wireless networks. In: 2014 12th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), pp. 685–692. IEEE (2014)
116. Worsley, K.J.: Local maxima and the expected Euler characteristic of excursion sets of χ^2, F and t fields. Adv. Appl. Probab. 13–42 (1994)
117. Worsley, K.J.: Boundary corrections for the expected Euler characteristic of excursion sets of random fields, with an application to astrophysics. Adv. Appl. Probab. 943–959 (1995)
118. Worsley, K.J.: Estimating the number of peaks in a random field using the Hadwiger characteristic of excursion sets, with applications to medical images. Ann. Stat. 640–669 (1995)
119. Zarubin, D., Pokorny, F.T., Song, D., Toussaint, M., Kragic, D.: Topological synergies for grasp transfer. In: Hand Synergies—How to Tame the Complexity of Grapsing, Workshop, IEEE International Conference on Robotics and Automation (ICRA 2013), Karlsruhe, Germany. Citeseer (2013)
120. Zarubin, D., Pokorny, F.T., Toussaint, M., Kragic, D.: Caging complex objects with geodesic balls. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2999–3006. IEEE (2013)

121. Zomorodian, A., Carlsson, G.: Computing persistent homology. Discrete Comput. Geom. **33**(2), 249–274 (2005)

Author Biographies

Mikael Vejdemo-Johansson received his Ph.D. in Mathematics—Computational Homological Algebra—from the Friedrich-Schiller University in Jena, Germany in 2008. Since then he has worked on research into Topological Data Analysis in research positions at Stanford, St Andrews, KTH Royal institute of Technology and the Jozef Stefan Institute. The bulk of his research is into applied and computational topology, especially persistent cohomology and applications, topological software and applications of the Mapper algorithm. In addition to these topics, he has a wide spread of further interests: from statistical methods in linguistics, through network security and the enumeration of necktie knots, to category theory applied to programming and computer science. He has worked with data from the World Color Survey, from motion capture and from political voting patterns, and published in a spread of journals and conferences, including Discrete and Computational Geometry, IEEE Transactions on Visualization and Computer Graphics and Nature Scientific Reports.



Primoz Skraba received his Ph.D. in Electrical Engineering from Stanford University in 2009. He is currently a Senior Researcher at the Jozef Stefan Institute and an assistant professor of CS at the University of Primorska in Slovenia. He also spent two years as a visiting researcher at INRIA in France. His main research interests are in applied and computational topology as well as its applications to computer science including data analysis, machine learning, sensor networks, and visualization. His work has worked with a variety of types of data. Some examples including cross-lingual text analysis, wireless network link data as well as the study of personal mobility patterns. He has published in numerous journals and conferences including the Journal of the ACM, Discrete and Computational Geometry, IEEE Transactions on Visualization and Computer Graphics, the Symposium of Computational Geometry and the Symposium of Discrete Algorithms.