

Learning from the Shape of Data

Sarita Rosenstock*

To make sense of large data sets, we often look for patterns in how data points are “shaped” in the space of possible measurement outcomes. The emerging field of topological data analysis (TDA) offers a toolkit for formalizing the process of identifying such shapes. This article aims to discover why and how the resulting analysis should be understood as reflecting significant features of the systems that generated the data. I argue that a particular feature of TDA—its functoriality—is what enables TDA to translate visual intuitions about structure in data into precise, computationally tractable descriptions of real-world systems.

1. Introduction. “Learning from the shape of data” describes an expansive portion of scientific activity. One common example is curve fitting, in which a data set is visualized on a two-dimensional grid, and we infer that the underlying mechanism generating the data can be characterized by a function with a similarly shaped plot.

As new techniques are developed to gather, store, and analyze large quantities of high-dimensional information, it is increasingly difficult to visually identify and interpret relevant shapes. While we can scale up familiar curve-fitting tools, such as linear regression, we know there is more structure to be harnessed from large data sets than these methods can reveal.

One relatively new method of identifying “shapes” in data sets is topological data analysis (TDA). Topology is the study of the properties of shapes that are invariant under continuous deformations, such as stretching, twisting, bending, or rescaling. TDA aims to identify the essential “structure” of a data set as it “appears” in an abstract space of measurement outcomes.

The simplest application of TDA is a type of *cluster analysis*—a method of identify “clusters” of data points that are “more similar” to one another than the wider body of data. While this is relatively conducive to interpretation (as revealed “groupings” in the system being analyzed), TDA can also identify more complex shapes including “holes,” “voids,” and “tendrils” with no intuitive interpretation.

*To contact the author, please write to: Australian National University, School of Philosophy; e-mail: sarita.rosenstock@anu.edu.au.

Philosophy of Science, 88 (December 2021) pp. 1033–1044. 0031-8248/2021/8805-0024\$10.00
Copyright 2021 by the Philosophy of Science Association. All rights reserved.

This article is an investigation into why and how the resulting analysis should be understood as reflecting significant features of the systems that generated the data. In particular, I will argue that the relevance and utility of TDA stems from a particular feature: the *functoriality* of the relationship between the shapes it picks out and their symbolic representations.

In section 2 I describe TDA in detail. Section 3 explains what functoriality means and how it justifies the use of TDA despite interpretational challenges. In section 4, I relate this discussion to philosophical work on the contents of and relationships among physical theories. Section 5 examines the role of spatial reasoning in TDA and how its functoriality enables integrating this informal activity into a formal data analytic framework.

2. Topological Data Analysis. The phrase “topological data analysis” is used to refer to a variety of data science practices that use tools from algebraic topology to make inferences about the “shape” of data clouds as they appear in the “space” of possible observations. Here, the term *data* refers to a set of real vectors corresponding to a series of observations. This is an adequate definition for capturing natural language use of the term, but one might object that it does not necessarily capture what data are. One of the goals of TDA is to circumvent some of the arbitrariness involved in presenting data as real vectors. A *data cloud* can thus be thought of as a visual representation of this set of vectors as “points” in a (high-dimensional generalization of) space. The abstract “space” where data live is generally some form of *metric space*, or set X of points (including at least the data points) together with a notion of “distance” $d(\cdot, \cdot)$ between the points. For example, I may have data about the weights of a collection of potatoes. The distance between these data points would just be the pairwise difference in weight between two potatoes according to a fixed unit (e.g., pounds).

A characteristic problem of analyzing large data sets is deciding how to combine many different types of measurements into a shared metric space. I can also add information about the length, color, number of eyes, and so on, for each potato, creating an n -dimensional space, where n is the number of potato attributes. The “distance” between two data points is now some combination of the distances given by weights, lengths, color, and so on. But how should the notions of distance given by each variable combine into “distance” in the total space of possible variable values? The standard way of aggregating one-dimensional metrics into a shared metric space is to imagine each metric as an axis in an n -dimensional Cartesian grid, with distance given by the Cartesian distance as follows. Let $x = (x_1, \dots, x_n)$ and $y = (y_1, \dots, y_n)$ be two sets of potato measurements. Then $d(x, y) = \sqrt{(x_1 - y_1)^2 + \dots + (x_n - y_n)^2}$. Setting aside the fact that there are other viable options for constructing distances from these values, notice that this expression does not include units. Should weight be presented in pounds or tons? Of course we know how to translate between these two units, and we

consider the choice more of notational convenience than being theoretically meaningful. But if we are looking to the “shape” of data for information about the system being measured, the data cloud will look much more “flat” if we use tons rather than pounds. It is thus desirable to consider properties of the data cloud that do not depend on the particular choice of metric space or unit but that are shared by a variety of plausible modeling choices.

Such considerations motivate the use of *topological*, as opposed to geometric, methods. Topology is the mathematical field that studies properties of shapes that remain constant when stretching, twisting, or otherwise deforming. Topologists attend to more general features of metric spaces that would be present under different modeling assumptions, called *topological invariants*. Since data sets are finite, although they may suggest some underlying shape, they likely will not do so uniquely. This is the standard curve-fitting problem in higher dimensions: for any discrete set of points, there are an infinite number of continuous curves (or shapes) that contain (or approximate) the locations of those points. As with the curve-fitting problem, external considerations guide the choice of a continuous object, rather than just the bare, uninterpreted set of data points. One may have a priori reasons to expect that the “right” curve is quadratic, for example.

2.1. Clusters. The simplest example of TDA, and the one most broadly used by data scientists generally, is a type of cluster analysis. The idea behind cluster analysis is to ask: Do my data points naturally divide into sub-categories of data points more similar to one another than the overall space? Such a situation indicates that there is some nontrivial structure underlying the data associated with such groupings, which one may interpret as “natural kinds” in the space. Cluster analysis is in this way closely related to regression analysis—clusters point toward a correlation among variables, one of the main “signals” data scientists hope to read off of large data sets. For example, biological species are sometimes individuated as “homeostatic property clusters” of organisms that are stably more similar to one another than to other organisms (Boyd 1999).

In scientific contexts, external considerations about the type of data under consideration tend to influence how one chooses to carve a data set into clusters. For example, only features considered relevant to fitness will likely factor into the similarity notion that underlies species clustering. Moreover, traditional clustering algorithms such as k-means will require a prespecification of the number of clusters to be identified, which will likely come from preconceived notions of the expected number of groupings. For example, a clustering of voter data might presuppose that voters will split into two clusters along partisan lines.

Even in the absence of such guidance, natural clusters may be easily “seen” when the data are graphed. With larger and higher dimensional data sets to

analyze, these heuristics are less useful, and data scientists would prefer a principled algorithmic approach to clustering. This would amount to a function that takes metric spaces (X, d) —here understood as data sets $X = \{x_1, \dots, x_n\}$ with a notion of “distance” $d(x_i, x_j)$ —as inputs and outputs *partitions* of those data into clusters of data points that are “close together.”

2.2. Constructing Shapes. The most common method to construct a shape from a data cloud is roughly as follows. Enclose each data point in a “ball” of radius ε centered on that point. As ε gets larger, the cloud will cease to look like isolated points and start to gain shape. Once it gets too large, though, we are left with a single shapeless blob. We use this idea to construct a *simplicial complex*, beginning with the data points as vertices.¹ Where two balls intersect, we add an *edge* between them. When three balls intersect, we add a *face* enclosed by the three edges. This process continues, creating higher dimensional *n-faces* where $n + 1$ balls intersect. The result is called a *Čech complex* (see fig. 1).²

This is an intuitively plausible way to construct a discrete shape from a data cloud. A clustering can be “read off” of a Čech complex by grouping data points according to whether they are connected in a single component of the complex. This may be complicated by the presence of noise—a single anomalous data point might connect otherwise robustly distinct clusters. This can either be sidestepped by looking at only regions that are highly connected or avoided altogether by filtering and “cleaning” the data before analysis.

2.3. Holes and Voids. Identifying the clusters of a simplicial complex is a special case of a more general phenomenon of *homology*. Homology is a method of classifying shapes by looking at how many “holes” the shape has. No matter how much you stretch and twist it, a circle will always have a “hole” in it, a sphere will always have a void or cavity, an inner tube will always have the “donut hole” as well as a void in the interior that inflates.

When we look at the connected components of a Čech complex, we are considering the H_0 -homology of the complex (considered as a topological space). We can similarly attend to the H_1 -homology of the complex by looking for “holes” or the H_2 -homology by looking at “cells,” and so on, to higher dimensions with less intuitive interpretations.

Example 1 (Cosmology). Van de Weygaert et al. (2011) study the homology of density level sets of an ensemble of randomly generated cosmic mass distributions. They analyze the evolution of H_1 , H_2 , and H_3 -homology

1. See Hatcher (2002, sec. 2.1), for a precise definition of a simplicial complex.

2. In practice, TDA employs a more computationally tractable approximation thereof, called a *witness complex*. See Carlsson (2009, sec. 2) for details.

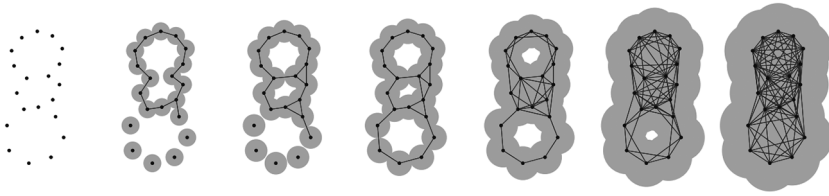


Figure 1. Constructing a Čech complex as ε increases, from Bubenik (2015).

over time in n -body simulations, revealing characteristic patterns of different dark energy models. They show how homology can track cosmological structures of independent interest to physicists, such as matter power spectra and non-Gaussianity in the primordial density field.

2.4. Persistence. The motivating idea behind the construction of a Čech complex is that we can imagine data as being uniformly sampled (with noise) from some underlying “shape” in the metric state space, and we can use these data points to infer the global structure of the “object” we are sampling from. The more samples we look at, the more accurate our picture of the shape will be. For sufficiently small ε -balls, the complex will not have any more structure than the bare data set. Similarly, when the balls get too large, there is nothing more to look at than a giant blob. The “right” choice of ε is at some intermediate size, but how should it be chosen? If we chose an ε that is too small, we will get a shape with a lot more holes, disconnected components, and so on, than we think are meaningful. In other words, we retain some of the noisy features of the data cloud that we were trying to eliminate. But we risk going too far and making ε large enough to obscure both noise and meaningful information from the data.

A natural way to solve this problem is to look at many different choices of ε and use external considerations to decide which gives the best resolution of the data shape. Two more problems arise when we do this, though. For one, the whole point of data analysis is to simplify and compress information about a system, and having a variety of different models we can choose from does not simplify matters. Second, there may be different features that arise at different resolutions that are equally significant, and this multilevel picture can get lost if we have to choose a single model among the many possibilities. For example, data may be dense in some regions but sparse in others, where relevant shapes require larger ε -balls to be “seen.”

The key insight that unlocked the power of TDA was the idea of “topological persistence,” introduced to data analysis in Edelsbrunner, Letscher, and Zomorodian (2002). Briefly: instead of picking a particular resolution to look at, we look at them all but take advantage of a trick from algebraic

topology to connect complexes at different scales in a sophisticated and efficient way. The result is the association of a data cloud with a *persistence module* that encodes how the cloud changes structurally as ϵ increases. Homology is then computed for these modules, and the result is typically expressed as a *homological barcode*, as in figure 2. The “bars” begin when a feature is “born” and end when it “dies.” Short intervals in barcodes are often attributed to either measurement noise or inadequate sampling, whereas long, “persistent” bars are thought to reveal real geometric features of the space being sampled.

Not only is this decomposition more computationally tractable to analyze than (sets of) complexes, but the barcode itself provides a visual summary of behavior as ϵ increases. When the number of features is large, data analysts will also sometime use *persistence diagrams* instead of barcodes.

2.5. Stability. One way to interpret ϵ is as a modeling parameter, corresponding to the resolution or scale we use to construct a shape from the data cloud. The persistent features of a Čech complex are those that are *stable*, or robust under perturbations of the parameter value. Longer bars in barcodes represent features that appear for a wider range of ϵ values, indicating that these features are robust and unlikely to constitute mere noise. Cohen-Steiner, Edelsbrunner, and Harer (2007) made this precise by proving that for a large

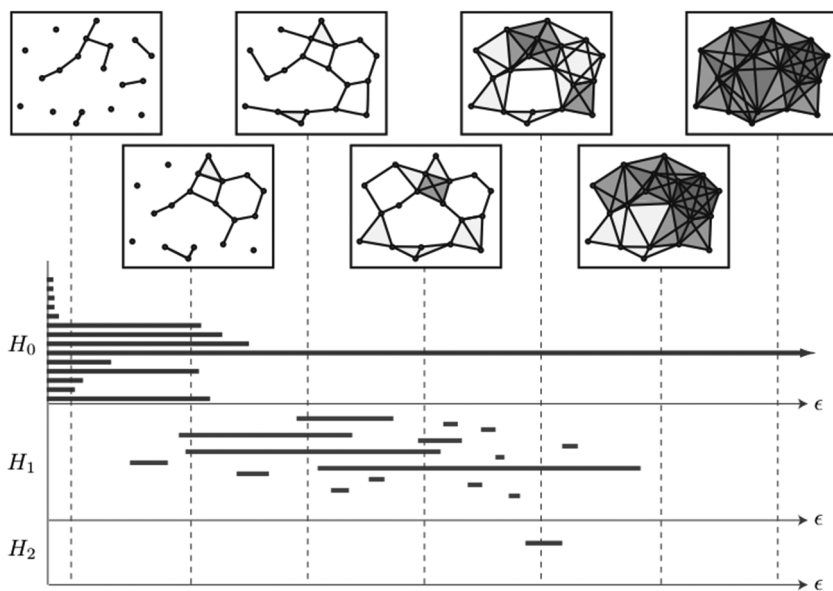


Figure 2. Example of a homological barcode, from Ghrist (2008).

class of constructions (including Čech complexes), persistence diagrams are *stable*, meaning that small perturbations of the initial data set result in correspondingly small changes in the resulting persistence diagram.

We can use this same method to consider stability across other indexing parameters as well at fixed resolution, as in the following example.

Example 2 (Arteries). Bendich et al. (2016) employ TDA to study the structure of arteries in the human brain. They uniformly sample a large number of points from a blood vessel diagram (weighted by thickness of vessel) and construct a Čech complex from this data cloud, analyzing the H_0 and H_1 persistence diagrams over the growing size of ε -balls in the Čech complex. They look at persistent H_0 over a stack of “horizontal slices” of the artery diagram (see fig. 3).

The authors found significant correlation between certain features of these homological barcodes and the age and sex of the subjects, with the age correlation a significant improvement over previous attempts at analyzing similar data. For example, older brains tended to have the longest bars in the latter barcodes.

In this example, persistence is indexed over the parameter of height. One can also analyze persistence of homological features over time.

Example 3 (Time-Series Data). Perea and Harer (2015) demonstrate that persistent H_1 -homology over time can be used to detect periodicity in time-series data by embedding them into a higher dimensional space. Note that in the absence of such an embedding, time series data display no “loops” (since prior points in time are never revisited), so as they stand, they are not conducive to analysis of homology. It is fairly common for data analysts to modify their data to match their methods in this way, rather than the other way around.

We can thus understand persistence modules as assembling a sequence of $(n - 1)$ -dimensional models indexed by an n th parameter, such as resolution or time. Dimensionality reduction is a common feature of data analysis techniques. Data often come in the form of large vectors, and the goal is often to

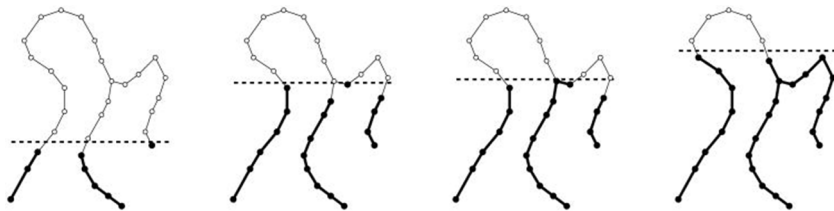


Figure 3. Horizontal slices of the artery diagram, from Bendich et al. (2016).

compress them—express as much of the original information as possible within as few dimensions as possible. This amounts to selecting features or parameters of interest and suppressing the rest in order to highlight general patterns. Reducing data models to two to three dimensions also makes them more visualizable, making them more useful to researchers to observe patterns, as well as easier to communicate to the public. Persistence modules provide the benefits of low-dimensional visualizability without throwing away the information in the extra dimensions.

3. Functoriality. Most practitioners will admit that the interpretation of homology in data is unclear. While increasing in popularity of late, TDA (beyond mere cluster analysis) is still relatively niche. It is often reserved for situations in which traditional data analysis tools have failed to bear fruit, and TDA is one of many attempts to gain insight into the data.

Data scientists rarely feel the need to justify their use of TDA beyond the fact that it seemed to pick up on a relevant pattern in a particular situation. But when pressed, or in more comprehensive theoretical contexts, the use of TDA is usually explained by the fact that homology has a particularly nice property that makes it a reliable data analysis tool: *functoriality*.

To understand this, we need to look a bit deeper into how TDA functions. TDA summarizes the shape of a Čech complex built from a data cloud in terms of a *homology group* $H_n(X)$. For each group, $H_n(X)$ essentially characterizes how many “holes” are present in each dimension. This makes it easy to describe the shape computationally, as groups are more easily described symbolically than as shapes. But in order for this symbolic representation to be useful, we need to be able to identify which “holes” in our complex correspond to which symbolic representation, and we need to be able to track the holes as we evolve the complexes. We can do this because homology is *functorial* in the sense that, more than just translating complexes to groups, it tells us how to translate maps between complexes into maps between groups while preserving all relevant topological information.

The functoriality of homology enables us to do three important things, which are essential to its utility in analyzing data: identify local structures, connect complexes as parameters vary, and compare complexes constructed from different samples. We can identify local structures via inclusion maps that pick out particular clusters, holes, and voids. We can then evolve these complexes by varying parameters of interest and see which features persist. Finally, we can perform an additional robustness check on our results by comparing clusters generated with different subsamples of our data, in a way analogous to bootstrapping in statistics (Chazal et al. 2015).

Thus, data scientists study persistent homology, not because they think of “counting holes” as the right way to characterize data but rather because TDA has a particular feature—functoriality—that makes it a reliable tool

to use. Since persistent homology has this nice property, data scientists will often shoehorn questions about data into the shape of a homology problem in order to make it tractable. For example, they might add extra edges to a Čech complex to turn open chains into closed loops. Or they might chose a particular dimensional reduction in which loops arise, as in Perea and Harer (2015).

One can also modify TDA to examine how clusters are shaped. For example, “tendrils” emanating from the core of a cluster can be tracked via the persistent H_0 -homology of the resulting data cloud once that core is removed. Nicolau, Levine, and Carlsson (2011) use this technique to classify breast cancer types (see fig. 4). While the recent proliferation of these methods might be dismissed as mere hammer nailing, it should rather be said that since we have very few tools to work with, we had better hope this problem can become nail shaped.

If I am correct about the significance of TDA’s functoriality, then we should expect that other fruitful data analytic methods can be understood functorially. Indeed, Bubenik and Scott (2014) express persistent homology as a special case of a more general kind of functor, and Carlsson and Mémoli (2013) demonstrate how a functorial account of clustering algorithms (including H_0 persistent homology) provides conceptual clarity.

4. Category Theory. The role of functoriality in justifying the use of TDA is suggestive of recent literature in the philosophy of physics advocating for a functorial account of intertheoretic relations. This literature is inspired by Halvorson (2013), who argues that one should understand the content of a scientific theory as a *category* of models of that theory, that is, as a collection



Figure 4. Visualization of data featuring tendrils.

of theoretical models plus relationships (structure-preserving functions) between the models. On this view, the appropriate way to understand relationships between theories is using a *functor*—a map that takes models to models and relations to relations in a consistent way. Once framed in this way, philosophers can use tools from *category theory* to enrich their understanding of these theories and how they relate to one another (Weatherall 2017; Rosenstock 2019).

We can conceive of TDA as a special case of this general category theoretic framework for characterizing scientific theories or, as I prefer to think of them, *representational frameworks*. We begin with a “metric space” representational framework for our empirical data. This consists of (finite) metric spaces, along with relationships between metric spaces (isometries, embeddings, etc.), forming category *FinMet*. We also have a “topological” representational framework of “shapes” that our data might have and structure-preserving maps between them forming a category *Comp* of simplicial complexes. And we have an algebraic category, *HomAlg*, of homological algebras.

In this language, we articulate a “reading” of shapes from a data set as a functor $F: \text{FinMet} \rightarrow \text{Comp}$, such as the functor F_δ that takes a metric space its Čech complex of radius δ . We can transform this topological framing into an algebraic framing via a functor from *Comp* to *HomAlg* (the “homology” functor). And we can construct a category *PDiag* of persistence diagrams, associated with our underlying data model again by a functor from *FinMet* to *PDiag*.

There are lessons to be learned from this relationship between TDA and the philosophy of physics literature in both directions. Philosophers benefit from a fruitful example outside of physics, one that incorporates many “levels” of abstraction from initial data to more abstract representations. Conversely, formal philosophical work can help elaborate the sense in which theoretical content is “preserved” in these functorial transformations. In particular, Rosenstock (2021) illustrates how reflection on the structure of a data set influences and constrains the ways in which it can be clustered.

5. Spatial Inference. The goal of data analysis is to identify patterns in data that provide concise, comprehensible summaries of the system that point toward features of significance in broad classes of systems. Such recognition of patterns of sufficient generality without overfitting is the holy grail of artificial intelligence and machine learning research. In the meantime, scientists rely heavily on visual intuition to guide inquiry, experimenting with parameters and data filtering until it “looks right.”

TDA removes some of the arbitrariness of this process by enforcing a consistent methodology to the identification of patterns once these discretionary setup choices are made. But intuitions are not abandoned entirely at this stage, since the resulting analysis still has to fit with preconceived

notions of natural categories and interesting patterns in order to be of interest to practitioners. Patterns found through random applications of TDA might lead scientists to look for corresponding features of interest in a system, but if these cannot be found, the shapes identified in the data remain merely curiosities. In example 2, if barcodes did not track gender and age but some other feature that we do not independently classify as a natural kind, they would likely be omitted from the published analysis.

The difficulty of interpreting higher dimensional homology thus requires extensive human discretion to be empirically useful. TDA is a second-line resource for data that are particularly intractable to analyze, which puts creativity at the center of its application. We might wonder whether such an informal process of intuitive speculation about the shape of data can be incorporated into a formal epistemic story about the structure of topological data models. Here, we can learn much from the vast literature on diagrammatic reasoning in Euclidean geometry. Critics of the rigor of reasoning from diagrams in geometric ‘proofs’ point to the fact that such proofs use a particular illustration to make an inference about all possible illustrations. However, philosophers of mathematical practice have recently come to appreciate the role of diagrams in generating and communicating geometric knowledge. Manders (2008) argues that ancient geometers were careful to rely on diagrams only for demonstrations about what he calls *co-exact* features—those that are relatively insensitive to the range of variation in possible visual representations, such as part-whole and boundary-interior relationships (and of course, homology). Mumma (2010) takes this a step further and develops a formal account of Euclidean proofs that includes both sentential and diagrammatic components.

Similarly, data analysts are concerned with ensuring that inferences about data rely only on real structural features of observations, rather than incidental features of how data visualized. At issue is the level of generality one can adopt when making inferences from a single visual representation of data, picked somewhat arbitrarily from an ensemble of possible alternative, equally valid representations. TDA resolves this issue by requiring that the analyzed features of data models be functorial with respect to maps that preserve what they take to be the relevant structural features of models and persistent across parameters when the “right” value is not known.

6. Conclusion. This article argues that the functoriality of homology is critical to TDA’s utility in revealing and interpreting structural features of data sets. In brief, topological features of data sets are visually salient to humans and aid in our reasoning in understanding. The functoriality of persistent homology ensures that reasons we had for thinking topological features were meaningful are preserved in the translation from data cloud to homological barcode, while enabling various robustness tests on the resulting analyses.

There are promising future directions for exploring the relationship between TDA and recent philosophical work on the content of and relationships among physical theories.

REFERENCES

- Bendich, P., J. S. Marron, E. Miller, A. Pieloch, and S. Skwerer. 2016. "Persistent Homology Analysis of Brain Artery Trees." *Annals of Applied Statistics* 10 (1): 198–218.
- Boyd, R. 1999. "Homeostasis, Species, and Higher Taxa." In *Species: New Inter-disciplinary Essays*, ed. R. A. Wilson, 141–85. Cambridge, MA: MIT Press.
- Bubenik, P. 2015. "Statistical Topological Data Analysis Using Persistence Landscapes." *Journal of Machine Learning Research* 16:77–102.
- Bubenik, P., and J. A. Scott. 2014. "Categorification of Persistent Homology." *Discrete and Computational Geometry* 51 (3): 600–627.
- Carlsson, G. 2009. "Topology and Data." *Bulletin of the American Mathematical Society* 46 (2): 255–308.
- Carlsson, G., and F. Mémoli. 2013. "Classifying Clustering Schemes." *Foundations of Computational Mathematics* 13 (2): 221–52.
- Chazal, F., B. T. Fasy, F. Lecci, A. Rinaldo, A. Singh, and L. Wasserman. 2015. "On the Bootstrap for Persistence Diagrams and Landscapes." *Modeling and Analysis of Information Systems* 20 (6): 111–20.
- Cohen-Steiner, D., H. Edelsbrunner, and J. Harer. 2007. "Stability of Persistence Diagrams." *Discrete and Computational Geometry* 37 (1): 103–20.
- Edelsbrunner, H., D. Letscher, and A. Zomorodian. 2002. "Topological Persistence and Simplification." *Discrete and Computational Geometry* 28:511–33.
- Ghrist, R. 2008. "Barcodes: The Persistent Topology of Data." *Bulletin of the American Mathematical Society* 45 (1): 61–75.
- Halvorson, H. 2013. "The Semantic View, If Plausible, Is Syntactic." *Philosophy of Science* 80 (3): 475–78.
- Hatcher, A. 2002. *Algebraic Topology*. Cambridge: Cambridge University Press.
- Manders, K. 2008. "The Euclidean Diagram, 1995." In *The Philosophy of Mathematical Practice*. Oxford: Oxford University Press.
- Mumma, J. 2010. "Proofs, Pictures, and Euclid." *Synthese* 175 (2): 255–87.
- Nicolau, M., A. J. Levine, and G. Carlsson. 2011. "Topology Based Data Analysis Identifies a Subgroup of Breast Cancers with a Unique Mutational Profile and Excellent Survival." *Proceedings of the National Academy of Sciences* 108 (17): 7265–70.
- Perea, J. A., and J. Harer. 2015. "Sliding Windows and Persistence: An Application of Topological Methods to Signal Analysis." *Foundations of Computational Mathematics* 15 (3): 799–838.
- Rosenstock, S. 2019. "A Categorical Consideration of Physical Formalisms." PhD diss., University of California, Irvine.
- . 2021. "Clustering Schemes for Diverse Data Models." Unpublished manuscript, Australian National University.
- van de Weygaert, R., et al. 2011. "Alpha, Betti and the Megaparsec Universe: On the Topology of the Cosmic Web." In *Transactions on Computational Science XIV: Special Issue on Voronoi Diagrams and Delaunay Triangulation*, ed. M. L. Gavrilova, C. J. K. Tan, and M. A. Mostafavi, 60–101. Berlin: Springer.
- Weatherall, J. O. 2017. "Categories and the Foundations of Classical Field Theories." In *Categories for the Working Philosopher*, ed. E. Landry, 329–48. Oxford: Oxford University Press.