Topological Data Analysis and Applications

Samuel Salander

April 2023

1 Introduction

Topological data analysis (TDA) is a data analysis framework that employs techniques from topology. It can be extremely difficult to get meaningful insights from high-dimensional and noisy data. TDA addresses this problem quite elegantly with defining characteristics of topology like invariance with respect to metric and functoriality [1]. These, respectively, lend the techniques an unbiasedness and adaptability that are very desirable in certain circumstances when compared to other dimensionality-reduction techniques like Support Vector Machines.

TDA draws upon tools from algebraic topology and similar fields related to the concept of shape in order to investigate the shape of data. The main tool that has emerged from this framework is persistent homology, which is elementary algebraic topology applied to point cloud data.

For example, the trajectory of a simple predator-prey system governed by the Lotka–Volterra equations forms a closed circle in state space, which TDA gives us a way to indirectly see [4]. In our own investigation, the system will be stochastically changing over time, and therefore we will use the changes in

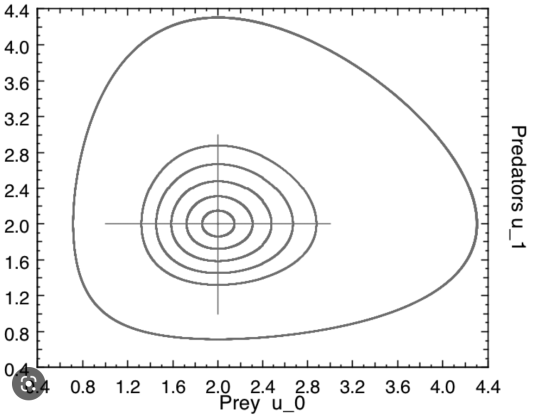


Figure 1: Lotka-Volterra state space diagram

the shape of the data to guide us rather than seek a unifying shape as in this example.

Algorithms for data analysis often require setting various parameters. Without prior knowledge, this can be an arbitrary and difficult task. A large insight of persistent homology is to use the information from all parameter values by encoding this information into an understandable form, more specifically, into a homology group. The assumption made is that features that persist for a wide range of parameters are reliable features.

In this paper, we will use public stock market data to demonstrate the implementation of TDA techniques and visualize the results. As we will discuss later, a weakness of TDA is large data volume, and the dataset we will work with is sizeable but not vast accordingly.

2 Theory

2.1 Explanation of Terms

We provide here a list of useful mathematical terms that we will use extensively in this paper.

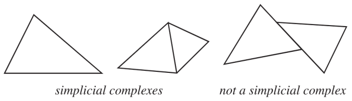


Figure 2: Simplicial Complex

- Point Cloud: any dataset can be viewed as a point cloud by plotting each point in an N-dimensional space with axes defined as the N variables of the dataset.

- Simplicial Complex: a simplicial complex K in R^N is a collection of triangles in R^N such that every face of a triangle of K is in K, and the intersection of any two triangles of K is a face of each of them.

- Homology: The set of topological invariants of a topological space, represented by its homology groups, the k^{th} of which describes the number of holes in the space with a k-dimensional boundary. For example, the homology groups of the n-dimensional sphere are the group of integers for k=0,n and the trivial group otherwise since it has a single connected component and exactly one n-dimensional hole. The rank of a homology group is known as a Betti number.

- Nerve Complex: The set of finite subsets of the index set of a family of sets.

- Homotopy Equivalence: two spaces X and Y are homotopy equivalent if they can be transformed into one another by bending, shrinking, and expanding.

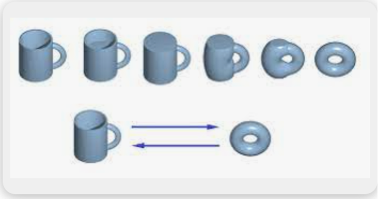


Figure 3: Example of homotopically equivalent spaces, from Wolfram Alpha

- Vietoris–Rips Complex: A simplicial complex constructed from a point cloud in any metric space which is meant to capture topological information about the point cloud it is drawn from. It is the nerve of the set of e-balls centered at points of the point cloud (for some radius e $>$ 0). By a well-known lemma about nerves, so-called Čech complexes are homotopy-equivalent to the union of balls in the complex, which gives it an advantage over the less computationally expensive Vietoris–Rips complex, though that computation factor is significant and so we use Vietoris-Rips. Recently, the so-called α-complex and witness complex have been used in an attempt to increase the scalability of TDA while maintaining the useful homotopy-equivalence property.

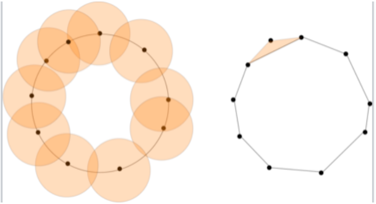


Figure 4: Example of Vietoris-Rips Complex, from Wolfram Alpha

- Filtration: Nested complexes resulting from the application of some pruning function on a complex, this function applied stochastically typically representing the passage of stochastic time.
- Persistent Homology Group: a multiscale analog of a homology group, which tracks "persistent" topological features along a filtration. Its ranks are called persistent Betti numbers. It is defined for any k as the product of the k^{th} homology groups of a complex along the filtration.
- Persistence Module: a vector space U_t for each t $\in Z$ and a linear map u_t^s: $U_s \to U_t$, whenever s<=t such that u_t^t=1 for all t and $u_t^s u_s^r$=u_t^r when r<=s<=t

- Structure Theorem:for a finitely generated persistence module C with field F coefficients, the persistence module is given by:

$$H(C;F) \simeq \oplus_i x^{t_i} * F[x] \oplus (\oplus_j x^{r_j} * (F[x]/(x^{s_j} * F[x]))) \tag{1}$$

- Persistence Barcode: a multiset of intervals in R derived from the persistence module via the structure theorem, representing the beginning and end of the appearance of a topological feature in a filtration.

- Persistence Diagram: A diagram of points derived from a persistence barcode with points above the diagonal in the first quadrant of R^2. Each point (a,b) represents a feature appearing at stage a and persisting to stage b in the filtration.

- Wasserstein Distance: Between persistence diagrams X and Y is defined as

$$W_p[L_q](X,Y) = \inf_{\phi:X\to Y}[\sum_{x\in X}(||x-\phi(x)||_q)^p]^{1/p} \tag{2}$$

where ϕ ranges over bijections from X to Y.

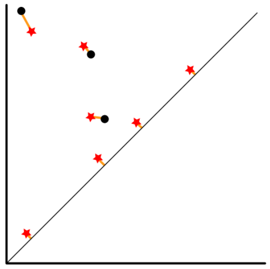


Figure 5: Example of pairing used for distance calculations between persistence diagrams, from Chaval (2013)

- Bottleneck Distance: Between diagrams X and Y, given by setting p=∞ in Wasserstein definition:

$$W_\infty[L_q](X,Y) = \inf_{\phi:X\to Y} \sup_{x\in X}(||x-\phi(x)||_q) \tag{3}$$

- Stability: A measure of robustness to noise, bounded for a simplicial complex X, functions f,g:X→R, and D is a map taking a continuous function to the persistence diagram of its k^{th} homology:

$$W_\infty(D(f),D(g)) <= ||f-g||_\infty \tag{4}$$

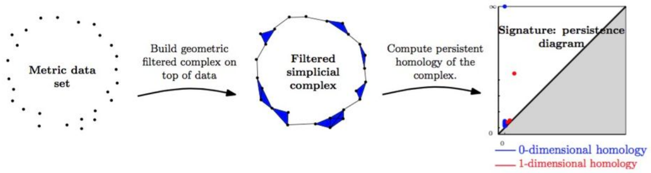

Metric data set
Build geometric filtered complex on top of data
Filtered simplicial complex
Compute persistent homology of the complex.
Signature: persistence diagram
∞
0
0
0-dimensional homology
1-dimensional homology

Figure 6: Pipeline from data to Persistence Diagram, from Chazal (2013)

2.2 General Pipeline and Intuition

As suggested by the ordering of these terms, the methods of modern TDA are built upon a pipeline of transforming data from a point cloud to a persistence barcode, or equivalently, a persistence diagram (see Figure 6). A dataset is represented as a point cloud (see implementation for details, these vary according to exact application), and some kind of complex is constructed in its place (we will use Vietoris-Rips because it is easy to understand and has the computational advantage, although as discussed before, other options exist), and the structure theorem is applied to get the persistence diagram, or equivalently the persistence barcode or persistence Betti numbers [4].

2.3 Persistent Homology Theory

Here we give a more thorough treatment of the important tool of persistent homology used in TDA. We write a filtration of simplicial complexes (read Vietoris-Rips complexes for our purposes) as

$$K_0 \subseteq K_1 \subseteq ... \subseteq K_n \tag{5}$$

Then the precise definition of the p^{th} persistent homology group $H_p^{i,j}$ is:

$$H_p^{i,j} = Z_p(K_i)/(B_p(K_j) \cap Z_p(K_i)) \tag{6}$$

for cycle and boundary groups Z and B. The elements of these groups are called homology classes. These classes are said to be born in K_i if it is not contained in K_{i-1} and die if it merges with another older class. The specification that the class it merges with be older for its death to occur is called the Elder Rule [2]. For an infinite field, the infinite number of class elements always have the same persistence, the collection over all classes of such intervals is not particularly meaningful. Instead, the desired multiplicities and a multiset of intervals in the extended real line are given by the structure theorem. This multiset is what we introduced above as the persistence barcode, also easily visualized in the form of a persistence diagram [2].

2.4 MAPPER Theory

An interesting variant of TDA is given by the MAPPER algorithm of Carlsson et al. This method produces a graph, but nonetheless falls under the umbrella of

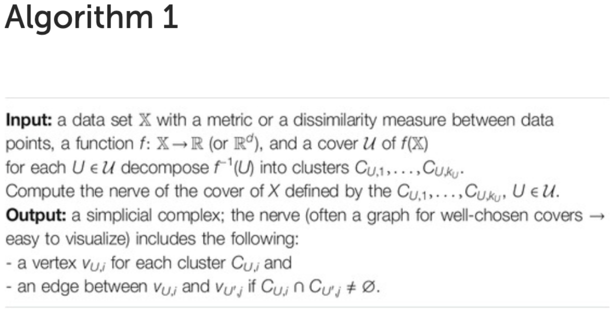
TDA. Together, they are the workhorses of the entire family of techniques and are far-and-away the most important with which to be familiar [3]. We give the following high-level description of MAPPER, from Dey and Wang (2015):Let X and Z be topological spaces and let f : X → Z be a continuous map. Let

$$U = U_{\alpha \alpha \in A} \tag{7}$$

be a finite open covering of Z. Consider the cover

$$f^*(U) = f^{-1}(U_\alpha)_{\alpha \in A} \tag{8}$$

of X, which is called the pull-back of U along f. The mapper construction arising from these data is nerve complex of the pullback cover M(U,f) := N(f^*(U)). To use mapper we need a metric space X, a filter function f : X → R or X → Z, and a covering U of Z. The output of the algorithm is a graph whose nodes represent clusters of data and whose edges denote clusters that share data, which yields information about global features in the data [3]. The choice of the function f above can be suited to the dataset and what one hopes to highlight. For example, one might use a simple distance function for data around one-dimensional structures, or eigenfunctions of graph Laplacians, eccentricity functions, or centrality functions to reduce the need for prior information about the dataset.



Algorithm 1

Input: a data set $\mathbb{X}$ with a metric or a dissimilarity measure between data points, a function f: $\mathbb{X} \rightarrow \mathbb{R}$ (or $\mathbb{R}^d$), and a cover $\mathcal{U}$ of $f(\mathbb{X})$
for each $U \in \mathcal{U}$ decompose $f^{-1}(U)$ into clusters $C_{U,1}, \ldots, C_{U,k_U}$.
Compute the nerve of the cover of X defined by the $C_{U,1}, \ldots, C_{U,k_U}$, $U \in \mathcal{U}$.
Output: a simplicial complex; the nerve (often a graph for well-chosen covers → easy to visualize) includes the following:
- a vertex $v_{U,i}$ for each cluster $C_{U,i}$ and
- an edge between $v_{U,i}$ and $v_{U',j}$ if $C_{U,i} \cap C_{U',j} \neq \emptyset$.

Figure 7: Overview of the MAPPER algorithm, from Fasy (2014)

3 Algorithms

In this section, we apply MAPPER and persistent homology techniques to different chronological sections of an expansive set of stock market data. This data is taken from the Python package yfinance, which in turn scrapes the data from Yahoo Finance. The idea for this investigation was presented in an article by da Oliveira [5], but ours is a more faithful predictive exercise in that it draws its data only from the past.

3.1 MAPPER Implementation

We begin with the MAPPER algorithm proposed by Carlsson et al. We import the Python packages kmapper, UMAP-Learn, sklearn, NumPy, and Matplotlib. We normalize the adjusted (accounting for stock splits and other irregular factors) closing price of each stock, calculate the percent return for each stock over the duration of our sample, perform the mapping, and color code the nodes according to the mean, maximum, minimum, or standard deviation of the percent return for the data in each node. Note also that adjusting the beginning and ending time for the period under investigation is easy to change if one is interested in a different period.

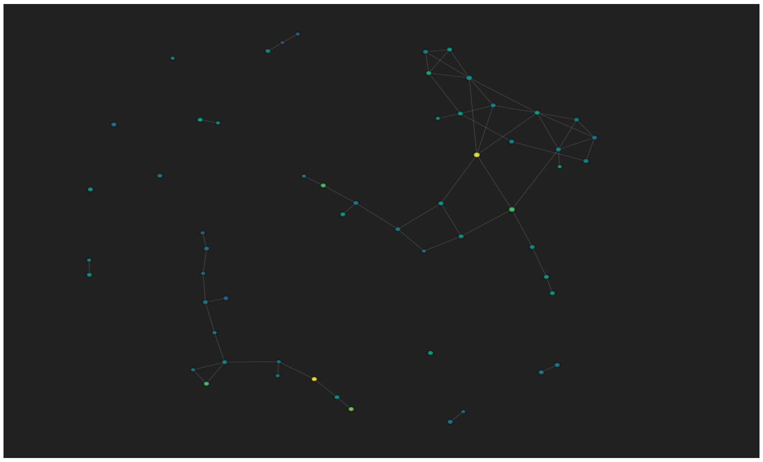


Figure 8: Output of MAPPER on Stock Data with Max node color function

3.2 Persistent Homology Implementation

Now we investigate the persistent homology method. Here, we use the package Ripser, a persistence homology software originally written in C++ (and requiring Cython to run on Python), and Persim as an accompanying tool. We input a time range of stock data, and for every day of data compute the Vietoris-Rips complex of the data for the previous W number of days and the W days before that (where in our code we use W=20), and output a graph of the Wasserstein distance between the two for that day. The results are somewhat mixed, with it working quite well for the crash of 2008 and still succeeding but less spectacularly for 2020. The intuition here is that before the visible crash in price takes place, the topological features of the data are changing more rapidly underneath the surface.

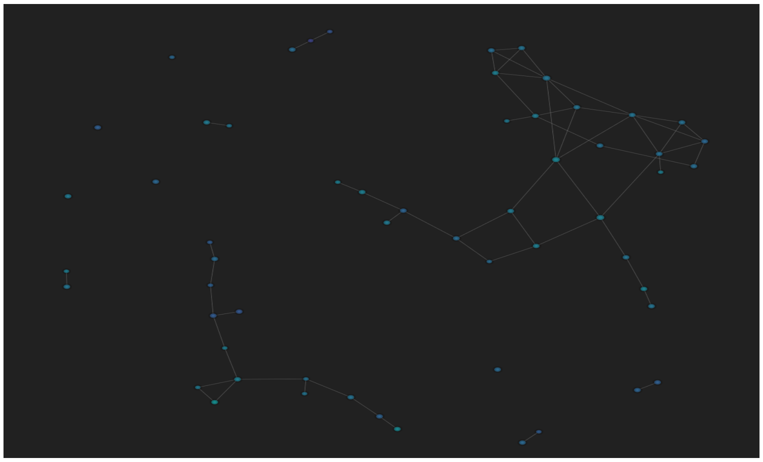


Figure 9: Output of MAPPER on Stock Data with Mean node color function

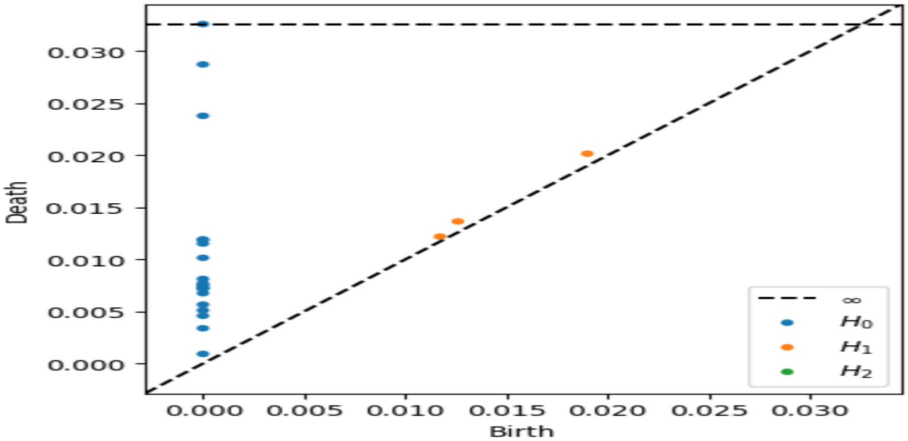


Figure 10: An Example Persistence Diagram for a Day in the Dataset

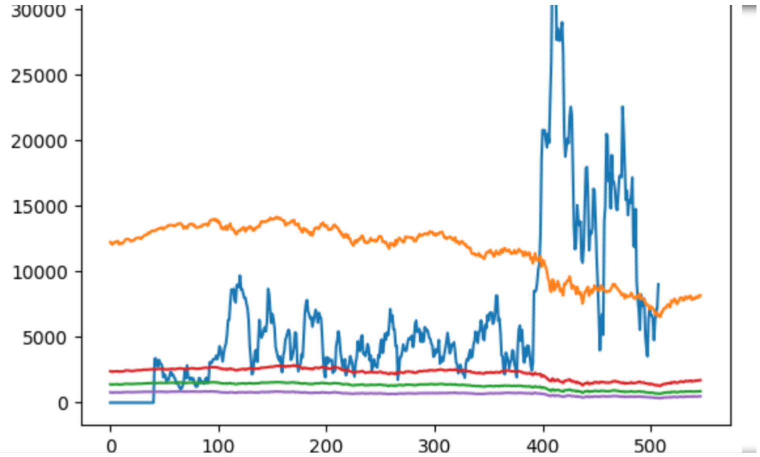


Figure 11: Output of Wasserstein test for the Crash of 2008. Here it works quite well at indicating a substantial drop in prices in the following weeks and months.

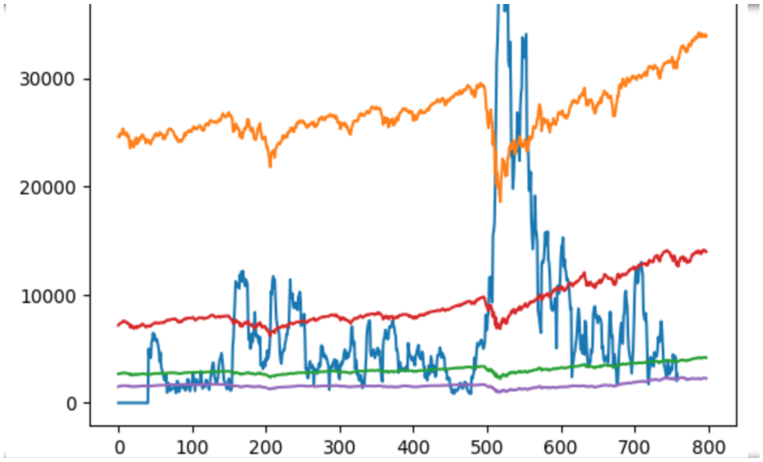


Figure 12: Output of Wasserstein test for the Crash of 2020. Here it works reasonably well at predicting the drop, though peaks in the middle of the crash, making it a bit less useful. Even here, though, we see a useful prediction for the mini-crash of late 2018.

4 Conclusion

We have seen the usefulness of TDA as a dimensionality-reduction and analysis tool. Finding relevant low-dimensional features in a dataset is a vague and ill-defined problem, and the motivation of TDA is to leverage the unbiasedness and other properties of topological operations to deliver sound and demonstrable low-dimensional features and information about how long they last. However, TDA has thus far found limits to its practical use. In particular, its main weakness is scaleability. There is a tradeoff in the selection of complex type in the first step of the TDA pipeline, with our Vietoris-Rips complex providing ease of computation at the expense of useful mathematical properties [2]. As mentioned earlier, this crucial step of the pipeline is the subject of much research now, and it may be a key advance to make these techniques more realistic for a wider range of applications involving enormous amounts of data. As it stands currently, TDA is a mathematically elegant concept that still needs to be expanded upon for application.

5 Bibliography

- 1. Carlsson, G. (2009). Topology and Data. Bull. Amer. Math. Soc. 46 (2), 255–308. doi:10.1090/s0273-0979-09-01249-x

- 2. Front. Artif. Intell., 29 September 2021 Sec. Machine Learning and Artificial Intelligence Volume 4 - 2021 — https://doi.org/10.3389/frai.2021.667963

- 3. Dindin, M., Umeda, Y., and Chazal, F. (2020). "Topological Data Analysis for Arrhythmia Detection through Modular Neural Networks," in Canadian Conference on Artificial Intelligence (Springer), 177–188. doi:10.1007/978-3-030-47358-717

- 4. Fasy, B. T., Lecci, F., Rinaldo, A., Wasserman, L., Balakrishnan, S., and Singh, A. (2014). Confidence Sets for Persistence Diagrams. Ann. Stat. 42 (6), 2301–2339. doi:10.1214/14-aos1252

- 5. Da Oliveira et al. (2019) Detecting stock market crashes with topological data analysis