Lecture Notes on Topological Data Analysis

STAT 37411, The University of Chicago

January 13, 2022

Contents

Preface

This set of lecture notes is being produced from a set of lectures on Topological Data Analysis given in the winter 2022 offering of STAT 37411 at The University of Chicago, taught by Brad Nelson. Students are being asked to contribute a transcription of one lecture each to these notes. Students who have contributed so far are:

Additional online material for this course can be found at stat37411.github.io.

These notes are a work in progress. Suggestions or corrections can be sent to:
Brad Nelson at bradnelson@uchicago.edu

Chapter 1

Introduction

In this chapter, we first give some motivation for the sorts of problems that topological data analysis might try to solve, and then pursue a non-rigorous overview of persistent homology.

1.1 Motivation

Topological data analysis (TDA) is a field that uses topological techniques to summarize and understand data. One line of work in TDA seeks to understand structure in data, for instance identifying clusters or holes in data. For example, in figure 1.1 we see points sampled near a figure-8. Our human intuition looking at this figure is to see that while there is some randomness in the points, they do indeed lie near a figure-8. We might say that the underlying space has a single connected component despite the fact that the sample is a discrete set of 200 points, and two holes where points are not sampled, despite the fact that there are many gaps or holes between points. One of the goals of topological data analysis is to produce a summary, or mathematical signature, for the point cloud which will contain this same qualitative information. This can be used as part of an exploratory data analysis pipeline, and the results can be used to inform models of the data. Perhaps one of the most well known applications of these techniques in the field was the discovery of a Klein bottle in natural image image patches [2]. This line of work leads to many interesting questions such as under what conditions we can recover the topological features of a manifold, and how robust these topological signatures are to perturbations of data [3].

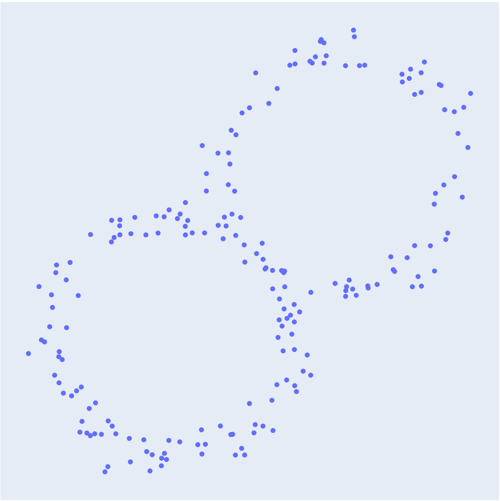

Figure 1.1: 200 points sampled near a figure-8.

Another aspect to topological data analysis is to create topologically meaningful features for machine learning. For example, in figure 1.2, we see images of the digits zero, ”0”, and one, ”1”. Topologicially, the representations these two digits are different; both have a single connected component, but ”0” has a hole in the middle, whereas ”1” does not. A topological machine learning model should be able to distinguish between these two digits based on this observation alone. Of course, reality is much more messy, as there may be gaps in a pen stroke, extreme variations in handwriting, or artifacts of digitization. This line of work has led to a variety of applications in fields such as materials discovery [4] and molecular property prediction [1].

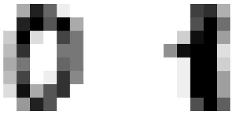

Figure 1.2: Left: a digital image of an hand-written zero. Right: a digital image of a hand-written one.

1.2 A Topological Signature For Point Cloud Data

We'll focus on the problem of obtaining a topological signature for point clouds such as in figure 1.1. A natural approach which we employed visually is to group together points based on some notion of proximity. In practice, we can build a generalization of a graph called *simplicial complex* based on this notion of proximity. Briefly, simplicial complex $\mathcal{X}$ consists of a collection of vertices (0-simplices) $\mathcal{X}_0$, edges (1-simplices) $\mathcal{X}_1$, triangles (2-simplicies) $\mathcal{X}_2$, and generally convex hulls of $k+1$ points (k-simplices) $\mathcal{X}_k$. We denote a k-simplex as $(x_0, \dots, x_k)$, where $x_0, \dots, x_k \in \mathbf{X}_0$, which can be thought of as the convex hull of these $k+1$ vertices. Given a data set $\mathbf{X}$ and a metric $d : \mathbf{X} \times \mathbf{X} \to \mathbb{R}_+$, the Vietoris-Rips complex $\mathcal{R}(\mathbf{X}; r)$ has $\mathcal{R}(\mathbf{X}; r)_0 = \mathbf{X}$, an edge set

$$\mathcal{R}(\mathbf{X}; r)_1 = \{(x_i, x_j) \mid d(x_i, x_j) \leq r\}, \tag{1.1}$$

and all possible k-simplices $k = 2, \dots,$ built from this edge set.

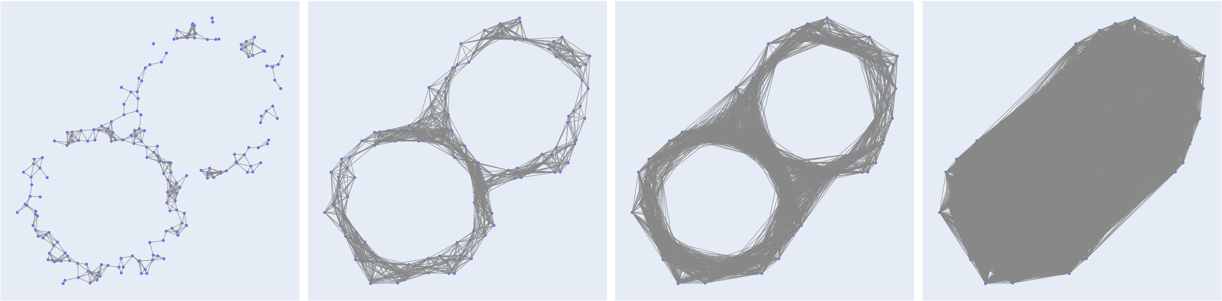

Figure 1.3: Vietoris-Rips complexes at various connectivity parameters. From left to right: $r = 0.2$, $r = 0.5$, $r = 1.0$, and $r = 2.0$.

In figure 1.3 we see these Vietoris-Rips complexes built on our figure-8 samples. When the connectivity parameter r is too small (on the far left), we do not yet have a single connected component. When r is too large (on the far right), all structure is filled in. However, for a range of parameters we see something that looks a lot like a figure-8 with a single connected component and two loops.

If we focus on a parameter r which captures the desired topology of the space, we can obtain an algebraic signature of the space called *homology.* At a high-level, homology in dimension k, denoted $H_k(\mathcal{X})$, is a vector space which captures topological information about k-dimensional features of $H_k(\mathcal{X})$. The dimension of $H_k(\mathcal{X})$, $\dim H_k(\mathcal{X})$, also known as the k-th Betti number of $\mathcal{X}$ β_k, counts k-dimensional topological features of $\mathcal{X}$: connected components in dimension 0, loops in dimension 1, and k-dimensional voids in dimension k. For our figure-8, we have $\dim H_0 = 1$, and $\dim H_1 = 2$.

Homology is more powerful than a tool for counting topological features. It is a *functor*, which means that if we have a map between topological spaces $f : \mathcal{X} \to \mathcal{Y}$, homology produces an *induced map* $H_k(f) : H_k(\mathcal{X}) \to H_k(\mathcal{Y})$ which encodes a notion of how topological features in $\mathcal{X}$ map to features in $\mathcal{Y}$. This is critical for what we need next.

We still have a problem, which is that we had to choose the parameter r. Instead of focusing on choosing the exact right value of r to use, it is easier to just consider all possible values of r and identify robust features which *persist* for a large range of parameters. When considering all possible values of r, we obtain a *filtration* of Vietoris-Rips complexes

$$\mathcal{R}(\mathbf{X}; r_0) \subseteq \mathcal{R}(\mathbf{X}; r_1) \subseteq \dots \tag{1.2}$$

Note that since the sample $\mathbf{X}$ is finite, there are only a finite set of r_i where the Vietoris-Rips complex actually changes. Each inclusion is a map from the smaller simplicial complex to the next, so the homology functor produces a diagram of vector spaces connected by linear maps

$$H_k(r_0) \xrightarrow{H_k(\iota)} H_k(r_1) \xrightarrow{H_k(\iota)} \dots \tag{1.3}$$

where $H_k(\iota)$ is the map induced by inclusion.

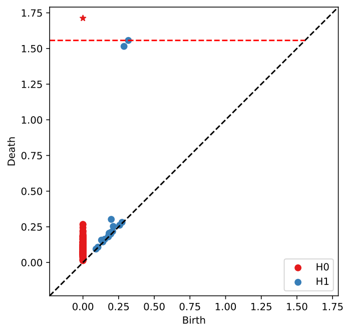

Figure 1.4: Persistence diagram for figure-8 sample, $r = 0.5$. Persistence pairs for H_0 are red, and for H_1 are blue.

If the diagram in equation (1.3) had only a single linear map, we could classify it up to change of basis in each homology vector space by its rank. For longer chains of linear maps, we can generalize the rank to a classification in terms of what is called a persistence barcode, or persistence diagram. In short, we obtain a set of birth-death pairs $H_k(\mathcal{R}(\mathbf{X}; r)) \cong \{(b_i, d_i)\}_{i=1}^n$ where the pair (b_i, d_i) has an associated homology vector that appears at parameter b_i for the first time, and then maps through the diagram until entering the kernel of the map at parameter d_i. The set of pairs can be visualized in a two-dimensional plane, as is done in figure 1.4 for the Vietoris-Rips filtration built on our sampling of the figure-8 in figure 1.1. Each point corresponds to a homology vector (red points are H_0 vectors and blue points are H_1 vectors), where the birth parameter of the vector is indicated on the horizontal axis, and the death parameter is indicated by the vertical axis. Features which are short-lived (topological noise) disappear shortly after they appear, so are located near the dashed line $d = b$. Robust features persist for large ranges of the parameter, so are far above the diagonal. For the figure-8, we see a single red H_0 point above the dashed red line indicating a connected component that persists as $r \to \infty$, as well as two blue H_1 points well above the diagonal indicating there are two robust loops.

1.2.1 Construction of Homology

To make homology a little less mysterious, we'll give a first pass at explaining how homology of a simplicial complex $\mathcal{X}$ is computed. The first step is to form a *chain complex* $C_*(\mathcal{X}) = \{C_k(\mathcal{X}), \partial_k\}_{k=0}^\infty$. Every $C_k(\mathcal{X})$ is a vector space with a basis element for each k-simplex of $\mathcal{X}$ and the boundary map $\partial_k : C_k(\mathcal{X}) \to C_{k-1}(\mathcal{X})$ is a linear map that sends the vector associated with a k-simplex to a linear combination of vectors in its boundary (with ∂_0 defined to be 0). These boundary maps have the property that $\partial_k \circ \partial_{k+1} = 0$, which means that $\ker \partial_k \subseteq \operatorname{img} \partial_{k+1}$.

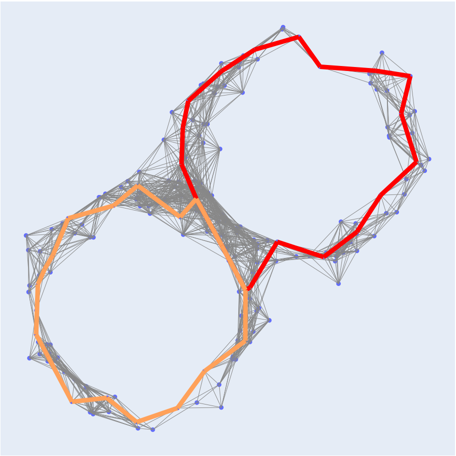

Figure 1.5: Generators for H_1 visualized as colored cycles in the figure-8.

Homology in dimension k is defined as the quotient vector space $\ker \partial_k / \operatorname{img} \partial_{k+1}$. Each non-zero vector in $H_k(\mathcal{X})$ is an equivalence class of vectors in $C_k(\mathcal{X})$, and we can choose a representative in $C_k(\mathcal{X})$ which generates the vector in H_k. Examples of these generators for H_1 of our figure-8 can be seen in figure 1.5.

There are two aspects to interpreting these representatives that require some care: first, the choice of basis for homology is not unique, and second, the choice of representative for a homology class is not unique either. Another detail that requires some attention is the choice of field when forming the chain complex $C_*(\mathcal{X})$. Because we want exact kernels and images, it is often best to avoid floating point arithmetic used for computation when the field is $\mathbb{R}$, and in practice finite fields or rational numbers are used. Depending on the choice of field, we may even end up with different dimensions in homology! In later lectures, we will cover homology and its variants in more detail.

1.3 A Topological Signature for Images

Now, let's turn to how we might produce a classifier for images of digits based on topological features, again using persistent homology. The digits in figure 1.2 are very different data than the point cloud in figure 1.1, but we can again use persistent homology by choosing a different filtration. We will think of images as functions on a square, which we triangulate to form a simplicial complex with vertex set in

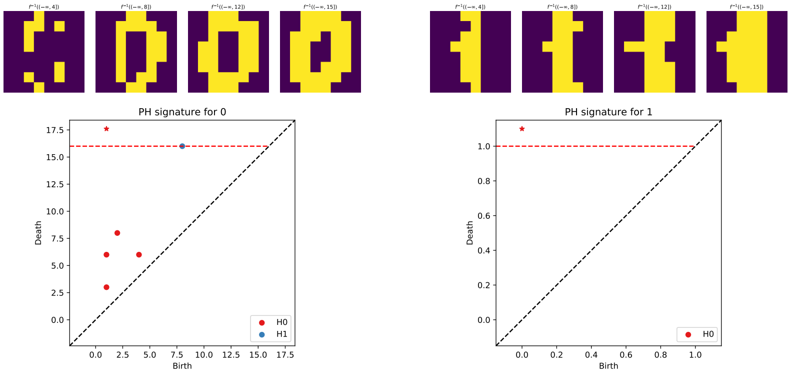

Figure 1.6: Sublevel set filtrations and persistence diagrams of the zero and one digits in figure 1.2.

correspondence with the pixels of the image. In a black and white image, pixel intensity is a real-valued function $f : \mathcal{X} \to \mathbb{R}$, and we can consider sublevel sets $f^{-1}((-\infty, a])$ of the image to form a filtration since $f^{-1}((-\infty, a]) \subseteq f^{-1}((-\infty, b])$ if $a \leq b$.

In figure 1.6, we see how the persistent homology of a "0" digit computed from a sublevel set filtration has a robust H_1 vector, whereas the "1" digit has no H_1 vectors. We also see that the "0" digit has several less robust H_0 features, likely due to varying pen pressure when the digit is drawn. How might we classify the digits based on these observations? Let's produce two features. The first will be the maximum length of a H_1 vector,

$$\max\{|d_i - b_i| \mid (b_i, d_i) \in H_1(\mathcal{X})\}$$

The second will be the sum of the lengths of *finite* H_0 vectors

$$\sum_{(b_i,d_i)\in H_0(\mathcal{X})} |d_i - b_i| \mathbb{I}_{|d_i - b_i| < \infty}$$

If we plot these two features computed over many examples of the digits zero and one, as in figure 1.7,

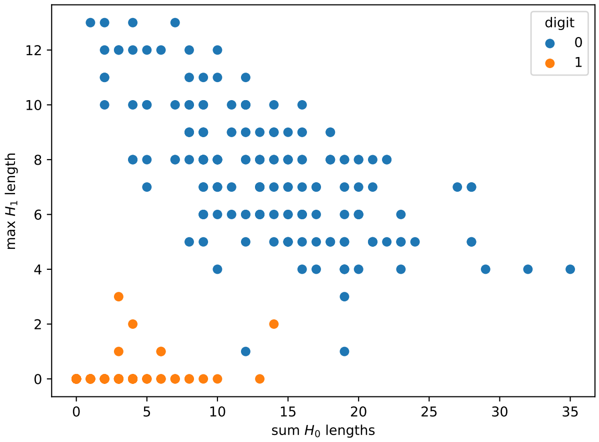

Figure 1.7: Two persistent-homology derived features computed for digital images of the digits "0" and "1".

we see a nice separation between the digits. Note that for some "1" digits, there are short H_1 vectors, and some "0" digits have a small sum of H_0 lengths, but between these two features we can visually classify the digits.

1.4 Further Questions

We now seen two different examples of problems that topological data analysis might try to address. We will cover these problems in more detail as well as others in this course. Here are some other questions that we may wish to consider as we proceed:

1. **Constructions** What different constructions of topological spaces can be used in addition to Vietoris-Rips complexes, and this triangulated grid for images? How might these relate to each other?

2. **Stability** How much does persistent homology of the Vietoris-Rips construction change if the input point cloud changes a little? What if the pixel values in an image change a little? Is the output stable with respect to perturbations of the input?

3. **Sampling** If points are sampled from some ground truth space/manifold, can the homology of the space be recovered from the sampling?

4. **Computation** How can topological spaces be represented on a computer? How is homology *really* computed? How can this be done efficiently?

5. **Generalizations and Alternatives** What about more than one filtration parameter? In what other ways might topological signatures be constructed from input data?

6. **Modeling** How can topological data analysis be applied to real problems? What are some examples and are there any principles for success?

1.5 A Brief History

Chapter 2

Graphs and Clustering

2.1 Graphs

A graph $G(V, E)$ is a collection of nodes $V = \{1, \ldots, N\}$ and edges $E \subset V \times V$. We will use the convention that the number of nodes in a graph is $N = |V|$ and the number of edges in the graph is $M = |E|$. We will generally consider undirected, unweighted graphs.

2.2 Path Components and Union-Find

2.3 Spectral Clustering

2.3.1 The Graph Laplacian

First, we recall the definition of the incidence matrix $B \in \mathbb{R}^{N\times M}$, which encodes the relationships between nodes and edges:

$$\begin{cases} B[i,k] & = -1 \qquad e_k = (i,j) \\ B[j,k] & = +1 \qquad e_k = (i,j) \\ B[\cdot,k] & = 0 \qquad \text{otherwise} \end{cases} \tag{2.1}$$

For an undirected graph we can choose the sign arbitrarily for the edge $e_k = (i,j)$, as long as $B[i,k] = -B[j,k]$.

The graph Laplacian $L \in \mathbb{R}^{N\times N}$ is defined as $L = BB^T$.

Exercise 2.3.1. *The graph Laplacian L can also be written $L = D - A$ where D is the diagonal degree matrix of the graph G, and A is the indicdence matrix of G.*

Proposition 2.3.1. *L satisfies the following properties:*

1. *$x^T L x = \sum_{(i,j)\in E}(x_j - x_i)^2$*
2. *L is symmetric, positive semi-definite*
3. *The null eigenspace of L is spanned by indicators on connected components.*

Proof. Item 1: $x^T L x = (B^T x)^T (B^T x)$. $B^T x$ is a M-dimensional vector whose k-th entry is $(B^T x)[k] = x_j - x_i$. The quadratic form $x^T L x$ is just the inner product of this vector with itself, so

$$x^T L x = \sum_{(i,j)\in E} (x_j - x_i)^2 \tag{2.2}$$

Item 2: symmetry is easy, since $L = BB^T$. Positive semi-definite means that $x^T L x \geq 0$ for any $x \in \mathbb{R}^N$. From item 1, we know that this is the sum of squares $\sum_{(i,j)\in E}(x_j - x_i)^2$. Every entry in the sum is non-negative, so the sum is non-negative.

Item 3: because L is symmetric its eigenvalues are real and there exists an orthogonal eigenbasis $\{(v_i, \lambda_i)\}_{i=1}^N$ for L, so $Lv_i = \lambda_i v_i$, and $v_i^T v_j = 0$ if $i \neq j$. Let $\mathbb{I}_C \in \mathbb{R}^N$ denote the indicator vector on a path-connected component $C \subseteq V$:

$$\mathbb{I}_C[i] = \begin{cases} 1 & i \in C \\ 0 & i \notin C \end{cases} \tag{2.3}$$

Note that $v_i^T L v_i = \lambda_i \|v_i\|_2^2$, and $v_j^T L v_i = 0$. As a result, if $x^T L x = 0$, then x is in the null eigenspace of L. We can verify that

$$\begin{aligned} \mathbb{I}_C^T L \mathbb{I}_C^T &= \sum_{(i,j)\in E} (\mathbb{I}_C[j] - \mathbb{I}_C[i])^2 \\ &= \sum_{(i,j)\in E} (0)^2 \\ &= 0 \end{aligned}$$

because any edge (i, j) connects two vertices in the same path component. Either $i, j \in C$, in which case $\mathbb{I}_C[j] - \mathbb{I}_C[i] = 1 - 1 = 0$, or thare are in a different path component and $\mathbb{I}_C[j] - \mathbb{I}_C[i] = 0 - 0 = 0$. We conclude that $\mathbb{I}_C$ is in the null eigenspace of L.

Now, suppose that x is a vector in the null eigenspace of L. Then $x^T L x = 0$. This means the sum $\sum_{(i,j)\in E} (x_j - x_i)^2 = 0$. Because the sum is zero, and all terms in the sum are non-negative, every term in the sum must be zero. This means $x[j] - x[i] = 0$ for all $(i, j) \in E$, which implies that $x[j] = x[i]$ for any two vertices in the same path component. As a result, any eigenvector in the nullspace is in the span of indicators of connected components.

Furthermore, since vertices belong to a unique path component, if $C, D \subseteq V$ are distinct path components then the vectors $\mathbb{I}_C$ and $\mathbb{I}_D$ are orthogonal. □

As s a result of proposition 2.3.1, we can identify path-connected components of a graph by computing the null eigenspace of the graph Laplacian.

2.3.2 Clustering within Path Components

We'll now restrict our attention to a graph G which has a single path component. In this case, the null eigenspace is spanned by the constant vector $\mathbb{I}$. We would generally like to be able to cluster within path components. What we would like to do is to partition the vertex set V into $S, \bar{S} \subset V$, where $S \cup \bar{S} = V$ and $S \cap \bar{S} = \emptyset$ as to minimize the number of edges that connect the two sets. Let

$$E(S, \bar{S}) = \{(i, j) \in E \mid i \in S, j \in \bar{S} \text{ or } i \in \bar{S}, j \in S\} = (S \times \bar{S} \cup \bar{S} \times S) \cap E$$

denote the set of edges that connect S and $\bar{S}$. Let $v_S = (\mathbb{I}_S - \mathbb{I}_{\bar{S}})/2$, and note that

$$\begin{aligned} v_S^T L v_S &= \sum_{(i,j)\in E} ((\mathbb{I}_S[j] - \mathbb{I}_S[i] + \mathbb{I}_{\bar{S}}[i] - \mathbb{I}_{\bar{S}}[j])/2)^2 \\ &= \sum_{(i,j)\in S\times\bar{S}\cap E} (1)^2 + \sum_{(i,j)\in \bar{S}\times S\cap E} (1)^2 \\ &= |E(S, \bar{S})| \end{aligned}$$

Note that if we're seeking to minimize $|E(S, \bar{S})|$ that we're trying to minimize this quadratic form subject to some constraints. One way to approach this is to look at the eigenvector v_1 associated with the smallest non-zero eigenvalue of the graph Laplacian, λ_1 and to partition the graph based on the sign of the entries in the vector v_i.

$$S = \{i \mid v_1[i] > 0\} \tag{2.4}$$

Note that there is a sign ambiguity in eigenvectors, but it doesn't matter. We recover the same partition $S, \bar{S}$ either way.

Normalization of eigenvectors.

The Cheeger inequality gives a notion of how well a cut based on the smallest non-zero eigenvalue approximates an optimal cut.

Bibliography

[1] Zixuan Cang and Guo-Wei Wei. TopologyNet: Topology based deep convolutional and multi-task neural networks for biomolecular property predictions. *PLOS Computational Biology*, 13(7):e1005690, July 2017.

[2] Gunnar Carlsson, Tigran Ishkhanov, Vin de Silva, and Afra Zomorodian. On the Local Behavior of Spaces of Natural Images. *International Journal of Computer Vision*, 76(1):1–12, January 2008.

[3] Frédéric Chazal, David Cohen-Steiner, Leonidas J. Guibas, Facundo Mémoli, and Steve Y. Oudot. Gromov-Hausdorff Stable Signatures for Shapes using Persistence. In *Computer Graphics Forum*, volume 28, pages 1393–1403. Wiley Online Library, 2009.

[4] Yasuaki Hiraoka, Takenobu Nakamura, Akihiko Hirata, Emerson G. Escolar, Kaname Matsue, and Yasumasa Nishiura. Hierarchical structures of amorphous solids characterized by persistent homology. *Proceedings of the National Academy of Sciences*, June 2016.