INTRODUCTION TO TOPOLOGICAL DATA ANALYSIS

ANGELA LIU

ABSTRACT. This paper is an introduction to topological data analysis (TDA) that is accessible for those who have little to no experience in topology. As TDA utilizes many techniques found in topology, the beginning sections will cover important topological definitions that are used in TDA such as metric spaces and simplicial complexes. Then, we will use the Python GUDHI library to analyze the New York taxi data right around when the COVID 19 caused lockdowns to demonstrate how to use an accessible TDA tool.

CONTENTS

1. Introduction to Topological Data Analysis 1
2. Topologies, Metric Spaces and Distances 2
3. Simplicial Complexes 4
4. The Nerve Theorem 5
5. The GUDHI Library 7
6. Conclusion 9
7. Acknowledgments 9
8. Bibliography 9
References 9

1. INTRODUCTION TO TOPOLOGICAL DATA ANALYSIS

As the amount and variety of data continues to grow, it often becomes too difficult to manage and extract meaningful information from data, especially in high dimensions. Thus, Topological Data Analysis (TDA), which provides a framework for analyzing data, has been a rapidly growing field. Due to its ability to extract information from high-dimensional data, TDA has become a popular way to analyze data in fields such as bioinformatics, finance, 3D shape analysis, and many more. The underlying principle of TDA is to use algebraic topology to give shape to the data, thus allowing it to be analyzed easier. Though TDA is a relatively young field that was only formally established in the early 21st century, there has been a large effort to provide easily accessible data structures and algorithms for TDA. An example of this includes the GUDHI library for Python and C++, which will later be used in the paper to analyze the New York Taxi data during March 2020, right when the COVID 19 pandemic began to cause people to enter lockdown.

As TDA continues to grow as a field, there have been numerous approaches to utilize it. In this paper, however, we will focus on the following pipeline that has formed the backbone of many standard approaches.
To begin, we must first understand what data means in the specified context. We assume the input is a finite set of data points with a defined notion of distance between them.

A continuous shape is created to showcase the topology or geometry of the data.

Topological information is extracted from the shape created, which reveals new features of the data.

2. TOPOLOGIES, METRIC SPACES AND DISTANCES

Before using TDA, we must first understand the mathematics that lay the groundwork for the field.

**Definition 2.1.** A topology on a set $X$ is a collection of subsets of $X$, denoted as $T$, that satisfy the following properties:

1. $\emptyset$ and $X$ are in $T$.
2. An arbitrary union of elements of a subcollection of $T$ is in $T$.
3. The intersection of elements of an arbitrary finite subcollection of $T$ is in $T$.

A set $X$ is a topological space if there has been a topology $T$ specified.

**Definition 2.2.** Consider a topological space $X$ with topology $T$. The subset $U \subset X$ is an open set of $X$ if $U \in T$.

**Definition 2.3.** Consider a topological space $X$. The subset $U \subset X$ is a closed set of $X$ if $X \setminus U$ is open.

With open and closed sets defined, we can define compactness. Compactness is a useful tool in topology as compact sets have properties similar to finite sets, thus making their behavior more predictable and easier to work with.

**Definition 2.4.** A cover of a space $X$ is a collection of subsets $A$ of $X$ such that the union of all sets in $A$ is equal to $X$. If all the subsets in $A$ are open subsets of $X$, then $A$ is an open cover of $X$.

**Example 2.5.** Consider the real numbers $\mathbb{R}$. An example of an open cover of $\mathbb{R}$ is the collection of open sets $(-n, n)_{n \in \mathbb{N}}$.

**Definition 2.6.** A space $X$ is compact if every open cover of $X$ also has a finite subcover of $X$.

Free-floating data points often do not reveal much on their own, thus making it necessary to link data points to discover underlying topological properties of the data that were previously unseen. This can be done with a defined notion of distance that quantifies the closeness of data points. For example, a Vietoris Rips complex (which will be discussed more in depth later) connects data points that are within a specified distance of each other.

**Definition 2.7.** A metric space $(M, d)$ is a set $M$ with distance function $d : M \times M \to \mathbb{R}_+$. For any $a, b, c \in M$, the following are true:

1. $d(a, b) = 0$ if and only if $a = b$. Otherwise, $d(a, b) > 0$.
2. $d(a, b) = d(b, a)$.
3. $d(a, c) \leq d(a, b) + d(b, c)$. 
Example 2.8. Consider a metric space $(M, d)$ and let $S$ be the set of its non-empty, compact subsets. Consider $a \in M$ and $X \in S$. Then, $d(a, X) = \inf\{d(a, b) | b \in X\}$. We can find what is called Hausdorff distance, which is calculated with the following equation. We will also prove that the Hausdorff distance is a metric on $S$.

We can define the Hausdorff distance between two compact sets $A, B \in S$ as

$$d_H(A, B) = \max\{\sup\{d(x, B) | x \in A\}, \sup\{d(y, A) | y \in B\}\}. \tag{2.9}$$

**Proof.** To prove that the Hausdorff distance is a metric on $S$, we will prove that it satisfies the three conditions of a metric that are laid out above.

1. First, we will prove that if $d_H(A, B) = 0$, then $A = B$. From definition 2.2, we know that $\sup\{d(x, B) | x \in A\} \geq 0$ and $\sup\{d(y, A) | y \in B\} \geq 0$. Since $d_H(A, B) = 0$, we know $\sup\{d(x, B) | x \in A\} = \sup\{d(y, A) | y \in B\} = 0$ so for all $a \in A$ and $b \in B$, $d(a, B) = d(b, A) = 0$. Thus, all elements of $A$ are in $B$ and all elements of $B$ are in $A$, so $A = B$.

Next, we will prove that if $A = B$, then $d_H(A, B) = 0$. Since $A = B$, we know $\sup\{d(x, B) | x \in A\} = \sup\{d(y, A) | y \in B\} = 0$. Thus, $d_H(A, B) = \max\{\sup\{d(x, B) | x \in A\}, \sup\{d(y, A) | y \in B\}\} = 0$.

Now, since $d_H(A, B) = 0$ if and only if $A = B$, it follows that $d_H(A, B) > 0$ if $A \neq B$.

2. We will now prove that $d_H(A, B) = d_H(B, A)$. We know

$$d_H(A, B) = \max\{\sup\{d(w, B) | w \in A\}, \sup\{d(x, A) | x \in B\}\}$$

and

$$d_H(B, A) = \max\{\sup\{d(y, A) | y \in B\}, \sup\{d(z, B) | z \in A\}\}.$$ 

Since $A$ and $B$ have not changed, $\sup\{d(x, A) | x \in B\} = \sup\{d(y, A) | y \in B\}$ and $\sup\{d(w, B) | w \in A\} = \sup\{d(z, B) | z \in A\}$. Thus,

$$\max\{\sup\{d(w, B) | w \in A\}, \sup\{d(x, A) | x \in B\}\}$$

$$= \max\{\sup\{d(y, A) | y \in B\}, \sup\{d(z, B) | z \in A\}\}$$

so $d_H(A, B) = d_H(B, A)$.

3. Finally, we will show for compact sets $A, B, C \in S$, $d_H(A, C) \leq d_H(A, B) + d_H(B, C)$. From definition 2.2, we know $d(a, C) \leq d(a, C) \leq d(a, b) + d(b, c)$ for all $c \in C$. Thus, $d(a, C) \leq d(a, b) + d(b, C) \leq d(a, b) + d_H(B, C)$ for all $b \in B$. Therefore, $d(a, C) \leq d(a, b) + d_H(B, C) \leq d_H(A, B) + d_H(B, C)$ for all $a \in A$. Finally, we know $d_H(A, C) \leq d_H(A, B) + d_H(B, C)$.

Thus, we have proven that the Hausdorff distance is a metric for the above metric space, providing us with a method to quantify how close two data sets are on the same metric space. This is just one way to complete the first step of the TDA pipeline. A further investigation into metric spaces will show that there are other ways to measure this distance. One of these is the Gromov-Hausdorff Distance, which can be used to quantify the closeness of two compact metric spaces that are not from the same ambient space.
3. Simplicial Complexes

We will now discuss simplicial complexes, which are vital for understanding how to connect our data.

**Definition 3.1.** A $k$-simplex is a set of $k + 1$ vertices denoted as $[p_0, p_1, ..., p_k]$.

**Definition 3.2.** The faces of a $k$-simplex are the simplicies that are proper subsets of the $k$-simplex.

For example, the faces of a pentahedron, also known as a square pyramid, would include squares, triangles, edges, vertices, and the empty set.

**Definition 3.3.** A geometric simplicial complex $K$ in $\mathbb{R}^d$ is a collection of simplicies that satisfy the following conditions:

1. Each face of a simplex of $K$ is also a simplex of $K$.
2. The intersection of two arbitrary simplicies of $K$ is either empty or a face of both simplicies.

Aside from a geometric simplicial complex, there is another way to construct a complex without having to put it into a Euclidean space.

**Definition 3.4.** For a set $V$, an abstract simplicial complex $\tilde{K}$ with the vertex set $V$ is a set of finite subsets of $V$ that satisfy the following conditions:

1. Each of the simplicies in $V$ is also in $\tilde{K}$.
2. Any subset of an element of $\tilde{K}$ is also in $\tilde{K}$.

The elements of $\tilde{K}$ can be referred to as either faces or simplicies of $\tilde{K}$.

$K$ is called a geometric realization of $\tilde{K}$ if the combinatorial description of $K$ is the same as $\tilde{K}$. Thus, this relation allows geometric simplicial complexes to be viewed with underlying combinatorial structure and for abstract simplicial complexes to be treated as topological spaces. This flexibility makes simplicial complexes a useful tool to analyze data since they are suited for computations as abstract simplicial complexes and can be seen as topological spaces as geometric simplicial complexes.

Now that we know what simplicial complexes are and why they are used in TDA, we can look into some examples of common complexes that are built from data. A couple of these include the Vietoris-Rips complex and the Čech complex.

Consider a metric space $(M, p)$, a real number $\epsilon \geq 0$, and a set of points $X$ in the metric space. The Vietoris-Rips complex, denoted as $V_\epsilon(X)$, is given by:

\[
V_\epsilon(X) = \{ \sigma \in X | d(x, y) \leq \epsilon \forall x \neq y \in \sigma \}.
\]

In other words, the complex is created by connecting data points within the set that are within $\epsilon$ distance of each other, forming various simplicies.

Figure 1, shown on the next page, shows two Vietoris-Rips complexes of the same data set. Consider an $\epsilon$. The figure on the left represents $V_\epsilon(X)$ while the one on the right represents $V_{2\epsilon}(X)$.

From the two diagrams, we notice that the simplicies of $V_\epsilon(X)$ include triangles, edges, and vertices. Moreover, we note that $V_{2\epsilon}(X)$ include the simplicies of $V_\epsilon(X)$ and two more edges. Thus, we know that $V_\epsilon(X)$ is a subset of $V_{2\epsilon}(X)$. This makes sense intuitively, because any points within $\epsilon$ of each other must also be within $2\epsilon$ of each other.
Figure 1. On the left, we have the complex $V_{\epsilon}(X)$. On the right, we have the complex $V_{2\epsilon}(X)$.

Another common complex is the Čech complex, which is closely related to the Vietoris-Rips complex. Consider a metric space $(M, p)$, a real number $\epsilon \geq 0$, and a set of points $X$ in the metric space. The Čech complex, denoted as $\check{C}ech_{\epsilon}(M)$, is defined as the set of simplices $[x_0, x_1, \ldots, x_l]$ where all of the $k+1$ balls with radius $\epsilon$ have a non-empty intersection. Figure two illustrates the difference between the the Čech and Vietoris Rips complexes.

Figure 2. On the left, we have the Čech complex $\check{C}ech_{2\epsilon}(M)$ generated by the five points, and the Vietoris Rips complex $V_{2\epsilon}(X)$ on the right. Since only 4 of the 5 points have a non-empty intersection, the Čech complex consists of a tetrahedron and a line. On the other hand, since the fifth point is still within $2\epsilon$ of one of the other four points, so the simplex generated is a 5-cell.

4. The Nerve Theorem

One of the most important theorems of TDA is the Nerve Theorem. It provides a way to convert continuous topological spaces into combinatorial structures that are more suitable for computation. To understand the Nerve Theorem, there are a few topological terms we will need to define first.

**Definition 4.1.** The nerve $C(\mathcal{U})$ of a cover $\mathcal{A} = (A_i)_{i \in I}$ of a space $X$ is an abstract simplicial complex that satisfies the following conditions:

1. the vertices of the complex are the $A_i$’s, and
2. $A_{i_0}, \ldots, A_{i_n} \in C(\mathcal{U})$ if and only if $\bigcap_{j=0}^{k} A_{i_j} \neq \emptyset$.
Figure 3. The left represents a point cloud and a cover. The right shows the nerve. Each vertex represents the non-empty intersection of two sets and each edge corresponds to a set.

Definition 4.2. Let $f$ and $g$ be continuous maps of $X$ onto $Y$. Then, $f$ and $g$ are \textbf{homotopic} if there is a continuous map $h : X \times [0,1] \to Y$ such that for any $x \in X$,

1. $h(x,0) = f(x)$ and
2. $h(x,1) = g(x)$

In this case, the map $h$ is a \textbf{homotopy} between $f$ and $g$.

Definition 4.3. The spaces $X$ and $Y$ are \textbf{homotopy equivalent} if there exist maps $f : X \to Y$ and $g : Y \to X$ such that $f \circ g$ is homotopic to the identity map of $Y$ and $g \circ f$ is homotopic to the identity map of $X$.

There are two commonly used methods to determine whether homotopy equivalences can be produced from different-looking spaces. The first method involves contracting subspaces to points, while the second relies on changing the ways in which different parts of a space are put together. For example, an infinity symbol is homotopy equivalent to a line segment with two circles at the end since the line segment can be collapsed into a point, thus creating the infinity symbol.

Definition 4.4. A space is \textbf{contractible} if it is homotopy equivalent to a point.

With these definitions, we will introduce and provide a basic proof of the Nerve theorem.

Theorem 4.5. Let $X$ be a topological space and let $\mathcal{U} = (U_i)_{i \in I}$ be a cover of open sets of $X$ such that the intersection of any subcollection of $\mathcal{U}$ is either empty or contractible. Then, $X$ and the nerve $C(\mathcal{U})$ are homotopy equivalent.

Proof. First, we will prove that convex sets in euclidean spaces are contractible. Let $A$ be a convex set of topological space $X$. Let $f : A \times [0,1] \to A$ be defined such that for $t \in [0,1]$ and all $a \in A$,

$$H(b, t) = (1 - t)b + ta.$$ 

Thus, we know $H(b, 0) = b$ and $H(b, 1) = a$ for all $b \in A$. This means the identity map on $A$ is homotopic to the constant map at $a$. It follows that $A$ and $a$ are homotopy equivalent so $A$ is contractible.

Consequently, we know that if $\mathcal{U}$ is a collection of convex sets, then $C(\mathcal{U})$ and $\bigcup_{i \in I} U_i$ are homotopy equivalent. \hfill $\square$
5. The GUDHI Library

As mentioned previously, we will now use the GUDHI Python library to create a Vietoris Rips complex on a data set. For my example, I used the taxi data for March 2020, which is when the COVID 19 pandemic started. Since the dataset was very large, I first queried the data to only include taxi zones that had less than a 75% drop in pickups after March 20th (about when lockdown started). By the end, I had 64 taxi zones. From here, I found the centroid of each zone using the geopandas library and created a new dataset that only had the 64 taxi zones. This data set also contained information on the shape, geometry, area, borough, and other features of the 64 taxi zones.

Next, I decided to plot the centers of the taxi zones to visualize how far apart they were. This way, I could see whether they were all clustered together or separately. The x variable represents the longitude and the y variable represents the latitude.

```python
df = pd.read_csv('/content/drive/My Drive/NYC/Centers.csv')

centers_data = df[['x', 'y']].copy()
centers_data.head()

centers_data['centers'] = centers_data.values.tolist()

ax1 = centers_data.plot.scatter(x='x', y='y', c="DarkBlue")
```

**Figure 4.** First I loaded my dataframe and made a new one with just the center’s x and y coordinates. Then I created a column of the dataframe that makes a list of the x and y values. This column is later used as my data points for the Vietoris Rips complex.

![Figure 4](image)

**Figure 5.** Here is the 2D representation of the 64 taxi zones that had less than a 75% decrease in pickups in March 2020 after lockdown started.

Now, we can see that a lot of the points are clustered towards the top of the graph while the rest are more spread out on the bottom of the graph. We will now use the GUDHI library to create a Vietoris Rips complex over this data to find the simplices this data creates.

To begin, we will create a skeleton using the `RipsComplex()` function.

Due to computational reasons, it is common to define a maximal dimension of simplicities for the Rips Complex. For our example, we will set this value to equal 3.
skeleton = gd.RipsComplex(points = centers_data['centers'], max_edge_length = 0.02)

**Figure 6.** Using the Centers column of the data set as the points, I set my max edge length to 0.02. This means that only points that are at most 0.02 away from each other are included in the skeleton.

Rips_simplex_tree = skeleton.create_simplex_tree(max_dimension = 3)

**Figure 7.** With the create_simplex_tree function, I can set the maximum dimension to be 3 with the max_dimension parameter.

Now, our Vietoris Rips complex has been created. We can view information about the complex with some simple commands.

Rips_simplex_tree.num_vertices()

Rips_simplex_tree.num_simplices()

**Figure 8.** With these two commands, we can find out that all 64 data points are vertices in the complex, and there are 181 simplicies in the 3D simplicial complex.

Finally, we can create a list of all the simplicies in the complex and print them out. The Vietoris Rips complex of the data set has 181 simplicies in 3 dimensions. Since that is a lot of simplicies, I have only included a few below to demonstrate what the outcome of using the GUDHI library to generate a Vietoris Rips Complex may look like.

rips = Rips_simplex_tree.get_filtration()

rlist = list(rips)
for simplex in rlist :
    print(simplex)

**Figure 9.** Finally, we get the filtrations and print out the list.

Here are some results:

((0], 0.0)
((1], 0.0)
((2], 0.0)
((12, 23], 0.11430584518535474)
((12, 13, 23], 0.11430584518535474)
((16, 35, 38, 39], 0.019904346485146263)

Each pair has a list of values and a filtration value, also known as the diameter of the simplex. From this selection of outputs I have chosen from the list of outcomes, we can see that the Vietoris Rips Complex has individual points as simplicies with filtration value 0, as well as lines, triangles, and even tetrahedrons since the max dimension can be 3.
6. Conclusion

Topological Data Analysis is a field with limitless possibilities. Although it is still a young field, there have been many efforts to make it accessible, thus allowing us to use this powerful tool to analyze the rapidly growing amounts of data. As this paper is meant to be an introduction to TDA, there are other powerful tools including the mapper algorithm and persistence homology that can further improve the tools used for data analysis.

The data set used in my example was provided by the NYC Taxi and Limousine Commission.

7. Acknowledgments

It is my pleasure to thank my mentor, Micah Gay, for her endless amount of support and guidance throughout the process of writing this paper. I would also like to thank Professor Danil Rudenko for organizing an amazing Apprentice Program that I was able to learn a lot from. Finally, Professor Peter May for running the REU and giving me the chance to write this paper.

8. Bibliography

References