# Artworks and Articles Meet Mapper and Persistent Homology

Hongyuan Zhang, Alicia Ledesma Alonso, Cherie Li, Chris Won*

07/31/18

**Abstract**

Since its recent birth, topological data analysis has proven to be a very useful tool when studying large and complex datasets. In this paper, we explore persistent homology tools and Mapper to explore two main datasets from the following sources: Metropolitan Museum of Art (MET) and arXiv. After learning the preliminary theory of topology, the two datasets were studied independently. For the MET project, we use Mapper to guide feature selection. We then use persistent homology to help differentiate between two subsets of artworks. For the arXiv project, we use persistent homology to derive a general sense of the shape of the data. We next use Mapper to further explore this idea by analyzing trends and certain features in the Mapper visualizations. By using these tools, detailed insights are given in understanding the complexity of each dataset.

## 1 Introduction

In the past decade, topological data analysis (TDA) has been a tool of key interest to researchers who want to analyze and visualize large and complex datasets. Specifically, TDA allows us to explore the "shape" of our data, through which we can filter out noise and reveal its real structure. There are two primary tools of TDA: persistent homology and the Mapper algorithm. The concept of persistent homology was first introduced by Edelsbrunner[12], followed by Carlsson and Zomorodian[44]. More recently, Singh, Mémoli and Carlsson introduced the Mapper algorithm in 2007 [39].

TDA has three key facets: coordinate freeness, deformation invariance, and compressed representations[29]. By *coordinate freeness*, we mean that TDA allows us to compare data from different coordinate systems. Rather than having the embedded coordinate system, we are more interested in the distance between data points (see Section 2.1). By *deformation invariance*, we mean that in topology, we study invariant shapes under non-extreme deformation (no cutting or gluing). A classic example of this invariance is that topologists treat a coffee cup and

*Research supported by NSF award number HRD-1619654. Any opinions, findings, and conclusions or recommendation expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

a donut as the same thing because you can transform one to another without cutting or gluing. The key topological feature that they share is a hole (the handle of the coffee cup and the hole in the center of the donut), and it is preserved. This idea allows differentiation between real structure and noise. By *compressed representations*, we mean reduced, finite representations of infinite shapes or a large amount of points. Ideally, compressed representations retain topological invariants of the original representations.

The Mapper algorithm[39] and persistent homology[12][44] have been applied to data analysis in several fields, including biology[32][42][28][37][33][10][26][38][5], network analysis[36][8][41], manufacturing[17], and natural language processing[43]. In this paper we apply these tools to a dataset describing artworks in the Metropolitan Museum of Art[1] and a dataset derived from research papers in the open preprint archive found at arXiv.org[3].

# 2 Preliminaries

## 2.1 Basic Topology

The Mapper algorithm and persistent homology's foundation in topological theory allows us to interpret our data with freedom. For example, our data does not have to come from the Euclidean space. Instead, we have a broader assumption that our data comes from a metric space, a type of topological space. We rely on the following definitions from Munkres[31], Kurlin[25] and Hatcher[18].

**Definition 1** (Topology)**.** A topology on a set $X$ is a collection $\mathscr{T}$ of subsets of $X$ having the following properties:

(1) $\emptyset$ and $X$ are in $\mathscr{T}$.

(2) The union of the elements of any subcollection of $\mathscr{T}$ is in $\mathscr{T}$.

(3) The intersection of the elements of any finite subcollection of $\mathscr{T}$ is in $\mathscr{T}$.

A set $X$ for which a topology $\mathscr{T}$ has been specified is called a topological space.

If $X$ is a **topological space** with topology $\mathscr{T}$, we say that a subset $U$ of $X$ is an open set of $X$ if $U$ belongs to the collection $\mathscr{T}$.

In the Mapper algorithm, data points are often projected onto the real number line $\mathbb{R}$ with the standard topology.

**Definition 2** (Basis)**.** If $X$ is a set, a basis for a topology on $X$ is a collection $\mathscr{B}$ of subsets of $X$ (called basis elements) such that

(1) For each $x \in X$, there is at least one basis element $B$ containing $x$.

(2) If $x$ belongs to the intersection of two basis elements $B_1$ and $B_2$, then there is a basis element $B_3$ containing $x$ such that $B_3 \subset B_1 \cap B_2$.

**Definition 3** ($\mathscr{T}$ generated by $\mathscr{B}$)**.** If $\mathscr{B}$ satisfies these two conditions, then we define the topology $\mathscr{T}$ generated by $\mathscr{B}$ as follows: A subset $U$ if $X$ is said to be open in $X$ if for each $x \in U$, there is a basis element $B \in \mathscr{B}$ such that $x \in B$ and $B \subset U$. Note that each basis element is itself an element of $\mathscr{T}$.

**Definition 4** (Standard Topology on $\mathbb{R}$)**.** If $\mathscr{B}$ is the collection of all open intervals in the real line, $(a,b) = \{x | a < x < b\}$, the topology generated by $\mathscr{B}$ is called the standard topology on the real line. Whenever we consider $\mathbb{R}$, we shall suppose it is given this topology unless we specifically state otherwise.

Furthermore, we can project the data onto $\mathbb{R}^2$. $\mathbb{R}^2$ is the equivalent to $\mathbb{R} \times \mathbb{R}$. Assuming we have standard topology on each $\mathbb{R}$, the topology on $\mathbb{R}^2$ is the product topology.

**Definition 5** (Product Topology)**.** Let $X$ and $Y$ be topological spaces. The product topology on $X \times Y$ is the topology having as basis the collection $\mathscr{B}$ of all sets of the form $U \times V$, where $U$ is an open subset of $X$ and $V$ is an open subset of $Y$.

Our applications of tools from TDA rely on a pairwise distance function that describes distances between data points, or, a "metric".

**Definition 6** (Metric)**.** A metric on a set $X$ is a function $d : X \times X \to R$ having the following properties:

(1) $d(x,y) \geq 0$ for all $x, y \in X$; equality holds if and only if $x = y$.

(2) $d(x,y) = d(y,x)$ for all $x, y \in X$.

(3) (Triangle inequality) $d(x,y) + d(y,z) \geq d(x,z)$, for all $x, y, z \in X$.

Given a metric $d$ on $X$, the number $d(x,y)$ is often called the distance between $x$ and $y$ in the metric $d$.

A set with a metric is a topological space, automatically, with a basis made up of open balls, or "$\epsilon$-balls" [31].

**Definition 7** ($\epsilon$-ball)**.** Given $\epsilon > 0$, consider the set $B_d(x,\epsilon) = \{y | d(x,y) < \epsilon\}$ of all points $y$ whose distance from $x$ is less than $\epsilon$. It is called the $\epsilon$-ball centered at $x$.

**Definition 8** (Metric Topology)**.** If $d$ is a metric on the set $X$, then the collection of all $\epsilon$-balls $B_d(x,\epsilon)$, for $x \in X$ and $\epsilon > 0$, is a basis for a topology on $X$, called the metric topology induced by $d$.

**Definition 9** (Metric Space)**.** If $X$ is a topological space, $X$ is said to be metrizable if there exists a metric $d$ on the set $X$ that induces the topology of $X$. A metric space is a metrizable space $X$ together with a specific metric $d$ that gives the topology of $X$.

The Mapper algorithm and computation of persistent homology begins with data points from a point cloud.

**Definition 10** (Point Cloud)**.** A point cloud is a finite metric space.

Finally, we introduce one kind topological equivalence. The concept of equivalence is crucial in TDA, since our goal is to reveal the original data's structure via simplicial complexes, a compressed representation of the data. Using this notion of equivalence, we could say a coffee cup and a donut are equivalent. Certain simplicial complexes are considered equivalent to the original data by the following definition.

**Definition 11.** A map $f : X \to Y$ is called a homotopy equivalence if there is a continuous map $g : Y \to X$ such that $f \circ g$ and $g \circ f$ both give the identity map. The spaces X and Y are said to be homotopy equivalent or to have the same homotopy type.

## 2.2 Simplicial Complexes and Homology

Simplicial complexes can be used as compressed representations of the structures of point clouds. The building blocks of simplicial complexes are simplices. We rely on the following definitions from Edelsbrunner [14].

**Definition 12** (k-simplex)**.** Let $u_1, u_2, \cdots, u_{k+1}$ be $k+1$ linearly independent points in $\mathbb{R}^n$. Then the set $\{\sum_{i=0}^{k} \lambda_i u_i | \sum_{i=0}^{k} \lambda_i = 1, \lambda_i \geq 0\}$ is a k-simplex.

We call 0-simplex a vertex, 1-simplex, edge, 2-simplex , triangle, and 3-simplex, tetrahedron (Figure 1). We can connect simplexes to form simplicial complexes along their faces.

Figure 1: Examples of 0, 1, 2, 3 simplex[43]

**Definition 13** (face)**.** A face $\tau$ of a simplex $\sigma$ is a simplex spanned by a non-empty subset of $\{u_i\}$. We write $\tau \leq \sigma$ if $\tau$ is a face.

**Definition 14** (simplicial complex)**.** A simplicial complex is a finite collection of simplices $K$ such that $\sigma \in K$ and $\tau \leq \sigma$ implies $\tau \in K$, and $\sigma, \sigma_0 \in K$ implies $\sigma \cap \sigma_0$ is either empty or a face of both.

A simplicial complex can also be defined in a broader sense. That is, non-geometrically.

**Definition 15** (abstract simplicial complex)**.** An abstract simplicial complex is a finite collection of sets $A$ such that $\alpha \in A$ and $\beta \subset \alpha$ implies $\beta \in A$.

We can build a geometric simplicial complex $K$ from an abstract simplicial complex $A$ and vice versa. We call $K$ a geometric realization of $A$, and $A$ a vertex scheme of $K$.

We give a special name to a sum of $n$-simplicies.

**Definition 16** (p-chain)**.** Let $K$ be a simplicial complex and $p$ a dimension. A $p$-chain is a formal sum of $p$-simplices in $K$, denoted as $c = \sum a_i \sigma_i$ where the $\sigma_i$ are the $p$-simplices and the $a_i$ are the coefficients.

In computational topology, we mostly work with coefficients $a_i$ that are either 0 or 1, called modulo 2 coefficients.

**Definition 17** (group of p-chains)**.** The $p$-chains together with the addition operation form the group of $p$-chains denoted as $C_p = C_p(K)$.

For each dimension $p$, there is a $p$-chain group. To relate these $p$-chain groups, we have the boundary map.

**Definition 18** (boundary map)**.** The boundary of a $p$-simplex is the sum of its $(p-1)$-dimensional faces. The boundary of a $p$-chain is the sum of the boundaries of its simplices. Hence, taking the boundary maps a $p$-chain to a $(p-1)$-chain, and we write $\delta_p : C_p \to C_{p-1}$.$\delta_p$ is a homomorphism.

Thus, for a simplicial complex, its chain groups can be connected by boundary maps: $\cdots \xrightarrow{\delta_{p+2}} C_{p+1} \xrightarrow{\delta_{p+1}} C_p \xrightarrow{\delta_p} C_{p-1} \xrightarrow{\delta_{p-1}} \cdots$. We are interested in examining the topological features of a simplicial complex. In order to identify topological features, we focus on a subgroup of $p$-chains.

**Definition 19** (p-cycle)**.** A $p$-cycle is a $p$-chain with empty boundary, $\delta c = 0$. The group of $p$-cycles is denoted as $Z_p = Z_p(K)$, which is a subgroup of $C_p$.

Notice that the group of $p$-cycles is the kernel of the $p$-th boundary map, $Z_p = ker\delta_p$. To identify each different feature, we focus on the $p$-cycles that do not represent the same feature.

**Definition 20** (p-boundary)**.** A $p$-boundary is a $p$-chain that is the boundary of a $(p+1)$-chain, $c = \delta d$ with $d \in C_{p+1}$. The group of $p$-boundaries is denoted by $B_p = B_p(K)$, which is again a subgroup of $C_p$.

Notice that the group of $p$-boundaries is the image of the $(p+1)$-st boundary map, $B_p = im\delta_{p+1}$.We formalize the idea of "a boundary of a boundary is empty" in the following theorem.

**Theorem 1** (Fundamental Lemma of Homology)**.** $\delta_p \delta_{p+1} d = 0$ for every integer $p$ and every $(p+1)$-chain $d$.

Thus every $p$-boundary is also a $p$-cycle and $B_p \leq Z_p \leq C_p$. We now define the homology group and identify different topological features from it.

**Definition 21** (homology group and betti number)**.** The $p$-th homology group,$H_p$, is the quotient group $Z_p/B_p$. The $p$-th Betti number is the rank of this group, $\beta_p = rank(H_p)$.

Elements of $H_p$ are called homology classes. Two cycles representing the same homology class are said to be homologous. This means their difference is a boundary.[18] If the different between two cycles is a boundary, they contain the same topological feature. For example, in Figure 2, c2 and c3 contain the same hole but their difference is c1, boundary of the yellow triangle.

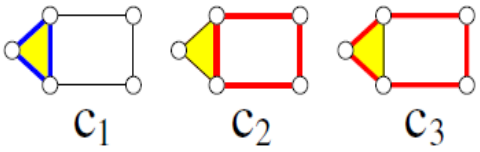



Figure 2: Example of homologous cycles (c2 and c3) [43]

$\beta_p$ is the number of topological features in the $p$th dimension. Intuitively, $\beta_0$ indicates the number of connected components, $\beta_1$ indicates the number of loops, $\beta_2$ the number of voids, and $\beta_n$ for $n$-dimensional analogues[43].

# 3 Topological Data Analysis

With the basic topological concepts in hand, we now introduce the pipelines of persistent homology and the Mapper algorithm.

## 3.1 Persistent Homology

In persistent homology, we compute betti numbers for a sequence of evolving simplicial complexes built from a point cloud, where the data points serve as vertices. How do we build a simplicial complex from a point cloud? There are multiple ways to do so; for the purpose of this paper, we introduce two such ways.

**Definition 22** (Vietoris-Rips (Rips) complex)**.** Given a collection of points $\{x_\alpha\}$ in Euclidean space$\mathbb{E}^n$, the Rips complex, $\mathscr{R}_\epsilon$ is the abstract simplicial complex whose $k$-simplices correspond to unordered $(k+1)$-tuples of points $\{x_\alpha\}_0^k$ which are pairwise within distance $\epsilon$.[16]

**Definition 23** (Čech complex)**.** Given a collection of points $\{x_\alpha\}$ in Euclidean space $\mathbb{E}^n$, the Čech complex, $\mathscr{C}_\epsilon$, is the abstract simplicial complex whose $k$-simplices are determined by unordered $(k+1)$-tuples of points $\{x_\alpha\}_0^k$ whose closed $\epsilon/2$-ball neighborhoods have a point of common intersection.[16]

The following theorem guarantees that we can recover the topology of the input point cloud from the Čech complex, since the Čech complex is a nerve.

**Definition 24** (Nerve)**.** For finite collection of sets, F, the nerve consist of all non-empty subcollections whose sets have a non-empty common intersection, Nr$F = \{X \subseteq F | \cap X \neq \emptyset\}$.[14]

**Theorem 2** (The Nerve Theorem)**.** Let F be a finite collection of closed, convex sets in Euclidean space. Then the nerve of F and the union of the sets in F have the same homotopy type.[14]

According to the Nerve Theorem, $\mathscr{C}_\epsilon$ is homotopically equivalent to the union of $\epsilon/2$-balls about the point set $\{x_\alpha\}$, whereas $\mathscr{R}_\epsilon$ is not. However, the Rips complex is computationally less expensive than the corresponding Čech complex, and in the context of persistent homology, the Rips complex is an acceptable approximation to Čech complex[16]. Thus, when applying persistent homology to our datasets, we build Rips complexes.

After we choose an $\epsilon$ and build a simplicial complex from the point cloud, we can compute betti numbers for this complex. However, the betti numbers only represent the structure of the underlying point cloud at a particular $\epsilon$ value; under which $\epsilon$ is the resulting complex "best" to represent the structure? In fact, in persistent homology, we compute betti numbers for a sequence of evolving simplicial complexes with increasing $\epsilon$ to see the "evolution" better. This sequence is called a filtration.

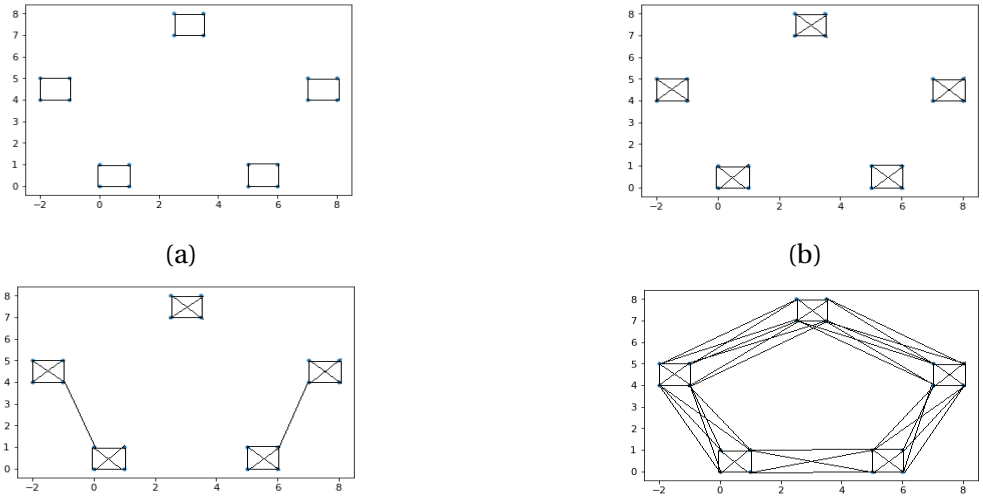
**Definition 25** (filtration)**.** An increasing sequence of $\epsilon$ produces a filtration, i.e., a sequence of increasing simplicial complexes $VR(\epsilon_1) \subset VR(\epsilon_2) \cdots$, with the property that a simplex enters the sequence no earlier than all its faces.[43]

In persistent homology, we track the birth and death (becoming filled-in) of homology generators along the filtration. A generator of the 0th homology group represents a connected component, that of the 1st homology group represents a loop, etc. We are interested in the "persistent" generators, by which we mean the ones that persist through a long range of $\epsilon$ values. Persistent ones are likely to be the real structure of data and short-lived ones are likely to be noise in data. We can use barcode graphs[16] to display this information. Persistence

diagrams[13] can also help visualize this same information. In our paper, we mainly use barcode graphs; see the following example for a sample barcode graph and its interpretation. In a barcode graph, $x$-axis is $\epsilon$, each bar represents a homology generator and the interval of the bar indicates the range of $\epsilon$ in which it exists. Generally, in a $p$-dimension barcode graph, the number of bars at a particular $\epsilon$ corresponds to $\beta_p$ at this $\epsilon$, since betti numbers are defined as ranks of homology groups and the barcodes track homology generators.

### 3.1.1 Examples

We use javaPlex 4.3.4[40] and Dionysus 2 [30] to generate the barcode graphs in this paper. Consider the example of five unit squares in $\mathbb{R}^2$ (Figure 3, Figure 4). We illustrate the approximate filtration of Rips-complexes through increasing epsilon. We start out with 20 separate bars in the 0th dimension when $\epsilon = 0$ since there are 20 vertices, each considered as a separate connected component. There are no loops yet, so there there are no bars in Dimension 1 at this point. At $\epsilon = 1$ (Figure 3a), five squares form, so there are 5 connected components, and 15 bars die in Dimension 0. In the first dimension, 5 bars appear, corresponding to loops formed by each square. At $\epsilon = \sqrt{2}$ (Figure 3b), these five bars end since the five squares are filled in when their diagonals are connected. Since there are still five connected components, there is no change in Dimension 0. As $\epsilon$ increases, edges form between squares, the number of connected components decreases (Figure 3c), and at some point, the five squares form a loop (Figure 3d). Thus in Dimension 1, one bar appears. As $\epsilon$ further increases, this loop also gets filled in and the bar in Dimension 1 dies. Thus, finally there is no bar in Dimension 1 and one bar in Dimension 0.



(a) (b) (c) (d)

Figure 3: Filtration example

As a second example, we generate Rips-complex barcode graphs for 50 randomly sampled points from a sphere of radius 1. We use javaPlex[40] to compute the barcodes. See Figure 5 for the visualization of these points in $\mathbb{R}^3$ and the barcode graphs for the first three dimensions.

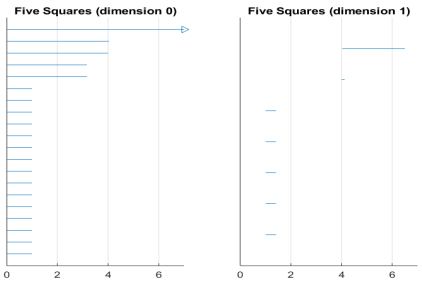
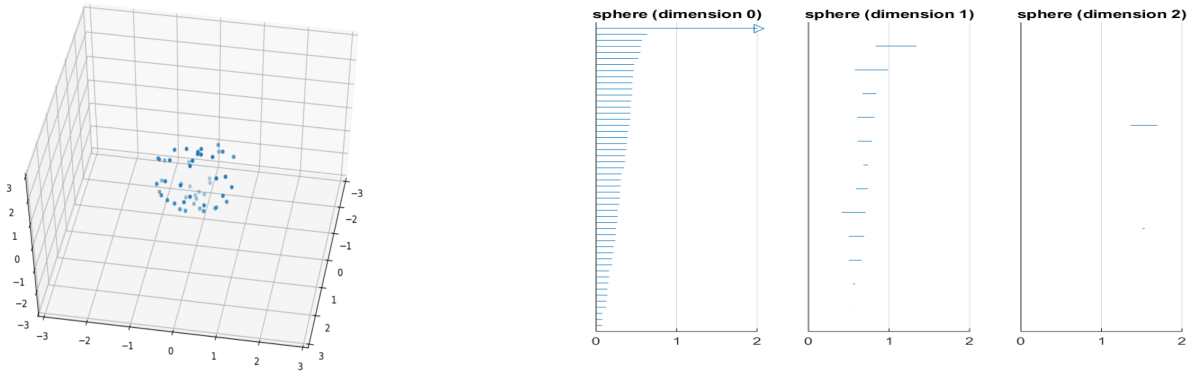



Figure 4: Barcode graphs for 5 squares




(a) Visualization of the 50 points

(b) Barcode graphs

Figure 5: Sphere example for persistent homology

For a sphere, $\beta_0 = 1$, $\beta_1 = 0$, $\beta_2 = 1$ and for higher dimensions, the betti number is 0, since there is one connected component, no loop, and one void in a sphere. Thus we expect this sampling from the sphere to exhibit some similar features. Indeed, checking Figure 1(b) shows some evidence that the point cloud where the 50 points are sampled from may be a sphere. We set the maximum $\epsilon$ to 2 since this is the maximum possible distance between any pair of points. Thus at $\epsilon = 2$, it is guaranteed that every point is connected to the rest. In Dimension 0, there is no similarly persistent bar as the top bar; this indicates that there is one persistent connected component. In Dimension 1, there are no persistent bars, and thus no persistent loops. In Dimension 2, there is one relatively persistent bar that represents a void born around $\epsilon = 1.4$ and filled in around $\epsilon = 1.7$. Remember the persistent bars are likely to be the real structure of the original point cloud. In this way, barcode graphs encapsulate the homology of a filtration and, we can differentiate between the real structure and noise from an input point cloud.

## 3.2 The Mapper Algorithm

### 3.2.1 Introduction

The Mapper algorithm was originally proposed in a paper written by Singh, Mémoli, and Carlsson from Stanford University [39]. This algorithm visually simplifies a plot of multidimensional data points with strategic clustering. A part of the Mapper algorithm's robustness has to deal

with its independence from clustering algorithms and metrics, meaning the algorithm is able to generate outputs on any clustering algorithm and any metric. Hence, the algorithm intends on visualizing more general properties of the data set within the topological framework with its motivations from more theoretical perspectives.

### 3.2.2 Description of the Algorithm

The Mapper algorithm, essentially, takes an entire point cloud of data and produces a visualization as a simplicial complex. In its applications, we treat the simplicial complex more as a network of nodes from the point of view of exploratory analysis. In contrast, computations of persistent homology uses a sample of the initial data and generates a filtration of simplicial complexes and calculates persistent features across these. The steps are as follows:

If the data is high-dimensional, we first want to project it into a lower dimension using a filter function. Depending on the filter function $f : X \to \mathbb{R}$ we choose, where $X$ is our point cloud of data, the preserved structure of the data will be different. The filter function can be very straightforward. For example, in the original Mapper paper, the authors illustrate a projection from three dimensional coordinates representing a hand to the y-coordinate of each point. As opposed to projecting onto the x-coordinate or z-coordinate, the y-coordinate best preserves the original shape of the hand. Alternatively, the function can be more involved, such as using MDS (multidimensional scaling) or PCA (principal component analysis), where we want to summarize the data across dimensions. See Figure 10 for a simple projection onto the Length and Width columns of an artwork.

We then create an open cover of this lower-dimensional projection (often in $\mathbb{R}$ or $\mathbb{R}^2$). The definition of open cover of $X$ is a collection of subsets $X$ such that $X$ is in the union of these subsets. For example, if we have data on the real line on the open interval (0, 100) with overlap .5, the intervals are (0, 12.5), (7.5, 22.5), (17.5, 32.5) and so on. The user can change this overlap proportion to anywhere from 0- 1. If the overlap is 0, the resulting simplicial complex will have completely disconnected nodes, whereas if it is 1, each node will be connected. When we take the preimage of this cover, we also produce an open cover of the original point cloud since the filter function is continuous. [7]

For our purposes, the overlap given above is a good cover, since every nonempty intersection $\cap_{i\in\sigma} U_i$, for $\sigma \subseteq \{1, ..., n\}$, is contractible [9]. According to the Nerve theorem, the nerve of this cover, which is our output, has the same homotopic type as the original data.

The user can specify the number of covering intervals. Depending on the number of intervals, the graph can preserve more detail or less detail about the original data. In the original paper, this choice is referred to as the 'resolution'. While it is not proof that we have a "good cover" of the data, we can at least check if there is any information we are missing for one particular resolution. The 'ideal' number of intervals or overlap percentage depends on what we want to reveal about the data.

Once the data is projected in lower dimension and we have defined a cover on the interval,

we want to cluster on the "pullback of the cover". The pull back of the cover is simply the inverse of each set of the cover. That is, the preimage of intervals of the cover are subsets $X$ and $Y$ of the original data set. If $X \cap Y$ is nonzero, then there will clearly exist clusters in $X$ and $Y$ that will have nonzero intersection. The points in the inverse of each interval are clustered and form a node. Nodes are connected when the intersection between any two clusters in any two hypercubes is nonzero. This is idea of partial clustering. [39]

Lastly, we color these nodes by some continuous color function, such as Object Year, Area, etc. In the MET case, we color by whether an artwork is Public Domain or not, which is a boolean variable. We can also color by categorical function but the interpretation is less clear as to what a mix of colors is. In the arXiv case, we color by academic categories. We then look at the simplicial complex for any flares in the data or any uniform clusters of color. By flare we mean if part of the simplicial complex branches off from the main body or there is a split in the complex. If there is a distinguishing feature, we can examine the nodes in that cluster/flare for further statistical analysis or draw conclusion from their color.

### 3.2.3 Examples

We use the Python implementation of Mapper - Kepler Mapper 1.1.6 [21] to generate Mapper outputs in this paper. One straightforward application of Mapper is topological summary of 3D shapes. In the following example, we take the point cloud of a sphere (Figure 6). The filter function maps each point to its x, y-coordinates and the coloring function is the z-coordinate. We expect to see red and blue in the center, referring to the points farthest away from the equator of the sphere.

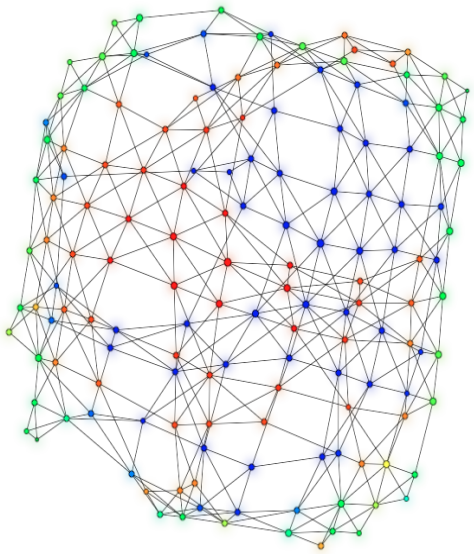
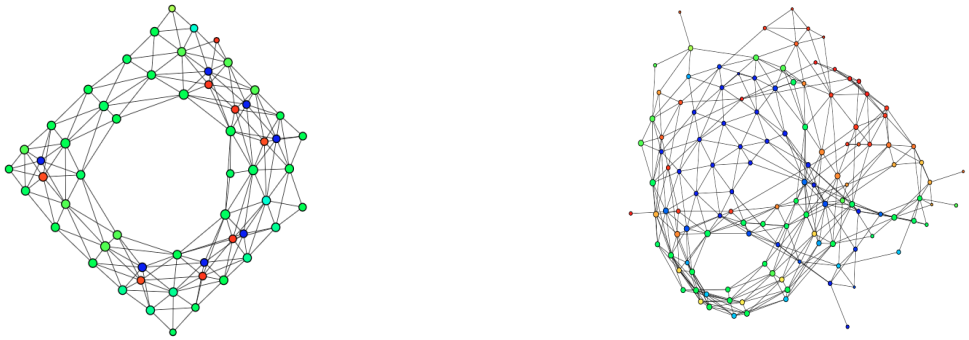


Figure 6: xy- projection of sphere, colored by z-coordinate

For torus 7, 8, we can see loop structure in both xy and yz projection. Notice how choice of lower resolution (Figure 7) can result in missing the structure of the torus.

(a) xy projection, colored by z-coordinate (b) yz-projection, colored by x-coordinate

Figure 7: Low resolution torus (n_cubes = 4, overlap = .6)



(a) xy-projection, colored by z-coordinate (b) yz-projection, colored by x-coordinate

Figure 8: High resolution torus, n-cubes = 7, overlap = .5

# 4 Applications

We employ these two TDA tools to analyze two datasets, the samples from the Metropolitan Museum of Art database and the arXiv database.

## 4.1 The MET Database

For the MET database, we use Mapper to examine the significant feature(s) that predict(s) if an artwork is in public domain or not. We also apply persistent homology to differentiate two groups of artworks. For initial explorations, see Appendix.

### 4.1.1 Data Collection and Cleaning

The original MET data [1] contains 43 columns and 450, 000 entries. From these we selected a subset of 13 columns of approximately 50,000 rows. We selected these columns for multiple reasons. One, many of the other columns we chose not to select contained upwards of 200,000 missing values. Secondly, other columns were difficult to use to classify the data since they had too many unique values (for example, the title of each artwork, or its link on the MET website). Third, on a basic level, we were interested in defining a relationship between our data points using these columns. Before cleaning our dataset, we have columns as shown in Table 1.

Next we clean our dataset (see Table 3). In the process of cleaning "Dimensions", we find that the values come in many formats. We choose to focus on artworks that have length and width measurements only, consolidated in one set of parentheses (for example, rows 0, 1, 2 in Table 2), since this is the most common format from those that we parse. During the process of cleaning "Dimensions", if a value is of this interested format, we first record the extracted dimensions in two newly created columns, "Length" and "Width", respectively, and then replace the value with the product of "Length" and "Width". If a value is not of the interested format, we just remove the entire entry from the dataset. Further,in some cases, multiple dates for "Artist Begin Date" or "Artist End Date" are included, in which case we replace these with the average. We think this is reasonable, as most of the dates were close together, except those where one contributor was a historical figure.

| Index | Is Highlight | Is Public Domain | Object ID | Department | Object Name |
|---|---|---|---|---|---|
| 173839 | FALSE | FALSE | 276173 | Photographs | Negative |
| 177828 | FALSE | FALSE | 281400 | Photographs | Polaroid |
| 409320 | FALSE | TRUE | 670612 | Photographs | Photograph |
| 232631 | FALSE | FALSE | 360381 | Drawings and Prints | Book |
| 255125 | FALSE | TRUE | 383756 | Drawings and Prints | Block |

| Index | Art. Begin | Art. End | Obj. Begin | Obj. End |
|---|---|---|---|---|
| 173839 | 1903 | 1975 | 1934 | 1975 |
| 177828 | 1903 | 1975 | 1974 | 1975 |
| 409320 | 1865 | 1931 | 1890 | 1931 |
| 232631 | 1764 \| 1767\|1777... | 1838\|1855 \|1865... | 1804 | 1838 |
| 255125 | 1471 | 1528 | 1493 | 1528 |

| Index | Medium | Dimensions |
|---|---|---|
| 173839 | Film negative | 8 x 10 in. |
| 177828 | Instant color print | 7.9 x 7.9 cm |
| 409320 | Gelatin silver print | Image: 4 in. ÃČ? 2 15/16 in. (10.1 ÃČ? 7.5 cm) |
| 232631 | Etching | 26 3/8 x 19 1/8 x 1 7/8 in. (67 x 48.5 x 4.8 cm) |
| 255125 | Black ink on carved pearwood | 15 1/2 ÃČ? 11 1/8 ÃČ? 1 in. (39.4 ÃČ? 28.3 ÃČ? 2.6 cm) |

| Index | Credit Line | Classification |
|---|---|---|
| 173839 | Walker Evans Archive, 1994 | Negatives |
| 177828 | Walker Evans Archive, 1994 | Photographs |
| 409320 | Purchase, Alfred Stieglitz Society Gifts, 2015 | Photographs |
| 232631 | The Elisha Whittelsey Collection, The Elisha Whittelsey Fund, 1969 | Books |
| 255125 | Gift of Junius Spencer Morgan, 1919 | Blocks |

Table 1: Random sample of 5 from original MET Dataset with selected features

| | Dimensions |
|---|---|
| 0 | sheet: 6 7/8 x 9 7/8 in. (17.5 x 25.1 cm) |
| 1 | 6 1/4 x 9 3/4 in. (15.9 x 24.8 cm) |
| 2 | sheet: 4 x 6 1/4 in. (10.2 x 15.8 cm) |
| 3 | 11 5/8 x 8 15/16 x 1 9/16 in. (29.6 x 22.7 x 4... |
| 4 | sheet: 18 3/8 x 13 11/16 in. (46.6 x 34.7 cm) |
| 5 | mount: 19 1/8 x 12 13/16 in. (48.5 x 32.5 cm)\... |
| 6 | 29 1/4 x 19 1/4 in. (74.3 x 48.9 cm) |
| 7 | sheet (trimmed within platemark): 3 7/8 x 5 11... |
| 8 | Sheet: 21 1/4 x 14 3/4 in. (54 x 37.5 cm)\... |
| 9 | Sheet: 10 1/2 x 5 1/16 in. (26.7 x 12.9 cm) |

Table 2: Dimensions column of random sample of original data

| Feature | Type | Cleaning Method |
|---|---|---|
| Is Highlight | Boolean | NA |
| Is Public Domain | Boolean | NA |
| Object ID | Integer | NA |
| Department | String | NA |
| Object Name | String→Set of strings | parse each value to extract a set of meaningful words, ignoring "and", "or", punctuation marks and other delimiters |
| Artist Begin Date | String→Integer | parse each value to extract integer(s) if multiple integers, take average |
| Artist End Date | String→Integer | same as “Artist Begin Date” |
| Object Begin Date | Integer | NA |
| Object End Date | Integer | NA |
| Medium | String→Set of strings | same as “Object Name” |
| Dimensions | String→Float | if the value is of interested format, 1) parse it to extract the two dimensions and 2) replace the original string with the product of these two dimensions if not, remove the whole entry |
| Credit Line | String→Set of strings | same as “Object Name” |
| Classification | String→Set of strings | same as “Object Name”, but ignore subclassfication after the dash |

Table 3: Cleaning Process

In using each tool of TDA, we analyze different subsets of this cleaned dataset, since we have a different research question for each tool. We will refer to this cleaned dataset as “the full dataset” in the rest of the paper. While using Kepler Mapper, we focus on the subset in which ’Artist Begin Date’ and ’Artist End Date’ are not missing, since we want to include more variables in our logistic regression model. We further discover that there are only 167 highlighted artworks in this dataset. Thus, “Is Highlight" does not differentiate between two artworks in a meaningful way, and we omit this feature. We will refer to this dataset as “the Mapper dataset” in the rest of the paper. The Mapper dataset has 12 columns and 50015 rows. We finally choose a random sample of 3000 to analyze.

In using persistent homology, we focus on the Mapper dataset and another subset of the full dataset in which entries have missing values for both “Artist Begin Date” and “Artist End Date”. We again omit “Is Highlight" from the latter dataset since 1) there are few highlighted artworks and 2) this feature is omitted from the Mapper dataset. We will refer to the latter dataset as “the contrast dataset” in the rest of the paper. The contrast dataset has 10 columns and 46878 rows. Finally we drop “Artist Begin Date" and “Artist End Date" from the Mapper dataset to make a more reasonable comparison between the Mapper dataset and the contrast dataset. We also

remove any row with missing value for any of the variables for these two datasets.

### 4.1.2 Methods and Results

#### 4.1.2.1 Mapper Application

At the initial stage of this project, we are inspired by Kraft's usage[24] of Mapper to guide logistical regression, a statistical model in which the dependent variable is binary. We think "Is Public Domain" is a good variable to model since it is binary. We think that it would be interesting to identify the feature(s) that discriminate between the group of public-domain artworks and the group of non-public-domain artworks. However, at this stage, we don't know how to incorporate Mapper into this process. Later, we read a paper Guo[17] in which Mapper is used to guide feature selection. Then we decide to follow the pipeline proposed by Guo to select important features associated with "Is Public Domain". The basic pipeline is:

1. Use Mapper to identify interesting subgroups
2. Perform the Wilcoxon Rank-Sum Test and proportion test on these subgroups to select relatively significant features
3. Compare the prediction accuracy of this subset of features to that of the full set of features

The Wilcoxon rank-sum test is a non-parametric test of the null hypothesis that two sets of measurements are drawn from the same distribution[22]. The proportion test tests whether one population proportion p1 equals a second population proportion p2[35]. Notice that in the first step, we drop "Is Public Domain" from our dataset since we want to examine the independent variables. "Is Public Domain" is the dependent variable and we use it as a color function in our Mapper output.

As introduced in the previous section, Mapper gives its users freedom in choosing parameters; however, this freedom comes with challenge. The challenge that using Kepler Mapper with our dataset poses is threefold.

The first challenge is choosing a filter function. We choose to use MDS (Multidimensional scaling)[6] as our filter function, projecting to $\mathbb{R}$. One advantage of MDS is that we can input a precomputed distance matrix with our metric. Also, MDS aims to preserve information about the distance between points in the original space. Since our data is mixed and for our purpose of feature selection, it is not meaningful to only project onto one or two numerical variables. For example, if two artworks have similar Length and Width, they are not necessarily related. In the initial stage of this project, we did project our data onto some feature as an exploration step (Appendix).

The second challenge is choosing a clustering algorithm. The Kepler Mapper default clustering method is DBSCAN (Density-based spatial clustering of applications with noise)[15]. The parameters are epsilon (the maximum radius for points to be considered close together), the metric, and minimum samples (the minimum points that a cluster can have). Since the default

is Euclidean metric, we have to pass our own metric (see the next paragraph) as parameter. We noticed that the pairwise distance matrix contains relatively large distances (with median about 4.5 and mean about 4.3), therefore the original epsilon value (0.5) was too small. Since our sample size is 3000 and we want to have reasonably large clusters, we set epsilon to 5 and mininum samples to 10. Choosing these parameters is a disadvantage of DBSCAN. However, we chose to use DBSCAN instead of K-family clustering algorithms because we do not have to specify the clusters in advance when we initialize the clusterer. Choosing number of clusters beforehand is considered as an undesired characteristic of a clustering algorithm in the original Mapper paper[39]. Also, for numerical data, while K-means has tendency to find convex clusters, DBSCAN is density based so is sensitive to other patterns[19]. At the initial stage of this project, we did explore K-family algorithms (see Appendix).

Another challenge is designing a metric to measure the distance (dissimilarity) between a pair of artworks. Since we have mixed types of data (numerical and categorical), we combine metrics for each type together to create our custom metric. Our custom metric is a sum of 13 subscores of distance, one for each feature. For categorical variables, we calculate the Jaccard distance[20]. For numerical variables, we calculate a subscore as $\frac{x_i - y_i}{maxDiff_i}$, where $x$, $y$ are two artworks, $x_i$, $y_i$ are their $i$th numerical feature's values, and $maxDiff_i$ is $max(x_i - y_i)$ for all $x, y$ in the dataset. For "Object ID", we output 0 if the two artworks are the same (they have same ID) and 1 otherwise. For "Department", since there are only 10+ unique values, we decide to calculate a subscore using the same method as the one for "Object ID". Notice that each subscore is within the range $[0, 1]$ and thus the maximum possible distance between two artworks is 13. See initial explorations for metric design in Appendix. In the next two propositions, we prove that our custom metric is a metric.

**Proposition.** Let $d_1$ be a metric on $X$ and $d_2$ a metric on $Y$. Define $d_3 : (X \times Y) \times (X \times Y) \to \mathbb{R}$ as $d_3((x_1, y_1), (x_2, y_2)) = d_1(x_1, x_2) + d_2(y_1, y_2) \forall x_1, x_2 \in X$ and $\forall y_1, y_2 \in Y$. $d_3$ is a metric.

*Proof.* We prove that $d_3$ satisfies every requirement for a metric.

Requirement 1: for a metric $d$, $d(x, y) \geq 0 \forall x, y \in X$.

Proof: since $d_1$ and $d_2$ are metrics, $\forall x_1, x_2 \in X$, $d_1(x_1, x_2) \geq 0$ and $\forall y_1, y_2 \in Y$, $d_2(y_1, y_2) \geq 0$. Thus $\forall x_1, x_2 \in X$ and $\forall y_1, y_2 \in Y$, $d_3((x_1, y_1), (x_2, y_2)) = d_1(x_1, x_2) + d_2(y_1, y_2) \geq 0$.

Requirement 2: for a metric $d$, if $x = y$, $d(x, y) = 0$.

Proof: since $d_1$ and $d_2$ are metrics, if $x_1 = x_2$, $d_1(x_1, x_2) = 0$ and if $y_1 = y_2$, $d_2(y_1, y_2) = 0$. Notice that by definition $(x_1, y_1) = (x_2, y_2)$ if and only if $x_1 = x_2$ and $y_1 = y_2$. Thus for $(x_1, y_1) = (x_2, y_2)$, $d_3((x_1, y_1), (x_2, y_2)) = d_1(x_1, x_2) + d_2(y_1, y_2) = 0$.

Requirement 3: for a metric $d$, if $d(x, y) = 0$, $x = y$.

Proof: since $d_1$ and $d_2$ are metrics, if $d_1(x_1, x_2) = 0$, $x_1 = x_2$ and $d_2(y_1, y_2) = 0$, $y_1 = y_2$. If $d_3((x_1, y_1), (x_2, y_2)) = d_1(x_1, x_2) + d_2(y_1, y_2) = 0$, then $d_1(x_1, x_2) = d_2(y_1, y_2) = 0$, then $x_1 = x_2$ and $y_1 = y_2$ and $(x_1, y_1) = (x_2, y_2)$.

Requirement 4: for a metric $d$, $d(x, y) = d(y, x) \forall x, y \in X$.

Proof: since $d_1$ and $d_2$ are metrics, $d_1(x_1, x_2) = d_1(x_2, x_1) \forall x_1, x_2 \in X$ and $d_2(y_1, y_2) = d_2(y_2, y_1)$, $\forall y_1, y_2 \in Y$. Thus, $d_3((x_1, y_1), (x_2, y_2)) = d_1(x_1, x_2) + d_2(y_1, y_2) = d_1(x_2, x_1) + d_2(y_2, y_1) = d_3((x_2, y_2), (x_1, y_1))$.

Requirement 5: for a metric $d$, $d(x, y) + d(y, z) \geq d(x, z) \forall x, y, z \in X$.

Proof: since $d_1$ and $d_2$ are metrics, $\forall x_1, x_2, x_3 \in X, d_1(x_1, x_2) + d_1(x_2, x_3) \geq d_1(x_1, x_3)$ and $\forall y_1, y_2, y_3 \in Y, d_2(y_1, y_2) + d_2(y_2, y_3) \geq d_1(x_1, x_3)$. Thus $\forall (x_1, y_1), (x_2, y_2), (x_3, y_3) \in X \times Y$, $d_3((x_1, y_1), (x_2, y_2)) + d_3((x_2, y_2), (x_3, y_3)) = d_1(x_1, x_2) + d_2(y_1, y_2) + d_1(x_2, x_3) + d_2(y_2, y_3) = d_1(x_1, x_2) + d_1(x_2, x_3) + d_2(y_1, y_2) + d_2(y_2, y_3) \geq d_1(x_1, x_3) + d_2(y_1, y_3) = d_3((x_1, y_1), (x_3, y_3))$. □

**Proposition.** The function $d : X \times X \to R$, defined as $d(x_1, x_2) =$

$$\begin{cases} 1 & x_1 \neq x_2 \\ 0 & x_1 = x_2 \end{cases}$$

is a metric, where $a \in \mathbb{R}$ is a constant.

*Proof.* We prove that $d$ satisfies every requirement for a metric.

Requirement 1 Proof: since the least value that $d$ outputs is 0, this requirement is satisfied.

Requirement 2, 3 and 4 Proof: by definition of $d$, this requirement is satisfied.

Requirement 5 Proof: there are 5 cases for $x_1, x_2, x_3 \in X$ and each satisfies Requirement 5. □

Also notice that our custom metric for numerical variables is indeed a metric since it is just a constant times the Euclidean distance for $\mathbb{R}$. Thus, our final metric, a combination of our metric for categorical variables (Jaccard distance), metric for numerical variables and metric for "Object ID" and "Department" (metric in the above proposition), by the first proposition, is indeed a metric.

Yet another challenge is Kepler Mapper is not compatible with non-numerical variables. Initially we think about assigning each string a meaningful numerical value. However, many of the categorical variables do not have a natural order. Thanks to Professor Ortiz, we realized that we could use dictionaries as an intermediate step. Then we defined dictionaries for each categorical variable. The values of the dictionaries are sets of strings (see Table 3); the keys were numerical values, which are simply the category codes for each variable. After creating the dictionaries, we replace the string values with the assigned numerical values in our dataset. Note that Mapper does not actually use the numerical value in the computation of the clustering, but rather only to reference the original value when calculating pairwise distances between data points.

Using this set of parameters for Kepler Mapper with "Is Public Domain" as the color function, we obtain an output graph that has two flares (see Figure 2). Nevertheless, these flares are not persistent through different resolutions or different samples of our subset (we tried a total of 12 random samples). However, there are almost always a red subgroup and a blue subgroup (such as the highlighted parts in Figure 9). Thus they are persistent features of our data. They are of two extreme colors, red and blue, indicating their extreme percentage of public-domain artworks. We choose to focus on these highlighted subgroups to guide feature selection.

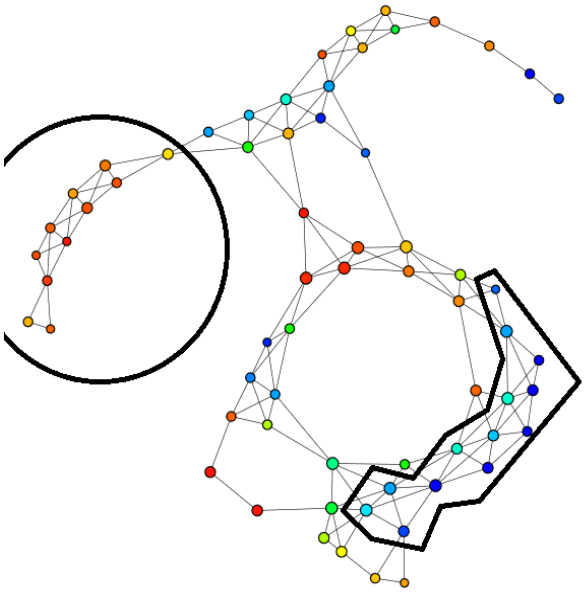


Figure 9: Mapper output with MDS as lens, DBSCAN as clustering algorithm and the custom metric. Number of hypercubes is 10 and percentage of overlapping is 0.35. Colored by "Is Public Domain". Red nodes have a high percentage of public-domain artworks in them, and blue nodes have a low percentage.

We now focus on the artworks in these two subgroups. We perform a proportion test on "Is Public Domain" of these artworks, and the p-value is significant at the 0.001 significance level, indicating there does exist a difference in the percentage of public domain artworks in these two subgroups. We then perform the Wilcoxon Rank-Sum Test to the continuous, numerical features and proportion test to dummy variables of "Department" (Table 4 and Table 5).

| | Features | | | | | | |
|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| p-value | 3.84e-31 | 1.01e-24 | 1.49e-30 | 5.16e-28 | 9.77e-59 | 4.81e-50 | 1.39e-61 |

Table 4: Results for Wilcoxon Rank-Sum Test. Features 1-7 correspond to: Artist Begin Date, Artist End Date, Object Begin Date, Object End Date, Area, Length, Width

| | Features | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| p-value | 5.65e-53 | 0.006 | 3.54e-42 | 8.58e-289 | 3.16e-74 | 4.02e-62 | 2.32e-07 | 3.20e-06 | 0.00072 |

Table 5: Results for proportion test. Features 1-9 correspond to: American Decorative Arts, Arms and Armor, European Sculpture and Decorative Arts, Drawings and Prints, Photographs, Arts of Africa, Oceania, and the Americas, Robert Lehman Collection, The Cloisters, Modern and Contemporary Art

Every feature could be considered statistically significant if we rely on p-values above, since the number of artworks in the focused subgroups is very large. Notice, however, that the dummy

variable “Drawings and Prints” has an outstanding p-value compared to other variables, 8.58e-289. Thus, we hypothesize that this single feature is a powerful discriminating feature between these two subgroups. To test our hypothesis, we build two logistic regression models on “Is Public Domain”, one based on “Drawings and Prints” that we call the “reduced model” and one based on the set of numerical features and dummy variables that we call the “full model”. We train and test our models on the sample of 3000 by using 80% of the sample as train data and 20% as test data. In this step, we use the train_test_split() method provided by sklean[34]. The model accuracy score, evaluated by the score() method of sklearn[34], of the reduced model is 52.17% and that of the full model is 73.83%. We repeat the train and test procedure on the Mapper dataset, and observe that the model accuracy score of the reduced model is 55.30% and that of the full model is 72.58%.

#### 4.1.2.2 Persistent Homology Application

Inspired by Carstens[8] and Zhu[43], we decide to use persistent homology to differentiate between the Mapper dataset and the contrast dataset. Interestingly, the two datasets are of similar size. The basic pipeline is:

1. Select random samples from each dataset
2. Identify their respective persistent intervals
3. Examine the artworks in these persistent intervals and identify the difference(s) between the artworks in different group’s persistent intervals

Notice that in this stage, we change our metric to accommodate the two datasets by dropping the subscores which evaluate the dissimilarity for “Artist Begin Date” and “Artist End Date" and adding in a metric measuring the dissimilarity for “Is Public Domain”, since in these two datasets we keep this boolean variable. As the metric for “Object ID” and “Department”, we output 1 if two artworks have the same value and 0 otherwise. We generate 12 random samples each of 150 artworks for each dataset (thus a total of 24 samples) and compute a distance matrix for each using our modified custom metric. For each sample, we compute barcodes for Dimension 1, the dimension of interest.

We notice that for Dimension 1 most of the 24 samples have at least one barcode in the interval $[4,5)$. The length of these barcodes is about 1, and for both groups these barcodes are among the longest (most persistent) ones considering all barcodes. Thus for both groups this interval is the most persistent. Further, we observe that for the group with missing Artist Begin Date and Artist End Date, there are, in general, more persistent barcodes (barcodes whose length is around 1) and particularly more persistent barcodes in this interval (Table 6, Table 7). This intuitively means that there are more loops in this relatively persistent interval.

Next, for each dataset we choose 3 of the 12 random samples to analyze the barcodes in $[4,5)$ (see Figure 10 for the barcode graphs of these 6 samples below and compare the graphs with Table 6 and Table 7).

| | Samples | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| num of barcodes aounrd [4,5) | 1 | 2 | 3 | 2 | 2 | 5 | 2 | 1 | 1 | 0 | 2 | 2 |
| num of persistent barcodes | 1 | 2 | 4 | 4 | 6 | 4 | 4 | 1 | 5 | 1 | 3 | 2 |

Table 6: Number of barcodes around [4,5) and number of barcodes of length around 1 for samples of the Mapper dataset

| | Samples | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| num of barcodes around [4,5)] | 5 | 6 | 4 | 4 | 4 | 11 | 3 | 1 | 4 | 1 | 5 | 4 |
| num of persistent barcodes | 7 | 6 | 7 | 5 | 5 | 12 | 6 | 2 | 7 | 1 | 10 | 7 |

Table 7: Number of barcodes around [4,5) and number of barcodes of length around 1 for samples of the contrast dataset

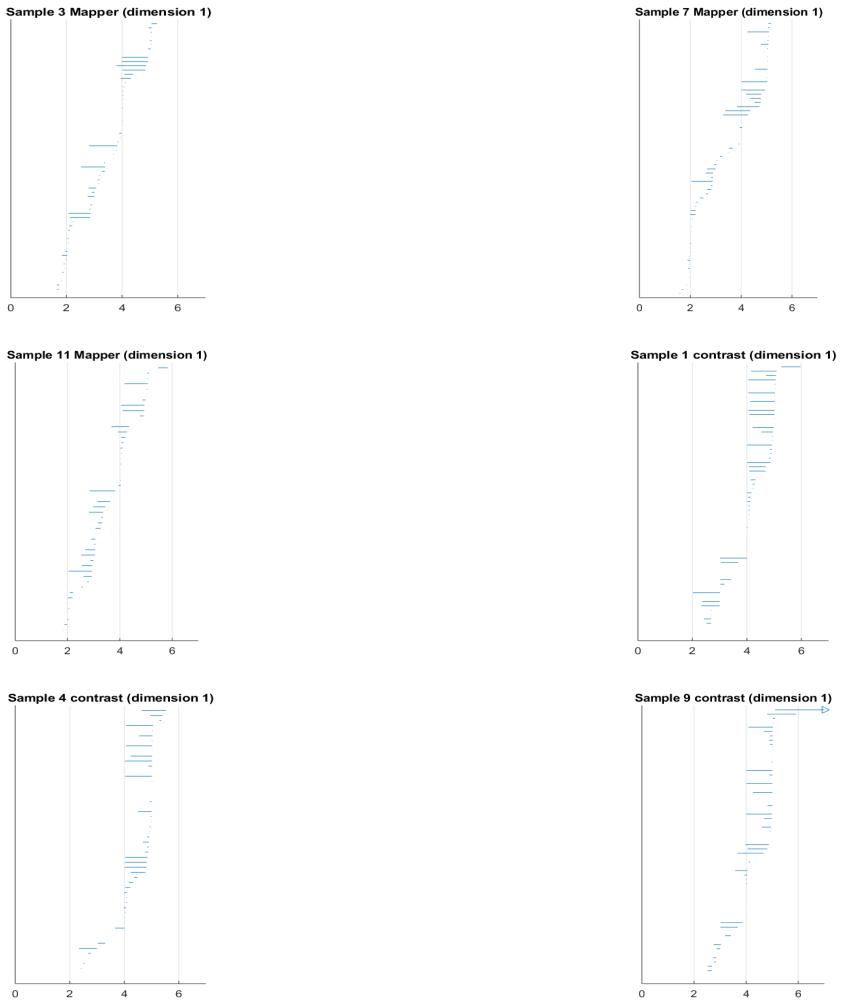
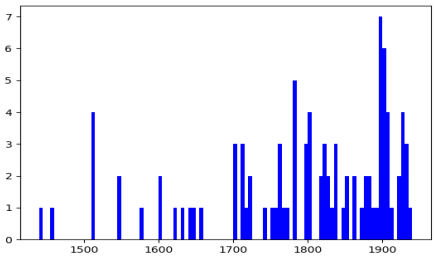
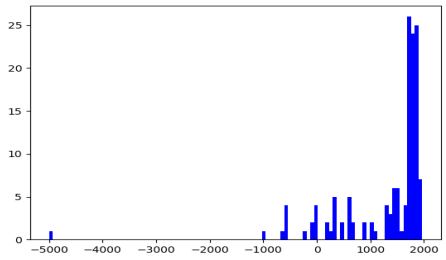



Figure 10: Dimension 1 barcodes for 6 selected samples

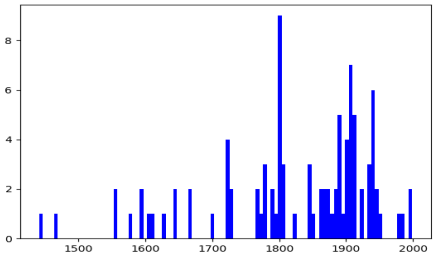
Recall that barcodes track the birth and death of a homology generator, a homology class. For any barcode, javaPlex chooses one class representative and gives the indices of the artworks which are the vertices of this representative. We choose to analyze the artworks which form the loops in $[4,5)$ in these 6 samples (recall that barcodes in Dimension 1 represent loops). See Appendix for the roadblocks we encounter when we try to extract these artworks. We observe that most of the artworks that form the Mapper dataset's loops are in the department "Drawings and Prints", whereas there are more various kinds of artworks in the loops of the contrast dataset. However, the artworks in the loops of the contrast dataset have less various credit lines and Medium. The distribution of each numerical variable is different for the artworks from the two datasets (Figure 11, Figure 12).



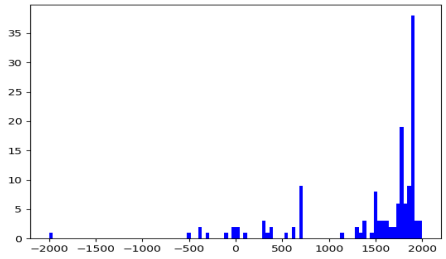

(a) Distribution of Object Begin Date for artworks in the loops of the Mapper dataset




(b) Distribution of Object Begin Date for artworks in the loops of the contrast dataset

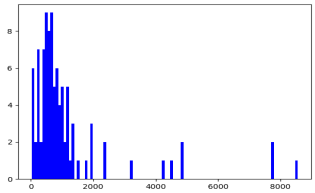



(c) Distribution of Object End Date for artworks in the loops of the Mapper dataset

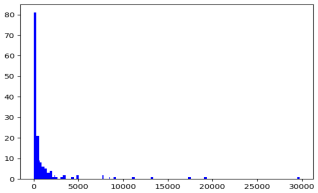



(d) Distribution of Object End Date for artworks in the loops of the contrast dataset

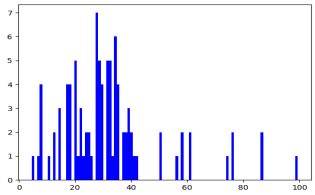
Figure 11: Distribtion of Object End Date for artworks in the loops



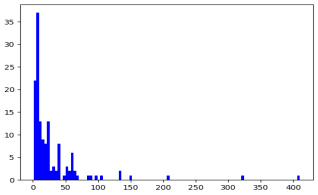

(a) Distribution of Area for artworks in the loops of the Mapper dataset



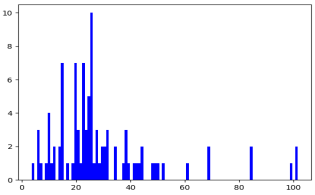

(b) Distribution of Area for artworks in the loops of the contrast dataset



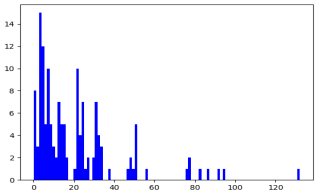

(c) Distribution of Length for artworks in the loops of the Mapper dataset




(d) Distribution of Length for artworks in the loops of the contrast dataset




(e) Distribution of Width for artworks in the loops of the Mapper dataset




(f) Distribution of Width for artworks in the loops of the contrast dataset

Figure 12: Distribution of Area (Length × Width), Length, Width for artworks in the loops

### 4.1.3 Discussion

From the Mapper application, we see that TDA can guide feature selection. The relatively close model accuracy scores indicate the Mapper algorithm's effectiveness. From the persistent homology example, we see that TDA can guide classification. Given a sample from the Mapper dataset and a sample from the contrast dataset, one can potentially differentiate between them by comparing the number of persistent barcodes and the distributions of numerical variables. Our applications show that TDA is a good approach to big and complex data.

However, our study has several limitations. Since we only focus on artworks with 2 dimensions, our results cannot be applied to a larger dataset. Moreover, due to the availability of data, we have limited kinds of numerical features, which can lead to omitted variable bias in our statistical models.

## 4.2 The arXiv e-print Database

### 4.2.1 Introduction

For this part of the project, we focused our attention on the arXiv (pronounced "archive"), which is an online repository hosted by Cornell University for academic paper submissions, mostly in the natural sciences and other quantitative fields. This online archive consists of electronic preprints (e-prints) that have been approved for publication. The subject of the these preprints consist of scientific papers ranging from subjects in STEM and are available to the public for free. More importantly, the service also offers an API framework to facilitate the programming tasks involving data retrieval from the repository. Thus, we took the opportunity of this open access and chose a small subset of the data to analyze using TDA. We formed a topological space of our point cloud of papers and experimented with various metrics to achieve persistent homology results.

Initially, we began the project by exploring the API documentation that arXiv offered to the public. We explored a python module that would allow easy access to the arXiv API and exploration thereof. However, we reached a roadblock when we fooled around with extracting too many files at once from the online API. We soon found out that this was making our script look like a "robot" which was strictly prohibited on arXiv's terms. As a result, the arXiv blocked us from retrieving and storing data directly from the online API. It was a learning experience in that we learned the significance of reading the terms and conditions of any service we partake in. Despite this obstacle, we approached other solutions. We instead looked at attaining bulk data from Amazon S3, a cloud computing web service, which allowed us to download a significant amount of data from arXiv for a very small fee. After doing so, we were able to download roughly 2000 pdf files (2286 files to be exact) ranging from all subjects of the arXiv in the year 2000.

### 4.2.2 Data Collection and Cleaning of the 125 TDA Papers

Data collection for this arXiv was made quick and possible due to the free API usage made available to the public on arXiv.org[3]. Following guidelines from the API and exploring a python module called "arxiv", we were able to extract a subset of query results personalized that whatever parameters we gave. In the beginning, we mainly concerned ourselves with one subject: topological data analysis. We specialized our code to retrieve 125 papers containing the phrase "topological data analysis" in the abstract. From these, we used various python modules to clean the data by parsing out only wanted features of the query results. See the following table summarizing the features extracted.

| Feature | Type | Cleaning Method |
|---|---|---|
| Authors | Set→List of Strings | parse out delimiters and create column for cleaned strings |
| Number of Authors | List of Strings→Integer | compute the number of authors from "Authors" per paper |
| Weights | Float | Computed as $\omega = \frac{1}{n-1}$ where n is the number of authors, if =1, $\omega = 0$ |
| Summary | Set→List of Strings | parse out delimiters and create a column for cleaned strings |
| Length of Summary | List of Strings→Integer | compute the number of words in each summary from "Summary" |
| Published Date | String→Float | parse out delimiters and create a column of time values computed as time = $\frac{x}{avg(x)}$ where x is the number of seconds from 1970 to the published date |
| Academic Category | String→Integer | parse out academic categories and create an array for use as a color function |

#### 4.2.3 Methods and Results of the 125 TDA Papers

When we were able to download and save 125 papers whose abstract contained the following phrase: "topological data analysis", we were able to extract the author names, academic category, summary detail, published date, and updated date. We kept only the author names and dates at the time. By computing the length of number of authors per paper, we constructed a weight function similar to the one used in the paper by Carstens and Horadam.[8]

$$\omega = \frac{1}{n-1} \tag{1}$$

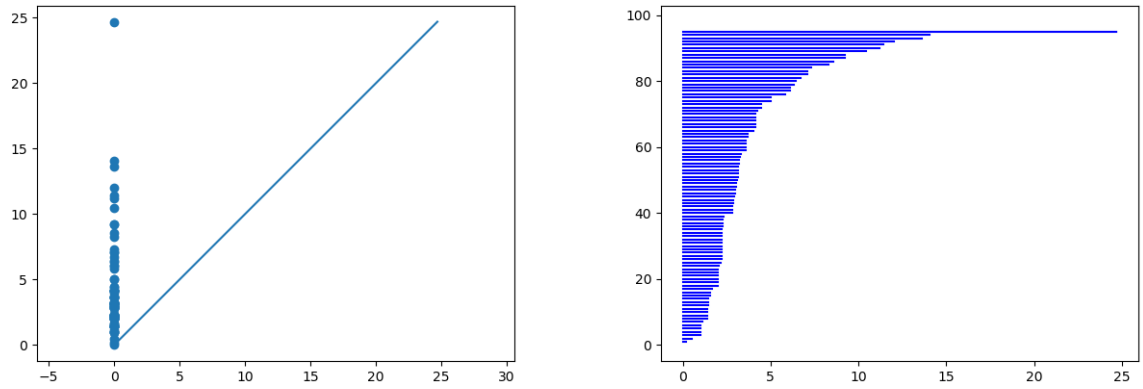
where n = number of authors.

This weight function allowed us to describe the "closeness" of the relationships between coauthors in a paper. If the paper only had one author, then we disregarded this paper out of the DataFrame. We establish that if there are many coauthors in a paper, the relationship between the coauthors would not be as strong as papers with fewer coauthors. This is assuming that when a lot of people work together, the people in this large group may not know each other as well than for people in a smaller group. As a result if $\omega$ was very close to 0, then the relationship between coauthors would be weak, versus, if $\omega$ was very large, the relationship between coauthors would be close. We added these values as a new column in our DataFrame. Other

than this, we were able to add more columns to the dataframe. By using a python module called "PyPDF2" (cite) we were able to extract the number of pages for each article. Using "refextract" (cite) we were able to extract references and the amount of references each paper has.

After creating a stable DataFrame to work with, we created a csv file and stored the information locally on our computers. We next created a distance matrix using "scipy" to compute pairwise distances from predefined metrics (cite). From this, we explored Dionysus, a C++ library with python bindings, that specializes in computing persistent homology (cite). We created persistent diagrams and barcodes for multiple dimensions, mainly the 0th and 1st dimension. Then, we used this to analyze the higher dimensional shape and features of our data by spotting the homology generators. We used Mapper to help us understand further what our data looks like in 3D space. This allowed us the possibility to connect ideas from our Dionysus results to our Mapper visualization. When we explored various metrics to define distances between pairs of points in our data, we worked with predefined metrics, such as the Euclidean metric, before exploring customized metrics.

We used a 3 dimensional array consisting of the following parameters: weights, page numbers, and reference numbers. Figure 13a and Figure 13b portray the persistent diagram and barcodes, respectively, for the 0th dimension of the 125 TDA papers using the Euclidean metric. From the barcodes, we analyze to find persistent bars that indicate generators for the 0th homology group which give us the 0th Betti number. Note that the persistent diagram has one point that is very far apart form the normal line, thus indicating a persistent bar, which we can also clearly see in the barcodes diagram. From these results, we can say the 0th Betti number is 1.

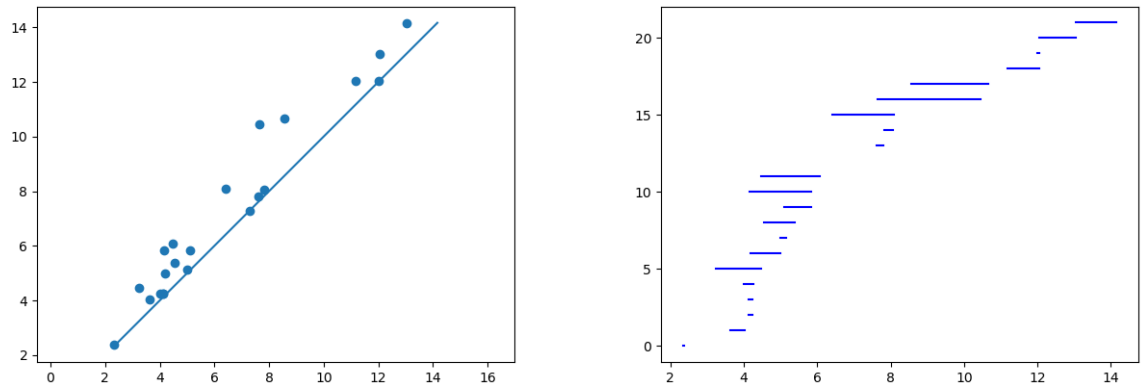



(a) Persistence diagram

(b) Barcodes

Figure 13: Persistent homology results for the 0th homology group using the Euclidean metric

For Figure 14a and 14b, we have a bit more complicated results but the same principles apply. We can see a persistent barcode from $\epsilon \approx 7.5$ to $\epsilon \approx 11$ in Figure 14b. This signifies that the first Betti number is at least 1 (with judgment).




(a) Persistence diagram

(b) Barcodes

Figure 14: Persistent homology results for the 1st homology group using the Euclidean metric.

When we look at the Mapper output for this dataset using the euclidean metric, we do indeed see a loop in one of our clusters as seen in Figure 15. However, we cannot confirm that this loop is a result from the Dionysus output. We can, however, analyze the clusters that arrive from the Mapper algorithm, DBSCAN, and the lenses to try to understand why we get certain loops or clustering.

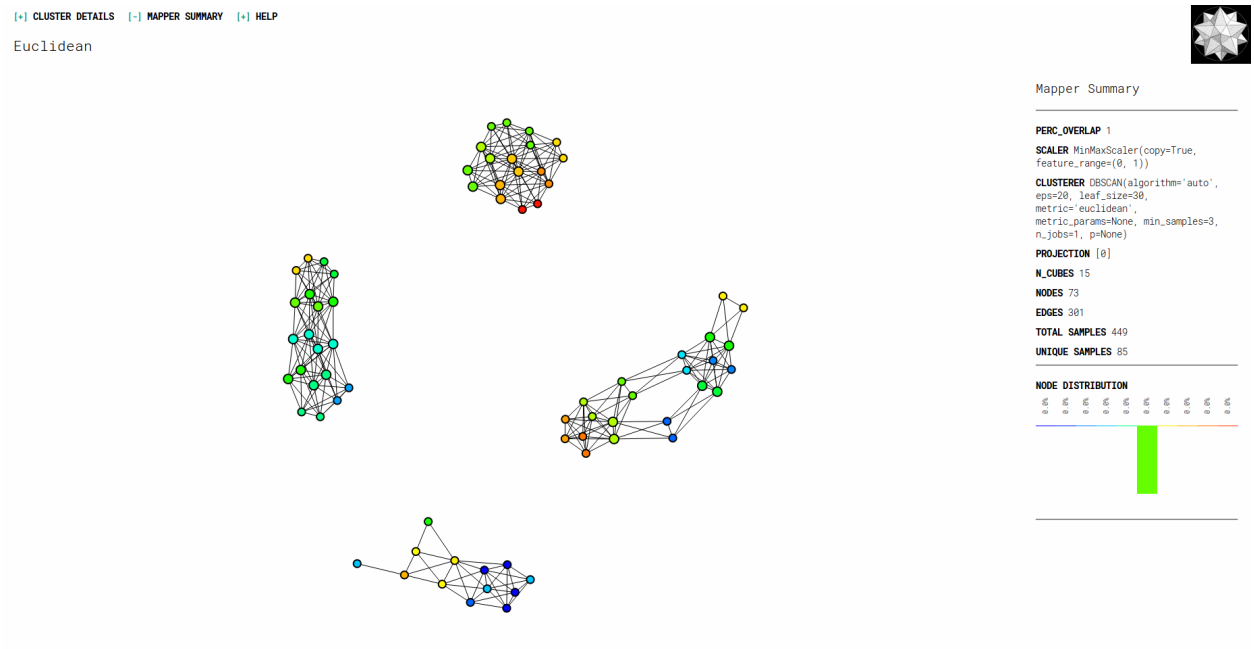
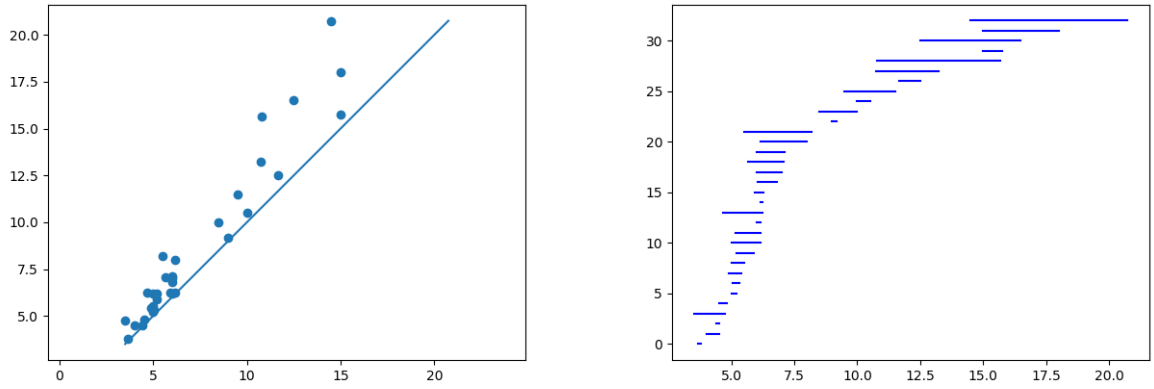



Figure 15: Mapper output for the 125 TDA papers using the Euclidean metric, colored by weights

Next, we explored a different predefined metric called the Manhattan metric, also commonly known as the "city-block" metric. In Figure 16a we see the persistence diagram for the

first dimension. As one can see, as the radius of our epsilon increases, more points seem to appear farther from the normal line. This is shown as persistent barcodes near $\epsilon \approx 20$ in Figure 16b. This result for the first dimension is promising, since we can say that there is at least one homology generator for this dimension. The 1st Betti number appears to be at least 3 (given judgment).




(a) Persistence diagram

(b) Barcodes

Figure 16: Persistent homology results for the 1st homology group using the Manhattan metric.

Following this, we explored its Mapper result, and from the visualization, we see primarily four clusters. Since the weight was also projected onto and used as a coloring function, there is a pattern of colors in the clusters as there was in the Euclidean Mapper output. Figure 17. After comparing the Manhattan results to the Euclidean results, we decided to create our own metric.

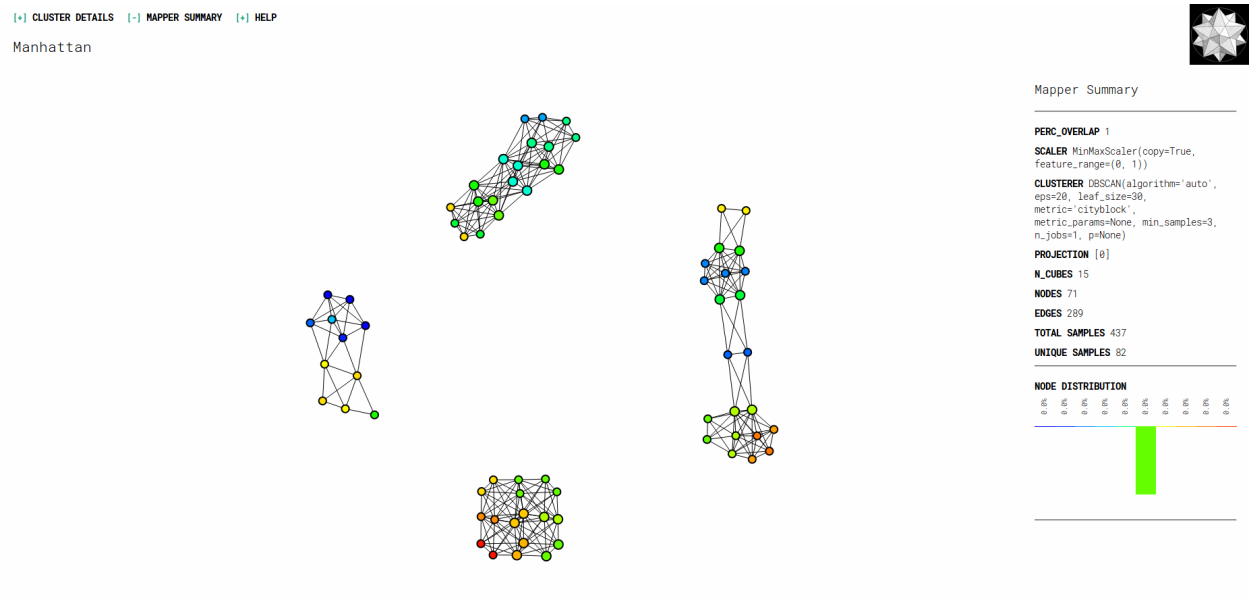



Figure 17: Mapper output for the for the 125 TDA papers using the Manhattan metric

To create a new metric, we considered the persistent homology results from the Euclidean and the Manhattan metric. We started off simply by deciding to manipulate predefined metrics so that our customized metric would be a linear combination of other predefined metrics. We knew that since a metric space is a vector space that the vector axioms would still apply. We were inspired by the absolute value of a difference of values from the Manhattan metric. It produced promising results as seen in Figure 16a and Figure 16b. Therefore, we decided to do something similar. With that being said we defined the following metric and produced interesting results with our dataset using this metric.

**Proposition**: The following formula, named $d_{arXiv}(x,y)$, is a metric in the space of papers in our point cloud:

$$d(x,y) = \sum_{i=1}^{n} | \frac{1}{x_i+1} - \frac{1}{y_i+1} | . \tag{2}$$

*Proof.* We show that Equation(1) follows the following four conditions of a metric space:

1. $d_{arXiv}(x,y) \geq 0$
2. $d_{arXiv}(x,y) = 0$ if and only if $x = y$
3. $d_{arXiv}(x,y) = d_{arXiv}(y,x)$
4. $d_{arXiv}(x,z) \leq d_{arXiv}(x,y) + d_{arXiv}(y,z)$

First, let $x, y \in \mathbb{R}^n$ be arbitrary vectors.
For the first condition, note that the components of $x$ and $y$ are real numbers. Let $x_1$ and $y_1$ be arbitrary real numbers as components of $x$ and $y$ respectively. Then,

$$| \frac{1}{x_1+1} - \frac{1}{y_1+1} | \geq 0 \tag{3}$$

since the terms in the absolute value are real. Since $x_1$ and $y_1$ are abitrary, this results applies for all $x_i$ and $y_i$ in $1 \leq i \leq n$. Thus, the first condition is satisfied.
For the second condition, we first assume the forward direction. Say $d_{arXiv}(x,y) = 0$. Then,

$$\begin{aligned} 0 &= \sum_{i=1}^{n} | \frac{1}{x_i+1} - \frac{1}{y_i+1} | \\ &= | \frac{1}{x_1+1} - \frac{1}{y_1+1} | + | \frac{1}{x_2+1} - \frac{1}{y_2+1} | + \ldots + | \frac{1}{x_n+1} - \frac{1}{y_n+1} |. \end{aligned} \tag{4}$$

As before, it must be that the absolute value of the difference of two real numbers is either 0 or greater. So, if the finite sum is equal to 0, it must be that each term is equal to 0. Thus for any $i$ in $1 \leq i \leq n$, we have that

$$| \frac{1}{x_i+1} - \frac{1}{y_i+1} | = 0 \tag{5}$$

and so it must be that

$$\frac{1}{x_i+1} = \frac{1}{y_i+1} \tag{6}$$

which leads to $x_i = y_i$ for all $i$. For the reverse direction, we first assume that $x = y$. If this is the case, then $x_i = y_i$ for all $i$ in $1 \leq i \leq n$, and we simply note as before that $|\frac{1}{x_i+1} - \frac{1}{y_i+1}| = 0$, and so $0 = \sum_{i=1}^{n} |\frac{1}{x_i+1} - \frac{1}{y_i+1}| = d_{arXiv}(x, y)$.
For the third condition, recall that for any real value $a, b \in \mathbb{R}$, we have that by the definition of taking the absolute value that $|a-b| = |b-a|$. Since $\frac{1}{x_i+1}$ and $\frac{1}{y_i+1}$ are both real values for all $i$, then it must be that

$$|\frac{1}{x_i+1} - \frac{1}{y_i+1}| = |\frac{1}{y_i+1} - \frac{1}{x_i+1}| \tag{7}$$

for all $i$ in $1 \leq i \leq n$. Thus,

$$\sum_{i=1}^{n} |\frac{1}{x_i+1} - \frac{1}{y_i+1}| = \sum_{i=1}^{n} |\frac{1}{y_i+1} - \frac{1}{x_i+1}| \tag{8}$$

and so $d_{arXiv}(x, y) = d_{arXiv}(y, x)$. For the fourth condition, recall the following similar inequality:

$$|a-b| \geq |a| - |b| \tag{9}$$

We start off by first computing the following: $d_{arXiv}(x, y) + d_{arXiv}(y, z)$. We apply the definition of our metric defined above and expand the sum in 10.

$$\begin{aligned} d_{arXiv}(x, y) + d_{arXiv}(y, z) &= \sum_{i=1}^{n} |\frac{1}{x_i+1} - \frac{1}{y_i+1}| + \sum_{i=1}^{n} |\frac{1}{y_i+1} - \frac{1}{z_i+1}| \\ &= |\frac{1}{x_1+1} - \frac{1}{y_1+1}| + |\frac{1}{x_2+1} - \frac{1}{y_2+1}| + \ldots + |\frac{1}{x_n+1} - \frac{1}{y_n+1}| \\ &\quad + |\frac{1}{y_1+1} - \frac{1}{z_1+1}| + |\frac{1}{y_2+1} - \frac{1}{z_2+1}| + \ldots + |\frac{1}{y_n+1} - \frac{1}{z_n+1}| \end{aligned} \tag{10}$$

We then apply equation 9 to equation 10 and simplify. Note that the $\frac{1}{y_i+1}$ terms for $i \in 1 \leq i \leq n$ cancel.

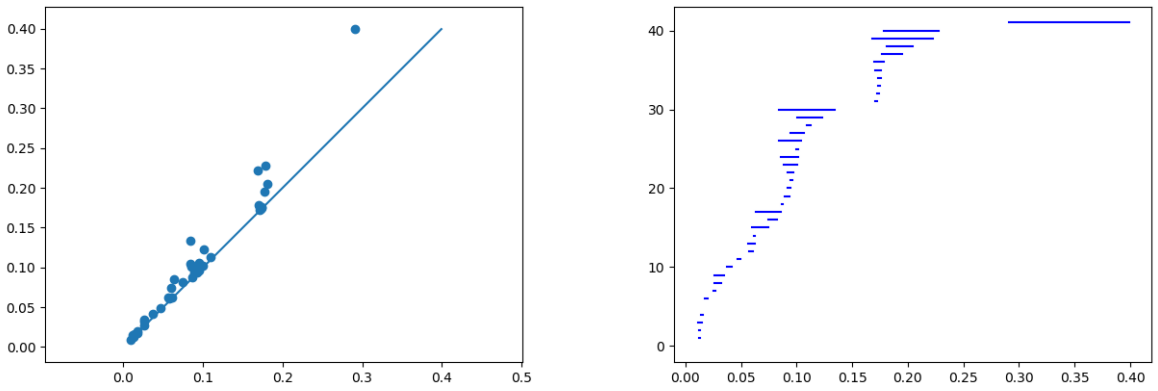
$$\begin{aligned} &= (|\frac{1}{x_1+1} - \frac{1}{y_1+1}| + |\frac{1}{x_2+1} - \frac{1}{y_2+1}| + \ldots + |\frac{1}{x_n+1} - \frac{1}{y_n+1}| + |\frac{1}{y_1+1} - \frac{1}{z_1+1}| \\ &\qquad + |\frac{1}{y_2+1} - \frac{1}{z_2+1}| + \ldots + |\frac{1}{y_n+1} - \frac{1}{z_n+1}|) \\ &\geq (|\frac{1}{x_1+1}| + |\frac{1}{x_2+1}| + \ldots + |\frac{1}{x_n+1}| - |\frac{1}{y_1+1}| - |\frac{1}{y_2+1}| - \ldots - |\frac{1}{y_n+1}| + |\frac{1}{y_1+1}| \\ &\qquad + |\frac{1}{y_2+1}| + \ldots + |\frac{1}{y_n+1}| - |\frac{1}{z_1+1}| - |\frac{1}{z_2+1}| - \ldots - |\frac{1}{z_n+1}|) \\ &= (|\frac{1}{x_1+1} - \frac{1}{y_1+1}| + |\frac{1}{x_2+1} - \frac{1}{y_2+1}| + \ldots + |\frac{1}{x_n+1} - \frac{1}{y_n+1}| + |\frac{1}{y_1+1} - \frac{1}{z_1+1}| \\ &\qquad + |\frac{1}{y_2+1} - \frac{1}{z_2+1}| + \ldots + |\frac{1}{y_n+1} - \frac{1}{z_n+1}|) \\ &\geq (|\frac{1}{x_1+1}| + |\frac{1}{x_2+1}| + \ldots + |\frac{1}{x_n+1}| - |\frac{1}{z_1+1}| - |\frac{1}{z_2+1}| - \ldots - |\frac{1}{z_n+1}|) \end{aligned} \tag{11}$$

Note that in equation 11 we also have the following:

$$(|\frac{1}{x_1+1}|+|\frac{1}{x_2+1}|+\ldots+|\frac{1}{x_n+1}|-|\frac{1}{z_1+1}|-|\frac{1}{z_2+1}|-\ldots-|\frac{1}{z_n+1}|)=d_{arXiv}(x,z) \quad (12)$$

Thus, $d_{arXiv}(x,z) \leq d_{arXiv}(x,y)+d_{arXiv}(y,z)$ as we have shown. Since this formula satisfies the four conditions listed above, and since $x$ and $y$ are arbitrary, the result follows. □

This metric proved to be an interesting metric once we saw the barcodes it produced for the 125 papers (see Figure 18b). In the persistence diagram showing the 1st dimension in Figure 18a we see at least three instances where persistent barcodes appear.




(a) Persistence diagram

(b) Barcodes

Figure 18: Persistent homology results for the 1st homology group using $d_{arXiv}$

In our Mapper output, we do in fact see one hole in Figure 19. However, as before, we cannot make the conclusion that the persistent barcode seen in the persistent diagram occurring around $\epsilon \approx 0.3$ is the loop generated in the Mapper visualization output. One can speculate, however, how this loop did manage to appear by thinking about the clustering algorithm used and any lenses. In this case, we projected on the weights column in our DataFrame. Moreover, we happened to color by these values as well. Other than this projected, we used the Isolation Forest lens and the Multidimensional scaling lens.

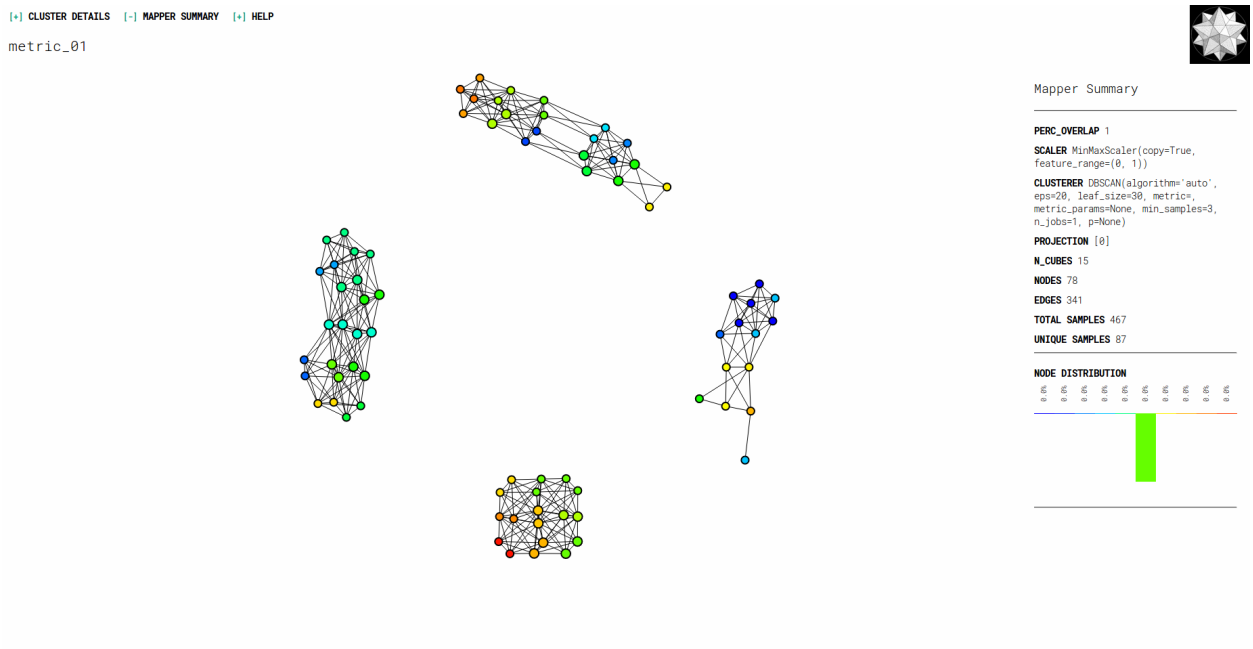



Figure 19: Mapper output for the 125 TDA papers using $d_{arXiv}(x, y)$, colored by weights

### 4.2.4 Discussion of the 125 TDA Papers

When we first encountered results from the Euclidean metric, we saw potential for interpretation. From the 0th dimension, after given a certain epsilon ($\epsilon \approx 15$) we would achieve just one cluster, a large simplex connecting all the 125 TDA papers. When we look at the 1st dimension, it is quite interesting to try to interpret the loops that could possibly have formed in our small subset. As a time constraint, we were not able to program our script to achieve this. However, we can still interpret this results as proof that our data has some form of shape. We could not achieve any results in the 2nd dimension, i.e. we could not create persistent homology for the 2nd homology group. This tells us that though our data has shape of loops, it does not have any voids in 3D space.

As for the Manhattan metric, our results turned out a bit more promising. Unlike the Euclidean metric where persistent barcodes for the 1st homology group where spread out, in the Manhattan metric, we noticed a trend that as our epsilon grew, more and more loops would develop. The idea that this could happen is what sparked our interest when we decided to create our own metric. When we look at the persistent homology results for $d_{arXiv}$, there does indeed appear to be at least four persistent barcodes that represent loops in our point cloud of papers. These signify that there is some type of relationship among certain types of authors.

When we interpret the Mapper visualization output for all of these three metrics, we take into consideration of what clustering algorithm we used, the epsilon parameter we chose, any projections (or lenses), and a color function (if one was defined). In all the three Mapper outputs given, a color function was defined by coloring by weights (the same weights from equation 1). As one can see in all three Mapper outputs, there is a pattern of how closely different nodes

are close to each other based on color. This is simply because the weight values were also part of the data. to understand this, one must also understand any projections and the clustering algorithm used. In all three of the Mapper outputs, we projected onto the weights column in our DataFrame, onto the Euclidean L2-norm, and used an Isolation Forest lens. Given the persistent homology results, and given the three unique lenses we used, it is quite interesting to see a loop in the Mapper result that uses $d_{arXiv}(x, y)$. We can say that after looking at the persistent homology results and Mapper, this metric holds some promise for a larger subset.

#### 4.2.5 Roadblock

Initially, we tried to increase our subset by simply changing our script to read more query results from the online API that arXiv provides. However, we hit a roadblock when suddenly one day, we discovered that our IP address was no longer able to access arXiv's online API service. We immediately realized that our scripts that we were running were starting to look like "robots", i.e. arXiv's servers were detecting our requests as "mindless searches". We learned that this caused stress on arXiv's servers and that arXiv strictly prohibited this on their online website. We were not quite aware of this restriction until we became banned.

#### 4.2.6 Data Collection and Cleaning of the 2,285 Papers

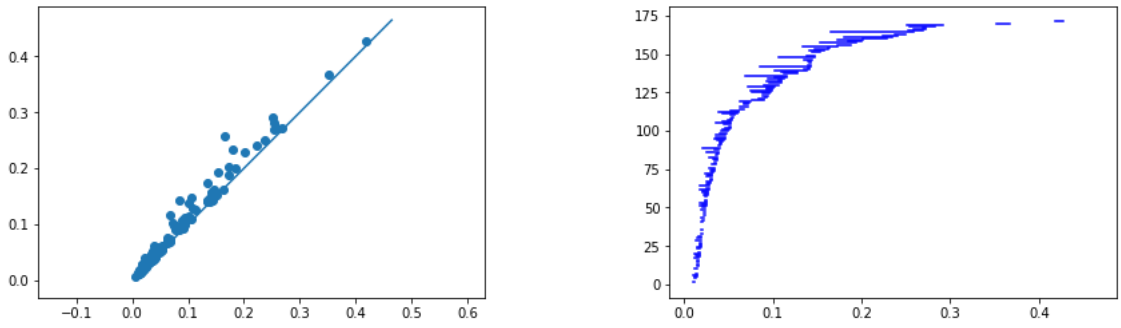
After being banned from storing data directly from arXiv's API due to a naïve neglect of arXiv's restriction on automated downloads, we resorted to Amazon S3 and downloaded a considerable amount of papers (for $1.00). To this dataset, we performed a similar search as we did before with the 125 papers. We stored all of this information into a a dictionary and converted it into a dataframe where we could manipulate and add new information columns from previous columns. We started off with collecting author names and published date. From this, we used the other python modules, "PyPDF2" and "refextract", to get the number of pages per article and the number of references as well. Overall, we decided to collect 2,285 papers from the year 2000 ranging in the following subjects: astrophysics, condensed matter, physics, math, high energy physics, computer science, nuclear, nonlinear sciences, quantum physics, quantum cosmology, and general relativity.

#### 4.2.7 Methods and Results of the 2,285 Papers

With the freedom to search over 2000 papers, we explored a much larger dimensional dataset. In the end, we decided to create a dataframe that was 6 dimensional, containing the following features for each paper: weights (as defined by equation 1), number of pages, number of references, length of authors, length of summaries, and a time value depicting the published date. With this large dataset, computing persistent homology became a challenge to our computers. As a result, when exploring Dionysus, we resorted to exploring various random samples of the dataset. From these random sampling, we chose certain graphs that best represented the general shape of the dataset given certain parameters.

For the metric $d_{arXiv}(x, y)$ we explored two different epsilon values: 0.5 and 1.0. The epsilon values chosen are quite different than the epsilon values chosen for the 125 TDA papers. This

is simply because the distance matrix produced had different magnitudes of values compared to the distance matrices for the 125 TDA papers. Figure 20a shows the persistent diagram of the 1st dimension after applying the metric $d_{arXiv}(x, y)$ to the dataset with an epsilon of 0.5. From the persistence diagram, a point around $\epsilon \approx 0.2$ can be spotted to appear far away from the diagonal. Figure 20b shows the respective barcodes, and as understood from the persistent diagram, a persistent barcode appear to start around $\epsilon \approx 0.2$.

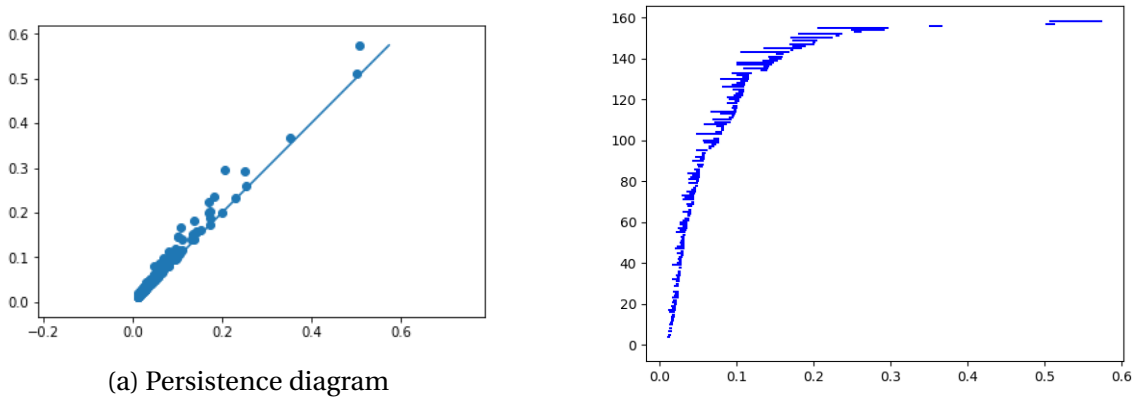


(a) Persistence diagram

(b) Barcodes

Figure 20: Persistent homology for the 1st homology group using $d_{arXiv}(x, y)$ with an epsilon of 0.5 for a random sample the larger dataset containing 2,285 papers

For epsilon 1.0, at a glance, we get very similar results for the 1st homology group that do not look far off from the results produced by epsilon 0.5. In Figure 21b it is quite harder to see if there is a persistent barcode or not. However, if we look at the persistence diagram in Figure 21a, there seems to be a potential candidate around $\epsilon \approx 0.2$. Despite changing the epsilon, we can attain a persistent homology generator around this epsilon of 0.2.



(a) Persistence diagram

(b) Barcodes

Figure 21: Persistent homology for the 1st homology group using $d_{arXiv}(x, y)$ with an epsilon of 1.0 for a random sample the larger dataset containing 2,285 papers

Once we explored the territory of Mapper, interesting results arose. The color function was made by creating an array of the academic categories that arXiv has to offer. For example, all

papers belonging to "math" would be colored one certain color. While, all papers belonging to "physics" would be colored another certain color.

In Figure 22 we explored the dataset with the metric $d_{arXiv}(x,y)$. The DBSCAN clustering algorithm was applied with an epsilon of 0.5. As for the lenses, we applied three lenses: the Isolation Forest lens, the Multidimensional Scaling lens, and a projection onto the number of pages column in our dataframe.

In Figure 23 we explored the dataset with the metric $d_{arXiv}(x,y)$. The DBSCAN clustering algorithm was applied with an epsilon of 1.0. As for the lenses, we applied three lenses: the Isolation Forest lens, the Multidimensional Scaling lens, and a projection onto the number of pages column in our dataframe.

In Figure 24 we explored the dataset with the metric $d_{arXiv}(x,y)$. The DBSCAN clustering algorithm was applied with an epsilon of 0.5. As for the lenses, we applied three lenses: the Isolation Forest lens, the Multidimensional Scaling lens, and a projection onto the number of references column in our dataframe.

In Figure 25 we explored the dataset with the metric $d_{arXiv}(x,y)$. The DBSCAN clustering algorithm was applied with an epsilon of 1.0. As for the lenses, we applied three lenses: the Isolation Forest lens, the Multidimensional Scaling lens, and a projection onto the number of references column in our dataframe.

In Figure 26 we explored the dataset with the metric $d_{arXiv}(x,y)$. The DBSCAN clustering algorithm was applied with an epsilon of 0.5. As for the lenses, we applied three lenses: the Isolation Forest lens, the Multidimensional Scaling lens, and a projection onto the time column in our dataframe.

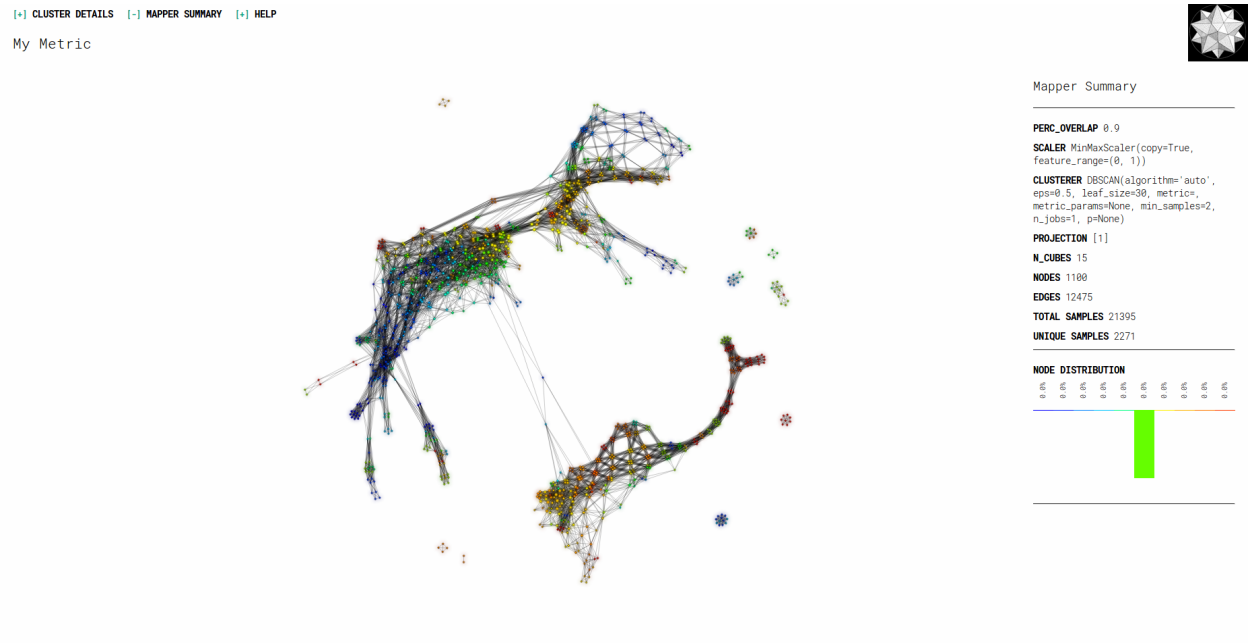



Figure 22: Mapper output projecting onto the number of page numbers using $d_{arXiv}(x,y)$ with an epsilon of 0.5 for larger dataset, colored by academic categories

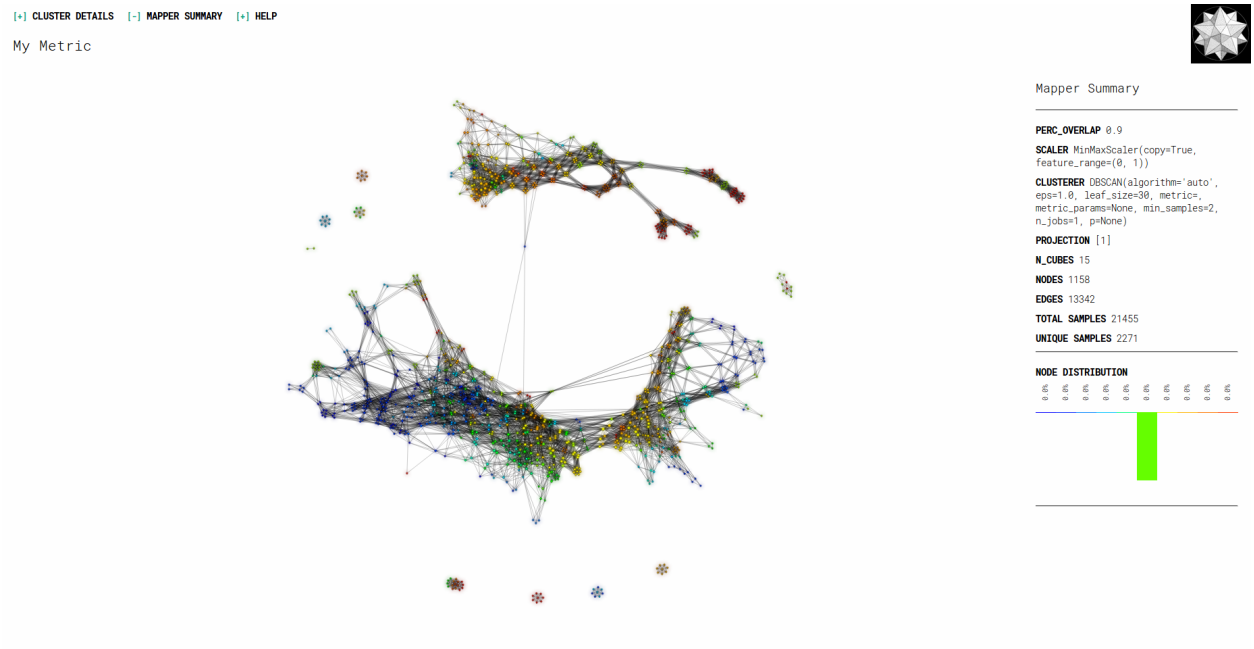



Figure 23: Mapper output projecting onto the number of page numbers using $d_{arXiv}(x, y)$ with an epsilon of 1.0 for larger dataset, colored by academic categories

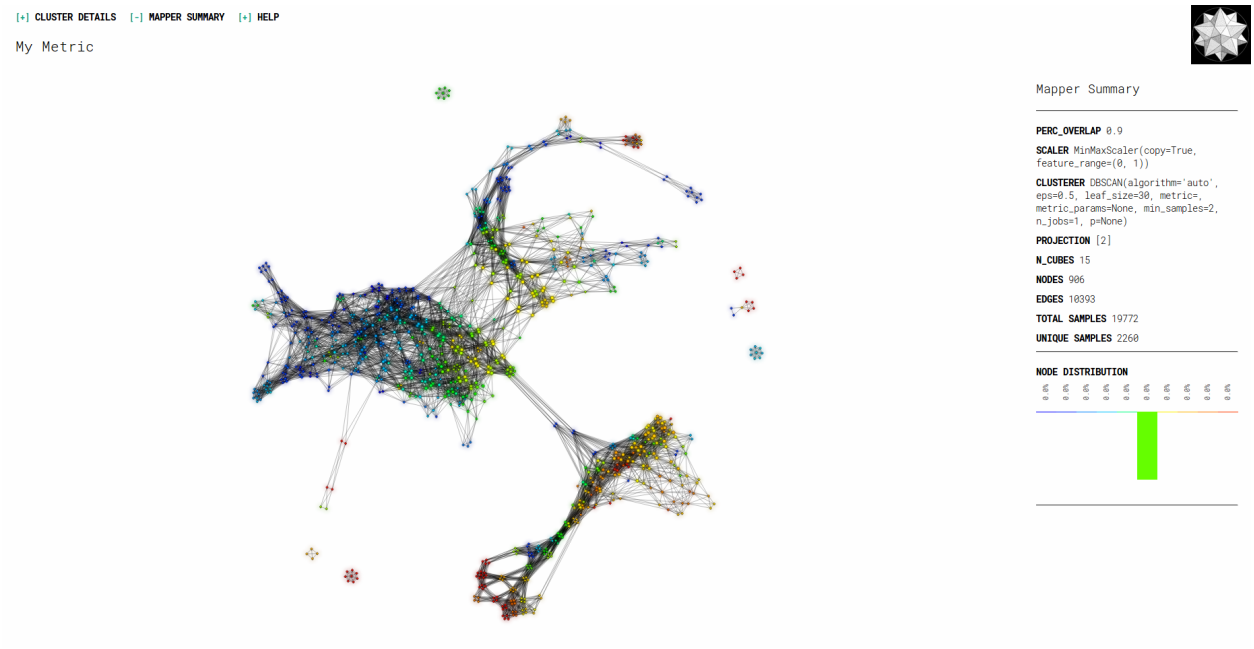



Figure 24: Mapper output projecting onto the number of references using $d_{arXiv}(x, y)$ with an epsilon of 0.5 for larger dataset, colored by academic categories

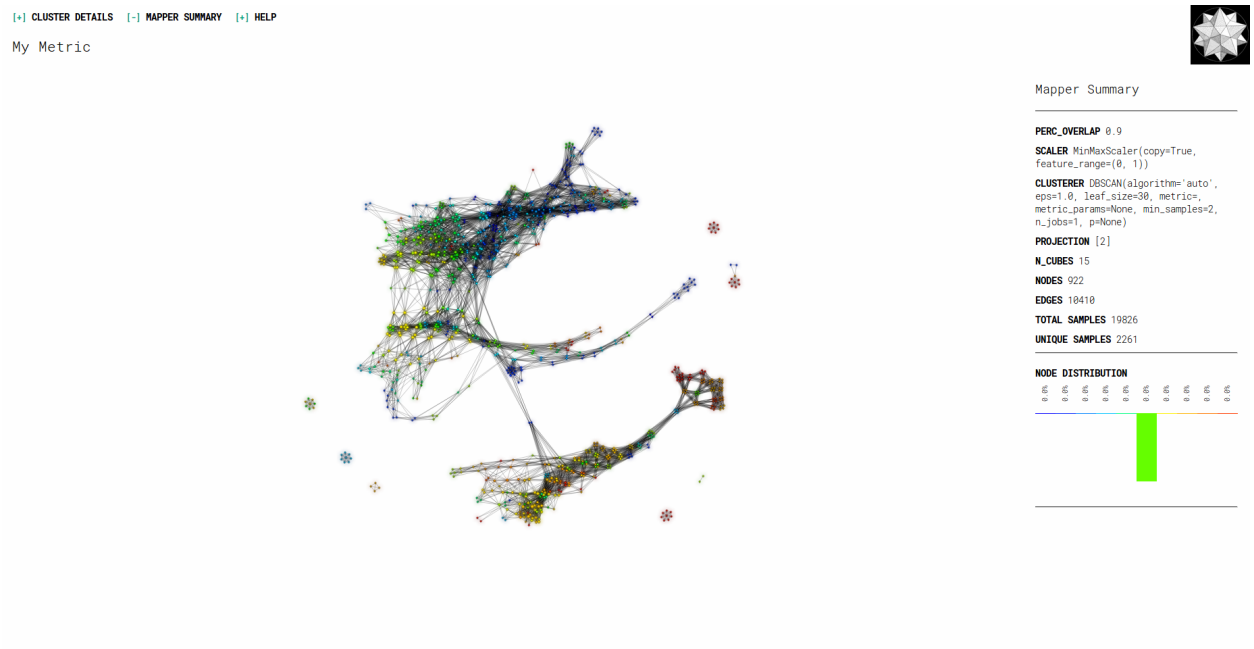



Figure 25: Mapper output projecting onto the number of references using $d_{arXiv}(x, y)$ with an epsilon of 1.0 for larger dataset, colored by academic categories

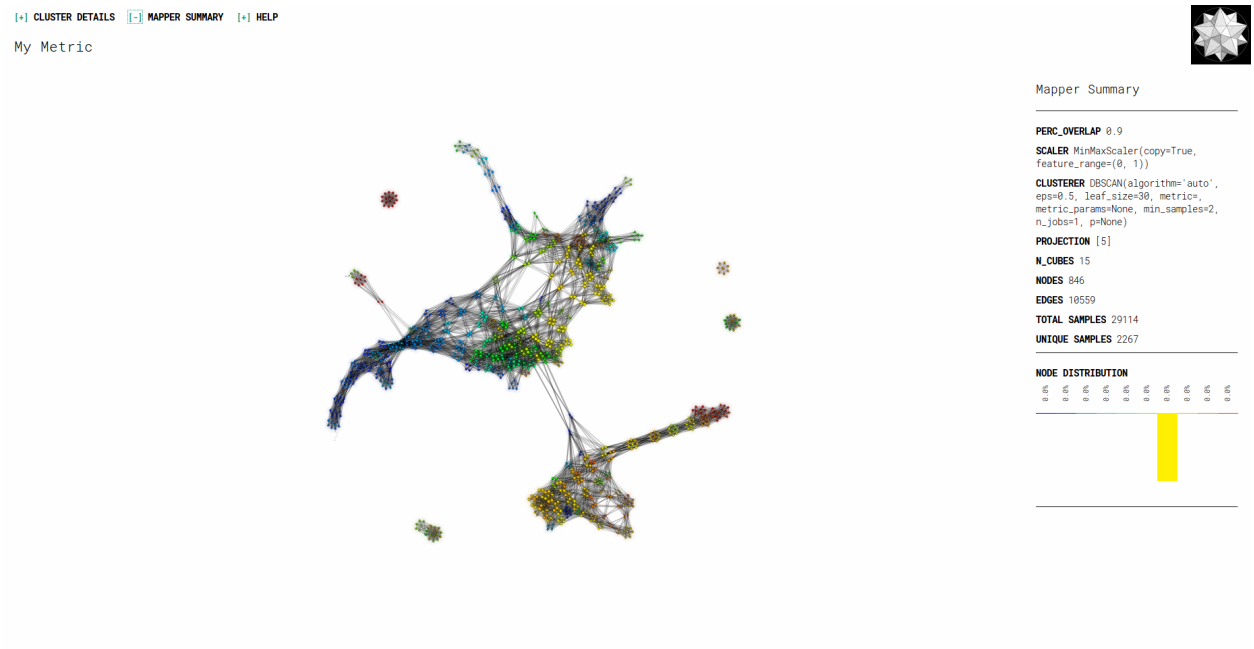



Figure 26: Mapper output projecting onto the time value representing the published date using $d_{arXiv}(x, y)$ with an epsilon of 0.5 for larger dataset, colored by academic categories

#### 4.2.8 Discussion of the 2,285 Papers

From looking at the persistent homology results for the 2,285 papers, we can make the conclusion that our dataset has a shape and that there appears to be at least one loop in this point cloud of papers. From this, we know that there are hidden relationships in our higher dimensional data. From both epsilon values (0.5 and 1.0) we were able to retrieve persistent homology results that indicate some form of relationship as epsilon increased. When we look at Mapper, the visualization also confirms that there is a relationship in this topological space we have created from the 2,285 papers. If we look at the Mapper results, in all outputs, it is interesting to see clustering based on color. Because academic categories was not a feature (column) in our dataframe, its correspondence to the topological space is independent.

In Figure 22, we can see two main clusters connected by a few papers (nodes). In the cluster on the top left, there seems to be a separation of flares by colors. The cluster also seems to be "wanting" to split into two smaller clusters as there seems to be a "pinch" when we near the green/yellow nodes. We can interpret this as a certain divide in forms of papers in the academic fields representing these green/yellow nodes. One could speculate that these fields tend to publish two very drastically different types of papers. It is interesting to note that there are some uniformly colored, smaller clusters surrounding the two big clusters.These small clusters represent papers that are very similar to each other though not similar to the rest of the papers as whole. When we look at the cluster on the bottom right, we can see a big red flare shaped like a "y". We can interpret the large clustering based on color to indicate the following conclusion: papers in similar academic fields tend to produce papers with similar page numbers.

More specifically, papers in the blue, physics and mathematics, tend to follow this trend. The red represents Nuclear Physics and since this is still a fairly new field, fewer papers have been published and so fewer nodes appear. It is interesting to see that these papers are clustered so closely together. The reason we can say that papers in clusters have similar number of pages is because if one recalls, one of our Mapper projections is a lens representing the number of pages per paper. If we look at Figure 23, we see a very similar shape in the visualization. Like in the persistent homology results, increasing the epsilon value from 0.5 to 1.0 did not drastically change the results. The same applies for the Mapper visualization here. We still see two main clusters separated by a one or a few nodes (papers). There is still the same color distinction as the previous Mapper result.

Now, if we look at Figure 24, we are now seeing a projection onto the number of references instead of the number of pages. The epsilon here is 0.5, and the color function is the same (array of academic categories). Interestingly enough, the shape that appears in this output is somewhat similar to the Mapper output that used a projection of the number of pages per paper. We see two main clusters connected by one or a few nodes (papers), and there again seems to be a color distinction. As we have seen in the previous two Mapper outputs, the cluster on the top left seems to have three main colors (blue, green, yellow). As before, this cluster has the appearance of wanting to "split" into two smaller clusters. Also like before, we have red colored flares in the bottom right cluster. From all of this, given the apparant colored clusters and shape, we can conclude the following: papers with in similar academic fields tend to produce papers with

similar amounts of references listed. It is quite interesting to see this trend in both the number of pages per paper and the number of references per paper. One could speculate that the longer the paper is the more references this paper has. This research question, however, would be for the future. Again, as before, increasing the epsilon from 0.5 to 1.0 does not change the Mapper output drastically. In Figure 25 we see a very similar shape as was seen in the previous Mapper output with an epsilon of 0.5. The top left cluster still appears to want to "separate" and there still appears to be a red flare on the bottom right. The two clusters are also connected by a few nodes.

Finally, if we look at Figure 26, we the same similar shape. There appears to be two clusters connected by a few nodes. The top left cluster is clustered by color and appears to want to "separate". The bottom right cluster has a red flare. Because we are dealing published time dates, and we still see this trend of colored clusters, one can make the conclusion that certain papers in certain academic fields tend to publish on similar times together. It is simpler to think of this idea for the Nuclear Physics category since it is not as old as Mathematics and Physics. To see this clustering of red nodes confirms this idea for papers published

# 5 Summary and Conclusions

In this paper, we have explored the topics of Topological Data Analysis and some aiding tools such as the Mapper algorithm. After being acquainted with the concepts from general topology and algebraic topology, we emulated the computational process for Persistent Homology and visual outputs with the datasets provided by the MET and arXiv. Mostly utilizing the Python programming language, we were able to obtain some original interpretations of the structure of the provided data clouds.

In the MET project, Mapper has been shown as an effective tool to guide feature selection. Coloring the Mapper output by the variable that we would like to predict helps identify associated significant features. Persistent homology has been employed to classify two subsets of artworks. Using our results, we may differentiate between samples from these two subsets by looking at their barcode graphs and distribution of features.

In the arXiv project, interesting results have been found using the metric $d_{arXiv}(x, y)$ both in Dionysus and in Mapper. With the 125 TDA papers we found promising results that led to further exploration in the 2,285 papers. After gathering all the data results from Mapper and Dionysus, we've noticed a few trends in the shape of the dataset that we can summarize. We notice from our dataset that papers in physics and math (blue and green) tend to be similar in number of pages and have similar number of references listed. The same goes for the red flares, which indicate the papers published in Nuclear physics. Interpreting these results means that we have found trends in how papers in different fields tend to publish or submit.

# 6 Future Work

Concerning the MET database: Since our data had categorical string values, the next step could be to use word classification techniques and machine learning. There have already been implementations of Mapper with semantic meaning, so it is reasonable to think that more underlying relationships in the dataset can be revealed with a metric that captures more information (see Appendix). We might expect to see more persistent topological features given a more developed metric.

By creating mathematical models representing topological features in our data, such as a loop or void, one may further classify the data to make a comparison to standard statistical models in terms of their accuracy and prediction.

As many of the models currently being employed in machine learning have their theoretical backgrounds in statistics, the application of topological approach also brings a diverse yet unique perspective within the realm of data analysis. A significant advantage of topological approach versus the conventional statistical approach is that the framework of TDA allows more flexibility. For an instance, the topological alternative to data analysis can be regarded as more general procedure in comparison to the statistical methods, as the statistical models are often restricted in their dimensions and the assumptions that must follow in order for the distributions to correctly map the predictions. With all these advantages paired with the unique insight offered by the topological framework, further exploration of such discipline should be a worthwhile research.

Concerning the arXiv dataset: Near the end of the project timeline, we delved into experimenting with the Jaccard metric in analyzing the similarity between author names of papers and the similarity between words in summaries of papers. By adding the similarity between authors of papers and the similarity between words in summaries of papers, one is able to describe a metric that measures how similar two papers are in terms of their content. A future exploration in applying persistent homology and understanding its possible Mapper output would be another promising research project.

# 7 Acknowledgements

We would like to thank Professor Ortiz for his mentorship and the opportunity to work on this project. Thank you for all the detailed feedback and support. We would also like to thank the authors of Kepler Mapper [21], Javaplex [40], and Dionysus [30] as well the authors of the Python packages Pandas [2] , sklearn [34], scipy [22], and statsmodels [23]. Finally, we would like to thank Grinnell College for funding this MAP.

## 7.1 Appendix

### 7.1.1 Metric

1. Our initial metric for categorical variables is Levenshtein distance[27], since at that stage, we try to measure the distance between two string values without parsing them. However, since this distance measures the minimum edits needed for transforming a string to the other string, we think it is not an accurate metric for our categorical variables. Then we decide to extract meaningful words from strings and count the number of common words in two strings. To limit the distance between 0-1, we divide it by the length of the longer string. However, this metric does not satisfy the triangle inequality.

2. Around Week 7, we find that our metric does not satisfy the “only if” part of Requirement 1 for a metric (see Definition 6)., since two different artworks may have the same values for all the other features. Since "Object ID" is unique for each artwork, it can be used to differentiate between two artworks. Then we decide to include "Object ID" in our metric.

3. We think that there is possible future work with nltk, a metric that can explore the semantic relationship between words in categorical variables. We start defining a metric based on the path-similarity between words. There are also some challenges. Firstly, some specialized words are not in the given corpus. For these we have to decide whether to drop these values or to group them in a larger category (for example, the Medium 'silverpoint' would be grouped under 'metalpoint'). Another challenge is automatically selecting which meaning of each word to use. However, we feel this direction has potential because it should give more nuanced distance between points.

### 7.1.2 Mapper

1. We have tried different filter functions before we decide to use MDS. Initially, we try to project to columns of the dataset. See Figure 27 for an example.

2. Since our data is mixed, we have wanted to find clustering algorithm that would handle both categorical and numerical data. We use both K-prototypes and K-modes. Figure 27 shows output from Mapper using K-modes clustering algorithm [11]. K-modes is technically meant to be used only with categorical data, we included numerical data. Two disadvantages emerge with K-family clustering algorithms. One is computational time and the other is pre-initalizing clusters.

### 7.1.3 Persistent Homology

1. We observe that “Artist Begin Date” and “Artist End Date” are the two features with the most missing values among all the selected features. Thus, we decide to compare the Mapper dataset and the contrast dataset.

2. Initially, we generate barcode graphs using live.ripser.org[4]. To extract more information from the barcodes, we explore Dionysus, but we could not figure out the ReducedMatrix Class (probably due to our inexperience in Python). Finally we switch to javaPlex, and are able to extract elements that form the homology generators indicated by the barcodes.

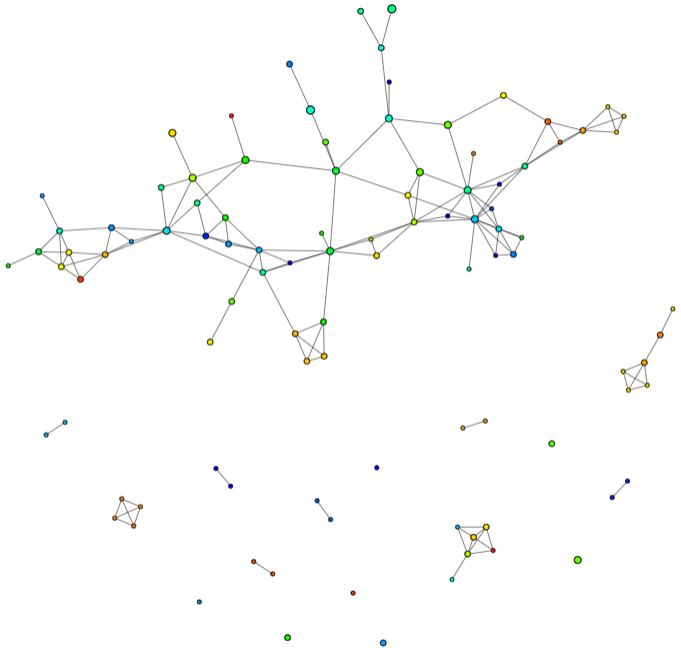


Figure 27: Initial exploration with kmodes clustering algorithm, projected onto Width and Length columns, colored by row index in dataset

# References


[1] The metropolitan museum of art's open access initiative. `https://github.com/metmuseum/openaccess`. Accessed: 6/25/2018.

[2] Pandas: Flexible and powerful data analysis / manipulation library for Python, providing labeled data structures similar to r data.frame objects, statistical functions, and much more. [Online; accessed <today>].

[3] arXiv-api group. arxiv api. [Online; accessed <today>].

[4] Ulrich Bauer. Ripser:a lean c++ code for the computation of vietorisâĂŞrips persistence barcodes, 2016. [Online; accessed <today>].

[5] Paul Bendich, J. S. Marron, Ezra Miller, Alex Pieloch, and Sean Skwerer. Persistent homology analysis of brain artery trees. *The Annals of Applied Statistics*, 10(1):198âĂŞ218, 2016.

[6] I. Borg and P.J.F. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer, 2005.

[7] L. Carlsson, G. Carlsson, and M. Vejdemo-Johansson. Fibres of Failure: Classifying errors in predictive processes. *ArXiv e-prints*, February 2018.

[8] C.J. Carstens and K.J Horadam. Persistent Homology of Collaboration Networks. *Mathematical Problems in Engineering*, 2013(815035):2–3, April 2013.

[9] C. Curto. What can topology tell us about the neural code? *ArXiv e-prints*, May 2016.

[10] Pablo G. CÃąmara, Arnold J. Levine, and RaÃžl RabadÃąn. Inference of ancestral recombination graphs through topological data analysis. *PLOS Computational Biology*, 12(8), 2016.

[11] Nico de Vos. Kmodes: Python implementations of the k-modes and k-prototypes clustering algorithms., 2016–. [Online; accessed <today>].

[12] Edelsbrunner, Letscher, and Zomorodian. Topological persistence and simplification. *Discrete & Computational Geometry*, 28(4):511–533, Nov 2002.

[13] Edelsbrunner, Letscher, and Zomorodian. Topological persistence and simplification. *Discrete & Computational Geometry*, 28(4):511–533, Nov 2002.

[14] Herbert Edelsbrunner and John Harer. *Computational topology: an introduction*. American Mathematical Soc., 2010.

[15] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pages 226–231. AAAI Press, 1996.

[16] Robert Ghrist. Barcodes: The persistent topology of data. *Bulletin of the American Mathematical Society*, 45(01):61âĂŞ76, 2007.

[17] Wei Guo and Ashis G. Banerjee. Identification of key features using topological data analysis for accurate prediction of manufacturing system outputs. *Journal of Manufacturing Systems*, 43:225 – 234, 2017. High Performance Computing and Data Analytics for Cyber Manufacturing.

[18] Allen Hatcher. *Algebraic Topology*. Cambridge University Press, 2002.

[19] Zhexue Huang. A fast clustering algorithm to cluster very large categorical data sets in data mining. 09 1998.

[20] P. Jaccard. Distribution de la flore alpine dans le bassin des dranses et dans quelques rÃľgions voisines. *Bulletin de la SociÃľtÃľ Vaudoise des Sciences Naturelles*, 37:241–272, 1901.

[21] Hendrik Jacob van Veen and Nathaniel Saul. Keplermapper. http://doi.org/10.5281/zenodo.1054444, Nov 2017.

[22] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. [Online; accessed <today>].

[23] Jonathan Taylor Josef Perktold, Skipper Seabold. statsmodels: statsmodels is a python module that provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration. [Online; accessed <today>].

[24] Rami Kraft. Illustrations of Data Analysis Using the Mapper Algorithm and Persistent Homology. 2016.

[25] Vitaliy Kurlin. A one-dimensional homologically persistent skeleton of an unstructured point cloud in any metric space. *Computer Graphics Forum*, 34(5):253–262.

[26] Sunghyon Kyeong, Jae-Jin Kim, and Eunjoo Kim. Novel subgroups of attention-deficit/hyperactivity disorder identified by topological data analysis and their functional network modular organizations. *Plos One*, 12(8), 2017.

[27] V. I. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707, February 1966.

[28] Li Li, Wei-Yi Cheng, Benjamin S. Glicksberg, Omri Gottesman, Ronald Tamler, Rong Chen, Erwin P. Bottinger, and Joel T. Dudley. Identification of type 2 diabetes subgroups through topological analysis of patient similarity. *Science Translational Medicine*, 7(311), 2015.

[29] P. Y. Lum, G. Singh, A. Lehman, T. Ishkanov, M. Alagappan, J. Carlsson, G. Carlsson, and Mikael Vilhelm Vejdemo Johansson. Extracting insights from the shape of complex data using topology. *Scientific Reports*, 3, 2 2013.

[30] Dmitriy Morozov. Dionysus: Dionysus is a c++ library for computing persistent homology. [Online; accessed <today>].

[31] James R Munkres. *Topology*. Prentice Hall, 2000.

[32] Monica Nicolau, Arnold J. Levine, and Gunnar Carlsson. 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):7265–7270, 2011.

[33] Jessica L. Nielson, Jesse Paquette, Aiwen W. Liu, Cristian F. Guandique, C. Amy Tovar, Tomoo Inoue, Karen-Amanda Irvine, John C. Gensel, Jennifer Kloke, Tanya C. Petrossian, and et al. Topological data analysis for discovery in preclinical spinal cord injury and traumatic brain injury. *Nature Communications*, 6(1), 2015.

[34] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[35] PennState. Comparing two proportions. [Online; accessed <today>].

[36] G. Petri, M. Scolamiero, I. Donato, and F. Vaccarino. Topological Strata of Weighted Complex Networks. *PLoS ONE*, 8:e66506, June 2013.

[37] Abbas H Rizvi, Pablo G Camara, Elena K Kandror, Thomas J Roberts, Ira Schieren, Tom Maniatis, and Raul Rabadan. Single-cell topological rna-seq analysis reveals insights into cellular differentiation and development. *Nature Biotechnology*, 35(6):551âĂŞ560, Jan 2017.

[38] M. Rucco, L. Falsetti, D. Herman, T. Petrossian, E. Merelli, C. Nitti, and A. Salvi. Using Topological Data Analysis for diagnosis pulmonary embolism. *ArXiv e-prints*, September 2014.

[39] G. Singh, F. Mémoli, and G. Carlsson. Topological methods for the analysis of hbigh dimensional data sets and 3d object recognition. *Eurographics Symposium on Point-Based Graphics*, 2007.

[40] Andrew Tausz, Mikael Vejdemo-Johansson, and Henry Adams. JavaPlex: A research software package for persistent (co)homology. In Han Hong and Chee Yap, editors, *Proceedings of ICMS 2014*, Lecture Notes in Computer Science 8592, pages 129–136, 2014. Software available at `http://appliedtopology.github.io/javaplex/`.

[41] Machacon Herchel Thaddeus. A topological data analysis approach to visualizing ebola tweets. *Japan Journal of Medical Informatics*, 36(5):253–269, 2016.

[42] Yuan Yao, Jian Sun, Xuhui Huang, Gregory R. Bowman, Gurjeet Singh, Michael Lesnick, Leonidas J. Guibas, Vijay S. Pande, and Gunnar Carlsson. Topological methods for exploring low-density states in biomolecular folding pathways. *The Journal of Chemical Physics*, 130(14):144115, 2009.

[43] Xiaojin Zhu. Persistent homology: An introduction and a new text representation for natural language processing. pages 1953–1959, 08 2013.

[44] Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33(2):249–274, Feb 2005.