A New Comparison Metric for Computational Topology

Pejmon Shariati

An Honors Thesis for the Department of Mathematics

Tufts University, 2021

Contents

1 Acknowledgements

I would like to thank my thesis advisors Professor Moon Duchin and Professor Thomas Weighill for supporting me this past year and guiding me in my research. I would also like to thank them and Professor Kasso Okoudjou for agreeing to be part of my thesis committee.

2 Introduction

There is often a common yet important set of questions that a data scientist, a mathematician, or even an engineer must ask themselves when dealing with any kind of data. Some of which include - what should I do with this data? How should I visualize it? What can I do to gain the most descriptive information from it? Broadly speaking, you could possibly use machine learning, statistics, or modeling to better understand your data. Specifically within machine learning there are some related methods of interest such as PCA (principal component analysis), SVM (support vector machines), neural networks, and TDA (topological data analysis). For the purposes of this paper we focus on TDA - the study of the "shape of data." Persistent homology is a popular technique whose ultimate goal is to track the appearance and disappearance of homology classes (i.e. holes and connected components). See for example, [13]. However, our attention is directed towards an algorithm called "Mapper." It was developed by Gurjeet Singh, Facundo Mémoli, and Gunnar Carlsson for topological data analysis projects. It is used to extract simple descriptions of high dimensional data sets [10]. In other words, it is an efficient method for summarizing data. My goal is to extend the use of Mapper from summarizing individual data sets to comparing different data sets with each other. We will devise a method to accurately measure the difference between Mapper objects.

The method will be based on converting Mapper objects into metric measure spaces and then using their weighted path induced distance matrices and the Gromov-Wasserstein metric to compare them. We will then use this method to perform experiments on synthetic data in the form of shapes that are familiar to us - circles, lines, etc. Once we are confident in the robustness and stability of our method, we will apply it to an example of real world data to see if we can gain any descriptive information. From here, we theorize the possible future benefits of our method on other real world data.

3 Background

Definition 3.1 (Mapper)**.** [1] Given a data set X we choose a real valued function $f : X \to \mathbb{R}$ known as a filter. From here we choose a covering or a collection of intervals $\{I_i\}_{i=1}^{t}$ whose union is a superset of $f(X)$. In addition, we need to decide the length of these intervals and how much they overlap with each other. Next, for each i we must consider $f^{-1}(I_i)$ and apply a clustering algorithm of our choice to this set. This will give us a cluster which we will identify as a node. If two clusters, $c_i \subset f^{-1}(I_i)$, $c_j \subset f^{-1}(I_j)$ $(i \neq j)$ have a nonempty intersection, so $c_i \cap c_j \neq \emptyset$, then the Mapper algorithm assigns an edge between those two nodes. At the end of this process we end up with a Mapper object.

Now that we have given a formal definition of the Mapper algorithm, we would like to discuss some of its applications. Using data from a 1970's diabetes study at Stanford University, Singh et al used Mapper to determine the division of diabetes into adult onset and juvenile onset forms [10]. In another study involving cancer research, Monica Nicolau, Arnold J. Levine, and Gunnar Carlsson used Mapper to identity a subgroup of breast cancers. This study even mentions how Mapper has already been used to gain further insight into the subtleties of the folding patterns of RNA [7]. Already we can see some of the medical benefits of an algorithm such as Mapper. Another application includes a method implemented by Kleiman and Ovsjanikov who sought to present a robust method to find region-level correspondences between shapes, which are invariant to changes in geometry and applicable across multiple shape representations [5]. In other words, with the aid of Mapper they could identify a shape such as a human hand regardless of its position and configuration in space.

These are just a few examples that illustrate the importance and versatility of an algorithm such as Mapper. Suppose we wanted to use Mapper on the following circle in figure 1. We also want to use the height or the y-axis as the filter function.

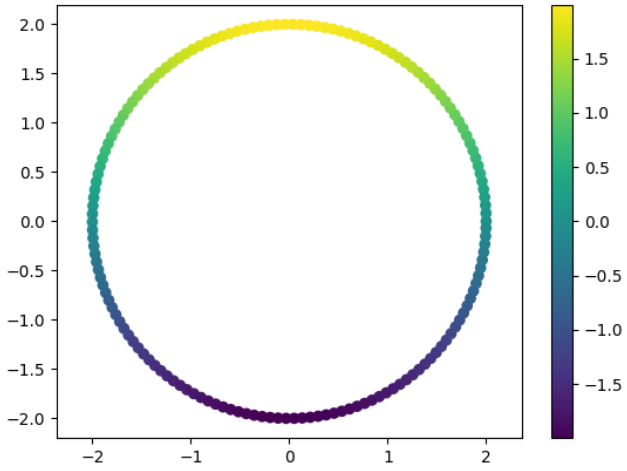

Figure 1: This is the 2D circular data set where each data point is colored by its filter value which is its y-coordinate.

For our parameters, our cover comprised of six intervals with a percent overlap of 30 %. When we run Mapper on this shape we end up with the following output in figure 2:

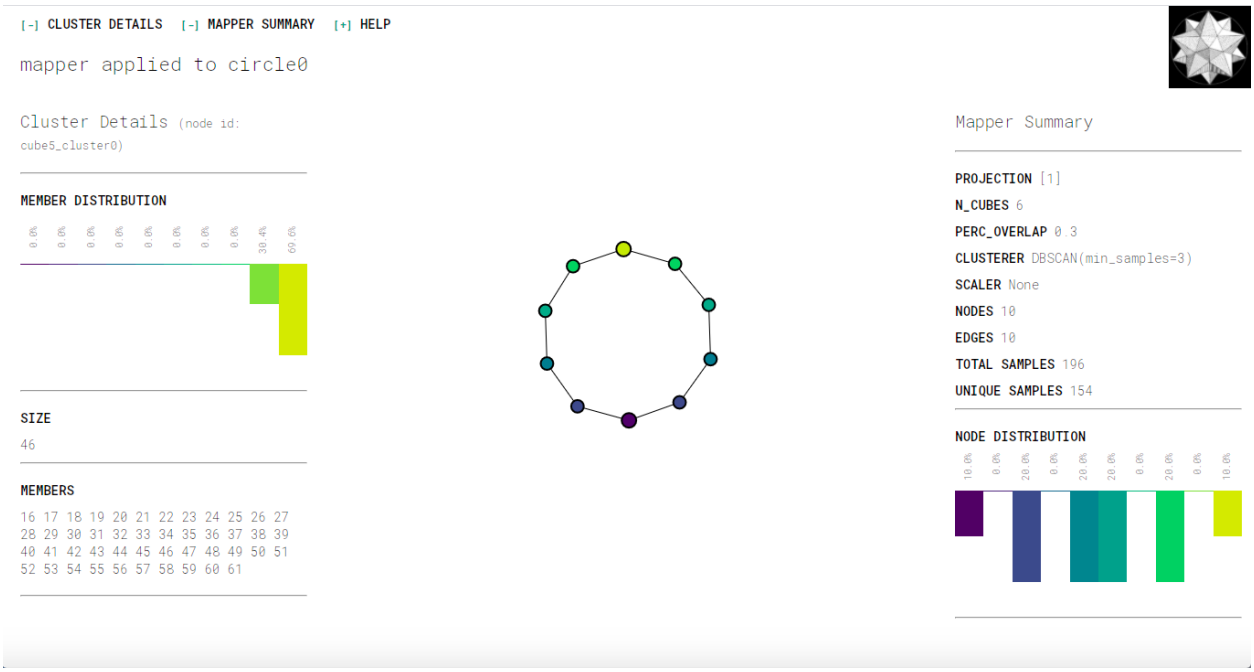

Figure 2: In this output we can hover over nodes and see which of the data points they contain as well as the size of the node itself. On the right hand side, you can see the clustering algorithm used on the data, the number of intervals in the 1-dimensional cover, and the percent overlap.

Here we specifically chose to color the nodes by filter value. We used the same viridis color map to illustrate the height of each point in the original data set. The nodes are colored by average filter value. Mapper is intended to take a point cloud of data and guess its shape in the form of a graph. Therefore, figure 2 is exactly what we would anticipate a successful application on circular point cloud data to be. Mapper has simplified the data set by outputting a graph with less nodes while still maintaining the shape of a circle. This gives an idea of how useful it can be when dealing with real world data whose structure may be difficult to discern. More examples can be found in section 5.1.

4 Related Work

There are papers related to our work that are worth mentioning. We have already discussed the study by Kleiman and Ovsjanikov which was primarily concerned with shape matching. See [5] for further details.

In another work by Michael McCabe, he uses an optimal transport based metric called the Network Augmented Wasserstein (NAW) distance to compare two different Mapper outputs [6]. In his paper, he uses this metric to evaluate distances between Mapper graphs and demonstrates its value for model drift analysis by using it to transform the model drift problem into an anomaly detection problem over dynamic graphs [6]. Similar to our construction, he does transform each Mapper output into a metric measure space and gives

each space a measure that will be defined in the same way as ours as we will see in the next section. However, he uses a different metric known as the Hausdorff distance between the nodes within a particular Mapper output. He then implements an extrinsic metric in his definition of NAW. One of the key differences in our methods is that in our implementation we set the distance between any pair of nodes in different connected components of a Mapper output to be the maximum distance between any pair of nodes within any one connected component. McCabe takes a different approach [6]. Furthermore, his metric also takes edge importance into account in terms of connected components and the density of the neighboring nodes [6]. This is not really a concern of ours because as we will see our metric is more concerned with shape preservation.

In another work by Francisco Belchí and his collaborators, they use a so called "clustering" distance to determine the instability of clustering algorithms used in the Mapper procedure [1]. In a similar vein much like us, they perform several experiments to measure the instability of any Mapper-type algorithm and propose techniques to control it. They try to view instability as a function of the Mapper input parameters and determine a good Mapper output as a local minimum [1]. This would give someone an idea of an appropriate set or range of parameters that would yield a good Mapper output. They do acknowledge however that since all clustering algorithms present some kind of instability, Mapper will inevitably bear similar difficulties. They also took into account the size of each cluster unlike persistent homology which is only concerned with their presence. This led them to study the ramifications of choosing a certain clustering algorithm which can be different on different parts of the cover. Eventually, they discuss that a clustering quality function is required to measure the efficacy of a particular clustering by assigning it a notional cost [1]. From here it is rather intuitive then that the goal is to find a clustering procedure that minimizes such a cost. They compare clusterings using a metric called the minimal matching distance. By assigning a distance between two clusterings and utilizing the quality function mentioned earlier, they can then use these tools to define instability. Eventually, they formalize a function that provides a numerical output which indicates the instability [1]. Their next step involved taking these ideas of comparing clusterings and applying them to something they call Mapper functions - their new way of describing Mapper outputs [1]. This is comparable to how they represented clusterings as functions. Part of their construction attempts to extrapolate the minimal matching metric to comparing Mapper outputs. The rationale behind their method is based on the information of the clusters within the resulting simplicial complexes after running Mapper. The instability of a Mapper procedure and the derivation of its upper bounds is then based on their concept of distance between these Mapper functions. While Belchí and his collaborators research the importance of clustering algorithms within the Mapper procedure in relation to instability, we take a different approach. Every experiment in this thesis is used with the clustering algorithm DBSCAN, and we do not discuss any effects of clustering. Furthermore, unlike our method, Belchí's is not concerned with the shape of the Mapper graphs. One of the other key differences in our methods is that Belchí and his collaborators compare Mapper graphs on the same data, while we compare the Mapper graphs of different data sets. In addition, our focus is on other parameter choices such as the number of intervals in the cover, the amount of noise added to our original data, and the overall shape of our Mapper outputs. We determine instability with regard to these areas.

5 Method

Definition 5.1 (Metric Space)**.** [2] A set X is called a metric space if for any two points p and q in X there is defined a real number $d(p,q)$, called the distance between p and q such that the following three properties are satisfied:

1. (Non-negativity) $d(p,q) > 0$ if $p \neq q$ and $d(p,q) = 0$ if $p = q$
2. (Symmetry) $d(p,q) = d(q,p)$
3. (Triangle Inequality) $d(p,q) \leq d(p,w) + d(w,q)$ for all $w \in X$

The above definition $d : X \times X \to [0,\infty)$ is called a metric on X.

Definition 5.2 (Probability Measure)**.** [9] We will assume for this thesis that our spaces are discrete. A function $\mu : P(X) \to \mathbb{R}$, where $P(x)$ is the power set of X, is said to be a probability measure on X if

1. (Non-negativity) $\mu(A) \geq 0$ for every $A \in P(X)$.
2. $\mu(X) = 1$
3. (Additivity) $\mu(A \cup B) = \mu(A) + \mu(B)$ if $A \cap B = \emptyset$
4. $\mu(\emptyset) = 0$

Definition 5.3 (Metric Measure Space)**.** [4] A metric measure (mm) space is defined to be a triple (X, d, μ) where (X, d) is a separable metric space and μ is a measure on X.

Definition 5.4 (Gromov-Wasserstein)**.** [6] The Gromov-Wasserstein metric is defined between mm-spaces. It measures how far two mm-spaces are from isomorphism. Assume we have two finite mm-spaces (G_X, d_X, μ_X) and (G_Y, d_Y, μ_Y). Consider the matrices of pairwise distances for each space, $M^X \in \mathbb{R}^{n\times n}$ and $M^Y \in \mathbb{R}^{m\times m}$. We define $\Omega(\mu_X, \mu_Y)$ to be the set of valid transportation plans or

$$\Omega(\mu_X, \mu_Y) := \{\mu \in \mathbb{R}^{n\times m} \mid \mu\mathbf{1} = \mu_Y, \mu^T\mathbf{1} = \mu_X\}$$

So the discrete Gromov Wasserstein metric is defined to be [8] [3]:

$$\min_{\mu} \left\{ \sum_{i,j,k,l} \frac{1}{2}|M^X_{i,k} - M^Y_{j,l}|^2 \ \mu_{i,j}\, \mu_{k,l} : \mu \in \Omega(\mu_X, \mu_Y) \right\}$$

In our context, our mm-space will come from the Mapper graph with the weighted path induced distance matrix. Suppose we have a data set X. We first run Mapper to obtain the Mapper object and its graph G_X. Consider the set of nodes Π in the graph G_X. For each $n \in \Pi$, its weight $N(n)$ will be equal to the number of data points it contains. Therefore, we can now define a measure on the nodes of G_X:

$$\mu_X(n_0) = \frac{N(n_0)}{\sum N(n)}$$

where $N(n_0) =$ the weight of node n_0. Based on this formulation it is now apparent that we are simply normalizing the node weights. We will also define a function F on each node:

$$F(n) = \text{average of the filter values in node } n$$

Therefore we can define the edge weight:

$$\text{edge weight of } (n, m) = |F(n) - F(m)|$$

After determining the weight of each edge, we use Dijkstra's algorithm to determine the lightest path between each pair of nodes. We define our metric d_X on G_X to be the weight of the lightest path between any pair of nodes. If two nodes are in different connected components of the Mapper output we define their distance to be equal to the maximum distance between any pair of nodes within any one connected component. We store these distances in a matrix $M^X \in \mathbb{R}^{n \times n}$ called the weighted path induced distance matrix. Therefore, if indices i and j of M^X correspond to the nodes n and m then

$$M^X_{i,j} = d_X(n, m)$$

We constructed the weighted path induced distance matrix in this manner because we want our distance metric to be stable under minor or small refinements in the cover. In other words, we only care about changes in the shape of the Mapper output, not the number of nodes in the Mapper output. We could weigh each edge equally but then subdividing edges would not have a small effect on the mm-space.

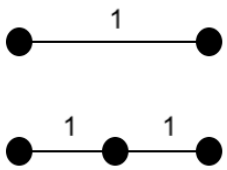

Figure 3: This is an example of what we mean by subdividing edges. In the first line its two end points are distance 1 apart but the second line's endpoints are distance 2 apart. The distance has doubled but notice the shape is still the same. Our method should therefore indicate a low discrepancy between the two since their shapes are the same. Furthermore, if we were to subdivide the first line into two edges then our method would not give each edge a weight of 1.

We have now just described an abstract process for converting a Mapper object into an mm-space (G_X, d_X, μ_X).

Summary: To recap, we start off with two data sets X and Y. Once we run the Mapper algorithm on each set we end up with two Mapper objects and their graphs G_X and G_Y. We can then convert the Mapper objects into mm-spaces and use the Gromov-Wasserstein metric to measure how far they are from isomorphism. Now that we have devised a method for comparing the Mapper objects of two different data sets, we will show the results of certain experiments in the next chapter.

6 Experiments

For these experiments, we use an implementation of Mapper called "Kepler Mapper," [11] [12] the Python Optimal Transport (OT) Library [3], and the Gromov-Wasserstein metric [8]. We first used this method of comparing Mapper objects on shape data. By shape data, we mean two dimensional data points in the shape of circles, lines, figure eights, and the letter Y. For every experiment in this section we chose the height of the data or its projection onto the y-axis as the filter function for the Mapper algorithm. We also resorted to the default Kepler Mapper coloring scheme for the nodes. Most of our experiments with the exception of the real world data example are designed to have a right answer. In other words, we know what the Mapper objects should appear to be in every case because they are obtained from an original set of data similar in shape with added noise. Therefore, we can tell when a certain Mapper object may have a bad set of parameters - when it does not resemble the shape of the original point cloud data. After we establish that the experiments with the shape data were successful, we can then progress to applying Mapper to real world data where its shape is unknown beforehand. See `https://github.com/pshariati/thesis.git` for our code.

6.1 "4-3-2-1" Experiment

For our first experiment, we ran our method on four circles, three lines, two Y-s, and a singular figure eight. For each shape, we applied normally distributed noise with a standard deviation of 0.1. We show scatter plots for all ten cases of the shape data:

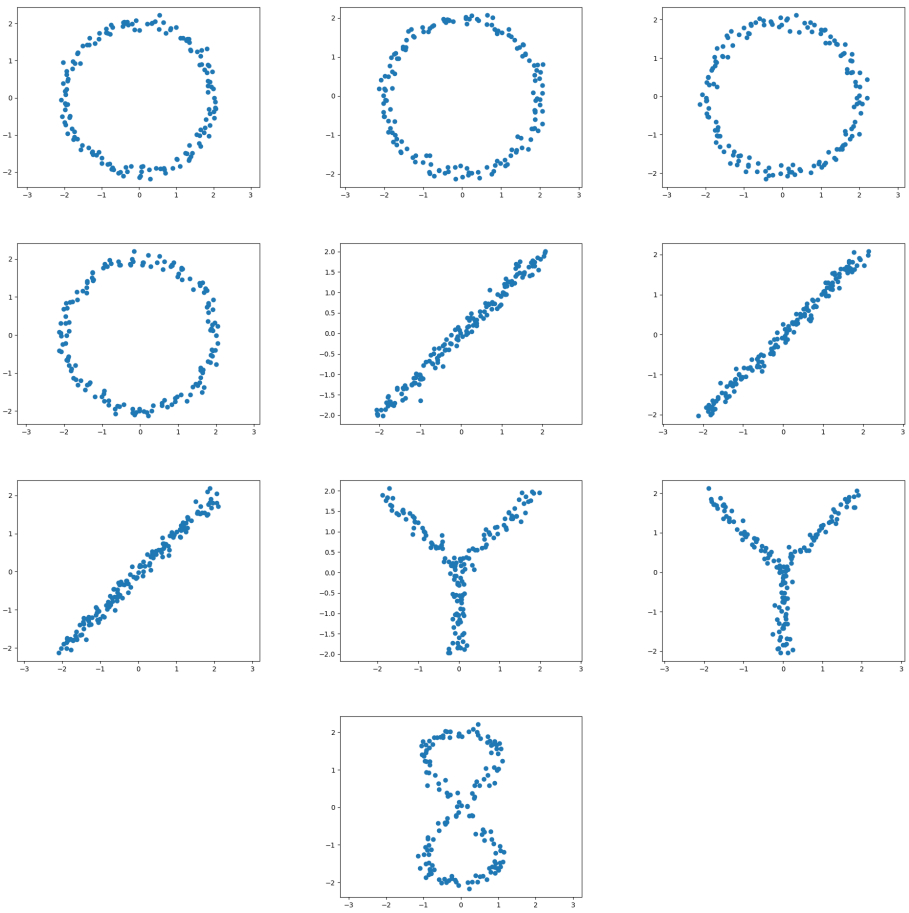

Figure 4: Scatter plots of the ten cases of shape data.

Each cover comprised of six intervals and the percent overlap for each cover was 30 %. The following figure is the Gromov-Wasserstein distance matrix output of this experiment:

$$\begin{bmatrix} 0.00 & 0.25 & 0.26 & 0.05 & 0.94 & 0.93 & 0.92 & 1.30 & 1.21 & 0.49 \\ 0.25 & 0.00 & 0.06 & 0.05 & 0.84 & 0.86 & 0.83 & 1.38 & 1.41 & 0.54 \\ 0.26 & 0.06 & 0.00 & 0.27 & 0.81 & 0.81 & 0.81 & 1.57 & 1.17 & 0.47 \\ 0.05 & 0.05 & 0.24 & 0.00 & 1.54 & 0.86 & 0.84 & 1.16 & 1.24 & 0.58 \\ 0.94 & 0.84 & 0.81 & 1.54 & 0.00 & 0.03 & 0.04 & 0.56 & 0.37 & 0.28 \\ 0.93 & 0.85 & 0.87 & 0.86 & 0.03 & 0.00 & 0.05 & 0.44 & 0.37 & 0.31 \\ 0.92 & 0.83 & 0.83 & 0.84 & 0.04 & 0.05 & 0.00 & 0.40 & 0.37 & 0.33 \\ 1.30 & 1.38 & 1.46 & 1.16 & 0.56 & 0.44 & 0.40 & 0.00 & 0.08 & 1.21 \\ 1.21 & 1.41 & 1.26 & 1.24 & 0.37 & 0.37 & 0.37 & 0.08 & 0.00 & 1.26 \\ 0.49 & 0.54 & 0.47 & 0.58 & 0.28 & 0.31 & 0.33 & 1.21 & 1.26 & 0.00 \end{bmatrix}$$

Figure 5: GW distance matrix

The ten rows and ten columns of this matrix correspond to the Mapper graphs of our

original shape data. We can ask for a heat map of the matrix above along with a visual of each shape's Mapper output to obtain a better description of what we are seeing:

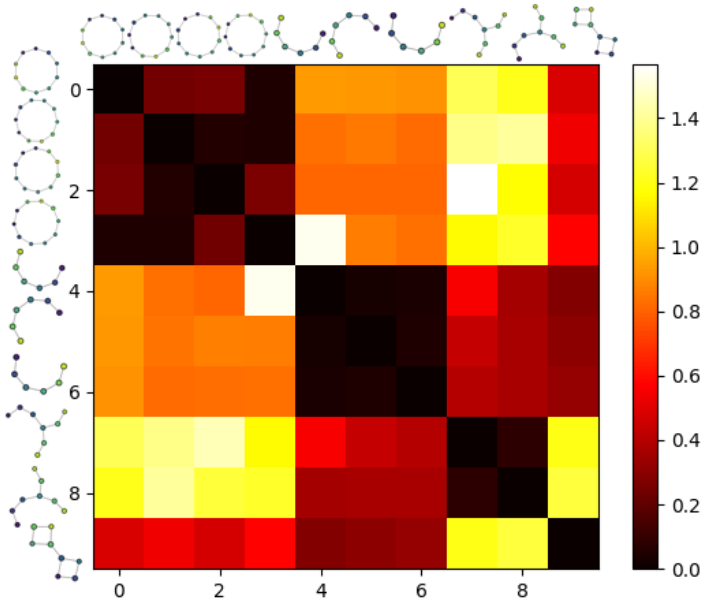

Figure 6: This is the corresponding heat map of the GW distance matrix output of the "4-3-2-1" Experiment. Notice how the distances between similar shapes are smaller than between different shapes.

We would expect to see darker squares (small distance) between each pair of similar shapes which agrees with our heat map. This experiment demonstrates how our method can distinguish shape types.

Remark 6.1. Note that the heat map, or really the distance matrix, is not symmetric because even though the theoretical Gromov-Wasserstein metric is symmetric the computational algorithm used only approximates it. It is also evident where the triangle inequality fails in the Gromov-Wasserstein distance matrix output M represented by the heat map. Observe that the $(1,0)$-entry of M is 0.25, yet the sum of the $(1,3)$ and $(3,0)$ entries is only 0.1. This is a violation of the triangle inequality since $M[1,0] > M[1,3] + M[3,0]$. This is important because one of the requirements of a metric measure space is for it to obey the triangle inequality as we pointed out in Definitions 4.1 and 4.3. The fact that it fails here highlights the computational pitfalls of the Gromov-Wasserstein metric.

6.2 "Ten Noisy Circles" Experiment

The next experiment involved running our method on ten circles. We increased the amount of noise in each circle until the Mapper output of the last few were unrecognizable or completely destroyed. To be specific, suppose we have a variable called q_i and $q_0 = 0$. For circle

$i \in \{1, 2, ..., 9\}$ we applied normally distributed noise with a standard deviation of $q_i = q_{i-1} + 0.35/10$. We show scatter plots for the ten cases of circular data:

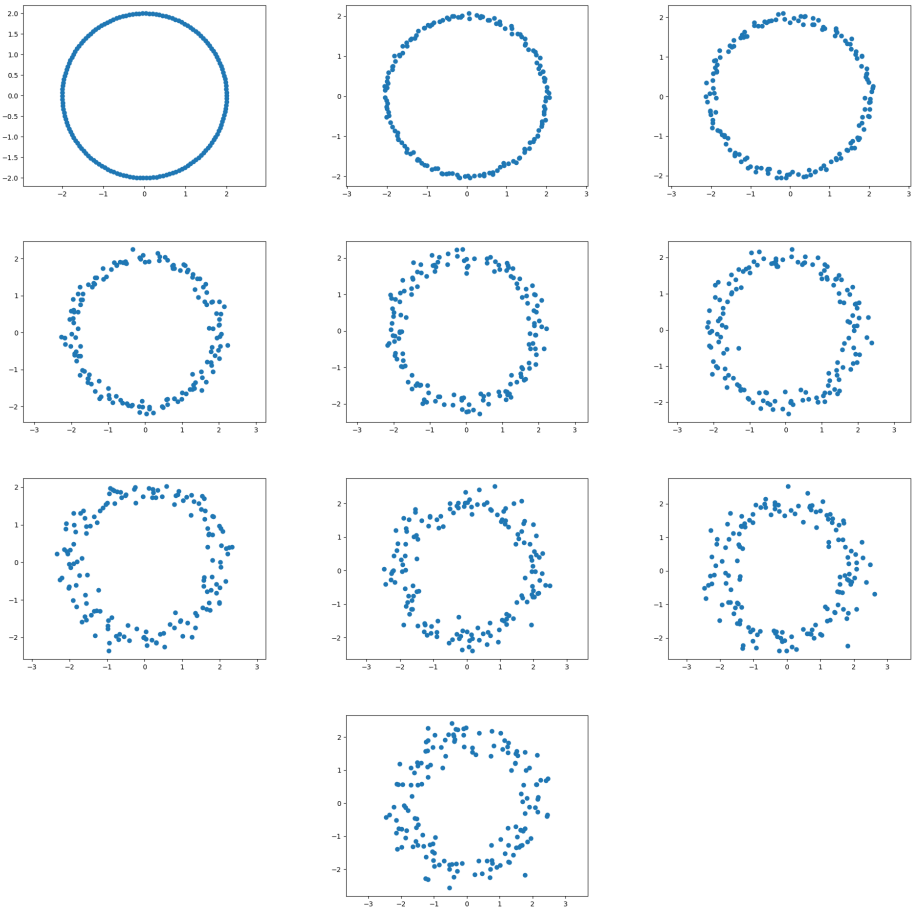

Figure 7: Scatter plots of the ten cases of circular data with an increasing amount of noise.

We kept the number of intervals in each cover at six with a percent overlap of 30 % for all ten circles. The heat map below is the result of the experiment:

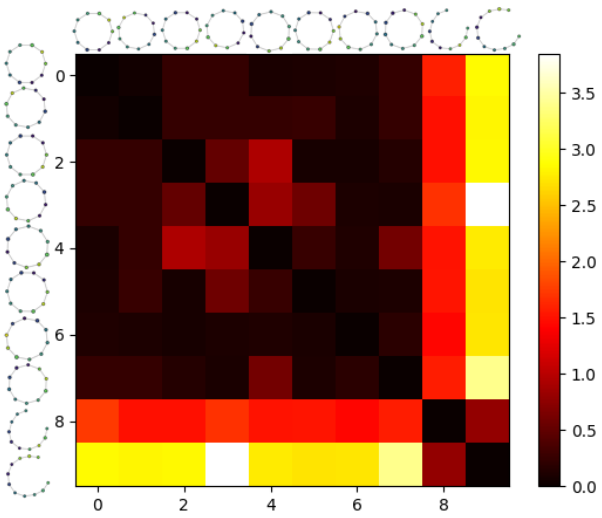

Figure 8: This is the corresponding heat map of the GW distance matrix output of the "Ten Noisy Circles" Experiment.

We expected to see more or less similar distances between each pair of circles which agrees with our heat map. However, we can see that there was greater distance between the ninth and tenth "circles," where the topology breaks and both are actually now lines, and every other circle. The distance between the ninth and tenth circles is low as expected since both of their Mapper outputs are in the shape of a line. This experiment demonstrates how our method can be used to determine stability with respect to noise. If this was Mapper on real data then we would have determined that our circle representation is stable under Gaussian noise of standard deviation 0.315 but not beyond.

6.3 "Ten Interval Sizes" Experiment

Our next experiment involved ten circles where for each circle we applied normally distributed noise with a standard deviation of 0.1, but increased the number of intervals in its cover while always keeping the percent overlap at 30 %. This will result in an increase in the number of nodes per circle thus an increase in its size. Starting with 2 intervals in the first circle, we increased the number of intervals by 1. The heat map below is the result of the experiment:

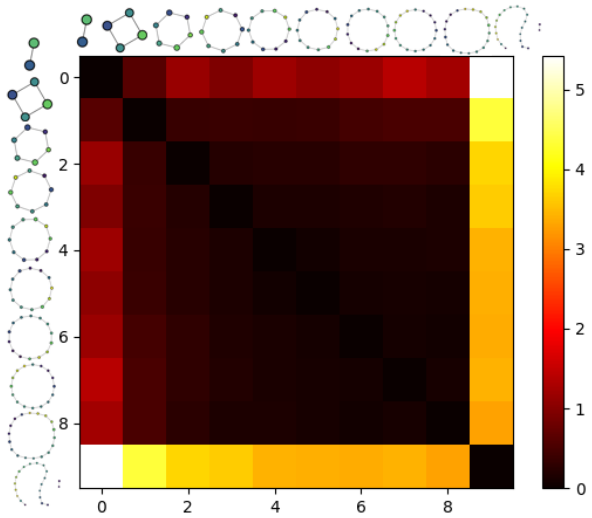

Figure 9: This is the corresponding heat map of the GW distance matrix output of the "Ten Interval Sizes" Experiment.

What we can see here is that the pairwise distances between circles with a small difference in the number of nodes tend to have smaller distances. This explains why we see darker and darker squares as we go from the top left corner to the bottom right. However, we notice that the pairwise distance between the tenth "circle," which is now a collection of two lines, and every other circle is relatively quite large. We also notice that the pairwise distance between the first "circle," whose shape is deformed and is a line, and every other circle is also quite large. This experiment demonstrates how we can determine stable parameter ranges. If this was real data we might conclude that a stable choice for the number of intervals was six to ten because of how dark and similar the corresponding squares in the heat map are. An important observation to be made is that there is little difference between circles two through nine. If we were just using the path metric this would not be the case because if every edge had weight 1 then smaller circles would have smaller diameters and larger circles would have larger diameters. This would result in the GW-distance between two circles to be proportional to their size difference and not their shape. This is why it is important to weight the edges how we did so we can distinguish between shape types and not shape size.

7 Real Data Experiment

We will now give an application of our method to real data. The following data set consists of demographic and electoral data for Chicago's 2069 precincts. The columns for this data set include the percent of the population in each racial and ethnic category as reported by the 2010 Census as well as the percentage in each of the sixteen income brackets. There is also one additional column corresponding to vote shares for the candidate Chuy Garcia in the 2015 mayoral race. Our filter function will be the vote shares for Garcia. Below is the output:

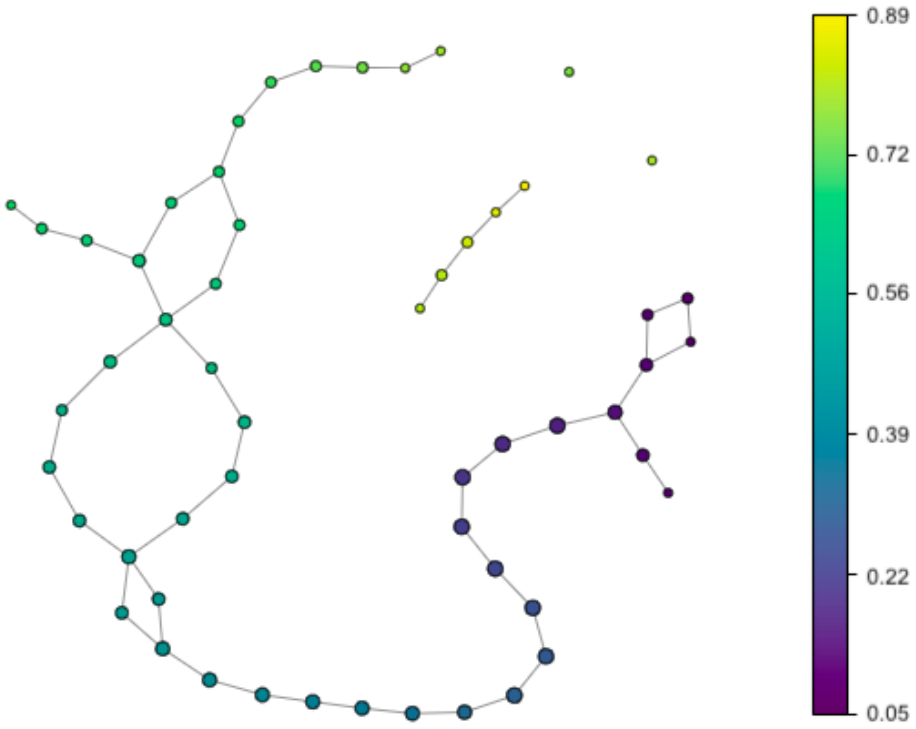

Figure 10: This is the Mapper output of the Chicago demographic and electoral data when the column for the vote share for Garcia in the 2015 mayoral race is used as the filter function. The color bar indicates which color is associated with which voting percentage for Garcia in a given precinct.

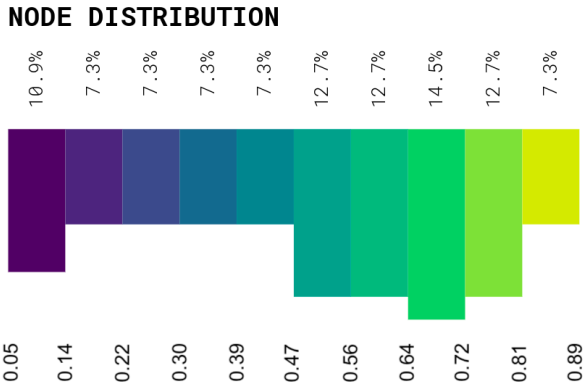

Figure 11: This is the node distribution of the graph in figure 10. The bottom row of values represent Garcia's popularity in a given precinct - the percentage of voters who voted for Garcia. The top row of values indicate the percentage of precincts with respect to that popularity. For example, based on the first bar of the histogram we can see that for 10.9 % of precincts in Chicago, Garcia was able to gain between 5 and 14 % of each precinct's vote.

Instead of using the default coloring scheme from Kepler Mapper, we specifically chose to color the nodes by filter value. We used a viridis color map which goes from purple to yellow or from low to high support for Garcia in our case. The nodes are colored by average filter value. We can see that the darker colored tail end of the Mapper output branches off into two different sets of nodes. This indicates that there are two different groups of people or two different disjoint sets of precincts that have low support or voting percentages for Garcia. By "different" this could be with regard to the racial makeup of the precincts and the income of the residing citizens. In fact, after looking at the original data file, we are able to determine that in one branch there is a precinct with a non-Hispanic white population of 49 % but in the other branch there is a precinct with a non-Hispanic Black population of 95 % but a non-Hispanic white population of only 1 %. As we increase the voting percentage, the Mapper graph takes on the shape of a line. Therefore, we can determine that there is no diversity in these precincts or voters with respect to a specific range of voting percentages. This changes when we reach the portion of the graph where there is medium support or nodes where each precinct had around half its population vote for Garcia. Specifically, we can see a loop of ten nodes on the left hand side which indicates that at some level of support we begin to see two different groups of people or two different disjoint sets of precincts that support Garcia with respect to a specific voting percentage. In fact, after looking at the original data file, we are able to determine that on one side of the loop there is a precinct with a Hispanic population of 86 % but on the other side of the loop there is a precinct with a non-Hispanic Black population of 80 % but a Hispanic population of only 2 %. Eventually the two branches "converge" to form a loop indicating that at some higher level of support the group of people become similar again or the set of precincts is more uniform. This same logic can be applied to explain the two sets of nodes branching off from the next loop in the graph. We can also see that there is one separate component in the shape of a line with the highest amount of support possible whose precincts are disjoint from the rest of the graph. After looking at the original data file, we are able to determine that in one of these high support nodes there are two precincts with a Hispanic population of 77 % and 60 %.

For our experiment with this data we increased the amount of noise in the original data set in ten instances similar to how we added noise to our ten circles from our earlier experiment. To be specific, suppose we have a variable called q_i and $q_0 = 0$. For data set $i \in \{1, 2, ..., 9\}$ we applied normally distributed noise with a standard deviation of $q_i = q_{i-1} + 0.1/10$. We kept the number of intervals in each cover at forty with a percent overlap of 30 % for all ten data sets. The heat map below is the result of the experiment:

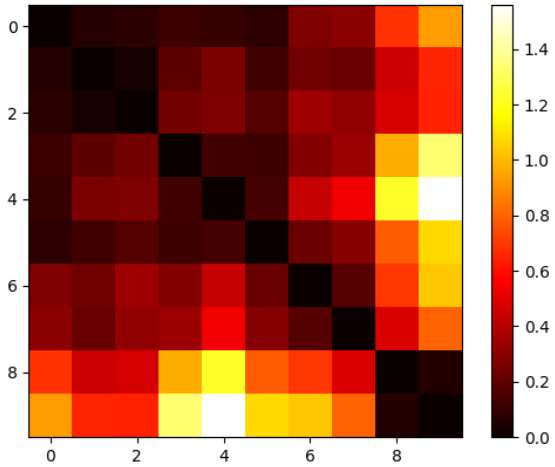

Figure 12: This is the corresponding heat map of the GW distance matrix output of the noisy Chicago experiment.

Now that we have a visual representation of Garcia's election data as a Mapper object we can discuss the features that comprise it such as the figure eights, loops, lines, or Y's. The goal of the noisy Chicago experiment was to help determine if these features were robust to small changes in the data. Suppose that one of the loops present was held together by only one precinct. We would like to know if small changes to the data regarding that precinct would transform the loop into a line. A large distance between Mapper outputs would indicate that significant alterations have been made to the structure of the original Mapper output based on our method. The upper left hand six by six portion of the map is dark red or black indicating low distance. However, once we applied normally distributed noise with a standard deviation of 0.06 the distance slightly increased and then with a standard deviation of 0.08 the distance relatively greatly increased and the rest of the heat map is evidently brighter.

8 Conclusion

In this thesis, we designed and tested a method for comparing Mapper objects. We saw a variety of experiments, specifically our first three, that demonstrated how our method could be used to determine stability with respect to certain parameter choices such as added noise or changes in the cover. We wanted to preserve shape structure as well as distinguish shape types as much as possible. Our experiments on the shape data appear to be successful according to our heat maps. With regard to this aspect of the thesis we have succeeded. Our experiment with the real data also appeared to be successful in terms of hoping to obtain a heat map similar to the "Ten Noisy Circles" experiment. As far as future applications of our method, we could use it in the case of electoral data to determine if there is a significant shift

in vote shares by analyzing how different the Mapper output is with respect to the change in data.

References

[1] Francisco Belchí, Jacek Brodzki, Matthew Burfitt, and Mahesan Niranjan. A numerical measure of the instability of mapper-type algorithms, 2019.

[2] P. Fitzpatrick. *Advanced Calculus*. Pure and applied undergraduate texts. American Mathematical Society, 2009.

[3] Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021.

[4] Chang-yu Guo. Mappings of finite distortion between metric measure spaces. *Conformal Geometry and Dynamics of the American Mathematical Society*, 19:95–121, 04 2015.

[5] Yanir Kleiman and Maks Ovsjanikov. Robust structure-based shape correspondence, 2018.

[6] Michael McCabe. Mapper comparison with wasserstein metrics, 2018.

[7] 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.

[8] G. Peyré, Marco Cuturi, and J. Solomon. Gromov-wasserstein averaging of kernel and distance matrices. In *ICML*, 2016.

[9] G.G. Roussas. *An Introduction to Measure-theoretic Probability*. Elsevier Science, 2004.

[10] Gurjeet Singh, F. Mémoli, and G. Carlsson. Topological methods for the analysis of high dimensional data sets and 3d object recognition. In *PBG@Eurographics*, 2007.

[11] Hendrik Jacob van Veen, Nathaniel Saul, David Eargle, and Sam W. Mangham. Kepler mapper: A flexible python implementation of the mapper algorithm. *Journal of Open Source Software*, 4(42):1315, 2019.

[12] Hendrik Jacob van Veen, Nathaniel Saul, David Eargle, and Sam W. Mangham. Kepler Mapper: A flexible Python implementation of the Mapper algorithm, October 2020.

[13] Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. In *Proceedings of the Twentieth Annual Symposium on Computational Geometry*, SCG '04, page 347–356, New York, NY, USA, 2004. Association for Computing Machinery.