

Spring 4-27-2022

Topological Data Analysis with Mapper

Gretchen Langenbahn
langeng@bgsu.edu

Follow this and additional works at: <https://scholarworks.bgsu.edu/honorsprojects>



Part of the [Data Science Commons](#)

How does access to this work benefit you? Let us know!

Repository Citation

Langenbahn, Gretchen, "Topological Data Analysis with Mapper" (2022). *Honors Projects*. 732.
<https://scholarworks.bgsu.edu/honorsprojects/732>

This work is brought to you for free and open access by the Honors College at ScholarWorks@BGSU. It has been accepted for inclusion in Honors Projects by an authorized administrator of ScholarWorks@BGSU.

Topological Data Analysis with Mapper

Gretchen Langenbahn

HONORS PROJECT

Submitted to the Honors College
at Bowling Green State University in partial fulfillment of
the requirements for graduation with

UNIVERSITY HONORS

4/27/2022

Dr. Umar Islambekov Department of Mathematics & Statistics

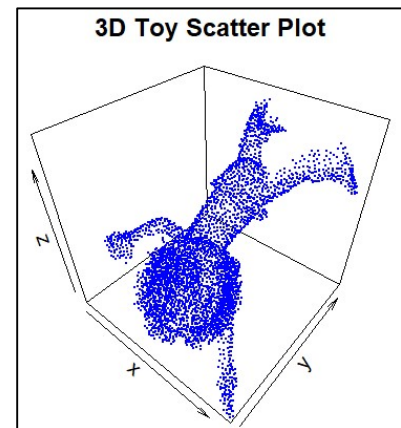
Dr. Jong Kwan “Jake” Lee Department of Computer Science

Data visualization is one of the most important aspects of data analysis as it allows for further interpretation and exploration of data. Data visualization allows for much easier human interpretation of data through graphs and maps. The difficulty of visualization can vary based on data complexity. High dimensional data sets are data sets where the number of variables (or features) is high. While it is often more difficult, visualization of high dimensional data is often the most rewarding as high dimensional data is the most difficult to interpret and visualization often shows hidden connections.

Visualization of high dimensional data sets can be challenging, as more variables complicates visualization methods. That is to say, each variable is a new dimension on the graph that needs to be visualized in the 3D or 2D space. In order for all the variables of a high dimensional data set to be expressed on a 2D plane a lot of dimension reduction needs to happen. Different visualization methods have different ways of going about this. One such visualization method is called Principal Component Analysis, or PCA. An advantage of PCA analysis is that it is very easy to implement. One key disadvantage is that PCA is heavily linearly based and has difficulty capturing non-linearities in the data. Because of this, PCA loses a lot of information to oversimplification. There are other methods of visualization, such as factor analysis, all of which have advantages and disadvantages.

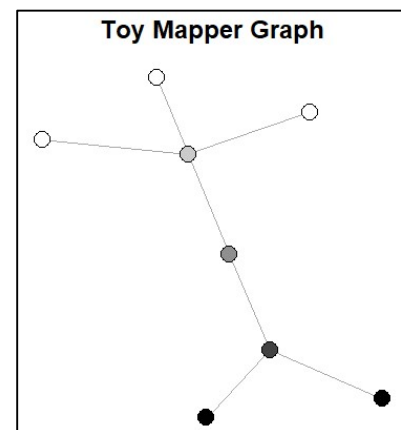
The visualization method discussed in this paper is called Mapper. Mapper uses a topological approach to data interpretation, which makes it more flexible and capable of expressing complexities in data. Because of this flexibility, Mapper is able to capture data shape relationships often lost in other analysis methods. One thing of note in Mapper is that it is not only a visualization method, but also a clustering technique. However, unlike more common

clustering methods, Mapper is a soft clustering method. In soft clustering methods, clusters are allowed to overlap and share data points. Whereas in hard clustering methods data clusters are binary and do not overlap at all. For instance, when using a K-means clustering algorithm all data points are sorted by how close they are to each K center. It only matters which K center the point is closest to, the second closest is irrelevant. In the Mapper algorithm, on the other hand, the data set is divided into overlapping subsets, and as long as a point is in a subset, it is counted as part of that cluster. This results in the same point being in multiple clusters, which helps Mapper to depict the shape of the data.



Graph 1: 3D Toy Scatter Plot

Mapper is a type of clustering algorithm that depicts the shape of the data in a graph with two key features: nodes and edges. Graphs 1 and 2 are depictions of the same data. That data is a 3-dimensional coordinate data set of a toy figure. If you look at Graph 1 you can see the shape of the toy in a 3D space. Graph 2 is a Mapper graph of the same data set.



Graph 2: 3D Toy Mapper Graph

Notably, in the Mapper graph the shape of the toy is still easily seen as a head, arms, chest, and legs. Each dot on Graph 2 is called a node, and each line between the nodes is called an edge. What Mapper does is that it partitions a data set into subsets and forms clusters out of all the points in each subset. Those clusters are then added as nodes on a graph. But since it is possible for the same point to be in multiple nodes, Mapper also adds a line between the two nodes if they share a common, which is called an edge.

A Mapper graph has the ability to depict a lot of different information. The standard Mapper graph simply has the nodes and edges depicted. This shows the general shape of the data but doesn't really aid much in visual interpretation of the data. With some labeling effort it is possible to color the nodes of the graph by some node factor. The standard factor we used to color each node was to count the number of points in each cluster that belonged to each categorical variable. So, if there was the most of class x in cluster 1 then cluster 1 would be colored based on class x . This wasn't a perfect method, however, as the proportions of classes x and y in the total data set might differ. Say there were 90 x and only 10 y . In that case, there would be very few nodes colored for y clusters. To fix this issue, we took a proportion table of the data, say .9 for x and .1 for y , and compared the proportions of each cluster to the proportion of the data instead. So, if a cluster was made up of above, or equal to, .1 percent of y instead of x , then the cluster would be colored for class y . But it is possible to make graphs with more than the nodes colored. Amongst other effects, it is also possible to adjust the size of a node or edge depicted. We adjusted the size of nodes and edges based off of the number of points represented by each.

To build a model in Mapper the first necessary input, is to specify a filter function defined on the data points. A filter function can be univariate or multivariate. The values assigned to the data points by the filter function can be regarded as quantitative attributes attached to them. In this project we started with the first filter function that occurred to us, norm analysis, and then we used some of the more common filter functions for TDA analysis: centrality, PCA, and eccentricity. We found those filter functions reading "An Introduction to Topological Data Analysis: Fundamental and Practical Aspects for Data Scientists," by Frédéric

Chazal and Bertrand Michel. PCA stands for Principal Components Analysis, which is a method for dimension reduction. Centrality analysis gives the sum of distance from a given point to all other points. Eccentricity analysis works in a similar way to centrality, though eccentricity uses the maximum distance instead of the sum. And norm analysis is simply the norm of a point in the Euclidean space.

Along with selecting the filter function, there are three variables it is necessary to specify: the number of intervals, the percentage of overlap, and the number of bins when clustering. The number of intervals could be anything above one, however, the higher the number the longer the processing time. Because of this we usually kept number of intervals under ten. These intervals cover the range of the filter function. The percentage of overlap effects how much overlap there is between the intervals. Higher overlap percentage usually causes there to be more connectivity between preimages of the filter function, so more edges. And the opposite when overlap was decreased. The number of bins when clustering, like number of intervals, doesn't have a limit, but going over ten resulted in longer process time. One downside to using Mapper is that there is no optimization method for selecting any of the four variables. There are a lot of different filter functions, each have advantages and disadvantages, but there is not a standard for selection. The other three variables also do not have optimization methods, and small changes strongly effect the graph. The solution to this problem is to experiment with the different possible combinations, and to remember that there isn't an optimal graph.

One other issue with Mapper is that the variables have to be numeric. While Mapper is good at interpreting noisy data sets, not being able to process categorical variables did limit

some of the available data sets. For instance, the original plan for this project was to use a data set from the BGSU learning commons. However, that data set was largely comprised of categorical variables, which Mapper would not have been able to use. Therefore we were unable to use that data set and had to find others.

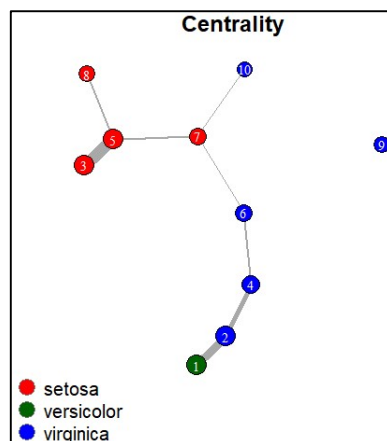
The first data set we used was called Iris. The Iris data set is a data set built into R, and it is commonly used as a learning tool to learn about new model methods. Which is what we used it as, to learn more about Mapper and adjusting the parameters for the models. Iris has one hundred and fifty values, and each value is a flower with data recorded about each flower. There are five variables in the data set: Sepal.Length, Sepal.Width, Petal.Length, Petal.Width, and Species. Starting with Species, Species is a categorical variable that lists the species of flower. There are three options of flower species: Setosa, Virginica, and Versicolor. As previously stated, there are one hundred and fifty values in the data set, and each species of flower makes up fifty values. Species was used as the class variable for the model generation. The next variable was Sepal.Length, which was the measured length of each flower's sepal. The range of Sepal.Length was 4.3 to 7.9. The next variable was Sepal.Width, which was the measured length of each flower's sepal width. The range of Sepal.Width was 2.0 to 4.4. The next variable was Petal.Length, which was the measured length of each of the flower petals' length. The range of Petal.Length was 1.0 to 6.9. The next variable was Petal.Width, which was the measured length of each of the flower petals' width. The range of Petal.Width was 0.1 to 2.5.

What we expected when mapping Iris was for each species of flower to be clustered together. To show this the model we used had each node colored by the majority of which type

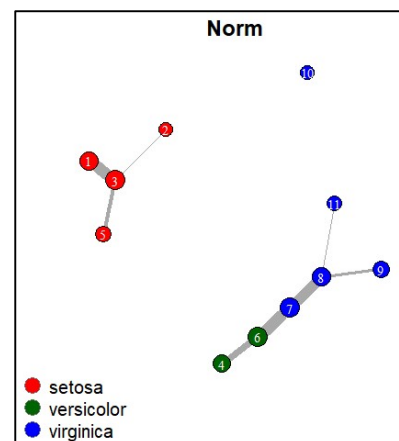
of flower was in that node. We also expected there to be some overlap between the Virginica and Versicolor species, as those species are similar to each other in terms of size. Setosa, on the other hand, is more unique than the other two, running at a much smaller average in all four variables, so it was expected to be more isolated and not connect to the other flower nodes. We also wanted to limit outliers, meaning no solitary nodes, in the model. Lastly, we wanted the cluster sizes to be generally equal, as the data was well proportioned.

To start with, we selected the filter function, because it was the variable that effected the graph the most. Looking at Graphs 3 through 6 it is possible to see each filter function

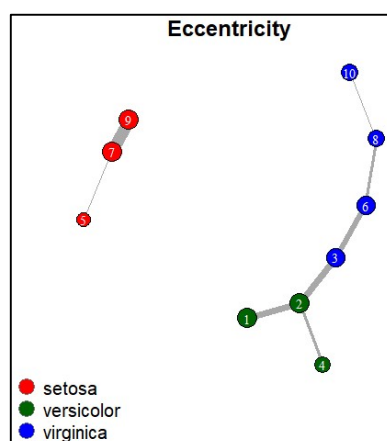
being used, and how strongly the filter function effects the graph. We based our selection process off of which graph looks the most similar to our expectations. For instance, we selected the eccentricity filter function, Graph 5, to build the next models of Iris. Eccentricity was selected because Graph 5 is the most similar to what our expectations were, meaning nodes grouped by flower types and isolated setosa flowers.



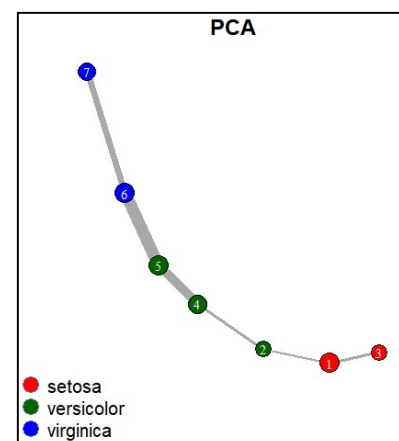
Graph 3: Iris with Centrality Filter



Graph 4: Iris with Norm Filter



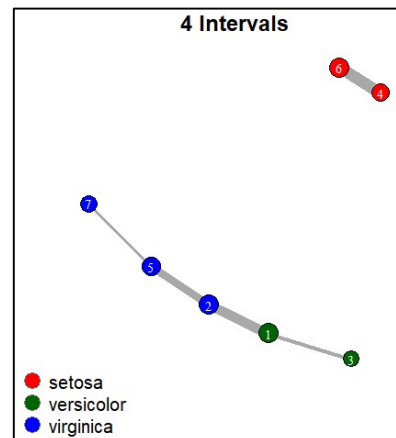
Graph 5: Iris with Eccentricity Filter



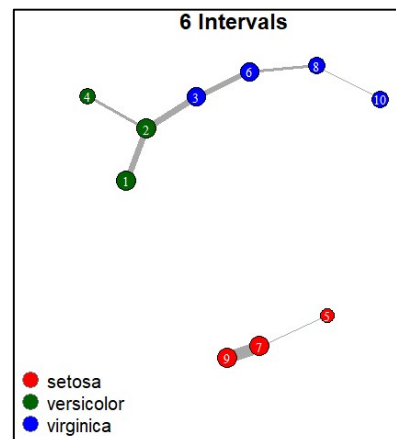
Graph 6: Iris with PCA Filter

Graph 4, the norm function, had an outlier that was not expected. PCA, depicted in Graph 6, was also okay as it linked the nodes in decreasing order, however it still linked setosa when we didn't want that. Lastly Graph 3, centrality, seemed to be the worst of the lot, as it linked all nodes together in a disjointed way.

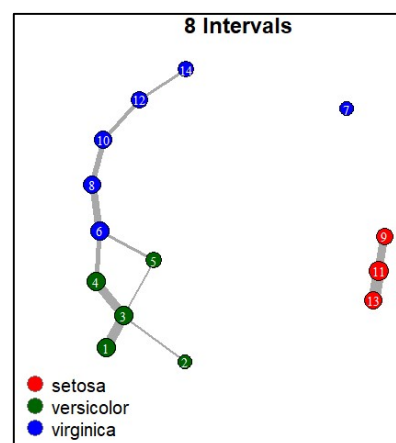
The next variable selected was number of intervals, as seen in Graphs 7 through 9. As a general rule, as the number of intervals is increased the number of clusters, and thereby nodes on the graph, increase. With the opposite being true with decreasing the number of intervals. As you can see in Graph 7, the four intervals used limited the number of nodes. And in Graph 9, the number of nodes increase. However, Graph 9 also has an outlier when we expected there to be none, as well as adding another edge between versicolor and virginica. Graph 8, on the other hand, does not have an outlier, and it has fewer edges connecting versicolor and virginica nodes than Graph 7. Therefore, we selected Graph 8 as the optimal graph out of these three. Keeping eccentricity as our selected filter function, the next models will be built using six intervals.



Graph 7: Iris with Four Intervals

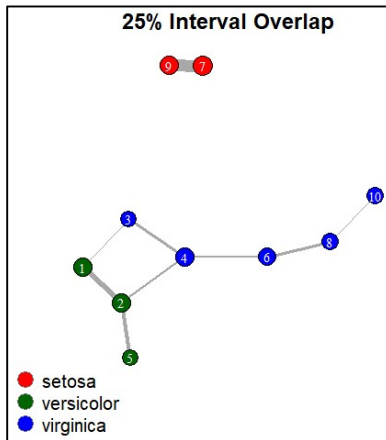


Graph 8: Iris with Six Intervals

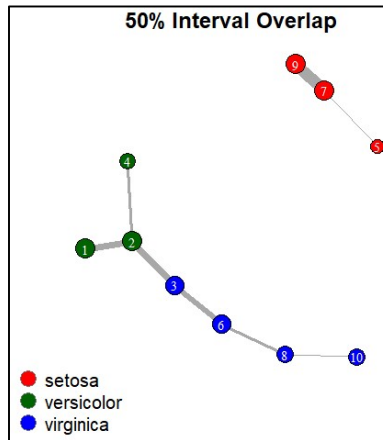


Graph 9: Iris with Eight

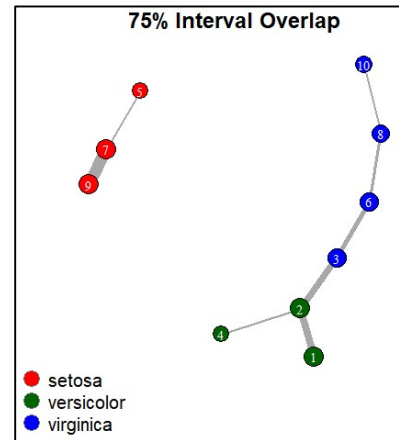
Graph 10: Iris with 25% Overlap



Graph 11: Iris with 50% Overlap



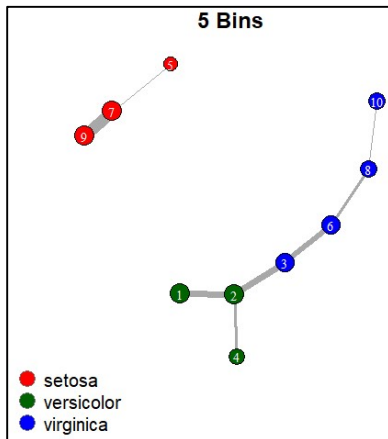
Graph 12: Iris with 75% Overlap



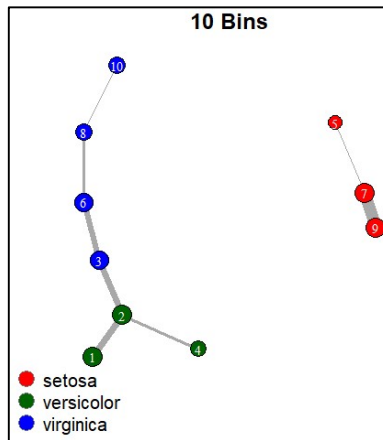
The next variable selected was percentage of overlap, as seen in Graphs 10 through 12. Looking at these graphs you can see percentage of overlap mostly effects the number of edges. There is also some effect on the number of nodes. In Graph 10 you can see a reduction in the number of nodes and edges, this is a good model with no outliers and the expected grouping of flower types. However, in comparison to the other two models, the number of edges between versicolor and virginica increase, which lowers the quality of the graph. The interesting thing about Graphs 11 and 12 is that they seem to be identical. This means that even with increased overlap setosa flowers are still too dissimilar from the other two flowers to connect. This helped support our decision to discount graphs with setosa overlap. Though while Graphs 11 and 12 appear to be identical they are not, as the number of edges vary. Graph 11 had fewer edges connecting Nodes 2 and 3 then Graph 12. So, we selected 50% interval overlap for the next models because Graph 11 had fewer edges connecting versicolor and virginica.

The final variable to select was number of bins when clustering, which can be seen in Graphs 13 through 15. In Graph 15 the number of nodes has increased and there is now an

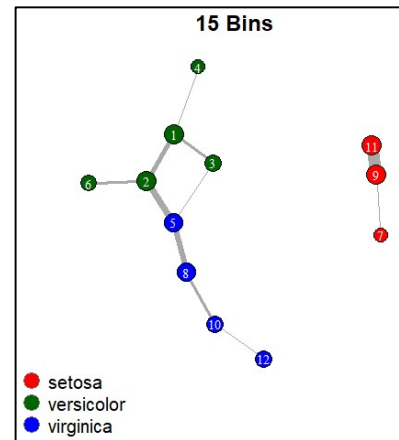
Graph 13: Iris with 5 Bins



Graph 14: Iris with 10 Bins



Graph 15: Iris with 15 Bins



increased number of edges between versicolor and virginica. This is not in favor of our expectations for the model, so not an optimal graph. Graphs 13 and 14 are interesting because they are, once again, seemingly identical. That means anything under 10 bins will not affect the model nodes. Yet, once again, there is differences in the number of edges between Nodes 2 and 3 in Graphs 13 and 14. Graph 14 has fewer edges between versicolor and virginica, so we selected Graph 14 as our final model. Once again, there is no optimization method for Mapper, so Graph 14 isn't necessarily the best graph. It was just the one selected for this modeling run through.

The second data set was called Fetal Health. We found Fetal_health on Kaggle, a website for sharing data sets. Fetal_health is a high dimensional data set with twenty-two different variables and 2126 different values. The data itself is from a study meant to classify the health of a fetus in order to study, and prevent, child and mother mortality. The first variable is fetal_health, which is a categorical variable. fetal_health is what we used as the class variable for the model. fetal_health has three categories, labeled one to three: healthy or normal fetus, on watch for possible sickness, and pathological. The next variable was

baseline.value, which represents the baseline fetal heart rate. The range of baseline.value was 106 to 160. The next variable was accelerations, which represents the number of accelerations of the fetal heart per second, if there were no accelerations zero was recorded, which can be filtered out if categorizing. The range of accelerations was 0.000 to 0.019. The next variable was fetal_movement, which recorded the number of movements of the fetus per second. Once again, if there were no movements zero was recorded, which, again, can be filtered out. The range of fetal_movement was 0.000 to 0.481. The next variable was uterine_contractions, which records the number of contractions per second. The range of uterine_contractions was 0.000 to 0.015, if there were no uterine contractions zero was recorded. The next variable was light_decelerations, which counts the number of light decelerations, or LDs, per second. The range of light_decelerations was 0.000 to 0.015, if there was no LDs then zero was recorded. The next variable was severe_decelerations, which counts the number of severe decelerations, or SDs, per second. The range of severe_decelerations was 0.000 to 0.001, the majority of this variable was zero which meant there were no severe deceleration. The next variable was prolonged_decelerations, which counts the number of prolonged decelerations, or PDs, per second. The range of prolonged_decelerations was 0.000 to 0.005, where zero represented no prolonged decelerations. The next variable was abnormal_short_term_variability, which records the percentage of time where the fetus experience abnormal short-term variability. The range of abnormal_short_term_variability was 12 to 87. The next variable was mean_value_of_short_term_variability, which represents the mean time the fetus experience short term variability. The range of mean_value_of_short_term_variability was 0.2 to 7.0. The next variable was percentage_of_time_with_abnormal_long_term_variability, which records

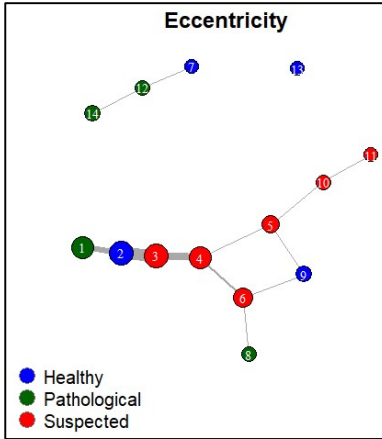
the percentage of time where the fetus experience abnormal long-term variability. The range of `percentage_of_time_with_abnormal_long_term_variability` was 0 to 91, with zero representing the fetus experiencing no long-term variability. The next variable was `mean_value_of_long_term_variability`, which records the mean of the amount of time where the fetus experience abnormal long-term variability. The range of `mean_value_of_long_term_variability` was 0.0 to 50.7 with zero representing the fetus experiencing no long-term variability. The next variable was `histogram_width`, which represents the width of the histogram made using all values from the record, made from previous examinations, of that fetus. The range of `histogram_width` was 3 to 180. The next variable was `histogram_min`, which represents the minimum value of the histogram of that fetus. The range of `histogram_min` was 50 to 159. The next variable was `histogram_max`, which represents the maximum value of the histogram of that fetus. The range of `histogram_max` was 122 to 238. The next variable was `histogram_number_of_peaks`, which represents the number of peaks of the histogram of that fetus. The range of `histogram_number_of_peaks` was 0 to 18. The next variable was `histogram_number_of_zeroes`, which represents number of zeros of the histogram of that fetus, which means the number of times there is no data from the fetus's exam. The range of `histogram_number_of_zeroes` was 0 to 10, zero in this case meaning the histogram had no missing values. The next variable was `histogram_mode`, which represents the mode value of the histogram of that fetus. The range of `histogram_mode` was 60 to 187. The next variable was `histogram_mean`, which represents the mean value of the histogram of that fetus. The range of `histogram_mean` was 73 to 182. The next variable was `histogram_median`, which represents the median value of the histogram of that fetus. The range of `histogram_median`

was 77 to 186. The next variable was `histogram_variance`, which represents the variance value of the histogram of that fetus. The range of `histogram_variance` was 0 to 269. The next variable was `histogram_tendency`, which represents the trend value of the histogram of that fetus. This variable had three values from -1 to 1. Negative one meaning a negative trend in the histogram and positive one being a positive trend in the histogram. Zero meaning either no trend or consistent values in the histogram.

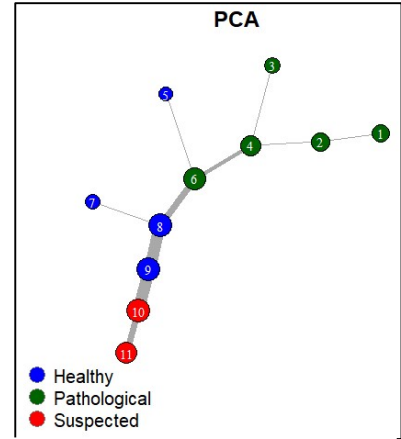
Our main expectation for the graphs of Fetal Health is that neighboring classes would be grouped together. So, healthy nodes would be connected with suspected nodes which would connect with pathological nodes. The only aspect we do not want to see on a graph is a connection between a healthy node and a pathological node, meaning a direct edge between the two nodes. The data is very imbalanced as well, so clusters of varying sizes are okay, we just want to limit single outliers.

We started to model fetal health the same way we began with the Iris data set, by selecting the filter function, as depicted in Graphs 16 through 19. At first glance, it might be assumed that Graph 17, with PCA filtering, is the best as it has no outliers. That is an incorrect assumption because the nodes connecting the three classes aren't the suspected nodes, it is the healthy nodes. And the healthy nodes being connected to the pathological nodes is the main thing we want to avoid in these models. Eccentricity, which is depicted in Graph 16, is also not a good graph as it is very scattered with overlap between the healthy nodes and the pathological nodes. The centrality graph, Graph 19, has more outliers than Graph 18 and has a connection between the healthy and pathological nodes. Graph 18 seems to be the best option of the four. Graph 18 is a model built off of the norm function. It has clustering between all of

the class categories and no connection between the healthy and the pathological nodes. Because Graph 18 was the best, we built the next models using the norm function.

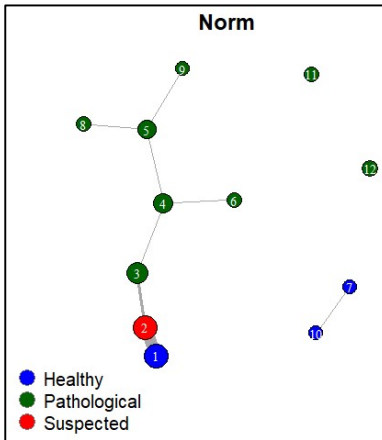


Graph 16: Fetal Health with Eccentricity

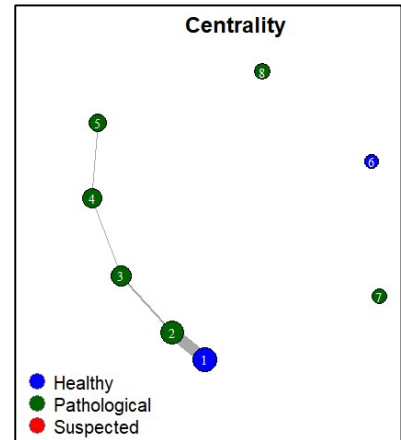


Graph 17: Fetal Health with PCA

To continue to refine the model, we moved onto selecting the number of intervals. Graphs 20 through 22, have three different models with three different numbers of intervals. Graph

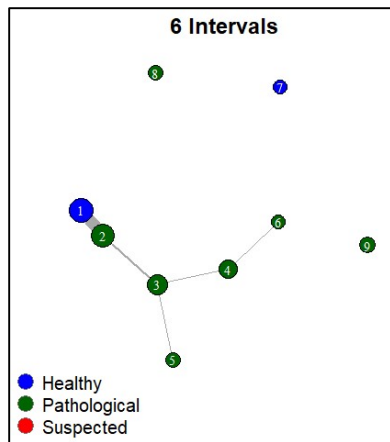


Graph 18: Fetal Health with Norm

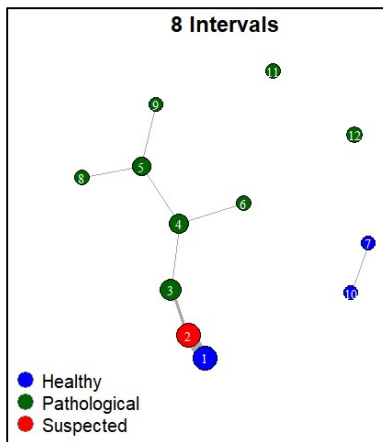


Graph 19: Fetal Health with Centrality

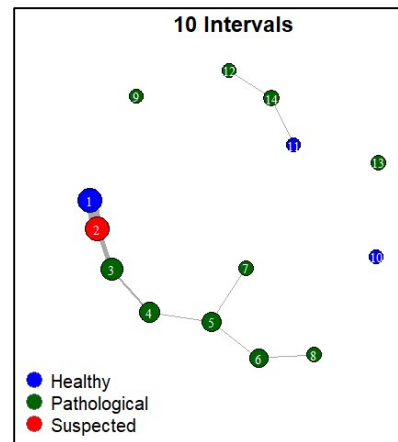
20 removes the suspected node, seen in Graphs 21 and 22, which results in a connection between a healthy node and a pathological node. Graph 22 has a similar flaw; in that it has a connection between the healthy and the pathological nodes. Graph 21 does not have this connection, making it the best fit of the three. As a result, we used eight intervals for future models.



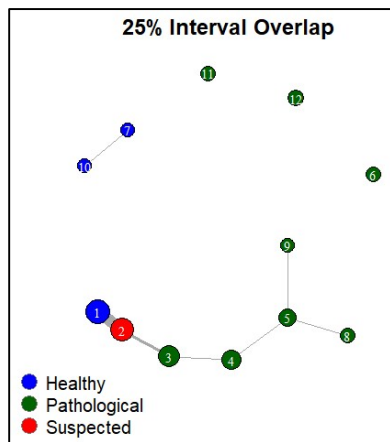
Graph 20: Fetal Health with Six Intervals



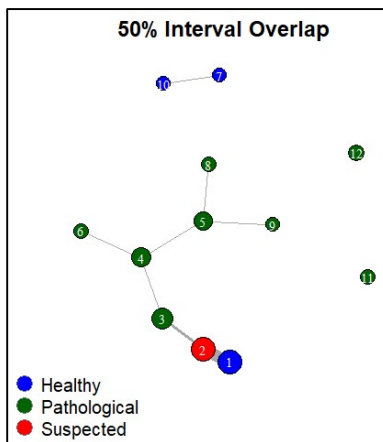
Graph 21: Fetal Health with Eight Intervals



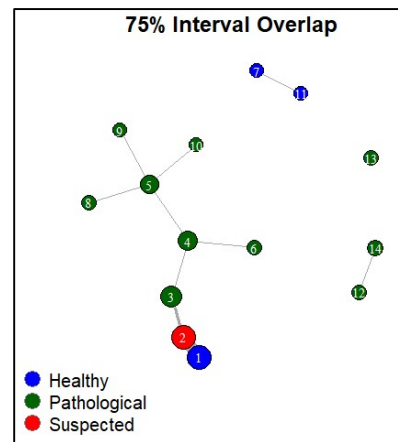
Graph 22: Fetal Health with Ten Intervals



Graph 23: Fetal Health with 25% Interval Overlap



Graph 24: Fetal Health with 50% Interval Overlap

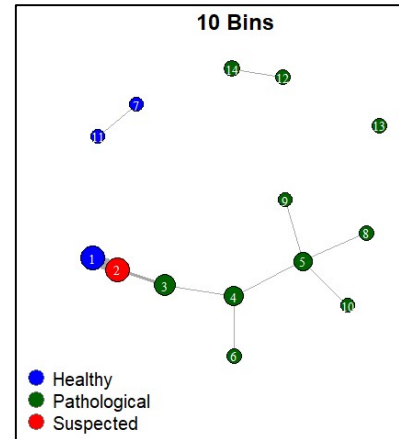


Graph 25: Fetal Health with 75% Interval Overlap

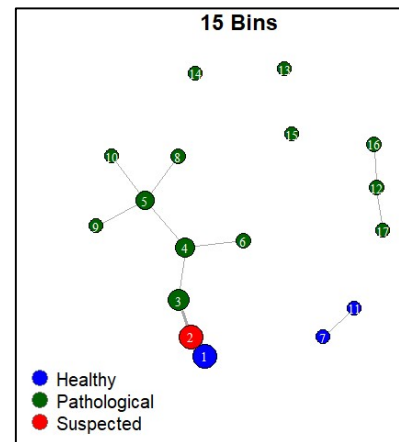
The next variable was percentage of overlap, depicted in Graphs 23 through 25. None of the three graphs, 23 through 25, violate any of our major assumptions; all like groups are together, and no connection between the healthy nodes and the pathological nodes. The major difference between the graphs is the number of outliers and lone nodes. While all three graphs are valid, Graph 25, with 75% overlap, seems to be the best of the three as it only has one isolated node. And one isolated node can be seen as better than the two isolated nodes in Graph 24. Therefore, the next models will use 75% interval overlap.

The last variable to select is the number of bins when clustering, which are depicted in Graphs 26 through 28. None of these graphs violate our main assumptions. That said, Graph 27 and 28 both have three independent outliers, while Graph 26 only has one. As a result of this, we selected Graph 26 as the best fit of these three graphs. That makes Graph 26 our final model of fetal health.

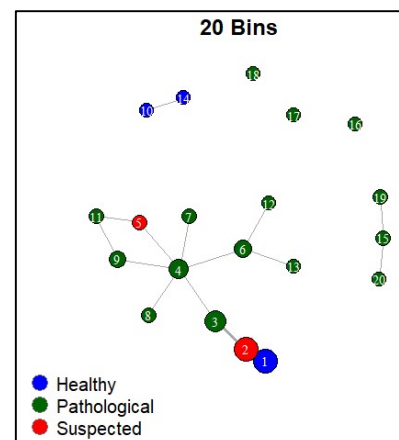
Mapper is a visualization method that uses a topological approach to data analysis. Mapper is good because its topological approach allows for flexibility in modeling that other visualization methods don't allow. While Mapper is good at expressing complexities in data, it is hard to tune. Because of this it is necessary to test and retest Mapper graphs until the user is satisfied. For instance, with the two data sets used in this paper, the final graphs produced are not the optimized versions for these data sets. There is no optimal graph. They are simply an option of graph. For further study the data sets can be retested with the final variable values as the starting values, or could try new variables as class variables, or could try new filter functions. There is a lot more possibility for exploration with Mapper. And given that Mapper is a useful tool in the data science field, it is important to understand it.



Graph 26: Fetal Health with Ten Bins



Graph 27: Fetal Health with Fifteen Bins



Graph 28: Fetal Health with 20 Bins

Works Cited

- Chazal, Frédéric, and Bertrand Michel. "An Introduction to Topological Data Analysis: Fundamental and Practical Aspects for Data Scientists." *Frontiers in artificial intelligence* vol. 4 667963. 29 Sep. 2021, doi:10.3389/frai.2021.667963
- Chazal, Frederic, and Bertrand Michel. "Mapper Algorithm with the R-Package TDAmapper." *Mapper Algorithm with the R-Package TDAmapper*, May 2016, <http://bertrand.michel.perso.math.cnrs.fr/Enseignements/TDA/Mapper.html>.
- Jacob van Veen, Hendrik, et al. "Keplermapper 2.0.1 Documentation a Scikit-TDA Project." *Background - KeplerMapper 2.0.1 Documentation*, 2021, <https://kepler-mapper.scikit-tda.org/en/latest/theory.html>.
- Kraft, Rami. "Illustrations of Data Analysis Using the Mapper Algorithm and Persistent Homology." (2016).
- Munch, E. "A User's Guide to Topological Data Analysis". *Journal of Learning Analytics*, vol. 4, no. 2, July 2017, pp. 47–61, doi:10.18608/jla.2017.42.6.
- Singh, Gurjeet Kaur Chatar et al. "Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object Recognition." *PBG@Eurographics* (2007).
- "TDA Mapper Part 1: Introduction." *YouTube*, uploaded by Isabel K. Darcy, 18 July 2020, https://www.youtube.com/watch?v=DD0_zPIEsgY.
- "TDA Mapper Part 2: Examples." *YouTube*, uploaded by Isabel K. Darcy, 18 July 2020, <https://www.youtube.com/watch?v=dApjJpQZY0Y>.
- "TDA Mapper Part 3: Summary." *YouTube*, uploaded by Isabel K. Darcy, 18 July 2020, <https://www.youtube.com/watch?v=5aTcZ0tcQbA>.