Joshua Sheldon, Justin Barnwell, Michelle Arubi
Dr. Mitra
CSE 5290
February 8, 2024

Project 5 - Graduate Project Proposal

Topological Data Analysis (TDA)

Scope

Our primary objective is to use the Mapper algorithm to cluster samples of MNIST digits, evaluate their topological relatedness/homological similarity, and display that through a graph. This is fundamentally the success condition, fulfilling the objective of the project.

However, should we have additional time, we could also use persistence homology to identify the features of each number 0-9, and construct graphs displaying how each number is arranged from a topological perspective.

Current Understanding of Project

The objective of this project is to come to a more full understanding of topological data analysis (TDA). TDA is a field which aims to identify the topological features, or shape, of data, identify the defining/impactful features, and use that information to make inferences on the clustering of data, the relationship/closeness between different pieces of data, the future trajectory of a set of continuously collected data, and more. Along with this field are many algorithms, strategies, and tools used to analyze data which allow for the broad spectrum of insights to be made in a single field. Generally, we aim to explore the concepts and implementations of different aspects of TDA.

Mapper

The specific application outlined by this project is to apply the Mapper algorithm to the MNIST dataset. The Mapper algorithm has 3 key steps:

1. **Reduce the dimensionality of the data** - Attempt to map high-dimensional points to lower dimensions, such as 2D. This typically removes information that doesn't define the data or inform the relationships between data points, and makes it easier to cluster data points. On a per MNIST digit (hereafter referred to as "digit") basis, this could be easy, as we can consider each pixel a data point and only carry information about the position of the pixel and whether it is filled in or not. However, as we attempt to compare digits to each other, we need to consider how to represent images in a way where similar digits are clustered together (which may involve the evaluation of the persistence homology of the digit).

2. **Form covers from reduced dimensionality data** - Now that we're in a space that is less complex, we can effectively start the clustering by creating covers. These covers will collectively cover the whole set of data points. More notably, these covers will have overlap, which indicates the relationship between data points. With our digits, this is where we start to see which numbers are closest to each other. We'd assume the 10 digits for each number would cluster together, and then see connections between numbers with similar homologies. For instance, 9 and 6 both are effectively the same shape, so there will likely be considerable overlap in those clusters. Additionally, the 9 and 6 clusters may be nearby or otherwise close to the 0 cluster, the 5 and 3 clusters will likely have overlap, etc.
3. **Construct a graph from covers** - Once we have our covers, we can create a graph, where the nodes are clusters/covers, and the edges indicate which clusters have overlap. Information about the individual and collective attributes of the data points inside each cluster and the connections between the clusters can reveal information about the data set. One helpful strategy is to organize this graph in a visually similar format to the input data (if it is easily visualizable) to more easily understand which clusters contain which data points.

Persistence Homology

Another tool in TDA is persistence homology. Persistence homology helps us learn what shapes in the data are defining or significant. The central steps of persistence homology:

1. Create *n*-dimensional spheres (with diameter *d*) around every data point, where *n* is the number of dimensions in the input space.
2. Increase *d* until we reach trivial homology, this is when there are no more holes, and for each step of *d*:
    a. Connect pairs of points that are no further apart than *d*. This creates simplices, and the collection of all simplices made of the points is called the simplicial complex.
    b. For every sphere that overlaps, fill in the simplex created by the points of all overlapping spheres. For instance, if the spheres of 3 points overlap, a triangle (2-simplex) will be formed between them, and we fill it in. This creates a simplicial complex, particularly the Rips complex of the data. Note that before the spheres overlap, the same points may make a simplex, it just won't be filled in.
    c. Evaluate the homology of the simplicial complex, or in simpler terms, record the holes formed by our filled and unfilled simplices.
3. Evaluate the persistence of all holes. As *d* grows, some holes will form, and then disappear. The persistence of a hole is the range/pair (*d1*, *d2*), where *d1* is the *d* value at a point in time where the hole appears, and *d2* is the *d* value where the hole disappears.
4. Distinguish noise from features. Holes with a short persistence (small *d2* - *d1*) are noise. These are holes that are minor variations in the defining features of the shape of the data. Holes with a long persistence (large *d2* - *d1*) are features. These are holes that are stable, meaning they persist even with alterations of the data. For example, consider the

two holes in an 8. Those are features, and even with slight changes to handwriting (noise), you can still identify an 8 through those two holes.

Once we’ve identified the features, we’ve identified the defining characteristics of the data. Another metric that can be deduced after getting the series of Rips complexes is the Wasserstein distance. This is essentially the distance between two complexes, or a metric of how much you would have to modify one complex to get to the other complex. The Wasserstein distance is useful not only as a metric to compare the fundamental similarity of two spaces (for instance, seeing how similar two pictures are for Google reverse image search), but also for evaluating the stability of data over time (how much is the data changing from point to point?)

Data Augmentation Strategy

The Mapper algorithm creates a low-dimensional representation of the data which captures its underlying shape, or topological structure, making it easier to visualize and understand complex relationships within the data. The result is a topological summary of the original dataset, which can reveal features like loops, clusters, and voids that are important for understanding the shape of the data. The Mapper algorithm is particularly useful in exploratory data analysis, where the goal is to discover patterns and structures without making a priori assumptions about the data's form or distribution.

When applied to the MNIST dataset for TDA, the Mapper algorithm will employ a unique approach to augment and understand the intrinsic geometric and topological structure of the high-dimensional data represented by the handwritten digits. This augmentation is not in the traditional sense of image transformations but rather in the creation of a simplified representation that captures the essential topological features of the data. The subsections below describe how the Mapper algorithm will enhance our understanding of the MNIST dataset through its analysis process:

1. **Dimensionality Reduction and Filtering:** Initially, the high-dimensional pixel data of MNIST images undergoes a dimensionality reduction step. This step is crucial for Mapper's functionality, as it transforms the complex, high-dimensional space into a more manageable, lower-dimensional representation. Techniques such as PCA (Principal Component Analysis), t-SNE, or UMAP could be employed here. The choice of dimensionality reduction technique and the subsequent filter function—whether it's based on feature extraction or inherent data properties—plays a significant role in how the data is augmented for analysis. The filter function projects the data points into a lower-dimensional space, effectively augmenting the dataset by emphasizing the features most relevant to the topological analysis.
2. **Covering and Clustering:** After dimensionality reduction, the Mapper algorithm constructs a cover of the filter function's output space, which consists of overlapping regions. Each region in this cover corresponds to a subset of the dataset that shares certain characteristics as determined by the filter function. The data within each region is then clustered, identifying groups of data points that are close to each other in the

reduced space. This step further augments the MNIST data by organizing it into a collection of interconnected clusters that reflect the underlying topological structure of the dataset. The choice of clustering algorithm and parameters can influence the resolution at which these topological features are identified, effectively tuning the augmentation process to highlight different aspects of the data's shape.

3. **Graph Construction and Visualization:** The culmination of the Mapper algorithm's process is the construction of a graph that represents the topological structure of the MNIST dataset. Nodes in this graph represent the clusters identified in the previous step, while edges indicate relationships between these clusters, such as proximity or overlap. This graph is a powerful augmentation of the original MNIST data, transforming thousands of high-dimensional images into a comprehensible visual representation that highlights the topological connections between different digits. For example, digits that share similar structural features, such as 6s and 9s, or 0s and 8s, might be represented by closely connected nodes or clusters within the graph, illustrating their topological similarity.

Through the Mapper algorithm, the MNIST dataset is augmented from a collection of individual, high-dimensional images into a structured, interpretable representation that reveals the complex topological relationships between different digits. This augmented view of the data enables deeper insights into the dataset's structure, facilitating discoveries that go beyond traditional clustering or classification tasks. By highlighting the continuity and connections within the data, the Mapper algorithm opens new avenues for understanding and analyzing the inherent shapes and patterns present in the MNIST dataset.

Input Data

For this project, we have been instructed to use the MNIST dataset. We pulled this data from the Common Visual Data Foundation's mirror of the original dataset, which at the time of data retrieval was unavailable. The MNIST dataset is composed of four archives:

1. Training Set Images (`train-images-idx3-ubyte.gz`)
2. Training Set Labels (`train-labels-idx1-ubyte.gz`)
3. Test Set Images (`t10k-images-idx3-ubyte.gz`)
4. Test Set Labels (`t10k-labels-idx1-ubyte.gz`)

Inside of each of these archives is one file in the IDX file format. The IDX file format is designed for representing complex mathematical structures, but in MNIST's case the application is quite simple.

Image Format

MNIST images are simple, grayscale, and low resolution. So, instead of providing hefty `.png` or `.jpg` files, with support for features like color and transparency, MNIST represents each pixel with an unsigned byte (0-255), on a scale from white (0) to black (255). Also, MNIST images are standardized, so instead of providing many files, MNIST simply collates all images into one archive file. The first 4 bytes of this file represent a magic number, the next 4 bytes represent the number of images, the next 4 bytes represent the number of rows in each image, and the next 4 bytes represent the number of columns in each image. Every byte after this represents a pixel, and to pull an image you simply read a number of bytes equal to the number of rows * the number of columns in the correct position. Assuming images start with an index of 1, you can calculate the offset of the beginning of an image and the number of bytes to read as so:

- Offset: 16 + (imageIndex - 1)
- Length: ((0x008 to 0x0012) * (0x0012 to 0x0016))

Label Format

MNIST labels are also incredibly simple, as each label must be on the interval [0, 9]. So, MNIST labels are stored in a single IDX file, just like the images. However, the label files have an 8 byte header, where the first 4 bytes is a magic number and the next 4 bytes represent the number of items. Every byte after that is a label (as an unsigned byte, 0-255 is plenty enough space). So, to get the index of the label of an image in the file (where image indices are assumed to start at 1), we plan to use:

- Index of Label Byte: 8 + (imageIndex - 1)

Output Data

Our primary output will be a series of graphs generated by the Kepler Mapper tool, visualizing the topological structure of the MNIST dataset as interpreted through the Mapper algorithm. These graphs will highlight the clustering of similar digits and the connections between different clusters, illustrating the topological proximity of digits that share visual similarities (e.g., some 9s and 0s).

1. **Mapper Graphs**: The core output will consist of Mapper graphs that represent the high-dimensional data of MNIST digits in a simplified, interpretable form. Each node in these graphs will represent a cluster of digits that are similar in their topological features, while edges will indicate relationships between these clusters.
2. **Persistence Diagrams**: For a deeper analysis, we will generate persistence diagrams for each digit class (0-9). These diagrams will help us identify and understand the topological features that persist across scales, distinguishing between noise and significant structures (e.g., loops or holes in the digits 8 and 0).
3. **Homology Feature Videos**: Depending on the scope and time availability, we plan to create videos that demonstrate the evolution of simplicial complexes for each digit as the scale parameter changes. This dynamic visualization will provide insights into how topological features emerge, evolve, and contribute to the identification of each digit.
4. **Comparative Analysis**: We will also include a comparative analysis of the topological structures of different digits. This may involve calculating Wasserstein distances between persistence diagrams to quantify the similarity between digit topologies.

These outputs will not only serve to illustrate the practical applications of TDA and the Mapper algorithm but also enhance our understanding of the underlying structure of the MNIST dataset. Our findings will be documented in a detailed report, accompanied by the generated visualizations to support our conclusions.

This approach to data augmentation and output visualization will ensure a comprehensive analysis of the MNIST dataset, allowing for robust findings and a deeper understanding of the topological relationships between handwritten digits.

References

- Bala Krishnamoorthy (2/11/2020): An Introduction to Mapper (youtube.com)
- Chazal, Frédéric, and Bertrand Michel. “An Introduction to Topological Data Analysis: Fundamental and Practical Aspects for Data Scientists.” arXiv, February 25, 2021. http://arxiv.org/abs/1710.04019.
- Dwaraknath, Anjan. “Topological Data Analysis and Deep Learning,” n.d.
- https://arxiv.org/pdf/1910.08345.pdf
- http://yann.lecun.com/exdb/mnist/
- Introduction to Persistent Homology by Matthew Wright
- Mapping Firms' Locations in Technological Space: A Topological Analysis of Patent Statistics (youtube.com)
- The Mapper Algorithm | Overview & Python Example Code (youtube.com)
- Topological Data Analysis (TDA) by Shaw Talebi
- Wagenknecht, Adam. “Topological Data Analysis of Weight Spaces in Convolutional Neural Networks,” n.d.