Introduction to Topological Data Analysis - II

Priyavrat Deshpande

Chennai Mathematical Institute

January 26, 2022

Outline

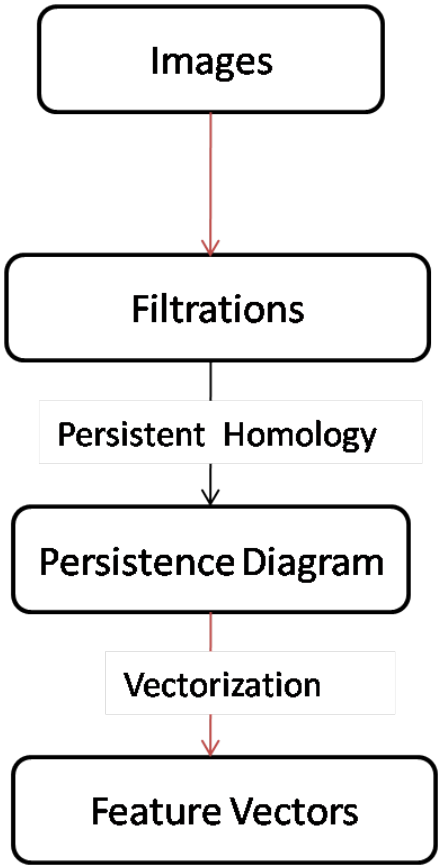
1. Classification using TDA
2. The mapper algorithm

Image Classification using PH

In image classification problems, we can often identify distinct shape features that characterize images in each class. By using tools from TDA, we can classify images based on these properties.

The aim is to develop a 'pipeline' to extract various topological descriptors from images which can serve as a basis for classification.

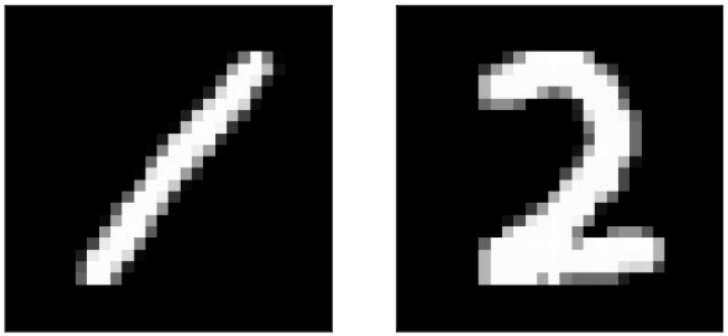
Topological Pipeline: Outline



Two key steps in the pipeline are:

- Constructing filtrations from the image.
- 'Vectorization' of the persistent diagram.

The MNIST dataset of hand written digits will be used to expand on these steps.
This consists of 70000 grayscale images of dimension 28×28 pixels.



Topolgoy of digits

Consider digits as subspaces of $\mathbb{R}^2$.

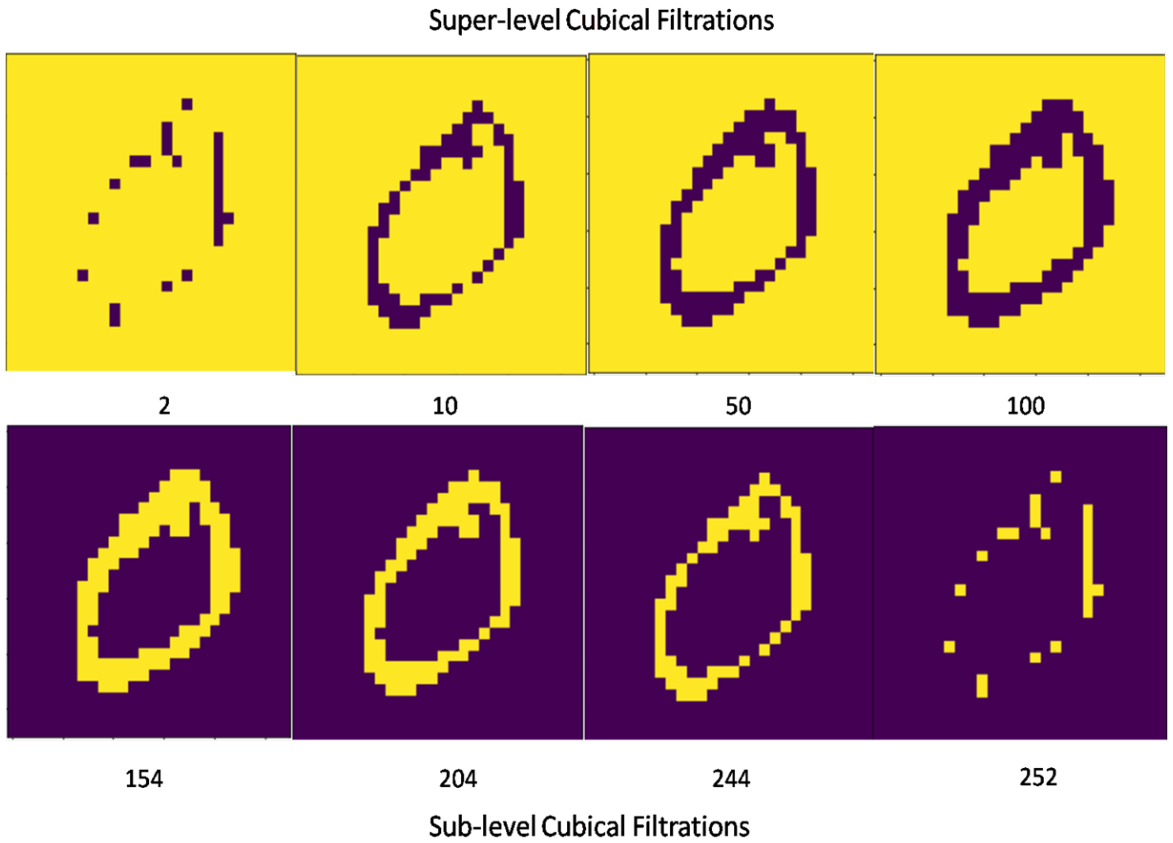
Following are the groups of topologically equivalent digit-shapes

1. $1,\ 2,\ 3,\ 5,\ 7.$
2. $0,\ 4,\ 6,\ 9.$
3. $8.$

Considering these shapes alongwith the information of how they could be drawn will help distinguish all of them.

Topological Pipeline : Filtrations

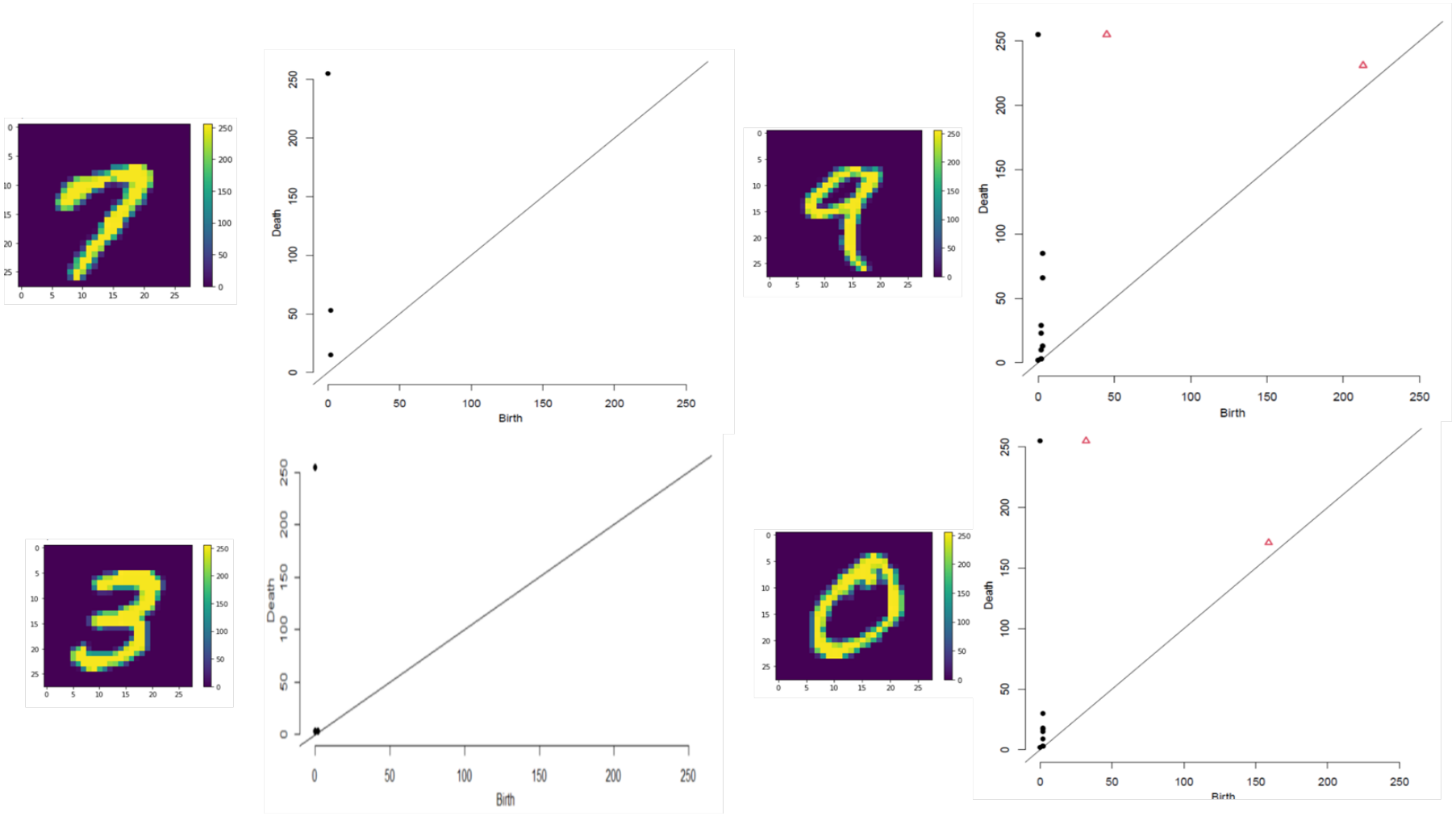
Grayscale images have a natural pixel structure and can be represented by a pixel map on a rectangular grid. As a result, the sublevel and superlevel sets of this map can be interpreted as a cubical filtration.



In the image above, cubical complexes are colored purple.

Topological Pipeline : Filtrations

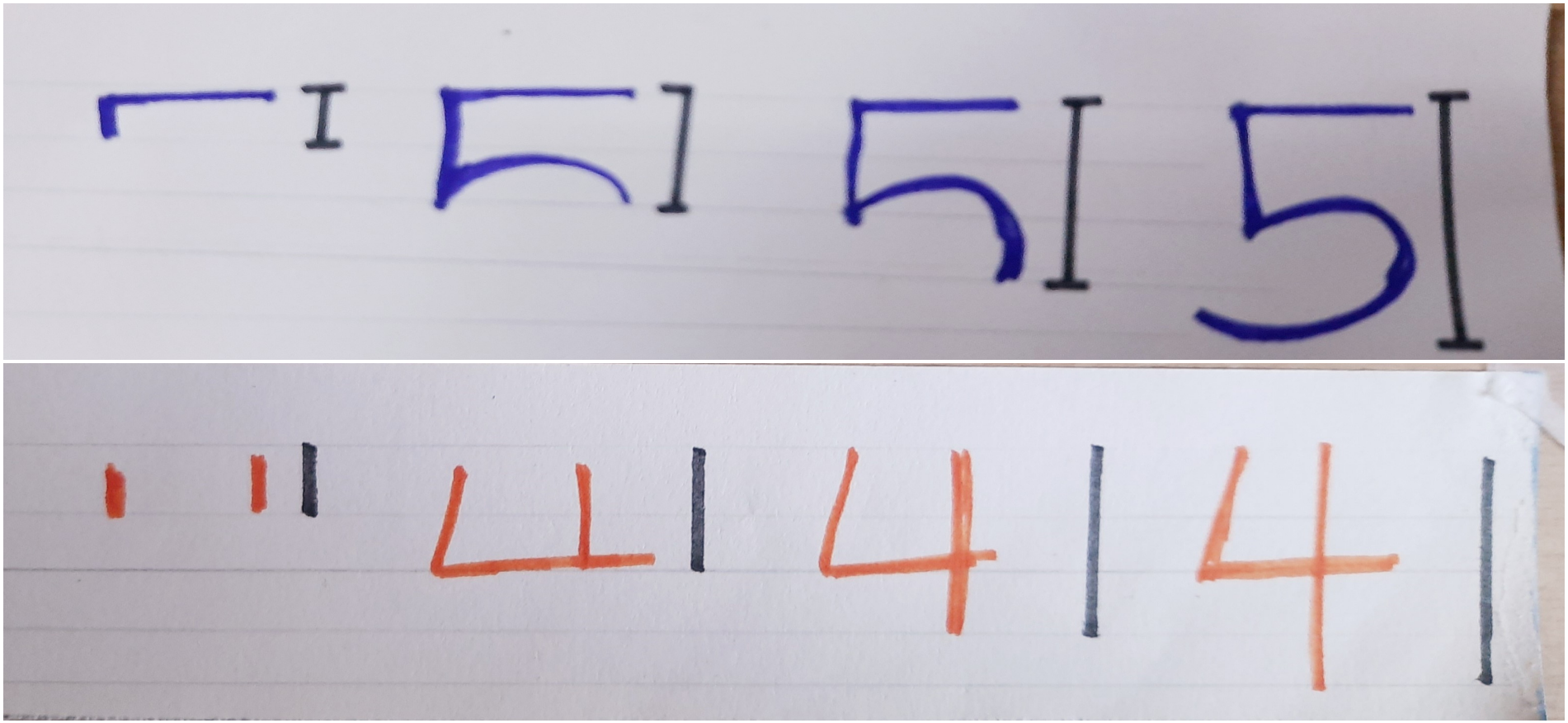
Grayscale filtration alone does not help in distinguishing between digits in the same homotopy class.



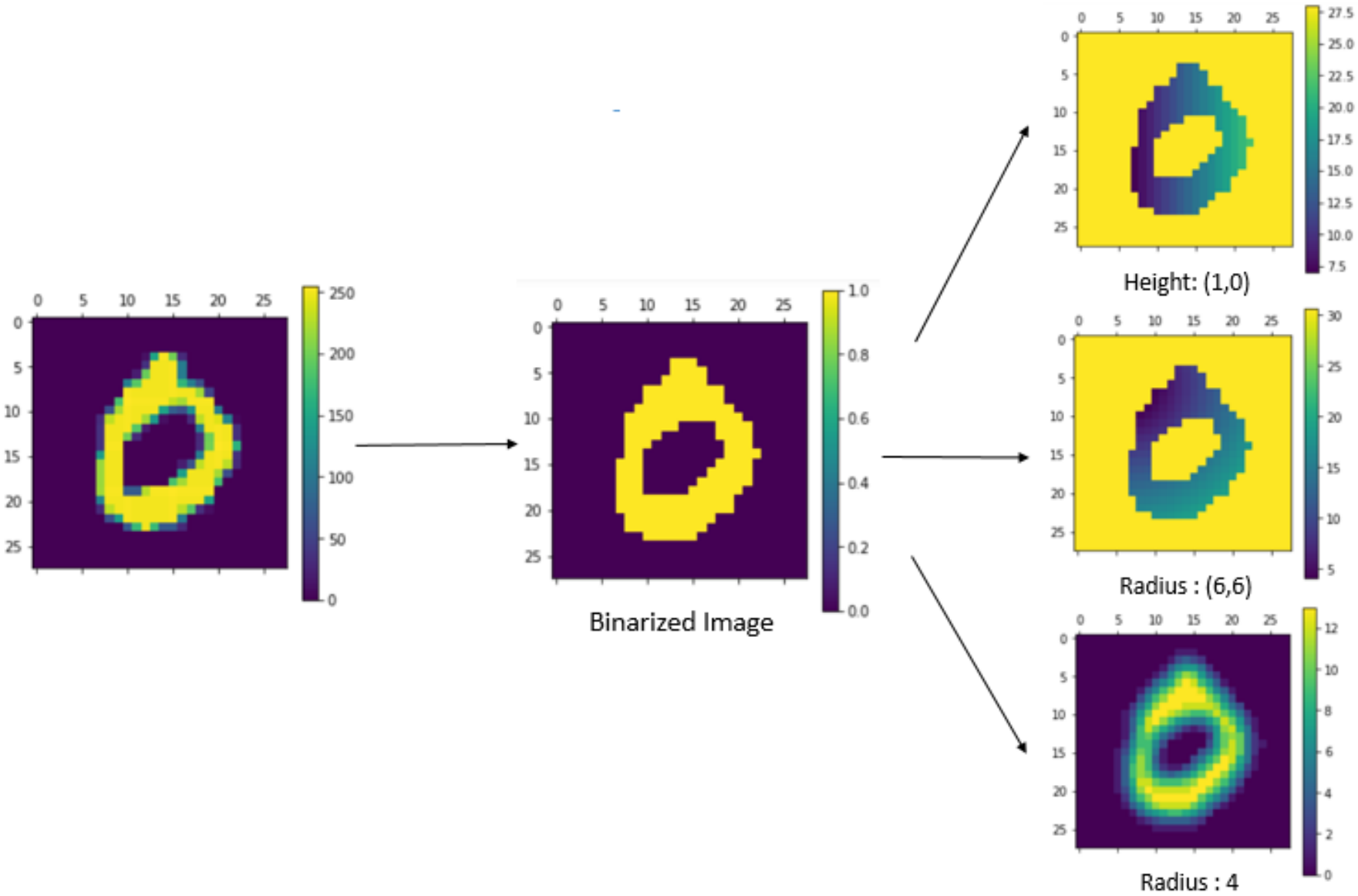
Persistence diagrams corresponding to super-level set cubical complexes.

Topological Pipeline : Filtrations

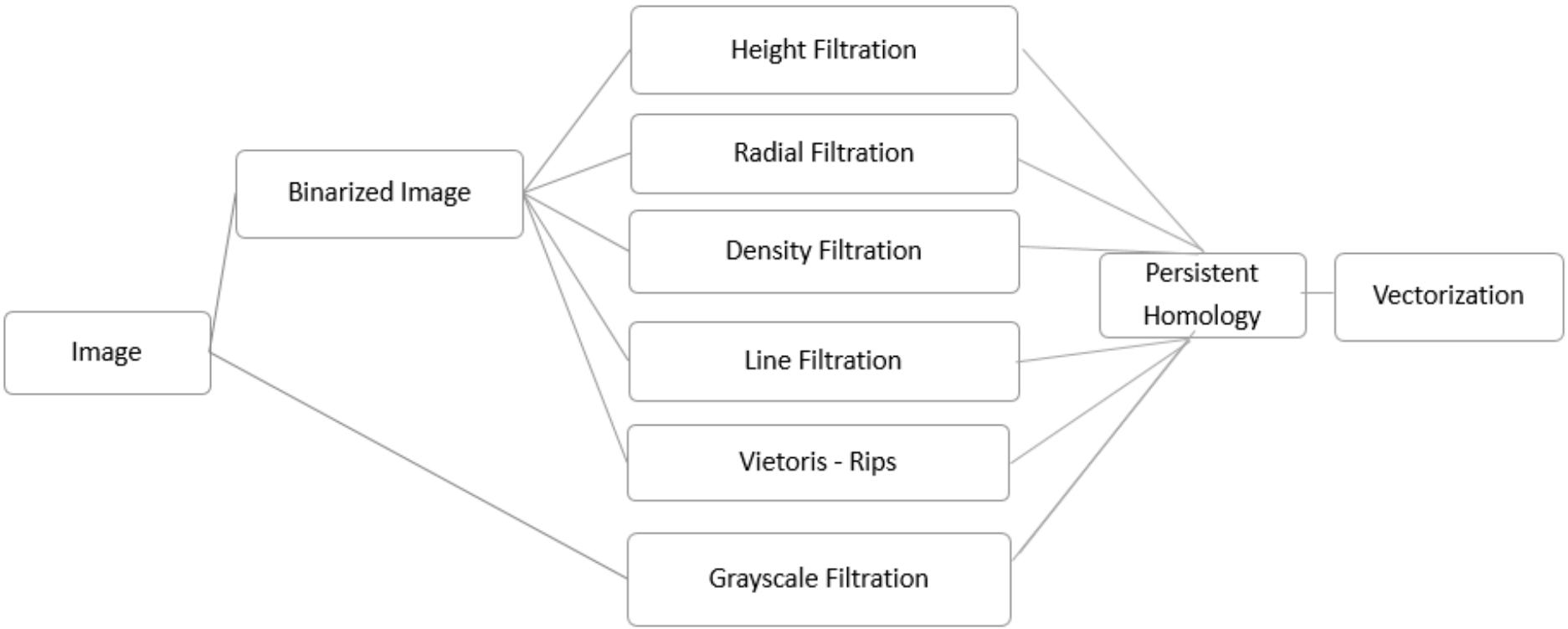
Other filtrations which take into account how the digits are built help distinguish between digits of the same homotopy type.



Topological Pipeline: Filtrations

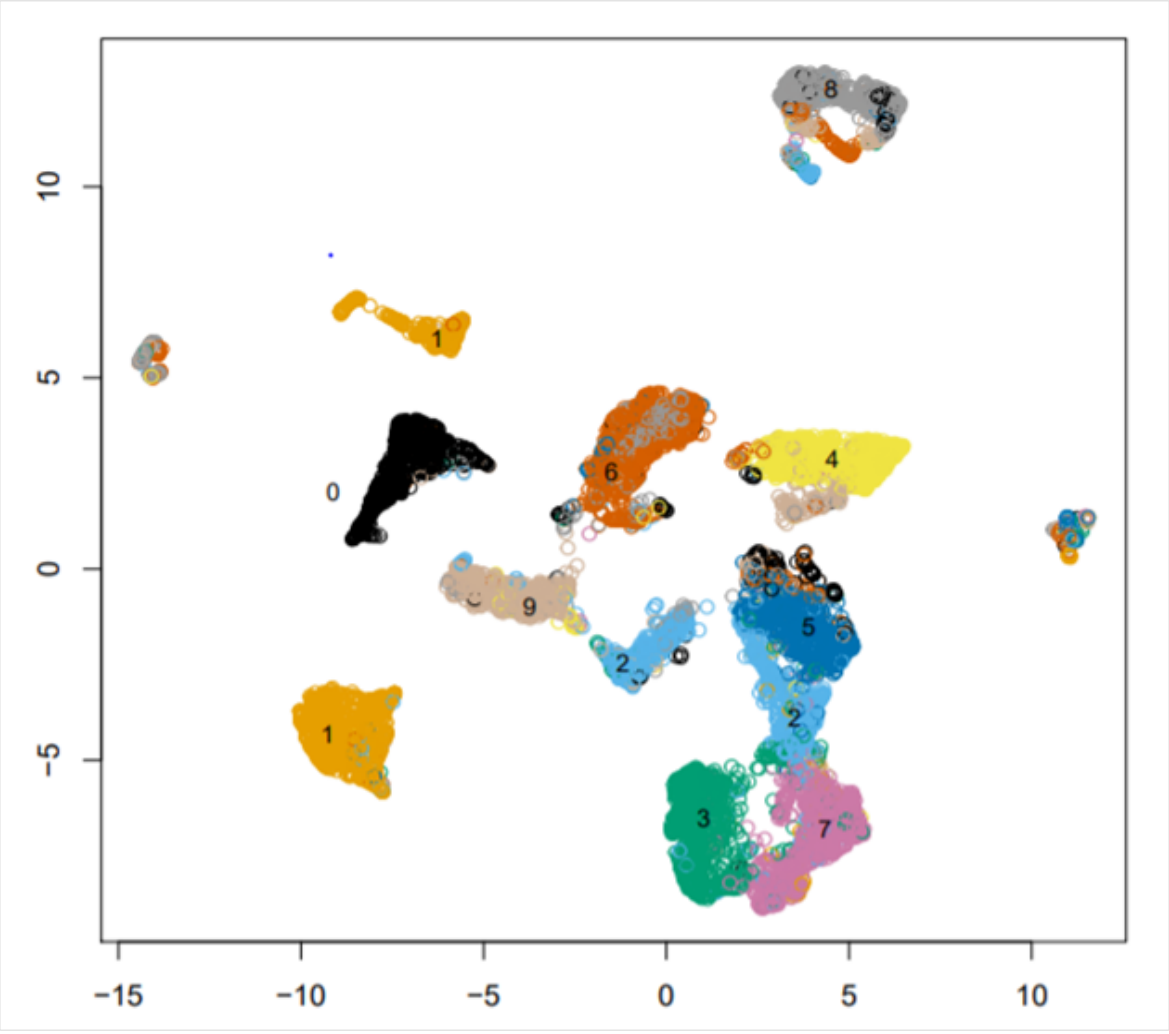


Topological Pipeline



Visualizing the Pipeline Data

2-dimensional projection of 52-dimensional pipeline output (persistent entropy vectorisation) using UMAP.

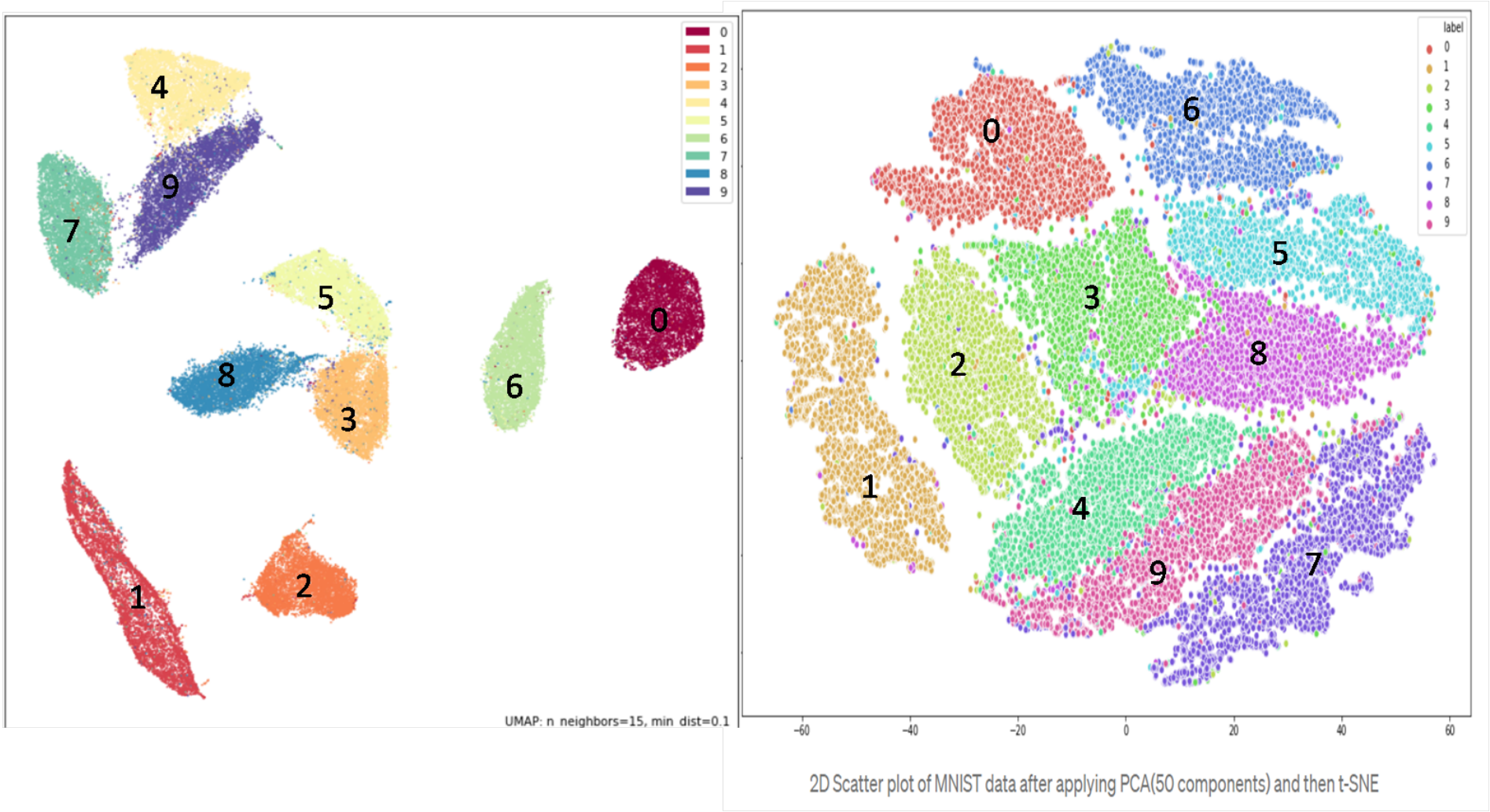


Visualizing the Pipeline Data

The two clusters each of digits '1' and '2' represent different ways in which these digit are written.

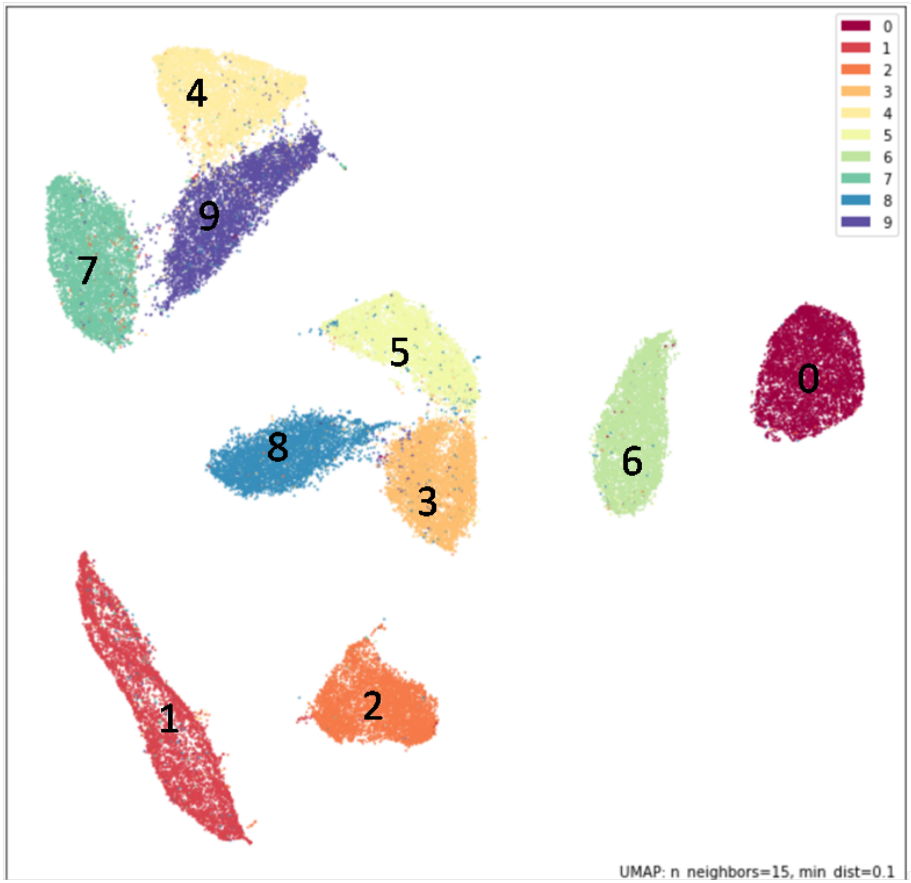
The MNIST Data

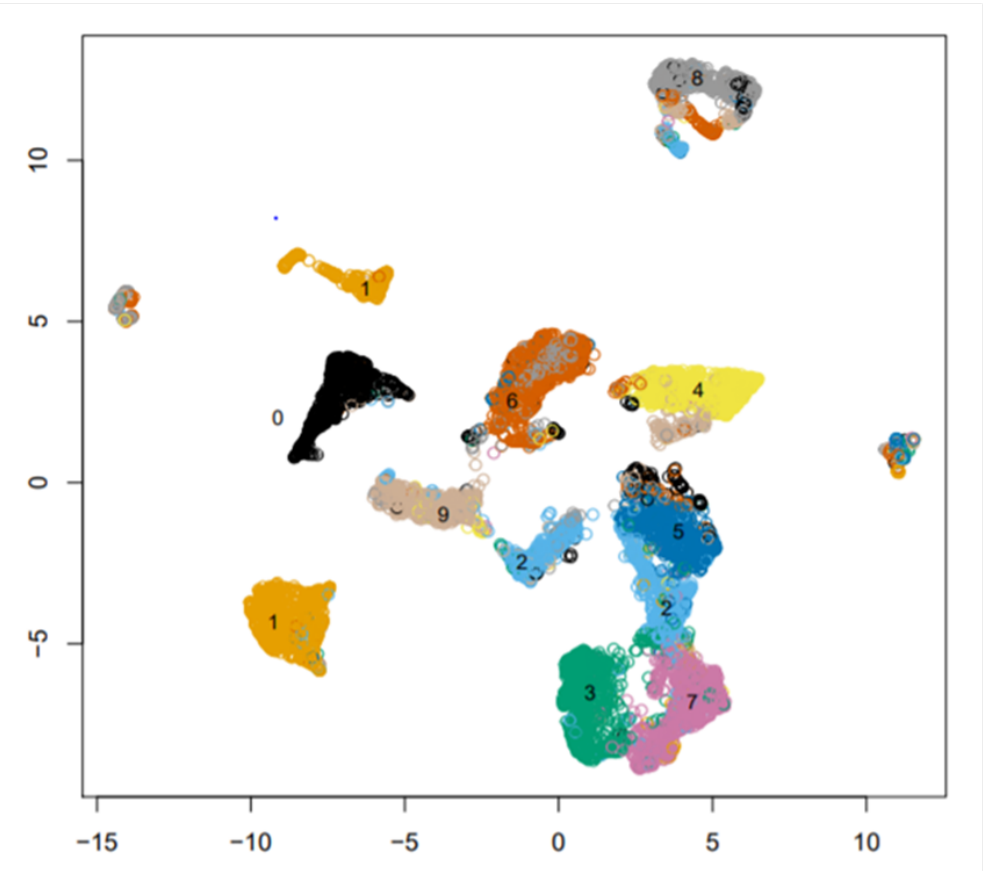
Reference 2-dimensional projections generated by considering 784 pixel values as a vector.



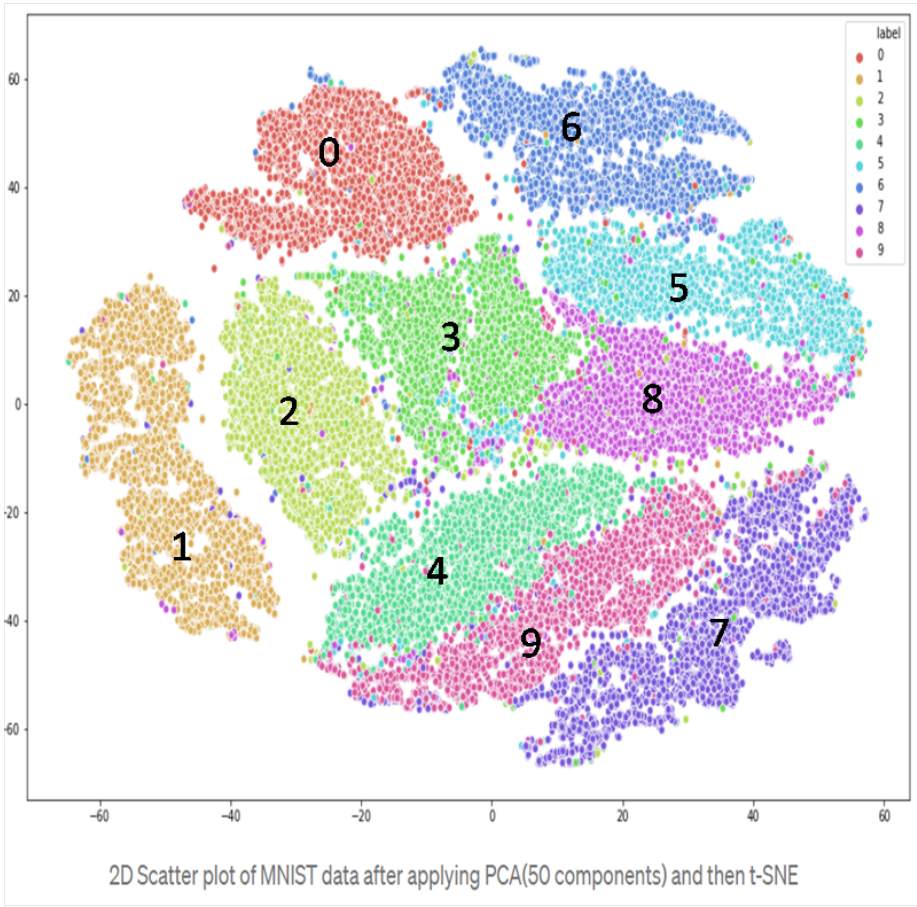
(Left) PCA followed by t-SNE ; (Right) UMAP

Visualization comparison

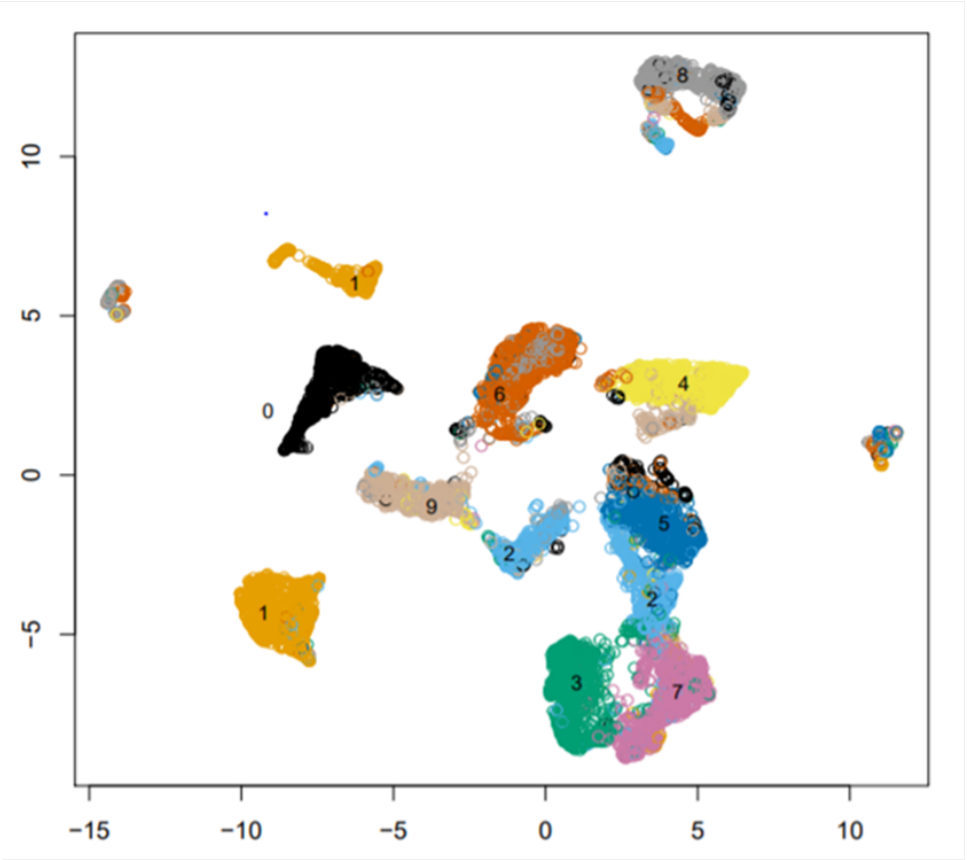




Visualization comparison



2D Scatter plot of MNIST data after applying PCA(50 components) and then t-SNE



Show the interactive Bokeh plot.

Classification of MNIST

Classification using Random Forest Classifier: Number of trees = 1000

Dimension	Description	Accuracy
703	Reference classifier	96.3%
52	MNIST Pipeline with persistent entropy	96.48%
202	MNIST Pipeline using all 4 vectorization	97.16%

For reference classifier, all the pixel values except those that are 0 for all images were considered as a vector.

S.No	Binarisation	Dimension	Filtrations	Vectorisation	Accuracy
1	0.2	50	Height, Radial, Density, Line	Persistent Landscape	94.92
2	0.4	52	Height, Radial, Density, Line, V-R	Entropy	96.15
3	0.3	52	Height, Radial, Density, Line, V-R	Entropy	96.21
4	0.2	52	Height, Radial, Density, Line, V-R	Entropy	96.48
5	0.2	202	Height, Radial, Density, Line, V-R	All Vectorisations	97.16

Outline

1. Classification using TDA
2. The mapper algorithm

Mapper

G. Singh, F. Mémoli and G. Carlsson, Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object Recognition. Point Based Graphics 2007, Prague.

Instead of trying to determine the shape of the entire data set, Mapper's goal is to compress the data into a form that can be easily visualized.

Mapper

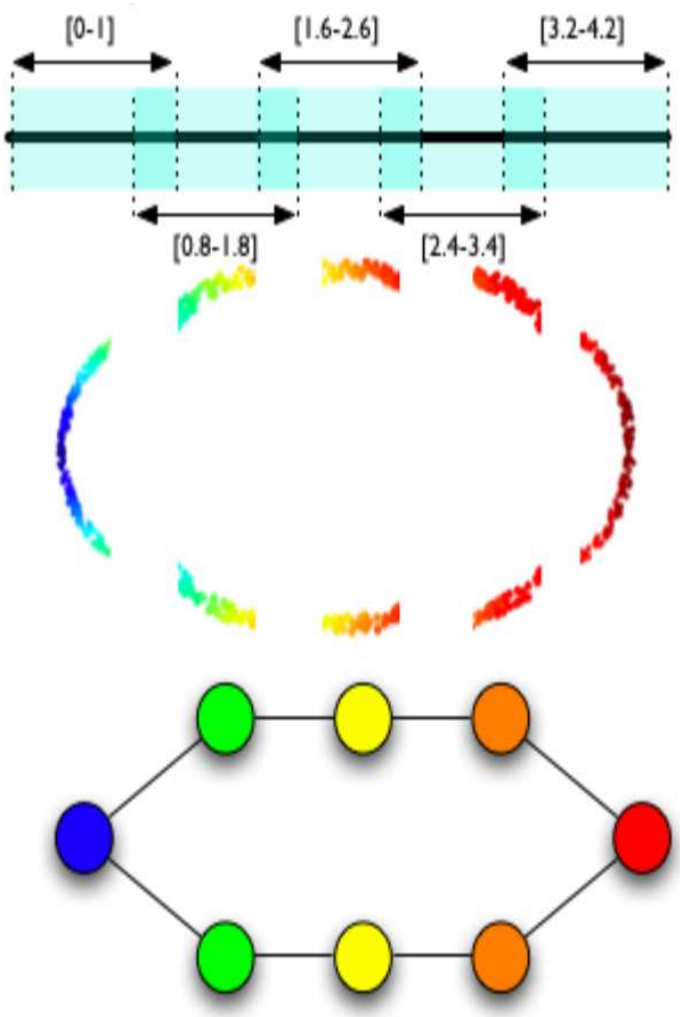
G. Singh, F. Mémoli and G. Carlsson, Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object Recognition. Point Based Graphics 2007, Prague.

Instead of trying to determine the shape of the entire data set, Mapper's goal is to compress the data into a form that can be easily visualized.

Ayasdi

Singh and Carlsson co-founded Ayasdi, a leading AI solutions provider based on TDA. All the examples/ images are from the Ayasdi blog.

Morse theoretic idea



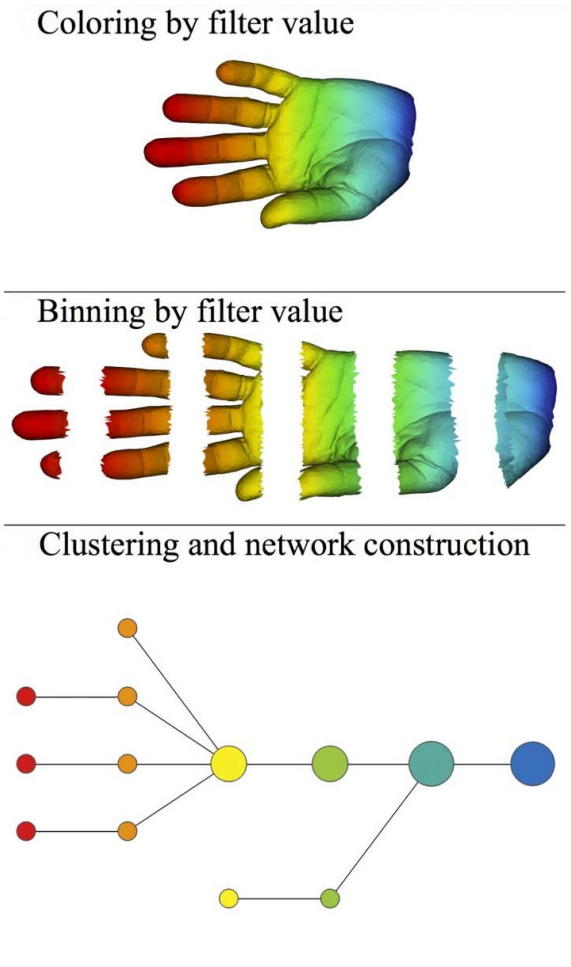


Image credits: previous paper.

Working of mapper

- Input

Working of mapper

- Input
  1. A point cloud X.

Working of mapper

- Input
  1. A point cloud X.
  2. A real valued **filter function** $f : X \to \mathbb{R}$.

Working of mapper

- Input
  1. A point cloud X.
  2. A real valued **filter function** $f : X \to \mathbb{R}$.
  3. A finite cover $\{\mathcal{U}_i\}$ of $\mathbb{R}$.

Working of mapper

- Input
  1. A point cloud X.
  2. A real valued **filter function** $f : X \to \mathbb{R}$.
  3. A finite cover $\{\mathcal{U}_i\}$ of $\mathbb{R}$.
  4. A choice of clustering algorithm.

Working of mapper

- Input
  1. A point cloud X.
  2. A real valued **filter function** $f : X \to \mathbb{R}$.
  3. A finite cover $\{\mathcal{U}_i\}$ of $\mathbb{R}$.
  4. A choice of clustering algorithm.
- Algorithm

Working of mapper

- Input
  1. A point cloud X.
  2. A real valued **filter function** $f : X \to \mathbb{R}$.
  3. A finite cover $\{\mathcal{U}_i\}$ of $\mathbb{R}$.
  4. A choice of clustering algorithm.
- Algorithm
  1. For each i break $X_i := f^{-1}(U_i)$ into clusters.

Working of mapper

- Input
  1. A point cloud X.
  2. A real valued **filter function** $f : X \to \mathbb{R}$.
  3. A finite cover $\{\mathcal{U}_i\}$ of $\mathbb{R}$.
  4. A choice of clustering algorithm.
- Algorithm
  1. For each i break $X_i := f^{-1}(U_i)$ into clusters.
  2. For each i and each cluster c of X_i, there is a node (i, c).

Working of mapper

- Input
  1. A point cloud X.
  2. A real valued **filter function** $f : X \to \mathbb{R}$.
  3. A finite cover $\{\mathcal{U}_i\}$ of $\mathbb{R}$.
  4. A choice of clustering algorithm.
- Algorithm
  1. For each i break $X_i := f^{-1}(U_i)$ into clusters.
  2. For each i and each cluster c of X_i, there is a node (i, c).
  3. An edge between (i_1, c_1) and (i_2, c_2) if c_1, c_2 overlap.

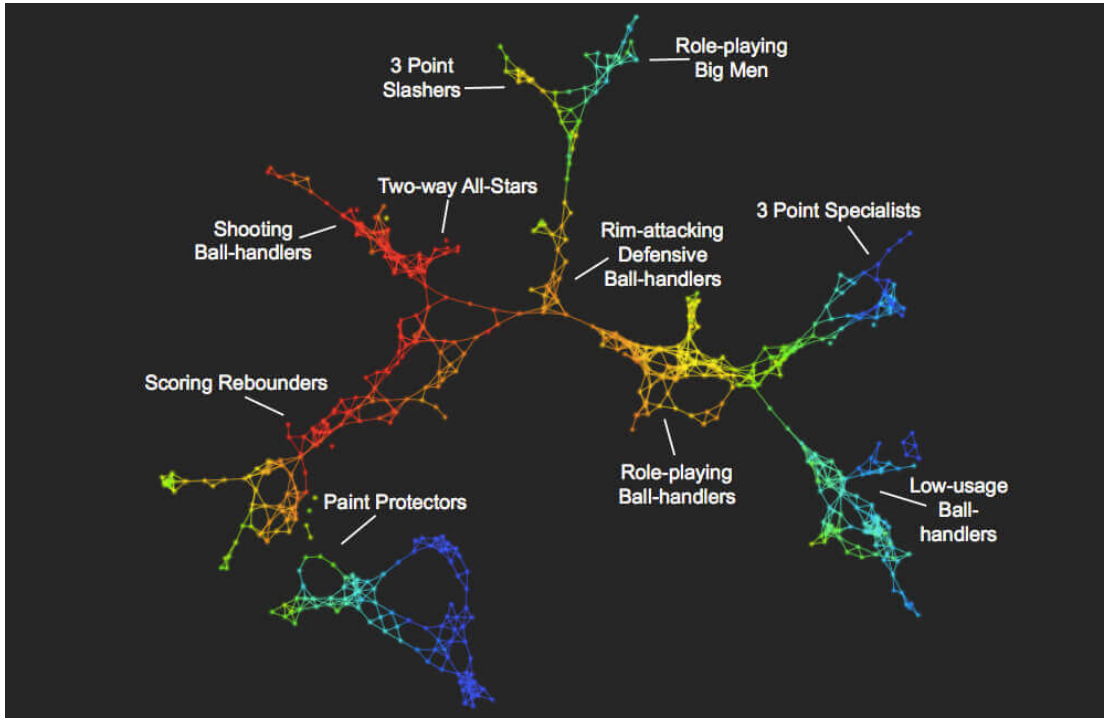
Working of mapper

- Input
  1. A point cloud X.
  2. A real valued **filter function** $f : X \to \mathbb{R}$.
  3. A finite cover $\{\mathcal{U}_i\}$ of $\mathbb{R}$.
  4. A choice of clustering algorithm.
- Algorithm
  1. For each i break $X_i := f^{-1}(U_i)$ into clusters.
  2. For each i and each cluster c of X_i, there is a node (i, c).
  3. An edge between (i_1, c_1) and (i_2, c_2) if c_1, c_2 overlap.
  4. In general, put an n-simplex if $c_0, \ldots, c_n$ intersect.

Basketball analysis

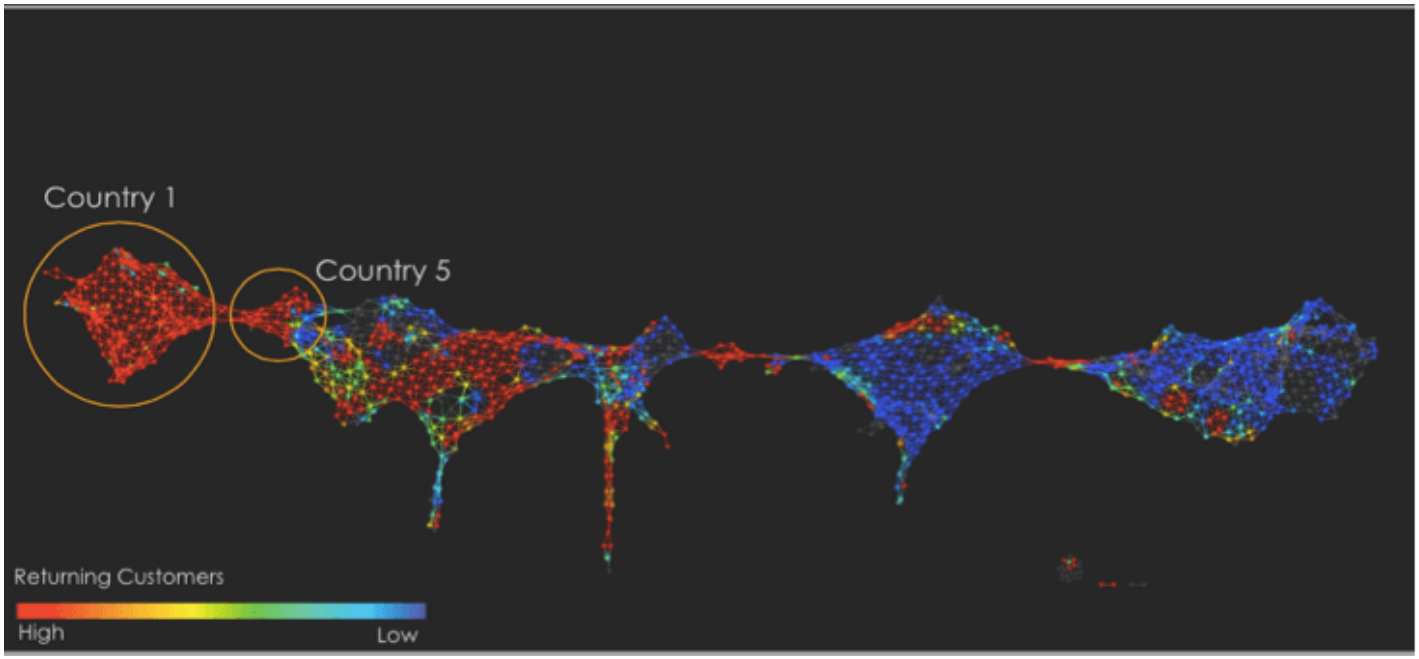
Muthu Alagappan, Sloan Sports Conference, 2012. Playing positions 13 not 5.

1. Method: data from NBA players (2010 - 2011).
2. Stats: 7 basic - rebounds, assists, turnovers, steals, points scored etc.
3. Filter function: An $\mathbb{R}^2$-valued function based on SVD.



Returning customers

The network below is constructed from purchases made by a retailer's customers. The nodes are colored by the percentage of customers who are returning. Built using only the customer's first purchase information. A clear localization of the purchase patterns associated to different countries is seen. Prediction about the likelihood of a given customer becoming a return customer is possible.



Type 2 diabetes subgroups

2551 patients with Type II diabetes. For each patient, 73 clinical variables and 7097 genetic variables. Very clear clustering into three new subtypes.