The Power of Topological Data Analysis for Machine Learning

MATH 471 Topology

Charles Zhang

Introduction: Cubical Complex

An elementary interval I_a is a subset of $\mathbb{R}$ of the form $[a, a+1]$ or $[a, a] = \{a\}$ for some $a \in \mathbb{R}$. These two types are called respectively **non-degenerate** and **degenerate**. To a non-degenerate elementary interval we assign two degenerate elementary intervals

$$d^+ I_a = [a+1, a+1] \quad \text{and} \quad d^- I_a = [a, a]$$

An elementary cube is a subset of the form

$$I_{a_1} \times \cdots \times I_{a_N} \subset \mathbb{R}^N$$

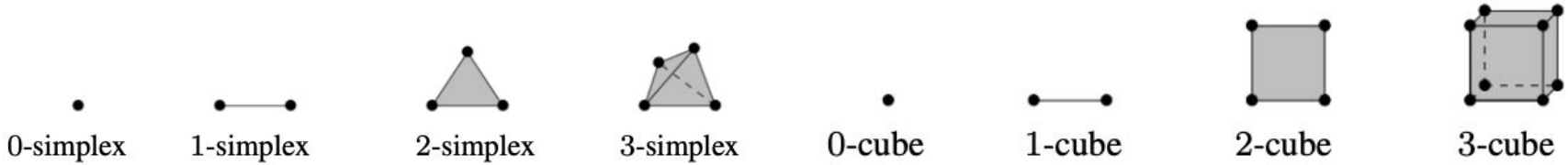
where each I_{a_i} is an elementary interval. We refer to the total number of its nondegenerate factors $I_{a_{k_1}}, \ldots, I_{a_{kn}}$ as its dimension and, assuming

$$a_{k_1} < \cdots < a_{k_n}$$

we define for $i = 1, \ldots, n$ the following two elementary cubes

$$d_i^{\pm} I^N = I_{a_1} \times \cdots \times d^{\pm} I_{a_{k_i}} \times \cdots \times I_{a_N}$$

A **cubical complex** is a finite set of elementary cubes of $\mathbb{R}^N$, and a subcomplex of X is a cubical complex whose elementary cubes are also in X. This is quite similar with simplicial complex, and more generally, every face of a cube in cubical complex is also in the cubical complex; and the intersection of any two cubes of cubical complex is either empty or a common face.



Introduction

Given a topological space $X \subset \mathbf{R}^n$ in terms of a cubical complex, we define $C_q(X)$ to be the free abelian group generated by all the q -dimensional elementary cubes in X. Thus the elements of this group, called q -chains are formal linear combinations of elementary q -cubes. We put

$$C_q(X) = 0 \text{ if } q < 0 \text{ or } q > n$$

The **boundary map** $\partial_q : C_q(X) \to C_{q-1}(X)$ is a group homomorphism defined on every elementary q -cube as an altemating sum of its $(q-1)$ -dimensional faces and extended by linearity to all q -chains. Note that $\partial_0 = 0$ since $C_{-1}(X) = 0$. The boundary map satisfies the very important property

$$\partial^2 = 0$$

i.e., $\partial_q \circ \partial_{q+1} = 0$ for all q. As for a simplicial complex of dimension n, the algebraic information extracted from the given cubical structure can be expressed as a finitely generated free chain complex (C, ∂)

$$0 \xrightarrow{\partial_{n+1}} C_n(X) \xrightarrow{\partial_n} \dots \xrightarrow{\partial_{k+2}} C_{k+1}(X) \xrightarrow{\partial_{k+1}} C_k(X)$$
$$\xrightarrow{\partial_k} C_{k-1}(X) \xrightarrow{\partial_{k-1}} \dots C_0(X) \xrightarrow{\partial_0} 0$$

A chain $z \in C_q(X)$ is called cycle or, more precisely, q -cycle if $\partial_q z = 0$. A chain $z \in C_q(X)$ is called boundary if there exists $c \in C_{q+1}(X)$ such that $\partial_{q+1} c = z$. The set of all q -cycles is the subgroup $Z_q := \operatorname{Ker} \partial_q$ of $C_q(X)$ while the set of all boundaries is the subgroup $B_q := \operatorname{Im} \partial_{q+1}$. $\partial^2 = 0$ implies that $\operatorname{Im} \partial_{q+1} \subset \operatorname{Ker} \partial_q$ and hence the quotient group $H_q(X) := \operatorname{Ker} \partial_q / \operatorname{Im} \partial_{q+1}$, called the $q-$ th cubical homology group of X is well defined and has the same interpretation as the simplicial homology groups. Moreover, one can remark that the cubical set X considered here can be triangulated and its simplicial homology is isomorphic to the cubical homology of X defined above. Thus, the two theories are equivalent for cubical sets.

The homology type of X (also of (C, ∂)) is by definition the sequence of abelian groups

$$H(X) = H(C) := \left\{ H_q(X) \right\}, \quad q = 0, 1, 2, \dots$$

Introduction

Persistent homology is defined for filtrations of cubical complexes as well. Hence, we now introduce a way to represent images as cubical complexes and then explain how we build filtrations of cubical complexes from binary images. A d -dimensional image is a map $\mathcal{I} : I \subseteq \mathbb{Z}^d \longrightarrow \mathbb{R}$. An element $v \in I$ is called a voxel (or pixel when $d = 2$) and the value $\mathcal{I}(v)$ is called its intensity or greyscale value. In the case of binary images, which are made of only black and white voxels, we consider a map $\mathcal{B} : I \subseteq \mathbb{Z}^d \longrightarrow \{0, 1\}$. In a slight abuse of terminology, we call the subset $I \subseteq \mathbb{Z}^d$ an image.

There are several ways to represent images as cubical complexes. Here, we choose the approach in which a voxel is represented by a d -cube and all of its faces (adjacent lower-dimensional cubes) are added. We get a function on the resulting cubical complex K by extending the values of the voxels to all the cubes σ in K in the following way:

$$\mathcal{I}'(\sigma) := \min_{\sigma \text{ face of } \tau} \mathcal{I}(\tau)$$

A grayscale image comes with a natural filtration embedded in the grayscale values of its pixels. Let K be the cubical complex built from the image I. Let

$$K_i := \left\{\sigma \in K \mid \mathcal{I}'(\sigma) \leq i\right\}$$

be the i -th sublevel set of K. The set $\{K_i\}_{i \in \mathrm{Im}(I)}$ defines a filtration of cubical complexes, indexed by the value of the grayscale function $\mathcal{I}$. The steps we factor through to obtain a filtration from a grayscale image are then:

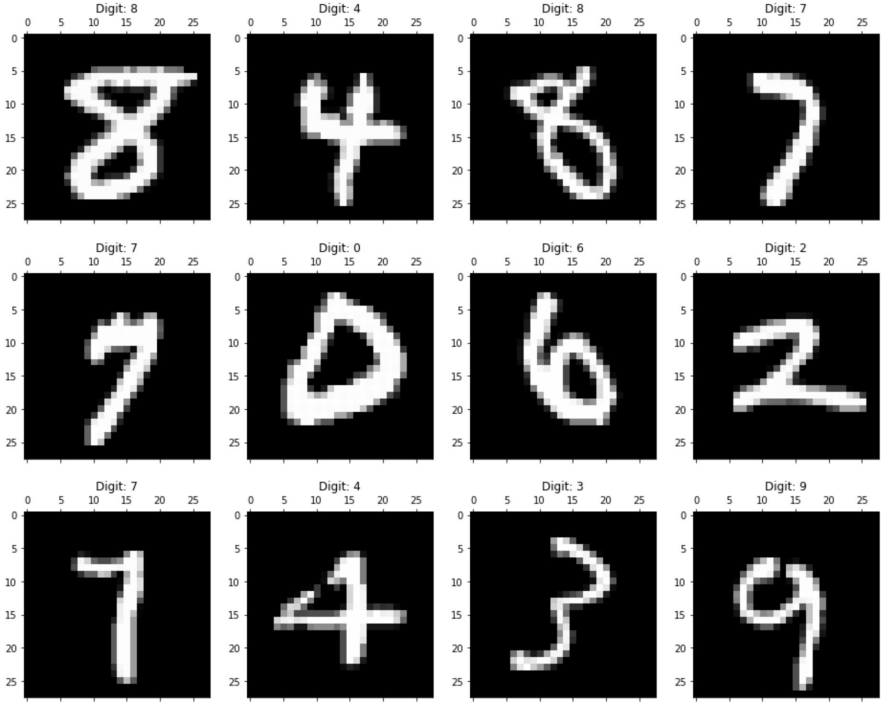
Image $\rightarrow$ Cubical complex $\rightarrow$ Sublevel sets $\rightarrow$ Filtration.

Introduction

Machine learning: a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention.

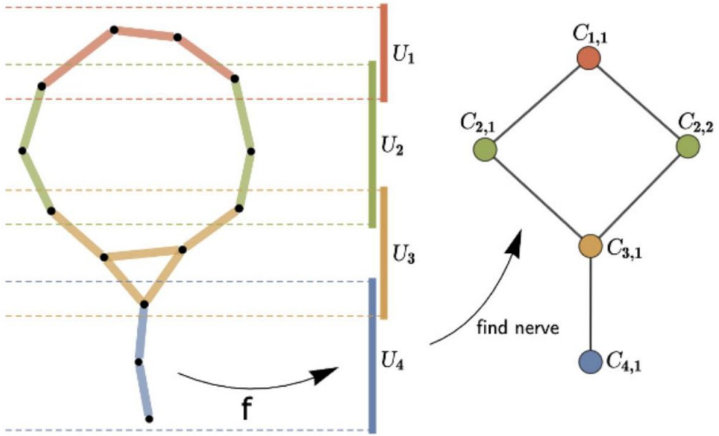
Supervised Learning: the machine learning task of learning a function that maps an input to an output based on example input-output pairs. It infers a function from labeled training data consisting of a set of training examples.

Dataset: MNIST consists of 70,000 images of handwritten digits of size 28 × 28

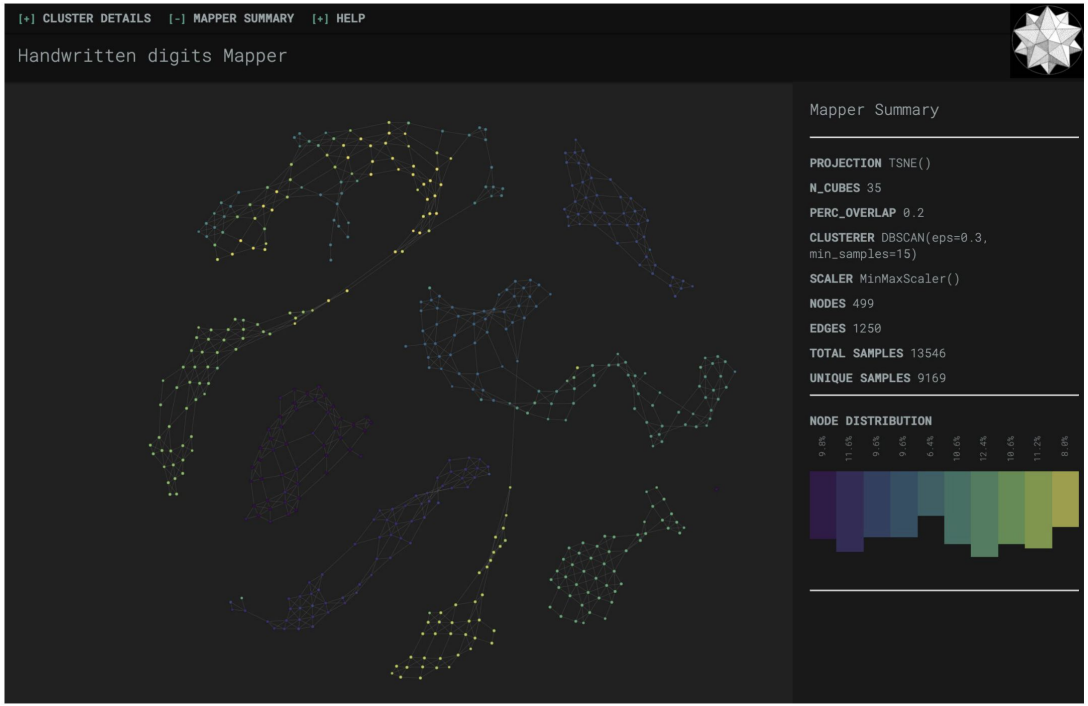


Shape Analysis — Mapper

1. Map to a lower-dimensional space using a filter function f, or lens. Common choices for the filter function include projection onto one or more axes via principal component analysis (PCA) or density-based methods.
2. Construct a cover $(U_i)_{i \in I}$ of the projected space typically in the form of a set of overlapping intervals which have constant length.
3. For each interval U_i cluster the points in the preimage $f^{-1}(U_i)$ into sets $C_{i,1}, \ldots, C_{i,k_i}$
4. Construct the graph whose vertices are the cluster sets and an edge exists between two vertices if two clusters share some points in common.

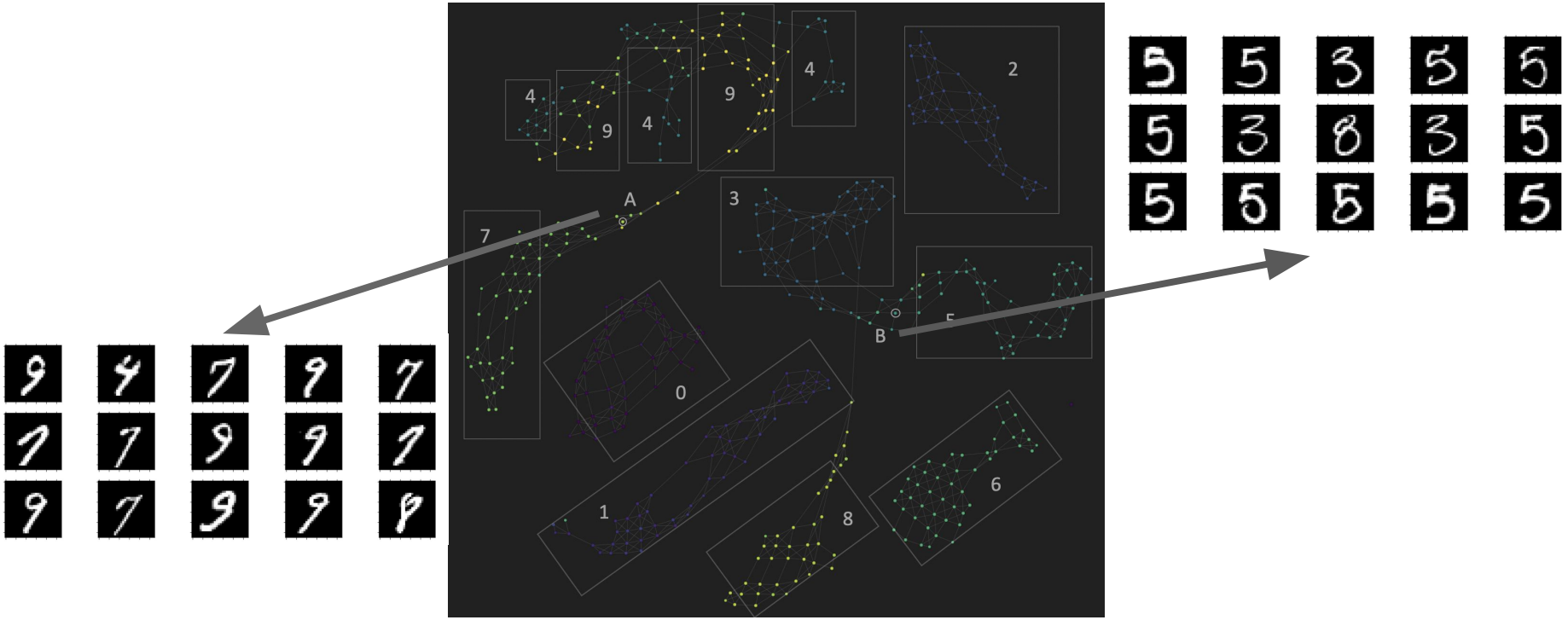


Shape Analysis — Mapper

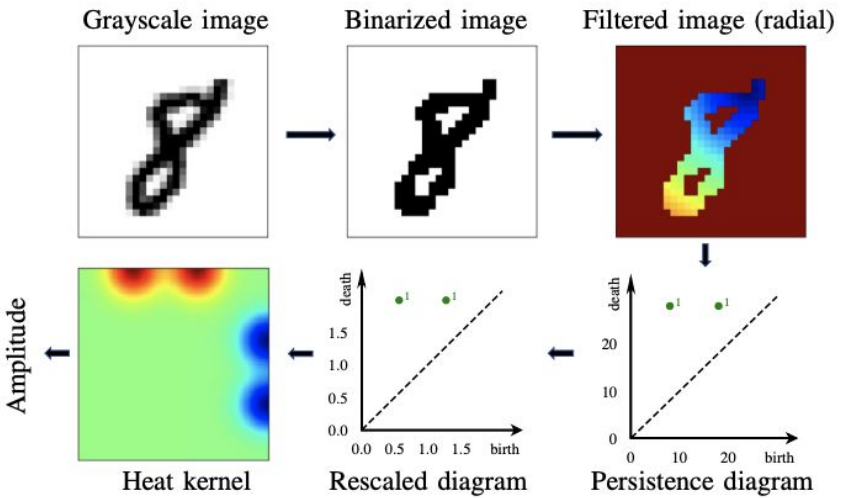


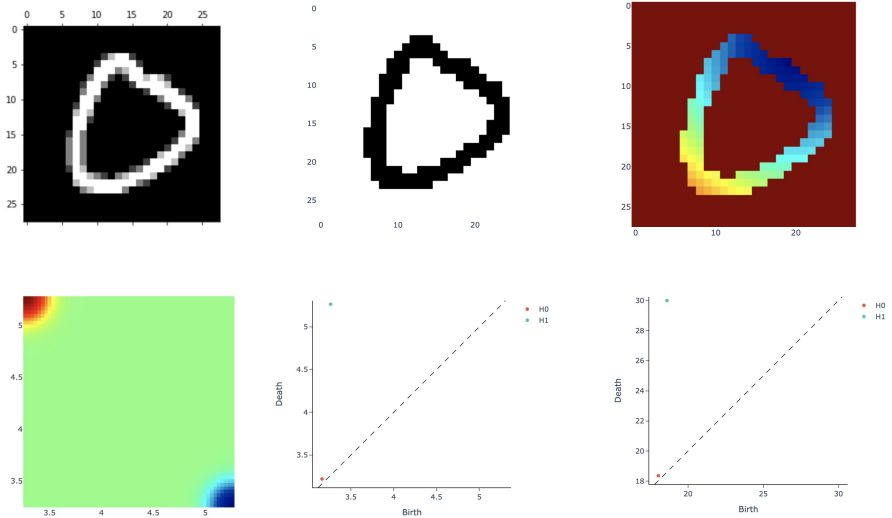
[Interactive page]

Shape Analysis — Mapper



Feature Extraction





Feature Extraction: Filtration

1) **Binary filtration**: The binary filtration consists of computing the persistence diagram straight from the binary image, i.e. by considering the binary values of the pixels as a twolevel filtration. It is related to computing the homology of the image.

2) **Height filtration**: The height filtration is inspired by Morse theory and by the Persistent Homology Transform [8]. For cubical complexes, we define the height filtration $\mathcal{H} : I \longrightarrow \mathbb{R}$ of a d -dimensional binary image I by choosing a direction $v \in \mathbb{R}^d$ of norm 1 and defining new values on all the voxels of value 1 as follows: if $p \in I$ is such that $\mathcal{B}(p) = 1$, then one assigns a new value $\mathcal{H}(p) :=< p, v >$ the distance of p to the hyperplane defined by v. If $\mathcal{B}(p) = 0$ then $\mathcal{H}(p) := H_\infty$, where H_∞ is the filtration value of the pixel that is the farthest away from the hyperplane.

3) **Radial filtration**: The radial filtration $\mathcal{R}$ of I with center $c \in I$, inspired from [9], is defined by assigning to a voxel p the value corresponding to its distance to the center

$$\mathcal{R}(p) := \begin{cases} \|c - p\|_2 & \text{if } \mathcal{B}(p) = 1 \\ \mathcal{R}_\infty & \text{if } \mathcal{B}(p) = 0 \end{cases}$$

where $\mathcal{R}_\infty$ is the distance of the pixel that is the farthest away from the center.

4) **Density filtration**: The density filtration gives each voxel a value depending on the number of neighbors it has at a certain distance. For a parameter r, the radius of the ball we want to consider, the density filtration is:

$$\mathcal{D}_e(p) := \#\{v \in I, \mathcal{B}(v) = 1 \text{ and } \|p - v\| \leq r\}$$

where the norm can be any norm on $\mathbb{R}^d$, but we choose the $L1$ -norm in our implementation.

5) **Dilation filtration**: The dilation filtration of I defines a new grayscale image $\mathcal{D} : I \longrightarrow \mathbb{R}$ as follows: a vertex p in I is assigned the smallest distance to a vertex of value 1 in I :

$$\mathcal{D}(p) := \min \{\|p - v\|_1, \mathcal{B}(v) = 1\}$$

6) **Erosion filtration**: The erosion filtration is the inverse of the dilation filtration. Instead of dilating the object it erodes it. To obtain the erosion filtration, one applies the dilation filtration to the inverse image, where 0 and 1 are switched. Note that in the case of extended persistence [8], the two filtrations return the same information by duality, but it is not the case here as we consider only standard persistence.

7) **Signed distance filtration**: The signed distance filtration is a combination of the erosion and dilation filtrations that returns both positive and negative values. It takes positive values on the 1 -valued voxels by taking the distance to the boundary and negative values on the 0 -valued voxels, by attributing them the negative distance to the same boundary.

Feature Extraction: Amplitude

1) **Betti curves**: The Betti curve $B_n : I \longrightarrow \mathbb{R}$ of a barcode $D = \left\{\left(b_j, d_j\right)\right\}_{j \in I}$ is the function that return for each step $i \in I$, the number of bars $\left(b_j, d_j\right)$ that contains i

$$i \mapsto \#\left\{\left(b_j, d_j\right), i \in \left(b_j, d_j\right)\right\}$$

2) **Persistence landscapes**: Introduced in [11], the k -th persistence landscape of a barcode $\{(b_i, d_i)\}_{i=1}^n$ is the function $\lambda_k : \mathbb{R} \longrightarrow [0, \infty)$ where $\lambda_k(x)$ is the k -th largest value of $\left\{f_{(b_i,d_i)}(x)\right\}_{i=1}^n$, with

$$f_{(b,d)}(x) = \begin{cases} 0 & \text{if } x \notin (b, d) \\ x - b & \text{if } x \in \left(b, \frac{b+d}{2}\right) \\ -x + d & \text{if } x \in \left(\frac{b+d}{2}, d\right) \end{cases}$$

The parameter k is called the layer. Here we consider curves obtained by setting $k = 1$ and $k \in \{1, 2\}$.

3) **Heat kernel**: In [10], the authors introduce a kernel by placing Gaussians of standard deviation σ over every point of the persistence diagram and a negative Gaussian of the same standard deviation in the mirror image of the points across the diagonal. The output of this operation is a real-valued function on $\mathbb{R}^2$. In this work, we consider two possible values for σ, 10 and 15 (in the unit of discrete filtration values).

4) **Wasserstein amplitude**: Based on the Wasserstein distance, the Wasserstein amplitude of order p is the Lp norm of the vector of point distances to the diagonal:

$$A_W = \frac{\sqrt{2}}{2}\left(\sum_i (d_i - bi)^p\right)^{\frac{1}{p}}$$

In this project, we use $p = 1, 2$.

5) **Bottleneck distance**: When we let p go to ∞ in the definition of the Wasserstein amplitude, we obtain the Bottleneck amplitude:

$$A_B = \frac{\sqrt{2}}{2} \sup_i (d_i - b_i)$$

6) **Persistent entropy**: The persistent entropy of a barcode, introduced in [11], is a real number extracted by taking the Shannon entropy of the persistence (lifetime) of all cycles:

$$PE(D) = \sum_{i=1}^{n} \frac{l_i}{L(B)} \log\left(\frac{l_i}{L(B)}\right)$$

where $l_i := d_i - b_i$ and $L(B) := l_1 +, \ldots + l_n$ is the sum of all the persistences.

Experiment

Random forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean/average prediction (regression) of the individual trees.

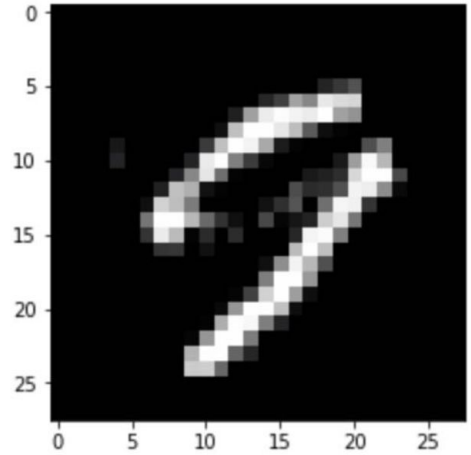
Accuracy of this Project: 0.97

With 476 feature vectors

Accuracy for Same ML Classifier: 0.94 - 0.96

With 784 feature vectors

```
Predicted 9 as 4
<matplotlib.image.AxesImage at 0x7f9b1ea58198>
```



Conclusion and Future Steps

- Understand more about the nature, fundamental structure and underlying relationships of the data by Mapper
- the clusters Mapper algorithm has produced can be considered as input to a machine learning algorithm helping with select features that best discriminate data to classify images.
- Combined a wide range of different TDA techniques for images based on different filtrations and diagram features
- Less features while maintaining high accuracy for classification tasks
- Can be conduct similarly to other novel datasets
- Can be combined with state-of-the-art deep learning algorithms to improve accuracy

THANK YOU