Topological Data Analysis (TDA) for Image Classification

Virtual

River Reger
river.reger@gmail.com

Technical Presentation
2021 November 29

Overview

Mathematics of Machine Learning: Problem Statement

Definition

Given an ideal function $f : X \to Y$, which associates **data** in $X \neq \phi$ with **measurements** in $Y \neq \phi$, the problem of **supervised learning** is to **learn** a **target function** $\hat{f} : X \to Y$, which approximates f, using a finite sample $S = \{(x_i, f(x_i))\}_{i=1}^{n}$, a **training set**.

- When Y is finite, we say that this is a **supervised classification problem**, otherwise, we say this is a **supervised regression problem**.
- In practice we often encode Y to be isomorphic to $\mathbb{R}^n$ or $\mathbb{Z}_n$ for some $n \in \mathbb{N}$. (e.g., if the possible measurements are "red" and "not red", we encode this as $Y = \{0, 1\}$).

Mathematics of Machine Learning: Generalization

- There are infinitely many $\hat{f}$ for which $\hat{f}(x_i) = f(x_i)$ for all $x_i \in S$.
- For the purpose of machine learning, we are more interested not in fitting $\hat{f}$ to S, but in generalizing $\hat{f}$ to X; i.e., approximating f on X.
- For this reason, we often partition S into two sets S_{train} and S_{test}.
- We first find $\hat{f}$ which performs well on S_{train}, then we validate this on S_{test}.
- This does not guarantee accuracy on X. It is an open problem of research to establish useful conditions under which training and test set performance will generalize to the entire set X.

Mathematics of Machine Learning: Hypothesis Space

- The **hypothesis space** for the supervised learning problem is $\mathcal{H} = \{f : X \to Y\}$.
- It is not computationally feasible to search over the space of all such functions for f, especially when the X is infinite or extremely large. Instead we choose a class of target functions $\hat{H}$.
- Once a class of target functions has been selected, a **loss function** $J : \hat{\mathcal{H}} \times \mathcal{H} \to \mathbb{R}$ is selected.
- A popular example of this is the mean squared error:

$$J(\hat{f}, f) = \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - \hat{f}(x_i))^2.$$

- In the case where $Y = \{0, 1\}$, a popular loss function is **logarithmic loss**:

$$J(f, \hat{f}) = -f(x_i) \log(\hat{f}(x_i)) + (1 - f(x_i)) \log(1 - \hat{f}(x_i)).$$

Mathematics of Machine Learning: Optimization

- The task of learning $\hat{f}$ is to minimize J.
- There is copious literature on the subject of optimization and loss functions in a machine learning context.
- One of the most popular methods of optimization for machine learning is Adam, or adaptive moment estimation.
- It is beyond the scope of this talk to delve into the details of Adam.
- The selection of an optimization algorithm can impact the performance of machine learning significantly.
- See details of Adam here: `https://arxiv.org/pdf/1412.6980.pdf`

Mathematics of Machine Learning: Pipeline

- Define the problem in terms of an ideal function f and a dataset S.
- Select a target function class; i.e., $\hat{H}$.
- Select a loss function J; e.g., mean squared error.
- Select an optimization method; e.g., Adam.
- Learn $\hat{f}$ by optimizing J over S (i.e., implement the machine learning architecture design).
- Analyze the results usually with statistics-based methods such as area under curve (AUC) and iterate on the design.

Note: if enough computational resources are available, hyperparameter search can be used to automate some of the iteration.

Traditional Assumptions of Data Analytics

- Qualitative information is required - we wish to classify data/datasets by describing global properties (i.e., features).
  - Loss/error functions are quantitative in nature.
- Metrics usually have no basis in physics (counterexample to this would be Physics-informed Neural Networks).
  - Optimization of machine learning algorithms is sensitive to metric choice (e.g., mean squared error).
- Coordinates are not usually natural (unlike state vector coordinates from physics).
  - The representation of data matters significantly, especially in terms of coordinates (normalized data vs. raw data).
- Preference of summaries over individual parameter selection.
  - Parameter selection and hyperparameter search look for optimal parameters, but this is computationally expensive.

Topological Data Analysis

- Qualitative information is required - we wish to classify data/datasets by describing global properties.
  - Topology captures global qualitative information via connectivity information about the underlying surface (e.g., manifold) that data resides within.
- Metrics usually have no basis in physics.
  - Less sensitivity to metric choice in topology since spatial information is not dependent on true distance, but relative placement.
- Coordinates are not usually natural.
  - Topology is coordinate free by definition. We place topological structure on a coordinate space, but that structure is not the significant factor.
- Preference of summaries over individual parameter selection.
  - The powerhouse of topological data analysis is persistent homology, which looks at all possible parameter choices as a summary versus endlessly searching a large hyperparameter space.

Spaces Under Study

Definition

Let $X \neq \phi$ and $d : X \times X \to [0, \infty)$. Then d is said to be a **metric** on X, and the pair (X, d) a **metric space** if for all $x, y, z \in X$:

1. $d(x, y) = 0$ iff $x = y$
2. $d(x, y) = d(y, x)$
3. $d(x, z) \leq d(x, y) + d(y, z)$

Definition

Let $X \neq \phi$ and $\mathcal{T} \subseteq X$. Then $\mathcal{T}$ is a **topology** on X, and the pair $(X, \mathcal{T})$ is said to be a **topological space**, if

1. $X, \phi \in \mathcal{T}$
2. $\mathcal{T}$ is closed under arbitrary unions
3. $\mathcal{T}$ is closed under finite intersections.

The members of $\mathcal{T}$ are said to be **open**.

Continuity

Definition

Let $(X, \mathcal{T}_X)$ and $(Y, \mathcal{T}_Y)$ be topological spaces and $f : X \to Y$. Then f is **continuous** at x_0 if for every $V(f(x)) \in \mathcal{T}_Y$, there exists $U(x) \in \mathcal{T}_X$ such that $f(U) \subseteq V$.

Definition

Let (X, d) be a metric space and $\epsilon > 0$, then the open ball $B(x; \epsilon) = \{y : d(x, y) < \epsilon\}$

Definition

Let (X, d) and (Y, d_Y) be metric spaces. Then $f : X \to Y$ is continuous at $x_0 \in X$ if for every $\epsilon > 0$, there exists $\delta > 0$ such that $f(B(x_0; \delta)) \subseteq B(y; \epsilon)$.

Point Clouds and Homeomorphism

Definition

A **point cloud** is a finite metric space (i.e., (X, d) such that $X = \{x_1, \ldots, x_n\}$). As such all datasets encoded in computers can be seen as mathematical point clouds.

Definition

Let $f : X \to Y$ be bicontinuous (continuous f and f^{-1}). Then f is a **homeomorphism** and we say X is homeomorphic to Y; $X \cong Y$. If a property holds for X and for any homeomorphism $f(X)$, then we say that property is a **topological invariant**.

- Since every dataset is a point cloud (i.e., a metric space), every dataset has a topology, dependent on the choice of metric.
- E.g., 32×32 black and white images can be seen as lying in $\mathcal{M}^{32\times 32}([0, 1])$. We can endow this space with a metric (e.g., Euclidean distance), which makes it a point cloud. Since it is also a point cloud, it has an underlying topology.
- $\mathcal{M}^{32\times 32}([0, 1]) \cong [0, 1]^{1024}$.

Homotopy

Definition

If $f, g : X \to Y$ are continuous, we say that they are **homotopic** if $H : X \times [0,1] \to Y$ is such that $H(x,0) = f(x)$ and $H(x,1) = g(x)$. Furthermore, we say that $f : X \to Y$ is a **homotopy equivalence** if there exists $G : Y \to X$ such that $f \circ g$ is homotopic to id_X and $g \circ f$ is homotopic to id_Y. If X is homotopy equivalent to Y then we say they are **homotopic** spaces.

- See *Algebraic Topology* by Hatcher for a more thorough treatment of this topic.
- It is enough for us to note with some hand-waving that there is a group $H_k(X, A)$ for any commutative group A and non-negative integer k, such that when X and Y are homotopic, $H_k(X, A)$ is **isomorphic** (i.e., operation-preserving) to $H_k(Y, A)$.
- We call $H_k(X, A)$ the **homotopy group** of X.

Homotopy

With respect to $H_k(X, A)$:

- If we require A to be a field, then $H_k(X, A)$ is a vector space.
- We denote the dimension of this vector space as $\beta_k(X, A)$, which will be referred to as the $k-$th Betti number.
- Informally, the $k-$th Betti number corresponds to the number of independent $k-$dimensional surfaces.
- If two spaces are homotopy equivalent, then all their Betti numbers are equal.
- The profound observation of TDA, is that data may be studied by studying the inherent independent $k-$dimensional surfaces.
- This structure is not one that has been engineered, but one that is inherent in the data's topological structure.
- Computationally, it is not feasible/efficient to compute $H_k(X, A)$.

Homology

- Homology is a computationally feasible analog for homotopy equivalences.
- The rigorous definition of homology for general topology relies on infinitely generate modules over $\mathbb{Z}$.
- This definition is not useful from a data analytics perspective because it is computationally impossible to guarantee we could feasibly compute an approximation in general.
- However, using a simple combinatorial structure called **simplicial complexes**, we can determine the homology of a given point cloud efficiently.
- We will see that we can actually extract Betti numbers from simplicial complexes.

Complexes

Definition

An **abstract simplicial complex** is a pair (V, Σ) where V is finite and Σ is a family of non-empty subsets of V such that:

$$\sigma \in \Sigma \quad \text{and} \quad \tau \subseteq \sigma \quad \text{implies} \quad \tau \in \Sigma$$

- Intuitively, simplicial complexes express a space of points, segments, triangles, tetrahedrons, and their higher dimensional analogues.
- These provide a particularly simple way to approximate topological spaces (in terms of homotopy/homology).
- Simplicial complexes admit a topology as well as an associated vector space (beyond the scope for this talk).
- It is computationally efficient to determine the homology of simplicial complexes compared to the original topological space.
- Rigorously, this is done by computing $H_k^{\text{simp}}(X, \mathbb{Z})$, associated with the simplicial complex $X = (V, \Sigma)$.
- $H_k^{\text{simp}}(X, \mathbb{Z})$ is isomorphic to the homology of X, which can be generate for a given point cloud.
- See Gunnar Carlsson's "Topology and Data."

Persistent Homology

- In order to guarantee that the homology of the point cloud X corresponds with the homology of a simplicial complex (V, Σ) is to build this complex in such a way that there is a homotopy equivalence between X and (V, Σ).
- One such complex is the Vietoris-Rips simplicial complex.

Definition

For a metric space (X, d), the Vietoris-Rips simplicial complex associated with $\epsilon > 0$, whose vertex set is X and where $\{x_0, \ldots, x_k\}$ spans a $k-$dimensional subset iff $d(x_i, x_j) \leq \epsilon$ for all $i, j \leq k$.

By varying ϵ, we can study the homology of a point cloud at varying scales. We thus do not tie ourselves to one homology, but study the **persistent homologies** across sufficient choices of ϵ to summarize the topological information within the data.

VR Complexes

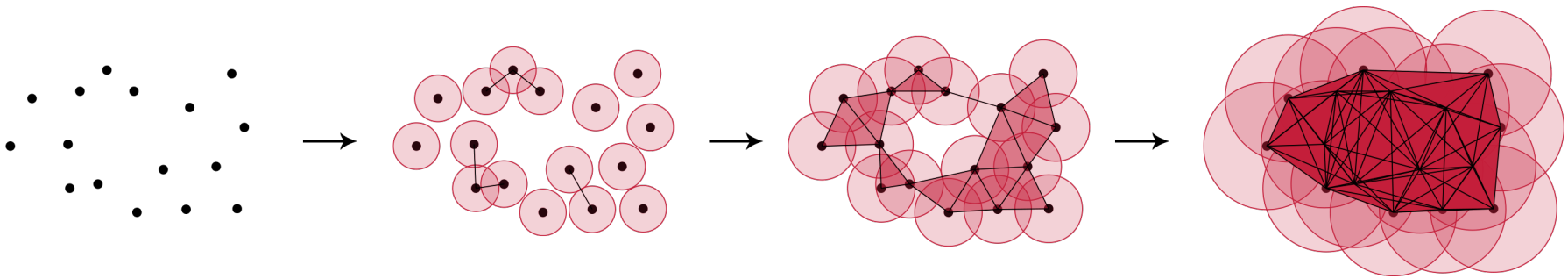

Figure: Sample simplicial complex structure using the Vietoris-Rips construction. As ϵ varies, the homology groups of the complex change.

Persistence Diagrams

- This is a representation of the persistent homology information.
- Persistence Diagrams (PDs) can be encoded in terms of two-dimensional vectors (b_i, d_i), where b_i is the birth and d_i is the death of the i^{th} homology feature.
- PDs have a natural metric associated with them called the **bottleneck distance**, which is numerically stable and computationally efficient to compute.
- We often design algorithms for TDA on PDs under the bottleneck distance.

Persistence Diagram Example

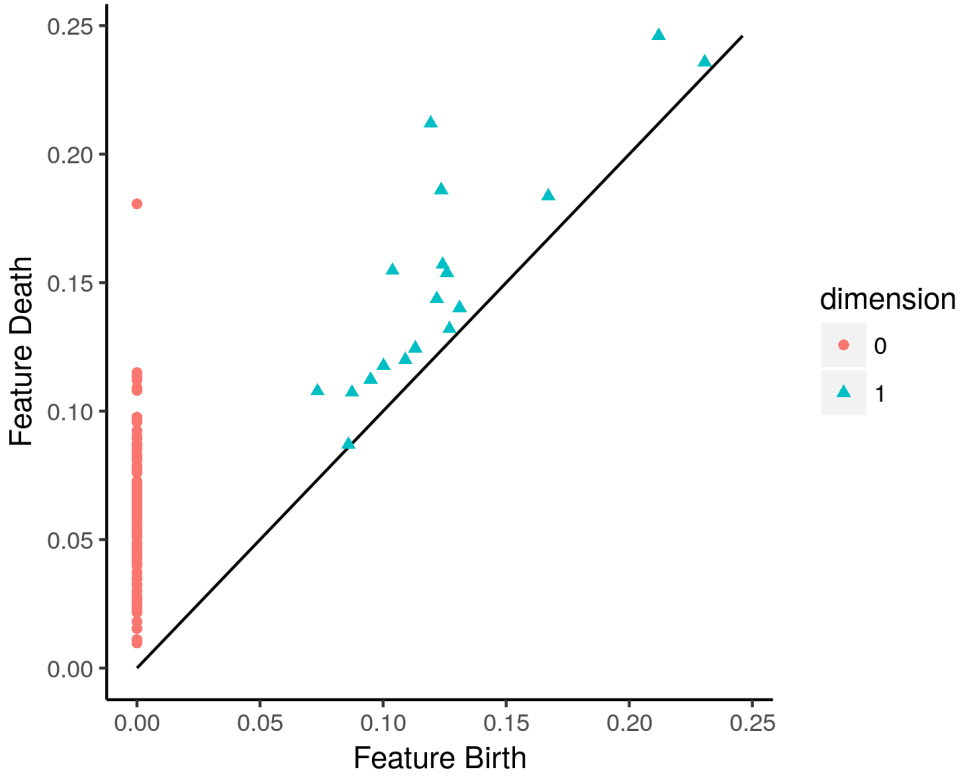

Figure: Sample persistence diagram visualized in $\mathbb{R}^n$ as a set of (birth,death) pairs.

Persistence Barcodes

- Persistent Barcodes are an alternative representation to PDs, mostly used for visualization.
- Instead of considering (b_i, d_i) pairs as points, we look at them as segments/intervals and organize them with respect to the k^{th} Betti number (y-axis).

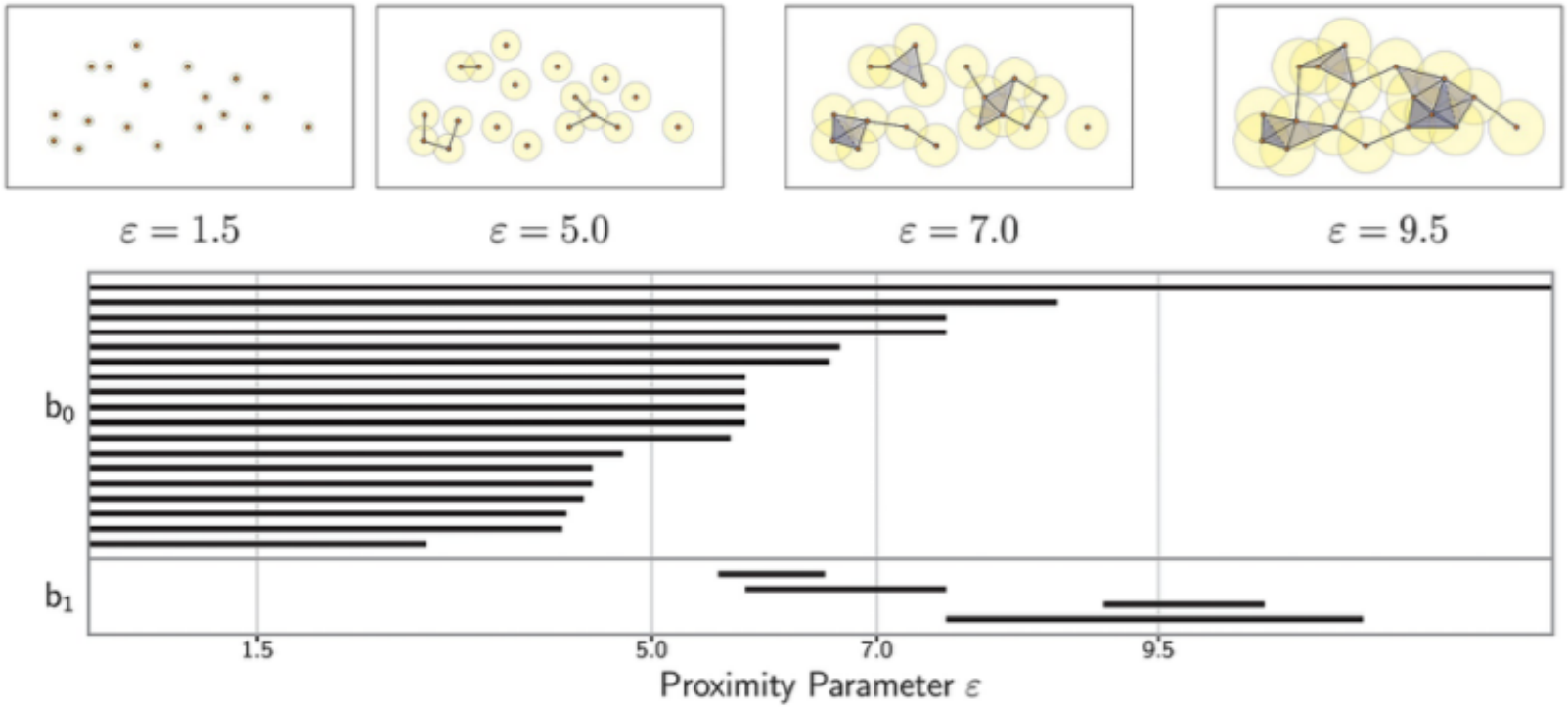

Figure: Sample barcode. As ϵ varies, the homology groups of the complex change and this is captured by line segments representing [birth, death]

Persistence Images

- Persistence Images (PIs) provide a vector representation of PDs stable with respect to input noise, which are efficient to compute, and whose resolution can be adjusted.
- The user makes three choices for this representation:
  - Resolution of the output PI.
  - Probability distribution which affects noise stability (e.g., Gaussian, Rayleigh).
  - Weighting function, which controls the relative importance of persistence coordinates (e.g., sigmoidal functions).
- PIs lend themselves to being studied under traditional image processing techniques as well as convolutional neural networks.

Persistence Image Example

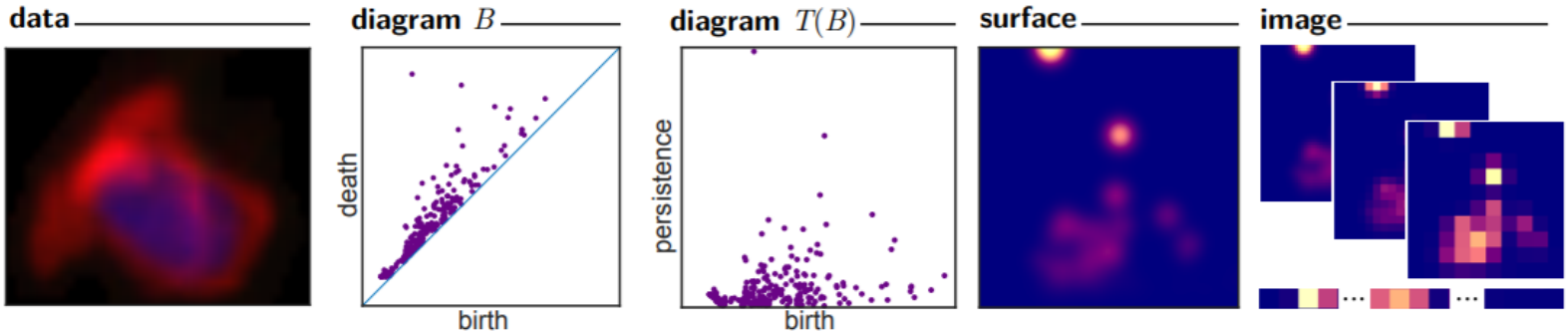

Figure: Sample persistence image pipeline from PD to Persistence Surface to PI at varying resolutions.

Open Source Tools for Computing Persistent Homology

- There have been a number of open source software packages developed for the efficient computation of persistent homology of data.
- These have been developed mostly by academic mathematicians in Python, C++, R, and Julia.
- Most popular ones include dionysus, scikit-tda, gudhi, ripser, and mapper.
- A full list of these tools has been compiled by Henry Adams.

Iris Data Classification

- Very small dataset (150 points with 4 features, 3 labels, balanced).
- 4 features: sepal length/width (cm), petal length/width (cm)
- 3 classes: setosa, versicolor, and virginica
- Very simple and computationally easy, which makes it ideal for quick sanity checks.
- Analsis Tools Considered:
  - PCA Clustering (qualitative)
  - K Nearest Neighbor Classifier Accuracy (quantitative)
  - K Nearest Neighbor Confusion Matrix (quantitative)
  - Persistence Diagrams (qualitative)
  - Bottleneck Distance Matrix (quantitative)

PCA Clustering

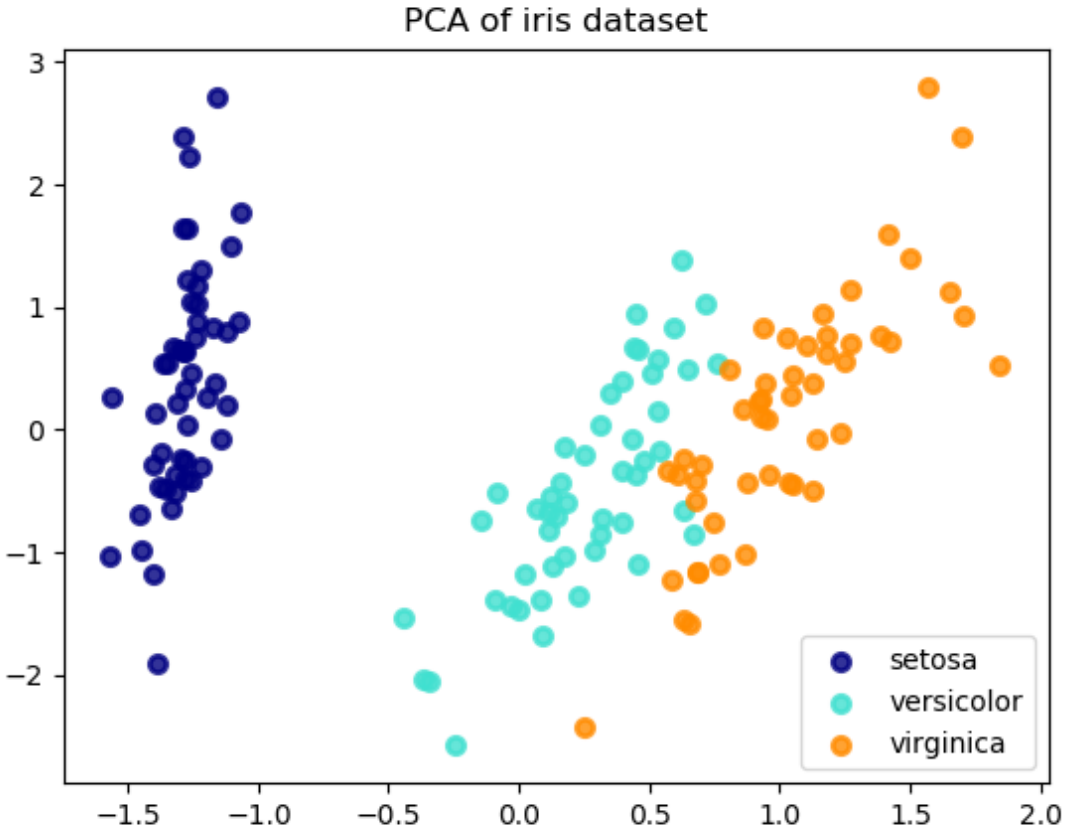

Figure: PCA Clustering of the Iris Dataset. Visually, we can see that the setosa label has high separability from versicolor and virginica, which are much closer after PCA.

K Nearest Neighbors Classifier

- Pre-processing:
  - We split the data into a 30% train and 70% test sets and keep the data balanced.
  - We also shuffle the data.
  - We scale the data such that it is within $(0, 1)$ using a min-max scaler.
- The function class we consider is trivial since it is dependent only on the training data given.
- We can view this as a weighting function that weights the k-nearest points to the input with $\frac{1}{k}$ and all other points 0.
- Classification Accuracy: 83 percent
- Training Accuracy: 91 percent

Confusion Matrix for Iris Data

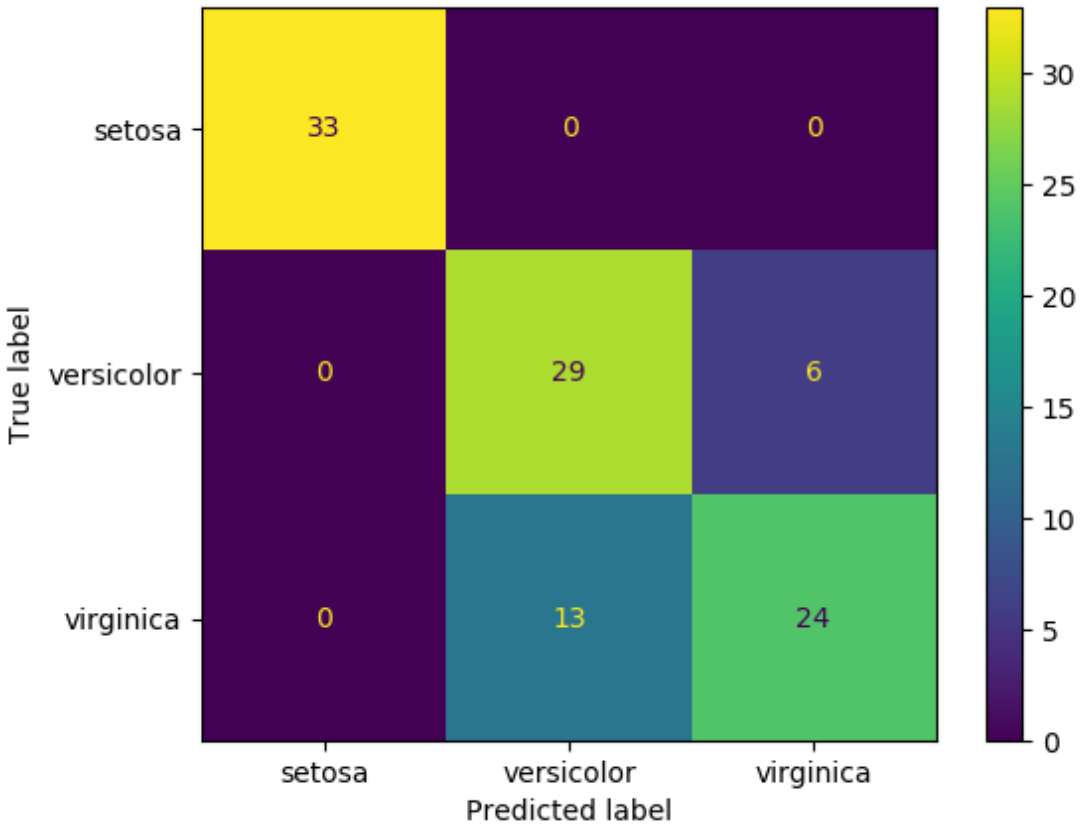

Figure: Confusion Matrix of Iris Data. We see that setosa is classified perfectly, while versicolor and virginica have some errors in classification.

Persistence Diagrams of Iris Features: Setosa

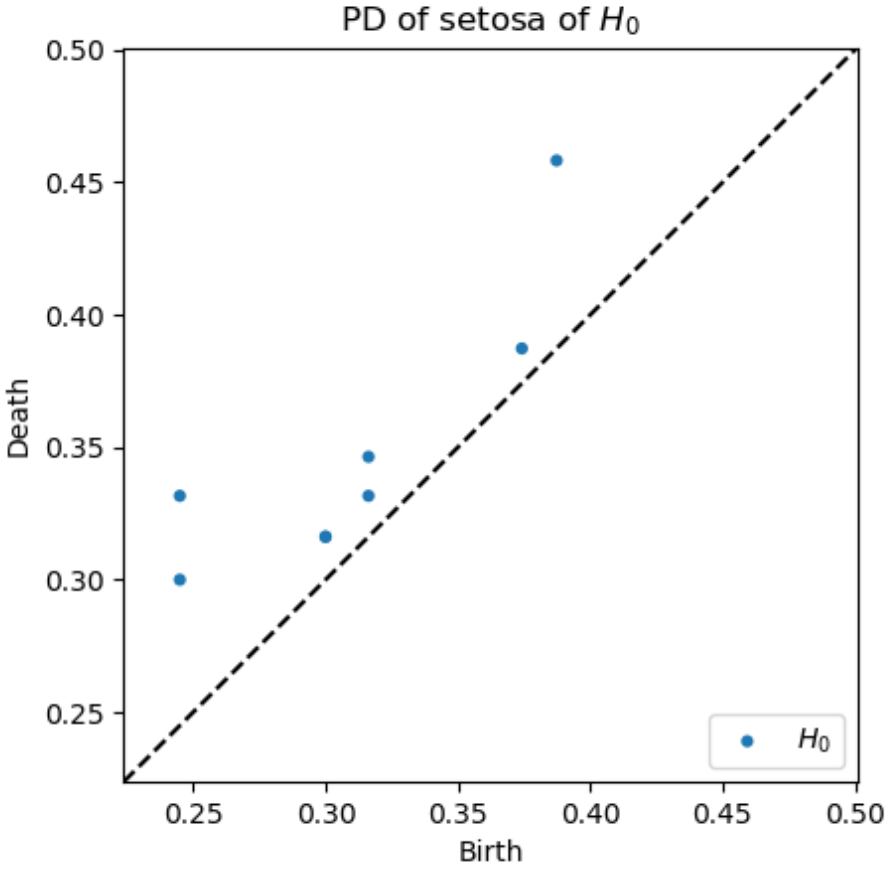

Figure: Persistence Diagram for Setosa

Persistence Diagrams of Iris Features: Versicolor

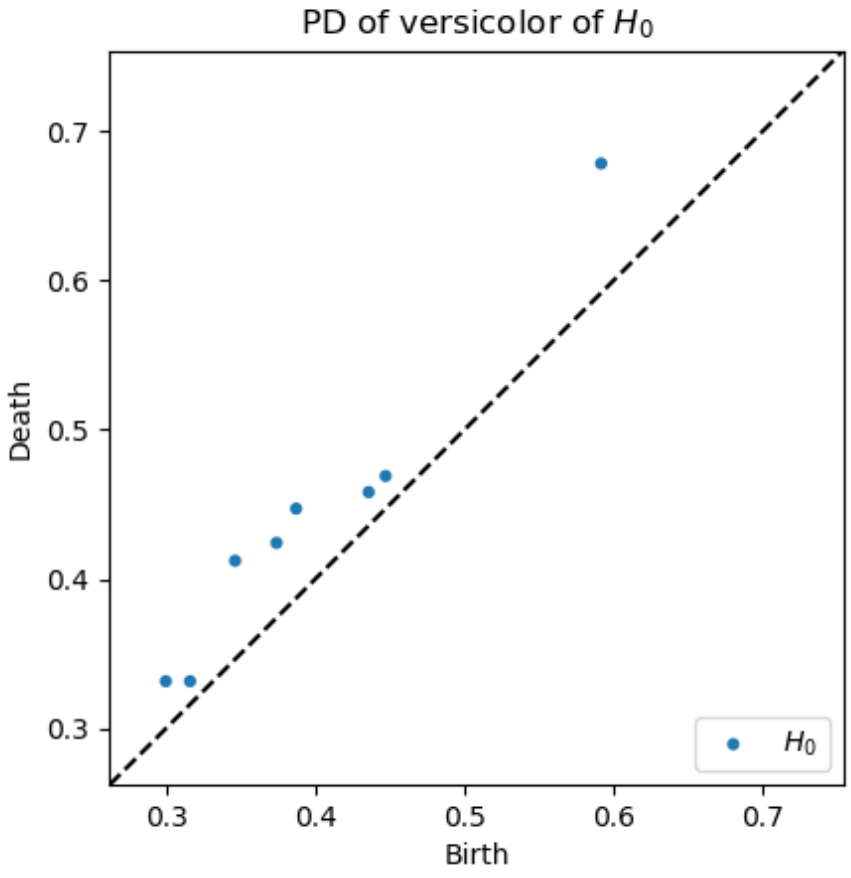

Figure: Persistence Diagram for Versicolor.

Persistence Diagrams of Iris Features: Virginica

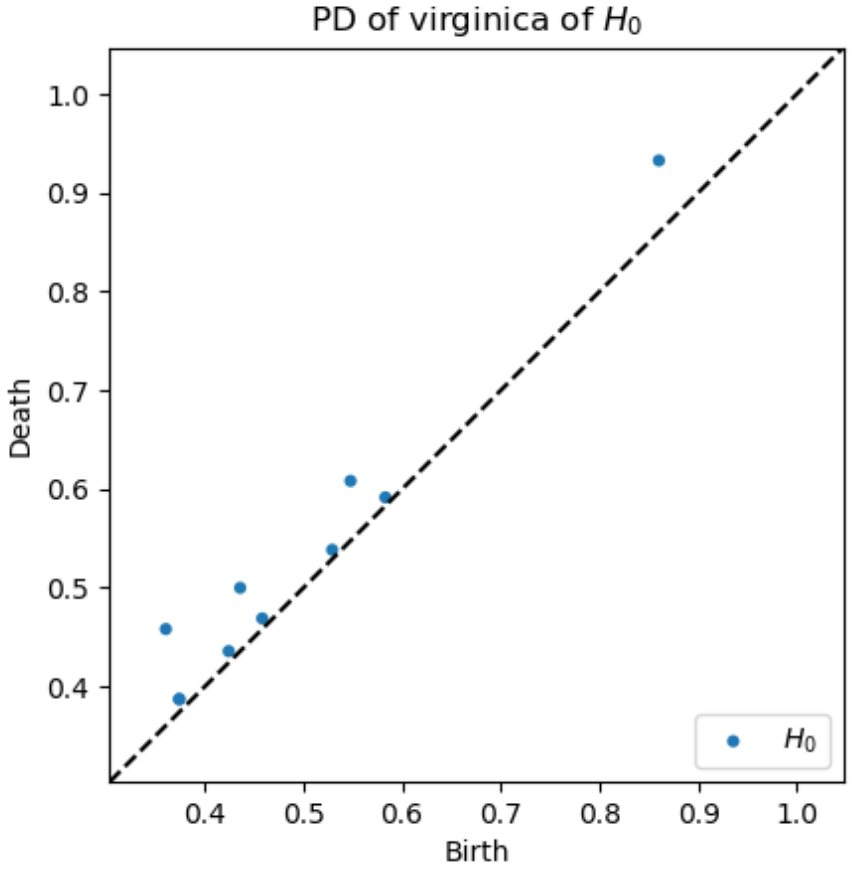

Figure: Persistence Diagram for Virginica.

Pairwise Bottleneck Distances

- The bottleneck distances for each of the iris varieties:
  - With themselves: 0 (control)
  - Setosa, Virginica: 0.04335675
  - Setosa, Versicolor: 0.04335675
  - Virginica, Versicolor: 0.04331252

A downside to the Bottleneck Distance as a measure is that quantitatively it can be hard to distinguish for simple data what a good distance is.

NIST Data Classification

- Small dataset (1797 points with 64 features, 10 labels, balanced).
- 64 features: 8×8 images with pixels in the range $[0, 16]$.
- 10 classes: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Simple and computationally able to run locally, which makes it ideal for proof-of-concept on short timelines where image analysis is needed quickly.
- Analsis Tools Considered:
  - PCA Clustering (qualitative)
  - K Nearest Neighbor Classifier Accuracy (quantitative)
  - K Nearest Neighbor Confusion Matrix (quantitative)
  - Persistence Diagrams (qualitative)
  - Bottleneck Distance Matrix (quantitative)

PCA Clustering

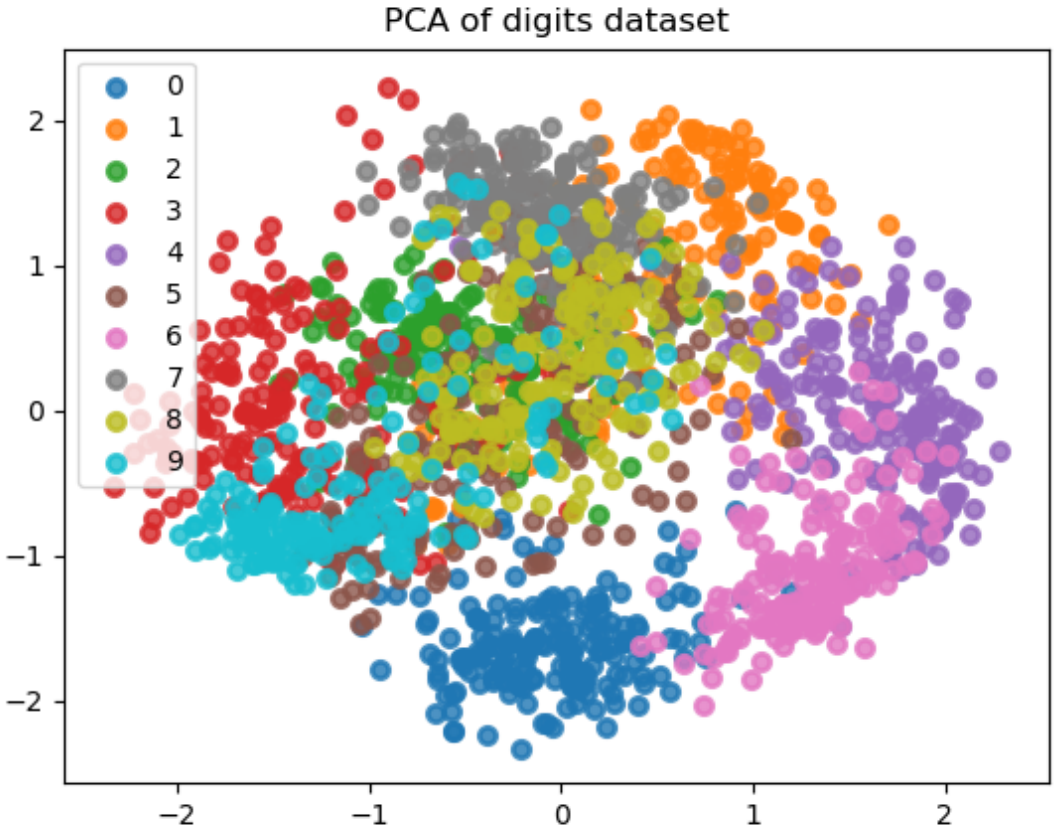

Figure: PCA Clustering of the NIST Digits Dataset. Visually, we can see that there is some clustering, but the data has a pretty high degree of similarity even after PCA.

K Nearest Neighbors Classifier

- Pre-processing:
  - We split the data into a 30% train and 70% test sets and keep the data balanced.
  - We also shuffle the data.
  - We scale the data such that it is within $(0, 1)$ using a min-max scaler.
- The function class we consider is trivial since it is dependent only on the training data given.
- We can view this as a weighting function that weights the k-nearest points to the input with $\frac{1}{k}$ and all other points 0.
- Classification Accuracy: 76 percent
- Training Accuracy: 83 percent

Confusion Matrix for Digits Data

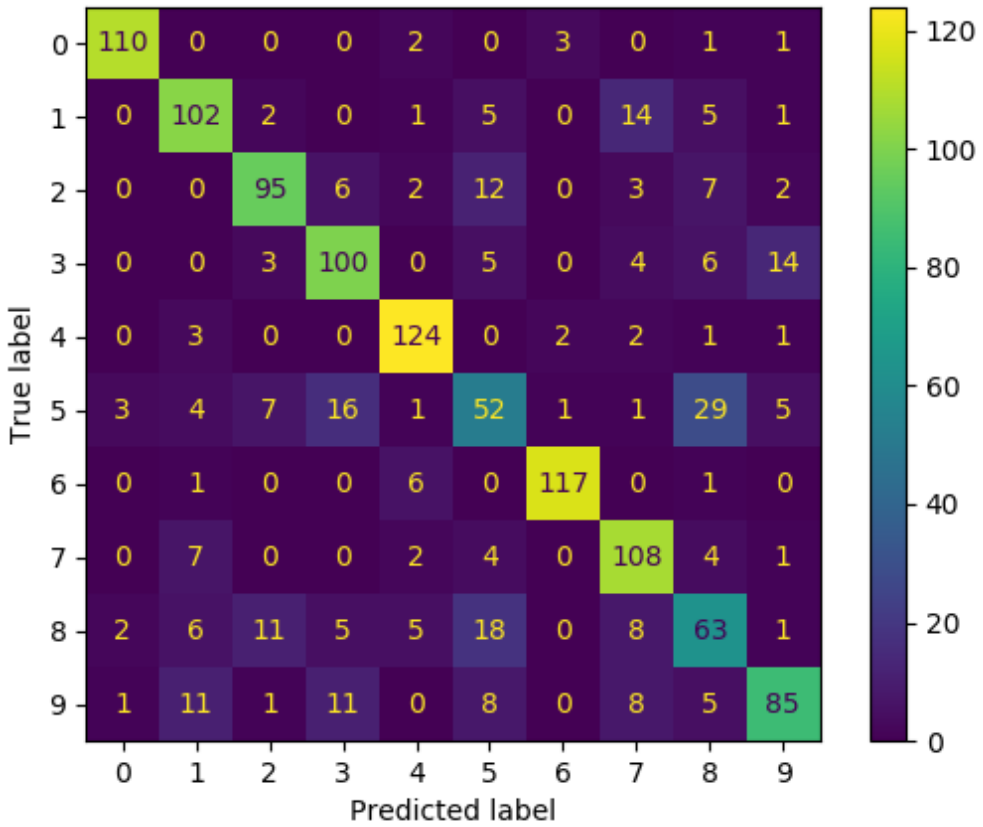

Figure: Confusion Matrix of Digit Data. We see many interesting artifacts: 0's, 4's, and 6's are classified nicely, 1's have some confusion with 7's (vice versa), 2's with 5's, 3's with 9's, 5's with 8's, 8's with 2's and 5's, and 9's with 1's and 3's.

Persistence Images of Digits: 0's

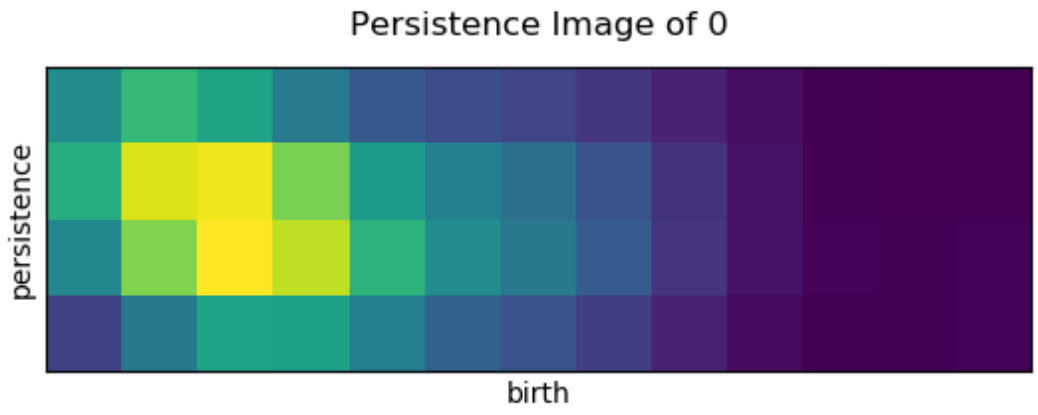

Figure: Persistence Image for 0.

Persistence Images of Digits: 0's

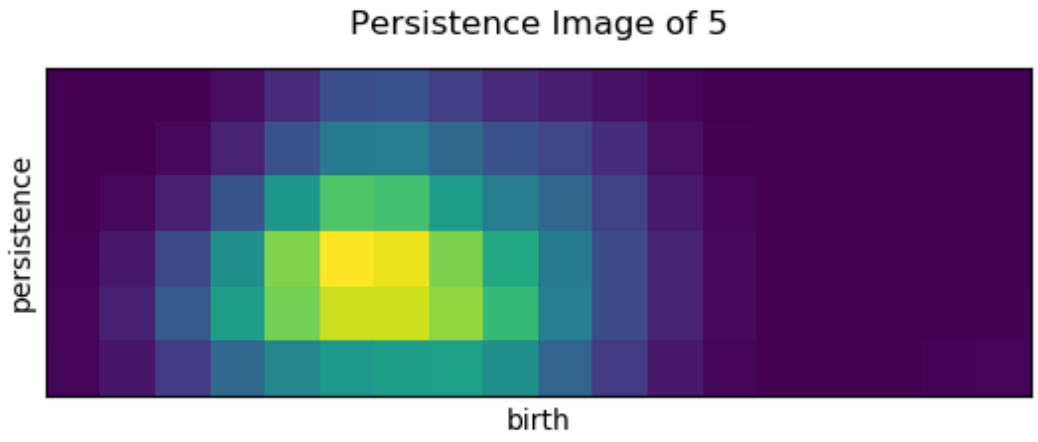

Figure: Persistence Image for 5.

Persistence Images of Digits: 8's

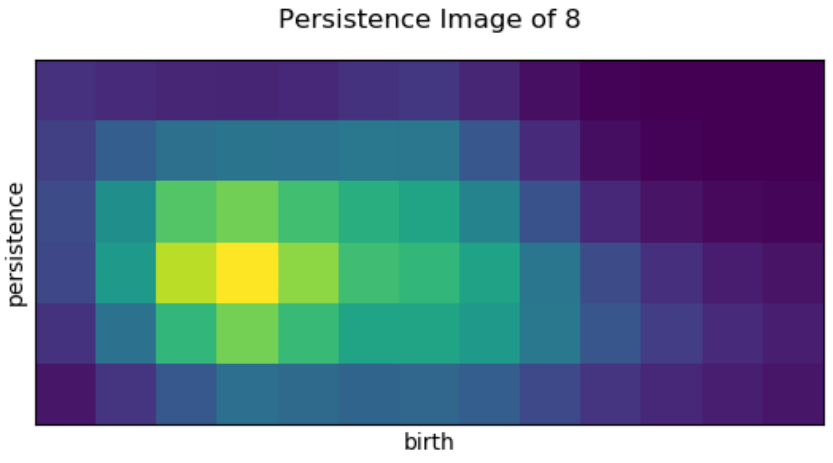

Figure: Persistence Image for 8.

Persistence Diagram of Digits: 0's, 5's, & 8's

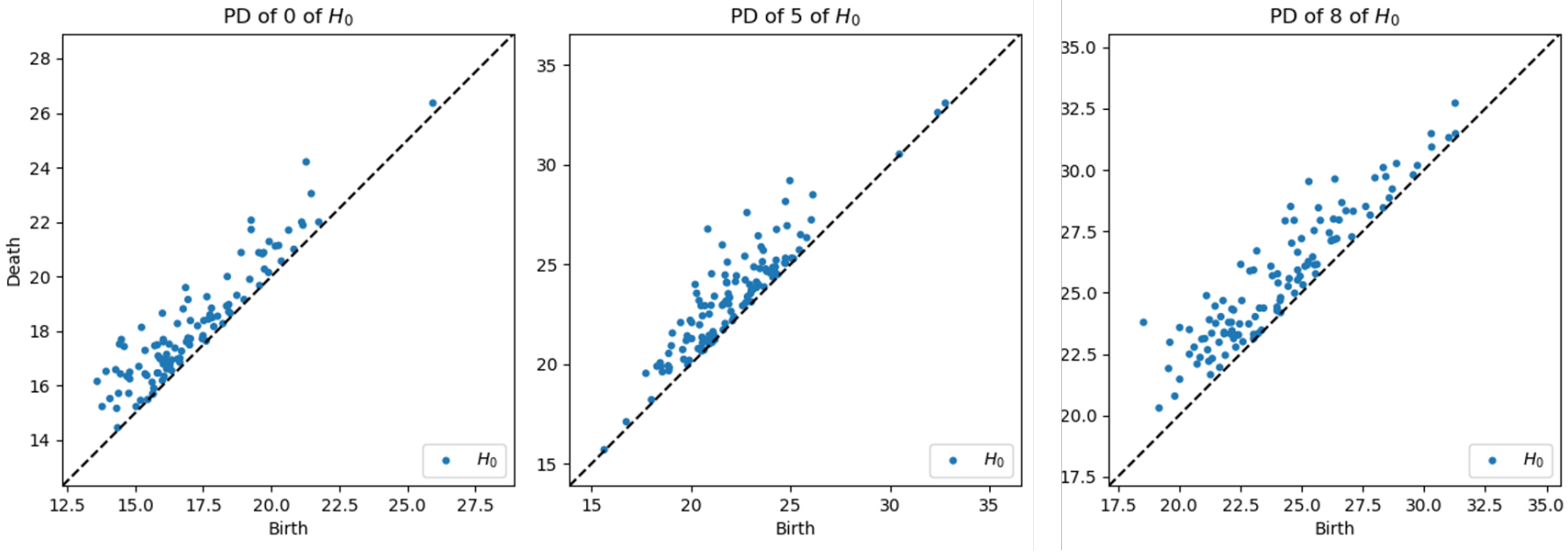

Figure: Persistence Image for 0, 5, 8. Notice the birth/death are similar for 5 and 8 versus 0.

Pairwise Bottleneck Distances

- The bottleneck distances:
  - With themselves: 0 (control)
  - 0, 5: 2.56941795
  - 0, 8: 2.21272278
  - 5, 8: 2.11571217

A downside to the Bottleneck Distance as a measure is that quantitatively it can be hard to distinguish for simple data what a good distance is.

Breast Cancer Data Classification

- Small dataset (569 points with 30 features, 2 labels, nearly balanced).
- 30 features: numeric such as radius, texture, concavity, etc.
- 2 classes: benign and malignant
- Very simple and computationally easy, but with far more dynamic feature set.
- Analsis Tools Considered:
  - PCA Clustering (qualitative)
  - K Nearest Neighbor Classifier Accuracy (quantitative)
  - K Nearest Neighbor Confusion Matrix (quantitative)
  - Persistence Diagrams (qualitative)
  - Bottleneck Distance Matrix (quantitative)

PCA Clustering

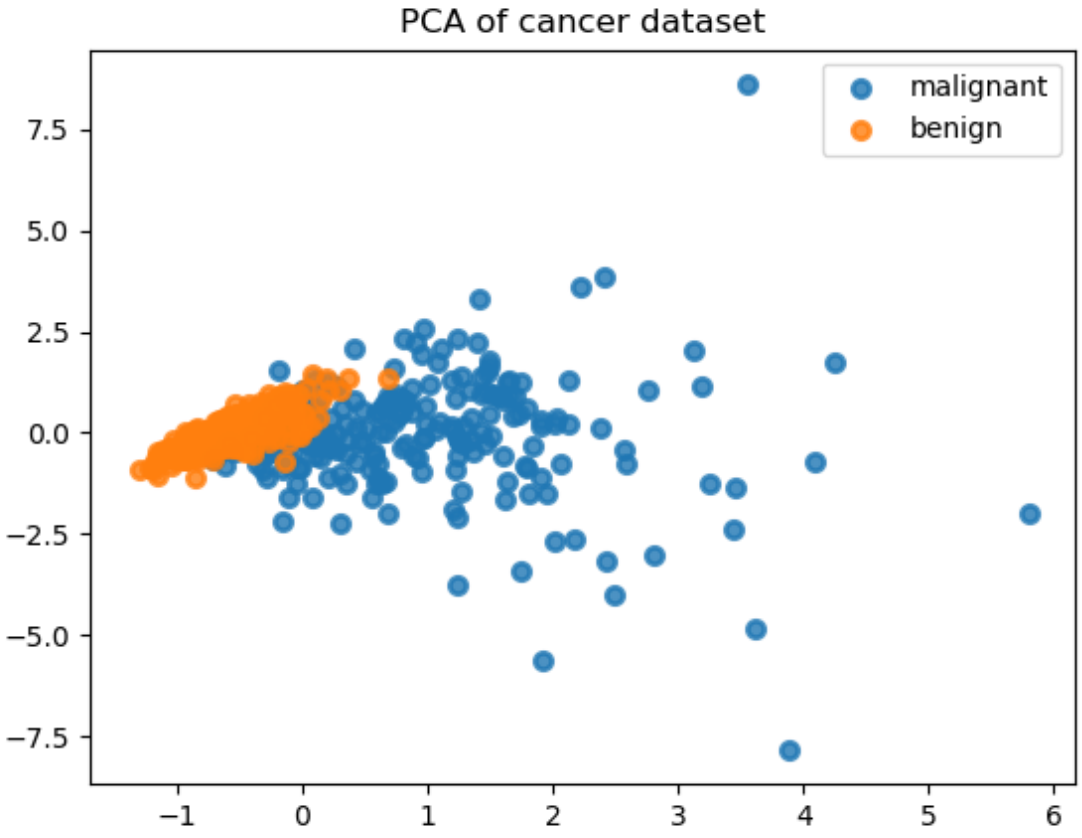

Figure: PCA Clustering of the Breast Cancer Dataset. Visually, we can see that there is some separation, but the data lies close even after PCA.

K Nearest Neighbors Classifier

- Pre-processing:
  - We split the data into a 30% train and 70% test sets and keep the data balanced.
  - We also shuffle the data.
  - We scale the data such that it is within $(0, 1)$ using a min-max scaler.
- The function class we consider is trivial since it is dependent only on the training data given.
- We can view this as a weighting function that weights the k-nearest points to the input with $\frac{1}{k}$ and all other points 0.
- Classification Accuracy: 91 percent
- Training Accuracy: 94 percent

Confusion Matrix for Cancer Data

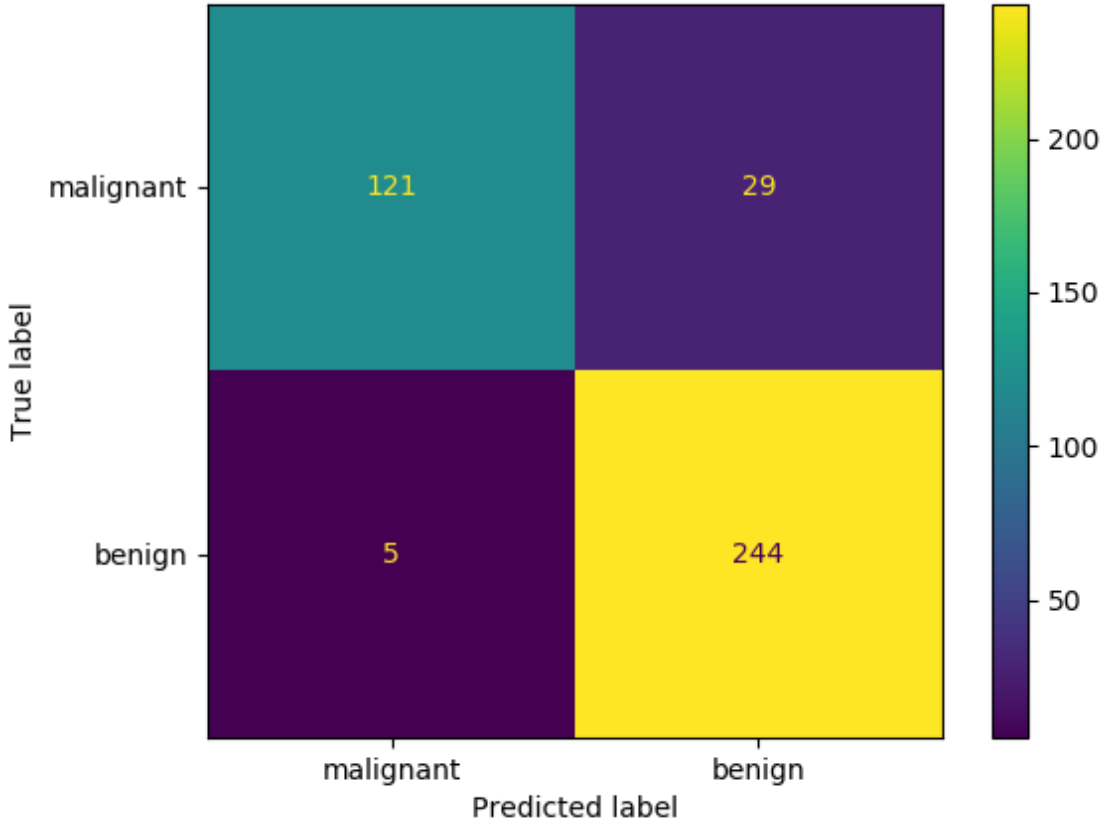

Figure: Confusion Matrix of Cancer Data. The data is pretty nicely classified in this binary case.

Persistence Diagrams of Digits: Benign

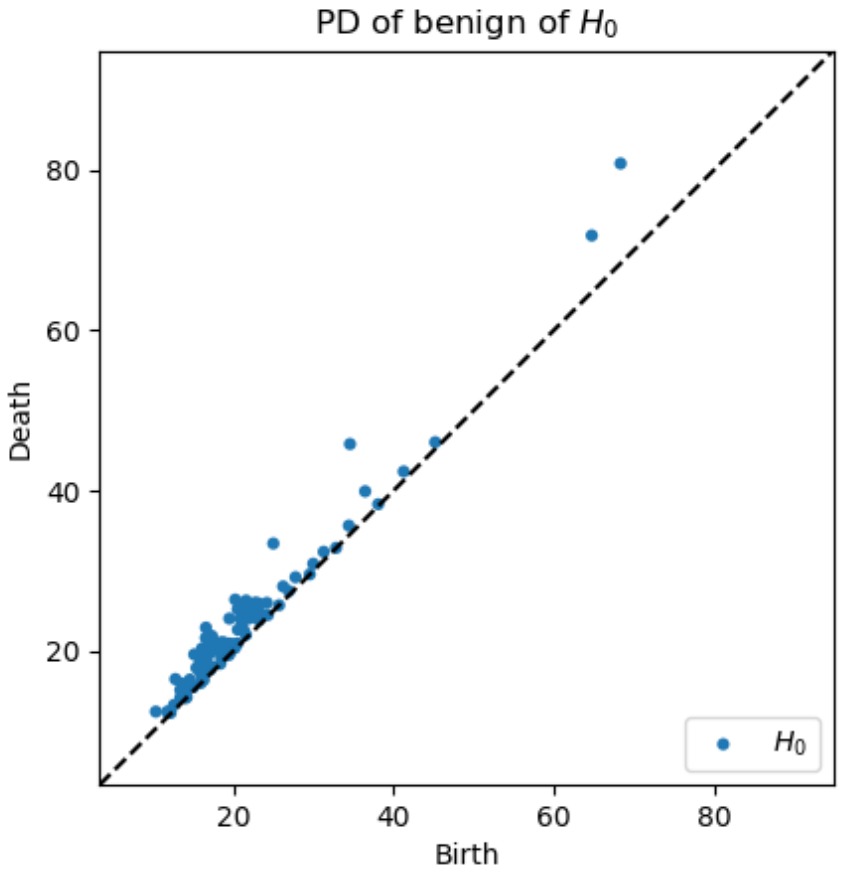

Figure: Persistence Diagram for Benign.

Persistence Diagrams of Digits: Malignant

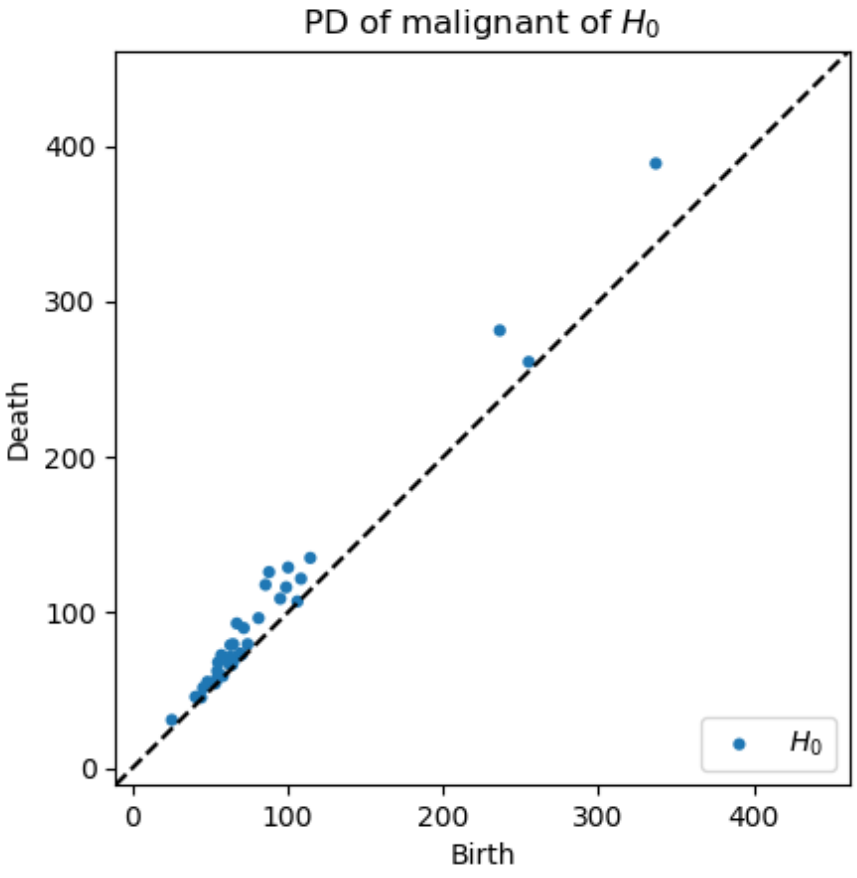

Figure: Persistence Diagram for Malignant.

Persistence Images of Digits: Benign

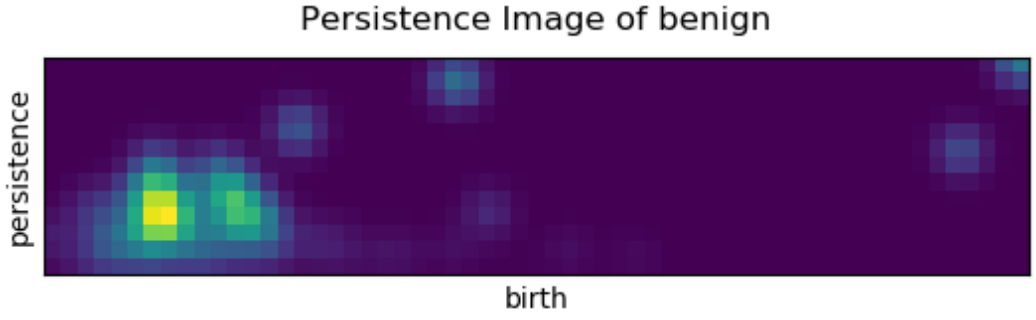

Figure: Persistence Image for Benign.

Persistence Images of Digits: Malignant

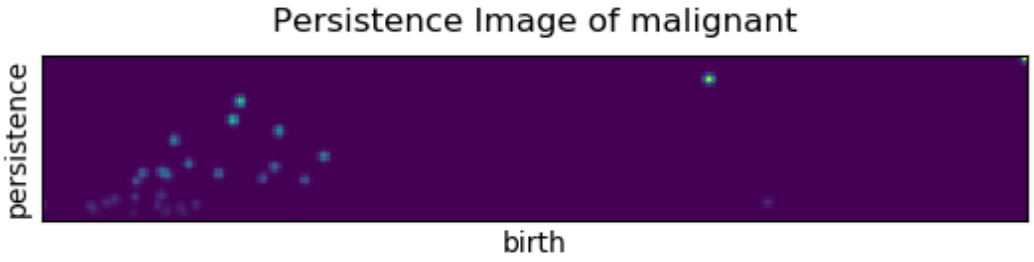

Figure: Persistence Image for Malignant.

Pairwise Bottleneck Distances

- The bottleneck distances:
  - With themselves: 0 (control)
  - Benign, Malignant: 25.95355225

A downside to the Bottleneck Distance as a measure is that quantitatively it can be hard to distinguish for simple data what a good distance is. In the binary case, unless you are comparing for hyperparameter search, this metric is close to useless.

Conclusions

- Traditional Data Analysis has seen great progress over the past decade due to the copious amounts of data available paired with ever increasing availability of high performance computing resources.
- TDA offers new and rigorous tools from mathematics for analyzing datasets at a global scale.
- As seen in the simple dataset examples, differences in data can be explained by the inherent homology information that is encoded within the dataset.
- This is a unique tool that can be added to data scientists' toolboxes for more detailed analysis of the underlying shape of data.
- The bottleneck metric must be used with caution as an actual scoring metric.
- The analysis from TDA tends to be less quantitative and more qualitative.

Future Work

- Since we can extract vectors and images as topological information, more work is needed on the practical meaningfulness of these features under traditional techniques including:
  - Convolutional Neural Networks (CNNs) on Persistence Images
  - Image Segmentation (noise/signal separation) on PIs.
  - Support Vector Machines (SVMs) and other classical ML algorithms using persistence diagrams as vector sets.
- There is a large gap in academia and practice in modalities beyond RGB images or small numerical vectors. Particular areas of interest could include:
  - Non-image based data (radar waveforms, GDELT text-based data, sentiment analysis, etc.)
  - Imagery: SAR (especially 3D SAR), HSI, MWIR, LWIR, Lidar
  - Military messaging systems (Link-16 messages), scenario data (laydown parameters)
  - Cybersecurity threat analysis

Questions?