Question 3

Programmatically illustrate the usage of persistence homologies to solve problems such as regression or classification that arise in the field of data science

Response

In this response, we will build a toy example to illustrate the usage of persistence homologies to classify 3 dimensional shapes. Many software packages that boast a wide range of TDA techniques exist today - and for this particular application, we will use a python package called `Giotto-tda`. This response is a walkthrough of a particular example in their tutorials suite. Before we delve into the actual application, we establish some background.

1. Background

Having defined and examined the alpha-shape complex in question 1, we will now take a moment to define a different complex as an effort towards a comprehensive exploration of the available tools.

Definition 8 (Vietoris-Rips complex)**.** *Given a scale parameter $\epsilon > 0$, the Vietoris-Rips complex, $\mathrm{VR}(\mathcal{X};\epsilon)$ of a point cloud $\mathcal{X}$ at scale ϵ is a simplicial complex whose vertex set is the set of points in $\mathcal{X}$ and for which a family $\{x_1,\ldots,x_k\}$ forms a $k-$subsimplex if and only if*

$$d(x_i,x_j) \leq \epsilon$$

for all $0 \leq i < j \leq k$.

Notice that we have a natural filtration $\mathrm{VR}(\mathcal{X};\epsilon_1) \hookrightarrow \mathrm{VR}(\mathcal{X};\epsilon_2)$ whenever $\epsilon_1 \leq \epsilon_2$. This observation yields the fact the Vietoris-Rips complex has a simplicial filtration and as such the procedure outlined in question 2 could be applied to the family $\{\mathrm{VR}(\mathcal{X};\epsilon)\}_\epsilon$ to compute it's persistence barcode.

We also introduce an equivalent notion to the persistence barcode - called a persistence diagram

Definition 9 (Persistence Diagram)**.** *A $\mathcal{P}-$interval (i,j) found in a persistence barcode, can be interpreted as a point on the half-plane defined by the inequality $x \leq y$ and vice versa. Thus every interval present in a barcode corresponds uniquely to a point in this half-plane. This alternate interpretation yields the persistence diagram of a filtered simplicial complex Δ.*

Finally, we note that although persistence barcodes (or equivalently, diagrams) are useful in displaying homological information - they can not be used directly in data science or machine learning applications. For instance, there is not a well-defined way to perform algebra on persistence diagrams. As such, there is a need to "vectorize" the information contained in these persistence barcodes. Many methods have been introduced to solve this issue. Some notable ones include convolving the diagram with a kernel function to produce a heat-map of the persistence information, while another approach introduces the notion of a "persistence landscape", which is a family of L^p functions that extract the persistence information from a persistence diagram.

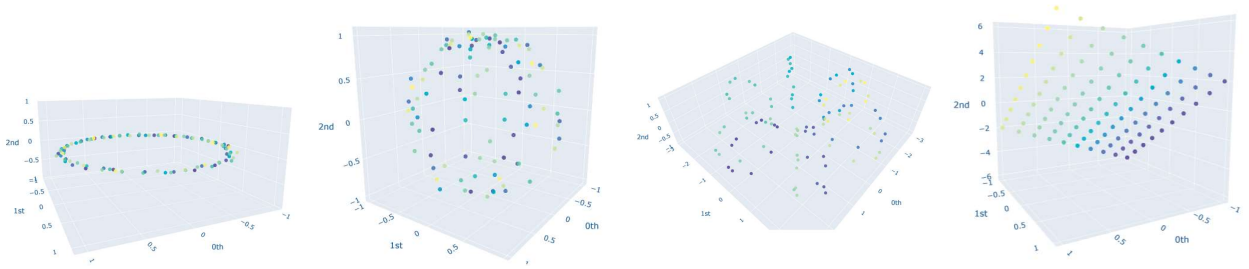
In our example today, we will introduce a much simpler notion, known as persistence entropy. Intuitively, this takes the persistence diagram in each homology dimension, and produces a number that quantifies how far away it is from having no homology in that dimension. More precisely,

Definition 10 (Persistence Entropy)**.** *Given a persistence diagram as a collection $D = \{(b_i,d_i)\}_{i\in I}$ with $b_i < d_i < \infty$, the persistence entropy of D is defined as*

$$E(D) = -\sum_{i\in I} p_i \log(p_i), \qquad \text{where } p_i = \frac{d_i - b_i}{L_D} \text{ and } L_D = \sum_{i\in I} d_i - b_i$$

2. Application

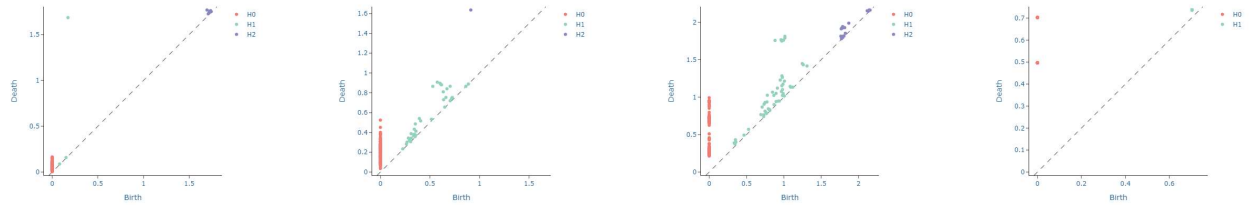
It is typical that in most applications that pose a shape classification problem, data is collected in the form of a point cloud. In our application, we will synthetically create point-clouds of four different shapes, namely a circle (on the x-y plane), a sphere, a torus and a plane. We then sample 100 points from each shape 10 different times to form our feature matrix. This way we would have 40 different 100×3 matrices that make up our data. For classification purposes, we will issue a label of 0 to the circle, 1 to the sphere, 2 to the Torus and 3 to the plane. Below is a sample point cloud of each of the four shapes:



((a)) Circle (label 0) ((b)) Sphere (label 1) ((c)) Torus (label 2) ((d)) Plane (label 3)

Figure 5: Sample point cloud for each shape

We now generate persistence diagrams for each of our point cloud via constructing a Vietoris-Rips filtration. Reproduced below are the persistence diagrams for each of the shapes above.



((a)) Circle (label 0) ((b)) Sphere (label 1) ((c)) Torus (label 2) ((d)) Plane (label 3)

Figure 6: Persistence diagram for each shape

We see that the circle poses a strong dimension 1 homology, whereas a sphere produces a strong dimension 2 homology. We also see that the Torus exhibits multiple (two) strong dimension 1 homologies, and the plane has no significant homologies present in dimensions greater than 0. We will capture this observation mathematically via calling the entropy function on each of our persistence diagram. This gives us three numbers for each persistence diagram, with each number corresponding to the entropy in a homology dimension. Since we only compute up to dimension 2 homologies, we can conveniently plot the entropies as points in $\mathbb{R}^3$. The plot is presented in figure 7 below.

We can see that there are clearly four families of points, each corresponding to a different shape in our initial point cloud data. Having successfully vectorized our persistence diagrams, we could use any of the various classifiers present in a robust machine learning package like scipy. In this case, since we can already visually confirm the presence of clusters, we can use something like the `RandomForestClassifier` that is robust for such clustering problems. In this case, upon training we obtain convergence of our model with a 'out-of-bag' validation accuracy of 100%.

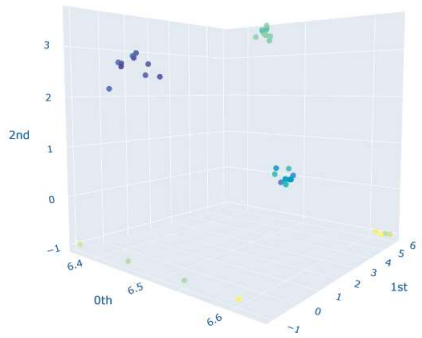


Figure 7: Entropy values for each persistence diagram

3. Upshots and shortcomings

One direct advantage of this approach is the significant reduction in the size of the feature matrix for our machine learning model. More precisely, in a more traditional approach, the entire point cloud would serve as the input to the machine learning model, which would mean that each input representing a point cloud would be a 300 dimensional vector. With the sheer lack of input samples, we can expect our convergence to be very slow. In comparison, using our approach, we devise a model that takes only three parameters, namely the entropies in each dimension of homology. Thus there is a tremendous reduction in dimensionality while preserving the accuracy of the model.

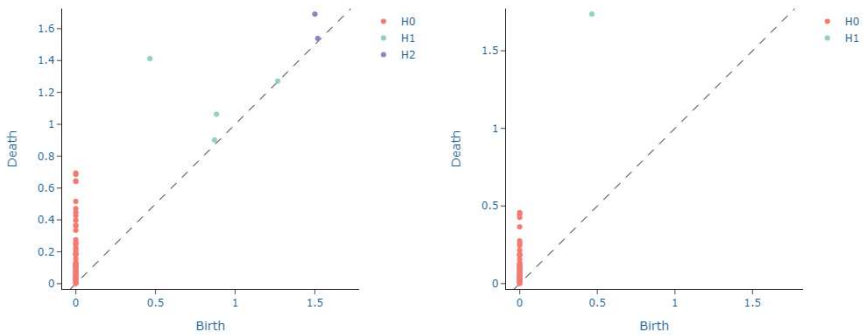
Another thing worth noting is that the computation of lower dimensional homologies might suffer from computational errors and noise that come from the embedding of our point-cloud in a higher dimension. For instance, we could see in figure 6(a) that though the cloud represents a circle, the mere fact that the circle is embedded in a three dimensional space gives rise to numerical errors that manifest as non-zero homologies in dimension 2. To combat this, we can use techniques like robust subspace recovery (RSR), to first recover a lower dimensional subspace that has enough data points from our cloud $\mathcal{X}$. Running our persistence homologies on this smaller and cleaner point cloud should eradicate some of these 'phantom' homologies that were present in our original computations. Shown below is an example of such an improvement. Consider an unlabelled point cloud of a circle corrupted with some higher dimensinal noise. We could use an RSR scheme to first recover the circle by labelling the points in $\mathcal{X}$ that lie on the x-y plane.

((a)) Unlabelled data $\mathcal{X}$ ((b)) Labelled data RSR($\mathcal{X}$)

Figure 8: Using RSR to first recover an underlying subspace

Running our homology computations certainly reduces the amount of the noise in our persistence diagrams

as illustrated below:



((a)) Persistence diagrams before labelling ((b)) Persistence diagrams after labelling

Figure 9: Clean dimension 1 homologies and removed dimension 2 homologies