Computation of Persistent Homology Using the Delaunay-Rips Filtration

An efficient family of simplicial complexes for topological data analysis

Amish Mishra and Francis Motta

Abstract

Topological Data Analysis is an emerging area rooted in theories from Algebraic Topology, which enables researchers to extract discriminating geometric and topological features from data. We give an overview of some of the popular methods of extracting features from point-cloud data, which first requires one to construct a 1-parameter family of spaces on the data using the geometry of the cloud. We demonstrate their benefits and shortcomings and introduce a new, more efficient construction that we name the Delaunay-Rips Filtration. We justify conditions on the data that guarantee stability of our method when computing persistent homology. Aided by intuitive examples, we also provide an empirical run-time comparison of the two existing methods with our new algorithm on the computation of the persistence diagrams of some synthetic data sets.

1 Introduction

Welcome to our paper!

2 Background

2.1 Topological Data Analysis

We are living in an information age where data-driven decision making is a huge area of interest. With so much data at our hands, many questions naturally arise.

- How do we extract relevant information from the data?
- How do we even know what is relevant and what is not?
- If we are unable to visualize large quantities of data, especially data in high dimensions, then how do we know what sort of data set we are inspecting?
- Further, how can we compare information extracted from one data set with another?

These are the sorts of difficult and fascinating questions tackled in the field of Topological Data Analysis (TDA). From the name itself, TDA hints at leveraging ideas borrowed from topology with data analysis techniques to measure and quantify qualitative features of data. At a more nuanced level, TDA appears as the child of Algebraic Topology, Computer Science, Statistics, Data Analysis, and Computational Geometry. Results from each field have found beautiful applications in TDA and have shed new light on applicability of theoretical results from mathematics.

2.2 Simplicial Complexes

The idea behind extracting topological features from a given point-cloud requires there to be a method of assigning some sort of “shape” to the data. Only then are we able to study its associated topological properties. The scenario we face is that we have a finite dimensional metric space out of the point-cloud that we need to assign a shape to. With the human eye, we may be able to make out some appropriate surface our data set can live on. However, this situation pokes at a fundamental question that arises when computing: how can we get a computer to do what a human can do? We need to introduce the idea of a *simplicial complex*, a computational efficient way for a computer to build a surface onto a data set.

Definition 2.1. *A **simplicial complex** is a collection K of non-empty subsets of a set K_0 such that $\{v\} \in K$ for all $v \in K_0$, and if $\sigma \in K$ then $\tau \in K$ for all $\tau \subseteq \sigma$. The elements of K_0 are called **vertices** of K, and the elements of K are called **simplices**. Additionally, we say that a simplex has **dimension** p or is a p-simplex if it has cardinality of $p+1$. We use K_p to denote the collection of p-simplices. The k-skeleton of K is the union of the sets K_p for all $p \in \{0, 1, \dots, k\}$. If τ and σ are simplices such that $\tau \subset \sigma$, then we call τ a **face** of σ, and we say that τ is a face of σ of **codimension** k' if the dimensions of τ and σ differ by k'. The **dimension** of K is defined as the maximum of the dimensions of its simplices. A map of simplicial complexes, $f : K \to L$, is a map $f : K_0 \to L_0$ such that $f(\sigma) \in L$ for all $\sigma \in K$. [EH10]*

2.3 Simplicial Homology

2.4 Vietoris-Rips Complex

One of the simplest ways to build a complex on a data set X is by considering the pairwise distance between the points. The approach described here is an algorithmic, bottom-up approach that adds higher and higher dimensional simplices to the complex for a fixed scale. For a given scale $\varepsilon > 0$, if $d(x, x') \leq 2\varepsilon$ for $x, x' \in X$, then we add the edge between x and x' into our complex. Once all of the edges are added, we add the higher dimensional simplices if their faces are already in the complex. That is, we add the k-simplex $\sigma = \{x_0, x_1, \dots, x_k\}$ to the complex if every subset $u \subset \sigma$ is already in the complex. Formally, we define the Vietoris-Rips complex [Ott+17] for scale $\varepsilon > 0$

$$VR_\varepsilon(X) = \{\sigma \subseteq X \mid d(x, x') \leq 2\varepsilon,\ \forall x, x' \in \sigma\}.$$

2.5 Delaunay Triangulation

Although the Vietoris-Rips complex is simple to implement, constructing it on data sets with large numbers of points results in computation drawback. As the scale increases, we see that adding certain simplices does not affect the homology of the point cloud. We need some way to "weed" out these extraneous simplices as we construct our complex to increase computational efficiency. Turning to a tool of Computational Geometry, we incorporate the Delaunay Triangulation in our construction. Our definition is adapted from "A roadmap for the computation of persistent homology"[Ott+17]. Assume our data X lives in the space $\mathbb{R}^n$. Let $x \in X$. We define

$$V_x = \{p \in \mathbb{R}^d \mid d(p, x) \leq d(p, x')\ \forall x' \in X\}.$$

Each V_x is called a Voronoi cell. Note that $\{V_x\}_{x \in X}$ forms a cover of $\mathbb{R}^n$. This cover is known as the Voronoi decomposition of $\mathbb{R}^n$ with respect to X. To construct the Delaunay triangulation from this cover, we connect $x, x' \in X$ with an edge if V_x and $V_{x'}$ are neighbors (that is, the Vornoi cells share a wall). When the points in X are in general position, this gives us a graph (1-skeleton) on X that is known as the Delaunay Triangulation. Formally, we define [EH10]

$$Del(X) = \{\sigma \subset X \mid \bigcap_{u \in \sigma} V_u \neq \emptyset\}.$$

We will use $Del(X)$ as the underlying graph structure when defining the Delaunay-Rips complex in section 3.1.

As an alternate defintion to the Delaunay Triangulation that we will employ in section 3.5, we have the following

Definition 2.2. *For a set X of points in $\mathbb{R}^d$, a Delaunay triangulation is a triangulation $DT(X)$ such that no point in X is inside the circum-hypersphere of any simplex in $DT(X)$.*

2.6 Persistence

3 Delaunay-Rips Complex

3.1 Definition and Construction

The Delaunay-Rips complex is our alternative method of building a complex on a data set X. The idea is similar to the construction of the Delaunay-Čech complex defined in [BE16]. Delaunay-Rips utilizes the conceptual simplicity of the Vietoris-Rips complex while cutting down on the number of high dimensional and extraneous simplices. This computational speed-up is by virtue of using the Delaunay Triangulation as the "backbone" of building the Vietoris-Rips complex on X. The idea is that we build the Vietoris-Rips complex on X but only add edges if the edges occur in the Delaunay 1-skeleton of the point cloud. The higher dimensional k-simplices are then added the traditional way they are in section 2.4. Formally, we define the Delaunay-Rips Complex for a given scale $\varepsilon > 0$

$$DR_\varepsilon(X) = \{\sigma \subseteq Del(X) \mid d(x, x') \leq 2\varepsilon, \ \forall x, x' \in \sigma\}.$$

3.2 Example Data Set

We now illustrate an example construction of the Delaunay-Rips complex for increasing radius r from 0 to 4 in figure 1. We add triangles as soon as their 3 edges appear in the complex: this is illustrated as a shaded triangle in the figure. Notice how at $r = 4$ although many balls overlap, we have only added the simplices that appear in the Delaunay triangulation of the point cloud.

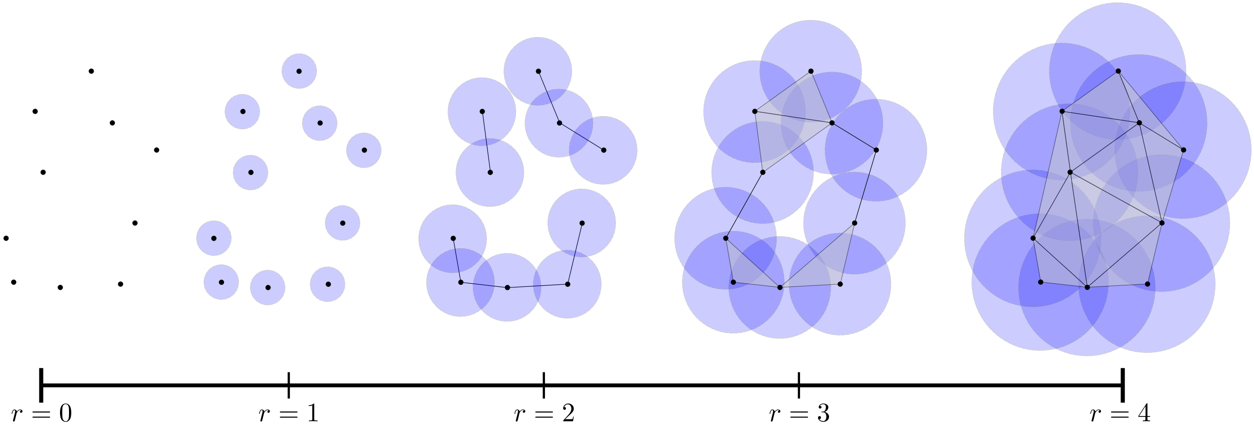

Figure 1: Example of constructing Delaunay-Rips complex on 10 points in $\mathbb{R}^2$.

3.3 Run-time Analysis Comparison

In this section, we show some empirical results of the performance of computing persistence using the Delaunay-Rips filtration on varying datasets. The runtimes are measured as the time between the data set being inputed into the corresponding algorithm and the persistence diagram being produced. Figure 2 shows the runtime compared to larger and larger number of points being sampled from a noisy 2-sphere. Each data point on the plot is the median of 10 trials. The box-and-whisker plots on each data point show the spread of the runtimes. Similarly, figure 3 shows the runtime compared to larger and larger number of dimensions the data is sampled from.

3.4 Persistence Diagram Instability

The Delaunay-Rips construction gains computational efficiency at the cost of stability. We demonstrate a simple, yet clear example of how this instability can arise. In figure 4, we can visually see a particular

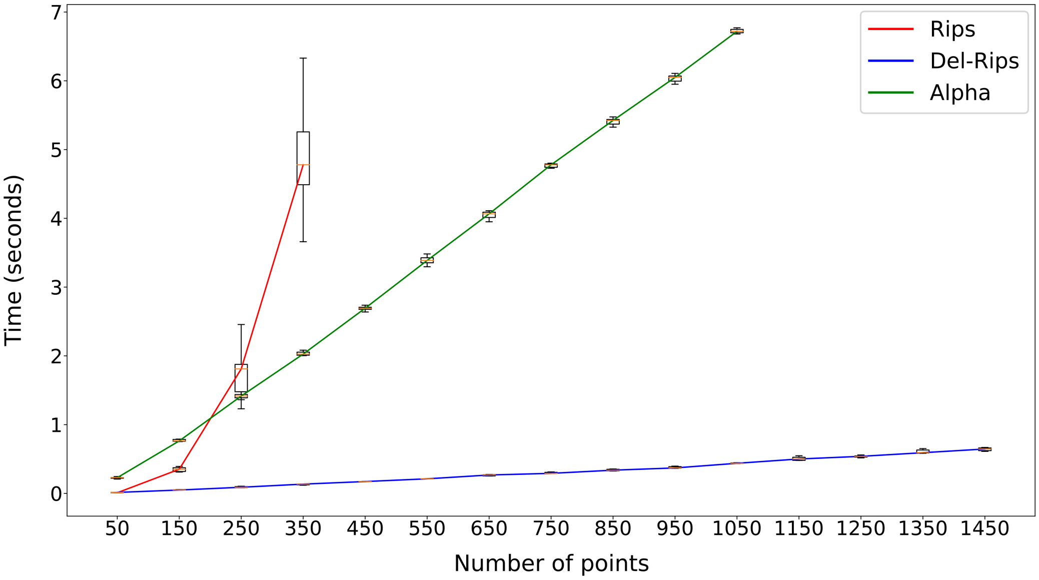

Figure 2: Runtime comparison of Rips, Alpha, and Delaunay-Rips as number of points are increased. Data set is taken from the surface of a 2-sphere of radius 1 with 0.1 noise.

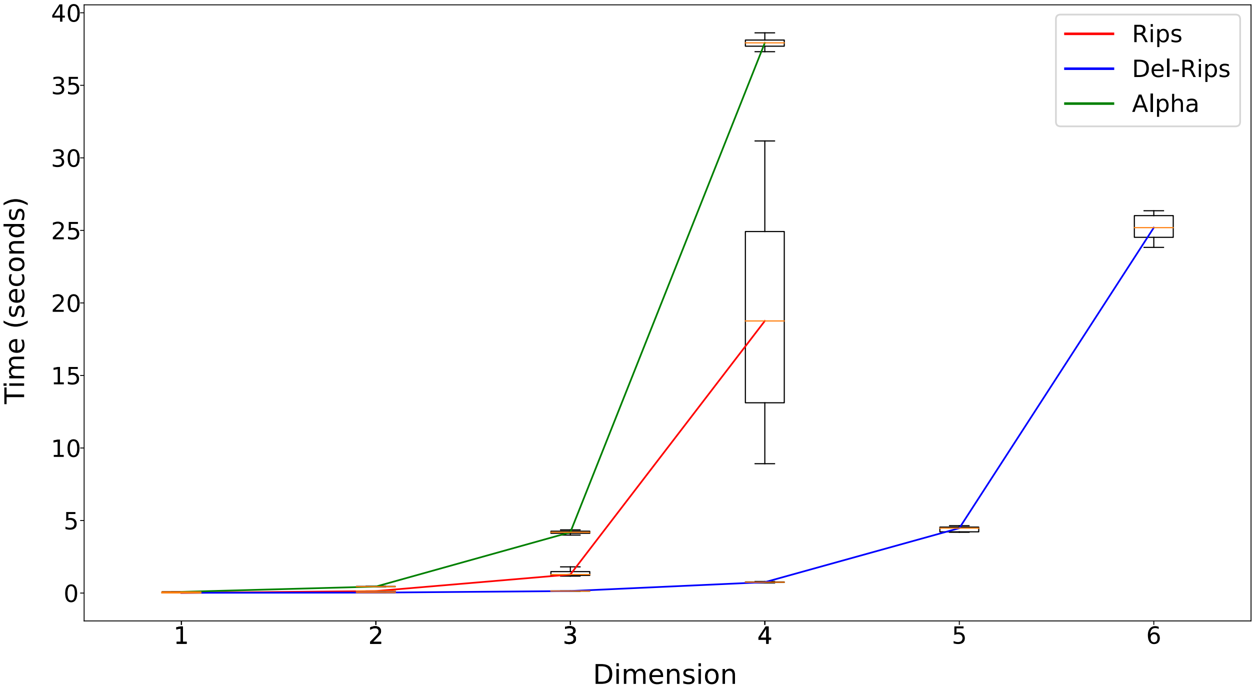

Figure 3: Runtime comparison of Rips, Alpha, and Delaunay-Rips as dimension of d-sphere increases. (Note that 1, 2, 3 on the x-axis correspond to 1-sphere, 2-sphere, 3-sphere). Data set is 100 points from the surface of a d-sphere of radius 1 with 0.1 noise.

configuration of four points giving a radically different Persistence Diagram as we move the right-most point across the circumcircle of the other 3 points. In the figure, we have marked the Delaunay Triangulation of the points to show case the point at which an edge flip occurs (namely when all four points lie on the same circle). We now proceed to formally prove the instability of this particular configuration of points.

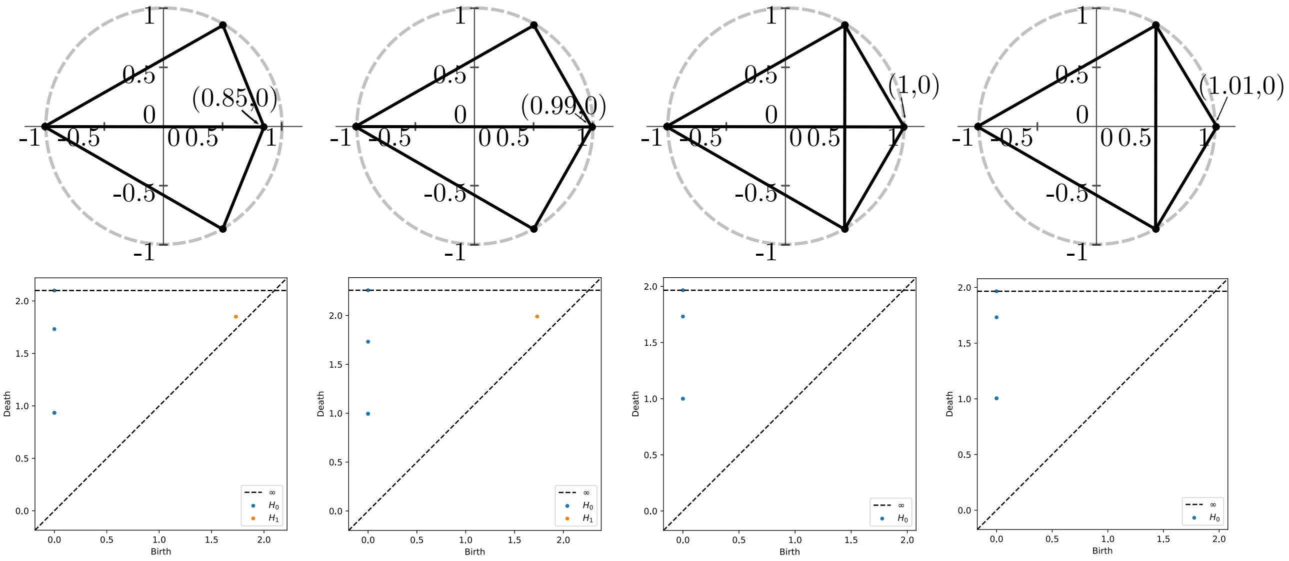

Figure 4: Persistence Diagrams of 4 point example

Definition 3.1. *Let X and Y be two non-empty subsets of a metric space (M, d). We define their* ***Hausdorff distance*** *$d_H(X, Y)$ by*

$$d_H(X,Y) := \max\left\{\sup_{x\in X} d(x,Y), \sup_{y\in Y} d(X,y)\right\}.$$

Definition 3.2. *If X and Y are two compact metric spaces, then the* ***Gromov-Haudorff distance*** *$d_{GH}(X, Y)$ is defined to be the infimum of all numbers $d_H(f(X), g(Y))$ for all metric spaces M and all isometric embeddings $f : X \to M$ and $g : Y \to M$.*

Definition 3.3. *Let X and Y be two persistence diagrams. Let $\eta : X \to Y$ be a bijection and $||x - y||_\infty := \max\{|x_1 - y_1|, |x_2 - y_2|\}$ for $x = (x_1, x_2), y = (y_1, y_2)$. We define the* ***bottleneck distance*** *between the diagrams as*

$$W_\infty(X,Y) = \inf_{\eta : X\to Y} \sup_{x\in X} ||x - \eta(x)||_\infty.$$

Let $(\mathcal{P}, d_{GH})$ be the space of point clouds equipped with the Gromhov-Hausdorff metric and let $(\mathcal{D}, W_\infty)$ be the space of Persistence Diagrams equipped with the bottle neck metric. Let

$$Pers_1 : \mathcal{P} \to \mathcal{D}$$

where $Pers_1(P)$ is the persistence diagram showing the H_1 classes of the point cloud P constructed using the Delaunay-Rips complex. Our example comes from 4 points taken in $\mathbb{R}^2$ where the instability is demonstrated as the discontinuity of $Pers_1$.

Let $P \in \mathcal{P}$ as $P = \{(-1, 0), (\frac{1}{2}, \frac{\sqrt{3}}{2}), (\frac{1}{2}, -\frac{\sqrt{3}}{2}), (1, 0)\}$. Note that the points all lie on the unit circle, so the Delaunay 1-skeleton has an edge between every pair of points (See figure). Thus, $Pers_1(P)$ has no H_1 class with non-zero persistence using the Delaunay-Rips filtration, as can be verified by the reader.

Fix $\varepsilon = 0.1$. We now show that for any $\delta > 0$, there exists $P' \in \mathcal{P}$ such that $d_{GH}(P, P') < \delta$, but $W_\infty(Pers_1(P), Pers_1(P')) \geq \varepsilon$. Take $P' = \{(-1, 0), (\frac{1}{2}, \frac{\sqrt{3}}{2}), (\frac{1}{2}, -\frac{\sqrt{3}}{2}), (1 - x, 0)\}$ with $0 < x < \delta < \frac{2-\sqrt{3}}{2}$. This is a small perturbation of P by pushing the point $(1, 0)$ inside the unit circle thereby putting the points in general position. We only work with $\delta < 2 - \sqrt{3}$ so that $Pers_1(P')$ maintains an H_1 class with non-zero persistence; for this example to work, we further need $\delta < \frac{2-\sqrt{3}}{2}$. We just compute the Hausdorff distance d_H between P and P' in the plane taking the isometric embedding of P to be the map that sends each of its points to itself in $\mathbb{R}^2$ and the same embedding for P'. Since the Gromov-Hausdorff distance is the infimum of $d_H(f(P), g(P'))$ over all isometric embeddings $f : P \to X$ and $g : P' \to X$ into any metric space X, $d_H(P, P')$ serves as an upper bound for $d_{GH}(P, P')$. We find that

$$d_{GH}(P, P') \leq d_H(P, P') = x < \delta.$$

Recall that $Pers_1(P)$ has no H_1 class with non-zero persistence. Thus, to compute $W_\infty(Pers_1(P), Pers_1(P'))$, we must match the H_1 class of $Pers_1(P')$ with the diagonal. The H_1 class of $Pers_1(P')$ has birth $\sqrt{3}$ and death $2 - x$ as calculated in the Appendix, section A.1. Using the max norm, we find

$$d := W_\infty(Pers_1(P), Pers_1(P')) = 2 - x - \sqrt{3} \geq 2 - \frac{2-\sqrt{3}}{2} - \sqrt{3} \geq 0.1 = \varepsilon.$$

Hence, our map $Pers_1$ is discontinuous at P. This gives us insight into when the Delaunay-Rips construction of the Persistence Diagram experiences instability–namely when points are not in general position. We now have motivation to ask if we have stability of the PD when the underlying Delaunay-Rips complex does not change under a perturbation of the point cloud.

3.5 Stability in a Neighborhood

1. What is the best our method can do? Use knowledge on stability of Delaunay Triangulation.

Let P denote our point cloud and P' denote the perturbed point cloud. We use $|xy|$ notation to denote the Euclidean distance between points $x, y \in \mathbb{R}^n$. Let V_x denote the closed Voronoi region for the point $x \in P$. Here is a lemma that should work in $\mathbb{R}^n$:

Lemma 3.4. *Two points $p_i, p_j \in P$ are strong Voronoi neighbors if and only if there exists an m such that*

$$\max\{|mp_i|, |mp_j|\} < \min_{k \neq i,j} |mp_k|.$$

We claim

Theorem 3.5. *Let $p_i, p_j \in P$ be strong Voronoi neighbors and $m \in V_{p_i} \cap V_{p_j}$. For $\varepsilon = \min\{\frac{1}{4}(\min_{k\neq i,j} |mp_k| - |mp_j|), \frac{1}{2}\min_{i\neq j} |p_ip_j|\}$, an ε-perturbation of P leaves p_i and p_j as strong Voronoi neighbors.*

Proof. Let $p_i, p_j \in P$ satisfy Lemma 3.4 with $m \in V_{p_i} \cap V_{p_j}$. Note that $0 < \varepsilon < \frac{1}{2}\min_{i\neq j} |p_ip_j|$ to ensure a unique correspondence between the points of P and the points of P'. We begin with the conclusion of Lemma 3.4:

$$\max\{|mp_i|, |mp_j|\} < \min_{k\neq i,j} |mp_k|$$

$$2\max\{|mp_i|, |mp_j|\} < 2\min_{k\neq i,j} |mp_k|$$

$$\min_{k\neq i,j} |mp_k| + 3\max\{|mp_i|, |mp_j|\} < 3\min_{k\neq i,j} |mp_k| + \max\{|mp_i|, |mp_j|\}$$

$$\frac{1}{4}\min_{k\neq i,j} |mp_k| + \frac{3}{4}\max\{|mp_i|, |mp_j|\} < \frac{3}{4}\min_{k\neq i,j} |mp_k| + \frac{1}{4}\max\{|mp_i|, |mp_j|\}$$

$$\max\{|mp_i|, |mp_j|\} + \frac{1}{4}(\min_{k\neq i,j} |mp_k| - \max\{|mp_i|, |mp_j|\}) < \min_{k\neq i,j} |mp_k| - \frac{1}{4}(\min_{k\neq i,j} |mp_k| - \max\{|mp_i|, |mp_j|\})$$

$$\begin{aligned} \max\{|mp_i|, |mp_j|\} + \varepsilon &\leq \max\{|mp_i|, |mp_j|\} + \frac{1}{4}(\min_{k\neq i,j} |mp_k| - \max\{|mp_i|, |mp_j|\}) \\ &< \min_{k\neq i,j} |mp_k| - \frac{1}{4}(\min_{k\neq i,j} |mp_k| - \max\{|mp_i|, |mp_j|\}) \leq \min_{k\neq i,j} |mp_k| - \varepsilon. \end{aligned} \tag{1}$$

Now, without loss of generality, let

$$|mp'_i| = \max\{|mp'_i|, |mp'_j|\}.$$

We note by the triangle inequality that

$$\max\{|mp'_i|, |mp'_j|\} \leq |mp_i| + |p_i p'_i| \leq \max\{|mp_i|, |mp_j|\} + \varepsilon. \tag{2}$$

Similarly, we have by the triangle inequality

$$\min_{k\neq i,j} |mp_k| - \varepsilon = \min_{k\neq i,j}(|mp_k| - \varepsilon) \leq \min_{k\neq i,j}(|mp_k| - |p_k p'_k|) \leq \min_{k\neq i,j} |mp'_k|. \tag{3}$$

Putting together equations 1, 2, 3, we have

$$\max\{|mp'_i|, |mp'_j|\} \leq \max\{|mp_i|, |mp_j|\} + \varepsilon < \min_{k\neq i,j} |mp_k| - \varepsilon \leq \min_{k\neq i,j} |mp'_k|.$$

Thus, we now apply Lemma 3.4 and have that p'_i and p'_j remain strong Voronoi neighbors. □

Lemma 3.6. *Given a subset $S = \{p_1, p_2, \dots, p_{d+1}\} \subset P$, there exists $\varepsilon > 0$ such that any ε-perturbation S' of S has $\{p'_1, p'_2, \dots, p'_{d+1}\} \subset P'$ as linearly independent.*

Proof. Let $\mathcal{S}$ be the unique $(d-1)$-sphere containing S with center O. We proceed by linear algebra, treating each p_i as a vector $\mathbf{p_i} \in \mathbb{R}^d$. We locate the center $O \in \mathbb{R}^d$ by finding the intersection of the perpendicular bisecting planes to the line segments $\overline{\mathbf{p_2 p_1}}, \overline{\mathbf{p_3 p_1}}, \dots, \overline{\mathbf{p_{d+1} p_1}}$. Thus, we will have d hyperplanes each in d variables: this is a system of linear equations that has a solution since our subset S is in general position (that is, no two perpendicular bisecting hyperplanes are parallel). The vector equation for the perpendicular bisecting hyperplane of $\overline{\mathbf{p_i p_1}}$ can be found by determining the normal vector and a point on the hyperplane. A point on the plane we can use is simply the midpoint of $\overline{\mathbf{p_i p_1}}$:

$$\mathbf{b_i} := \frac{\mathbf{p_i} + \mathbf{p_1}}{2}.$$

The normal vector to the plane is the directional vector based at the midpoint and pointing in the direction of one end of the line segment $\overline{\mathbf{p_i p_1}}$. We find the normal vector to be

$$\mathbf{n_i} := \mathbf{p_i} - \mathbf{b_i} = \frac{\mathbf{p_i} - \mathbf{p_1}}{2}.$$

Thus, the vector equation of the perpendicular bisecting hyperplane of $\overline{\mathbf{p_i p_1}}$ is

$$\mathbf{n_i} \cdot \mathbf{r} = \mathbf{n_i} \cdot \mathbf{b_i}$$

where $\mathbf{r}$ is the position vector for an arbitrary point on the hyperplane.
We are now ready to construct the following maps whose composition will output the circumcenter $O \in \mathbb{R}^d$ as a function of the $d+1$ points defining the sphere $\mathcal{S}$. We illustrate an example of this with 3 points in $\mathbb{R}^2$ in figure 5. This construction is instrumental to obtaining ε balls around each point that ensures the circumcenter O remains bounded. Define

$$f : \underbrace{\mathbb{R}^d \times \cdots \times \mathbb{R}^d}_{d+1} \to \mathcal{M}_d(\mathbb{R}) \times \mathbb{R}^d$$

$$f : (\mathbf{p_1}, \mathbf{p_2}, \dots, \mathbf{p_{d+1}}) \mapsto \left(\begin{bmatrix} \mathbf{n_2} \\ \mathbf{n_3} \\ \vdots \\ \mathbf{n_{d+1}} \end{bmatrix}, \begin{bmatrix} \mathbf{n_2} \cdot \mathbf{b_2} \\ \mathbf{n_3} \cdot \mathbf{b_3} \\ \vdots \\ \mathbf{n_{d+1}} \cdot \mathbf{b_{d+1}} \end{bmatrix} \right)$$

$$g : \mathrm{GL}_d(\mathbb{R}) \times \mathbb{R}^d \to \mathbb{R}^d$$

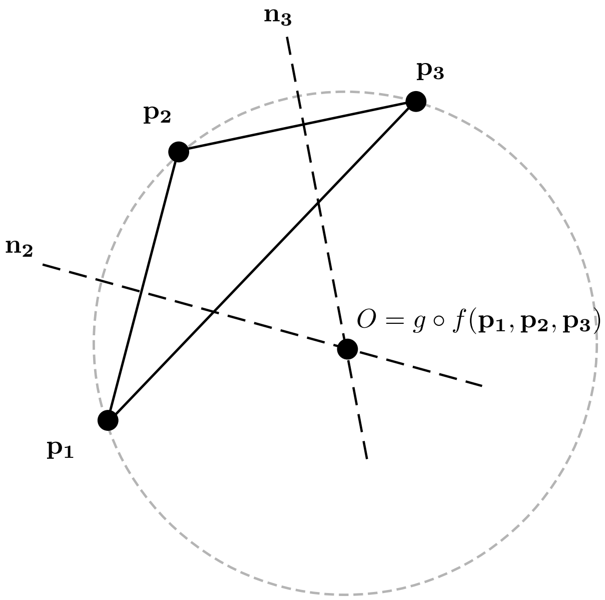

Figure 5: Locating the circumcenter of 3 points in $\mathbb{R}^2$ as a function of the 3 points.

$$g : (A, b) \mapsto A^{-1}b.$$

We prove that the map f is continuous. Let the domain of f have the the standard topology of $\mathbb{R}^{d(d+1)}$ (i.e. the open balls in $\mathbb{R}^{d(d+1)}$ form a basis for the topology). Observing that $\mathcal{M}_d \cong \mathbb{R}^{d^2}$, we have $\mathcal{M}_d(\mathbb{R}) \times \mathbb{R}^d \cong \mathbb{R}^{d^2} \times \mathbb{R}^d \cong \mathbb{R}^{d(d+1)}$. We let this space have the standard topology of $\mathbb{R}^{d(d+1)}$. Note that f is a polynomial in each component of it's output by how it is defined. Thus, f itself is continuous.

We carefully prove that the map g is continuous. Let $\mathbb{R}^d$ have the standard topology and $\mathrm{GL}_d(\mathbb{R}) \times \mathbb{R}^d$ have the subspace topology of $\mathcal{M}_d(\mathbb{R}) \times \mathbb{R}^d$. Note that taking the inverse of a matrix is a continuous function in the topology induced by the Frobenius norm (proven in detail in Section A.2). Mapping $b \mapsto b$ is a continuous function in the standard topology of $\mathbb{R}^d$. Since these two functions are defined on spaces with the same topology, recall that the product of two continous functions is continuous. Hence, $g(A, b) = A^{-1}b$ is a continuous function.

Observe that $g \circ f$ is the circumcenter of the sphere through the input points. This composition is continous since g and f are continous. Let $p = (\mathbf{p_1}, \mathbf{p_2}, \ldots, \mathbf{p_{d+1}}) \in \mathbb{R}^{d(d+1)}$ and let $O = g \circ f(p)$. Thus, for any $\varepsilon_1 > 0$, there exists a $\delta > 0$ such that when $p' = (\mathbf{p_1}', \mathbf{p_2}', \ldots, \mathbf{p_{d+1}}')$ satisfies $||p - p'|| < \delta$, then

$$||O - O'|| < \varepsilon_1$$

where $O' = g \circ f(p')$. Thus, we have that $B_\delta(p) \subset \mathbb{R}^{d(d+1)}$ is an open ball of radius δ centered at p. Let π_i be the projection map that projects p into its ith coordinate:

$$\pi_i : \underbrace{\mathbb{R}^d \times \cdots \times \mathbb{R}^d}_{d+1} \to \mathbb{R}^d$$

$$\pi_i : (\mathbf{p_1}, \mathbf{p_2}, \ldots, \mathbf{p_{d+1}}) \mapsto \mathbf{p_i}.$$

This map is an open map, thus $\pi_i(B_\delta(p))$ is an open set in $\mathbb{R}^d$. Since we have $\mathbf{p_i} \in \pi_i(B_\delta(p))$ by definition, we can find an open ball $B_{\delta_i}(\mathbf{p_i}) \subset \mathbb{R}^d$ for each $1 \leq i \leq d+1$. Letting $\varepsilon = \min\{\delta_i\}_{i=1}^{d+1}$, we have an ε neighborhood around each point $\mathbf{p_i}$ defining $\mathcal{S}$ that ensures the circumcenter O stays in an ε_1 ball.

□

Lemma 3.7. *For a Delaunay simplex $D = \{p_1, p_2, \ldots, p_{d+1}\} \subset P$, there exists $\varepsilon > 0$ such that any ε-perturbation D' of D has $\{p_1', p_2', \ldots, p_{d+1}'\} \subset P'$ as a Delaunay simplex.*

Proof. An illustrated example can be found in figure 6. Let $\mathcal{S}$ be the unique $(d-1)$-sphere containing D with center O. Since D is a Delaunay simplex, for all $p_k \in P \backslash D$, $d(O, p_1) < d(O, p_k)$. Let $t(p_k) := (d(O, p_k) - d(O, p_1))/4$ and select $p_{\min} \in P \backslash D$ that minimizes t. It follows that,

$$d(O, p_1) + 2t = d(O, p_{\min}) - 2t.$$

Since P is in general position and finite, $t > 0$. Let $\varepsilon_1 = t$, then we obtain from Lemma 3.6 that there is $\delta > 0$ such that each $d(p_i, p_i') < \delta$ for $i = 1, 2, \ldots, d+1$ implies $d(O, O') < \varepsilon_1$. Now, we take $\varepsilon = \min(\delta, \varepsilon_1)$ and D' is the ε-perturbation of D. Let $\mathcal{S}'$ be the unique $(d-1)$-sphere containing D' with center O'. Using the triangle inequality, we have the following chains of inequalities:

$$d(O', p_1') \leq d(O', O) + d(O, p_1) + d(p_1, p_1')$$

$$< \varepsilon_1 + d(O, p_1) + \delta \leq t + d(O, p_1) + t$$

$$= d(O, p_1) + 2t = d(O, p_{\min}) - 2t.$$

Since $d(O, p_{\min})$ minimized t, it must be that $d(O, p_{\min}) \leq d(O, p_k)$ for any $p_k \in P \backslash D$. Thus, we have

$$d(O, p_{\min}) - 2t \leq d(O, p_k) - t - t < d(O, p_k) - \varepsilon_1 - \delta$$

$$\leq d(O, p_k) - d(O, O') - d(p_k, p_k') \leq d(O', p_k').$$

We have concluded that

$$d(O', p_1') < d(O', p_k')$$

for any $p_k' \in P' \backslash D'$. Hence, D' is a Delaunay simplex. □

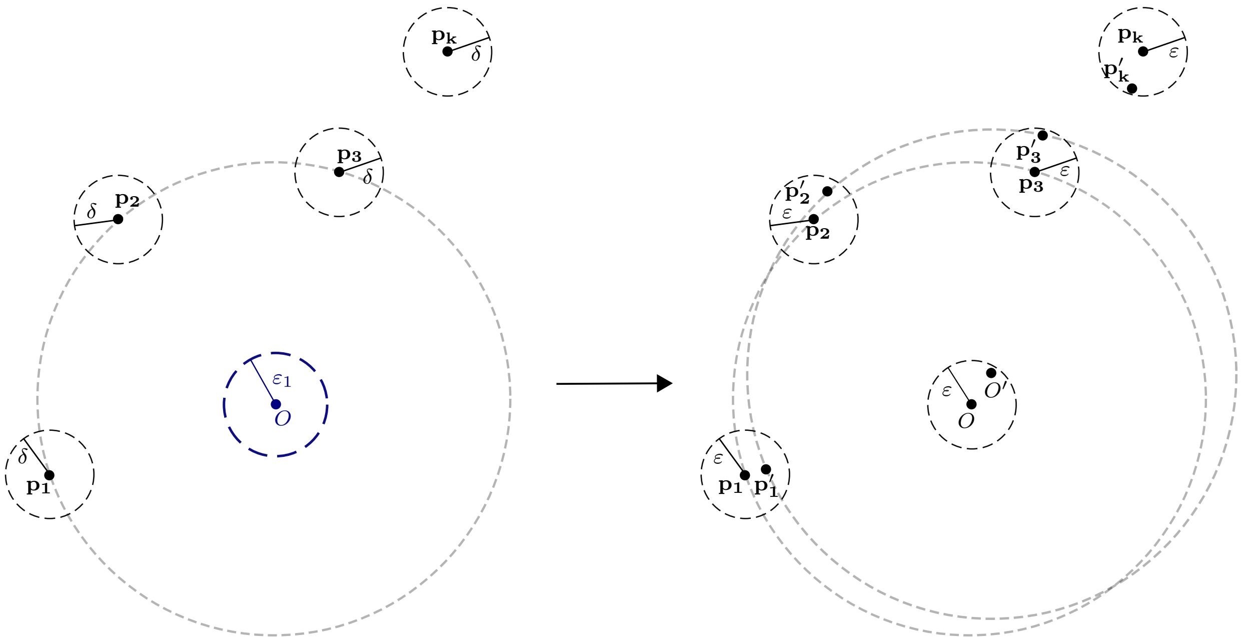

Figure 6: Maintaining a Delaunay triangle in $\mathbb{R}^2$. We take $\varepsilon = \min(\delta, \varepsilon_1)$. We illustrate in this figure how p_k' remains outside the circumcircle through p_1', p_2', and p_3'.

Lemma 3.8. *For a non-Delaunay simplex $D = \{p_1, p_2, \ldots, p_{d+1}\} \subset P$, there exists $\varepsilon > 0$ such that any ε-perturbation D' of D has $\{p_1', p_2', \ldots, p_{d+1}'\} \subset P'$ as a non-Delaunay simplex.*

Proof. An illustrated example can be found in figure 7. Let $\mathcal{S}$ be the unique $(d-1)$-sphere containing D with center O. Since D is a non-Delaunay simplex, there exists a $p_k \in P \backslash D$ such that $d(O,p_k) < d(O,p_1)$. Let $t := (d(O,p_1) - d(O,p_k))/4$ so that

$$d(O,p_k) + 2t = d(O,p_1) - 2t.$$

Since P is in general position, $t > 0$. Let $\varepsilon_1 = t$, then we obtain from Lemma 3.6 that there is $\delta > 0$ such that each $d(p_i,p_i') < \delta$ for $i = 1,2,\ldots,d+1$ implies $d(O,O') < \varepsilon_1$. Now, we take $\varepsilon = \min(\delta, \varepsilon_1)$ and D' is the ε-perturbation of D. Let $\mathcal{S}'$ be the unique $(d-1)$-sphere containing D' with center O'. Using the triangle inequality, we have the following chains of inequalities:

$$\begin{gathered}
d(O',p_k') \leq d(O',O) + d(O,p_k) + d(p_k,p_k') \\
< \varepsilon_1 + d(O,p_k) + \delta \leq t + d(O,p_k) + t \\
= d(O,p_k) + 2t = d(O,p_1) - 2t \\
= d(O,p_1) - t - t \leq d(O,p_1) - \varepsilon_1 - \delta \\
< d(O,p_1) - d(O,O') - d(p_1,p_1') \leq d(O',p_1').
\end{gathered}$$

Since $d(O',p_1')$ is the radius of $\mathcal{S}'$, we have that $p_k' \in \operatorname{int}(\mathcal{S}')$. Hence, D' is a non-Delaunay simplex. □

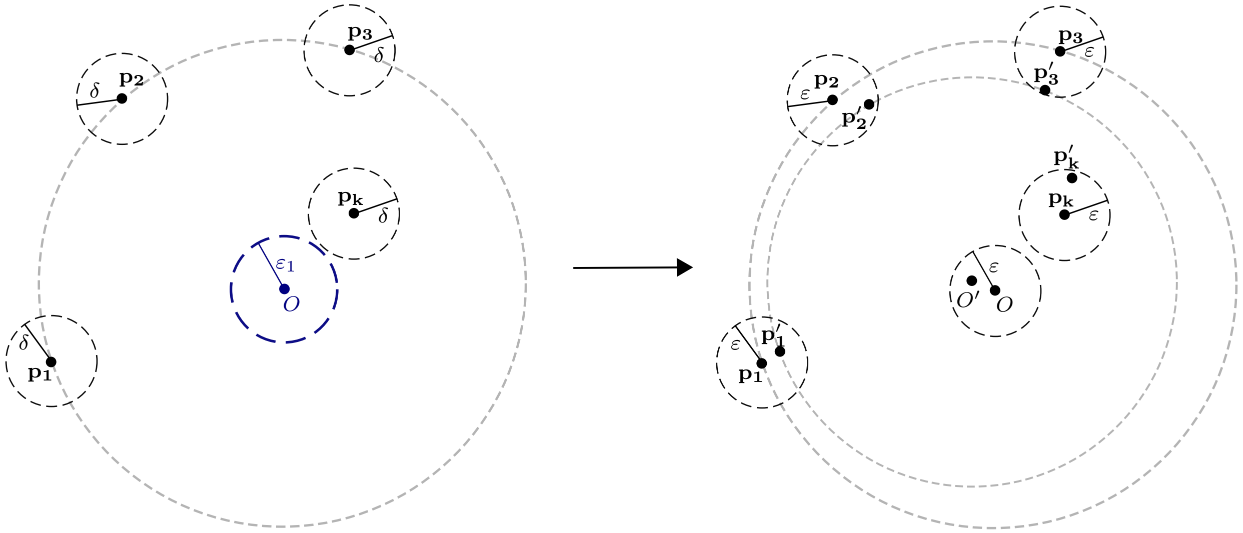

Figure 7: Maintaing a non-Delaunay triangle in $\mathbb{R}^2$. We take $\varepsilon = \min(\delta, \varepsilon_1)$. We illustrate in this figure how p_k' remains within the circumcircle through p_1', p_2', and p_3'.

Theorem 3.9. *Let $P \subset \mathbb{R}^d$ be in general position, that is no $d+2$ points lie on the same $(d-1)$-sphere and no $d+1$ points are affinely dependent. Then, there exists $\varepsilon > 0$ such that any ε-perturbation P' of P has the same Delaunay Triangulation as P. That is,*

$$Del(P) = Del(P').$$

Proof. Let D be a subset of P of size $d+1$. Depending if D is a Delaunay simplex in P, obtain ε_D from Lemma 3.7 or from Lemma 3.8. For any ε_D-perturbation D' of D, we have $Del(D) = Del(D')$. Now, find

$$\varepsilon = \min\{\varepsilon_D \mid \text{for every } D \subset P\}.$$

Hence, for any ε-perturbation P' of P, P' has the same Delaunay Triangulation as P. □

Now we know that for any point cloud in general position, there always exists an ε-perturbation that maintains the Delaunay Triangulation. With the underlying complex now fixed, we allow for the weights on the simplicies to change. This corresponds to choosing two monotonic functions on the same complex. Hence, we may apply the stability theorem for filtrations from [EH10].

Theorem 3.10. *Let K be a simplicial complex and $f, g : K \to \mathbb{R}$ two monotonic functions. For each dimension p, the bottleneck distance between the diagrams $X = Dgm_p(f)$ and $Y = Dgm_p(g)$ is bounded from above by the L_∞-distance between the functions,*

$$W_\infty(X, Y) \leq ||f - g||_\infty.$$

Hence, by application of this theorem, we conclude that the PDs produced using the Delaunay-Rips filtration are stable in an ε-perturbation of the original point cloud.

4 Application of Delaunay-Rips

1. Demonstrate value by talking about as dimensions change and number of points change.

2. Particular examples of how using special data sets affect the run-time of Rips/Alpha drastically but maybe not Del-Rips.

3. Performance: accuracy in ML algorithm, or classification. Instability may cause performance to go down even though run-time is unaffected.

4.1 Synthetic Data

Although we have special cases where instability in the PD may arise (as shown in section 3.4), we show here that this instability does not manifest itself when working with actual data sets. Our comparison is made in the following graphs. We compute the PD for a data set with 0.0001 noise (that is, the points are within a distance of 0.0001 of being exactly on the surface of the underlying shape). The tiny amount of noise makes it highly probable that our data set will be in general position. Then, we perturb the data set for increasing levels of noise and compute the PD for the perturbed data set using Delaunay-Rips, Rips, and Alpha. We plot the bottleneck distance between the PD of the perturbed data set with the PD of the original data set for each method. This captures how the PD evolves for increasing levels of noise in the data. The perturbation is made by moving each point in a random direction within a given magnitude (which is the noise level).

4.2 Machine Learning Classification

To test the robustness of Delaunay-Rips, we develop some ML classification models using Rips, Alpha, and Delaunay-Rips. Although the Persistence Diagrams produced using the Delaunay-Rips Filtration enjoy stability in a neighborhood (see Section 3.5), in reality, the underlying Delaunay Triangulation of the point cloud should be expected to change with varying levels of noise. The purpose of this section is to demonstrate how the instability of the Persistence Diagrams does not manifest itself in practical application. Here is the experiment:

1. Choose 6 shape classes of synthetic data: circle, sphere, torus, random, three clusters, and three clusters within three clusters.

2. Choose 3 filtration methods: Alpha, Rips, and Delaunay-Rips.

3. Fix a noise level ν and generate 100 point clouds for each shape class.

4. With the 600 point clouds, use one of the filtration methods to generate Persistence Diagrams for H_0, H_1, and H_2 classes for each point cloud.

5. Use a vectorization method to convert the PDs to a flattened vector. Concatenate the vectors corresponding to H_0, H_1, and H_2 diagrams into one vector for each point cloud. (This should produce 600 total vectors corresponding to each original point cloud.)

6. Build a dataframe with each vector labeled 0-5 where

$$0 : Circle,\ 1 : Sphere,\ 2 : Torus,\ 3 : Random,\ 4 : Clusters,\ 5 : Clusters\_in\_clusters.$$

7. Train an ML model on the 600 vectors to distinguish the 6 shape classes with 10-fold cross validation. Output the average accuracy of the model on the testing portion of the data.

8. Plot the average accuracy for increasing noise levels $\nu = 0.05, 0.10, ..., 0.75$. Thus, repeat steps 3-7.

9. Compare accuracy plots of using Alpha, Rips, and Delaunay-Rips as the filtration method to train ML model.

10. Further, change the ML model and the vectorization method (steps 5 and 7) to make a fair comparison of filtration method affecting the model accuracy rather than another factor.

One of our vectorization techniques was using Persistence Images from the python package "persim" from the scikit-tda library. Fixing a homology class and fixing a ν, we decided to fit a desired resolution to all of the PDs produced using Alpha, Rips, and Delaunay-Rips. For example, we iterate through all H_1 persistence diagrams for all three of our filtration methods that were produced using $\nu = 0.20$. Then, we find the maximum birth range and persistence range and use those values to set a 2×2 resolution grid to produce the Persistence Images. The purpose of doing this was to ensure that if we compared pixels of the PIs corresponding to different filtration methods, we would make a fair comparison because corresponding persistence pairs would land in corresponding pixels (we leverage this design in Figure 9 when comparing feature importance).

For our ML model, we decided to train a Random Forest Classifier from scikit-learn with 100 trees and minimum number of samples required to be at a leaf node set to 1. Figure 8 shows the median accuracy of our model for increasing noise levels (we plotted box-whisker plots to show the spread of the accuracy from 10-fold cross validation). As a baseline comparison, note that if our ML model was randomly classfying the test data, we expect to see accuracy of $1/6 \approx 17\%$. Since we are seeing over 70% median accuracy for each noise level, our model truly is finding distinguishing features between the 6 shape classes.

Further, we investigated the most important features that the Random Forest Classfier was detecting when constructing the model making classifications. We began with the PDs produced from the $\nu = 0.20$ noisy data. We generated our Persistence Images to have the following resolutions for each Persistence Diagram:

- H_0 diagram resolution: 5×1
- H_1 diagram resolution: 5×5
- H_2 diagram resolution: 5×5

Using this, each of our 600 vectors had a total of 55 features. We trained a Random Forest Classfier with the same parameters as before, but no cross-validation this time (or else we could not get a feature importance summary easily). Using a train-test split of 70-30, we generated the heatmaps in Figure 9. Notice how for the different filtrations we used to obtain the PDs, the PIs generated have similar corresponding pixels that the ML model found important. Remember that each pixel was a feature for our ML model. Hence, we have reason to believe that regardless of the filtration method used, our ML model learned the importance of similar features to classify the various datasets.

We also changed our vectorization technique to be persistence landscapes and our ML model to be a Support Vector Machine. Thus we assessed accuracy of our models with the following pairings:

1. persistence images + random forest
2. persistence images + SVM

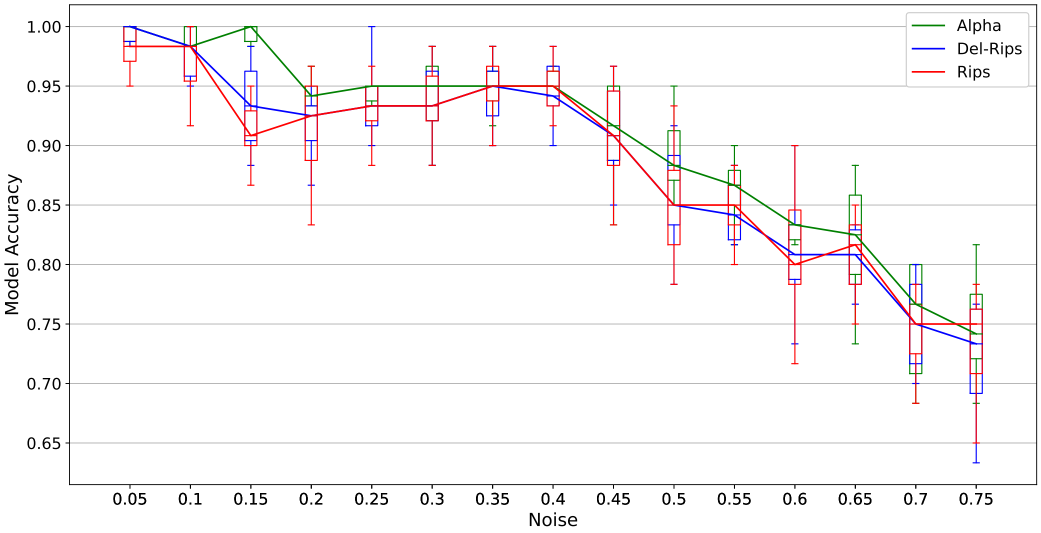

Figure 8

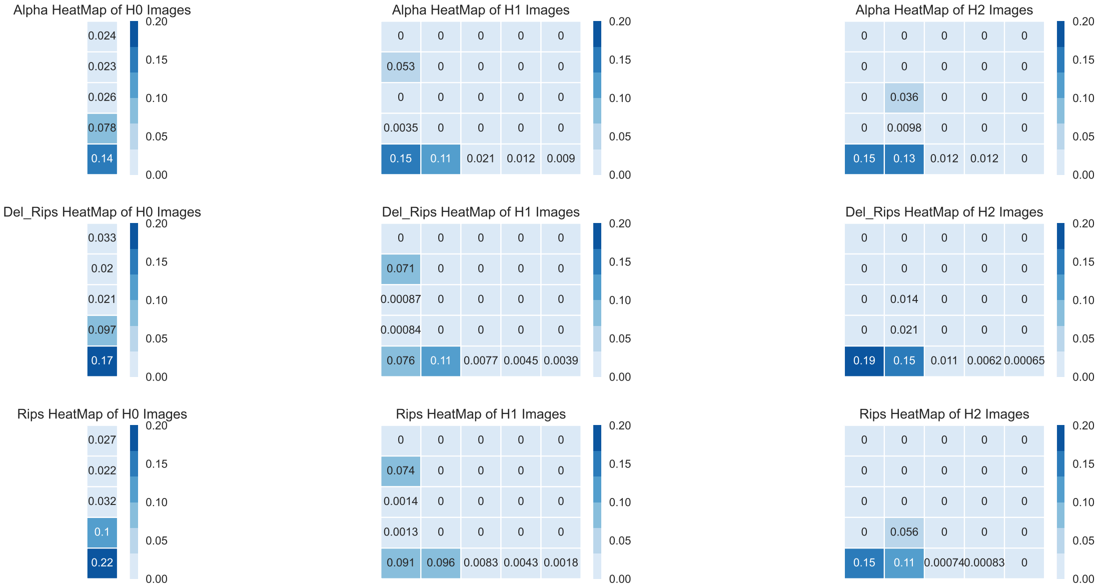

Figure 9

3. persistence landscapes + random forest

4. persistence landscapes + SVM

The results are summarized in Figure 10.

Figure 10

Highlight separately the speed up in the filtration step and then the PD step (since you have less simplices).

5 Conclusion

In this paper, we have defined the Delaunay-Rips complex and demonstrated its usage on some synthetic data. We compared the runtime of running the Delaunay-Rips method against Rips and Alpha on some synthetic data sets and showed the notable speed-up of Delaunay-Rips. We characterized the instability of this method by showing a 4 point example and carefully proving the instability in the corresponding persistence diagram for a small perturbation of the 4 points. We salvaged stability by imposing conditions on the data that would ensure the underlying Delaunay Triangulation would not change. This allowed us to leverage the stability theorem for Vietoris-Rips for Delaunay-Rips to guarantee stability in a neighborhood. We showed that practically, this instability is not a serious problem on data sets with sufficiently large number of points.

5.1 Further Questions

A Appendix: Math

A.1 Boundary Matrix Calculation for Instability

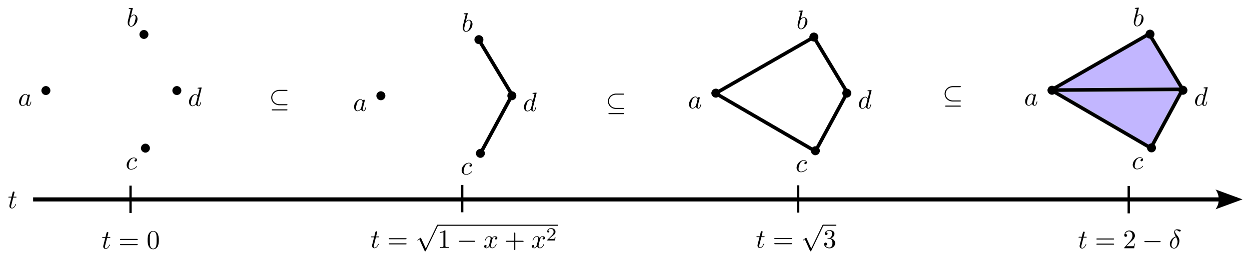

Figure 11: filtration

We have $P' = \{(-1, 0), (\frac{1}{2}, \frac{\sqrt{3}}{2}), (\frac{1}{2}, -\frac{\sqrt{3}}{2}), (1 - x, 0)\}$ with $0 < x < \delta < 2 - \sqrt{3}$. Our filtration has 4 key scale values, $t = 0 < \sqrt{1 - x + x^2} < \sqrt{3} < 2 - \delta$ as shown in Figure 11. We construct our boundary matrix B and reduce it to $\overline{B}$ using the standard algorithm:

$$B = \begin{array}{c} \\ a \\ b \\ c \\ d \\ bd \\ cd \\ ab \\ ac \\ ad \\ abd \\ acd \end{array} \begin{array}{c} \begin{array}{ccccccccccc} a & b & c & d & bd & cd & ab & ac & ad & abd & acd \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

$$\overline{B} = \begin{array}{c} \\ a \\ b \\ c \\ d \\ bd \\ cd \\ ab \\ ac \\ ad \\ abd \\ acd \end{array} \begin{array}{c} \begin{array}{ccccccccccc} a & b & c & d & bd & cd & ab & ac & ad & abd & acd \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}.$$

The persistence pairs for the H_0 class with their persistence diagram coordinate (birth/death pair) come out as follows:

$$(a, N/A) : (0, \infty)$$

$$(b, ab) : (0, \sqrt{3})$$

$$(c, cd) : (0, \sqrt{1 - x + x^2})$$

$$(d, bd) : (0, \sqrt{1 - x + x^2}).$$

The H_1 classes come out as

$$(ad, abd) : (2 - x, 2 - x)$$

$$(ac, acd) : (\sqrt{3}, 2 - x).$$

The only point with non-zero persistence is $(\sqrt{3}, 2 - x)$.

A.2 Continuity of Matrix Inverse

Let $V = (\mathcal{M}_n(\mathbb{R}), ||\cdot||)$ be the normed vector space of $n \times n$ matrices over $\mathbb{R}$ with the Frobenius norm defined as

$$||A||_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}^2}$$

where a_{ij} is the entry in the ith row and jth column of A. Recall the definition of the vector 2-norm

$$||v||_2 = \sqrt{\sum_{i=1}^{n} |v_i|^2}$$

for $v \in \mathbb{R}^n$. We use the vector 2-norm with the Frobenius matrix norm because of their compatibility: that is,

$$||Mv||_2 \leq ||M||_F ||v||_2.$$

We proceed by dropping the subscripts and using $||\cdot||$ for matrix and vector norms.

The heavy machinery used to establish the continuity of the matrix inverse was done in [Wil94] pages 92-93. However, their proof did not explicitly compute inequalities involving the Frobenius norm (that is, $||I|| = \sqrt{n}$ for $I \in V$ being the identity). For completeness, we provide derivations of similar results using the Frobenius norm.

Lemma A.1. *Given $A, E \in V$ with A invertible, the matrix $A + E$ is invertible if*

$$||A^{-1}E|| < 1.$$

Proof. We mimic the proof found in [Wil94] page 92. We write

$$A + E = A(I + A^{-1}E) \tag{4}$$

to see that $A + E$ is invertible if $I + A^{-1}E$ is invertible. From [Wil94] page 82, if λ is an eigenvalue of M, then

$$||M|| \geq |\lambda|.$$

Thus, if λ_i is an eigenvalue of $A^{-1}E$, then we have

$$||A^{-1}E|| \geq |\lambda_i|.$$

Since the eigenvalues of $I + A^{-1}E$ are of the form $1 + \lambda_i$, to ensure $I + A^{-1}E$ is invertible, we need $\lambda_i \neq -1$ for any i. One way to ensure this is to require $||A^{-1}E|| < 1$ so that

$$|\lambda_i| \leq ||A^{-1}E|| < 1.$$

This condition on $||A^{-1}E||$ guarantees the invertibility of $A + E$. □

Lemma A.2. *Given $A, E \in V$ with A invertible, let $x, b \in \mathbb{R}^n$ such that $Ax = b$ and $||A^{-1}||\ ||E|| < 1$. Then, there exists $h \in \mathbb{R}^n$ such that $(A + E)(x + h) = b$ and*

$$||(A + E)^{-1} - A^{-1}|| \leq \frac{\sqrt{n}||A^{-1}||^2\ ||E||}{1 - ||A^{-1}||\ ||E||}.$$

Proof. We mimic the proof found in [Wil94] pages 92-93. Begin by defining

$$F := A^{-1}E$$

$$G := (I + F)^{-1}.$$

Note that since $||A^{-1}E|| \leq ||A^{-1}||\ ||E|| < 1$, we have that $(I + F)$ is invertible by the proof of Lemma A.1. Hence, G indeed is well-defined. Multiplying each side by $(I + F)$ results in

$$I = G + FG$$

and using the reverse triangle inequality yields

$$||I|| = ||G + FG|| = ||G - (-FG)|| \geq ||G|| - || - FG|| = ||G|| - ||FG|| \geq ||G|| - ||F||\ ||G||$$

$$\sqrt{n} \geq ||G|| - ||F||\ ||G||$$

since $||I|| = \sqrt{n}$. Since $||F|| = ||A^{-1}E|| < 1$, we have

$$||G|| \leq \frac{\sqrt{n}}{1 - ||F||}.$$

We will use this inequality in a bit. Now, we rewrite $(A + E)(x + h) = b$ as

$$(A + E)h = -Ex.$$

Applying Lemma A.1, we have $A + E$ is invertible and using equation 4, we have the existence of h:

$$h = -(A + E)^{-1}Ex = -(I + F)^{-1}A^{-1}Ex = -GA^{-1}Ex.$$

Further,

$$||h|| \leq ||G||\ ||A^{-1}E||\ ||x|| \leq \frac{\sqrt{n}||A^{-1}E||\ ||x||}{1 - ||F||} = \frac{\sqrt{n}||A^{-1}E||\ ||x||}{1 - ||A^{-1}E||} \leq \frac{\sqrt{n}||A^{-1}||\ ||E||\ ||x||}{1 - ||A^{-1}||\ ||E||}$$

using the hypothesis that $||A^{-1}||\ ||E|| < 1$. Division by $||x||$ results in

$$\frac{||h||}{||x||} \leq \frac{\sqrt{n}||A^{-1}||\ ||E||}{1-||A^{-1}||\ ||E||}.$$

Using the substitutions

$$x = A^{-1}b,$$
$$h = (A+E)^{-1}b - x = (A+E)^{-1}b - A^{-1}b = ((A+E)^{-1} - A^{-1})b$$

we have

$$\frac{||((A+E)^{-1} - A^{-1})b||}{||A^{-1}||||b||} \leq \frac{||h||}{||x||} \leq \frac{\sqrt{n}||A^{-1}||\ ||E||}{1-||A^{-1}||\ ||E||}$$
$$\frac{||((A+E)^{-1} - A^{-1})b||}{||b||} \leq \frac{\sqrt{n}||A^{-1}||^2\ ||E||}{1-||A^{-1}||\ ||E||}.$$

Since this inequality is true for any b, we have that

$$\sup\left\{\frac{||((A+E)^{-1} - A^{-1})b||}{||b||}\ :\ b \neq 0\right\} \leq \frac{\sqrt{n}||A^{-1}||^2\ ||E||}{1-||A^{-1}||\ ||E||}.$$

It follows by the definition of the operator norm that

$$||(A+E)^{-1} - A^{-1}|| \leq \frac{\sqrt{n}||A^{-1}||^2\ ||E||}{1-||A^{-1}||\ ||E||}.$$

□

Theorem A.3. *Define the inverse of a matrix as the map*

$$\iota : V \to V$$
$$\iota(A) = A^{-1}.$$

This map is continuous in the topology on V induced by the Frobenius norm.

Proof. Let $\varepsilon > 0$ and $A \in V$ be arbitrary and invertible and choose $\delta = \min(\varepsilon/(2\sqrt{n}||A^{-1}||^2), 1/(2||A^{-1}||))$. We find $E \in V$ so that

$$||(A+E) - A|| < \delta.$$

Hence, we have

$$||(A+E) - A|| = ||E|| < \delta \leq 1/(2||A^{-1}||)$$

so that

$$||A^{-1}||\ ||E|| < 1/2 < 1.$$

Thus, Lemma A.2 applies and we have

$$||\iota(A+E) - \iota(A)|| = ||(A+E)^{-1} - A^{-1}|| \leq \frac{\sqrt{n}||E||\ ||A^{-1}||^2}{1-||E||\ ||A^{-1}||} \leq \frac{\sqrt{n}||A^{-1}||^2}{1-1/2}||E||$$
$$= 2\sqrt{n}||A^{-1}||^2||E|| < 2\sqrt{n}||A^{-1}||^2\delta \leq \varepsilon.$$

Since A was arbitrary, we have that ι (the matrix inverse) is a continuous function in the topology induced by the Frobenius norm. □

B Appendix: Tech

B.1 Pseudo-code Implementation

B.2 Github Repo of Actual, Clean Code

1. We want to compare the best implementation of Del-Rips with Ripser and Cechmate's Alpha.

Algorithm 1: An algorithm to compute Delaunay-Rips Filtration

```
Input: $P = (\mathbf{p_1}, \ldots, \mathbf{p_n})$, $dim$ = maximum homology dimension to compute
Output: Delaunay Rips Filtration
triangulation ← Delaunay($P$)
filtration ← $[([i], 0)]$ for $1 \leq i \leq n$ /* Add the 0-simplices into the filtration */
for each simplex ∈ triangulation do
    for $d \leftarrow 1$ to $dim + 1$ do
        $faces$ ← all $d$-simplex subsets of current simplex
        for each face ∈ faces do
            if face ∉ filtration and $d == 1$ then
                $value$ ← distance($face[0]$, $face[1]$) /* Calculate the euclidean distance
                    between the two points */
                filtration.append(($face$, $value$)) /* Append 1-simplices */
            end
            if face ∉ filtration and $dim > 1$ then
                find $subface$ of $face$ with greatest $value$
                filtration.append(($subface$, $value$)) /* Add higher order simplices */
            end
        end
    end
end
```

B.3 Machine Specs

1. Eluktronics laptop

Whole bibliography

[Wil94] James H. Wilkinson. *Rounding Errors in Algebraic Processes.* USA: Dover Publications, Inc., 1994. ISBN: 0486679993.

[EH10] Herbert Edelsbrunner and John Harer. *Computational Topology: An Introduction.* Jan. 2010. ISBN: 978-0-8218-4925-5. DOI: 10.1007/978-3-540-33259-6_7.

[BE16] Ulrich Bauer and Herbert Edelsbrunner. “The Morse theory of Čech and Delaunay complexes”. In: *Transactions of the American Mathematical Society* 369.5 (Dec. 2016), pp. 3741–3762. ISSN: 1088-6850. DOI: 10.1090/tran/6991. URL: http://dx.doi.org/10.1090/tran/6991.

[Ott+17] Nina Otter et al. “A roadmap for the computation of persistent homology”. In: *EPJ Data Science* 6.17 (2017). DOI: https://doi.org/10.1140/epjds/s13688-017-0109-5.