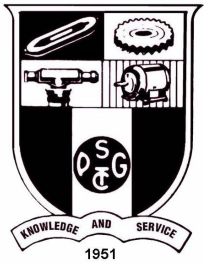
# Approximation of Persistence Homology for Large Point Clouds

**Karthikeya Subramanian**
**(Roll No. 21PA05)**

A Dissertation Submitted
in Partial Fulfilment of the Requirements
for the Degree of

## M.Sc. APPLIED MATHEMATICS
of Anna University




**July 2024**

DEPARTMENT OF APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCES

**PSG COLLEGE OF TECHNOLOGY**
(Autonomous Institution)
**COIMBATORE – 641 004.**

**PSG COLLEGE OF TECHNOLOGY**
(Autonomous Institution)
**COIMBATORE – 641 004.**

**Fourth Semester**
**Project Work**

**APPROXIMATION OF PERSISTENCE HOMOLOGY FOR LARGE POINT CLOUDS**

Bona fide record of work done by

**Karthikeya Subramanian**
**(Roll No. 21PA05)**

Submitted in Partial Fulfilment of the Requirements for the Degree of

# M.Sc. APPLIED MATHEMATICS

of Anna University

**July 2024**

| | |
|---|---|
| **Dr. Sai Sundara Krishnan** | **Dr.Shina Sheen** |
| Academic Guide | Head of the Department |

Submitted for Viva - Voice Examination held on _____

| | |
|---|---|
| **Internal Examiner** | **External Examiner** |

# Contents

# List of Figures

# ACKNOWLEDGEMENT

I am extremely grateful to **Dr. Prakasan K**, Principal, PSG College of Technology, for giving me this opportunity to do my project at IIIT, Bangalore.

I am deeply indebted to **Dr. Shina Sheen**, Professor and Head, Department of Applied Mathematics and Computational Sciences, for her continual support and ardent motivation.

I am indebted to **Dr. Sai Sundara Krishnan G**, Professor, Programme Coordinator, Department of Applied Mathematics and Computational Sciences, for his encouragement and persistent support.

I extend my gratitude to my tutor **Dr. Jeevadoss S**, Assistant Professor, Department of Applied Mathematics and Computational Sciences, for his guidance and support.

I am highly obliged to my academic guide **Dr. Sai Sundara Krishnan G**, Department of Applied Mathematics and Computational Sciences,for his guidance and expertise have been instrumental in shaping my research journey and continual support throughout my project tenure.

I would like to express my sincere gratitude to **Dr. Amit Chattopadhyay**, my esteemed mentor and Professor at IIIT-Bangalore, for his unwavering support, guidance, and motivation throughout my project. Without his invaluable insights and encouragement, I would not have been able to successfully complete this endeavor. I feel truly blessed to have had the privilege of learning from such an accomplished and inspiring mentor.

Finally, I express my sincere gratitude to all the staff of Department of Applied Mathematics and Computational Sciences, PSG College of Technology, Coimbatore and my family and friends for their encouragement and support.

# ABSTRACT

Topological data analysis (TDA) is a field that has recently emerged as a powerful approach to address modern data challenges. It utilizes algebraic topology to handle high dimensionality and structural complexity in diverse contexts. One of the most successful TDA tools is persistent homology, which employs the theory of homology to generate summaries of a dataset that capture its shape and size. The output of persistent homology is a persistence diagram, which encapsulates all the topological information of the data. This technique has been widely applied in various fields, such as biomedical imaging, information retrieval, machine learning, materials science, neuroscience and sensor networks.

However, one of the significant challenges in using persistent homology is its computational expense, which is known to be intensive. Despite recent advances, the fundamental nature of the procedure makes it largely non-parallelizable. For many large datasets that are now readily obtainable with modern powerful data acquisition techniques, computing persistent homology is often impossible. To address this issue, this project focuses on a proposed method that uses bootstrapping to compute persistent homology while taking the overall mean and their practical performance will be explored. This approach aims to explore new avenues for computing persistent homology in a computationally efficient manner.

# Chapter 1

# Basics of Topological Data Analysis

## 1.1 Complexes

There are many ways to represent a topological space, one being a decomposition into simple pieces. This decomposition qualifies to be called a complex if the pieces are topologically simple and their common intersections are lower-dimensional pieces of the same kind. Within these requirements, we still have a great deal of freedom. Particularly attractive are the extreme choices: few complicated or many simple pieces. The former choice lends itself to hand-calculations of topological invariants but also to the design of aesthetically pleasing shapes, such as car bodies and the like. The latter choice is preferred in computation and automation. Since we focus on computational aspects of topology, we favor the latter extreme choice of which the simplicial complex is the prime example. [1]

### 1.1.1 Simplices

A simplex is a geometric object defined as the convex hull of a set of affinely independent points in Euclidean space.

In formal notation, an n-simplex is defined as:

$$\Delta^n = \{\sum_{i=1}^{n} \lambda_i v_i \mid \sum_{i=1}^{n} \lambda_i = 1 \ , \lambda_i \geq 0\} \tag{1.1}$$

where $v_0, \ldots, v_n$ are affinely independent points in Euclidean space. The points $v_0, \ldots, v_n$ are called the vertices of the simplex $\Delta^n$, and the coefficients $\lambda_0, \ldots, \lambda_n$ are called the barycentric coordinates of a point in the simplex. The dimension of the simplex is $n$, and it has $n+1$ vertices. For example, a 0-simplex is a point, a 1-simplex is a line segment, a 2-simplex is a triangle, and a 3-simplex is a tetrahedron.

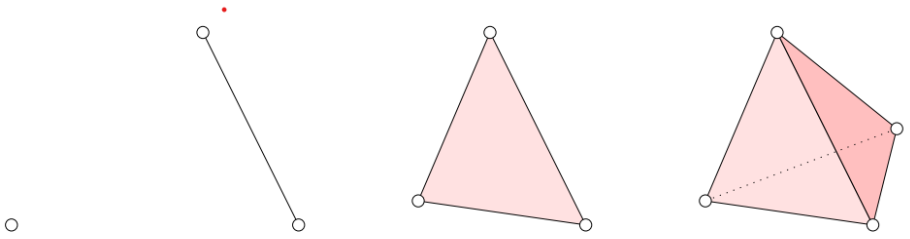


Figure 1.1: Simplex (as given in [1])

A face of $\sigma$ is the convex hull of a non-empty subset of the $u_i$ and it is proper if the subset is not the entire set. We sometimes write $\tau \leq \sigma$ if $\tau$ is a face and $\tau < \sigma$ if it is a proper face of $\sigma$. If $\tau$ is a (proper) face of $\sigma$ we call $\sigma$ a (proper) co-face of $\tau$. Since a set of size $k+1$ has $2^{k+1}$ subsets, including the empty set, $\sigma$ has $2^{k+1} - 1$ faces, all of which are proper except for $\sigma$ itself. The boundary of $\sigma$, denoted as $bd(\sigma)$, is the union of all proper faces, and the interior is everything else, $int(\sigma) = \sigma - bd(\sigma)$. A point $x \in \sigma$ belongs to $int(\sigma)$ if all its coefficients $\lambda_i$

are positive. It follows that every point $x \in \sigma$ belongs to the interior of exactly one face, namely the one spanned by the points $u_i$ that corresponds to positive coefficients $\lambda_i$.

### 1.1.2 Simplicial Complex

A simplicial complex is a finite collection of simplices $K$ such that $\sigma \in K$ and $\tau \leq \sigma$ implies $\tau \in K$, and $\sigma, \sigma_0 \in K$ implies $\sigma \cap \sigma_0$ is either empty or a face of both.

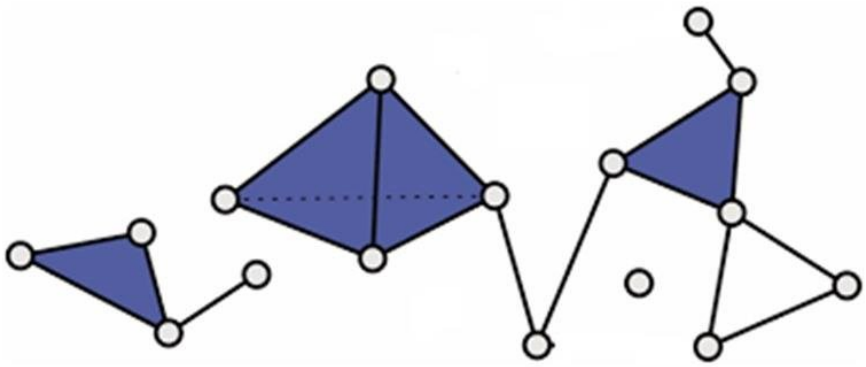


Figure 1.2: Simplicial Complex (as given in [1])

## 1.2 Homology

Homology groups provide a mathematical language for the holes in a topological space. Perhaps surprisingly, they capture holes indirectly, by focusing on what surrounds them. Their main ingredients are group operations and maps that relate topologically meaningful subsets of a space with each other. This section focuses on the various groups involved in the setup. [1]

**Chain Complexes**

Let $K$ be a simplicial complex and $p$ a dimension. A $p$-chain is a formal sum of $p$-simplices in $K$. The standard notation for this is $c = \sum_i a_i \sigma_i$, where the $\sigma_i$ are the $p$-simplices and the $a_i$ are the coefficients. In computational topology, we mostly work with coefficients $a_i$ that are either 0 or 1, called modulo 2 coefficients. Coefficients can, however, be more complicated numbers like integers, rational numbers, real numbers, elements of a field, or elements of a ring. Since we are working modulo 2, we can think of a chain as a set of $p$-simplices, namely those $\sigma_i$ with $a_i = 1$. But when we do consider chains with other coefficient groups, this way of thinking is more cumbersome, so we will use it sparingly.

Two $p$-chains are added componentwise, like polynomials. Specifically, if $c = \sum_i a_i \sigma_i$ and $c' = \sum_i b_i \sigma_i$ then $c+c' = \sum_i (a_i+b_i)\sigma_i$, where the coefficients satisfy $1+1 = 0$. In set notation, the sum of two $p$-chains is their symmetric difference. The $p$-chains together with the addition operation form the group of $p$-chains denoted as $(C_p, +)$ , or simply $C_p = C_p(K)$ if the operation is understood. Associativity follows from associativity of addition modulo 2. The neutral element is $0 = \sum_i 0\sigma_i$. The inverse of $c$ is $-c = c$ since $c + c = 0$. Finally, $C_p$ is abelian because addition modulo 2 is abelian. We have a group of $p$-chains for each integer $p$. For $p$ less than zero and greater than the dimension of $K$, this group is trivial, consisting only of the neutral element. To relate these groups, we define the boundary of a $p$-simplex as the sum of its $(p-1)$-dimensional faces. Writing $\sigma = [u_0, u_1, \ldots, u_p]$ for the simplex spanned by the listed vertices, its boundary is

$$\partial_p \sigma = \sum_{j=0}^{p} [u_0, \ldots, \hat{u}_j, \ldots, u_p],$$

where the hat indicates that $u_j$ is omitted. For a $p$-chain, $c = \sum a_i \sigma_i$, the

boundary is the sum of the boundaries of its simplices, $\partial_p c = \sum a_i \partial_p \sigma_i$. Hence, taking the boundary maps a $p$-chain to a $(p-1)$-chain, and we write $\partial_p : C_p \to C_{p-1}$. Notice also that taking the boundary commutes with addition, that is, $\partial_p(c + c') = \partial_p c + \partial_p c'$. This is the defining property of a homomorphism, a map between groups that commutes with the group operation. We will therefore refer to $\partial_p$ as the boundary homomorphism or, shorter, the boundary map for chains. The chain complex is the sequence of chain groups connected by boundary homomorphisms,

$$\cdots \xrightarrow{\partial_{p+2}} C_{p+1} \xrightarrow{\partial_{p+1}} C_p \xrightarrow{\partial_p} C_{p-1} \xrightarrow{\partial_{p-1}} \cdots$$

It will often be convenient to drop the index from the boundary homomorphism since it is implied by the dimension of the chain it applies to.

**Cycles and boundaries** We distinguish two particular types of chains and use them to define homology groups. A $p$-cycle is a $p$-chain with empty boundary, $\partial c = 0$. Since $\partial$ commutes with addition, we have a group of $p$-cycles, denoted as $Z_p = Z_p(K)$, which is a subgroup of the group of $p$-chains. In other words, the group of $p$-cycles is the kernel of the $p$-th boundary homomorphism, $Z_p = \ker \partial_p$. Since the chain groups are abelian so are their cycle subgroups. Consider $p = 0$ as an example. The boundary of every vertex is zero ($C_{-1} = 0$), hence, $Z_0 = \ker \partial_0 = C_0$. For $p > 0$, however, $Z_p$ is usually not all of $C_p$.

A $p$-boundary is a $p$-chain that is the boundary of a $(p+1)$-chain, $c = \partial d$ with $d \in C_{p+1}$. Since $\partial$ commutes with addition, we have a group of $p$-boundaries, denoted by $B_p = B_p(K)$, which is again a subgroup of the $p$-chains. In other words, the group of $p$-boundaries is the image of the $(p+1)$-st boundary homomorphism, $B_p = \operatorname{im} \partial_{p+1}$. Since the chain groups are abelian so are their boundary subgroups.

Consider $p = 0$ as an example. Every 1-chain consists of some number of edges, each with two endpoints. Taking the boundary cancels duplicate endpoints in pairs, leaving an even number of distinct vertices. Now suppose the complex is connected. Then for any even number of vertices, we can find paths that connect them in pairs and we can add the paths to get a 1-chain whose boundary consists of the given vertices. Hence, every even set of vertices is a 0-boundary and every odd set of vertices is not. If $K$ is connected this implies that exactly half the 0-cycles are 0-boundaries. The fundamental property that makes homology work is that the boundary of a boundary is necessarily zero.

**Fundamental Lemma of Homology**. $\partial_p \partial_{p+1} d = 0$ for every integer $p$ and every $(p+1)$-chain $d$.

Proof. We just need to show that $\partial_p \partial_{p+1} \tau = 0$ for a $(p+1)$-simplex $\tau$. The boundary, $\partial_{p+1}\tau$, consists of all $p$-faces of $\tau$. Every $(p-1)$-face of $\tau$ belongs to exactly two $p$-faces, so $\partial_p(\partial_{p+1}\tau) = 0$.
It follows that every $p$-boundary is also a $p$-cycle or, equivalently, that $B_p$ is a subgroup of $Z_p$. Figure 1.3 illustrates the subgroup relations among the three types of groups and their connection across dimensions established by the boundary homomorphisms.

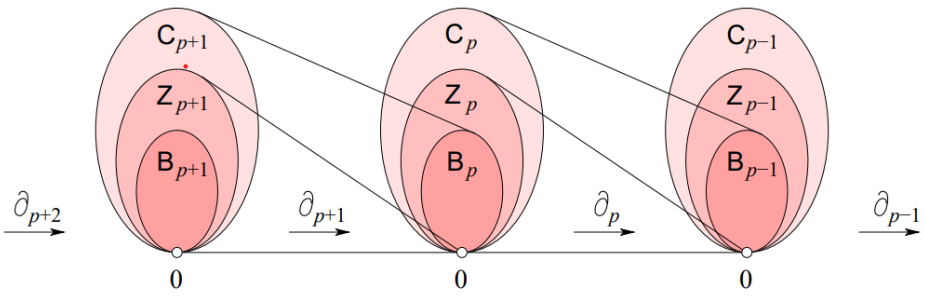



Figure 1.3: Chain Complex Homomorphisms (as given in [1])

**Homology groups** Since the boundaries form subgroups of the cycle groups, we can take quotients. In other words, we can partition each cycle group into classes of cycles that differ from each other by boundaries. This leads to the notion of homology groups and their ranks, which we now define and discuss.

**Definition.** The $p$-th homology group is the $p$-th cycle group modulo the $p$-th boundary group, $H_p = Z_p/B_p$. The $p$-th Betti number is the rank of this group, $\beta_p = \text{rank } H_p$.

Each element of $H_p = H_p(K)$ is obtained by adding all $p$-boundaries to a given $p$-cycle, $c + B_p$ with $c \in Z_p$. If we take any other cycle $c' = c + c''$ in this class, we get the same class, $c' + B_p = c + B_p$, since $c'' + B_p = B_p$ for every $c'' \in B_p$. This class is thus a coset of $H_p$ and is referred to as a homology class. Any two cycles in the same homology class are said to be homologous, which is denoted as $c \sim c'$. We may take $c$ as the representative of this class but any other cycle in the class does as well. Similarly, addition of two classes, $(c+B_p)+(c_0+B_p) = (c+c_0)+B_p$, is independent of the representatives and is therefore well-defined. We thus see that $H_p$ is indeed a group, and because $Z_p$ is abelian so is $H_p$.

The cardinality of a group is called its order. Since we use modulo 2 coefficients, a group with $n$ generators has order $2^n$. For example, the base 2 logarithm of the order of $C_p$ is the number of $p$-dimensional simplices in the complex. Furthermore, the group is isomorphic to $\mathbb{Z}_2^n$, the group of bit-vectors of length $n$ together with the exclusive-or operation. This is an $n$-dimensional vector space generated by $n$ bit-vectors, for example the $n$ unit vectors. The dimension is referred to as the rank of the vector space, $n = \text{rank}\,\mathbb{Z}_2^n = \log_2 \text{ord}\,\mathbb{Z}_2^n$. The cycles and boundaries exhibit the same vector space structure, except that their dimension is often less

than that of the chains. The number of cycles in each homology class is the order of $B_p$, hence the number of classes in the homology group is $\operatorname{ord} H_p = \operatorname{ord} \mathbb{Z}_p / \operatorname{ord} B_p$. (The above theory and definitions are from [1]).

# Chapter 2

# Filtrations and Persistence

A filtration is a sequence of topological spaces that captures the evolution of a space over some parameter such as distance, scale, or density. The sequence starts from some initial space and proceeds to some final space, with intermediate spaces representing intermediate stages of the evolution. For example, a filtration can be constructed by considering a set of nested subsets of a topological space, where each subset corresponds to a level set of some function defined on the space. Alternatively, a filtration can be constructed by considering a set of geometric shapes (e.g., balls or simplices) centered around each data point, where the radius of each shape is determined by some distance metric or density function. The idea is to capture the topological features of the space at different levels of resolution, and to track how these features evolve as the parameter changes. Filtrations are often used in conjunction with persistent homology to identify and quantify topological features that are persistent across multiple scales or parameters.[1]

## 2.1 $\check{C}$ech complexes

Consider a case in which the convex sets are closed geometric balls, all of the same radius $r$. Let $S$ be a finite set of points in $\mathbb{R}^d$ and write $B_x(r) = x + rB_d$ for the closed ball with center $x$ and radius $r$. The Cech complex $\check{C}(S, r)$ of $S$ and $r$ is isomorphic to the nerve of this collection of balls.

$$\check{C}ech(r) = \{\sigma \subseteq S \mid \bigcap_{x \in \sigma} B_x(r) \neq \varnothing\}$$

Clearly, a set of balls has a non-empty intersection iff their centers lie inside a common ball of the same radius. Indeed, a point $y$ belongs to all balls iff $\|x-y\| \leq r$ for all centers $x$. An easy consequence of Helly's Theorem is therefore that every $d+1$ points in $S$ are contained in a common ball of radius $r$ iff all points in $S$ are. This is Jung's Theorem which predates the more general theorem by Helly. The Čech complex does not necessarily have a geometric realization in $\mathbb{R}^d$ but it is fine as an abstract simplicial complex; see Figure 2.1. For larger radius, the disks are bigger and create more overlaps while.

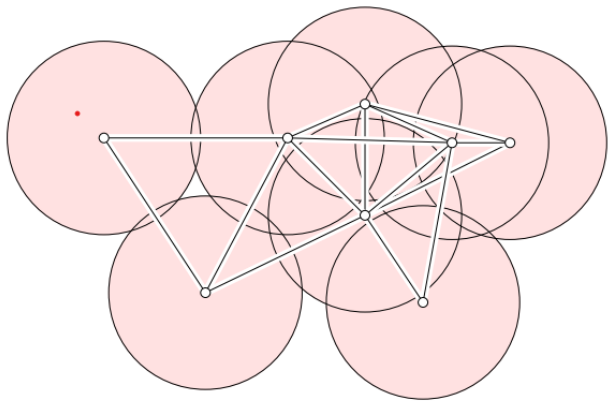


Figure 2.1: Cech Filteration (as given in [1])

Retaining the ones for smaller radius. Hence $Cech(\check{C}_{r_0}) \subseteq Cech(\check{C}_r)$ whenever $r_0 \leq r$. If we continuously increase the radius, from 0 to $\infty$, we get a discrete family of nested Čech complexes.

### 2.1.1 Smallest enclosing balls

Beyonds sets of two points it seems cumbersome to recognize the ones that form simplices in the Cech complex. Nevertheless, ˇ there is a fast algorithm for the purpose. Let $\sigma \subseteq S$ be a subset of the given points. We have seen that deciding whether or not $\sigma$ belongs to $Cech(\check{r})$ is equivalent to deciding whether or not $\sigma$ fits inside a ball of radius $r$. Let the miniball of $\sigma$ be the smallest closed ball that contains $\sigma$, which we note is unique. The radius of the miniball is smaller than or equal to $r$ iff $\sigma \in Cech(\check{r})$, so finding it solves our problem.

Observe that the miniball is already determined by a subset of $k+1 \leq d+1$ of the points, which all lie on its boundary. If we know this subset then we can verify the miniball by testing that it indeed contains all the other points. In a situation in which we have many more points than dimensions, the chance that a point belongs to this subset is small and discarding it is easy. This is the strategy of the Miniball Algorithm. It takes two disjoint subsets $\tau$ and $\upsilon$ of $\sigma$ and returns the miniball that contains all points of $\tau$ in its interior and all points of $\upsilon$ on its boundary. To get the miniball of $\sigma$ we call MiniBall$(\sigma, \emptyset)$.

When $\tau$ is empty, we have a set $\upsilon$ of at most $d+1$ points, which we know all lie on the boundary. Assuming the dimension, $d$, is a constant, we can compute their miniball directly and in constant time. To analyze the running time, we ask how often we execute the test "$u \notin B$". Let $t_j(n)$ be the expected number of such tests for calling MiniBall with $n$ points in $\tau$ and $j = d+1-|\upsilon|$ possibly open positions

**Algorithm 1** MiniBall($\tau, \upsilon$)

```
1: if τ = ∅ then
2:     compute the miniball B of υ directly
3: else
4:     choose a random point u ∈ τ
5:     B =MiniBall(τ − u, υ)
6:     if u ∉ B then
7:         B = MiniBall(τ − u, υ ∪ u)
8:     end if
9: end if
10: return B
```

on the boundary of the miniball. Obviously, $t_j(0) = 0$, and it is reassuring that the constant amount of work needed to compute the ball for the at most $d + 1$ points in $\upsilon$ is paid for by the test that initiated the call. Consider $n > 0$. We have one call with parameters $n - 1$ and $j$, one test "$u \notin B$", and one call with parameters $n - 1$ and $j - 1$. The probability that the second call indeed happens is at most $j$ out of $n$. Hence,

$$t_j(n) \leq t_j(n-1) + 1 + \frac{j}{n} \cdot t_{j-1}(n-1)$$

Setting $j = 0$ we get $t_0(n) \leq t_0(n-1) + 1$ and therefore $t_0(n) \leq n$. Similarly, $t_1(n) \leq t_1(n-1) + 2 \leq 2n$. More generally, we get $t_j(n) \leq (j+1)!n$, which is a constant times $n$ since $j \leq d+1$ is a constant. In summary, for constant dimension the algorithm takes expected constant time per point.

### 2.1.2 Vietoris-Rips complexes

Instead of checking all sub-collections, we may just check pairs and add 2- and higher-dimensional simplices whenever we can. This simplification leads to the Vietoris-Rips complex of $S$ and $r$ consisting of all subsets of diameter at most $2r$,

$$\text{Vietoris-Rips}(r) = \sigma \subseteq S \mid \text{diam}(\sigma) \leq 2r$$

Clearly, the edges in the Vietoris-Rips complex are the same as in the Cech complex. Furthermore, Cech($\check{r}$) $\subseteq$ Vietoris-Rips($r$) because the latter contains every simplex warranted by the given edges. We now prove that the containment relation can be reversed if we are willing to increase the radius of the Cech complex by a multiplicative constant.

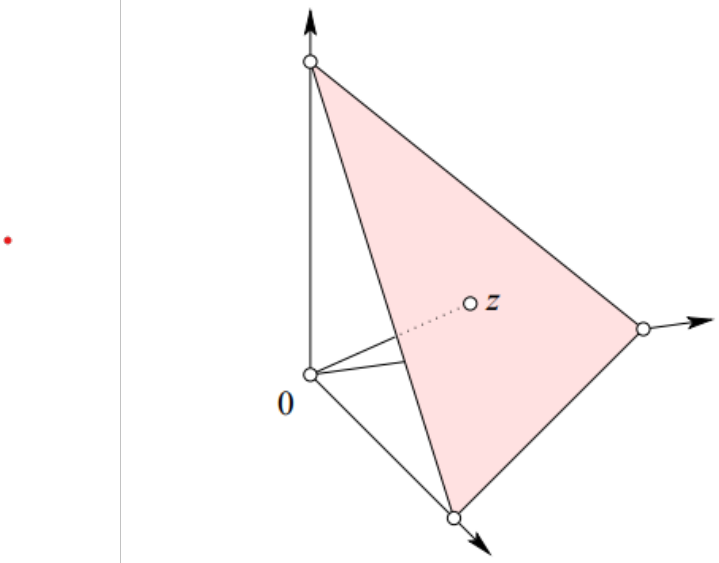



Figure 2.2: Vietoris-Rips Filteration (as given in [1])

**Vietoris-Rips Lemma.** Letting $S$ be a finite set of points in some Euclidean space and $r \geq 0$, we have Vietoris-Rips($r$) $\subseteq$ Cech($\check{r}/\sqrt{2}$). [1]

Proof. A simplex is regular if all its edges have the same length. A convenient representation for dimension $d$ is the standard $d$-simplex, $\Delta^d$, spanned by the endpoints of the unit coordinate vectors in $\mathbb{R}^{d+1}$; see Figure 2.2. Each edge of $\Delta_d$ has length $\sqrt{2}$. By symmetry, the distance of the origin from the standard simplex is its distance from the barycenter, the point $z$ whose $d+1$ coordinates are all equal to $\frac{1}{d+1}$. That distance is therefore $||z|| = 1/\sqrt{d+1}$. The barycenter is also the center of the smallest $d$-sphere that passes through the vertices of $\Delta_d$. Writing

$r_d$ for the radius of that sphere, we have $r_d^2 = 1 - ||z||^2 = d/(d+1)$. For dimension 1, this is indeed half the length of the interval, and for dimension 2, it is the radius of the equilateral triangle. As the dimension goes to infinity, the radius grows and approaches 1 from below. Any set of $d+1$ or fewer points for which the same $\sqrt{d}$-ball of radius $r_d$ is the miniball has a pair at distance 2 or larger. It follows that every simplex of diameter $\sqrt{2}$ or less belongs to $Cech(\check{r}d)$. Multiplying with $\sqrt{2r}$ we get Vietoris-Rips$(r) \subseteq Cech(\check{r}\sqrt{2r}d)$. Since $r_d \leq 1$ for all $d$, the latter is a subcomplex of $Cech(\check{r}_{\sqrt{2r}})$, which implies the claimed subcomplex relationship.

## 2.2 Persistent Homology

Persistent homology adapts classical homology form algebraic topology to finite metrics spaces. The construction of persistent homology starts with a $filteration$ which is a nested sequence of topological spaces: $X_0 \subseteq X_1 ... \subseteq X_n = X$. In particular, We focus on $Vietoris - Rips$ (VR) filtration for finite metric spaces $(\mathcal{X}, d_{\mathcal{X}})$. Let $\epsilon_1 \leq \epsilon_2 \leq ... \leq \epsilon_n$ be an increasing sequence of parameters. The $Vietoris - Rips$ Complex VR$(\mathcal{X}, \epsilon_i)$ at scale $\epsilon_i$ is constructed by adding a node for each $x_j \in \mathcal{X}$ and a $k-$simplex for each set $x_{j_i}$ with diameter less than $\epsilon_i$. Thus,

$$VR(\mathcal{X}, \epsilon_1) \longrightarrow VR(\mathcal{X}, \epsilon_2) \longrightarrow ... \longrightarrow VR(\mathcal{X}, \epsilon_n)$$

The sequence of inclusions induces maps in homology for any fixed dimension $r$. Let $H_r(\mathcal{X}, \epsilon_i)$ be the homology group of VR$(\mathcal{X}, \epsilon_i)$ with coefficients in a field . Then we have the following sequence of vector spaces:

$$H_r(\mathcal{X}, \epsilon_1) \longrightarrow H_r(\mathcal{X}, \epsilon_2) \longrightarrow ... \longrightarrow H_r(\mathcal{X}, \epsilon_n)$$

The collection of $H_r(\mathcal{X}, \epsilon_i)$, together with vector space homomorphisms $H_r(\mathcal{X}, \epsilon_i) \longrightarrow$

$H_r(\mathcal{X}, \epsilon_j)$ is called a persistence module.

When each $H_r(\mathcal{X}, \epsilon_i)$ is finite dimensional, the persistence module can be decomposed into rank one summands which corresponds to birth and death times of homology classes. Let $\alpha \in H_r(\mathcal{X}, \epsilon_i)$ be a non-trivial homology class $\alpha$ is born at $\epsilon_i$ if it is not in the image of $H_r(\mathcal{X}, \epsilon_{i-1}) \longrightarrow H_r(\mathcal{X}, \epsilon_i)$ it is dead entering $\epsilon_j$ if the image of $\alpha$ via $H_r(\mathcal{X}, \epsilon_i) \longrightarrow H_r(\mathcal{X}, \epsilon_{j-1})$ is not in the image $H_r(\mathcal{X}, \epsilon_{i-1}) \longrightarrow H_r(\mathcal{X}, \epsilon_{j-1})$, but the image of $\alpha$ via $H_r(\mathcal{X}, \epsilon_i) \longrightarrow H_r(\mathcal{X}, \epsilon_j)$ is in the image of $H_r(\mathcal{X}, \epsilon_{i-1}) \longrightarrow H_r(\mathcal{X}, \epsilon_j)$.The collection of birth-death intervals $[\epsilon_i, \epsilon_j)$ is called a *barcode* and it represents the persistent homology of VR filteration of $\mathcal{X}$.

Equivalently, we can regard each interval as an ordered pair of birth–death as coordinates and plot each in a plane $\mathbb{R}^2$ , which provides an alternate representation of a *barcode* as a *persistence diagram*.

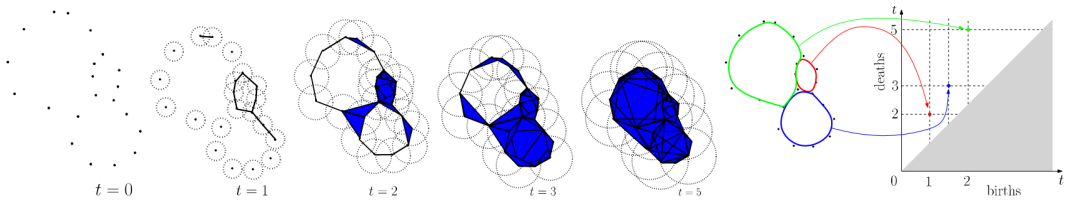



Figure 2.3: Filtration (as given in [1])

**Definition** A *persistence diagram* $D$ is a locally finite multi-set of points in the half-plane $\Omega = \{(x, y) \in \mathbb{R}^2 \mid x \leq y\}$ together with points on the diagonal $\partial\Omega = \{(x, x) \in \mathbb{R}^2\}$ counted with infinite multiplicity. Points in $\Omega$ are called $off - diagonal$ points. The persistence diagram with no off-diagonal points is called *empty persistence diagram* by $D_\emptyset$ .

(The above theory and definitions are from [1]).

# Chapter 3

# Persistence Diagrams

Having defined persistence diagrams, it is necessary to look at a few metrics and definitions.To measure similarities between persistent homology of two functions we use the following definition of a distance between persistence diagrams:

**Definition (Wasserstein distance).** The $p$-th Wasserstein distance between two persistence diagrams, $D_1$ and $D_2$, is defined as

$$W_{p,q}(D_1, D_2) = \inf_{\gamma} \left( \sum_{x \in D_1} ||x - \gamma(x)||_q^p \right)^{\frac{1}{p}} \tag{3.1}$$

where $\gamma$ ranges over all bijections from $D_1$ to $D_2$. The set of bijections is nonempty because of the diagonal.

$p - total\ persistence$ of D is defined as a $W_{p,q}(D, D_\emptyset)$ , where $D_\emptyset$ is the persistent diagram containing just the diagonal.

The set of persistent diagrams along with the finite $p - total\ persistence$ is denoted by $\mathcal{D}_p$. Under this metric , the space is completely separable i.e, a Polish space. [3]

## 3.1 Persistent Measures

An alternative approach is to define persistent diagrams as measures on $\Omega$ as a form of Dirac's measure. Let $\mu$ be Radon measure supported on $\Omega$ the $p-$ *total persistence* is defined as

$$pers_p(\mu) = \int_\Omega ||x - x^T|| d\mu(x) \tag{3.2}$$

Any Radon measure with finite $p-$*total persistence* is called a persistence measure. Space of all persistence measure is denoted by $\mathcal{M}_p$. The optimal transport distance between measures $\mu$ and $\nu$ is

$$OT_{p,q}(\mu, \nu) = \inf_\pi \left( \int ||x - y||_q^p d\Pi(x, y) \right)^{\frac{1}{p}} \tag{3.3}$$

$(\mathcal{M}_p, OT_p)$ forms a polish space which is more advantageous as in measures are linear objects, this can be used for statistical purposes. Though, Expectation of persistence measures forms a heat map. So,further quantization is needed for any relevant data. It is clear that, $\mathcal{D}_p$ is a closed subspace of $\mathcal{M}_p$ with $OT_{p,q} = W_(p, q)$ (Defined in [4]).

## 3.2 Two measures of centrality

**Frechet Mean of Persistent diagrams** is the diagram which minimizes the Frechet function

$$Fr_\rho(D) = \frac{1}{B} \sum_{i=1}^{B} W_p^p(D_i, D) \tag{3.4}$$

The previous works prove that this function is not convex and there is no guarantee to find the global minimizer, there is a greedy algorithm to find the local

minimizer. which makes the mean not unique (Defined in [5])

**Mean Persistence Measure** Let $D = \{D_1, D_2, ..., D_B\}$ be persistent diagram then the empirical mean of is simply $\bar{D} = \sum_{i=1}^{B} D_i$, So for any Borel set A in the first quadrant such that $A \subset \Omega$. The expectation is the average number of points in the persistent diagrams within the set $A$ [4]

# Chapter 4

# Experiments and Algorithms

Experiments ran using mean persistence measures and Fréchet means of persistence diagrams to estimate the persistent homology of large data sets using sub-sampling. In particular, let $\mathcal{X}$ be a large point cloud with a predefined probability distribution satisfying some standard assumptions.We take $B$ number of $i.i.d's$ consisting of $n$ points from $\mathcal{X}$ are chosen. Then compute the mean persistence measure and Fréchet mean which can be regarded as two types of averages of persistence diagrams of sub-sample sets.

### Step 1 : A large point cloud

We initially start with 5000 point, point cloud of an annulus. We denote it by $\mathcal{X}$

### Step 2 : Sub-Sampling

We further take $B-i.i.d$ (independent identically distributed)$\{X_1, X_2, ..., X_B\}$ from $\mathcal{X}$, each of size $n$. In this example, let $n = 400$ *and* $B = 3$

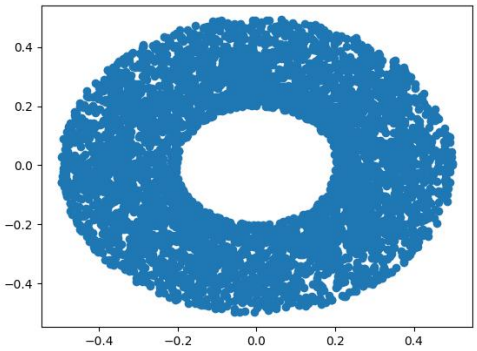
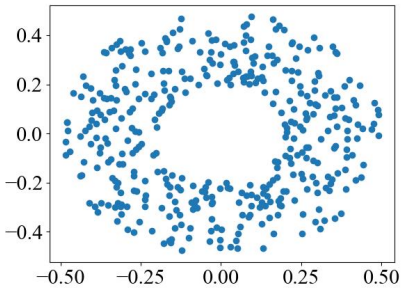
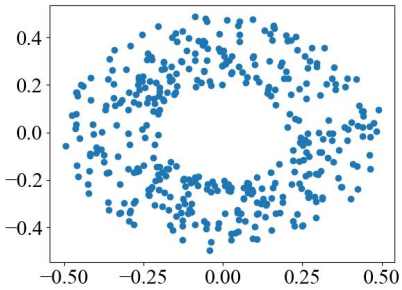


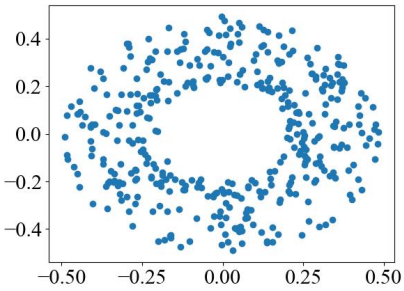

Figure 4.1: A point cloud of Annulus




(a) Sub-Sample 1




(b) Sub-Sample 2

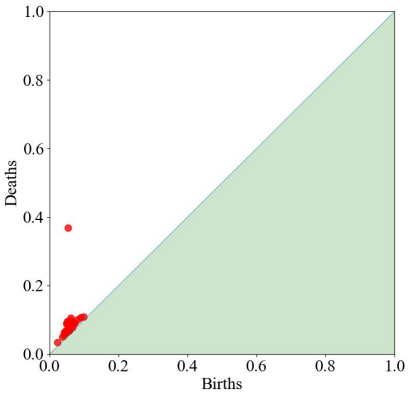



(c) Sub-Sample 3

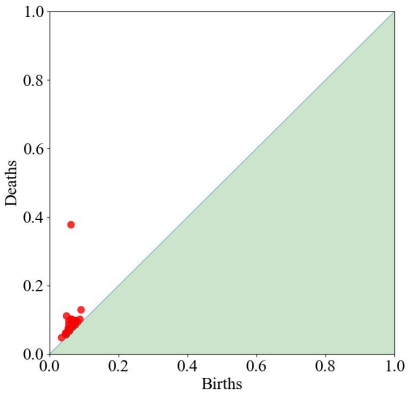
Figure 4.2: three $i.i.d$ of the point cloud

## Step 3 : Persistence Diagrams

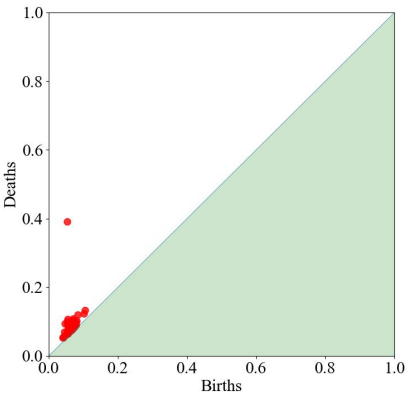
The Sub-samples are put through Vietoris-rips filtration. Persistent diagrams for each $X_i$ are generated. $D = \{D_1, D_2, ..., D_B\}$, each $D_i$ consist of $\Omega$ representing the off-diagonal points and $\partial\Omega$ representing the diagonal.




(a) PD for Sub-Sample 1




(b) PD for Sub-Sample 2




(c) PD for Sub-Sample 3

Figure 4.3: Persistent Diagrams

## Step 4 : Fréchet Mean

The Persistent diagram which minimizes the Fréchet function,

$$Fr_{\rho}(D_j) = \frac{1}{B}\sum_{i=1}^{B} W_p^p(D, D_i) \tag{4.1}$$

The Fréchet mean $D_f$, is not unique, as in the Fréchet function is concave in space of Persistent diagrams

$$D_f = \arg\min_{D_j \in D} Fr_{\rho}(D_j) \tag{4.2}$$

Hence, This is not the reliable approximation of the persistent diagram of the large data-set ($\mathcal{X}$), rather this can be used to initialize the clustering process. Randomly initialize the mean diagram. For example we can start at one of the $B$ persistence diagrams.Use the Hungarian algorithm to compute optimal pairings between the estimate of the mean diagram and each of the persistence diagrams update each point in the mean diagram estimate with the arithmetic mean over all diagrams—each point in the mean diagram is paired with a point (possibly on the diagonal) in each diagram if the updated estimate locally minimizes $Fr_p$ then return the estimate otherwise repeat (see [4])

**Algorithm 2** Greedy Algorithm for Fréchet mean

**Require:** A sequence $D_1, ..., D_B$

Draw $i$ from Uniform($\{1, .., B\}$)
Initialize $Y \leftarrow D_i$
Stop $\leftarrow$ False
**Repeat**
    $K = |Y|$
    **For** i=0,...B **do**
        $(y^j, x_i^j) \leftarrow Hungraian(Y, D_i)$
    **For** j=0,...K **do**
        $y^j \leftarrow mean_{i=1,...,m}(x_i^j)$
    **if** $Hungarian(Y, D_i) = (y_j, x_i^j)$**then** stop $\leftarrow$ true
**Until** $stop = true$
**Return Y**

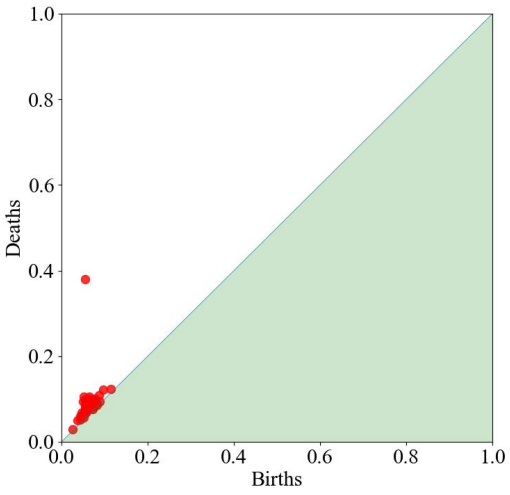



Figure 4.4: Fréchet Mean

### Mean Persistence Measure

The Empirical mean of the persistence diagrams is given by

$$\bar{\mu} = \frac{1}{B}\left(\sum_{i=1}^{B} \mu_i\right) \tag{4.3}$$

where each,$\mu_i$ represents a persistent diagram's off-diagonal elements $\mu_i = \{(x_1, n_1), ..., (x_k, n_k)\}$ where $x_i$ represents the coordinates and $n_i$ the multiplicity of the point.

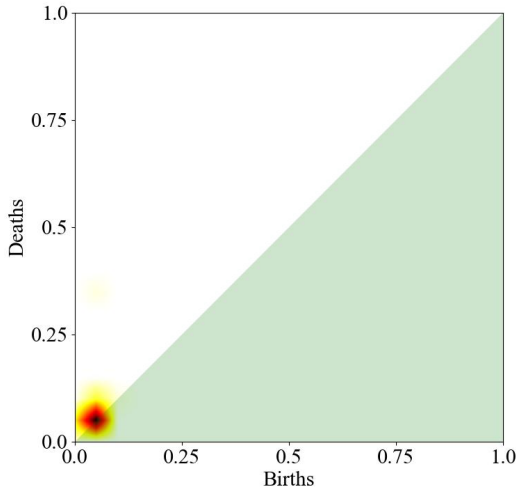



Figure 4.5: Mean Persistence

## 4.1 Quantization

Mean persistence measure generates a heat map instead of proper persistent diagram. A quantization is required to interpret the data. This algorithm's main idea is that at each iteration, the half space $\Omega$ is first partitioned into $k+1$ Voronoi cells $V(\mathbf{c}) = \{V_0, V_1, ..., V_k\}$ with respect to $k$ centroids and $V_{k+1}$ represents the diagonal, and then each centroid is updated by moving along the direction to the $p$-center of each Voronoi cell. Initialization of the centroid, is done by taking points from Fréchet mean diagram. Each $\mu_i$ represents the measure of each persistent diagram $D_i$ The computation time of this algorithm is $\frac{n}{\log n}$ [6]

**Algorithm 3** Quantization of EPDs

**Require:** A sequence $\mu_1, ..., \mu_n$ integer $k$

1: **Pre-Process** Divide indices $\{1, .., n\}$ into batches $(B_1, ..., B_T)$ of size $(n_1, ..., n_T)$

2: set $\bar{\mu}_t := \frac{1}{n_t} \sum_{i \in B_t} \mu_i$ for $1 \leq t \leq T$

3: Sample $c_1^{(0)} ... c_k^{(0)}$ ▷ (1)

4: **For** t=0,...,T-1 **do**

5: $\mathbf{c}^{(t+1)} = \frac{t}{t+1}\mathbf{c} + \frac{1}{t+1} \int_{V_j(\mathbf{c})} x \frac{d\mu(x)}{\mu(V_j(\mathbf{c}))}$

6: **end for**

7: **Output:** $c^{(T)}$ is the final points

# Chapter 5

# Conclusion

An effective approach to improving the performance of existing algorithms could involve leveraging a combination of previously established structures, such as weighted persistence diagrams or sublevel sets. By incorporating a weighted barycenter approach, we may be able to further optimize the output and potentially achieve superior results compared to previous methods. The weights can be assigned based on various factors, such as the importance or relevance of different features in the data. The barycenter approach enables us to compute a representative point for each diagram or set, which can then be combined to obtain a more informative and accurate result.

## 5.1 Weighted Wasserstein Barycenter

The weighted Wasserstein barycenter method is a technique for merging multiple persistence diagrams into a single diagram. It is based on the concept of Wasserstein distance, which measures the dissimilarity between probability distributions.

Given a set of $n$ persistence diagrams $D_1, D_2, ..., D_n$, the Wasserstein barycen-

ter method computes a weighted average of the diagrams using the Wasserstein distance as the weighting function. More specifically, the method solves the following optimization problem:

$$\min_{D} \sum_{i=1}^{n} w_i W_2(D, D_i)^2 \tag{5.1}$$

where $D$ is the merged persistence diagram, $w_i$ are the weights assigned to each input diagram, and $W_2(D, D_i)$ is the Wasserstein distance between $D$ and $D_i$.

The above optimization problem can be solved using various numerical techniques, such as the Sinkhorn algorithm, which is an efficient iterative algorithm for computing the Wasserstein distance. The Sinkhorn algorithm iteratively updates a matrix of weights until convergence, where the weights correspond to the optimal transportation plan between the diagrams.

After obtaining the merged persistence diagram using the Wasserstein barycenter method, we can further refine it using post-processing techniques, such as thresholding, clustering, or outlier removal, to obtain a more accurate approximation of the persistence diagram of the large point cloud.

Overall, the Wasserstein barycenter method provides an efficient and effective way to merge multiple persistence diagrams into a single diagram, while retaining the topological information of the original data.

Here's a detailed algorithm for computing the Wasserstein barycenter of a set of persistence diagrams.

The choice of the parameters $p$, $t$, and $\gamma$ depends on the specific problem and data set, and can be determined experimentally or by following established guide-

**Algorithm 4** Wasserstein barycenter of a set of persistence diagrams

**procedure** WASSERSTEIN BARYCENTER($D_1, \ldots, D_n, p, t, \gamma$)
    Initialize the barycenter diagram $B_0$ as the diagram with the same set of points as $D_1$, and zero persistence values.
    **for** $i = 1, \ldots, t$ **do**
        Compute the Wasserstein distances between the barycenter diagram $B_{i-1}$ and each of the input diagrams $D_1, \ldots, D_n$, using a suitable algorithm such as the sliced Wasserstein distance.
        **for** $j = 1, \ldots, n$ **do**
            Compute the weights $w_j$ , where $W_p(B_{i-1}, D_j)$ is the $p$-th power of the Wasserstein distance between $B_{i-1}$ and $D_j$.
        **end for**
        Normalize the weights $w_j$ so that they sum to one.
        Compute the weighted average of the input diagrams $D_1, \ldots, D_n$, using the weights $w_j$ to obtain the updated barycenter diagram $B_i$.
    **end for**
    **return** the final barycenter diagram $B_t$ as the Wasserstein barycenter of the input set of persistence diagrams.
**end procedure**

lines. The algorithm can be implemented efficiently using a suitable data structure to represent the persistence diagrams and a fast algorithm for computing the Wasserstein distance, such as the sliced Wasserstein distance or the Sinkhorn algorithm. Note that this is just one possible algorithm for computing the Wasserstein barycenter of a set of persistence diagrams, and there may be variations and improvements depending on the specific application and context.[7]

## 5.2 Challenges

Although this seems viable option , a few things are still to be seen.

- A method to compute weights, one of the options is $w_j = \exp\left(-\gamma W_p(B_{i-1}, D_j)^p\right)$
- Getting a persistence diagram out of this "weighted sum"

- Proving mathematically why and how this algorithm converges better.

## 5.3 Further Reading

After conducting extensive research and analysis in various related fields, I have identified a specific research problem that is currently the focus of my work. This problem represents a significant challenge in the field and has the potential to lead to valuable insights and innovations. However, there are still several areas that require further exploration and optimization before a satisfactory solution can be achieved. As such, I plan to continue my investigations by conducting additional analysis and experiments, exploring new approaches and methodologies, and leveraging existing knowledge and expertise to drive progress towards a resolution. By addressing these challenges head-on and developing effective solutions, I hope to contribute to the advancement of the field and make meaningful contributions to the wider scientific community.

# Appendix A

# Sinkhorn Algorithm

Given two probability measures $\mu$ and $\nu$ with finite second moments, the Sinkhorn algorithm provides an iterative method for computing the $p$th power of the Wasserstein distance $W_p(\mu, \nu)$ between them. The algorithm is based on the concept of entropic regularization and encourages sparsity in the optimal transport plan.

The Sinkhorn algorithm computes an approximation of $W_p(\mu, \nu)$ by solving the following optimization problem:

$$\min_{T \in U(\mu,\nu)} \int |x - y|^p T(x, y) d\mu(x) d\nu(y) - \epsilon H(T)$$

where $U(\mu, \nu)$ is the set of all joint probability measures with marginal distributions $\mu$ and $\nu$, $H(T)$ is the negative entropy of $T$, and $\epsilon$ is a parameter that controls the strength of the regularization.

The algorithm iteratively updates the matrix $T$ until convergence, using the following two steps:

**Dual update:** Compute the dual variables $u$ and $v$ from the current matrix $T$, using the formulae:

$$u(x) = -\epsilon \log(\mu(x)) + \epsilon \log \left( \sum_y \nu(y) e^{-v(y)/\epsilon} T(x,y) \right)$$
$$v(y) = -\epsilon \log(\nu(y)) + \epsilon \log \left( \sum_x \mu(x) e^{-u(x)/\epsilon} T(x,y) \right)$$

**Primal update:** Update the matrix $T$ using the formula:

$$T(x,y) = \exp \left( \frac{-u(x) - v(y) + |x-y|^p}{\epsilon} \right)$$

The algorithm terminates when the change in the objective function falls below a prescribed tolerance level.

# Appendix B

# Persistence Landscape

The persistence landscape[2] is a popular tool in topological data analysis that is used to summarize the persistent homology of a given data set. The persistence landscape is a function that maps each real number to a vector that summarizes the persistence intervals of a given topological feature, such as connected components or holes, at that threshold. The persistence landscape has been shown to be a powerful and robust tool for analyzing the topology of data sets, and has been used in a wide range of applications. One common approach to summarizing the persistent homology of a data set using the persistence landscape is to construct the average persistence landscape. The average persistence landscape is obtained by averaging the persistence landscapes of multiple realizations of the same data set or multiple data sets with similar characteristics. The idea behind the average persistence landscape is to obtain a more stable and robust summary of the persistent homology of the data.

However, there are cases where the persistence landscape falls short in constructing the average persistence landscape. One potential issue is that the persistence landscape can be sensitive to the choice of distance function used to define

the persistence diagram. This can lead to differences in the persistence landscapes of different realizations of the same data set, or different data sets with similar characteristics, even if they have similar persistent homology.

Another issue is that the persistence landscape can be sensitive to the choice of basis functions used to represent the persistence intervals. In practice, it is often necessary to choose a finite set of basis functions to represent the persistence intervals, which can limit the ability of the persistence landscape to capture the full complexity of the persistent homology.

To overcome these issues, several variants of the persistence landscape have been proposed, such as the stable persistence landscape and the multi-scale persistence landscape, which aim to address the sensitivity to the choice of distance function and basis functions, respectively. Additionally, other approaches to summarizing the persistent homology of data sets, such as persistent cohomology and persistent homology signatures, may provide alternative or complementary summaries to the persistence landscape.

It is constructed as,
Given a persistence diagram $D = \{(x, y)|\ y \geq x\ \forall x, y \in \mathbb{R}\}$
**First Step:**
Transform the coordinates to $m = \frac{y+x}{2}$ $h = \frac{y-x}{2}$

$$(x_i, y_i) \rightarrow (m_i, h_i) = \left(\frac{y_i + x_i}{2}, \frac{y_i - x_i}{2}\right)$$

**Second Step:**
A peak function is defined for each $(m_i, h_i)$ which is $f_i$

$$f_i(m) = \begin{cases} 0 & m \leq x_i \\ m - x_i & x_i \leq m \leq \frac{y_i + x_i}{2} \\ y_i - m & \frac{y_i + x_i}{2} \leq m \leq y_i \\ 0 & y_i \leq m \end{cases}$$

**Third Step:**

Collecting all such peak functions $f_i$, landscape function $\lambda_i$ is defined as,

$$\begin{aligned} \lambda_1(m) &= max\{f_1(m), ..., f_k(m)\} \\ \lambda_2(m) &= max\{f_1(m), .., \bar{f_i}(m), ., .\} \\ &\cdot \\ &\cdot \\ \lambda_j(m) &= 0 \quad \forall j > k \end{aligned} \tag{B.1}$$

This produces a decreasing sequence of functions $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \ . \ . \ .$

Collection of these functions $\wedge_p \ = \{\lambda_1, \lambda_2, ..., \lambda_k\}$

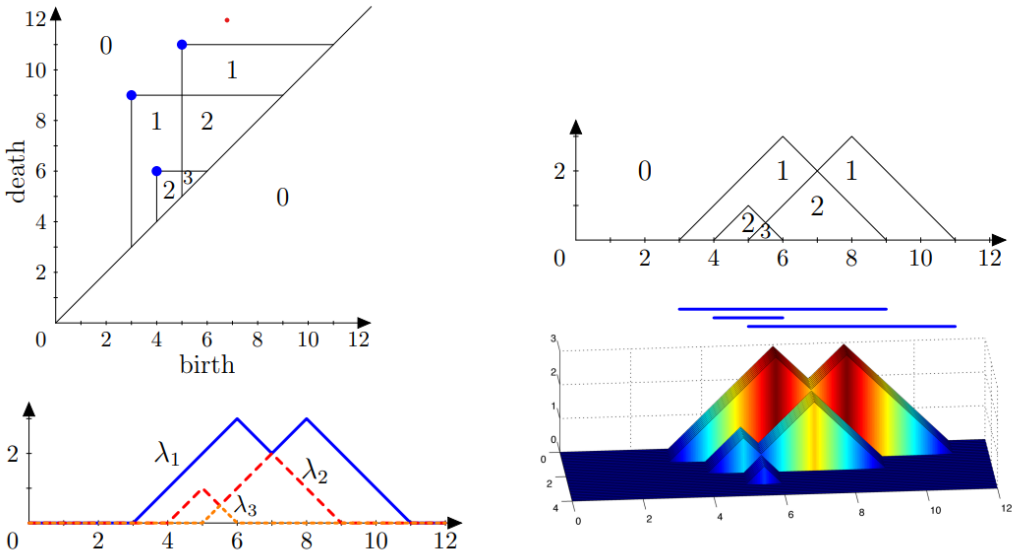



Figure B.1: Persistence Landscape (As given in [2])

# Appendix C

# Different Variations of Fréchet Function

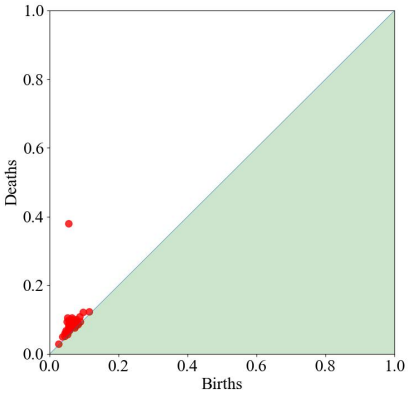
## C.1 Logarithmic Fréchet Function

The Fréchet function is semi-concave in the space, which requires an alteration in the function, in-order to arrive at a global minima. A natural candidate being logarithm

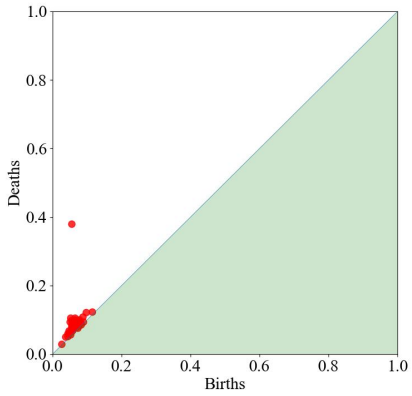
$$\bar{\mathcal{D}_f} = arg \min_{D_j \in \mathcal{D}} \frac{1}{B} \sum_{i=1}^{B} log(W_2^2(D_i, D_j)) \tag{C.1}$$

### C.1.1 Results

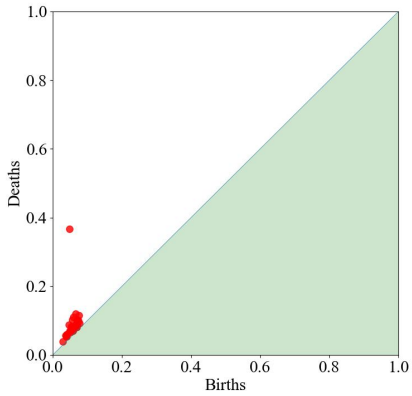
The proper Fréchet mean and logarithm approach both give the same persistence diagram as the mean. Whereas, the algorithmic Fréchet mean gives a much closer persistence diagram to the true persistence diagram




(a) Frechet mean $\mathcal{D}_f^A(0.15798)$




(b) Algo-Frechet Mean $\mathcal{D}_f(0.1537)$




(c) Log of Frechet mean $\bar{\mathcal{D}}_f(0.1537)$

Figure C.1: Variations of Frechet function

## C.2 Exponential Fréchet Function

The set of persistence diagram along with Wasserstein metric forms a metric space. Take exponential of said Wasserstein distance still preserves all the properties of the space.

Which allows us to define a Fréchet function with this metric.

$$\bar{\mathcal{D}_f} = arg \min_{D_j \in \mathcal{D}} \frac{1}{B} \sum_{i=1}^{B} e^{(W_2^2(D_i, D_j))} \tag{C.2}$$

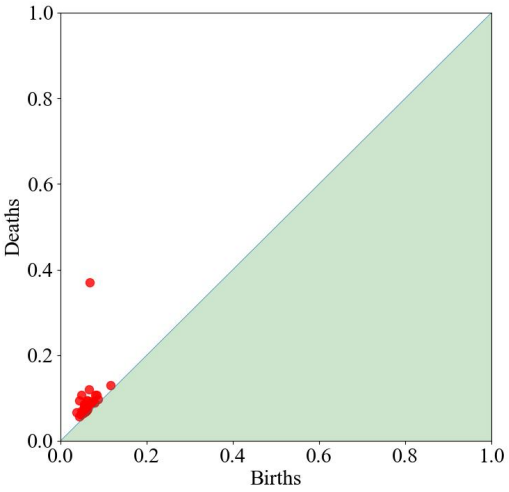



Figure C.2: Exponential-Fréchet Mean

### C.2.1 Results

After conducting thorough experimentation, it was observed that both the proper Fréchet mean and exponential approach resulted in the same persistence diagram as the mean. However, the algorithmic Fréchet mean method provided a much more accurate persistence diagram, closely resembling the true persistence diagram.

# Bibliography

[1] Herbert Edelsbrunner and John Harer. *Computational Topology: An Introduction*. American Mathematical Society, 2010.

[2] Peter Bubenik. The persistence landscape and some of its properties. *Springer*, Volume 15:97–117, 2020.

[3] Yuriy Mileyko, Sayan Mukherjee, and John Harer. Probability measures on the space of persistence diagrams. *Inverse Problems*, Volume 27:124007 – 124029, 2011.

[4] Vincent Divol and Théo Lacombe. Understanding the topology and the geometry of the space of persistence diagrams via optimal partial transport. *Journal of Applied and Computational Topology*, Volume 5:1–53, 2021.

[5] Elizabeth Munch, Katharine Turner, Paul Bendich, Sayan Mukherjee, Jonathan Mattingly, and John Harer. Probabilistic Fréchet means for time varying persistence diagrams. *Electronic Journal of Statistics*, Volume 9:1173 – 1204, 2015.

[6] Vincent Divol and Théo Lacombe. Estimation and quantization of expected persistence diagrams. *International Conference on Machine Learning*, 2021.

[7] Mathieu Carriere, Frederic Chazal, Yuichi Ike, Theo Lacombe, Martin Royer, and Yuhei Umeda. Perslay: A neural network layer for persistence diagrams

and new graph topological signatures,proceedings of the twenty third international conference on artificial intelligence and statistics. *PMLR*, Volume 108:2786–2796, 2020.