APPLICATIONS OF PERSISTENT COHOMOLOGY TO DIMENSIONALITY REDUCTION AND CLASSIFICATION PROBLEMS

By

Luis G. Polanco Contreras

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Computational Mathematics, Science and Engineering – Doctor of Philosophy
Mathematics – Dual Major

2022

ABSTRACT

APPLICATIONS OF PERSISTENT COHOMOLOGY TO DIMENSIONALITY REDUCTION AND CLASSIFICATION PROBLEMS

By

Luis G. Polanco Contreras

Many natural phenomena are characterized by their underlying geometry and topological invariants. Part of understanding such processes is being able to differentiate them and classify them through their topological and geometrical signatures. Many advances have been made which use topological data analysis to such end. In this work we present multiple machine learning tools aided by topological data analysis to classify and understand said phenomena.

First, feature extraction from persistence diagrams, as a tool to enrich machine learning techniques, has received increasing attention in recent years. In this paper we explore an adaptive methodology to localize features in persistent diagrams, which are then used in learning tasks. Specifically, we investigate three algorithms, CDER, GMM and HDBSCAN, to obtain adaptive template functions/features. Said features are evaluated in three classification experiments with persistence diagrams. Namely, manifold, human shapes and protein classification. In this area, our main conclusion is that adaptive template systems, as a feature extraction technique, yield competitive and often superior results in the studied examples. Moreover, from the adaptive algorithms here studied, CDER consistently provides the most reliable and robust adaptive featurization.

Furthermore, we introduce a framework to construct coordinates in finite Lens spaces for data with nontrivial 1-dimensional $\mathbb{Z}_q := \mathbb{Z}/\mathbb{Z}_q$ persistent cohomology, for $q > 2$ prime. Said coordinates are defined on an open neighborhood of the data, yet constructed with only a small subset of landmarks. We also introduce a dimensionality reduction scheme in $S^{2n-1}/\mathbb{Z}_q$ (Lens-PCA: LPCA) and demonstrate the efficacy of the pipeline $\mathbb{Z}_q$ -persistent

cohomology $\Rightarrow S^{2n-1}/\mathbb{Z}_q$ coordinates $\Rightarrow$ LPCA, for nonlinear (topological) dimensionality reduction. This methodology allows us to capture and preserve geometrical and topological information through a very efficient dimensionality reduction algorithm.

Finally, to make use of some of the most powerful tools in algebraic topology we improve on methodologies that make use of persistent 2-dimensional homology to obtain quasiperiodic scores that indicate the degree of periodicity or quasiperiodicity of a signal. There is a significant computational disadvantage in this approach since it requires the often expensive computation of 2-dimensional persistent homology.

Our contribution in this area uses the algebraic structure of the cohomology ring to obtain classes in the 2-dimensional persistent diagram by only using classes in dimension 1, saving valuable computational time in this manner and obtaining more reliable quasiperiodicity scores. We develop an algorithm that allows us to effectively compute the cohomological death and birth of a persistent cup product expression. This allows us to define a quasiperiodic score that reliably separates periodic from quasiperiodic time series.

A mi madre y mi hermano.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to Dr. Jose A. Perea for the opportunity to pursue a Ph.D. at Michigan State University. The advice I received during my time as his student helped me gather a deeper understanding and appreciation for topological data analysis and mathematics in general. He spent countless hours helping me develop and polish ideas, papers, and talks. I will be forever in debt for your patience and guidance through many difficult times and hurdles to overcome.

I would like to thank other committee members, Dr. Teena Gerhardt, Dr. Mattew Hedden, Dr. Mark Iwen, and Dr. Elizabeth Much. I gain valuable knowledge through lectures and casual discussions. I cherish the passion for research and teaching you all have shown to me.

I would like to offer my special thanks to Dr. Hitesh Gakhar and Dr. Joshua L. Mike and Danielle Barns for their support with writing, coding, and research discussions that help me navigate across many difficult problems and challenges.

I would like to express my gratitude to Dr. Andres Angel Cardenas for guiding me through my undergraduate and master's studies. His inspiration and passion help me forge my path. I can only strive to inspire others as he has been an inspiration to me.

I want to thank my friends Omar, Nicolas, Santiago, Mateo, Felipe, Sebastian, Sergio, Angelica, Dylan, and Jessica for always offering words of advice and putting a smile on my face all these years.

Last but not least, I would like to thank my mother and brother for their endless support and understanding. If anything, this work is a tribute to their unconditional love and sacrifices. Without your support, I would never make it.

Thank you!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

CHAPTER 1

INTRODUCTION

The main objective in this work is to examine multiple problems in the field of data analysis for which the usual tools do not provide sufficiently satisfactory solutions. We develop different approaches to solve these problems by leveraging the inherent advantages present in using geometric and topological tools. In Chapter 1 we tackle the main question of how can we take advantage of the information present in persistent diagrams and persistent homology to solve machine learning problems. To this end we develop a featurization method that allows us to extract meaningful feature vectors from persistent diagrams that can be fed into classical machine learning pipelines.

One of the central questions in Topological Data Analysis (TDA) is how to leverage topological information, like persistence diagrams [1], for machine learning purposes. This idea has been explored, for instance, in [2–5] and [6].

In particular, [5] establishes theoretical and computational tools to translate supervised machine learning tasks (e.g., classification and regression) with topological features, into the problem of approximating continuous real-valued functions on the space of persistence diagrams, $\mathcal{D}$, endowed with the Bottleneck distance. The main concept is that of templates. These are continuous real-valued compactly supported functions on $\mathbb{W} := \big\{(x_1, x_2) \in \mathbb{R}^2 \mid 0 \leq x_1 < x_2\big\}$, which (by integration against persistence measures, whereby a diagram is replaced by a sum of Dirac deltas) yield continuous functions on $\mathcal{D}$. The same work shows that one can construct countable families of template functions (a template system), which in turn give rise to dense subsets of $C(\mathcal{D}, \mathbb{R})$ with respect to the compact-open topology. 3.1.6 below indicates how a template system can be utilized to generate (polynomial) features for supervised machine learning problems on persistence diagrams.

In this paper we address the question of producing template systems that are attuned (adaptive) to the input data set and the supervised classification problem at hand. We explore and compare different strategies to assemble adaptive template systems; namely, Cover-Tree Entropy Reduction (CDER) [7], Gaussian Mixture Models (GMM) [8] and Hierarchical density-based spatial clustering of applications with noise (HDBSCAN) [9]. The conclusion is that CDER is the most consistently successful strategy out of the ones explored.

We present three different examples where we use adaptive template functions to extract features from persistence diagrams for supervised classification tasks. First, we explore a 6 class classification problem presented in [5]. In this problem, several random samples are taken from each of 6 manifolds, and persistence diagrams are computed as descriptors in each case. We then use our adaptive template functions and compare to the results provided in [5]. The average classification accuracy of adaptive templates for both the training and testing sets is comparable to that of [5]. On the other hand, the standard deviation of our results is much smaller, making our methodology more stable compared to the template systems proposed in [5].

We then report results on the SHREC 2014 synthetic data set [10], which involves a 15 class supervised learning problem. Each class in this data set corresponds to a human body in five different poses and three different shapes: male, female and child. The data points in this data set are 3D meshes. In [4] a heat kernel signature is computed for each mesh and for 10 different kernel parameter values. This defines 10 different classification tasks, each of which uses the corresponding heat sub-level set persistence diagrams as inputs. The results we obtain using adaptive template functions are contrasted with [5]. The results from this experiment highlight how CDER provides a more reliable method for obtaining adaptive templates when compared to GMM and HDBSCAN. Furthermore, when we select the heat kernel signature corresponding to the 6-th frequency, CDER adaptive templates generate a classification model with accuracy on par with the best results in [5] and [4]. When compared to the non-adaptive (tent) templates of [5], our classification results are superior.

Finally, we present results for a protein classification problem on the publicly available

Protein Classification Benchmark data set PCB00019 [11]. This data set contains spatial information for $1,357$ proteins as well as 55 distinct supervised classification tasks. The results on [3] are used as a benchmark, since they also use persistence diagrams, but the extracted features are hand-crafted to reflect chemical/physical properties of interest. In this experiment, adaptive template functions improve the average classification accuracy reported in [3] from 82% to around 98%.

In Chapter 2 we develop a dimensionality reduction algorithm that is consistent with standard methods in topological data analysis. We aim to present a dimensionality reduction algorithm that preserves topological invariants, moreover the methodology we present aim to preserve specific topological features in the persistent homology across the dimensionality reduction process.

Another central question in Topological Data Analysis (TDA) is how to use topological signatures like persistent (co)homology [1] to infer spaces parametrizing a given data set [12–14]. This is relevant in nonlinear dimensionality reduction since the presence of nontrivial topology—e.g., loops, voids, non-orientability, torsion, etc—can prevent accurate descriptions with low-dimensional Euclidean coordinates.

Here we seek to address this problem motivated by two facts. The first: If G is a topological abelian group, then one can associate to it a contractible space, EG, equipped with a free right G-action. For instance, if $G = \mathbb{Z}$, then $\mathbb{R}$ is a model for $E\mathbb{Z}$, with right $\mathbb{Z}$-action $\mathbb{R} \times \mathbb{Z} \ni (r, n) \mapsto r + n \in \mathbb{R}$. The quotient $BG := EG/G$ is called the classifying space of G [15]. In particular $B\mathbb{Z} \simeq S^1$, $B\mathbb{Z}_2 \simeq \mathbb{R}\mathbf{P}^\infty$, $BS^1 \simeq \mathbb{C}\mathbf{P}^\infty$ and $B\mathbb{Z}_q \simeq S^\infty/\mathbb{Z}_q$; here $\simeq$ denotes homotopy equivalence. The second fact: If B is a topological space and $\mathscr{C}_G$ is the sheaf over B (defined in [16]) sending each $U \subset B$ open to the abelian group of continuous maps from U to G, then $\check{H}^1(B; \mathscr{C}_G)$—the first Čech cohomology group of B with coefficients in $\mathscr{C}_G$—is in bijective correspondence with $[B\,,\, BG]$—the set of homotopy classes of continuous maps from B to the classifying space BG. This bijection is a manifestation of the Brown representability theorem [17], and implies, in not so many words, that Čech cohomology classes can be represented as coordinates with values in a classifying space (like S^1 or $S^\infty/\mathbb{Z}_q$).

For point cloud data—i.e., for a finite subset X of an ambient metric space (M, d)—one does not compute Čech cohomology, but rather *persistent cohomology.* Specifically, the persistent cohomology of the Rips filtration on the data set X (or a subset of landmarks L). The first main result of this paper contends that steps one through three below mimic the bijection $\check{H}^1(B; \mathscr{C}_{\mathbb{Z}_q}) \cong [B, S^\infty/\mathbb{Z}_q]$ for $B \subset M$ an open neighborhood of X:

1. Let (M, d) be a metric space and let $L \subset X \subset M$ be finite. X is the data and L is a set of landmarks.

2. For a prime $q > 2$ compute $PH^1(\mathcal{R}(L); \mathbb{Z}_q)$; the 1-dim $\mathbb{Z}_q$-persistent cohomology of the Rips filtration on L. If the corresponding persistence diagram $\mathsf{dgm}(L)$ has an element (a, b) so that $2a < b$, then let $a \leq \epsilon < b/2$ and choose a representative cocycle $\eta \in Z^1(R_{2\epsilon}(L); \mathbb{Z}_q)$ whose cohomology class has (a, b) as birth-death pair.

3. Let $B_\epsilon(l)$ be the open ball in M of radius ϵ centered at $l \in L = \{l_1, \ldots, l_n\}$, and let $\varphi = \{\varphi_l\}_{l \in L}$ be a partition of unity subordinate to $\mathcal{B} = \{B_\epsilon(l)\}_{l \in L}$. If $\zeta_q \neq 1$ is a q-th root of unity, then the cocycle η yields a map $f : \bigcup \mathcal{B} \longrightarrow L_q^n$ to the Lens space $L_q^n = S^{2n-1}/\mathbb{Z}_q$ defined as the quotient generated by the free $\mathbb{Z}_q$ action $(z_1, \ldots, z_n) \mapsto (e^{2\pi \mathrm{i}/q} \cdot z_1, \ldots, e^{2\pi \mathrm{i}/q} \cdot z_n)$. The map f can be expressed in homogeneous coordinates by the formula

$$B_\epsilon(\ell_j) \ni b \ , \ f(b) = \left[\sqrt{\varphi_1(b)} \zeta_q^{\eta_{j1}} : \cdots : \sqrt{\varphi_n(b)} \zeta_q^{\eta_{jn}} \right]$$

where $\eta_{jk} \in \mathbb{Z}_q$ is the value of the cocycle η on the edge $\{l_j, l_k\} \in R_{2\epsilon}(L)$.

If $X \subset \bigcup \mathcal{B}$, then $f(X) = Y \subset L_q^n$ is the representation of the data—in a potentially high dimensional Lens space—corresponding to the cocycle η. The second contribution of this paper is a dimensionality reduction procedure in L_q^n akin to Principal Component Analysis, called LPCA. This allows us to produce from Y, a family of point clouds $P_k(Y) \subset L_q^k$,

$1 \leq k \leq n$, $P_n(Y) = Y$, minimizing an appropriate notion of distortion. These are the Lens coordinates of X induced by the cocycle η.

This work, combined with [18, 19], should be seen as one of the final steps in completing the program of using the classifying space BG, for G abelian and finitely generated, to produce coordinates for data with nontrivial underlying 1^{st} cohomology. Indeed, this follows from the fact that $B(G \oplus G') \simeq BG \times BG'$, and that if G is finitely generated and abelian, then it is ismorphic to $\mathbb{Z}^n \oplus \mathbb{Z}_{n_1} \oplus \cdots \oplus \mathbb{Z}_{n_r}$ for unique integers $n, n_1, \ldots, n_r \geq 0$.

Part of understanding periodic process is being able to differentiate them from quasi-periodic occurrences. These phenomena are characterized by collections of underlying frequencies. In the case of periodic signals, the underlying frequencies have a common fundamental frequency, while on the other hand quasi periodic processes are determined by linearly independent frequencies.

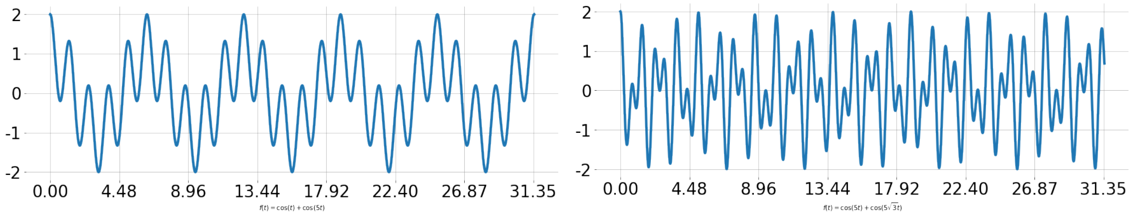

Figure 1.1 **Left:** Periodic signal with underlying frequencies $\{1, 5\}$. **Right:** Quasi-periodic signal with underlying frequencies $\{5, 5\sqrt{3}\}$.

Many tools have been made to leverage topological data analysis to classify and understand quasi-periodic phenomena, one of the most recent approaches comes from [20] and [21]. In the later, the authors developed a framework which allows them to extract topological invariants, such as persistent cohomology to construct different scores that quantify the degree of periodicity vs quasi-periodicity of a given signal. This framework makes ample use of [22], where the authors show how sliding window embedding produces a point cloud which accumulates around a torus of dimension equal to the number of independent frequencies in the signal.

With this point cloud in place it is possible to compute its persistent cohomology, using the Ripser algorithm [23], to recover topological signatures corresponding to such toroidal spaces. In particular, the methodology presented in [21] relies heavily on computing persistent cohomology up to dimension 2 and their corresponding persistent diagrams to differentiate between periodic and quasi-periodic signals.

However, it is possible to generate periodic signals whose time window embedding does not recover a 2-dimensional torus while producing persistent diagrams that suggest this toroidal structure. We construct in Chapter 4.4.2 one such example, by using the tools exhibited in [22]. More specifically we produce a periodic signal whose time delay embedding clusters around a space we refer as a "bunny". In other words a space with topological invariant similar to that of a sphere (S^2) with two loops (S^1) attach to it, similar to the ears of a bunny, thus the name.

One of the main goals in this work is to refine this methodology in order to overcome this type of false positive examples. To achieve this we use the richer algebraic structure of the cohomology ring in the form of the cup product [24]. This algebraic tool allow us to enhance persistent diagrams to further study interaction between loops and void in the topological structure of the spaces we analyse. In Chapter 4.2 and Chapter 4.3 we present an algorithm to obtain the persistence of cup chains, which are persistent chains that arise from the cup product of two classes in the persistent diagram. This algorithm allow us to effectively compute the cohomological death and birth of a persistent cup product expression by taking advantage of some of the efficiencies baked in the Ripser algorithm [23].

We use this approach to concretely obtain the persistence of classes in the 2-dimensional persistent cohomology while only using classes in dimension 1. In Chapter 4.5 we present the improved quasi-periodicity scores in [21] using persistent cup product and showcase its performance in both synthetic and real data sets.

In addition to this, we exhibit in Chapter 4.3 how the proposed algorithm provides computational improvements since it allow us to obtain the persistence of classes in 2-dimensional persistent diagram by only computing the generators for classes in the 1-dimensional persistent diagram. This procedure eliminates the need to enumerate the 3-simplices in the Rips complex or to make any computation/reduction using the 2-coboundary operator. This results in improving the computational efficiency of obtaining scores to separate periodic and quasi-periodic signals.

CHAPTER 2

BACKGROUND

Persistent homology has emerged as an increasingly useful tool is the area of machine learning and data science from the perspective of Topological Data Analysis. One of the main contributions it has provided to the field comes in the form of methodologies aimed to capture the geometrical and topological features to be used for machine learning tasks and dimensionality reduction problems.

We first provide some basic definitions as the building blocks for the work presented in here. Since we will be working with persistent homology we will need to introduce some basic notions in algebraic topology, most of these definition and results in this section are taken from [25] and [26].

Definition 2.0.1. *Given a set X an abstract simplicial complex K is a collection of subsets of $P(X)$ such that*

1. *Every $\sigma \in K$ is finite,*
2. *For any $\sigma \in K$, if $\tau \subset \sigma$ then $\tau \in K$.*

We will use the following commonly used notation from this point on. The sets in Δ are called **simplices** and the **dimension** of a simplex is defined as $\dim(\sigma) = |\sigma| - 1$, and since by definition each simplex is finite we define $\dim(K) = \max\{\dim(\sigma)\}$.

A **face** of σ is any $\emptyset \neq \tau \subsetneq \sigma$. The **$n$-th skeleton** of K is denoted by $K^{(n)} := \{\sigma \in \Delta : \dim(\sigma) \leq n\}$. $K^{(0)}$ is also called the set of **vertices** of K. A subcomplex L of K is an abstract simplical complex such that $L^{(n)} \subset K^{(n)}$ for all $n \in \mathbb{Z}$.

Definition 2.0.2. *A simplicial map $f : K \to L$ between simplicial complexes is a map such that $f(K^{(0)}) \subset L^{(0)}$.*

Now we define the algebraic structures required to define homology. So for a given simplical complex K and abelian group G we have the following.

Definition 2.0.3. *The n-th chain group $C_n(K;G)$ of K with coefficients in the group G is*

$$C_n(K;G) := \left\{\textstyle\sum_{i\in I} g_i\sigma_i : \sigma_i \in K^{(n)} \setminus K^{(n-1)}, g_i \in G \text{ and } g_i = 0 \in G \text{ for all but finitely many } i \in I\right\}.$$

We will refer to each $\tau \in C_n(K;G)$ as a chain. And each $\sigma \in K^{(n)}$ as a generator of $C_n(K;G)$.

We will use the notation $C_n(K)$ if there is no ambiguity about the group G being used. Amongst the more common choices for G are $\mathbb{Z}$ and $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$ for a prime number $q \in \mathbb{Z}$, for the computations and examples presented in this work we will be using $G = \mathbb{Z}_2$ unless we indicate otherwise.

Definition 2.0.4. *Let K be a simplicial complex and G a group. The n-th boundary map ∂_n is a group homomorphism $\partial_n : C_n(K;G) \to C_{n-1}(K;G)$ defined for any $\sigma = [v_0, \ldots, v_n] \in K^{(n)}$ as*

$$\partial_n([v_0, \ldots, v_n]) := \sum_{i=0}^{n} (-1)^n [v_0, \ldots, \hat{v_i}, \ldots, v_n], \tag{2.1}$$

where $[v_0, \ldots, \hat{v_i}, \ldots, v_n]$ denotes the i-th face of σ obtained from deleting the vertex v_i from the set $\{v_0, \ldots, v_n\}$.

Definition 2.0.5. *A chain complex is a collection $C_* = \{C_i, f_i\}_{i\in I}$ of groups C_i and morphisms $f_i : C_i \to C_{i-1}$ such that $f_{i-1}f_i = 0$. This condition implies that img$(f_i) \subset \ker(f_{i-1})$ for all $i \in I$.*

Example 1. *The collection $\{C_n(K), \partial_n\}$ of chain groups of a simplicial complex K together with the boundary maps defined above form a chain complex. A proof of this is completely analogous to Lemma 2.1 in [25].*

Definition 2.0.6. *The i-th homology group of a chain complex C_* with coefficients in a group G is defined as*

$$H_i(C_*; G) := \frac{\ker(f_i)}{img(f_{i+1})} \tag{2.2}$$

Example 2. *The homology groups of the chain complex $\{C_i(K;G), \partial_i\}_{i \in \mathbb{N}}$ are simply denoted by $H_i(K;G)$.*

Since we will we working with data sets we need to provide a method to obtain simplicial complexes out of them. To achieve this goal we can view a data set as a finite metric space. For example if the data set is a subset of $\mathbb{R}^n$ it has a natural metric associated to it. In most cases this will not be true but the user can always define a suitable metric for the problem at hand.

So given a finite metric space (X, d) we can construct multiple simplicial complexes. We will focus on two particular constructions for the time being.

Definition 2.0.7. *Given a finite metric space (X, d) and a real number $\epsilon > 0$ we define the Rips complex $R_\epsilon(X)$ to be the simplicial complex with vertices X and n-simplices defined to be the set*

$$\left\{(x_1, \ldots, x_n) : d(x_i, x_j) < 2\epsilon, 0 \leq i, j \leq n\right\}.$$

Definition 2.0.8. *Given $\epsilon > 0$ the Čech complex $C_\epsilon(X)$ on (X, d) is the simplicial complex with vertex set X and such that $(x_0, \ldots, x_n) \in C_\epsilon(X)$ if and only if*

$$\bigcap_{j=1}^{n} B_\epsilon(x_j) \neq \emptyset,$$

where $B_\epsilon(x_j) = \left\{y \in X : d(x_j, y) < \epsilon\right\}$.

There are some important relations between the Vieotris-Rips and Čech complex of a metric space. First of all we can see that $C_\epsilon(X) \subset R_\epsilon(X)$ since the Vietoris-Rips complex contains every simplex warranted by the given edges, which is not true for the Čech complex, thus making the Vietoris-Rips complex larger in general. But one can also prove that

$R_\epsilon(X) \subset C_{2\epsilon}(X)$ (see. Section III.3 in [26]), giving us in total

$$C_\epsilon(X) \subset R_\epsilon(X) \subset C_{2\epsilon}(X). \tag{2.3}$$

This is useful to us since the metric spaces that we will consider will be random samples of manifolds M. In which case, if the sample size of $X \subset M$ is large enough we can find $\epsilon > 0$ so that the open covering $\{B_\epsilon(x)\}_{x \in X}$ is in fact a good covering of M and therefore it will recover the topological information of the underlying M. This suggests that to recover the topological information we are forced to use the Čech complex, which is computationally expensive to compute for large sets of vertices. But Chapter 2.3 provides us a method to approximate the Čech complex using the Vietoris-Rips complex which is in fact much faster to compute since we don't need to verify intersections of multiple open balls.

Thus from this point on and for all the computation presented we will use the Vietoris-Rips complex, knowing that it provides a truthful computation of the topological features we will be working with.

2.1 Persistent homology

Definition 2.1.1. *A filtered complex $\mathcal{K}$ is a collection of simplicial complexes $\{K_i\}_{i=0}^n$ such that*

$$K_0 \subset K_1 \subset K_2 \subset \cdots \subset K_n$$

Definition 2.1.2. *A persistence complex is a family of chain complexes $\{C_*^i\}_i$ together with chain maps $f^i : C_*^i \to C_*^{i+1}$.*

A filtered complex $\mathcal{K}$ produces a persistence complex in a natural way using the chain maps induced by the inclusions.

$$\begin{tikzcd}
\cdots \arrow[r] & C_2(K_2) \arrow[r, "\partial_2"] & C_1(K_2) \arrow[r, "\partial_1"] & C_0(K_2) \arrow[r] & 0 \\
\cdots \arrow[r] & C_2(K_1) \arrow[u, "i^1"] \arrow[r, "\partial_2"] & C_1(K_1) \arrow[u, "i^1"] \arrow[r, "\partial_1"] & C_0(K_1) \arrow[u, "i^1"] \arrow[r] & 0 \\
\cdots \arrow[r] & C_2(K_0) \arrow[u, "i^0"] \arrow[r, "\partial_2"] & C_1(K_0) \arrow[u, "i^0"] \arrow[r, "\partial_1"] & C_0(K_0) \arrow[u, "i^0"] \arrow[r] & 0
\end{tikzcd}$$

Now we give the definitions of the main object that we will use throughout this work.

Definition 2.1.3. *Let R be a ring and I and totally ordered index set, then a persistence module is a family of R-modules M^i together with homomorphisms $\phi^{i,j} : M^j \to M^j$ for $i \leq j \in I$.*

Some of the most commonly used index sets are $I = \mathbb{Z}, \mathbb{R}^+, [a.b]$ with $a, b \in \mathbb{R}$.

The homology groups of the persistence complex obtained from a filtered complex is a persistence module where the maps are the ones induced by the inclusions.

$$PH_i(\mathcal{K}, \mathbb{Z}) : H_i(K_0; \mathbb{Z}) \xrightarrow{i^0_*} H_i(K_1; \mathbb{Z}) \xrightarrow{i^1_*} H_i(K_2; \mathbb{Z}) \xrightarrow{i^2_*} \cdots ,$$

more precisely one has the following definition.

Definition 2.1.4. *The i-th persistent homology groups are the images of the homomorphisms induced by the inclusion, $H_i^{r,s} = img(f_i^{r,s})$ where $f_i^{r,s} : H_i(K_r) \to H_i(K_s)$. The corresponding i-th persistent Betti numbers are the ranks of these groups, namely $\beta_i^{r,s} = rank(H_i^{r,s})$.*

Since these algebraic objects are in general difficult to understand, we would like to have a decomposition into simpler pieces. The atomic or basic pieces used for this decomposition are interval modules.

Definition 2.1.5. *An interval module is a persistence module $\mathcal{I}_{[r,s)} = \left\{ I_a, i_a^b \right\}$ where $I_a = G$ if $a \in [r, s)$ and 0 otherwise while $i_a^b : I_a \to I_b$ correspond to the identity whenever $r \leq a \leq b < s$.*

Under adequate conditions one can show (see [27]) that each persistent module obtained by computing persistent homology can be decomposed as

$$PH_i(\mathcal{K}) = \bigoplus_{[r,s)\in A} \mathcal{I}_{[r,s)} \tag{2.4}$$

2.2 Space of persistent diagrams

We start by defining a persistence diagram S_μ as the pair (S, μ) where

1. $S \subset \mathbb{W} := \left\{(x, y) \in \mathbb{R}^2 \mid 0 \le x < y\right\}$ such that for any $\epsilon > 0$ the set $u_\epsilon = \{(x, y) \in S \mid y - x > \epsilon\}$ is finite and

2. $\mu : S \to \mathbb{N}$.

We also define the persistence of a point $(x, y) \in S$ as $\mathsf{pers}((x, y)) := y - x$.

If we denote by $\mathcal{D}$ the set of persistence diagrams we can consider the bottleneck distance $d_B : \mathcal{D} \times \mathcal{D} \to \mathbb{R}^+$ to make $(\mathcal{D}, d_B)$ a complete metric space.

Definition 2.2.1. *A partial matching M between persistent diagrams S_μ and T_α is a bijection between subsets of S_μ and subsets of T_α, i.e. $M : S'_\mu \to T'_\alpha$ is a bijection where $S'_\mu \subset S_\mu$ and $T'_\alpha \subset T_\alpha$.*

If $(y, n) = M(x, m)$ we say that (x, m) is matched with (y, n). If $(z, k) \notin (S_\mu \setminus S'_\mu)$ or $(z, k) \notin T_\alpha \setminus T'_\alpha$ we call it unmatched.

Definition 2.2.2. *Given $\delta > 0$ and a partial matching M, then we have a δ-matching if:*

- *If $(x, m) \in S'_\mu$ is matched with (y, n) then $\|x - y\|_\infty < \delta$.*

- *If $(z, m) \in S_\mu \cup T_\alpha$ is unmatched, then $\mathsf{pers}(z) < 2\delta$.*

Definition 2.2.3. *The bottleneck distance $d_B : \mathcal{D} \times \mathcal{D} \to \mathbb{R}^+$ is defined as*

$$d_B(D_1, D_2) = \inf\{\delta > 0 : \textit{there is a } \delta\textit{-matching between } D_1, D_2\}.$$

In [28] the authors prove that d_B defines a metric on $\mathcal{D}$ and that $\mathcal{D}$ is the metric completion under the bottleneck distance of $\mathcal{D}_0 := \{(S, \mu) \in \mathcal{D} \mid S \text{ is finite}\}$.

CHAPTER 3

ADAPTIVE TEMPLATE SYSTEMS

3.1 Approximating Functions on Persistence Diagrams

The goal of this section is to provide the theoretical framework in which template functions are used as a means to approximating continuous functions on the space of persistence diagrams.

Definition 3.1.1. *The* ***bottleneck distance*** $d_B : \mathcal{D} \times \mathcal{D} \to \mathbb{R}^+$ *is defined as*

$$d_B(D_1, D_2) = \inf\{\delta > 0 : M : D_1 \to D_2 \textit{ is a } \delta\textit{-matching}\}.$$

In [28] it is shown that d_B defines a metric on $\mathcal{D}$ and that $\mathcal{D}$ is the metric completion of $\mathcal{D}_0 := \{(S, \mu) \in \mathcal{D} \mid S \text{ is finite}\}$ with respect to d_B. In [5] the authors present a complete characterization of (relatively) compact subsets of $(\mathcal{D}, d_B)$. In particular, one can prove that compact subsets of $(\mathcal{D}, d_B)$ have empty interiors (see Theorem 13 in [5]). This implies that $(\mathcal{D}, d_B)$ is not locally compact and therefore the compact-open topology on $C(\mathcal{D}, \mathbb{R})$, the space of real-valued continuous functions on $\mathcal{D}$, is not metrizable.

3.1.1 Template functions

Lemma 3.1.2. *Let* $C_c(\mathbb{W})$ *denote the set of real-valued continuous functions on* $\mathbb{W}$ *with compact support. For each* $f \in C_c(\mathbb{W})$*, the function* $\nu_f : \mathcal{D} \to \mathbb{R}$ *defined as*

$$\nu_f(S_\mu) = \sum_{x \in S} \mu(x) f(x)$$

is continuous.

Proof. See Lemma 23 in [5]. □

Definition 3.1.3. *A* ***coordinate system*** *for $\mathcal{D}$ is a collection $\mathcal{F} \subset C(\mathcal{D}, \mathbb{R})$ with the following property: for any two distinct $D, D' \in \mathcal{D}$, there exists $F \in \mathcal{F}$ such that $F(D) \neq F(D')$.*

Remark 3.1.4. *$\mathcal{F} = C(\mathcal{D}, \mathbb{R})$ is itself a coordinate system, but at least for computational purposes, it is too large to be of algorithmic use.*

Definition 3.1.5. *A* ***template system*** *for $\mathcal{D}$ is a set $T \subset C_c(\mathbb{W})$ such that $\{\nu_f : f \in T\}$ is a coordinate system for $\mathcal{D}$. The elements of T are called template functions.*

The main utility of template systems is that they can be used to construct dense subsets of $C(\mathcal{D}, \mathbb{R})$ with respect to the compact-open topology. In this topology, which is not metrizable as we mentioned above, two functions are deemed to be nearby if their values on compact sets are similar. Since the space of persistence diagrams is rather large and complicated, such comparisons (weaker than L^2 or $\|\cdot\|_\infty$) are desirable.

Theorem 3.1.6. *Let T be a template system for $\mathcal{D}$, let $\mathcal{C} \subset \mathcal{D}$ be compact, and let $F : \mathcal{C} \to \mathbb{R}$ be continuous. Then, for any $\epsilon > 0$ there exist $N \in \mathbb{N}$, a polynomial $p \in \mathbb{R}[t_1, \ldots, t_N]$, and template functions $f_1, \ldots, f_N \in T$ so that*

$$|p(\nu_{f_1}(D), \ldots, \nu_{f_N}(D)) - F(D)| < \epsilon \tag{3.1}$$

for all $D \in \mathcal{C}$.

Proof. See Theorem 29 in [5]. □

Corollary 3.1.7. *Let $T \subset C_c(\mathbb{W})$ be a template system for $\mathcal{D}$. Then, the collection of functions of the form*

$$\begin{aligned} \mathcal{D} &\longrightarrow \mathbb{R} \\ D &\mapsto p(\nu_{f_1}(D), \ldots, \nu_{f_N}(D)) \end{aligned}$$

where $N \in \mathbb{N}$, $p \in \mathbb{R}[t_1, \ldots, t_N]$, and $f_n \in T$, is dense in $C(\mathcal{D}, \mathbb{R})$ with respect to the compact-open topology.

The problem of constructing template systems of reasonable size (e.g., countable) is addressed by the following theorem.

Theorem 3.1.8. *Let $f \in C_c(\mathbb{W})$, $n \in \mathbb{N}$, $\mathbf{m} \in \mathbb{Z}^2$ and define*

$$f_{n,\mathbf{m}}(x) = f\left(nx + \frac{\mathbf{m}}{n}\right)$$

If f is nonzero, then $T = \{f_{n,\mathbf{m}} | n \in \mathbb{N},\ \mathbf{m} \in \mathbb{Z}^2\} \cap C_c(\mathbb{W})$ is a template system for $\mathcal{D}$.

Proof. See Theorem 30 in [5]. □

The goal of this paper is to investigate and identify data-driven methodologies for selecting the support of an initial template function f, as well as its most relevant re-scaled translates $f_{n,\mathbf{m}}$, so that the scalar features $\nu_{f_{n,\mathbf{m}}}$ can be used successfully in learning problems on persistence diagrams.

3.2 Adaptive template systems

In [5] two different template systems are suggested: tent functions and interpolating polynomials. Specifically, let

$$\widetilde{\mathbb{W}} = \{(x, y) \,|\, x \in \mathbb{R}, y \in \mathbb{R}_{>0}\}$$

be the conversion of $\mathbb{W}$ to the birth-lifetime plane. This conversion is defined by $(a, b) \in \mathbb{W} \mapsto (a, b - a) \in \widetilde{\mathbb{W}}$. The **template system of tent functions** in the birth-lifetime plane is defined as follows. Let $\boldsymbol{a} = (a, b) \in \widetilde{\mathbb{W}}$ and $0 < \delta < b$, then

$$g_{\boldsymbol{a},\delta}(x, y) = \max\left\{1 - \frac{1}{\delta}\max\{|x - a|, |y - b|\},\ 0\right\}$$

In a similar manner one can define a **template system of interpolating polynomials**. Given $\left\{(a_i, b_j)\right\}_{i,j} \subset \widetilde{\mathbb{W}}$ and $\left\{c_{i,j}\right\}_{i,j} \subset \mathbb{R}$, one can use Lagrange interpolating polynomials to construct a function f such that $f(a_i, b_j) = c_{i,j}$.

In general these two approaches require the user to input the meshes used in defining the template systems. By construction, such meshes define the support of the template functions. One shortcoming of this procedure, when applied to 3.1.6, is that without prior knowledge about the compact set $\mathcal{C} \subset \mathcal{D}$ the number of template functions that carry no information relevant to the problem can be high. This drawback is illustrated in 3.1.

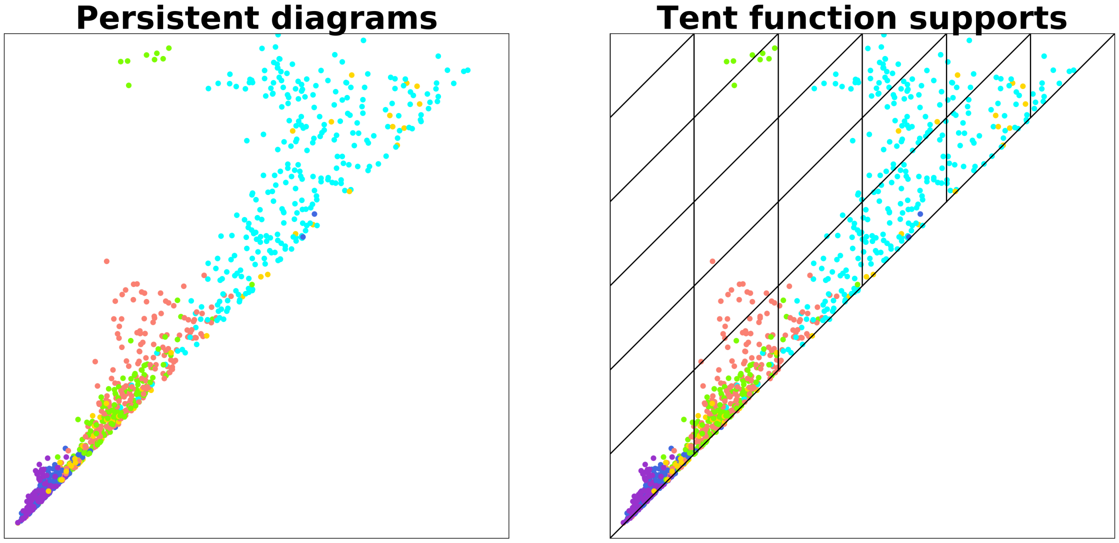

Figure 3.1 **Left:** Collection of persistent diagrams colored by class. **Right:** Mesh covering the collection on the left.

The main goal of this paper is to present a methodology to define the template system used in 3.1.6 that incorporates the prior information we have about the particular learning task. Such methodology is what we refer to as **adaptive template functions**.

Our approach to defining adaptive template functions is to first identify a collection of open ellipses in $\widetilde{\mathbb{W}}$ or $\mathbb{W}$ as in 3.2. Each ellipse in this collection will be the support for a template function defined in the following manner. Let $A \in M_{2\times2}(\mathbb{R})$ represent the quadratic form in two variables corresponding to an ellipse in the collection, and let $\boldsymbol{x} \in \mathbb{W}$ be its center. Then, the associated template function is

$$f_A(\boldsymbol{z}) = \begin{cases} 1 - (\boldsymbol{z} - \boldsymbol{x})^* A(\boldsymbol{z} - \boldsymbol{x}) & , \ (\boldsymbol{z} - \boldsymbol{x})^* A(\boldsymbol{z} - \boldsymbol{x}) < 1 \\ 0 & , \ (\boldsymbol{z} - \boldsymbol{x})^* A(\boldsymbol{z} - \boldsymbol{x}) \geq 1 \end{cases}$$

To obtain the collection of ellipses (A) mentioned above we will use and compare three different approaches; namely, Cover-Tree Entropy Reduction (CDER, [7]), Gaussian Mixture

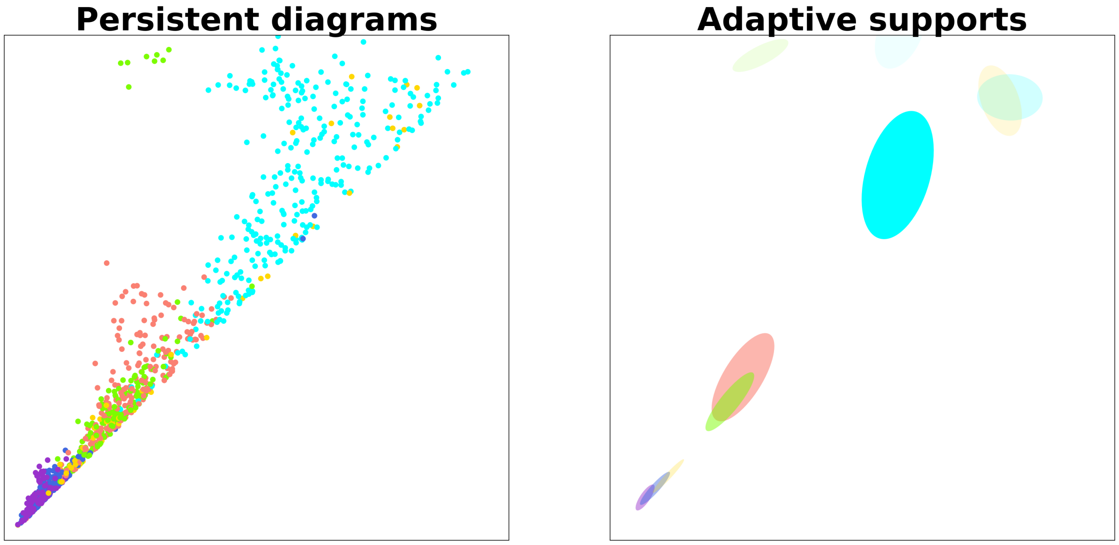

Figure 3.2 **Left:** Collection of persistent diagrams colored by class. **Right:** collection of open balls as supports for template functions.

Models (GMM) and Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN). We now provide a brief description of each method.

3.2.1 Cover-Tree Entropy Reduction - CDER

The main objective of this algorithm is to find a partial cover tree for a collection of labeled point clouds in $\mathbb{R}^n$. CDER searches for the cover tree in convex regions that are likely to have minimum local entropy. In this section we will explain the notion of entropy used by CDER, as it is relevant to explaining some of the results in 3.3.

Let $\chi = \{X_1, \ldots, X_N\}$ be a collection of point clouds $X_i \subset \mathbb{R}^d$ — which in our case will be persistence diagrams — and define $\underline{\chi} := \bigsqcup_{i=1}^{N} X_i$. We also have a labeling map at the level of point clouds

$$\lambda : \chi \to \{1, \ldots, L\} := \mathcal{L}.$$

Notice that we have a natural map $ind : \underline{\chi} \to \chi$ given by $ind(x) = X_i$ if $x \in X_i$. This allow us to define

$$\underline{\chi}_l := (\lambda \circ ind)^{-1}(l),$$

the set of all point in a point cloud labeled $l \in \mathcal{L}$.

Now we will assign weights to each point $x \in \underline{\chi}$ in the following manner:

1. Each label is equally likely among the data and $\underline{\chi}$ has a total weight of 1. Thus each label $l \in \mathcal{L}$ has an allocated weight of $1/L$.

2. Now consider
$$\lambda^{-1}(l) = \{\text{All point clouds with label } l\},$$
we assume again that each point cloud in $\lambda^{-1}(l)$ is equally likely, so each $X_i \in \lambda^{-1}(l)$ has an allocated weight of $1/(N_l L)$, where $N_l = \left\|\lambda^{-1}(l)\right\|$.

3. Finally each point in $x \in X_i$ is equally likely, and thus
$$w(x) = \frac{1}{\|X_i\| N_l L}$$

Definition 3.2.1. *Let* $\chi = \left\{X_1, \ldots, X_N \mid X_i \subset \mathbb{R}^d\right\}$ *be a collection of point clouds together with a label map* $\lambda : \chi \to \{1, \ldots, L\} := \mathcal{L}$ *and a weight function* $w : \underline{\chi} \to \mathbb{R}$. *For any convex and compact set* $\Omega \subset \mathbb{R}^d$ *we define the following quantities:*

1. The total weight of $l \in \mathcal{L}$ *in* Ω
$$w_l(\Omega) = \sum \left\{w(x) \mid x \in \Omega \cap \underline{\chi}_l\right\}.$$

2. the total weight of Ω
$$W(\Omega) = \sum_{l \in \mathcal{L}} w_l(\Omega)$$

Using the weights assigned above we can interpret $\dfrac{w_l(\Omega)}{W(\Omega)}$ as the probability of a point in Ω to have label l. This leads to the following definition.

Definition 3.2.2. *Let* χ, λ, *and* w *be as before. For any convex and compact set* $\Omega \subset \mathbb{R}^d$ *its entropy is defined by*
$$S(\Omega) = -\sum_{l \in \mathcal{L}} \frac{w_l(\Omega)}{W(\Omega)} \log_L \left(\frac{w_l(\Omega)}{W(\Omega)}\right).$$

This definition is borrowed from information theory. Notice that if all $w_l(\Omega)$ are roughly the same, then $S(\Omega) \approx 1$. But, if for example, Ω only contains points with a single label then we must have $S(\Omega) = 0$.

Remark 3.2.3. *Generally, we would expect the entropy to decrease as we select smaller subsets of* Ω*. However, this is not always the case.*

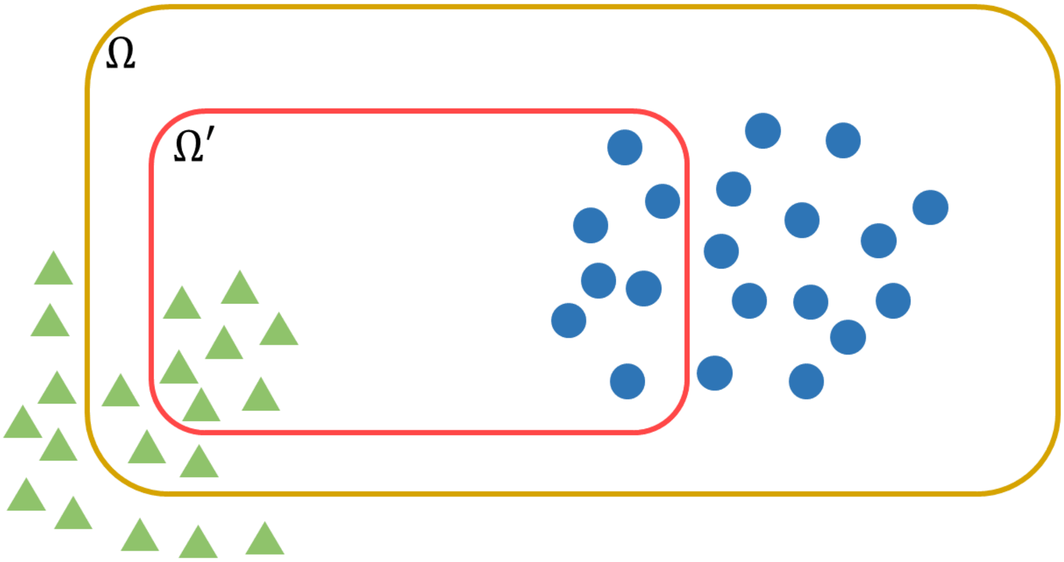

Figure 3.3 $\Omega' \subset \Omega$ but $S(\Omega) < S(\Omega')$.

For instance, consider point clouds X_1 *and* X_2 *such that* $|X_1| = 10$ *and* $|X_2| = 20$*, each point cloud with a different label* $\{0, 1\}$*, and compact sets* Ω, Ω' *as show in 3.3. We can easily see that* $\omega_0(\Omega') = \omega_1(\Omega') = 7/(20 \cdot 20 \cdot 2)$*, so* $W(\Omega') = 7/(20 \cdot 20 \cdot 2) + 7/(20 \cdot 20 \cdot 2) = 7/400$*. Finally* $S(\Omega') = 1$*. A similar calculation will show that* $S(\Omega) \approx 0.9182$*, and thus* $\Omega' \subset \Omega$ *but* $S(\Omega) < S(\Omega')$*.*

3.2.2 Gaussian Mixture Models - GMM

This algorithm is an implementation of the Expectation-Maximization (EM) algorithm to fit Gaussian models to a collection of points. Recall that an EM algorithm is an iterative method to solve maximum likelihood estimation of parameters for a given model; in our case Gaussian models.

The EM algorithm iterates over two steps: an expectation step and a maximization step. The first step defines the expected value of the log likelihood function using a given set of parameters for the model. The maximization step finds a new set of parameters that maximizes the previously described expected value.

3.2.3 Hierarchical Density-Based Spatial Clustering of Applications with Noise - HDBSCAN

HDBSCAN [9] is a hierarichal clustering algorithm extending BDSCAN. The latter, finds clusters as the connected components of a graph. The vertices of this graph are the elements in the data set after removing "noise points" and the adjacency is defined by a user-provided parameter ϵ.

HDBSCAN constructs a hierarchical collection of DBSCAN solutions by changing the parameter used to compute the adjacency in the DBSCAN algorithm. Once this hierarchy of solutions is obtained, the algorithm extracts from its hierarchy dendrogram a sumarized collection of significant clusters.

3.3 Experimental Results

Thus far we have developed a methodology for deriving adaptive template systems from labeled persistence diagrams. The main idea is to use algorithms such as CDER, GMM and HDBSCAN to identify the supports of the functions in the template system. We will use these adaptive templates to produce feature vectors in order to solve supervised classification problems. In this section we present three examples of supervised learning, where adaptive template functions yield featurizations that improve the classification accuracy or robustness of the classification model. Our implementation of adaptive template systems with CDER, GMM and HDBSCAN as well as the scripts to replicate all the results in this section can be found in the associated GitHub repository[1].

3.3.1 Manifolds

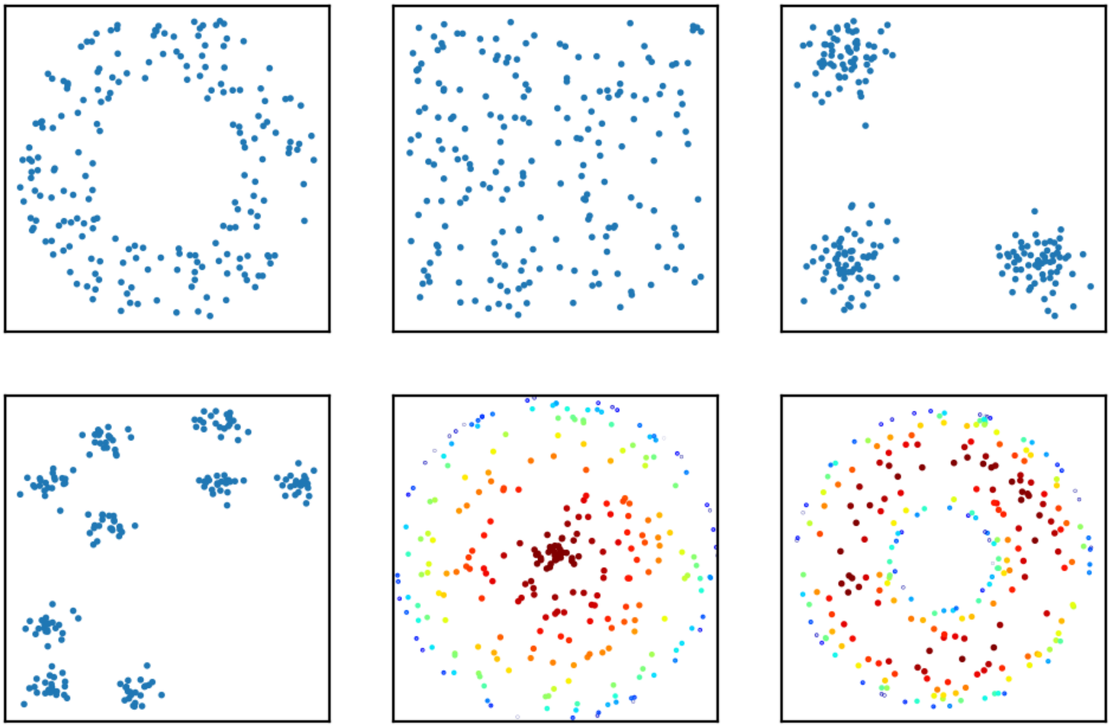

Figure 3.4 Example of the 6 manifolds, from top left to bottom right we have: annulus, cube, 3 clusters, 3 cluster of 3 clusters, S^2 (projected on the xy-plane) and torus (projected on the xy-plane).

For this example we revisit an experiment presented in [5] and [6]. We generated point clouds sampled from different manifolds in $\mathbb{R}^2$ or $\mathbb{R}^3$. Each point cloud has 200 points and the manifolds considered are the following: an **annulus** with inner radius 1 and outer radius 2 centered at $(0, 0)$, **3 clusters** of points drawn from normal distributions with means $(0, 0)$, $(0, 2)$ and $(2, 0)$ all with standard deviation 0.05, **3 cluster of 3 clusters** of points drawn from normal distributions with standard deviation 0.05 and means $(0, 0)$, $(0, 1.5)$, $(1.5, 0)$, $(0, 4)$, $(1, 3)$, $(1, 5)$, $(3, 4)$, $(3, 55)$ and $(4.5, 4)$, **cube** defined as $[0, 1]^2 \subset \mathbb{R}^2$, **torus** obtained

[1] https://github.com/lucho8908/adaptive_template_systems.git

from rotating a circle of radius 1 centered at $(2, 0)$ on the xz-plane around the z-axis and **sphere** $S^2 \subset \mathbb{R}^3$ with uniform noise in $[-0.05, 0.05]$ on the normal direction.

We used CDER, GMM and HDBSCAN to generate the supports of the functions that form our template systems. We reserved 33% of the data for testing, and trained a kernel ridge regression model on the remaining data (%67). In addition, we investigate the effect of increasing the number of point clouds sampled from each manifold.

			CDER		**GMM**		**HDBSCAN**	
	Train	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**
10	0.99 ± 0.9	0.96 ± 3.2	0.99 ± 0.001	0.98 ± 0.034	0.99 ± 0.001	0.91 ± 0.075	1.00 ± 0.000	0.90 ± 0.092
25	0.99 ± 0.3	0.99 ± 1.0	0.99 ± 0.001	0.99 ± 0.001	0.99 ± 0.004	0.99 ± 0.009	1.00 ± 0.000	0.89 ± 0.031
50	1.00 ± 0.0	0.99 ± 0.9	0.99 ± 0.001	0.99 ± 0.001	0.99 ± 0.002	0.99 ± 0.008	1.00 ± 0.000	0.95 ± 0.003
100	0.99 ± 0.1	0.99 ± 0.4	0.99 ± 0.001	0.99 ± 0.001	0.99 ± 0.004	0.99 ± 0.005	1.00 ± 0.000	0.97 ± 0.011
200	0.99 ± 0.1	0.99 ± 0.3	0.99 ± 0.002	0.99 ± 0.005	0.99 ± 0.002	0.99 ± 0.003	1.00 ± 0.000	0.98 ± 0.005

Table 3.1 Manifold classification: Each row corresponds to the number of samples taken from each manifold. The first two columns show the best results reported in [5].

In 3.1 we present the mean accuracy of the model on both the training and testing data after averaging the results over 10 experiments for each sampling size. Furthermore, 3.1 contains the results obtained from using CDER, GMM and HDBSCAN to find the adaptive templates as well as the best results presented by [5] on the same classification problems.

The first important feature to highlight is that for all the different sampling sizes the adaptive templates accuracy results show a smaller standard deviation than the results reported in [5]. At the same time, the mean accuracy of our methodology is comparable with the state of the art results (in [5]).

It is worth mentioning that across all the different methods used in this work to obtain adaptive template systems, CDER provides the most stable results. This is specially significant for the smaller sample size (the first row in 3.1).

3.3.2 Shape data

In this example we consider the synthetic SHREC 2014 data set [10], of which some instances are shown in Figure 3.5. We compare our result to the methods reported in [5] by extracting features using adaptive template systems for the same data from [4] and [5]. In [4] the authors defined a function on each mesh using a heat kernel signature, [29], for 10 parameters and computed persistent diagrams for dimensions 0 and 1.

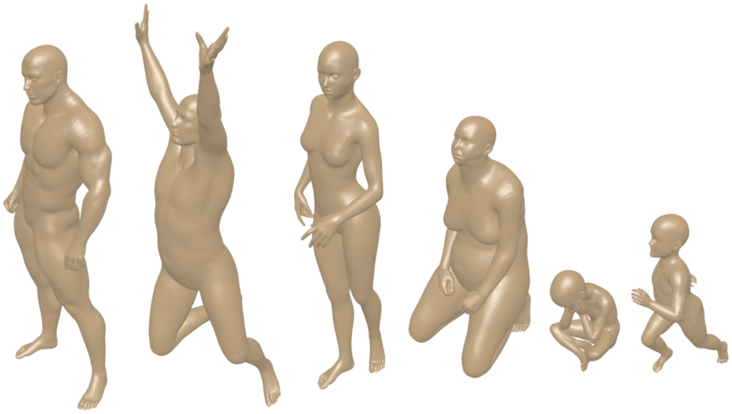

Figure 3.5 Examples of shapes and poses in the SHREC synthetic data set.

For each one of the 10 parameter values we have 300 pairs of persistence diagrams and the goal of the problem is to predict the human model. The models correspond to 5 different poses for people labeled as male, female and child; giving us a total of 15 labels. Lets us remark that each one of the 10 parameters yields a different classification problem.

	Polynomial	**RBF**	**Sigmoid**

Kernel Ridge	CDER = 0.6
	GMM = 0.3
	HDBSCAN = 0.1
		CDER = 0.4
		GMM = 0.4
		HDBSCAN = 0.1
			CDER = 0.2
			GMM = 0.4
			HDBSCAN = 0.4

SVM	CDER = 0.7
	GMM = 0.3
	HDBSCAN = 0.0
		CDER = 0.7
		GMM = 0.2
		HDBSCAN = 0.0
			CDER = 0.8
			GMM = 0.2
			HDBSCAN = 0.0

Random Forest	CDER = 0.6
	GMM = 0.3
	HDBSCAN = 0.1
		-
		-
		-
			-
			-
			-

Table 3.2 Shape classification: Portion out of the 10 problems, for which each adaptive template system yields the best classification results. The cells with a dash (-) indicate that no computations were carried out.

We considered CDER, GMM and HDBSCAN as methods to obtain adaptive template systems. For each one of these template systems we used three different kernel methods to solve the classification problems, namely, kernel ridge regression, kernel support vector machines (SVM) and random forest. Finally, three different kernels were examined, polynomial, radial basis function (RBF) and sigmoid kernels.

3.2 shows the portion, out of the 10 problems, for which a given adaptive template system yields the best classification results. For instance, the entry corresponding to ridge regression with a polynomial kernel (first row and first column). shows that the CDER template system yields the best classification accuracy in 6 our of 10 problems, GMM is the best for 3 out of 10 and HDBSCAN is superior in 1 out of the 10 problems.

With this interpretation of 3.2 in mind, we can see that CDER adaptive template systems yield more accurate classification results than GMM or HDBSCAN templates. This holds true for all but one, of the kernel and kernel method combinations explored in this experiment.

	Global features		**Local features**					
	Interpolating Polynomials		**Tent functions**		**Adaptive templates**			
Freq.	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**		
1	0.99 ± 0.3	0.90 ± 5.3	0.08±0.30	0.03±0.5	0.79±0.01	0.73±0.02	GMM	Kernel Ridge
2	1.00 ± 0.0	0.95 ± 2.4	0.08±0.40	0.03±1.00	0.84±0.00	0.8±0.03	GMM	
3	0.99 ± 0.5	0.90 ± 2.0	0.80±1.3	0.44±4.3	0.7±0.01	0.66±0.03	HDBSCAN	
4	0.98 ± 0.9	0.84 ± 3.9	0.89±1.5	0.69±4.9	0.7±0.01	0.67±0.03	GMM	
5	0.99 ± 0.4	0.93 ± 2.2	0.76±2.7	0.58±7.9	0.78±0.04	0.76±0.08	CDER	
6	0.98 ± 0.5	0.92 ± 1.8	0.96±0.67	0.89±1.7	0.97±0.01	0.92±0.03	CDER	SVM
7	0.99 ± 0.4	0.95 ± 1.4	0.98±0.60	0.94±2.5	0.96±0.02	0.94±0.05	CDER	
8	0.99 ± 0.4	0.94 ± 2.2	0.91±1.2	0.89±3.3	1±0.00	0.88±0.05	CDER	Random Forest
9	0.98 ± 1.3	0.92 ± 2.1	0.64±2.3	0.53±3.8	1±0.00	0.88±0.03	CDER	
10	0.97 ± 1.1	0.89 ± 4.6	0.27±3.4	0.18±5.6	1±0.00	0.9±0.08	CDER	

Table 3.3 Shape classification: Classification accuracy for adaptive templates (ours), tent functions and interpolating polynomials [5]

Having stablished how CDER, GMM and HDBSCAN compare across different kernel methods and kernels, next we compare our classification results with the state of the art. 3.3 contains our classification accuracy on the training and testing shape data, as well as the results obtained using the tent functions and interpolating polynomials from [5].

The first important feature to remark is that for 7 out of the 10 problems, the model obtained from adaptive coordinates has less overfitting than interpolating polynomials. Meaning that for most of the classification problems studied in this example, adaptive template systems provide a more robust classification model. An additional argument in favor of the robustness of our approach is that the standard deviation of all the models using adaptive templates is smaller than those presented in [5]. Moreover, when comparing adaptive template systems with tent functions (see 3.3) we attain better classification results on the testing set across all the problems presented. This highlights the benefits of adaptive versus non-adaptive local template systems.

It is pertinent to mention that the results in 3.3 correspond to a specific selection of parameters for each kernel and regularization in each classification method. In fact, from our methodology we can find a combination of kernel parameters and regularization that yields higher classification accuracy in the training set. But, for such models the overfitting issues are more noticeable.

Finally, since the end goal of this problem is to solve the multiclass classification problem in the synthetic SHREC 2014 data set. We can select the problem corresponding to the frequency 6 in the heat kernel signature (row 6 in 3.3) as the one that gives us the best classification accuracy while minimizing the overfiting concern. Such result is obtained using a CDER template system and a regularized SVM method.

3.3.3 Protein classification

We consider next the data set PCB00019 from the Protein Classification Benchmark Collection [11]. This problem set contains $1,357$ proteins and 55 classification tasks. Our results will be compared to those reported in [3].

To compare our results with the ones in [3], we report the average classification accuracy over the 55 classification tasks in the data set PCB00019.

3.4 shows the average classification accuracy and standard deviation for each of the classification tasks from PCB00019. Here we used a regularized kernel ridge regression with polynomial, RBF and sigmoid kernels. The last row in 3.4 displays the average classification

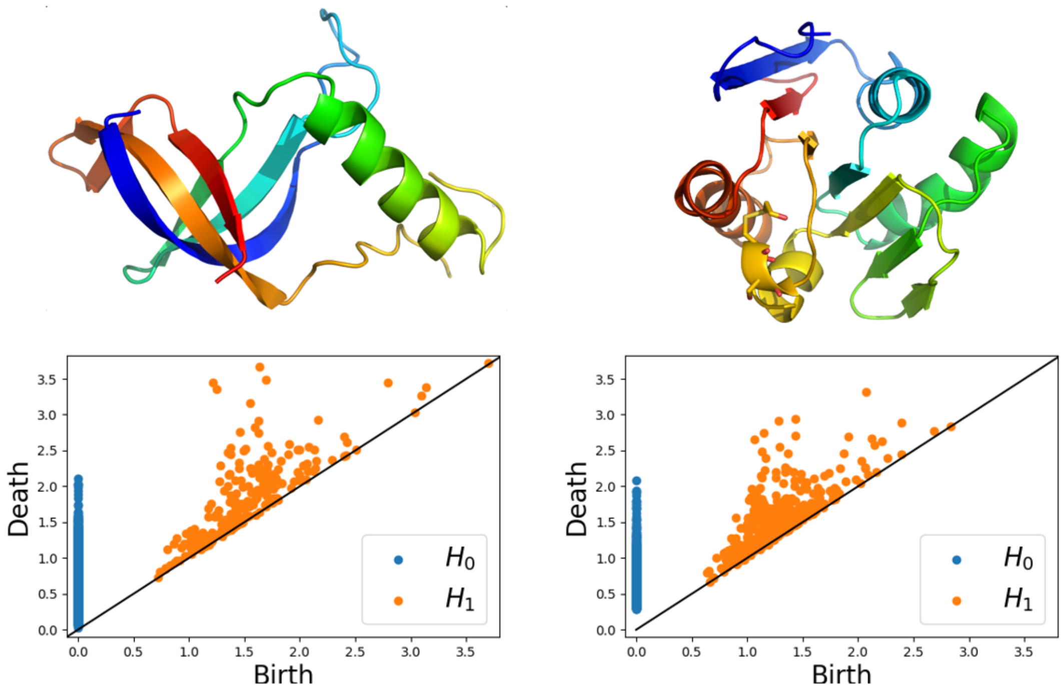

Figure 3.6 Examples of data points in PCB00019 and their corresponding persistent diagrams. **Top row:** Protein domains from [30]. **Bottom row:** Persistent diagrams in dimensions 0and 1.

		Train	**Test**
CDER	**Polynomial**	0.90 ± 0.07	0.98 ± 0.02
	RBF	0.91 ± 0.06	0.97 ± 0.02
	Sigmoid	0.90 ± 0.07	0.98 ± 0.02
Topological features in [3]		-	0.82 ± —-

Table 3.4 Protein classification: average classification accuracy for the 55 tasks in the data set PCB00019.

accuracy for the testing set reported in [3]. We note that the authors do not report standard deviation or average classification accuracy for the testing set.

It is meaningful to remark that in [3] topological features are used to solve the classification problems. Those features were constructed using persistence diagrams as to reflect relevant properties of the proteins in the given data set.

3.4 Discussion

This paper investigates the viability of utilizing data-driven methodologies to localize features in persistence diagrams. These features are used in subsequent supervised learning tasks for classification problems where shape is an important feature. We examined three different algorithms, CDER, GMM and HBDSCAN to produce adaptive template functions. Through extensive testing with real and synthetic data sets, we demonstrate that CDER provides a more robust collection of adaptive features while maintaining classification accuracy on par with the state of the art. In terms of time complexity, CDER also outperforms GMM and HDBSCAN for the problems here considered.

CHAPTER 4

PERSISTENT CUP PRODUCTS AND QUIASI-PERIODICITY DETECTION

4.1 Persistent cohomology

Given and abstract simplicial complex $\mathcal{K}$ any finite set σ in $\mathcal{K}$ is called a simplex of dimension $|\sigma| - 1$, for example a simplex with one element is called a 0-simplex, a simplex with two elements is a 1-simplex. We will denote a simplex as $[v_0, \ldots, v_n]$ and the points v_i are called the *vertices* of the simplex. Using this notation we will say that $[v_0, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n]$ is a *face* of $[v_0, \ldots, v_n]$ and $[v_0, \ldots, v_n]$ is a *coface* of $[v_0, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n]$.

Example 3. *This first example will provide the basic example that we will use throughout this paper.*

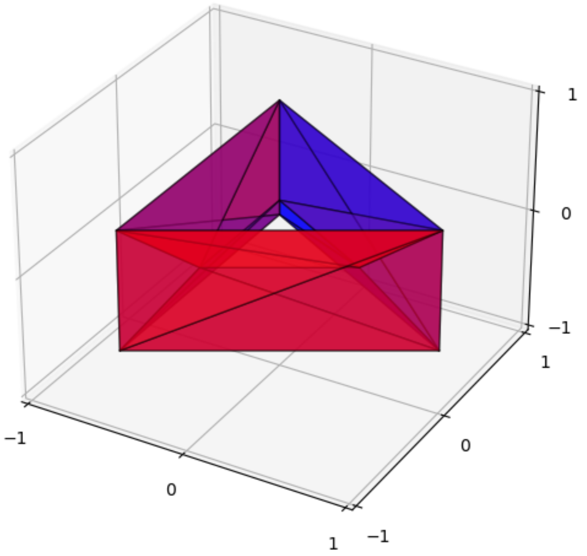

Figure 4.1 Simplicial complex structure on the torus

In this example we present a simplicial complex that represents a torus by letting the base *set* $S = \{0, \ldots, 8\}$. *We define the* 1*-simplices to be*

$$\begin{aligned}&\{[0,1],[1,2],[0,2],[3,4],[4,5],[3,5],[6,7],[7,8],[6,8],\\&[0,3],[3,6],[0,6],[1,4],[4,7],[1,7],[2,5],[5,8],[2,8],\\&[2,4],[1,3],[0,5],[5,7],[4,6],[3,8],[1,8],[0,7],[2,6]\},\end{aligned}$$

and the 2*-simplices to be defined as*

$$\begin{aligned}&\{[0,1,3],[1,3,4],[1,2,4],[2,4,5],[0,2,5],[0,3,5],\\&[4,6,7],[3,4,6],[4,5,7],[5,7,8],[3,5,8],[3,6,8],\\&[0,6,7],[0,1,7],[1,7,8],[1,2,8],[2,6,8],[0,2,6],[0,1,2]\}\end{aligned}$$

A family $\{K_i\}_{i \in I}$ of simplicial complexes indexed by a totally ordered set I and such that $K_i \subset K_j$ is a sub-simplex whenever $i \leq j$ is called a *filtered complex.* A spacial case of filtered complexes arises whenever $K_i = K_{i-1} \cup \{\sigma_i\}$, we call this type of filtration a *simplexwise filtration.* In general any filtered complex can be made into a simplexwise filtration by refining and reindexing maps as presented by [23], therefore from this point on we will consider any filtered complex to be a simplexwise filtrations and we will refer to it just as a filtration.

In computational topology one of the main object of study are Vietoris-Rips filtrations due to the computational properties they hold (see [26]). We will make use of this filtrations as well since they naturally arise as part of many computational tools such as Ripser (see [23]). The Vietoris-Rips filtration is usually applied to obtain persistent diagrams from data sets when seen as finite metric spaces, in other words if X is a finite subset of a metric space (M, d). We define the *Vietoris-Rips filtration of* X to be the family of simplicial complex S

$$R_\epsilon(X) = \{S \subset X : S \neq \emptyset \text{ and } \mathsf{diam}(S) \leq \epsilon\}$$

together with natural inclusions $R_\epsilon(X) \hookrightarrow R_{\epsilon'}(X)$ whenever $\epsilon < \epsilon'$.

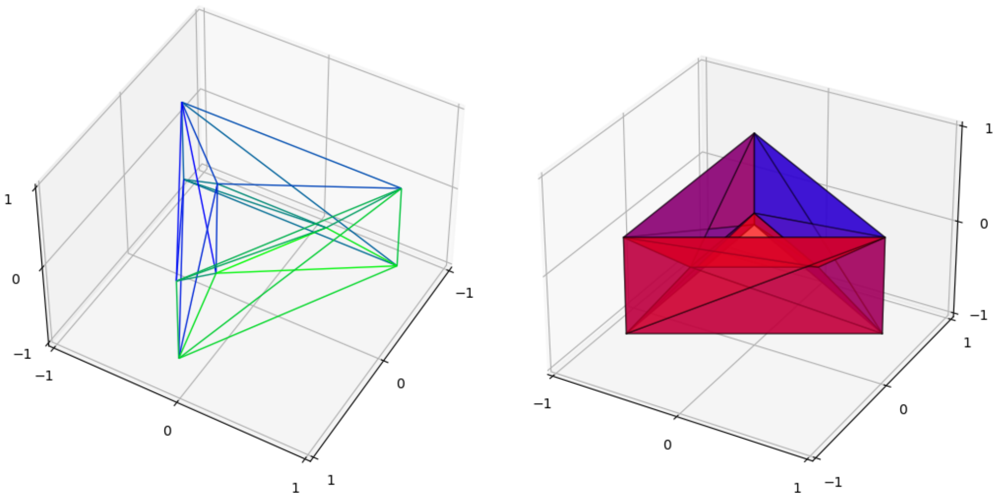

Figure 4.2 Filtration of a simplicial complex representation of a torus in $\mathbb{R}^3$.

Example 4. *Here we have filtered complex of a Torus (see Section 2.1 on [25]).*

The filtered complex is given by the coloration of the edges and vertices, meaning the edges in green enter the filtration sooner than those in blue, while the faces in blue enter the filtration sooner than those in red.

Given a simplicial complex K and a field $\mathbb{F}$ we can define the *group of p-cochains*, denoted by $C^p(K)$, to be the collection of homomorphisms from $C_p(K)$ to $\mathbb{F}$. The *p-coboundary* of an element $\sigma \in C^p(K)$, denoted as $\delta^p(\sigma) \in C^{p+1}(K)$ is defined by the values it takes in the chains $C_p(K)$, namely

$$\begin{aligned}(\delta^p(\sigma))\,([v_0,\ldots,v_{p+1}]) &:= \sigma\left(\partial_{p+1}([v_0,\ldots,v_{p+1}])\right)\\ &= \sum_{j=0}^{p+1}(-1)^j\sigma\left([v_0,\ldots,\hat{v_j},\ldots,v_{p+1}]\right)\end{aligned}$$

where ∂_p is as defined in Chapter 2.1.

With this map in mind we define the *p-cocycles* of K as $Z^p(K) = \ker(\delta^p)$ and the *p-coboundaries* of K to be $B^p(K) = \text{img}(\delta^{p-1})$. Using this notions the *p-th cohomology of K* is defined as

$$H^p(K;\mathbb{F}) = Z^p(K)/B^p(K) = \ker(\delta^p)/\text{img}(\delta^{p-1}). \tag{4.1}$$

Example 5. *Consider one of the complexes the filtration in Chapter 4 and fix* $\mathbb{F} = \mathbb{Z}_2$

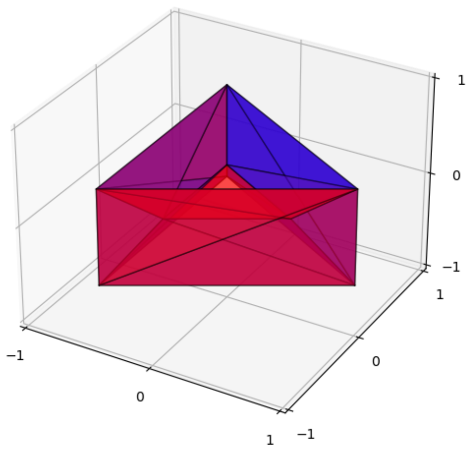

Figure 4.3 Cellular complex on the Torus.

This simplicial complex contains 9 *vertices,* $\{v_0, \dots, v_8\}$*,* 27 *edges denoted as* e_i *for* $i = 0, \dots, 26$ *and finally* 18 *faces similarly labeled* f_j *for* $j = 0, \dots, 17$*.*

It is easy to verify that this simplicial complex has the cohomology of a torus, namely $H^0(K; \mathbb{Z}_2) = \mathbb{Z}_2$*,* $H^1(K; \mathbb{Z}_2) = \mathbb{Z}_2 \times \mathbb{Z}_2$ *and* $H^2(K; \mathbb{Z}_2) = \mathbb{Z}_2$*.*

Given a filtration $\mathcal{K} = \{K_i\}_{i \in I}$ the inclusion maps $K_i \hookrightarrow K_j$ for $i < j$ induce homomorphisms $f^p_{i,j} : H^p(K_j; \mathbb{F}) \to H^p(K_i; \mathbb{F})$ giving place to a sequence

$$PH^p(\mathcal{K}; \mathbb{F}) := H^p(K_1; \mathbb{F}) \longleftarrow H^p(K_2; \mathbb{F}) \longleftarrow \cdots \longleftarrow H^p(K_m; \mathbb{F}) \qquad (4.2)$$

The *p-th persistent cohomology groups of* $\mathcal{K}$ are defined as $H^p_{i,j}(\mathcal{K}; \mathbb{F}) = \mathrm{img}(f^p_{i,j})$. In particular if $i = j$ we recover the standard cohomology groups for each element in the filtration, i.e. $H^p_{i,i}(\mathcal{K}; \mathbb{F}) = H^p(K_i; \mathbb{F})$ for any $i \in I$.

Moreover the sequence of cohomology groups in Chapter 4.2 form a pointwise finite-dimensional persistent vector space. Meaning that each cohomology group is a finite dimensional vector space over the field $\mathbb{F}$. Once this condition is met, in [31] is shown that any pointwise finite-dimensional persistence module is a direct sum of interval modules.

```
**Require:** $D = (m \times n)$ filtration boundary matrix.
**Ensure:** $R$ is reduce, $V$ is invertible upper triangular and $R = DV$.
**Initialize:** $(R, v) = (D, I)$
**for** $j = 1, \ldots, n$ **do**
    **while** $\exists i < j$ *such that* $low_R(i) = low_R(j)$ **do**
        $\lambda = R[\mathrm{low}_R(i), j]/R[\mathrm{low}_R(i), i]$
        $R[:, j]- = \lambda R[:, i]$
        $V[:, j]- = \lambda V[:, i]$
**return** $(R, V)$
```

Algorithm 4.1 Reduction algorithm (pHcol)

Each interval module is defined over an interval $I \subset \mathbb{R}$ and the corresponding interval module V is given by letting $V_t = \mathbb{F}$ for any $t \in I$, $V_t = 0$ for any $t \notin I$ and the maps between $\rho_{st} : V_s \to V_t$ correspond to the identity $\mathbb{F} \to \mathbb{F}$.

This decomposition of persistent cohomology in terms of interval modules presented in [31] allow us to represent the persistent cohomology as a *barcode* or *persistent diagram.* These representations are further explained and explored in Chapter 6 and furthermore in [32].

4.1.1 Computing persistent cohomology

The decomposition in interval modules (see [31]) can be computed by standard Ripser algorithm as shown in [23] and Algorithm 4.1. The output of such algorithm is a matrix decomposition $R = DV$, where the matrix D is assembled from the boundary maps, V is an invertible upper triangular matrix and R is used to encode the persistent diagram of $PH^*(\mathcal{K} : \mathbb{F})$. First, given a filtration $\mathcal{K}$ we assemble the coboundaries $\delta(\sigma)$ as columns ordered according to the filtration order into the *filtration boundary matrix D*. If the maximum dimension on $\mathcal{K}$ is d then we can compute all persistent cohomology up to dimension $d - 1$.

Algorithm 4.1 stopping condition is defined in terms of the matrix R being reduced as defined by [33]. More specifically, R is reduced if for any pair of non-zero columns i-th and j-th then $\mathrm{low}_R(i) \neq \mathrm{low}_R(j)$, where $\mathrm{low}_R(i)$ represents the index of the lowest row with a non-zero entry in the i-th column of R.

In [33] the authors prove that given a decomposition of the form $R = DV$ where V is invertible and upper triangular and R is in reduced form, then the pairings define by

- (i, j) if $\text{low}_R(j) = i$,
- (i, ∞) if the i-th column of R is zero and there is no j such that $\text{low}_R(j) = i$

are unique in general and define the *persistent diagram* or *barcode* of the filtration $\mathcal{K}$. Notice that this result does not imply uniqueness of the matrices R and V.

One of the optimization presented in [34] to use Algorithm 4.1 consist of replacing the matrix D by $D^{\perp}$, which is define entry wise as $D^{\perp}[i, j] = D[n + 1 - i, n + 1 - j]$. We will consider this matrix for decomposition purposes since this will coincide with the exact implementation presented in [23].

An important note worth highlighting is the fact that given a pair (i, j) in the persistent diagram of $PH^*(\mathcal{K}; \mathbb{F})$ we will say that i is the *cohomological death* of a class, while j is the *cohomological birth* of a class.

Example 6. *In this example we will take the filtration on Chapter 4 Example 4 and will compute its persistent cohomology using a decomposition of $D^{\perp}$. Since showcasing the entirety of the matrix $D^{\perp}$ would be too cumbersome, we will illustrate the process on a smaller submatrix. In particular, we will use the submatrix that contains the 0-coboundaries. In other words, each column of such matrix contains the coboundary of a vertex v_i in the filtration on Chapter 4.*

By using the reduction Algorithm 4.1 we obtain a reduced matrix $R^{\perp}$

and a matrix of elementary operations $V^{\perp}$.

From this submatrix alone is impossible to read the complete persistent cohomology of the filtered complex in Chapter 4, but using the same reduction algorithm applied to the complete matrix $D^{\perp}$ we can easily find that the pairs in the persistent cohomology are given by following the rules stated in [34].

$$\begin{array}{c|ccccccccc|}
D^{\perp} & v_8 & v_7 & v_6 & v_5 & v_4 & v_3 & v_2 & v_1 & v_0 \\
e_{26} & 1 & & & & & & & & 1 \\
e_{25} & 1 & & 1 & & & & & & \\
e_{24} & & & 1 & 1 & & & & & \\
e_{23} & & & & 1 & & 1 & & & \\
e_{22} & & & & & & 1 & 1 & & \\
e_{21} & & & & & & & 1 & & 1 \\
e_{20} & 1 & & & & & & 1 & & \\
e_{19} & & 1 & & & & & 1 & & \\
e_{18} & 1 & 1 & & & & & & & \\
e_{17} & 1 & & & 1 & & & & & \\
e_{16} & 1 & & & & 1 & & & & \\
e_{15} & & & & 1 & 1 & & & & \\
e_{14} & & & & 1 & & & 1 & & \\
e_{13} & & & & 1 & & & & 1 &
\end{array}, \quad
\begin{array}{c|ccccccccc|}
 & & & & & \vdots & & & & \\
e_{12} & & & & & & & 1 & 1 & \\
e_{11} & & & 1 & & & & & & 1 \\
e_{10} & & 1 & & & & & & 1 & \\
e_{9} & & & 1 & & & & & 1 & \\
e_{8} & & 1 & 1 & & & & & & \\
e_{7} & & & 1 & & & 1 & & & \\
e_{6} & & 1 & & & 1 & & & & \\
e_{5} & & 1 & & & & 1 & & & \\
e_{4} & & & & & 1 & 1 & & & \\
e_{3} & & & & & & 1 & & & 1 \\
e_{2} & & & & & 1 & & & 1 & \\
e_{1} & & & & & 1 & & & & 1 \\
e_{0} & & & & & & & & 1 & 1 \\
 & v_8 & v_7 & v_6 & v_5 & v_4 & v_3 & v_2 & v_1 & v_0
\end{array}$$

$$\begin{array}{c|ccccccccc|}
R^{\perp} & v_8 & v_7 & v_6 & v_5 & v_4 & v_3 & v_2 & v_1 & v_0 \\
e_{26} & 1 & & & & & & & & \\
e_{25} & 1 & & 1 & & & & & & \\
e_{24} & & & 1 & 1 & & & & & \\
e_{23} & & & & 1 & & 1 & & & \\
e_{22} & & & & & & 1 & 1 & & \\
e_{21} & & & & & & & 1 & & \\
e_{20} & 1 & & & & & & 1 & & \\
e_{19} & & 1 & & & & & 1 & & \\
e_{18} & 1 & 1 & & & & & & & \\
e_{17} & 1 & & & 1 & & & & & \\
e_{16} & 1 & & & & 1 & & & & \\
e_{15} & & & & 1 & 1 & & & & \\
e_{14} & & & & 1 & & & 1 & & \\
e_{13} & & & & 1 & & & & 1 &
\end{array}, \quad
\begin{array}{c|ccccccccc|}
 & & & & & \vdots & & & & \\
e_{12} & & & & & & & 1 & 1 & \\
e_{11} & & & 1 & & & & & & \\
e_{10} & & 1 & & & & & & 1 & \\
e_{9} & & & 1 & & & & & 1 & \\
e_{8} & & 1 & 1 & & & & & & \\
e_{7} & & & 1 & & & 1 & & & \\
e_{6} & & 1 & & & 1 & & & & \\
e_{5} & & 1 & & & & 1 & & & \\
e_{4} & & & & & 1 & 1 & & & \\
e_{3} & & & & & & 1 & & & \\
e_{2} & & & & & 1 & & & 1 & \\
e_{1} & & & & & 1 & & & & \\
e_{0} & & & & & & & & 1 & \\
 & v_8 & v_7 & v_6 & v_5 & v_4 & v_3 & v_2 & v_1 & v_0
\end{array}$$

$$\begin{aligned}
PH^0(\mathcal{K};\mathbb{Z}_2) = \{&[v_0,\infty),[v_8,e_{16}),[v_5,e_{13}),[v_2,e_{12}),[v_6,e_7),[v_7,e_5),[v_3,e_3),\\
&[v_4,e_1),[v_1,e_0)\}\\
PH^1(\mathcal{K};\mathbb{Z}_2) = \{&[e_9,\infty),[e_{26},f_{17}),[e_{25},f_{16}),[e_{24},f_{15}),[e_{23},f_{14}),[e_{22},f_{13}),\\
&[e_{21},f_{19}),[e_{20},f_{11}),[e_{19},f_{12}),[e_{18},f_{10}),[e_{17},f_9),[e_{15},f_8),[e_{14},f_7),\\
&[e_{11},f_6),[e_{10},f_5),[e_8,f_4),[e_6,f_3),[e_4,f_2),[e_2,f_1)\}\\
PH^2(\mathcal{K};\mathbb{Z}_2) = \{&[f_{18},\infty)\}
\end{aligned}$$

$$\begin{array}{c|ccccccccc}
V^{\perp} & v_8 & v_7 & v_6 & v_5 & v_4 & v_3 & v_2 & v_1 & v_0 \\
\hline
v_8 & 1 & & & & & & & & 1 \\
v_7 & & 1 & & & & & & & 1 \\
v_6 & & & 1 & & & & & & 1 \\
v_5 & & & & 1 & & & & & 1 \\
v_4 & & & & & 1 & & & & 1 \\
v_3 & & & & & & 1 & & & 1 \\
v_2 & & & & & & & 1 & & 1 \\
v_1 & & & & & & & & 1 & 1 \\
v_0 & & & & & & & & & 1
\end{array}$$

4.2 Persistent cup products

Given K a simplicial complex and a field $\mathbb{F}$, if we take a k-cochain $\phi \in C^k(K;\mathbb{F})$ and an l-cochain $\psi \in C^l(K;\mathbb{F})$ then *cup product* between ϕ and ψ is a $(k+l)$-cochain $\phi \smile \psi \in C^{k+l}(K;\mathbb{F})$. To define this cochain we need to define the values that it takes on any $(k+l)$-simplex $\sigma = [v_0, \dots, v_{k+l}]$ as

$$(\phi \smile \psi)(\sigma) := \phi([v_0, \dots, v_k])\psi([v_k, \dots, v_{k+l}]).$$

We can easily see that this product is bilinear, associative and graded commutative (see [24]). Since this cup product produces a map on cochains we need to do some extra work to define a cup product at the level of cohomology. To achieve such goal the following results is required (see Lemma 3.6 in [24]).

Lemma 4.2.1. *Let $\phi \in C^k(K;\mathbb{F})$ and $\psi \in C^l(K;\mathbb{F})$ then*

$$\delta(\phi \smile \psi) = \delta\phi \smile \psi + (-1)^k \phi \smile \delta\psi.$$

We can now define a cup product at cohomology level

$$\smile : H^k(K;R) \times H^l(K;R) \to H^{k+l}(K;R). \tag{4.3}$$

This cup product has many useful properties, among which we have that it is

- associative,
- distributive,
- anti-commutative, in other words $\phi \smile \psi = (-1)^{kl} \psi \smile \phi$,
- and natural with respect to continuous functions, i.e. given a continuous map $f : K \to L$ the induced map $f^* : H^k(L; \mathbb{F}) \to H^k(K; \mathbb{F})$ satisfies $f^*(\phi \smile \psi) = f^*(\phi) \smile f^*(\psi)$.

Example 7. *Consider the same complex K as in Chapter 5 and with the persistent cohomology as computed in Chapter 6 we can obtain the generators from the matrix $V^{\perp}$.*

For this example we will take the persistent classes $[e_9, \infty)$ and $[e_{21}, f_{19})$ which are generated by the cochains

$$\mathbb{1}_{[0,8]} + \mathbb{1}_{[2,8]} + \mathbb{1}_{[2,7]} + \mathbb{1}_{[0,6]} + \mathbb{1}_{[1,7]} + \mathbb{1}_{[1,6]}$$

and

$$\mathbb{1}_{[0,8]} + \mathbb{1}_{[6,8]} + \mathbb{1}_{[5,6]} + \mathbb{1}_{[3,5]} + \mathbb{1}_{[2,3]} + \mathbb{1}_{[0,2]}$$

respectively. Where $\mathbb{1}_{[0,8]}$, for example, represents the identification function that takes value 1 on the simplex $[0, 8]$ and 0 otherwise.

With this notation in mind and the definition at the beginning of this section then one can compute

$$\mathbb{1}_{[v_0,v_1]} \smile \mathbb{1}_{[v_2,v_3]}([v_4, v_5, v_6]) = \mathbb{1}_{[v_0,v_1]}([v_4, v_5])\mathbb{1}_{[v_2,v_3]}([v_5, v_6])$$

is non-zero if and only if $v_1 = v_5 = v_2$, therefore $\mathbb{1}_{[v_0,v_1]} \smile \mathbb{1}_{[v_2,v_3]} = \mathbb{1}_{[v_0,v_1,v_3]}$.

It is the easy to verify that the cup product of the two generator chosen above results in $\mathbb{1}_{[0,6,8]} + \mathbb{1}_{[1,6,8]}$.

To extend the notion of cup product to persistent homology we need to make use of the the property that cup products are natural with respect to continuous functions. With this in mind and using the inclusions in a given filtration $\mathcal{K} = \{K_i\}$ we will define a cup product on $PH^*(\mathcal{K} : \mathbb{F})$ by looking at the following diagram

$$\begin{tikzcd}
PH^k(\mathcal{K};\mathbb{F}) := H^k(K_1;\mathbb{F}) & H^k(K_2;\mathbb{F}) \arrow[l] & \cdots \arrow[l] & H^k(K_m;\mathbb{F}) \arrow[l] \\
\smile & \smile & & \smile \\
PH^l(\mathcal{K};\mathbb{F}) := H^l(K_1;\mathbb{F}) \arrow[d] & H^l(K_2;\mathbb{F}) \arrow[l] \arrow[d] & \cdots \arrow[l] & H^l(K_m;\mathbb{F}) \arrow[l] \arrow[d] \\
PH^{k+l}(\mathcal{K};\mathbb{F}) := H^{k+l}(K_1;\mathbb{F}) & H^{k+l}(K_2;\mathbb{F}) \arrow[l] & \cdots \arrow[l] & H^{k+l}(K_m;\mathbb{F}) \arrow[l]
\end{tikzcd} \tag{4.4}$$

Meaning that given persistent classes in $PH^k(\mathcal{K} : \mathbb{F})$ and $PH^l(\mathcal{K} : \mathbb{F})$ we can compute the cup product entry-wise and obtain a persistent cochain in $PH^{k+l}(\mathcal{K} : \mathbb{F})$. Notice that from this definition we do not know the persistence of this cup product expression. Since the persistence of a class is defined in term of its cohomological birth and death we need to define such notion for a general cochain in $PH^{k+l}(\mathcal{K} : \mathbb{F})$. The *cohomological birth* of a persistent cochain to be the largest point in the filtration where the given cochain is a cocyle, while the *cohomological death* will be the largest point in the filtration where the cochain is a coboundary.

The goal now and our contribution is to develop an algorithm that allow us to compute the cohomological birth and death of a persistent cochain in $PH^{k+l}(\mathcal{K} : \mathbb{F})$.

4.3 Algorithm

The goal of this section is to present an algorithm to compute the persistence of a cochain of the form $\alpha \smile \beta$ for any two classes. To achieve this goal we split this section in to main steps: one dedicated on how to compute cohomological death while the other one deals with cohomologcal birth.

4.3.1 Cohomological death

From this point on we will make heavy use of ideas in [34] and [35] to compute persistent cohomology using a decomposition $(D^{n-1})^{\perp} = R^{\perp}V^{\perp}$ where $R^{\perp}$ is reduced and $V^{\perp}$ is invertible and upper-triangular. Where for any matrix A define $low_A(j)$ as the index of the lowest non-zero entry in the j -th column of A; it is undefined if the column is zero. We say that matrix A is reduced if low_R is injective over its domain of definition.

Let $X_l^{(n)} := X^{(n-1)} \cup \sigma_1 \cup \cdots \cup \sigma_l$ for any $1 \leq l \leq s$. Then the cohomological birth of a cochain $\gamma \in C^n(X;\mathbb{Z}_q)$ coincides with the smallest $1 \leq l \leq s$ such that

1. $i_l^* \circ \delta^{n-1}(x) = i_l^*(\gamma)$ has no solution $x \in C^{n-1}(X;\mathbb{Z}_q)$, and

2. $i_{l-1}^* \circ \delta^{n-1}(x) = i_{l-1}^*(\gamma)$ can be solved.

Where $i_l : X_l^{(n)} \hookrightarrow X^{(n)}$ is the natural inclusion.

First of all notice that when written in the standard basis the composition $i_l^* \circ \delta^{n-1}$ corresponds to the $l \times t$ submatrix of $(D^{(n-1)})^{\perp}$ composed of the last l rows of $(D^{(n-1)})^{\perp}$, namely $(D^{(n-1)})^{\perp}[s+1-l : s, 1 : t]$. therefore, the previously presented condition can be restated as follows:

1. $(D^{(n-1)})^{\perp}[s+1-l : s, 1 : t]x[s+1-l : s] = \gamma[s+1-l : s]$ has no solution in $C^{n-1}(X_l^{(n)};\mathbb{Z}_q)$, and

2. $(D^{(n-1)})^{\perp}[s+1-(l+1) : s, 1 : t]x[s+1-(l+1) : s] = \gamma[s+1-(l+1) : s]$ has no solution.

Furthermore, these conditions can be transformed in terms of $R^{\perp}$. Let $\hat{x} := V^{\perp}x$ and $\hat{\gamma} := V^{\perp}\gamma$, resulting in:

1. $R^{\perp}[s+1-l : s, 1 : t]\hat{x}[s+1-l : s] = \hat{\gamma}[s+1-l : s]$ has no solution in $C^{n-1}(X_l^{(n)};\mathbb{Z}_q)$, and

2. $R^{\perp}[s+1-(l+1) : s, 1 : t]\hat{x}[s+1-(l+1) : s] = \hat{\gamma}[s+1-(l+1) : s]$ has no solution.

```
Precompute: $R^{\perp}$ and $low_{R^{\perp}}$
Initialize: $\hat{x} = (0, \dots, 0) \in \mathbb{Z}_q^t$ and $death = 0$
for $l = s, \dots, 1$ do
    if $l \in low_{R^{\perp}}$ then
        Let $j$ be $low_{R^{\perp}}(j) = l$
        $\hat{x}[j] = \hat{\gamma}[j] - \sum_{k \neq j} \hat{x}[k] R^{\perp}[l, k]$
    else
        if $\sum_k \hat{x}[k] R^{\perp}[l, k] == \hat{\gamma}[j]$ then
            Pass
        else
            $death = l$
            Break
return $death$
```

Algorithm 4.2 Cohomological death of a cochain

Using the fact that the matrix $R^{\perp}$ is reduced we have $low_{R^{\perp}}$ is injective on its domain. Thus we can use the following algorithm to compute the cohomology death of a cochain $\gamma \in C^n(X; \mathbb{Z}_q)$.

Example 8. *Let us consider the filtration $\mathcal{K}$ from Chapter 4. Algorithm 4.2 requires a decomposition $R^{\perp} = D^{\perp} V^{\perp}$ which was already computed in Chapter 6.*

Consider now the generators associated to the persistent classes used in Chapter 7, namely $[e_9, \infty)$ and $[e_{21}, f_{19})$. Now we are interested in computing the cohomological death of the persistent cochain $[e_9, \infty) \smile [e_{21}, f_{19}) = \mathbb{1}_{[0,6,8]} + \mathbb{1}_{[1,6,8]}$ using Algorithm 4.2. Furthermore, since the 2-simplex $[1, 6, 8]$ is not part of the filtered complex in Chapter 4, then we have $[e_9, \infty) \smile [e_{21}, f_{19}) = \mathbb{1}_{[0,6,8]}$.

So we are interested in solving the system of equations using the matrix $V^{\perp}$ and $\mathbb{1}_{[0,6,8]}$, in other words

A careful evaluation of Algorithm 4.2 will return 1, which means that the cohomological death of this cochain happens whenever f_{17} enters the filtration.

$$\begin{array}{r|ccccccccccccccccccccccccc|}
f_{18} & & & & & & 1 & & & & & & & & & & & & & & & & & & & \\
f_{17} & 1 & & & & & & 1 & 1 & & & & & & & & & & & & & & & & & \\
f_{16} & 1 & 1 & & & & & & & & & & & & & & 1 & & & & & & & & & \\
f_{15} & & 1 & 1 & & & & & & & 1 & & & & & & & & & & & & & & & \\
f_{14} & & & 1 & 1 & & & & & & & & & & & & & & & & & & & & & \\
f_{13} & & & & 1 & 1 & & & & & & & & 1 & & & & & & & & & & & & \\
f_{12} & & & & & 1 & & & & & & & & & & & & & & & & & & & & \\
f_{11} & & & & & & & & 1 & & & & & & & & & 1 & & & & & & & & \\
f_{10} & & & & & & & 1 & & 1 & & & & & & & & & & & & & & & & \\
f_{9} & & & & & & & & & 1 & & & & & & & & & & & & 1 & & & & \\
f_{8} & & & & & & & & & & 1 & & 1 & & & & & & & & & & & & & \\
f_{7} & & & & & & & & & & & & 1 & & & & & & & & & & & & & 1 \\
f_{6} & & & & & & & & & & & & & 1 & & & & & & & & & & & & \\
f_{5} & & & & & & & & & & & & & & & & 1 & & & & & & & & & \\
f_{4} & & & & & & & & & & & & & & & & & 1 & & 1 & & & & & & \\
f_{3} & & & & & & & & & & & & & & & & & & & 1 & & & & & & \\
f_{2} & & & & & & & & & & & & & & & & & & & & & 1 & & 1 & & \\
f_{1} & & & & & & & & & & & & & & & & & & & & & & & 1 & & \\
f_{0} & & & & & & & & & & & & & & & & & & & & & & & & & 1
\end{array}\; x = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

4.3.2 Complexity analysis

To study the complexity of our implementation of cohomological death computation we will focus on Algorithm 4.2. First of all it is important to highlight the fact that this algorithm resembles closely the standard backwards substitution algorithm. The main difference resides in the fact that instead of looping the entire matrix we only need to do the corresponding substitutions for the rows that appear in the set of lowest ones of $R^{\perp}$, i.e. the rows that we can encounter in the image of $low_{R^{\perp}}$.

Recall that in general $\delta^n : C^n \to C^{n+1}$, in this context this implies that the inner product has at most as many operations as generators in C^n. Furthermore, since the matrix representation $D^{\perp}$ of δ^n is more often than not not-square is it's safe to assume that it's reduced form $R^{\perp}$ will have at most $\min\left\{\dim(C^n), \dim(C^{n+1})\right\}$ pivots. This fact implies that the outer most loop in Algorithm 4.2 will iterate at most $\dim(C^n)$ times. With this considerations in mind we can see that the number of operation in Algorithm 4.2 is $O(\dim(C^n)^2)$.

To understand this result even better, let us consider a Rips filtration on a set with n points, here we know that the $\dim(C^k)$ grows exponentially on the number of points on 0-skeleton. This in particular means that we can expect $\dim(C^k) \sim n^k$. Furthermore, if we use δ^1 to compute the cohomological death of a cochain in C^2 the complexity of Algorithm 4.2 would become $O(n^2)$.

4.4 Examples

In this section we present different examples of persistent cup product computations. The first two examples we explore in here correspond to sliding window embedding of time series that results in topological spaces homeomorphic to a torus T^2 and to the wedge $S^2 \vee S^1 \vee S^1$. This first example allow us to illustrate the power of the persistent cup product presented in this paper, as it will help us differentiate persistent diagrams coming from samplings of these two spaces.

4.4.1 Torus

For this example we will take advantage of the fact that the sliding window embedding of a quasi-periodic function is homeomorphic to a 2-dimensional torus (see [20]). For this particular example we will use the function in Chapter 4.4.

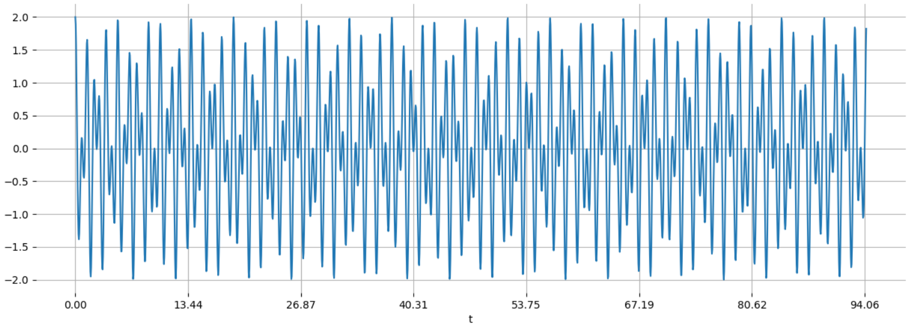

Figure 4.4 Time series $f(t) = \cos(5t) + \cos\left(5\sqrt{3}t\right)$.

Once we have the point cloud obtained from the sliding window embedding we use the Vietoris-Rips filtration of the point cloud to compute its persistent cohomology. Such computation is carried away using the python implementation of Ripser [23] provided as part of the package scikit-tda [36].

The persistent diagram corresponding to this computation is presented in Chapter 4.5.

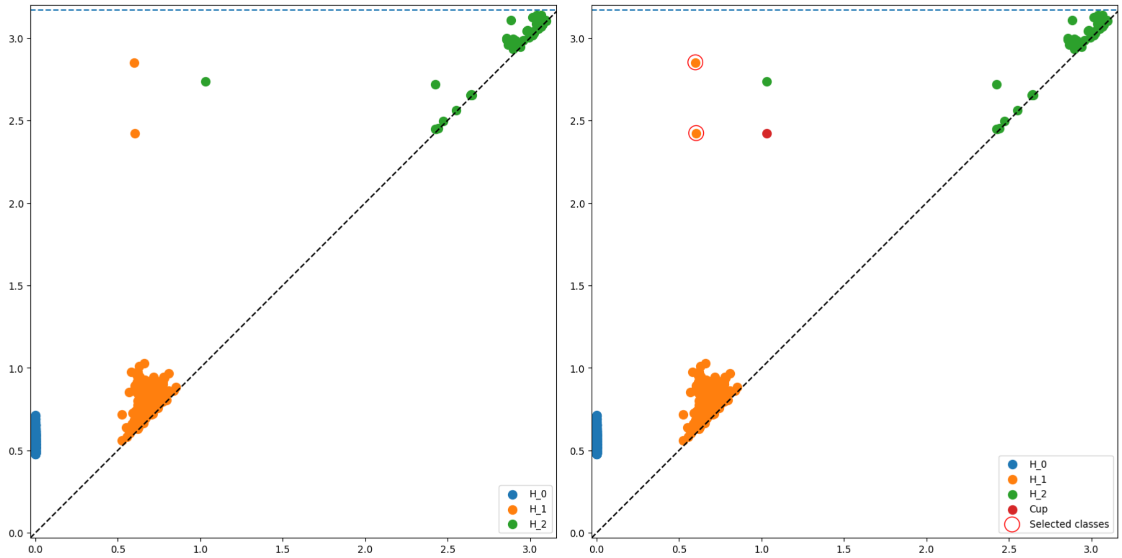

Figure 4.5 Persistence diagram of $PH^*(SW_{d,\tau}(f);\mathbb{Z}_2)$ and persistence of the cup product of the 2 most persistent classes in $PH^1(SW_{d,\tau}(f);\mathbb{Z}_2)$.

4.4.2 Periodic signal with non trivial 2-dimensional persistence.

The main goal of this example is to showcase a periodic signal which sliding window embedding (see [37]) produces a point cloud such that its persistent diagram has 2 high persistent classes in dimension 1 and 1 persistent class in dimension 2. One way in which this goal can be attained is if the point cloud (obtained from the sliding window embedding of a signal) has the same topology as gluing 2 circles to a sphere in different points.

More specifically, to obtain such point cloud we will first construct a time series using the results in [22] that will recover the desired geometric properties. To do so we require to choose an observation function as required by the methodology in [22], in this specific example we will use the Euclidean distance to a single random point in $\mathbb{R}^n$ as our observation function. By applying the results in [22] we obtain the time series in Chapter 4.6

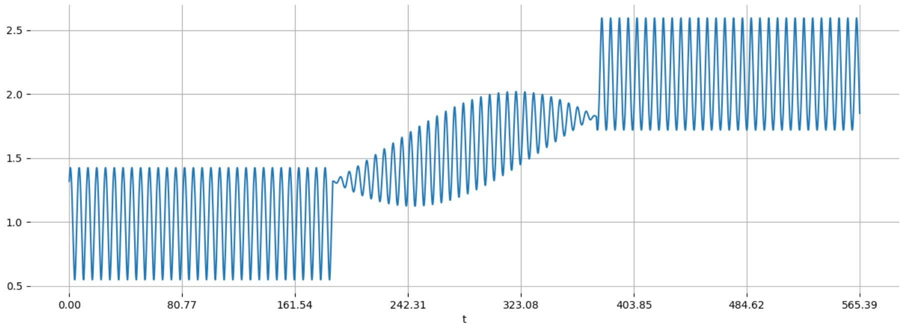

Figure 4.6 Time series.

Once this time series is at hand we can go ahead and compute it's sliding window embedding after fixing an adequate dimension embedding and window size for the embedding to recover the desired geometry. As suggested by [37] we make our dimension $d = 5$ and using this we define the sliding window size as $\omega = \frac{2\pi d}{d+1}$. Once the sliding window point cloud is computed we user Ripser to obtain the persistent diagram in Chapter 4.7.

First let us remark the fact that this persistent diagram has the topological features we required at the beginning of this example, namely it was obtained from a periodic signal (see Chapter 4.6) while presenting 2 clearly persistent classes in PH^1 and 1 persistent class in dimension 2.

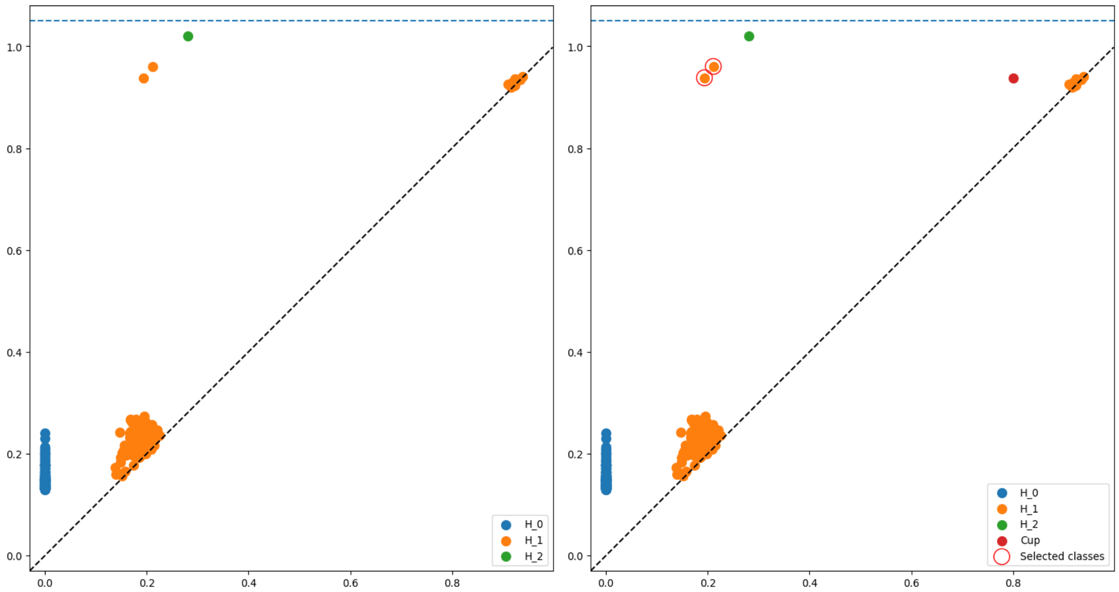

Figure 4.7 Persistence $PH^1(SW_{d,\tau}(f))$.

4.5 Quasi-periodicity detection

Following the work in [21] we define a quasi-periodicity score based on the geometry of the sliding window embedding of a signal. To do so we define dgm_n to be the n-dimensional persistence diagram for the Rips filtration on the sliding window embedding of the signal, therefore we define $mp_i(\mathsf{dgm}_n)$ as the i-th largest difference $b - a$ for any point $(a, b) \in \mathsf{dgm}_n$, or in other words, the persistence of the i-th largest class in dgm_n. Furthermore, we define $\mathrm{pcup}_{i,j}$ to be the persistence of the cup product of the i-th and j-th most persistent cohomology classes in dgm_n.

With this definitions in mind we define a *Quasi-periodicity Cup Score* as

$$CQPS = \sqrt{\frac{mp_2(\mathsf{dgm}_1)\mathrm{pcup}_{1,2}}{3}} \tag{4.5}$$

which is designed to differentiate quasi-periodic signals like the ones in Chapter 4.4.1 from periodic signal with non trivial 2-dimensional cohomology, as in Chapter 4.4.2. This score is based on the second largest 1-dimensional persistence times the persistence of the cup product between the largest and second largest classes in dgm_1.

Throughout this section we will be comparing this score with the Quasi-periodic score defined in [21] as

$$QPS = \sqrt{\frac{mp_2(\mathsf{dgm}_1)mp_1(\mathsf{dgm}_2)}{3}} \tag{4.6}$$

which make use of the most persistent class in the 2-dimensional cohomology of the filtration.

4.5.1 ROC for quasi-periodicity detection

The goal of this example is to provide an evaluation method for the Quasi-periodocity score in Chapter 4.5. To achieve this we produced a data set of 600 time series split as 150 periodic signals, 150 quasi-periodic signals as in Chapter 4.4.1, 150 periodic signal whose sliding window embedding has non trivial 1 and 2-dimensional cohomology as in Chapter 4.4.2 and finally 150 linear signal with Gaussian noise.

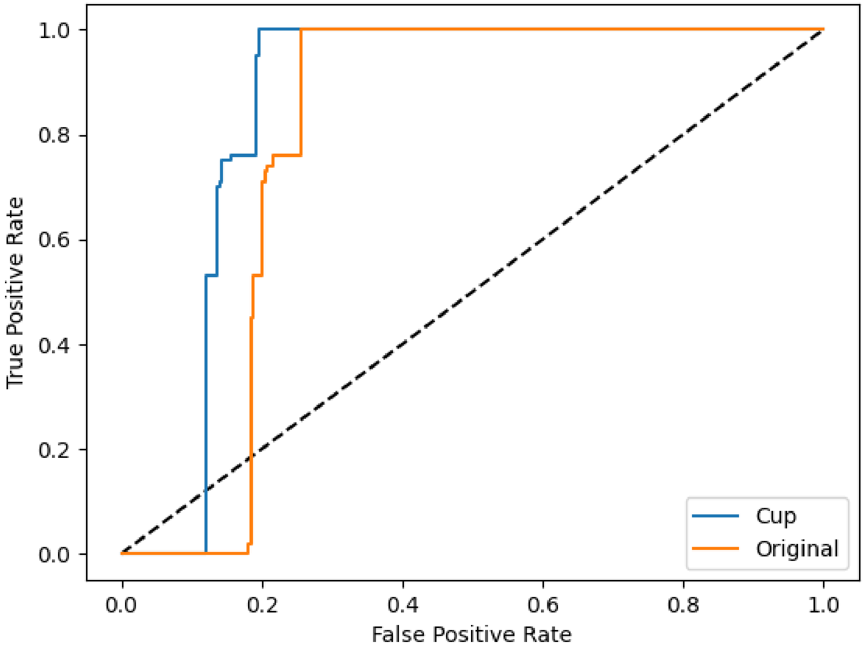

Figure 4.8 ROC curves for the origical Quai-periodicity score in [21] and Cup quasi-periodic score.

After computing the Quasi-periodicity score and Cup Quasi-periodicity score for each signal we solved the binary classification problem of quasi-periodic signal vs. the rest of them. Chapter 4.8 shows the ROC curve for each score where we observe an improvement when using our Cup Quasi-periodic Score over the classification results obtained with the basic QCP in Chapter 4.6. Furthermore, this improvement as measured by the area under the curve (AUC) corresponds to an increase from 0.8052 to 0.8755, when the theoretical maximum is 1.

4.5.2 Vocal folds

Finally we explore an example with real data in the form of Quasi-periodicity detection in video data. More specifically, we consider the data set of high speed vocal folds videos studied in [21] used to separate periodicity from biphonation phenomenons. In particular we apply our cup product pipeline to detect periodicity on the videos from quasi-periodicity. In biological terms the biphonetion phenomenon occurs when the vocal folds oscillation bifurcates between between two different frequencies, this behaviour results in a sliding window embedding that resembles a torus as in Chapter 4.

This data set consists to 7 high speed videos, 2 of which correspond to "normal" periodic vocal fold oscillation, 3 of them represent biphonation and the last 2 portrait irregular motion. Here we replicated the reprocessing and persistent diagram computation carried away in [21]

Video Name	QPS	CQPS
HerbstPeriodic	0.00448	0.0003570
NormalPeriodic	0.0216	0.0000371
APBiphonation	0.298	0.2828325
APBiphonation2	0.116	0.0985476
ClinicalAsymmetry	0.404	0.4025851
MucusPerturbedPeriodic	0.00447	0.0002006
HerbstIrregular	0.0398	0.0345596

Table 4.1 Quasi-periodicit score (PQS) and Quasi-periodicity Cup score (CQPS) across the 7 vocal folds videos.

In Chapter 4.1 we observed that our proposed Quasi-periodicity Cup Score (CQPS) maintains high values on the quasi-periodic signals as we would expect from the results in Chapter 4.5, where we showed for quasi-periodic signal we would expect a 2-dimensional class in cohomolohy with comparable persistence to the one for the cup product of the 2 most persistent classes in dimension 1.

More importantly, when we look at the scores for the periodic signals the CQPS score reduces its value compared to the original QPS score by an amount between 1 and 3 orders of magnitude. This increased separation between the scores for periodic and quasi-periodic signals makes the separation between this phenomenon easier to identify. This behaviour of our CQPS score is reinforced by the results in Chapter 4.5.1 where we see an improvement ion the classification power of CQPS when compared to QPS.

4.6 Discussion

The main contribution of this paper is in Chapter 4.3 where we present an algorithm that allow us to compute the persistence of persistent chains in general, allowing us to reach the two main goals we set for this work:

- to compute the persistence of cup product expressions and enrich persistent diagrams with such information, while providing scores to enhance quasi-periodicity detection, and
- to reduce the computational burden of obtaining said scores, by allowing the computation of persistence for 2-chains while only computing generators in 1-dimensional persistent cohomology.

We show the performance of this methodology to separate periodic from quasi-periodic signals in a synthetic data set to be comparable to that of the scores proposed in [21] as presented in Chapter 4.8. As well as in a real life data set in Chapter 4.5.2, for which we can clearly see a stronger separation between periodic and quasi-periodic signals when compared to the results in the original work [21], while displaying comparable results for the edge scenarios of signal that are in general hard to identify in Chapter 4.1 like the "Clinical Asymetry" cases.

It is important to point out that the computations presented in this work do not reflect the persistent cup product as a submodule of the persistent cohomology. We are computing the persistence of a specific persistent cochain, that arises from the cup product at cochain level of generators of persistent classes in the persistent cohomology. Nevertheless, investigating the possibility to extended some of these tools to aid in the computation of the cup product as a persistent submodule is an interesting research avenue for future projects.

Furthermore, the cup product persistence algorithm presented here can achieve higher theoretical performance compared the version presented in [21] that depends on computing 2-dimensional persistent cohomology as expressed in Chapter 4.3. Our specific implementation was made in Python 3.7 and therefore carries over some of the inefficiencies inherited by the language. Further work will focus on refining our implementation to take further advantage of the computational benefits benefits one can achieve with this algorithm as well as exploring the performance of this classification methodology on other quasi-periodic phenomena.

CHAPTER 5

LENS COORDINATES AND DATA COORDINATIZATION

5.1 Preliminaries

5.1.1 Persistent Cohomology

A family $\mathcal{K} = \{K_\alpha\}_{\alpha \in \mathbb{R}}$ of simplicial complexes is called a filtration if $K_\alpha \subset K_{\alpha'}$ whenever $\alpha \leq \alpha'$. If $\mathbb{F}$ is a field and $i \geq 0$ is an integer, then the direct sum $PH^i(\mathcal{K}; \mathbb{F}) := \bigoplus\limits_{\alpha} H^i(K_\alpha; \mathbb{F})$ of cohomology groups is called the i-th dimensional **$\mathbb{F}$-persistent cohomology** of $\mathcal{K}$. A theorem of Crawley-Boevey [38] contends that if $H^i(K_\alpha; \mathbb{F})$ is finite dimensional for each α, then the isomorphism type of $PH^i(\mathcal{K}; \mathbb{F})$—as a persistence module—is uniquely determined by a multiset (i.e., a set whose elements may appear with repetitions)

$$\mathsf{dgm} \subset \{(\alpha, \alpha') \in [-\infty, \infty]^2 : \alpha \leq \alpha'\}$$

called the **persistence diagram** of $PH^i(\mathcal{K}; \mathbb{F})$. Pairs (α, α') with large persistence $\alpha' - \alpha$, are indicative of stable topological features throughout the filtration $\mathcal{K}$.

Persistent cohomology is used in TDA to quantify the topology underlying a data set. There are two widely used filtrations associated to a subset X of a metric space (M, d), the **Rips filtration** $\mathcal{R}(X) = \{R_\alpha(X)\}_\alpha$ and the **Čech filtration** $\check{\mathcal{C}}(X) = \{\check{C}_\alpha(X)\}_\alpha$. Specifically, $R_\alpha(X)$ is the set of nonempty finite subsets of X with diameter less than α, and $\check{C}_\alpha(X)$ is the nerve of the collection $\mathcal{B}_\alpha$ of open balls $B_\alpha(x) \subset M$ of radius α, centered at $x \in X$. In other words, $\check{C}_\alpha(X) = \mathcal{N}(\mathcal{B}_\alpha)$. Generally $\mathcal{R}(X)$ is more easily computable, but $\check{\mathcal{C}}(X)$ has better theoretical properties (e.g., the Nerve theorem [39, 4G.3]). Their relative weaknesses are ameliorated by noticing that

$$R_\alpha(X) \subset \mathcal{N}(\mathcal{B}_\alpha) \subset R_{2\alpha}(X)$$

for all α, and using both filtrations in analyses: Rips for computations, and Čech for theoretical inference.

5.1.2 Lens Spaces

Let $q \in \mathbb{N}$ and let $\zeta_q \in \mathbb{C}$ be a primary q-th root of unity. Fix $n \in \mathbb{N}$ and let $q_1, \ldots, q_n \in \mathbb{N}$ be relatively prime to q. We define the **Lens space** $L_q^n(q_1, \ldots, q_n)$ as the quotient of $S^{2n-1} \subset \mathbb{C}^n$ by the $\mathbb{Z}_q$ right action

$$[z_1, \ldots, z_n] \cdot g := \left[z_1 \zeta_q^{q_1 g}, \ldots, z_n \zeta_q^{q_n g}\right]$$

with simplified notation $L_q^n := L_q^n(1, \ldots, 1)$. Notice that when $q = 2$ and $q_1 = \cdots = q_n = 1$, then the right action described above is the antipodal map of S^{2n-1}, and therefore $L_2^n = \mathbb{R}\mathbf{P}^{2n-1}$. Similarly, the infinite Lens space $L_q^\infty = L_q^\infty(1, 1, \ldots)$ is defined as the quotient of the infinite unit sphere $S^\infty \subset \mathbb{C}^\infty$, by the action of $\mathbb{Z}_q$ induced by scalar-vector multiplication by powers of ζ_q.

5.1.2.1 A Fundamental domain for $L_q^2(1, p)$

In what follows we describe a convenient model for both $L_q^2(1, p)$ and a fundamental domain thereof. This model will allow us to provide visualizations in Lens spaces towards the end of the paper. Let D^3 be the set of points $\mathbf{x} \in \mathbb{R}^3$ with $\|\mathbf{x}\| \leq 1$, and let D_+ (D_-) be the upper (lower) hemisphere of ∂D^3, including the equator. Let $r_{p/q} : D_+ \longrightarrow D_+$ be counterclockwise rotation by $2\pi p/q$ radians around the z-axis, and let $\rho : D_+ \longrightarrow D_-$ be the reflection $\rho(x, y, z) = (x, y, -z)$. Then, $L_q^2(1, p)$ is homeomorphic to $D^3/\sim$, where $\mathbf{x} \sim \mathbf{y}$ if and only if $\mathbf{x} \in D_+$ and $\mathbf{y} = \rho \circ r_{p/q}(\mathbf{x})$.

5.1.3 Principal Bundles

Let B be a topological space with base point $b_0 \in B$. One of the most transparent methods for producing an explicit bijection between $\check{H}^1(B; \mathscr{C}_{\mathbb{Z}_q})$ and $[B, L_q^\infty]$ is via the theory of

Principal bundles. We present a terse introduction here, but direct the interested reader to [40] for details. A continuous map $\pi : P \longrightarrow B$ is said to be a **fiber bundle** with fiber $F = \pi^{-1}(b_0)$ and total space P, if π is surjective, and every $b \in B$ has an open neighborhood $U \subset B$ as well as a homeomorphism $\rho_U : U \times F \longrightarrow \pi^{-1}(U)$, so that $\pi \circ \rho_U(x, e) = x$ for every $(x, e) \in U \times F$.

Let $(G, +)$ be an abelian topological group. A fiber bundle $\pi : P \longrightarrow B$ is said to be a **principal G-bundle** over B, if P comes equipped with a free right G-action $P \times G \ni (e, g) \mapsto e \cdot g \in P$ which is transitive in $\pi^{-1}(b)$ for every $b \in B$. Moreover, two principal G-bundles $\pi : P \longrightarrow B$ and $\pi' : P' \longrightarrow B$ are isomorphic, if there exits a homeomorphism $\Phi : P \longrightarrow P'$, with $\pi' \circ \Phi = \pi$ and so that $\Phi(e \cdot g) = \Phi(e) \cdot g$ for all $(e, g) \in P \times G$. Given an open cover $\mathcal{U} = \{U_j\}_{j\in J}$ of B, a **Čech cocycle**

$$\eta = \{\eta_{jk}\} \in \check{Z}^1(\mathcal{U}; \mathscr{C}_G)$$

is a collection of continuous maps $\eta_{jk} : U_j \cap U_k \longrightarrow G$ so that $\eta_{jk}(b) + \eta_{kl}(b) = \eta_{jl}(b)$ for every $b \in U_j \cap U_k \cap U_l$. Given such a cocycle, one can construct a principal G-bundle over B with total space

$$P_\eta = \left(\bigcup_{j\in J} U_j \times \{j\} \times G \right) / \sim$$

where $(b, j, g) \sim (b, k, g + \eta_{jk}(b))$ for every $b \in U_j \cap U_k$, and $\pi : P_\eta \longrightarrow B$ sends the class of (b, j, g) to $b \in B$.

Theorem 5.1.1. *If $\mathsf{Prin}_G(B)$ denotes the set of isomorphism classes of principal G-bundles over B, then*

$$\begin{array}{ccc} \check{H}^1(B; \mathscr{C}_G) & \longrightarrow & \mathsf{Prin}_G(B) \\ [\eta] & \mapsto & [P_\eta] \end{array}$$

is a bijection.

Proof. See 2.4 and 2.5 in [18] □

Now, let us describe the relation between principal G-bundles over B, and maps from B to the classifying space BG. Indeed, let $\jmath : EG \longrightarrow BG = EG/G$ be the quotient map. Given $h : B \longrightarrow BG$ continuous, the pullback h^*EG is the principal G-bundle over B with total space $\{(b, e) \in B \times EG : h(b) = \jmath(e)\}$, and projection map $(b, e) \mapsto b$. Moreover,

Theorem 5.1.2. *Let $[B, BG]$ denote the set of homotopy class of maps from B to the classifying space BG. Then, the function*

$$\begin{array}{ccc} [B, BG] & \longrightarrow & \mathsf{Prin}_G(B) \\ [h] & \mapsto & [h^*EG] \end{array}$$

is a bijection.

Proof. See [40], Chapter 4: Theorems 12.2 and 12.4. □

In summary, given a principal G-bundle $\pi : P \longrightarrow B$, or its corresponding Čech cocycle η, there exists a continuous map $h : B \longrightarrow BG$ so that h^*EG is isomorphic to (π, P), and the choice of h is unique up to homotopy. Any such choice is called a classifying map for $\pi : P \longrightarrow B$.

5.2 Main Theorem: Explicit Classifying Maps for L_q^∞

The goal of this section is to show how one can go from a singular cocycle $\eta \in Z^1(\mathcal{N}(\mathcal{U}); \mathbb{Z}_q)$ to an explicit map $f : \bigcup \mathcal{U} \longrightarrow L_q^\infty$. Let $J = \{1, \ldots, n\}$, let $\mathcal{U} = \{U_j\}_{j \in J}$ be an open cover for B, and let $\{\varphi_j\}_{j \in J}$ be a partition of unity dominated by $\mathcal{U}$. If $\eta = Z^1(\mathcal{N}(\mathcal{U}); \mathbb{Z}_q)$ and ζ_q is a primitive q-th root of unity, let $f_j : U_j \times \{j\} \times \mathbb{Z}_q \longrightarrow S^{2n-1} \subset \mathbb{C}^n$ be

$$f_j(b, j, g) = \left[\sqrt{\varphi_1(b)}\zeta_q^{(g+\eta_{j1})}, \ldots, \sqrt{\varphi_n(b)}\zeta_q^{(g+\eta_{jn})}\right]$$

If $b \in U_j \cap U_k$, then $f_j(b, j, g) = f_k(b, k, g + \eta_{jk})$ and we get an induced map $\Phi : P_\eta \longrightarrow S^{2n-1} \subset S^\infty$ taking the class of (b, j, g) in the quotient P_η to $f_j(b, j, g)$.

Proposition 5.2.1. Φ *is well defined and $\mathbb{Z}_q$-equivariant.*

Proof. Take $[b,j,g] \in P_\eta$ and consider a different representative of the class. Namely, an element $(b,k,g+\eta_{jk})$ such that $b \in U_j \cap U_k$. By definition of Φ, we have $\Phi([b,j,g]) = f_j(b,j,g)$ and $\Phi([b,k,g+\eta_{jk}]) = f_k(b,k,g+\eta_{jk})$. And since $f_j(b,j,g) = f_k(b,k,g+\eta_{jk})$, we have that

$$\Phi([b,j,g]) = \Phi([b,k,g+\eta_{jk}]),$$

which shows that Φ is well defined.

To see that Φ is $\mathbb{Z}_q$-equivariant, take $m \in \mathbb{Z}_q$ for any $m = 0,\dots,q-1$ and compute

$$\begin{aligned}
&\Phi([b,j,g]) \cdot m \\
&= \left[\sqrt{\varphi_1(b)}\zeta_q^{(g+m+\eta_{j1})}, \dots, \sqrt{\varphi_n(b)}\zeta_q^{(g+m+\eta_{jn})}\right] \\
&= f_j(b,j,g+m) = \Phi([b,j,g+m]) \\
&= \Phi([b,j,g] \cdot m).
\end{aligned}$$

□

Let $p : S^{2n-1} \longrightarrow L_q^n$ be the quiotient map. Since $\Phi : P_\eta \longrightarrow S^{2n-1} \subset S^\infty$ is $\mathbb{Z}_q$-equivariant, it induces a map $f : B \longrightarrow L_q^n \subset L_q^\infty$ such that $p \circ \Phi = f \circ \pi$. By construction of $\pi : P_\eta \longrightarrow B$, $f(\pi([b,j,g])) = f(b)$ for any $g \in \mathbb{Z}_q$. In particular for $0 \in \mathbb{Z}_q$

$$U_j \ni b \ , \ \ f(b) = \left[\sqrt{\varphi_1(b)}\zeta_q^{\eta_{j1}} : \dots : \sqrt{\varphi_n(b)}\zeta_q^{\eta_{jn}}\right] \tag{5.1}$$

Remark 5.2.2. *The notation $[a_1 : \dots : a_n]$ corresponds to homogeneous coordinates in the quotient $S^{2n-1}/\mathbb{Z}_q$. In other words, $[a_1 : \dots : a_n] = \{[a_1 \cdot \alpha, \dots, a_n \cdot \alpha] \in S^{2n-1} : \alpha \in \mathbb{Z}_q\}$.*

Theorem 5.2.3. *The map f classifies the $\mathbb{Z}_q$-principal bundle P_η associated to the cocycle $\eta \in Z^1(\mathcal{N}(\mathcal{U});\mathbb{Z}_q)$.*

Proof. First we need to see that f is well defined. Let $b \in U_j \cap U_k$, therefore

$$\begin{aligned}
p(\Phi([b,j,0])) &= \left[\sqrt{\varphi_1(b)}\zeta_q^{\eta_{j1}} : \dots : \sqrt{\varphi_n(b)}\zeta_q^{\eta_{jn}}\right] \\
&= p(\Phi([b,k,0)).
\end{aligned}$$

This shows that $f(b)$ is independent of the open set containing b.

Hence $(\Phi, f) : (P_\eta, \pi, B) \to (S^{2n-1}, \pi, L_q^n)$ is a morphism of principal $\mathbb{Z}_q$-bundles, and by [[40], Chapter 4: Theorem 4.2] we conclude that P_η and $f^*(S^{2n-1})$ are isomorphic principal $\mathbb{Z}_q$-bundles over B. □

5.3 Lens coordinates for data

Let (M, d) be a metric space and let $L \subset M$ be a finite subset. We will use the following notation from now on: $B_\epsilon(l) = \{y \in M : d(y, l) < \epsilon\}$, $\mathcal{B}_\epsilon = \{B_\epsilon(l)\}_{l \in L}$, and $L^\epsilon = \bigcup \mathcal{B}_\epsilon$. Given a data set $X \subset M$, our goal will be to choose $L \subset X$, a suitable ϵ such that $X \subset L^\epsilon$, and a cocycle $\eta \in Z^1(\mathcal{N}(\mathcal{B}_\epsilon); \mathbb{Z}_q)$. Chapter 5.1 yields a map $f : L^\epsilon \to L_q^\infty$ defined for every point in X, but constructed from a much smaller subset of landmarks. Next we describe the details of this construction.

5.3.1 Landmark selection

We select the landmark set $L \subset X$ either at random or through `maxmin` sampling. The latter proceeds inductively as follows: Fix $n \leq |X|$, and let $l_1 \in X$ be chosen at random. Given $l_1, \ldots, l_j \in X$ for $j < n$, we let $l_{j+1} = \underset{x \in X}{\operatorname{argmax}} \min\{d(x, l_1), \ldots, d(x, l_j)\}$.

5.3.2 A Partition of Unity subordinated to $\mathcal{B}_\epsilon$

Defining f requires a partition of unity subordinated to $\mathcal{B}_\epsilon$. Since $\mathcal{B}_\epsilon$ is an open cover composed of metric balls, then we can provide an explicit construction. Indeed, for $r \in \mathbb{R}$ let $|r|_+ := \max\{r, 0\}$, then

$$\varphi_l(x) := |\epsilon - d(x, l)|_+ \Big/ \sum_{l' \in L} |\epsilon - d(x, l')|_+ \qquad (5.2)$$

is a partition of unity subordinated to $\mathcal{B}_\epsilon$.

5.3.3 From Rips to Čech to Rips

As we alluded to in the introduction, a persistent cohomology calculation is an appropriate vehicle to select a scale ϵ and a candidate cocycle η. That said, determining $\eta \in Z^1(\mathcal{N}(\mathcal{B}_\epsilon), \mathbb{Z}_q)$ would require computing $\mathcal{N}(\mathcal{B}_\epsilon)$ for all ϵ, which in general is an expensive procedure. Instead we will use the homomorphisms

$$\begin{tikzcd} H^1(\mathcal{R}_{2\epsilon}(L)) \arrow[r, "i^*"] \arrow[rr, bend right=10, "\iota"'] & H^1(\mathcal{N}(\mathcal{B}_\epsilon)) \arrow[r] & H^1(\mathcal{R}_\epsilon(L)) \end{tikzcd}$$

induced by the appropriate inclusions. Indeed, let $\tilde{\eta} \in Z^1(\mathcal{R}_{2\epsilon}(L); \mathbb{Z}_q)$ be such that $[\tilde{\eta}] \notin \ker(\iota)$. This is where we use the persistent cohomology of $\mathcal{R}(L)$. Since the previous diagram commutes, then $[\tilde{\eta}] \notin \ker(i^*)$, so $i^*([\tilde{\eta}]) \neq 0$ in $H^1(\mathcal{N}(\mathcal{B}_\epsilon); \mathbb{Z}_q)$. We will let $[\eta] = i^*([\tilde{\eta}])$ be the class that we use in Chapter 5.2.3. However,

Proposition 5.3.1. *If $b \in \mathcal{B}_\epsilon(l_j)$ and $1 \leq k \leq n$, then*

$$\sqrt{\varphi_k(b)}\zeta_q^{\eta_{jk}} = \sqrt{\varphi_k(b)}\zeta_q^{\tilde{\eta}_{jk}}.$$

That is, we can compute Lens coordinates using only the Rips filtration on the landmark set.

Proof. First of all, $\mathcal{R}_{2\epsilon}(L)^{(0)} = \mathcal{N}(\mathcal{B}_\epsilon)^{(0)} = L$. If $b \notin B_\epsilon(l_k)$, then $\varphi_k(b) = 0$ and therefore the equality holds. If on the other hand $b \in B_\epsilon(l_k) \cap B_\epsilon(l_j)$, then $\{j, k\} \in \mathcal{N}(\mathcal{B}_\epsilon)^{(1)} \subset \mathcal{R}_{2\epsilon}(L)^{(1)}$. In which case, by definition of i^*, we have $\tilde{\eta}_{jk} = \eta_{jk}$. □

5.4 Dimensionality Reduction in L_q^n via Principal Lens Components

Chapter 5.1 gives an explicit formula for the classifying map $f : B \longrightarrow L_q^n$. By construction, the dimension of L_q^n depends on the number n of landmarks selected, which in general can be large. The main goal of this section is to construct a dimensionality reduction procedure in L_q^n to address this shortcoming. To this end, we define the distance $d_L : L_q^n \times L_q^n \longrightarrow [0, \infty)$ as

$$d_L([x], [y]) := d_H(x \cdot \mathbb{Z}_q\, , \, y \cdot \mathbb{Z}_q) \tag{5.3}$$

where d_H id the Hausdorff distance for subsets of S^{2n-1}.

Proposition 5.4.1. *Let $[x], [y] \in L_q^n$, then*

$$d_L([x],[y]) = d(x, \mathbb{Z}_q y) = \min_{g\in\mathbb{Z}_q} d(x, gy).$$

Proof. For $x, y \in \mathbb{C}^n$ let $\langle x, y\rangle_{\mathbb{R}} := \mathsf{real}(\langle x, y\rangle_{\mathbb{C}})$. By definition of Hausdorff distance, we have that

$$\begin{aligned} d_L([x],[y]) = \max \Bigg\{ &\max_{g\in\mathbb{Z}_q} \min_{h\in\mathbb{Z}_q} \arccos(\langle x\cdot g, y\cdot h\rangle_{\mathbb{R}})\ , \\ &\max_{h\in\mathbb{Z}_q} \min_{g\in\mathbb{Z}_q} \arccos(\langle x\cdot g, y\cdot h\rangle_{\mathbb{R}}) \Bigg\}. \end{aligned}$$

Notice that

$$\begin{aligned} \langle x\cdot g, y\cdot h\rangle_{\mathbb{R}} &= \mathsf{real}\left(\left\langle \zeta_q^g x, \zeta_q^h y\right\rangle_{\mathbb{C}}\right) \\ &= \mathsf{real}\left(\left\langle x, \zeta_q^{(h-g)} y\right\rangle_{\mathbb{C}}\right) \\ &= \langle x, y\cdot(h-g)\rangle_{\mathbb{R}} \end{aligned}$$

And since $\mathbb{Z}_q$ is Abelian, then

$$\begin{aligned} &\max_{h\in\mathbb{Z}_q} \min_{g\in\mathbb{Z}_q} \arccos(\langle x\cdot g, y\cdot h\rangle_{\mathbb{R}}) \\ &\qquad = \max_{h\in\mathbb{Z}_q} \min_{g\in\mathbb{Z}_q} \arccos(\langle x\cdot(g-h), y\rangle_{\mathbb{R}}) \\ &\qquad = \max_{h\in\mathbb{Z}_q} \min_{g\in\mathbb{Z}_q} \arccos(\langle x\cdot(-h), y\cdot(-g)\rangle_{\mathbb{R}}) \\ &\qquad = \max_{h'\in\mathbb{Z}_q} \min_{g'\in\mathbb{Z}_q} \arccos\big(\langle x\cdot h', y\cdot g'\rangle_{\mathbb{R}}\big). \end{aligned}$$

Thus

$$d_L([x],[y]) = \max_{g\in\mathbb{Z}_q} \min_{h\in\mathbb{Z}_q} \arccos(\langle x\cdot g, y\cdot h\rangle_{\mathbb{R}}).$$

Furthermore

$$d_L([x],[y]) = \max_{g\in\mathbb{Z}_q} d(x\cdot g, y\cdot\mathbb{Z}_q) = \max_{g\in\mathbb{Z}_q} d(x, y\cdot(-g)\mathbb{Z}_q).$$

Since $y \cdot \big((-g)\mathbb{Z}_q\big) = y \cdot \mathbb{Z}_q$ for any $g \in \mathbb{Z}_q$, we obtain $d_L([x],[y]) = \max_{g\in\mathbb{Z}_q} d(x, y\cdot\mathbb{Z}_q) = d(x, y\cdot\mathbb{Z}_q) = \min_{h\in\mathbb{Z}_q} d(x, y\cdot h)$. □

We will now describe a notion of **projection in** L_q^n onto lower-dimensional Lens spaces. Indeed, let $u \in S^{2n-1}$. Since $\zeta_q^k w \in \mathsf{span}_{\mathbb{C}}(u)^\perp$ for any $k \in \mathbb{Z}_q$ and $w \in \mathsf{span}_{\mathbb{C}}(u)^\perp$, then

$$L_q^{n-1}(u) := (\mathsf{span}_{\mathbb{C}}(u)^\perp \cap S^{2n-1})/\mathbb{Z}_q$$

is isometric to L_q^{n-1}. Let $P_u^\perp(v) = v - \langle v, u\rangle_{\mathbb{C}} u$ for $v \in \mathbb{C}^n$, and if $v \notin \mathsf{span}_{\mathbb{C}}(u)$, then we let

$$\mathcal{P}_u([v]) := \Big[P_u^\perp(v)/\|P_u^\perp(v)\|\Big] \in L_q^{n-1}(u)$$

It readily follows that $\mathcal{P}_u$ is well defined, and that

Lemma 5.4.2. *For $u \in S^{2n-1}$ and $v \notin \mathsf{span}_{\mathbb{C}}(u)$, we have*

$$d_L([v], \mathcal{P}_u([v])) = d\Big(v\ ,\ P_u^\perp(v)/\|P_u^\perp(v)\|\Big)$$

where d is the distance on S^{2n-1}. Furthermore, $\mathcal{P}_u([v])$ is the point in $L_q^{n-1}(u)$ closest to $[v]$ with respect to d_L.

Proof. From Chapter 5.4.1 we know that

$$\begin{aligned} d_L([v], P_u^\perp([v])) &= \min_{g\in\mathbb{Z}_q} d(v, P_u^\perp([v])\cdot g) \\ &= \min_{g\in\mathbb{Z}_q} d\left(v, \frac{P_u^\perp(v)}{\|P_u^\perp(v)\|}\cdot g\right). \end{aligned}$$

Let $g^* := \underset{g\in\mathbb{Z}_q}{\operatorname{argmin}}\ d\left(v, \frac{P_u^\perp(v)}{\|P_u^\perp(v)\|}\cdot g\right)$, so we have

$$d_L([v], P_u^\perp([v])) = \arccos\left(\left\langle v, \frac{P_u^\perp(v)}{\|P_u^\perp(v)\|}\cdot g^*\right\rangle_{\mathbb{R}}\right).$$

Notice that the argument of the arccos can be simplified as follows

$$\begin{aligned}\left\langle v, \frac{P_u^{\perp}(v)}{\|P_u^{\perp}(v)\|} \cdot g^* \right\rangle_{\mathbb{R}} &= \left\langle \langle v, u\rangle_{\mathbb{C}} u + P_u^{\perp}(v), \frac{P_u^{\perp}(v)}{\|P_u^{\perp}(v)\|} \cdot g^* \right\rangle_{\mathbb{R}} \\ &= \left\langle \langle v, u\rangle_{\mathbb{C}} u, \frac{P_u^{\perp}(v)}{\|P_u^{\perp}(v)\|} \cdot g^* \right\rangle_{\mathbb{R}} \\ &\quad + \left\langle P_u^{\perp}(v), \frac{P_u^{\perp}(v)}{\|P_u^{\perp}(v)\|} \cdot g^* \right\rangle_{\mathbb{R}} .\end{aligned}$$

since u and $P_u^{\perp}(v)$ are orthogonal in $\mathbb{C}^n$ then they are also orthogonal in $\mathbb{R}^{2n}$, making the then the firs summand on the right hand side equal to zero. Additionally since arccos as a real valued function is monotonically decreasing we have

$$g^* = \underset{g \in \mathbb{Z}_q}{\operatorname{argmax}} \ \frac{1}{\|P_u^{\perp}(v)\|} \left\langle P_u^{\perp}(v), P_u^{\perp}(v) \cdot g \right\rangle_{\mathbb{R}} .$$

Using the fact that the action of $\mathbb{Z}_q$ is an isometry (and therefore an operator of norm one) as well as the Cauchy-Schwartz inequality we obtain

$$\begin{aligned}\frac{\left\langle P_u^{\perp}(v), P_u^{\perp}(v) \cdot g \right\rangle_{\mathbb{R}}}{\|P_u^{\perp}(v)\|} &\leq \left| \frac{1}{\|P_u^{\perp}(v)\|} \left\langle P_u^{\perp}(v), P_u^{\perp}(v) \cdot g \right\rangle_{\mathbb{R}} \right| \\ &\leq \frac{1}{\|P_u^{\perp}(v)\|} \|P_u^{\perp}(v)\| \|P_u^{\perp}(v) \cdot g\| \\ &= \|P_u^{\perp}(v) \cdot g\| = \|P_u^{\perp}(v)\|.\end{aligned}$$

And the equality holds whenever $g = e \in \mathbb{Z}_q$, so we must have $g^* = e$.

Let $[w] \in L_q^{n-1}(u)$, so $w \in \mathsf{span}_{\mathbb{C}}^{\perp}(u)$ which implies that for any $h \in \mathbb{Z}_q$

$$\langle u, w \cdot h \rangle_{\mathbb{C}} = \sum_k u_k (\overline{\zeta_q^h w_k}) = \zeta_q^{-h} \sum_k u_k \overline{w_k} = \zeta_q^{-h} \langle u, w \rangle = 0.$$

In other words $w \cdot h \in \mathsf{span}_{\mathbb{C}}^{\perp}(u)$ for any $h \in \mathbb{Z}_q$.

Thus by the Cauchy-Schwartz inequality

$$\begin{aligned}\langle v, w \cdot h \rangle_{\mathbb{R}} &= \langle \langle v, u \rangle_{\mathbb{C}} u + P_u^{\perp}(v), w \cdot h \rangle_{\mathbb{R}} = \langle P_u^{\perp}(v), w \cdot h \rangle_{\mathbb{R}} \\ &\leq |\langle P_u^{\perp}(v), w \cdot h \rangle_{\mathbb{R}}| \leq \|P_u^{\perp}(v)\| \|w \cdot h\| \\ &= \|P_u^{\perp}(v)\| \|w\| = \|P_u^{\perp}(v)\|,\end{aligned}$$

since the action of $\mathbb{Z}_q$ is an isometry and $w \in S^{2n-1}$.

Finally since arccos is decreasing

$$d_L([v], P_u^{\perp}([v])) = \arccos\Big(\|P_u^{\perp}(v)\|\Big) \leq \arccos(\langle v, w \cdot h\rangle_{\mathbb{R}})$$

for all $h \in \mathbb{Z}_q$, thus $d_L([v], P_u^{\perp}([v])) \leq d_L([v],[w])$. □

This last result suggests that a PCA-like approach is possible for dimensionality reduction in Lens spaces. Specifically, for $Y = \{[y_1], \ldots, [y_N]\} \subset L_q^n$, the goal is to find $u \in S^{2n-1}$ such that $L_q^{n-1}(u)$ is the best $(n-1)$-Lens space approximation to Y, then project Y onto $L_q^{n-1}(u)$ using $\mathcal{P}_u$, and repeat the process iteratively reducing the dimension by 1 each time. At each stage, the appropriate constrained optimization problem is

$$\begin{aligned} u^* &= \underset{u \in \mathbb{C}^n, \|u\|=1}{\operatorname{argmin}} \sum_{j=1}^{N} d_L([y_j], \mathcal{P}_u([y_i]))^2 \\ &= \underset{u \in \mathbb{C}^n, \|u\|=1}{\operatorname{argmin}} \sum_{j=1}^{N} \left(\frac{\pi}{2} - \arccos(|\langle y_i, u\rangle|)\right)^2 \end{aligned}$$

which can be linearized using the Taylor series expansion of $\arccos(\theta)$ around 0. Indeed, $|\frac{\pi}{2} - \arccos(\theta)| \approx |\theta|$ to third order, and thus

$$u^* \approx \underset{u \in \mathbb{C}^n, \|u\|=1}{\operatorname{argmin}} \sum_{j=1}^{N} |\langle y_i, u\rangle|^2.$$

This approximation is a linear least square problem whose solution is given by the eigenvector corresponding to the smallest eigenvalue of the covariance matrix

$$\mathsf{Cov}\,(y_1, \ldots, y_N) = \begin{bmatrix} | & & | \\ y_1 & \cdots & y_N \\ | & & | \end{bmatrix} \begin{bmatrix} - & \overline{y}_1 & - \\ & \vdots & \\ - & \overline{y}_N & - \end{bmatrix}.$$

Moreover, for any $\alpha_1, \ldots, \alpha_N \in S^1 \subset \mathbb{C}$ we have that

$$\mathsf{Cov}\,(\alpha_1 y_1, \ldots, \alpha_N y_N) = \mathsf{Cov}\,(y_1, \ldots, y_N),$$

so $\mathsf{Cov}(Y)$ is well defined for $Y \subset L_q^n$.

5.4.1 Inductive construction of LPCA

Let $v_n = \mathsf{LastLensComp}(Y)$ be the eigenvector of $\mathsf{Cov}(Y)$ corresponding to the smallest eigenvalue. Assume that we have constructed $v_{k+1}, \ldots, v_n \in S^{2n-1}$ for $1 < k < n$, and let $\{u_1, \ldots, u_k\}$ be an orthonormal basis for $\mathsf{span}_{\mathbb{C}}(v_{k+1}, \ldots, v_n)^{\perp}$. Let $U_k \in \mathbb{C}^{n\times k}$ be the matrix with columns $u_1, \ldots, u_k$, and let $U_k^{\dagger}$ be its conjugate transpose. We define the **k-th Lens Principal component** of Y as the vector

$$v_k \;:=\; U_k \cdot \mathsf{LastLensComp}\left(\frac{U_k^{\dagger} y_1}{\|U_k^{\dagger} y_1\|}, \ldots, \frac{U_k^{\dagger} y_N}{\|U_k^{\dagger} y_N\|}\right)$$

This inductive procedure yields a collection $[v_2], \ldots, [v_n] \in L_q^n$, and we let $v_1 \in S^{2n-1}$ be such that $\mathsf{span}_{\mathbb{C}}\{v_1\} = \mathsf{span}_{\mathbb{C}}\{v_2, \ldots, v_n\}^{\perp}$. Finally

$$\mathsf{LPCA}(Y) := \{[v_1], \ldots, [v_n]\}$$

are the **Lens Principal Components** of Y. Let $V_k \in \mathbb{C}^{n\times k}$ be the n-by-k matrix with columns $v_1, \ldots, v_k$, and let $P_k(Y) \subset L_q^k$ be the set of classes $\left[\frac{V_k^{\dagger} y_j}{\|V_k^{\dagger} y_j\|}\right]$, $1 \leq j \leq N$. The point clouds $P_k(Y)$, $k = 1, \ldots, n$, are the **Lens Principal Coordinates** of Y.

5.4.2 Choosing a target dimension

The **variance recovered** by the first k Lens Principal Components $[v_1], \ldots, [v_k] \in L_q^n$ is defined as

$$\mathsf{var}_k(Y) := \frac{1}{N}\sum_{l=2}^{k}\sum_{j=1}^{N} d_L\left(\left[\frac{V_l^{\dagger} y_j}{\|V_l^{\dagger} y_j\|}\right], L_q^{l-1}(e_{l-1})\right)^2$$

where V_l is the n-by-l matrix with columns $v_1, \ldots, v_l$, $1 < l \leq k$, and $e_{l-1} \in \mathbb{C}^l$ is the vector $[0, \ldots, 0, 1, 0]$.

Therefore, the **percentage of cumulative variance** $p.\mathsf{var}(k) := \mathsf{var}_k(Y)/\mathsf{var}_n(Y)$, can be interpreted as the portion of total variance of Y along $\mathsf{LPCA}(Y)$, explained by the first k components.

Thus we can select the target dimension as the smallest k for which $p.\mathsf{var}_k(Y)$ is greater than a predetermined value. In other words, we select the dimension that recovers a significant portion of the total variance. Another possible guideline to choose the target dimension is as the minimum value of k for which $p.\mathsf{var}(k) - p.\mathsf{var}(k+1) < \gamma$ for a small $\gamma > 0$.

5.4.3 Independence of the cocycle representative

Let $\eta \in Z^1(\mathcal{N}(\mathcal{B}_\epsilon); \mathbb{Z}_q)$ be such that $[\eta] \neq 0$ in $H^1(\mathcal{N}(\mathcal{B}_\epsilon); \mathbb{Z}_q)$, and let $\eta' = \eta + \delta^0(\alpha)$ with $\alpha \in C^0(\mathcal{N}(\mathcal{B}_\epsilon); \mathbb{Z}_q)$. If $b \in U_j$, then

$$f_{\eta'}(b) = [\sqrt{\phi_1(b)}\zeta_q^{\eta_{j1}+\alpha_1} : \cdots : \sqrt{\phi_n(b)}\zeta_q^{\eta_{jn}+\alpha_n}]$$

If Z_α is the square diagonal matrix with entries $\zeta_q^{\alpha_1}, \zeta_q^{\alpha_2}, \ldots, \zeta_q^{\alpha_n}$, then $f_{\eta'}(b) = Z_\alpha \cdot f(b)$. Moreover, after taking classes in L_q^n, this implies that $f_{\eta'}(X) = Z_\alpha \cdot f(X)$. Since $\mathsf{Cov}(Z_\alpha \cdot f(X)) = Z_\alpha \mathsf{Cov}(f(X)) Z_\alpha^\dagger$ and Z_α is orthonormal, then if v is an eigenvector of $\mathsf{Cov}(f(X))$ with eigenvalue σ, we also have that $Z_\alpha v$ is an eigenvector of $\mathsf{Cov}(Z_\alpha \cdot f(X))$ with the same eigenvalue. Therefore

$$\mathsf{LastLensComp}(f_{\eta'}(X)) = Z_\alpha \mathsf{LastLensComp}(f(X)).$$

Since each component in LPCA is obtained in the same manner, we must have that $\mathsf{LPCA}(f_{\eta'}(X)) = Z_\alpha \mathsf{LPCA}(f(X))$. Thus, the lens coordinates from two cohomologous cocycles η and $\eta + \delta^0(\alpha)$ (i.e., representing the same cohomology class) only differ by the isometry of L_q^n induced by the linear map Z_α.

5.4.4 Visualization map for L_3^2

Given $v_1, \ldots, v_n \in S^{2n-1}$ representatives for the classes in $\mathsf{LPCA}(Y)$. We want to visualize $P_2(Y) \subset L_3^2$ in the fundamental domain described in Chapter 5.1.2.1. Let

$$P_2(Y) = \left\{ \left[\langle y_i, v_1 \rangle_{\mathbb{C}}, \langle y_i, v_2 \rangle_{\mathbb{C}} \right] \in S^3 \subset \mathbb{C}^2 : [y_i] \in Y \right\}$$

and define $G : P_2(Y) \longrightarrow S^3 \subset \mathbb{C}^2$ to be

$$G(z,w) := \left({\zeta_3}^{-k} z, \left(\arg(w) - \frac{\pi}{3} \right) \sqrt{1 - \|z\|^2} \right) \tag{5.4}$$

where $\arg(w) \in \left[0, \frac{2\pi}{3}\right)$, and k an integer such that

$$\arg(z) = k\frac{2\pi}{3} + \theta,$$

where θ is the remainder after division by $\frac{2\pi}{3}$.

Metric on the Moore space $M(\mathbb{Z}_3, 1)$**.** For $x, y \in \mathbb{C}$ with $|x|, |y| \leq 1$, we let

$$d(x,y) = \begin{cases} \sqrt{|\langle x, y \rangle_{\mathbb{R}}|} & \text{if } \|x\|, \|w\| < 1 \\ \min\limits_{\zeta \in \mathbb{Z}_3} \sqrt{|\langle x, \zeta y \rangle_{\mathbb{R}}|} & \text{if } \|x\| = 1 \text{ or } \|w\| = 1 \\ \min\limits_{\zeta \in \mathbb{Z}_3} \arccos(|\langle x, \zeta y \rangle_{\mathbb{R}}|) & \text{if } \|x\| = 1 \text{ and } \|w\| = 1 \end{cases} . \tag{5.5}$$

5.5 Examples

5.5.1 The Circle S^1

Let $S^1 \subset \mathbb{C}$ be the unit circle, and let X a random sample around S^1, with $10,000$ points and Gaussian noise in the normal direction. $L \subset X$ is a landmark set with 10 points obtained as described in Chapter 5.3.1.

Let a be the cohomological death of the most persistent class $PH^1(\mathcal{R}(L); \mathbb{Z}_q)$. For $\epsilon := a + 10^{-5}$ and $\eta = i^*(\eta') \in Z^1(\mathcal{N}(\mathcal{B}_\epsilon); \mathbb{Z}_q)$ we define the map $f : B_\epsilon \to L_3^{10}$ as in Chapter 5.1.

After computing LPCA for $f(X) \subset L_3^{10}$ and the percentage of cumulative variance $p.\mathsf{var}_Y(k)$ we obtain the row in Table 5.1 with label S^1 (see Chapter **??** for more details). We see that dimension 1 recovers $\sim 60\%$ of the variance. Moreover, Chapter 5.3 shows $P_2(f(X)) \subset L_3^2$ in the fundamental domain described in Chapter 5.1.2.1 trough the map in Chapter 5.4.

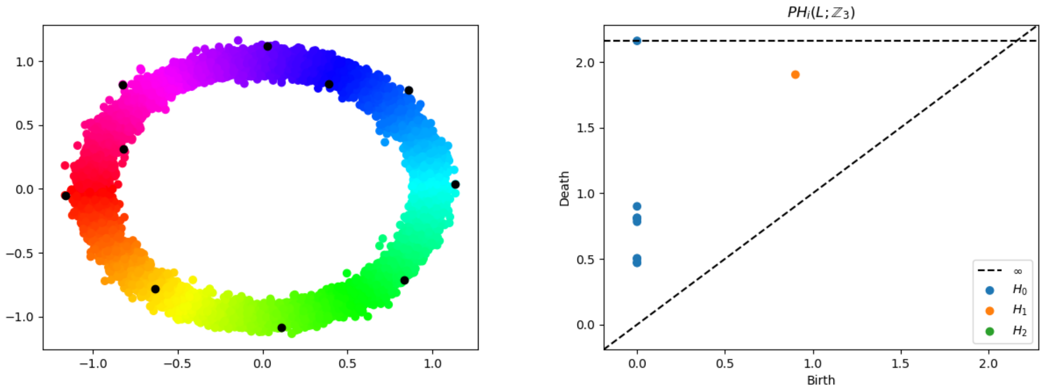

Figure 5.1 **Left:** Sample X, in black landmark set $L \subset X$. **Right:** $PH^i(\mathcal{R}(L);\mathbb{Z}_3)$ for $i = 0, 1, 2$.

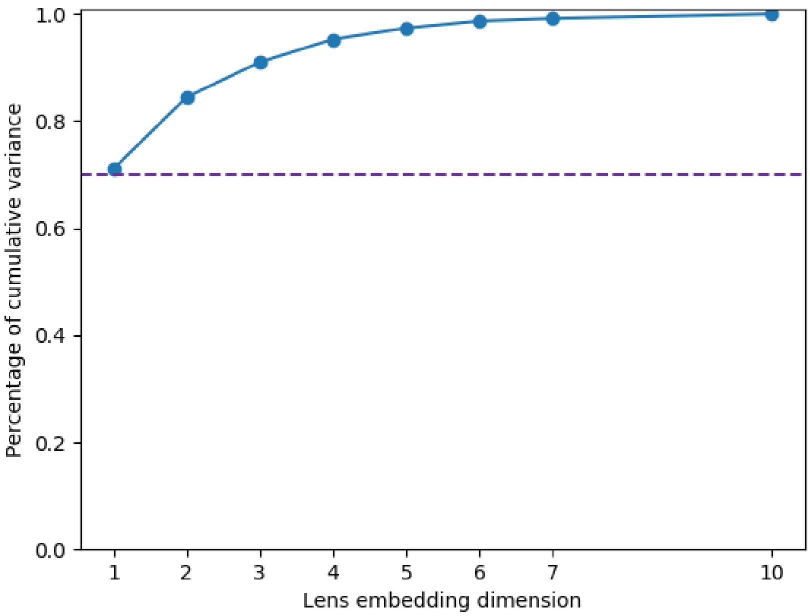

Figure 5.2 Percentage of recovered variance.

Dim. (n)	**1**	**2**	**3**	**4**	**5**
S^1	0.62	0.75	0.81	0.86	0.89
$M(\mathbb{Z}_3, 1)$	0.56	0.7	0.76	0.8	0.83
L_3^2	0.47	0.62	0.67	0.71	0.73

Table 5.1 Percentage of recovered variance in L_3^n.

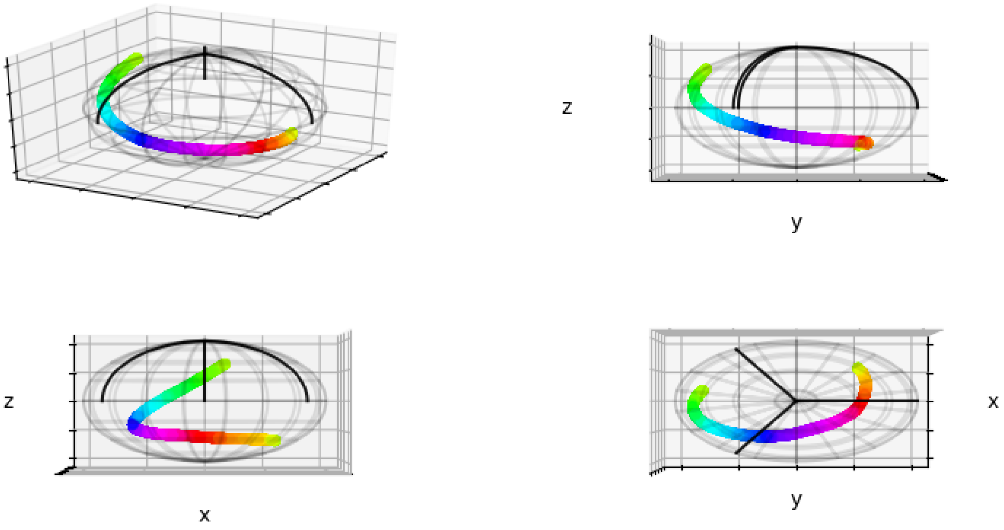

Figure 5.3 Visualization $P_2(f(X)) \subset L_3^2$.

One key aspect of LC (Lens coordinates) is that it is designed to highlight the cohomology class η used on Chapter 5.1. This is easily observed in this example; we selected the most persistent class in $PH^1(\mathcal{R}(L); \mathbb{Z}_3)$ and as a consequence in Chapter 5.3 we see how this class is preserved while all the information in the normal direction is lost in the process.

5.5.2 The Moore space $M(\mathbb{Z}_3, 1)$

Let G be an abelian group and $n \in \mathbb{N}$. The Moore space $M(G, n)$ is a CW-complex such that $H_n(M(G, n), \mathbb{Z}) = G$ and $\tilde{H}_i(M(G, n), \mathbb{Z}) = 0$ for all $i \neq n$. A well known construction for $M(\mathbb{Z}_3, 1)$ can be found in [39]. Chapter 5.5 defines a metric on $M(\mathbb{Z}_3, 1)$.

Chapter 5.4, on the left, shows a sample $X \subset M(\mathbb{Z}_3, 1)$ with $\|X\| = 15,000$ and 70 landmarks. The landmarks were obtained by minmax sampling after feeding the algorithm with an initial set of 10 point on the boundary on the disc. Chapter 5.5 shows the persistent cohomology of $\mathcal{R}(L)$ with coefficients in $\mathbb{Z}_2$ and $\mathbb{Z}_3$ side-by-side.

We compute $f : M(\mathbb{Z}_3, 1) \longrightarrow L_3^{70}$ analogously to the previous example and obtain a point cloud $f(X) \subset L_3^{70}$. The profile of recovered variance is shown in Table 5.1. Dimension

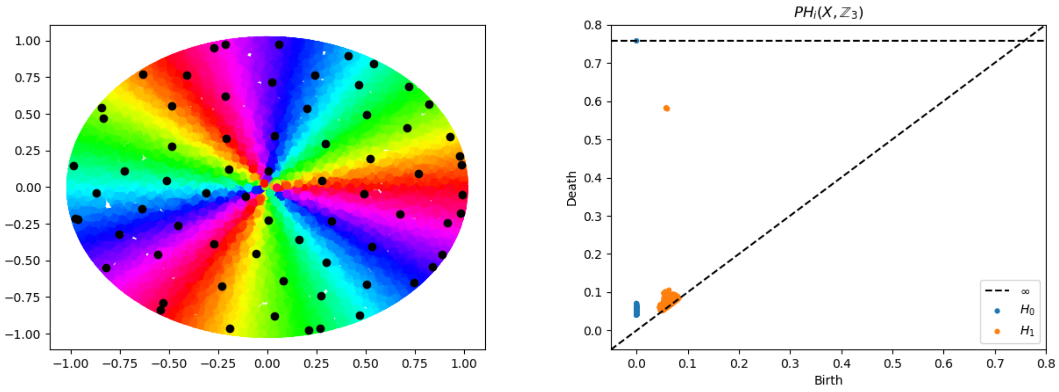

Figure 5.4 **Left:** $X \subset M(\mathbb{Z}_3, 1)$ with landmarks in black. **Right:** $PH^i(\mathcal{R}(L); \mathbb{Z}_3)$ for $i = 0, 1$.

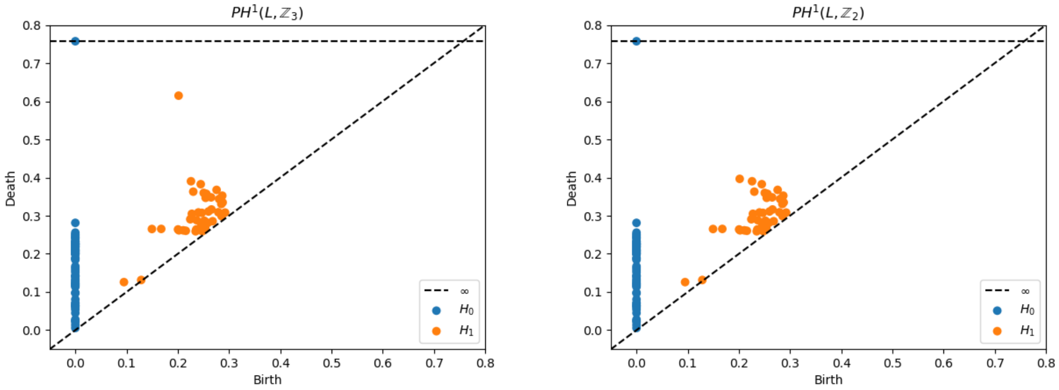

Figure 5.5 $PH^i(R(L); \mathbb{F})$ for $i = 0, 1$ and $\mathbb{F} = \mathbb{Z}_2, \mathbb{Z}_3$.

2 provides a low dimensional representation of $f(X)$ inside L^2_3 with 70% of recovered variance (see Figure 5.6).

Since f classifies the principal $\mathbb{Z}_3$-bundle P_η over $M(\mathbb{Z}_3, 1)$, then f must be homotopic to the inclusion of $M(\mathbb{Z}_q, 1)$ in L^∞_q. Chapter 5.7 shows $X \subset M(\mathbb{Z}_3, 1)$ mapped by f in L^2_3. Notice the identifications on X are handled by the identification on $S^1 \times \{0\} \subset D^3$ from the fundamental domain on Chapter 5.1.2.1. See https://youtu.be/_Ic730_xFkw for a more complete visualization.

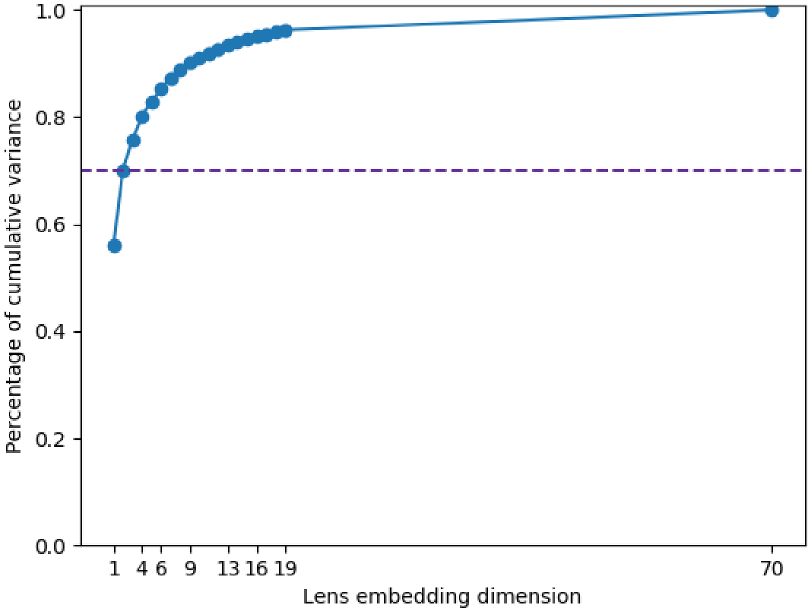

Figure 5.6 Percentage of recovered variance.

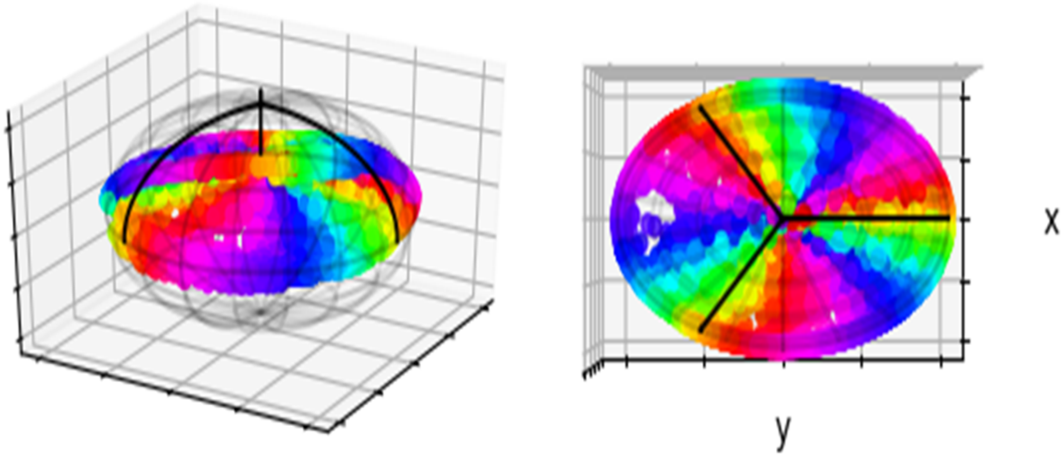

Figure 5.7 Visualization of the resulting $P_2(f(X)) \subset L_3^2$.

5.5.3 The Lens space $L_3^2 = S^3/\mathbb{Z}_3$

We use the metric defined in Chapter 5.3 on L_3^2 and randomly sample $15,000$ points to create $X \subset L_3^2$. Chapter 5.9(left) shows the sample set using the fundamental domain from

Chapter 5.1.2.1.

We can use $PH^i(\mathcal{R}(X);\mathbb{Z}_2)$ and $PH^i(\mathcal{R}(X);\mathbb{Z}_3)$ to verify that the sampled metric space has the expected topological features. Figure 5.8 contains the corresponding persistent diagrams.

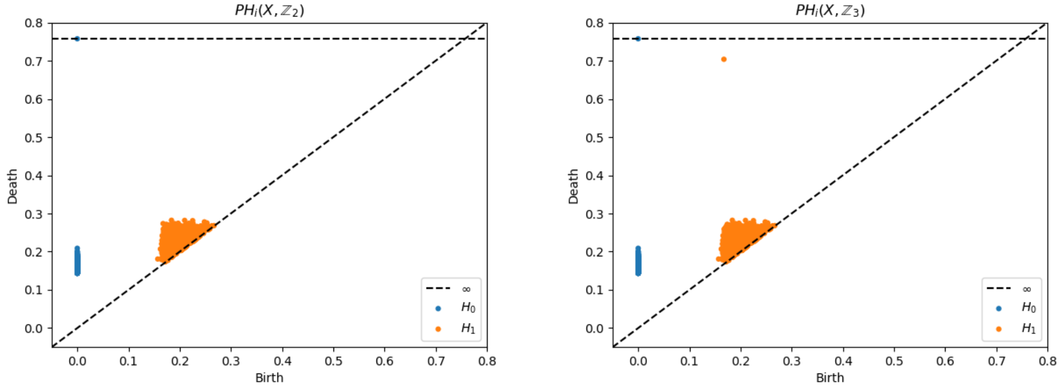

Figure 5.8 $PH^i(R(L);\mathbb{F})$ for $i = 0, 1$ and $\mathbb{F} = \mathbb{Z}_2, \mathbb{Z}_3$.

Just as in the previous examples define $f : L_3^2 \to L_3^\infty$ using the most persistent class in $PH^1(\mathcal{R}(L);\mathbb{Z}_3)$. The homotopy class of f must be the same as that of the inclusion $L_3^2 \subset L_3^\infty$, since f classifies the $\mathbb{Z}_3$-principal bundle P_η. Thus we expect L_3^2 to be preserved up to homotopy under LPCA. Chapter 5.9 offers a side and top view of $P_2(f(X)) \subset L_3^2$. Here we clearly see how the original data set X is transformed while preserving the identifications on the boundary of the fundamental domain.

Finally in Figure 5.10 we show the variance profile for the dimensionality reduction problem. We see that for dimension 4 we have recovered more than 70% of the total variance as seen in Table 5.1.

5.5.4 Isomap dimensionality reduction

We conclude this section by providing evidence that Lens coordinates (LC) preserve topological features when compared to other dimensionality reduction algorithms. For this purpose we use Isomap ([41]) as our point of comparison.

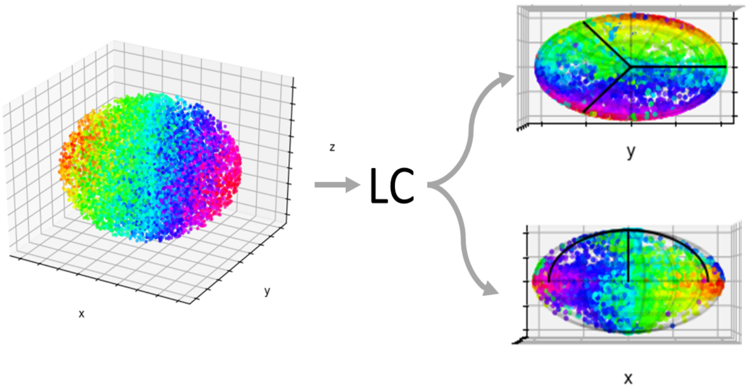

Figure 5.9 **Left:** $X \subset L_3^2$. **Right:** Lens coordinates.

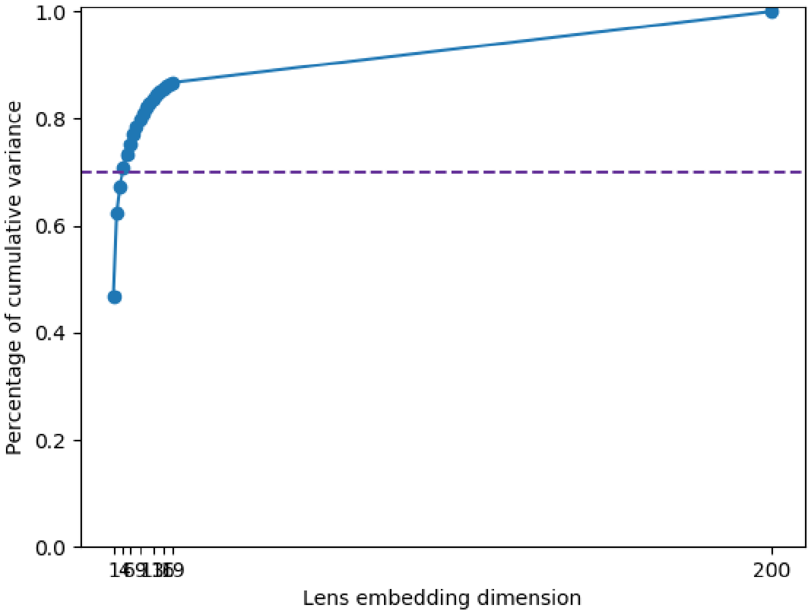

Figure 5.10 Percentage of recovered variance.

The Isomap algorithm consist of 3 main steps. The first step determines neighborhoods of each point using k-th nearest neighbors. The second step estimates the geodesic distances between all pairs of points using shortest distance path, and the final step applies classical MDS to the matrix of graph distances.

Let dgm be a persistent diagram. Define per_1 to be the largest persistence of an element in dgm, and let per_2 be the second largest persistence of an element dgm.

$\mathsf{per}_1/\mathsf{per}_2$		$\mathbb{Z}_2$	$\mathbb{Z}_3$
$M(\mathbb{Z}_q, 1)$	Isomap	1.0105	1.0105
	LC	1.7171	3.6789
L_3^2	Isomap	1.0080	1.0080
	LC	1.1592	2.8072

Table 5.2 In green we highlight the fraction that indicates which method better identifies the topological features.

For both $M(\mathbb{Z}_3, 1)$ and L_3^2 it is clear that the Isomap projection fails to preserve the difference between the cohomology groups with coefficients in $\mathbb{Z}_2$ and $\mathbb{Z}_3$. On the other hand the LC projections maintains this difference in both examples.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] J. A. Perea. “A Brief History of Persistence”. In: *preprint arXiv:1809.03624* (2018). https://arxiv.org/abs/1809.03624.

[2] P. Bubenik and P. Dlotko. “A persistence landscapes toolbox for topological statistics”. In: *arXiv e-prints*, arXiv:1501.00179 (2014), arXiv:1501.00179. arXiv: 1501.00179.

[3] Z. Cang et al. “A topological approach for protein classification”. In: *Computational and Mathematical Biophysics* 3 (2015).

[4] J. Reininghaus et al. “A Stable Multi-Scale Kernel for Topological Machine Learning”. In: *arXiv e-prints*, arXiv:1412.6821 (2014), arXiv:1412.6821. arXiv: 1412.6821.

[5] J. A. Perea, A. Munch, and F. A. Khasawneh. “Approximating Continuous Functions on Persistence Diagrams Using Template Functions”. In: *CoRR* abs/1902.07190 (2019). arXiv: 1902.07190. URL: http://arxiv.org/abs/1902.07190.

[6] H. Adams et al. “Persistence Images: A Stable Vector Representation of Persistent Homology”. In: *Journal of Machine Learning Research* 18.8 (2017), pp. 1–35. URL: http://jmlr.org/papers/v18/16-337.html.

[7] A. Smith et al. “Supervised Learning of Labeled Pointcloud Differences via Cover-Tree Entropy Reduction”. In: *arXiv e-prints*, arXiv:1702.07959 (2017), arXiv:1702.07959. arXiv: 1702.07959.

[8] D. Reynolds. “Gaussian Mixture Models”. In: *Encyclopedia of Biometrics*. Ed. by Stan Z. Li and Anil Jain. Boston, MA: Springer US, 2009, pp. 659–663. ISBN: 978-0-387-73003-5. DOI: 10.1007/978-0-387-73003-5_196. URL: https://doi.org/10.1007/978-0-387-73003-5_196.

[9] R. J. G. B. Campello, D. Moulavi, and J. Sander. “Density-Based Clustering Based on Hierarchical Density Estimates”. In: *Advances in Knowledge Discovery and Data Mining*. Ed. by J. Pei et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 160–172. ISBN: 978-3-642-37456-2.

[10] D. Pickup et al. “Shape Retrieval of Non-rigid 3D Human Models”. In: *Proceedings of the 7th Eurographics Workshop on 3D Object Retrieval*. 3DOR ’14. Strasbourg, France: Eurographics Association, 2014, pp. 101–110. ISBN: 978-3-905674-58-3. DOI: 10.2312/3dor.20141056. URL: https://doi.org/10.2312/3dor.20141056.

[11] P. Sonego et al. “A Protein Classification Benchmark collection for machine learning”. In: *Nucleic acids research* 35 (Feb. 2007), pp. D232–6. DOI: 10.1093/nar/gkl812.

[12] G. Carlsson. “Topological pattern recognition for point cloud data”. In: *Acta Numerica* 23 (2014), 289–368. DOI: 10.1017/S0962492914000051.

[13] J. A. Perea and G. Carlsson. “A Klein-Bottle-Based Dictionary for Texture Representation”. In: *International Journal of Computer Vision* 107 (Mar. 2014), pp. 75–97. DOI: 10.1007/s11263-013-0676-2.

[14] G. Carlsson et al. “On the Local Behavior of Spaces of Natural Images”. In: *Int. J. Comput. Vision* 76.1 (Jan. 2008), pp. 1–12. ISSN: 0920-5691. DOI: 10.1007/s11263-007-0056-x. URL: http://dx.doi.org/10.1007/s11263-007-0056-x.

[15] J. Milnor. “Construction of universal bundles, II”. In: *Annals of Mathematics* (1956), pp. 430–436.

[16] R. Miranda. *Algebraic curves and Riemann surfaces*. Vol. 5. American Mathematical Soc., 1995.

[17] E. H. Brown. “Cohomology theories”. In: *Annals of Mathematics* (1962), pp. 467–484.

[18] J. A. Perea. “Sparse Circular Coordinates via Principal $\mathbb{Z}$-Bundles”. In: *arXiv e-prints*, arXiv:1809.09269 (2018), arXiv:1809.09269. arXiv: 1809.09269 `[math.AT]`.

[19] J. A. Perea. “Multiscale projective coordinates via persistent cohomology of sparse filtrations”. In: *Discrete & Computational Geometry* 59.1 (2018), pp. 175–225.

[20] H. Gakhar and J. A. Perea. *Sliding Window Persistence of Quasiperiodic Functions*. 2021. arXiv: 2103.04540.

[21] C. J. Tralie and J. A. Perea. “(Quasi)Periodicity Quantification in Video Data, Using Topology”. In: *SIAM Journal on Imaging Sciences* 11.2 (2018), pp. 1049–1077. DOI: 10.1137/17M1150736. eprint: https://doi.org/10.1137/17M1150736. URL: https://doi.org/10.1137/17M1150736.

[22] B. Xu et al. “Twisty Takens: a geometric characterization of good observations on dense trajectories”. In: *Journal of Applied and Computational Topology* 3 (2019), 285–313. DOI: https://doi.org/10.1007/s41468-019-00036-9.

[23] U. Bauer. *Ripser: efficient computation of Vietoris-Rips persistence barcodes*. 2021. arXiv: 1908.02518.

[24] J. R. Munkres. *Elements of Algebraic Topology.* Addison Wesley Publishing Company, 1984. ISBN: 0201045869. URL: http://www.worldcat.org/isbn/0201045869.

[25] A. Hatcher. *Algebraic topology.* Cambridge: Cambridge Univ. Press, 2000. URL: https://cds.cern.ch/record/478079.

[26] H. Edelsbrunner and J. Harer. *Computational Topology: An Introduction.* Applied Mathematics. American Mathematical Society, 2010. ISBN: 9780821849255.

[27] F. Chazal et al. *The Structure and Stability of Persistence Modules.* SpringerBriefs in Mathematics. Springer Verlag, 2016, pp. VII, 116. URL: https://hal.inria.fr/hal-01330678.

[28] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer. "Stability of Persistence Diagrams". In: *Discrete & Computational Geometry* 37.1 (2007), pp. 103–120. ISSN: 1432-0444. DOI: 10.1007/s00454-006-1276-5. URL: https://doi.org/10.1007/s00454-006-1276-5.

[29] J. Sun, M. Ovsjanikov, and L. Guibas. "A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion". In: *Computer Graphics Forum* (2009). ISSN: 1467-8659. DOI: 10.1111/j.1467-8659.2009.01515.x.

[30] N. Fox, S. Brenner, and J. Chandonia. "SCOPe: Structural Classification of Proteins - Extended, integrating SCOP and ASTRAL data and classification of new structures". In: *Nucleic acids research* 42 (Dec. 2013). DOI: 10.1093/nar/gkt1240.

[31] W. Crawley-Boevey. "Decomposition of pointwise finite-dimensional persistence modules". In: *Journal of Algebra and its Applications* 14.05 (2015), p. 1550066.

[32] H. Edelsbrunner and J. Harer. *Computational Topology - an Introduction.* American Mathematical Society, 2010, pp. I–XII, 1–241. ISBN: 978-0-8218-4925-5.

[33] A. Zomorodian and G. Carlsson. "Computing Persistent Homology". In: *Discrete Computational Geometry* 33 (2005), pp. 249–274. URL: https://doi.org/10.1007/s00454-004-1146-y.

[34] V. de Silva, D. Morozov, and M. Vejdemo-Johansson. "Dualities in persistent (co)homology". In: *Inverse Problems* 27.12, 124003 (2011), p. 124003. DOI: 10.1088/0266-5611/27/12/124003. arXiv: 1107.5665.

[35] D. Cohen-Steiner, H. Edelsbrunner, and D. Morozov. "Vines and vineyards by updating persistence in linear time." In: *Symposium on Computational Geometry.* Ed. by Nina Amenta and Otfried Cheong. ACM, 2006, pp. 119–126. ISBN: 1-59593-340-9. URL: http://dblp.uni-trier.de/db/conf/compgeom/compgeom2006.html#Cohen-SteinerEM06.

[36] N. Saul and C. J. Tralie. *Scikit-TDA: Topological Data Analysis for Python*. 2019. DOI: 10.5281/zenodo.2533369. URL: https://doi.org/10.5281/zenodo.2533369.

[37] J. A. Perea and J. Harer. “Sliding Windows and Persistence: An Application of Topological Methods to Signal Analysis”. In: *Foundations of Computational Mathematics* 15 (2015), 799–838. DOI: https://doi.org/10.1007/s10208-014-9206-z.

[38] W. Crawley-Boevey. “Decomposition of pointwise finite-dimensional persistence modules”. In: *Journal of Algebra and its Applications* 14.05 (2015), p. 1550066.

[39] A. Hatcher. *Algebraic topology*. Cambridge University Press, 2002.

[40] D. Husemoller and D. Husemöller. *Fibre Bundles*. Graduate Texts in Mathematics. Springer, 1994. ISBN: 9780387940878. URL: https://books.google.com/books?id=DPr_BSH89cAC.

[41] J. B. Tenenbaum, V. de Silva, and J. C. Langford. “A Global Geometric Framework for Nonlinear Dimensionality Reduction”. In: *Science* 290.5500 (2000), p. 2319.