Multiscale 2-Mapper: Exploratory Data Analysis Guided by the First Betti Number

by

Halley Ann Fritze

A dissertation accepted and approved in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

in Mathematics

Dissertation Committee:

Dr. Dev Sinha, Chair

Dr. Peter Ralph, Core Member

Dr. Luca Mazucatto, Core Member

Dr. David Levin, Core Member

Dr. Tao Hou, Institutional Representative

University of Oregon

Spring 2025

DISSERTATION ABSTRACT

Halley Ann Fritze

Doctor of Philosophy in Mathematics

Title: Multiscale 2-Mapper: Exploratory Data Analysis Guided by the First Betti Number

The Mapper algorithm is a fundamental tool in exploratory topological data analysis for identifying connectivity and topological clustering in data. Derived from the nerve construction, Mapper graphs can contain additional information about clustering density when considering the higher-dimensional skeleta. To observe two-dimensional features, and capture one-dimensional topology, we construct *2-Mapper*. Computationally, we study how cover choice affects 2-Mapper and analyze this through a computational Multiscale Mapper algorithm. A common issue in using Mapper algorithms is parameter choice. We develop tools to choose 2-Mapper parameters reflect persistent Betti-1 information. We test our constructions on data produced from chaotic dynamical systems.

CURRICULUM VITAE

NAME OF AUTHOR: Halley Ann Fritze

GRADUATE AND UNDERGRADUATE SCHOOLS ATTENDED:

University of Oregon, Eugene
North Dakota State University, Fargo

DEGREES AWARDED:

Doctor of Philosophy, Mathematics, 2025, University of Oregon
Master of Science, Mathematics, 2021, University of Oregon
Bachelor of Science, Mathematics, 2017, North Dakota State University

AREAS OF SPECIAL INTEREST:

Computational Topology
Population Genetics

PROFESSIONAL EXPERIENCE:

Statistician, Stanford Genome Technology Center

GRANTS, AWARDS, AND HONORS:

Ronald E. McNair Scholar
Marie Vitulli Scholar

PUBLICATIONS:

H. Fritze, *Multiscale 2-Mapper – Exploratory Data Analysis Guided by the First Betti Number*, Accepted to ATMCS11. To be published in LaMatematica (2025)
H. Fritze et al., *Faithful Reeb Graph Reconstruction of a Tectonic Subduction Zone from Earthquake Hypocenters*, Accepted to ATMCS11. To be published in LaMatematica (2025)
H. Fritze et al., *A forest is more than its trees: haplotypes and ancestral recombination graphs*, Submitted to Genetics, Under 2nd Review (2025)

ACKNOWLEDGMENTS

I would first like to thank my advisor, Dr. Dev Sinha, for allowing me to explore all the wonderful aspects of topological data analysis, and independently pursue my mathematical interests. I would also like to thank Dr. Peter Ralph for being a wonderful mentor, collaborator, and teacher of all things programming. I would also like to acknowledge all of my other collaborators who made me feel welcome in the wonderful field of topological data analysis and inspire to continue research. For keeping my sanity across this crazy experience, I am indebted to the Fabulous Five. Long may Dave reign.

For Alex.

Thank you for joining me on this adventure.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Page

LIST OF TABLES

Table Page

CHAPTER 1

TOPOLOGICAL DATA ANALYSIS

Topological data analysis (TDA) uses geometric structure to understand data. As topological features can be robust to noise and are often interpretable, such an approach can be appropriate for real-world data can complement statistical analysis and machine learning. The most prevalent technique within TDA is persistent homology [Car09] of simplicial complexes defined using distances between data points. Another successful technique is Mapper [SMC07], which provides a graphical representation of a soft clustering label function. In this thesis we find compelling rationale for and application of a novel combination of these techniques. In the following chapter, we define these tools and discuss their applications.

1.1 Mapper

Developed by Singh, Mémoli, and Carlsson [SMC07], the Mapper algorithm is an unsupervised manifold learning technique that generalizes clustering and visualizes the labeling as a *Mapper graph*. Mapper is effective in identifying connectivity in a data set, and it has known applications in a variety of fields, including machine learning [Pur+23; Lov+23], political science [Hus+22], biology [Amé+23; CV21b; SBH20; NLC11], and neuroscience [ZCS23; Sag+18].

Definition 1.1 (Mapper)**.** Let $X \subseteq \mathbb{R}^d$ be a data set. Fix a choice of continuous reference function $f : X \to \mathbb{R}^m$, a cover $\mathcal{U}$ of the image $f(X)$, and clustering algorithm. Define the *Mapper graph* of X with respect to these choices as the graph whose vertices are given by the clusters $C_{\alpha,k}$ produced by the clustering algorithm for $f^{-1}(U_\alpha)$, taken over all α. There is an edge between $C_{\alpha,k}$ and $C_{\alpha',k'}$ if their intersection is nonempty.

The idea informally is to enhance clusters by allowing and accounting for overlap, which can in some cases capture essential shape, as in Figure 1.1(A). In its first published application, Mapper was used to identify clusters of cancer patients and in the process revealed a new genetic marker c-MYB$^+$, as in Figure 1.1(B). This marker was missed in previous meta-studies and, in particular, is not evident within hierarchical clustering.

With many parameters, Mapper is a flexible tool but one for which stability and other theoretical properties can be difficult to establish [CMO17]. One of our

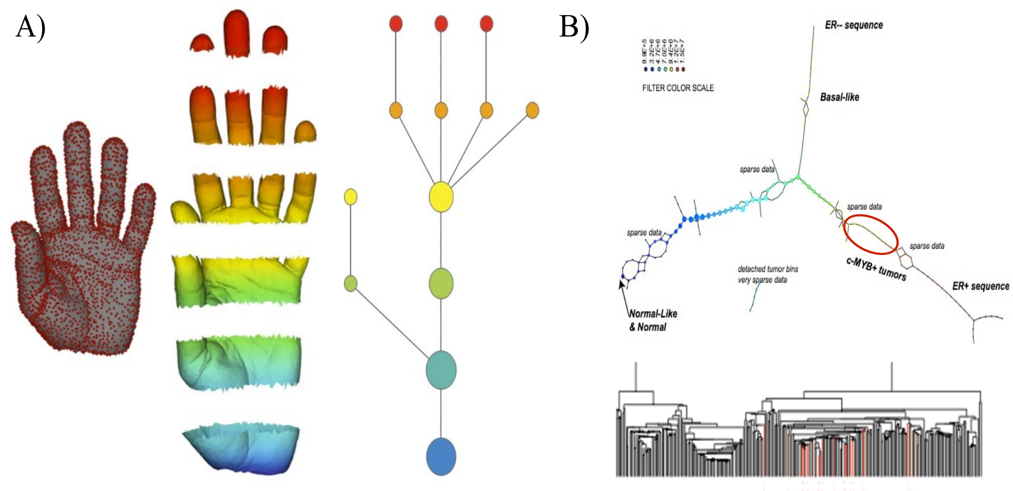

Figure 1.1. A) The Mapper algorithm for a point cloud of a hand. B) A Mapper graph (top) and hierarchical clustering (bottom) of micro-array breast cancer data. The classification for c-MYB$^+$ tumors are a distinct group circled in the Mapper graph but are spread out (highlighted in red) in hierarchical clustering [NLC11, Figures 3-4].

main results in Chapter 3 will be to understand ranges of parameters and make choices based on persistent homology, defined below.

In common implementations of Mapper there are some standard choices for each parameter. For example, real valued functions such as coordinate functions or principal components [Hot36] are the most common reference functions. [Amé+23; Pur+23; CM21; CMO17]. Additionally, most algorithms [Alv+24; Bui+20; CZW21] which find optimal cover parameters for mapper only apply to covers on $\mathbb{R}$. In higher dimensions, the standard choice is uniform hypercubes $\mathcal{U} = \{U_\alpha\}_{\alpha\in A}$ where $U_\alpha = \prod_{i=1}^{m}[a_{\alpha,i}, b_{\alpha,i}]$ for $a_{\alpha,i}, b_{\alpha,i} \in \mathbb{R}$. The clustering algorithm is typically DBSCAN [Est+96] or some form of agglomerative clustering.

In addition to parameter choices, one can alter the node color and size to correspond to some feature statistics for X. Multiple Mapper implementations exist in Python, including `gudhi` [Pro24], `giotto` [Tau+20], `mapper-interactive` [Zho+21], and `kepler-mapper` [Vee+20]. There is also a Mapper package for R called `TDAmapper` [SPM15]. The main implementations constructed in this work are written in Python and dependent on `giotto-tda` and `gudhi`. We plan implementation in R in the future.

Many structures in topological data analysis are derived from a simplicial *nerve*. If S is a set, let Δ^S denote the simplex with vertex set S.

Definition 1.2 (Nerve)**.** Let $X \subseteq \mathbb{R}^d$ and let $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ be a cover on X. We define the *nerve* of $\mathcal{U}$ as the simplicial complex $\mathcal{N}(\mathcal{U}) \subset \Delta^A$, where a collection of $n+1$ points $\sigma = \{\alpha_0, ..., \alpha_n\} \subseteq A$ spans an n-simplex in $\mathcal{N}(\mathcal{U})$ if and only if $U_{\alpha_0} \cap \cdots \cap U_{\alpha_n} \neq \emptyset$.

For example, the Mapper graph as defined in Definition 1.1 is the 1-skeleton of the nerve of the cover defined by clustered subsets derived from the cover of a data set (pulled back through its reference function). The nerve can provide a combinatorial encoding of homotopy type. Borsuk in 1948 gave a condition for a nerve to be homotopy equivalent to the space from which it is constructed.

Lemma 1.3 (Nerve Lemma [Bor48])**.** *Let X be a metric space with finite cover $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ by open sets. Then the the underlying space $|\mathcal{N}(\mathcal{U})|$ is homotopy equivalent to X if every non-empty intersection $\cap_{\alpha \in A} U_\alpha$ is contractible.*

Thus, when Borsuk's condition applies, the homology of the nerve will be that of the homology of X.

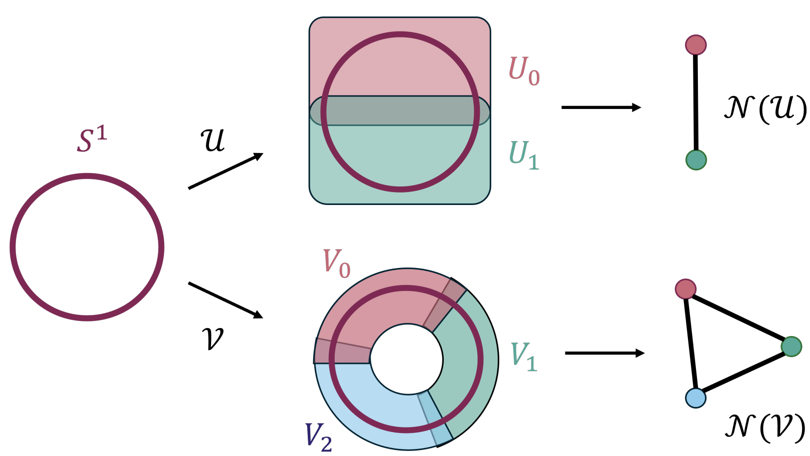

Figure 1.2. Given two covers $\mathcal{U}$ and $\mathcal{V}$ of S^1, we compute their nerves $\mathcal{N}(\mathcal{U})$ and $\mathcal{N}(\mathcal{V})$.

Figure 1.2 gives a simple example of computed nerves for covers $\mathcal{U}$ and $\mathcal{V}$ of S^1. We see that $U_0 \cap U_1$ is not contractible and its resulting nerve $\mathcal{N}(\mathcal{U})$ is not homotopy equivalent to S^1. In the second case, each pair of cover sets in $\mathcal{V}$ is contractible, satisfying Lemma 1.3, thus $\mathcal{N}(\mathcal{V})$ is homotopy equivalent to S^1. As a simplicial complex, the nerve of a cover gives a concrete approach to the homotopy type of a general metric space.

Data sets typically have natural metrics, in particular the standard Euclidean metric for numerical data. Though data sets are discrete, a nerve can "fill in" a homotopy type. We can construct a simplicial complex from a metric space through a nerve construction, with standard choices being the *Čech complex* and *Vietoris-Rips complex*.

Definition 1.4 (Čech Complex)**.** Let (X, d_X) be a metric space. Consider the cover $U_r = \{B_r(x) \colon x \in X\}$, which covers each point of X with a ball of radius $r > 0$. Define the *Čech complex*, $\check{C}_r(X)$, as the nerve of the cover U_r. That is, $\check{C}_r(X) = \mathcal{N}(U_r)$.

The Čech complex is simple to define but is costly to compute for large data sets. In practice it is more common to use the Vietoris-Rips complex.

Definition 1.5 (Vietoris-Rips Complex)**.** Let (X, d_X) be a metric space, and consider the cover of radius $r > 0$ balls about each point in X, $U_r = \{B_r(x) \colon x \in X\}$. Define the *Vietoris-Rips complex* as the simplicial complex $\mathrm{VR}_r(X)$ where an n-simplex $\sigma \in \mathrm{VR}_r(X)$ if and only if $d_X(p, q) \leq 2r$ for all pairs of vertices $p, q \in \sigma$.

Because the simplex condition for Vietoris-Rips complex depends only on distances between pairs, it is the completion of its 1-skeleton.

The Vietoris-Rips complex and Čech complexes are not necessarily equivalent. For a Vietoris-Rips complex and Cech complex defined with the same radial value $r > 0$ it is possible that for $x, y \in X$ we have $d_X(x, y) \leq 2r$ but $B_r(x) \cap B_r(y) = \emptyset$. But these two simplicial complexes do *interleave*. That is, for $r > 0$,

$$\mathrm{VR}_r(X) \subseteq \check{C}_r(X) \subseteq \mathrm{VR}_{2r}(X).$$

Choosing one of these to work with for a particular data set is mostly a question of efficiency. For a data set X of size n, constructing its Čech complex requires checking the intersection of every possible combination of balls over X, which is 2^n computations. As this is exponential in computation time, that limits its use for large data sets. The Vietoris-Rips complex, being constructed from its 1-skeleton, requires on the order of n^2 computations. There are alternative sparse simplicial complexes for TDA [SC04; Del34], and many software implementations for topological data analysis allow the user to choose amongst these [Pro24; Tau+20].

1.3 Topological Persistence

Persistent homology [Car09] synthesizes information from Vietoris-Rips, Čech, or other simplicial complexes. It is a robust measure of the morphology of a dataset and has been used in a wide range of applications, including biology [Tho+21; Zhe+21; Tak+17], neuroscience [Kan+17; Kan+16], and material science [Lee+17].

When computing the topological characteristics of a metric space through the Čech or Vietoris-Rips complex $K_r(X)$, one must choose the radius r for the cover. There is no canonical choice or even a reasonable heuristic for choosing this radius. Instead we study all (or at least many) such choices, which naturally fit together as a *filtration* of simplicial complexes. As in Figure 1.3, for $r_i \leq r_{i+1}$ there exist inclusions between $K_{r_i} \hookrightarrow K_{r_{i+1}}$ which are simplicial maps. We take the collection $\{K_{r_i}(X)\}$ for some set of positive increasing r_i, namely

$$K_{r_0}(X) \subseteq K_{r_1}(X) \subseteq \cdots \subseteq K_{r_k}(X).$$

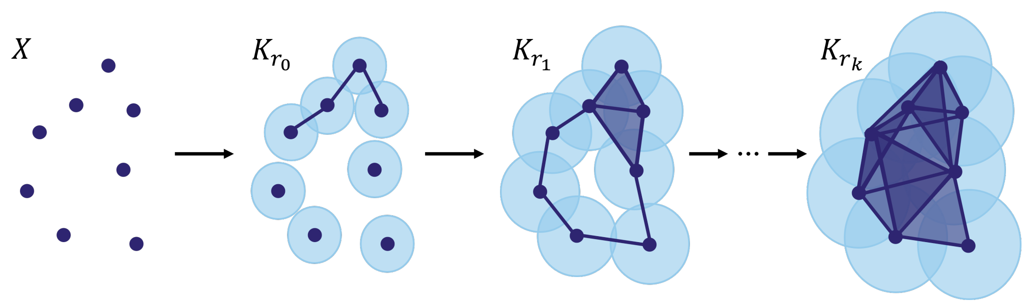

Figure 1.3. The Vietoris-Rips filtration computed from a point cloud X.

From each simplicial complex, we can compute its homology, generally using $\mathbb{Z}_2$ coefficients for computational simplicity. The inclusion maps between simplicial complexes induce linear maps between the homology vector spaces:

$$H_*(K_{r_0}(X;\mathbb{Z}_2)) \to H_*(K_{r_1}(X;\mathbb{Z}_2)) \to \cdots \to H_*(K_{r_k}(X;\mathbb{Z}_2)).$$

This sequence is known as a *persistence vector space*, and computing the homology across a simplicial filtration is referred to as *persistent homology*. Persistent homology highlights topological features which persist over the filtration parameter r.

As shown by Zomorodian and Carlsson, the notion of persistence is captured by an algebraic structure [ZC05]. When computed from finite data sets, a persistence vector space is a finitely generated $\mathbb{Z}_2[t]$-module where the t action encodes the map on homology induced by inclusion from $H_*(K_{r_i}(X;\mathbb{Z}_2)) \to H_*(K_{r_{i+1}}(X;\mathbb{Z}_2))$. Since $\mathbb{Z}_2[t]$ is a principle ideal domain, the persistent vector space can be decomposed as

$$\bigoplus_{i=0}^{k} H_p(K_{r_i};\mathbb{Z}_2) \cong \bigoplus_{a=0}^{m} \Gamma^{\alpha_a}\mathbb{Z}_2[t] \oplus \bigoplus_{b=0}^{l} \Gamma^{\beta_b}\mathbb{Z}_2[t]/(t^{n_b}) \tag{1.1}$$

where t^{n_b} divides $t^{n_{b+1}}$ and Γ shifts the grading. There is also a bijection between isomorphism classes in the above persistence vector space to interval modules defined as follows. Let $P(a,b) = \{P(a,b)_r\}_{r\in\mathbb{R}}$, where $a \in \mathbb{R}_+$, $b \in \mathbb{R}_+ \cup \{+\infty\}$ and $a < b$, consist of $P(a,b)_r = \mathbb{Z}_2$ when $r \in [a,b)$ and is 0 otherwise. In Equation (1.1), each free summand $\Gamma^{\alpha}\mathbb{Z}_2[t]$ is isomorphic to $P(\alpha,\infty)$, and each summand of the form $\Gamma^{\beta}\mathbb{Z}_2[t]/(t^n)$ is isomorphic to $P(\beta,\beta+n)$.

We can visualize this decomposition as either a *persistence barcode* [CAR+05], with each interval plotted from its birth to death in parallel, or a *persistence diagram* [ELZ02] which plots each interval as a single point whose x-value is birth and y-value is death. We color the points or bars based on the degree of the homology group they represent, and let $D_k(\mathcal{F})$ denote the persistence diagram constructed by taking the k-th degree homology on our simplicial filtration.

A persistence diagram and persistence barcode are given for a point cloud which could hypothetically have been sampled from an annulus in Figure 1.4. Represented in red are the H_0 class representatives, all born when the filtration is initialized, and as the radius of the Vietoris-Rips complex increases, more connected components merge together until there is a single connected component left. This is represented with the longest red bar in the persistence barcode or red point on the ∞-line in the persistence diagram. The 1-cycle representatives of H_1 are given in blue. From the point cloud we see two holes, and this structure is seen in the persistence barcode as the two longer bars. For the persistence diagram large cycles are represented by the points further from the diagonal. Points in a persistence diagram close to the diagonal (or short bars in a persistence barcode) are typically considered noise, though a threshold to define noise is only heuristic.

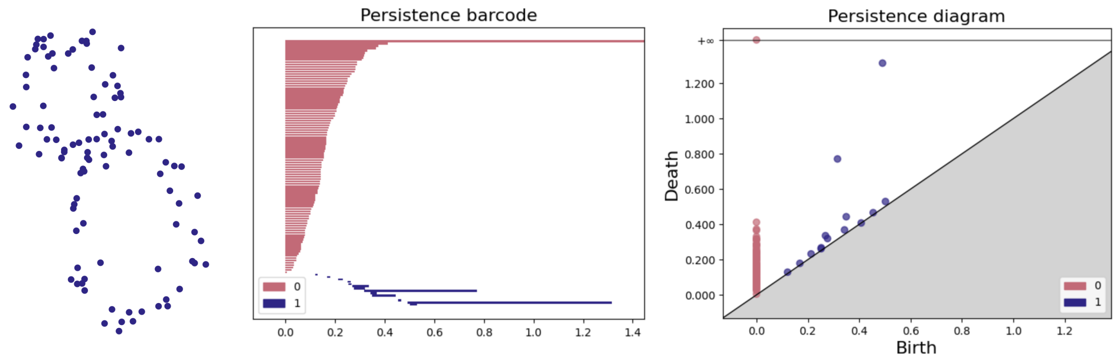

Figure 1.4. The persistence barcode (center) and persistence diagram (right) of a point cloud (left).

These representations of persistent homology have been useful tools in many applications, including classification tasks through image analysis. For example, in Figure 1.5, the persistent homology is computed for CT scans of both non-infected and COVID-19-infected patients [Iqb+21]. Using the differences between the persistence diagrams for H_1 and H_2, a classifier is built to predict infection based on a patient's CT scan, outperforming traditional image classifiers including ones given by machine learning.

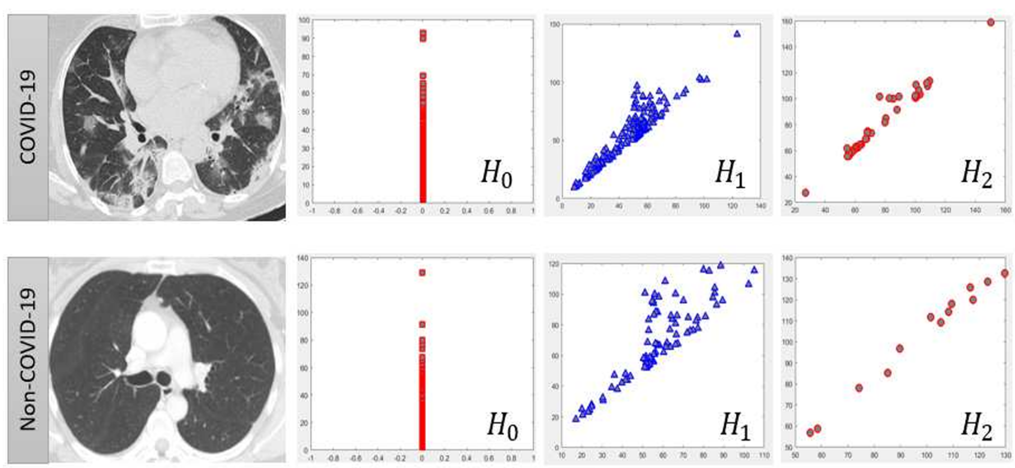

Figure 1.5. COVID-19-infected (top) and non-infected (bottom) CT scans and their computed persistence diagrams in degrees zero through two [Iqb+21, Figure 11]. The classification of COVID-19 using CT scans outperformed traditional machine learning models, including ResNet101 [Iqb+21, Table 1].

As data is often noisy, stability of any invariant applied to data must be analyzed. A significant proportion of literature [CEH07; Cha+08] has produced a wide variety of stability theorems for persistence homology. We highlight two such theorems, first defining a needed metric on persistence diagrams.

Definition 1.6 (Bottleneck Distance [CV21a])**.** Let $D_k(\mathcal{F})$ and $D_k(\mathcal{G})$ be two degree-k persistence diagrams. A bijection between them is a matching of each of their points to each other or to points in the diagonal $\{(x, x) \colon x \in \mathbb{R}_+\}$. Let $\Pi = \{\pi \colon D_k(\mathcal{F}) \to D_k(\mathcal{G})\}$ be the set of all such bijections. Define the *bottleneck distance* as

$$d_B(D_k(\mathcal{F}), D_k(\mathcal{G})) = \inf_{\pi \in \Pi} \sup_{x \in D_k(\mathcal{F})} ||x - \pi(x)||_\infty.$$

That is, given a matching between points in a persistence diagram, which can include matching to the diagonal which are needed when the cardinalities differ, the distance between degree k diagrams reflects the worst fit. But we take the smallest of such distances over all matchings. The bottleneck distance can only be computed between two diagrams of the same degree, however, it is common to take the cumulative sum of bottleneck distances over all degrees as an agglomerate statistic. The bottleneck distance is also a specific case ($p = \infty$) of the *Wasserstein distance.*

Under this metric there is a stability result. Each simplicial filtration $\mathcal{F}$, if constructed from filtration of Čech or Vietoris-Rips complexes, has an associated input function $f \colon X \to \mathbb{R}$. This input function has value $f(\sigma) = r_j$ for every simplex $\sigma \in K_{r_j} \backslash K_{r_{j+1}}$ for $K_{r_j} \neq K_{r_{j+1}}$, and f is a monotone simplicial map, meaning for $\sigma \subseteq \tau$ we have $f(\sigma) \leq f(\tau)$. Further, we can construct a simplicial filtration from any given monotone simplicial map f. For persistence diagrams constructed from simplicial filtrations, bottleneck distance is well-behaved.

Theorem 1.7 ([CEH07])**.** *Let $\mathcal{F}$ and $\mathcal{G}$ be two simplicial filtrations on a finite simplicial complex X constructed from simplicial maps $f, g \colon X \to \mathbb{R}$. Then for every $k \geq 0$,*

$$d_B(D_k(\mathcal{F}), D_k(\mathcal{G})) \leq ||f - g||_\infty.$$

This theorem only pertains to filtrations on a single simplicial complex X, but there are extensions of this theorem to the level of general finite metric spaces (or data sets) where we achieve stability through *Gromov-Hausdorff distance.*

Definition 1.8. Let (X, d_X) and (Y, d_Y) be compact metric spaces. The *Gromov-Hausdorff* distance is defined

$$d_{GH}(X,Y) = \frac{1}{2} \inf_{C \in \mathcal{C}} \sup_{(x_1,y_1),(x_2,y_2) \in C} |d_X(x_1,x_2) - d_Y(y_1,y_2)|,$$

where $\mathcal{C}$ is the set of *correspondences* between X and Y.

Similar to the bottleneck distance, Gromov-Hausdorff distance measures the dissimilarity between two metric spaces through the difference in distances between points in correspondence with one another. Thus the following result reveals that the bottleneck distance between two persistence diagrams will always be bounded by this dissimilarity on the original metric spaces.

Theorem 1.9 ([CV21a])**.** *Let (X, d_X) and (Y, d_Y) be finite compact metric spaces. Then for any fixed $k \geq 0$ we have*

$$d_b(D_k(VR(X)), D_k(VR(Y))) \leq d_{GH}(X,Y),$$

where $VR(X)$, $VR(Y)$ are Vietoris-Rips filtrations on X and Y, and d_{GH} is the Gromov-Hausdorff distance.

1.4 Multiscale Mapper

While Mapper has been successful as a tool in exploratory data analysis, it is often difficult to choose from the wide range of parameters. Moreover, it is unstable with respect to its many parameter choices, which makes it not suitable for inference. Small perturbations in choice of any of the parameters (reference function, cover, or clustering algorithm) can change the Mapper graph's topological structure. Carrière, Michel, and Oudot have shown statistical guarantees for topological properties of a Mapper graph under certain parameter choices [CMO17]. But the resulting constraints seem too conservative for most practical settings. Moreover, unlike persistence homology, there are no generally adopted metrics for comparing Mapper graphs. To develop an alternative notion of stability, Dey, Mémoli, and Wang (2015) set the theoretical foundations for *Multiscale Mapper* [DMW], producing a filtration of Mapper graphs.

This work is the theoretical inspiration for our implementation of *Multiscale 2-Mapper* in Chapter 3. We will use these ideas to integrate Mapper and persistent homology [Car09]. As we will see, using Multiscale Mapper, one can observe the structure of a Mapper graph across an entire filtration of parameter choices which can assist with understanding the underlying topological structure of a data set. We have made the first implementation of Multiscale Mapper, but before describing that work we cover the theoretical background and address some caveats.

While we defined Mapper graphs as constructed from data sets, Dey, Mémoli, and Wang generalize the Mapper graph to one constructed from topological spaces and *well-behaved* maps, $f\colon X \to Z$, between them. By well-behaved, we mean that for every path connected open set $U \subseteq Z$, the preimage $f^{-1}(U)$ has finitely many open path connected components.

Definition 1.10 (Generalized Mapper [DMW])**.** Let X and Z be topological spaces and let $f\colon X \to Z$ be a well-behaved continuous map. Let $\mathcal{U} = \{U_\alpha\}_{\alpha\in A}$ be a finite open cover of Z. The *Mapper complex* is then the nerve of the simplicial complex of the pullback cover: $\mathrm{M}(\mathcal{U}, f) := \mathcal{N}(f^*(\mathcal{U}))$.

The construction of Multiscale Mapper is through a filtration of covers, also called a *tower of covers*. For those familiar with category theory, a tower is a functor from the real numbers considered as a poset category. Explicitly, we have the following.

Definition 1.11 (Tower [DMW])**.** A *tower* $\mathcal{W}$ with *resolution* $s \in \mathbb{R}$ is a collection of objects $\mathcal{W} = \{W_\varepsilon\}_{\varepsilon\geq s}$ together with maps $w_{\varepsilon,\delta}\colon W_\varepsilon \to W_\delta$ so that $w_{\varepsilon,\varepsilon} = \mathrm{Id}_{W_\varepsilon}$ and $w_{\varepsilon'',\varepsilon'} \circ w_{\varepsilon',\varepsilon} = w_{\varepsilon'',\varepsilon}$ for all $s \leq \varepsilon \leq \varepsilon' \leq \varepsilon''$. We write $\mathrm{res}(\mathcal{W})$ to denote the resolution of the tower.

A tower of covers is constructed so that the Multiscale Mapper is a simplicial filtration. Implementation is straightforward, as one need only to compute the Mapper graph for each cover in the tower of covers. The map from each Mapper graph to the next in the filtration is induced by the maps between covers.

Definition 1.12 (Multiscale Mapper [DMW])**.** Let X and Z be topological spaces and $f\colon X \to Z$ be a well-behaved continuous map. Let $\mathcal{U}$ be a tower of covers of Z. Then, the Multiscale Mapper is defined to be the tower of simplicial complexes defined by the nerve of the pullback:

$$\mathrm{MM}(\mathcal{U}, f) := \mathcal{N}(f^*(\mathcal{U})). \tag{1.2}$$

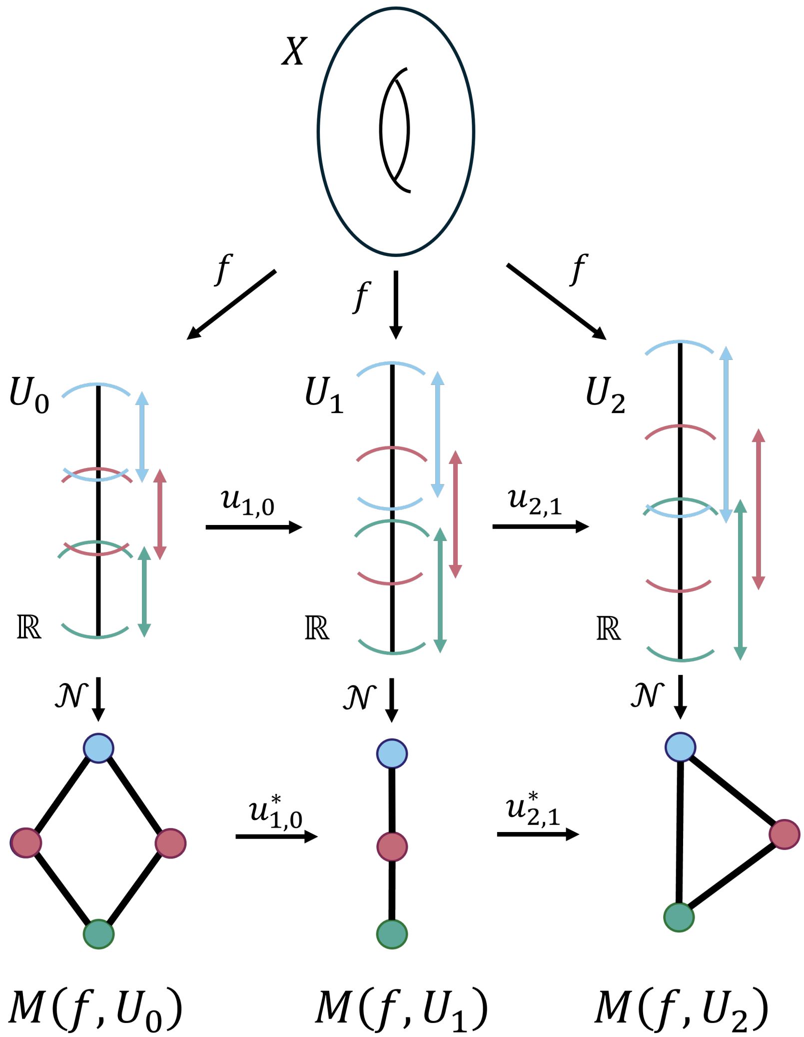

Figure 1.6. We construct the Multiscale Mapper, $\mathrm{MM}(f, \mathcal{U})$, for a torus X with height based reference function $f\colon X \to \mathbb{R}$ and tower of covers $\mathcal{U} = \{U_i\}_{0 \leq i \leq 2}$. In the middle row we have the tower of covers over $\mathbb{R}$ where we color each cover set individually so that they are easily matched with their respective vertices in each Mapper graph in the bottom row.

For a finite sequence $\mathrm{res}(\mathcal{U}) \leq \varepsilon_1 \leq \cdots \leq \varepsilon_k$, then Multiscale Mapper yields a sequence of simplicial complexes connected by simplicial maps

$$\mathcal{N}(f^*(U_{\varepsilon_1})) \overset{u^*_{\varepsilon_2,\varepsilon_1}}{\to} \mathcal{N}(f^*(U_{\varepsilon_2})) \overset{u^*_{\varepsilon_3,\varepsilon_2}}{\to} \cdots \overset{u^*_{\varepsilon_k,\varepsilon_{k-1}}}{\to} \mathcal{N}(f^*(U_{\varepsilon_k}))$$

Applying the homology functor H_* with binary coefficients, we can compute the persistence homology,

$$H_*(\mathcal{N}(f^*(U_{\varepsilon_1}))) \to H_*(\mathcal{N}(f^*(U_{\varepsilon_2}))) \to \cdots \to H_*(\mathcal{N}(f^*(U_{\varepsilon_k}))), \tag{1.3}$$

and represent the cycles that persist across this filtration as persistence diagrams and barcodes.

As an example, presented in Figure 1.6, we compute Multiscale Mapper on data sampled from a torus X. Choose a real-valued reference function $f\colon X \to \mathbb{R}$ to be the height function. Our tower of covers $\mathcal{U}$ all consist of three consecutive intervals all with various amounts of overlap. Each open interval of U_i includes into U_{i+1}, for each i, and each interval is denoted by colors (red, green, or blue) to visualize this inclusion. Multiscale Mapper consists of a Mapper graph on each cover in the tower, shown on the bottom row of Figure 1.6, with induced maps between each pair of Mapper graphs derived from the maps on the tower of covers $\mathcal{U}$. As the intervals in each cover increase we see the death of a cycle at U_1 and a birth of a new cycle at U_2.

While theoretically this example is clear, it is not immediate what to do with data sampled from a torus or other manifold rather than the manifold itself. We develop an approach and give the first implementation of Multiscale Mapper in this thesis.

The Multiscale Mapper construction allows for the comparison of filtrations of Mapper graphs through their persistence diagrams. In particular, for two interleaving towers of covers we can bound the bottleneck distance between their respective Multiscale Mappers via Corollary 1.14.

Definition 1.13 (Multiplicative Interleaving Towers of Covers [DMW])**.** Let $\mathcal{U} = \{U_\varepsilon\}$ and $\mathcal{V} = \{V_\varepsilon\}$ be two towers of covers of a topological space X such that their resolutions $\mathrm{res}(\mathcal{U}) = \mathrm{res}(\mathcal{V}) = r$. Given $\eta \geq 0$ we say that $\mathcal{U}$ and $\mathcal{V}$ are *η-multiplicatively interleaved* if one can find maps of covers $\zeta_\varepsilon\colon U_\varepsilon \to V_{\varepsilon\eta}$ and $\xi_{\varepsilon'}\colon V_{\varepsilon'} \to U_{\varepsilon'\eta}$ for all $\varepsilon, \varepsilon' \geq r$.

Corollary 1.14 ([DMW])**.** *For $\eta \geq 0$, let $\mathcal{U}$, $\mathcal{V}$ be two finite towers of covers of compact metric space Z with $\mathrm{res}(\mathcal{U}) = \mathrm{res}(\mathcal{V}) > 0$. Let $f\colon X \to Z$ be continuous and $\mathcal{U}$ and $\mathcal{V}$ be η-interleaved. Then $\mathrm{MM}(\mathcal{U}, f)$ and $\mathrm{MM}(\mathcal{V}, f)$ are η-interleaved. In particular, the bottleneck distance between the persistence diagrams $D_k\,\mathrm{MM}(\mathcal{U}, f)$ and $D_k\,\mathrm{MM}(\mathcal{V}, f)$ is at most η for all $k \in \mathbb{N}$.*

Dey, Mémoli, and Wang additionally show stability for Multiscale Mapper constructed with a *good tower of covers.*

Definition 1.15 (Good Towers of Covers [DMW])**.** Let $c \geq 1$ and $s > 0$. We say that a finite tower of covers $\mathcal{W} = \{W_\varepsilon\}$ for the compact metric space (Z, d_Z) is (c,s)-*good* if

(i) $\text{res}(\mathcal{W}) = s$ and $s \leq \text{diam}(Z)$;

(ii) $\text{diam}(W) \leq \varepsilon$ for all $W \in W_\varepsilon$ and all $\varepsilon \geq s$;

(iii) for all $O \subset Z$ with $\text{diam}(O) \geq s$, there exists $W \in W_{c \cdot \text{diam}(O)}$ such that $O \subseteq W$.

These towers of covers are a special family of covers that are used to guarantee stability for Multiscale Mapper under perturbations of both cover and reference function. As an example, $\mathcal{U} = \{U_\varepsilon\}_{\varepsilon \geq s}$ with $U_\varepsilon = \{B_{\varepsilon/2}(z) \colon z \in Z\}$ is a $(2,s)$-good tower of covers. In particular, Multiscale Mappers built from good towers of covers interleave with one another.

Theorem 1.16 ([DMW])**.** *Given a map $f \colon X \to Z$ let $\mathcal{V} = \{V_\varepsilon\}$ and $\mathcal{W} = \{W_\varepsilon\}$ be two (c,s)-good towers of covers of Z. Then the corresponding Multiscale Mappers* $\text{MM}(\mathcal{V}, f)$ *and* $\text{MM}(\mathcal{W}, f)$ *are c-multiplicatively interleaved.*

The authors additionally discuss heuristics for constructing a good tower of covers. In particular, they remark that towers of covers constructed from expanding balls centered on a lattice are $(3,s)$-good [DMW, Appendix B.2.1]. They describe this type of covers as an *ϵ-net*, however we will refer to this in Chapter 2 as a *lattice cover.*

Dey, Mémoli, and Wang followed up on their work in 2017 by analyzing the relationships between the homology groups of a topological space X and it's Multiscale Mapper $\text{MM}(f, \mathcal{U})$ [DMW17]. Their main result shows the one-dimensional homology of the Multiscale Mapper is "dominated by" that of the original space.

Theorem 1.17 ([DMW17])**.** *Assume that X and Z are compact topological spaces with continuous map $f \colon X \to Z$ between them. For a path connected tower of covers $\mathcal{U} = \{U_\varepsilon\}_{\varepsilon \geq s}$ on Z the Multiscale Mapper is defined*

$$\mathcal{N}(f^*(U_{\varepsilon_1})) \to \mathcal{N}(f^*(U_{\varepsilon_2})) \to \cdots \to \mathcal{N}(f^*(U_{\varepsilon_k}))$$

Then there exists a surjection from $H_1(X) \to H_1(\mathcal{N}(f^*(U_{\varepsilon_j})))$ *for each* $1 \leq j \leq k$. *Furthermore, all connecting maps of the* H_1*-persistence module*

$$H_1(\mathcal{N}(f^*(U_{\varepsilon_1}))) \to H_1(\mathcal{N}(f^*(U_{\varepsilon_2}))) \to \cdots \to H_1(\mathcal{N}(f^*(U_{\varepsilon_k}))),$$

are surjections.

While this is a nice theoretical result, it does not apply to data sets. Much of their result relies on the path-connectedness of the cover, however this property is lost when applying this cover on a data set. For example, for a sparse sampling on the torus, as in Figure 2.2, we see it's Mapper graphs (center and center right) are noisy and H_1 of either of these are certainly not at most $\mathbb{Z}_2^2$. To apply Multiscale Mapper in the context of data, we will need to adapt their ideas to the discrete setting, which will require carefully addressing some parameter choices.

In this thesis we lay both the theoretical and implementation foundations for Multiscale 2-Mapper, a tool for analyzing data with connectivity information chosen by persistence. We are beginning to apply it, in particular to climate data, as we share here and explain further in Chapter 4.

CHAPTER 2

2-MAPPER

The Mapper algorithm is a tool for understand clusters and relationships in high-dimensional data sets. It can be applied for data sets of any size, in particular when there is not enough data for training a machine learning model. As discussed in Chapter 1, Mapper requires choices in reference function, cover, and clustering algorithm. Reference (or *lens*) functions to the real numbers are the most common, but functions to any space are theoretically allowed, and implementations typically accommodate $\mathbb{R}^n$-valued lens functions. Indeed, there are several published applications of mapper that use higher dimensional reference functions, [NLC11; Sag+18; SBH20; CV21b; Hus+22].

General reference functions produce higher dimensional nerves, so there is information loss in only considering the 1-skeleton. We introduce a new mapper object, *2-Mapper*, which represents the two-skeleton of the nerve of the mapper cover, developing a Python implementation to display the output. This is similar to the original Mapper construction [SMC07] which defines the Mapper complex as the simplicial complex derived from the nerve of clusters. In our implementation, derived from `giotto-tda` [Tau+20], we develop additional cover structures, both because different geometries can be more or less useful in different contexts and with an eye towards employing persistence. In each section we include details of our Python implementation.

2.1 2-Mapper

Recall that the Mapper graph defined above is the 1-skeleton of the nerve of the cover defined by clustered subsets of a data set, defined with some choice lens function, cover, and clustering algorithm. In Definition 1.1 we spelled this out explicitly. For the same parameter choices, we now consider the 2-skeleton of the nerve of clusters. We call this construction the *2-Mapper complex*.

Definition 2.1 (2-Mapper)**.** Let $X \subseteq \mathbb{R}^d$ be a data set. For choice of continuous reference function $f : X \to \mathbb{R}^m$, cover $\mathcal{U}$ of the image $f(X)$, and clustering algorithm, we define the *2-Mapper complex* of X as the simplicial complex $M_2(f, \mathcal{U}) = \mathcal{N}^2(f^*(\mathcal{U}))$.

Concretely, 2-Mapper distinguishes between two cases of triangles in a Mapper graph. Such a triangle is "filled in" when it corresponds to a triple of nodes C_i, C_j, C_k that have $C_i \cap C_j \cap C_k \neq \emptyset$. A triangle "remains empty" when while there are pairwise non-empty intersections, the triple intersection $C_i \cap C_j \cap C_k$ is empty.

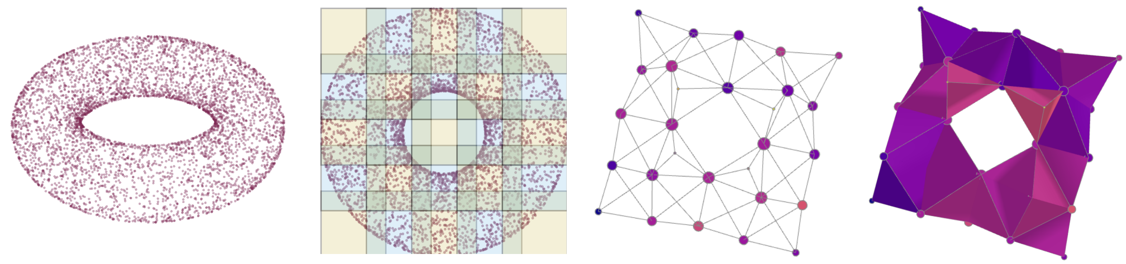

Figure 2.1. Left to Right: A point cloud (5000 points) of the torus in $\mathbb{R}^3$. Compute it's Mapper graph (center right) and 2-Mapper complex (right) by projecting to $\mathbb{R}^2$ using coordinate projection reference function and cubical cover with 5 intervals and 0.3 overlap fraction (center left).

Practically speaking, triple overlaps occur for most covers of $\mathbb{R}^m$ when $m \geq 2$, in particular using the default cover of hypercubes. For real valued lens functions $f : X \to \mathbb{R}$ and the default interval cover (1-dimensional hypercubes), it is possible to obtain a non-trivial 2-Mapper, however it requires that the overlap fraction g for consecutive intervals exceed 0.5, as we show in Proposition 2.4, and the overlap information is redundant. It is uncommon to choose $g > 0.5$ as it results in a surplus of extra connections within the Mapper graph [CMO17; Alv+24].

For well-connected Mapper graphs, as in Figure 2.1, the 2-simplices add a clarifying visual of a surface around cycles. While we could conjecture that Mapper graph (center right) likely just has one cycle at it's center, 2-Mapper (right) verifies this, and also clarifies the existence of a second cycle. Indeed, the addition of 2-simplices allows us to compute Betti-1 in a meaningful way. This is similar to the original construction of Mapper [SMC07, Section 5.2] in which they compute the Betti numbers of a torus data set. In the case of more sparse data sets, like in Figure 2.2, we have more evidence of topological features in 2-Mapper. We will gain substantial insight into such features when we also incorporate persistence below.

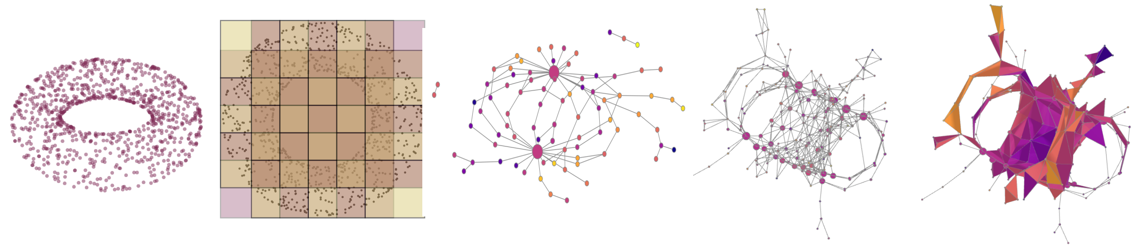

Figure 2.2. Left to Right: A sparse (1000 points) point cloud of the torus in $\mathbb{R}^3$. Compute it's Mapper graph (center right) and 2-Mapper complex (right) graphs by projecting to $\mathbb{R}^2$ using coordinate projection and cubical cover with 6 intervals and 0.5 overlap fraction (middle left). For reference we compute the Mapper graph (center) with height reference function with 6 interval cover with 0.5 overlap fraction.

2.2 Choices of Cover

Mapper is heavily dependent on the cover choice, and is unstable to perturbation of the cover [CMO17]. In this section we detail the construction of various covers including the new *lattice cover* for 2-Mapper, and discuss implementation in Python. For practical purposes, the distinction between closed or open covers does not matter. Whether or not we construct a cover open or closed is dependent on its computational efficiency. For the purposes of this section, we define all hypercube covers (*cubical cover* and *cubical lattice cover*) as closed and all ball covers (A_n^*-*lattice covers*) as open. Looking ahead, our main results are notions of persistence for hyperparameters, and so our covers are constructed with each cover set's center point specified. This is to guarantee containment among cover sets from nested covers in a tower of covers – see Section 3.2.

All Mapper software in existing TDA libraries [Tau+20; Pro24; Vee+20] only use a single cover type for their Mapper algorithms, namely the *cubical cover* or the *standard cover*, which we give in Definition 2.3. The cubical cover is used primarily because of its computational efficiency and familiarity. But this cover has more than intersections than are necessary for creating 2-simplices for 2-Mapper. For example, in Figure 2.3 the cubical cover in $\mathbb{R}^2$ (with $g < 0.5$) can only create 2-simplices within a four-fold intersection. This creates two pairs of 2-simplices which decompose a square, yielding a redundant visualization. If we a different cover configuration, like balls centered at points on the A_2^*-lattice in Figure 2.3, we can optimize the placement of 2-simplices for 2-Mapper.

To compute our covers for any data set X, we first need to consider bounds of our data.

Definition 2.2 (Bounding Box)**.** Let $X \subseteq \mathbb{R}^n$ be a compact topological space. Define the *bounding box* of X as $\prod_{i=1}^{n}[m_i, M_i]$ where $m_i = \min_{x\in X} \pi_i(x)$ and $M_i = \max_{x\in X} \pi_i(x)$ for coordinate projections π_i and $1 \leq i \leq n$.

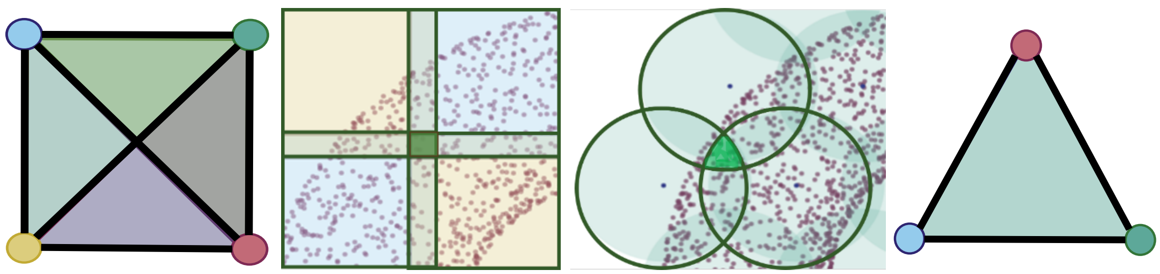

Figure 2.3. For a subset of data in $\mathbb{R}^2$, we cover that data with a cubical cover (center left) and A_2^*-lattice cover (center right). The cubical cover only has either two-fold or four-fold intersections. Its four-fold intersection is highlighted in green, and its resulting nerve contains four 2-simplices (left). The A_2^*-lattice cover with generating matrix $M_{A_2^*}$ only has at most a three-fold intersection and its nerve has only one 2-simplex (right).

Cubical Cover

For the definition of the cubical cover, we will work in the generality of compact topological spaces, which thus includes data sets.

Definition 2.3 (Cubical Cover)**.** Let $Z \subset \mathbb{R}^n$ be a compact topological space with bounding box $B = \prod_{i=1}^{n}[m_i, M_i]$. Define the half-unit $k \times k$ lattice within B as that centered at points of the form $\prod m_i + j_i \frac{M_i - m_i}{k}$ as the j_i's range over n-tuples of half integers between 0 and k.

The *cubical cover* $\mathcal{U}$ of Z constructed with k-intervals and overlap fraction g is the cover of boxes $\mathcal{U} = \{U_\alpha\}_{\alpha\in A}$ which are centered at the half-unit $k \times k$ lattices with lengths

$$l_i = \frac{M_i - m_i}{k - (k-1)g}.$$

The implementation of this cover can be found in [Tau+20; Pro24; Zho+21; Alv+24; Vee+20]. Because 2-Mapper is built from `giotto-tda`, we use their cubical cover construction.

For $n = 1$, the cubical cover over $\mathbb{R}$ is just a cover of uniform consecutive intervals, where each can be written $U_i = [L_i, R_i]$. For a bounding box $B = [m, M]$ over a compact topological space Z, the first interval is constructed as $U_1 = [m, m + l]$ where the interval length is

$$l = \frac{M - m}{k - (k-1)g}.$$

Each pair of consecutive intervals have an overlap proportion of g, so the distance between left endpoints is $|L_i - L_{i+1}| = (1-g)l$. Thus we can build the second interval as $U_2 = [m + (1-g)l, m + (1-g)l + l]$, and in general

$$U_i = [m + (i-1)(1-g)l, m + (i-1)(1-g)l + l]$$

We can also express the general cover set through a fixed center point

$$U_i = \left[c_i - \frac{l}{2}, c_i + \frac{l}{2}\right], \tag{2.1}$$

where $c_i = m_1 + (i-1)(1-g)l + \frac{1}{2}l$. We can create a cubical cover over $\mathbb{R}^n$ by performing this constructing in each dimension separately and then taking the product.

We emphasize through the following proposition that in the $n = 1$ case there are only "uninteresting" 2-Mapper complexes.

Proposition 2.4. *Let $Z \subset \mathbb{R}$ be a compact topological space with cubical cover $\mathcal{U}$, and $f = \mathrm{Id}_{\mathbb{R}}$. If the 2-Mapper, $M_2(f, \mathcal{U})$, constructed from Z contains a 2-simplex, then $k \geq 3$ and $g > 0.5$.*

Proof. Let $B = [m, M]$ be the bounding box of Z. Then we construct the cubical cover over Z as follows from `giotto-tda` [Tau+20]. Let each interval $U_i \in \mathcal{U}$ be defined as above with $U_i = [m + (i-1)(1-g)l, m + (i-1)(1-g)l + l]$, and suppose $M_2(f, \mathcal{U})$ contains a 2-simplex. Then, as stated previously, this would require $U_i \cap U_{i+1} \cap U_{i+2} \neq \emptyset$. Thus $k \geq 3$, and without loss of generality, suppose $U_1 \cap U_2 \cap U_3 \neq \emptyset$. Then

$$\bigcap_{i=1}^{3} U_i = \bigcap_{i=1}^{3} [L_i, R_i] = [L_3, R_1] = [m + 2(1-g)l, m + l]$$

so $L_3 < R_1$. Expanding this we have

$$\begin{aligned} m + 2(1-g)l &< m + l, \\ 2(1-g) &< 1, \\ 1 - g &< \frac{1}{2}, \\ \frac{1}{2} &< g. \end{aligned}$$

$\square$

For $n \geq 2$, a 2-Mapper complex created with a cubical cover over Z can contain 2-simplices for any $g \in (0,1)$ and $k \geq 2$. In each dimension there are pairs of consecutive intervals which intersect, meaning portions of the cover contain have 2^n cover sets intersecting, as in Figure 2.3.

Lattice Cover

In addition to the cubical cover, we provide a new cover called a *lattice cover*. This is similarly defined in [DMW] as an ϵ-*net* where the point sample P is a subset of a lattice Λ. We expand on this concept and implement this cover for 2-Mapper. Information about lattices and their properties can be found in Appendix A.

The first lattice that would be useful for a 2-Mapper construction would be the integer lattice $\mathbb{Z}^n$, due to its similarity to the cubical cover. We will call this cover the $\mathbb{Z}^n$-lattice cover, or *cubical lattice cover*, and construct it as follows. The generating matrix for $\mathbb{Z}^n$ is the identity matrix I^n. We want to construct a cover of hypercubes over $\mathbb{Z}^n$ with identical parameters k-intervals and overlap fraction g. In the construction of the cubical cover, the bounding box is subdivided into $\frac{M_i - m_i}{k}$ parts for each dimension i. To mimic this, we will scale $\mathbb{Z}^n$ by $c = \max_{1 \leq i \leq n} \frac{1}{k}|M_i - m_i|$ and build intervals centered at each lattice point in $c\mathbb{Z}^n$.

The construction of the cubical lattice cover, while similar to the cubical cover, is not equivalent. The difference is in side lengths of each dimension for the cover sets. The cubical lattice has uniform side lengths in all dimensions, while in the (standard) cubical cover each dimension of a cover set is scaled independently based on the bounding box. This difference in construction means that while the Mapper graphs constructed from either cover could look similar, there are no guarantee that two such Mapper graphs would even be homotopy equivalent. We go into more on the comparison of these two cover types in Chapter 4.

For simplicity, assume that $n = 1$ and we consider the integer lattice $\mathbb{Z}$ over $\mathbb{R}$. For each lattice point $\xi \in \mathbb{Z}$, we define the interval as $U_\xi = [c(\xi - \varepsilon), c(\xi + \varepsilon)]$, for some $\varepsilon > 0$. The cover should also have the consecutive intervals overlap with proportion gl where l is the interval length of each cover set. This allows us to directly compute ε as follows. Let $U_1 = [c(\xi_1 - \varepsilon), c(\xi_1 + \varepsilon)]$ and $U_2 = [c(\xi_2 - \varepsilon), c(\xi_2 + \varepsilon)]$, be consecutive intervals. Then $\xi_2 = \xi_1 + 1$, and so the overlap proportion between U_1 and U_2 is proportional to the interval length l by overlap fraction g. This means $gl = g(2\varepsilon) = U_1 \cap U_2$, and so

$$\begin{aligned}
2gc\varepsilon = U_1 \cap U_2 &= c(\xi_1 + \varepsilon) - c(\xi_2 - \varepsilon), \\
&= c(\xi_1 + \varepsilon) - c(\xi_1 + 1 - \varepsilon), \\
&= 2c\varepsilon - c,
\end{aligned}$$

implying that $\varepsilon = \frac{1}{2(1-g)}$, and the length of each interval is $l = 2c\varepsilon = \frac{c}{1-g}$. With this we can now formally define the cubical lattice cover.

Definition 2.5 (Cubical Lattice Cover)**.** Let $Z \subseteq \mathbb{R}^n$ be compact with bounding box $B = \prod_{i=1}^n [m_i, M_i]$. We define the *cubical lattice cover* over integer lattice $\mathbb{Z}^n$ with k-intervals and overlap fraction g as the cover $\mathcal{U} = \{U_\xi\}_{c\xi \in B \cap \mathbb{Z}^n}$ whose cover sets U_ξ are hypercubes defined

$$U_\xi = \prod_{i=1}^n \left[c\left(\xi_i - \frac{1}{2(1-g)}\right), c\left(\xi_i + \frac{1}{2(1-g)}\right)\right],$$

where $c = \max_{1 \le i \le n} \frac{1}{k}|M_i - m_i|$.

For the other lattice constructions, we instead focus on covers constructed as ϵ-nets as in [DMW]. For consistency across covers, we use the same parameters, namely k as the "number of intervals" into which each coordinate is divided and g as the fraction of overlap. For a general lattice cover, each cover set will a ball centered at each lattice point in the scaled lattice.

Definition 2.6 (Lattice Cover)**.** Let $Z \subseteq \mathbb{R}^n$ be compact with bounding box $B = \prod_{i=1}^n [m_i, M_i]$, and let Λ be a root lattice with generating matrix M and covering radius R – see Appendix A. We define the Λ*-lattice cover* with k-intervals and overlap fraction g as $\mathcal{U} = \{B(c\xi M, \varepsilon)\}_{c\xi M \in B \cap \mathbb{Z}^n}$ where $c = \max_{1 \le i \le n} \frac{M_i - m_i}{k}$ and $\varepsilon = cR(1 + g)$.

We focus solely on the dual root lattice A_n^* since, for $1 \leq n \leq 5$, it is the thinnest covering [Con+13, Chapter 2, Section 1.3]. A_n^* is also the thinnest *known* covering for up to $n \leq 23$. This means that the balls centered around points in an A_n^*-lattice cover have the smallest intersectional volume among all other lattice covers. The dual lattice A_n^* has a general generating matrix which embeds the lattice $\mathbb{R}^{n+1}$:

$$M_{A_n^*} = \begin{bmatrix} 1 & -1 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & -1 & 0 & \cdots & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdots & \cdot & \cdot \\ 1 & 0 & 0 & 0 & \cdots & -1 & 0 \\ \frac{-n}{n+1} & \frac{1}{n+1} & \frac{1}{n+1} & \frac{1}{n+1} & \cdots & \frac{1}{n+1} & \frac{1}{n+1} \end{bmatrix}.$$

There are special generating matrices for $n = 2, 3$ due to the isomorphisms $A_2^* \cong A_2$ and $A_3^* \cong D_3^*$:

$$M_{A_2^*} = \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}, \qquad M_{A_3^*} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 1 & 1 & 1 \end{bmatrix}.$$

We implement lattice covers for A_n^* with the three above generating matrices, compatible with `giotto-tda` Mapper pipelines. In Figure 2.4, we can see the geometry of the 2-Mapper complex is dependent on cover choice. We go into more depth on resulting 2-Mapper complexes with respect to cover choice in Chapter 4.

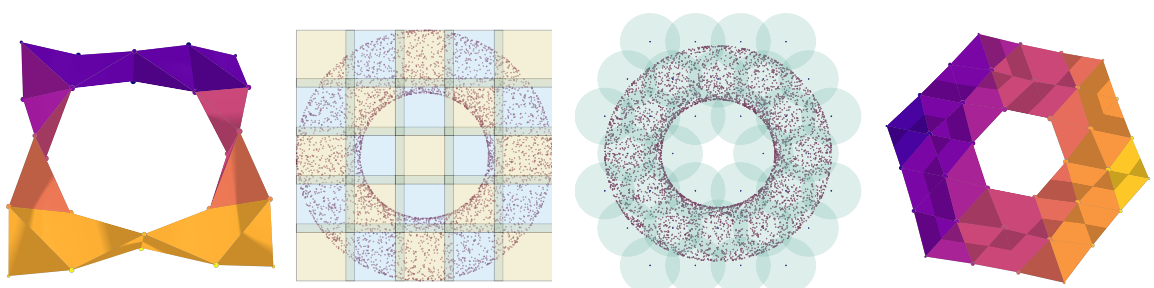

Figure 2.4. 2-Mapper complexes (left, right) of a torus point cloud produced from cubical cover (middle left) and A_2^*-lattice cover (middle right), respectively. Both 2-Mapper complexes were computed with coordinate projections to $\mathbb{R}^2$ as the lens and DBSCAN as the clustering algorithm.

Lattices are regularly implemented in computational mathematics [Con+13]. We develop them here as needed for both theory and practical work with 2-Mapper.

Let $X \subset \mathbb{R}^n$ be a data set with bounding box B, and for simplicity let $f = \mathrm{Id}\colon \mathbb{R}^n \to \mathbb{R}^n$. For each cover, we continue to use the number of subdivisions (intervals) in each coordinate and the percentage overlap as our parameters. Then we construct the lattice covers for these four cases ($\mathbb{Z}^n$, A_n^*, A_2^*, and A_3^*) as in Definitions 2.5 and 2.6.

Recall in Section 2.2, that to compute a lattice cover over X we choose a lattice Λ with generating matrix M and parameters k representing the "number of intervals" and overlap fraction g. With these parameters, the lattice cover is a subset of the scaled lattice $c\Lambda$ that covers the bounding box. To maintain computational efficiency, we only need to consider points in the lattice that are contained in the bounding box B. As each point in the scaled lattice can be written $c\xi M$ for $\xi \in \mathbb{Z}^n$, we can compute explicit bounds for the points ξ we should consider while constructing each lattice cover. The following results provide the necessary bounds on ξ are required to produce a lattice cover over X. We additionally provide the size of each lattice cover for comparison with one another. We will begin this analysis with the cubical lattice cover, and then move on to the A_n^*-lattice covers.

Recall our notation for a bounding box, namely $B = \prod_{i=1}^{n}[m_i, M_i]$, where $m_i = \min_{x\in X} \pi_i(x)$ and $M_i = \max_{x\in X} \pi_i(x)$ for coordinate projections π_i for all $1 \leq i \leq n$. Let $\vec{m} = (m_1, ..., m_n)$ and $\vec{M} = (M_1, ..., M_n)$ as the minimum and maximum points on B. For a lattice point $c\xi M \in c\Lambda$, to be considered as a center point in a lattice cover we must have $c\xi M \in B$. This can simply be expressed as

$$\vec{m} \leq c\xi M \leq \vec{M}, \tag{2.2}$$

where the inequality holds entry-wise, ie. for every $1 \leq i \leq n$

$$m_i \leq (c\xi M)_i \leq M_i.$$

From this expression we can explicitly compute bounds for each entry ξ_i, and this provides efficient ranges for $\xi \in \mathbb{Z}^n$ when computing the lattice cover for our 2-Mapper implementation. We begin with the simplest case, the cubical lattice cover, since its lattice is a scaled integer lattice.

Proposition 2.7. *Let $\mathcal{U}$ be the cubical lattice cover constructed from k-intervals and overlap fraction g over the data set X with bounding box B. Then each lattice point in the cubical lattice cover is of the form $c\xi \operatorname{Id}^n$ such that $\xi = (\xi_1, ..., \xi_n) \in \mathbb{Z}^n$ with*

$$\left\lfloor \frac{m_i}{c} \right\rfloor \leq \xi_i \leq \left\lceil \frac{M_i}{c} \right\rceil$$

for all $1 \leq i \leq n$.

Proof. The cubical lattice is constructed from the integer lattice $\mathbb{Z}^n$ which has generating matrix $M = \operatorname{Id}^n$. Then any point in the cubical lattice can be written $c\xi M = (c\xi \operatorname{Id}^n) = c\xi$, meaning we can bound $\xi \in \mathbb{Z}^n$ with $c\xi \in B$ as in Equation (2.2),

$$\begin{aligned}
\vec{m} \leq c\xi &\leq \vec{M} \\
m_i \leq c\xi_i &\leq M_i \\
\frac{m_i}{c} \leq \xi_i &\leq \frac{M_i}{c} \\
\left\lfloor \frac{m_i}{c} \right\rfloor \leq \xi_i &\leq \left\lceil \frac{M_i}{c} \right\rceil
\end{aligned}$$

for all $1 \leq i \leq n$. We must include floor and ceiling function on our final bounds to maintain that ξ is an integer tuple. □

To count the number of lattice points required for any given cover, we just need to count the number of integers in a given interval. For example, the number of integers ξ such that $m \leq \xi \leq M$ is $\lceil M - m + 1 \rceil$. Thus, given the bounds in Proposition 2.7, we can compute the maximum number of lattice points required to create the cubical lattice cover over the data set X as

$$\prod_{i=1}^{n} \left\lceil \frac{M_i - m_i}{c} + 1 \right\rceil$$

many points. Since $c = \max_{1 \leq i \leq k} \frac{M_i - m_i}{k}$, the size of the cubical lattice cover is similar in size to original cubical cover which both contain on the order of k^n many cover sets.

Next in our constructions we compute bounds on $\xi \in \mathbb{Z}^n$ for each variation of the A_n^*-lattice cover. We first begin with the smallest case, A_2^* with the generating matrix $M_{A_2^*}$.

Proposition 2.8. *Consider the A_2^*-lattice cover constructed from k-intervals and overlap fraction g over a compact topological space X with bounding box B. Then each lattice point $c\xi M_{A_2^*}$ with $\xi = (\xi_1, \xi_2) \in \mathbb{Z}^2$ has bounds*

$$\left\lfloor \frac{m_1 + \frac{1}{\sqrt{3}} m_2}{c} \right\rfloor \leq \xi_1 \leq \left\lceil \frac{M_1 + \frac{1}{\sqrt{3}} M_2}{c} \right\rceil,$$
$$\left\lfloor \frac{2m_2}{3c} \right\rfloor \leq \xi_2 \leq \left\lceil \frac{2M_2}{3c} \right\rceil.$$

Proof. The generating matrix for the A_2^*-lattice is

$$M_{A_2^*} = \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix},$$

To find proper values for ξ we simply run through inequalities.

$$\vec{m} = \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} \leq \xi M_{A_2^*} \leq \vec{M} = \begin{bmatrix} M_1 \\ M_2 \end{bmatrix},$$

which implies that

$$m_1 \leq \xi_1 - \frac{1}{2}\xi_2 \leq M_1$$
$$m_2 \leq \frac{\sqrt{3}}{2}\xi_2 \leq M_2.$$

Hence, after simplifying we have

$$m_1 + \frac{1}{\sqrt{3}} m_2 \leq \xi_1 \leq M_1 + \frac{1}{\sqrt{3}} M_2,$$
$$\frac{2m_2}{3} \leq \xi_2 \leq \frac{2M_2}{3}.$$

To construct the A_2^*-lattice cover, we consider points $c\xi M_{A_2^*}$ on the scaled lattice cA_2^*. With this scaling our inequalities for ξ are reduced to

$$\left\lfloor \frac{m_1 + \frac{1}{\sqrt{3}} m_2}{c} \right\rfloor \leq \xi_1 \leq \left\lceil \frac{M_1 + \frac{1}{\sqrt{3}} M_2}{c} \right\rceil$$
$$\left\lfloor \frac{2m_2}{3c} \right\rfloor \leq \xi_2 \leq \left\lceil \frac{2M_2}{3c} \right\rceil.$$

$\square$

With these inequalities now know, the A_2^*-lattice cover will have at most

$$\left\lceil \frac{2}{3c}(M_2 - m_2) + 1 \right\rceil * \left\lceil \frac{1}{c}(M_1 - m_1) + \frac{1}{c\sqrt{3}}(M_2 - m_2) + 1 \right\rceil$$

many cover sets. This has order of at most k^2 so it is similar in size to the cubical lattice and cubical covers.

Proposition 2.9. *Consider the A_3^*-lattice cover constructed from k-intervals and overlap fraction g over a compact topological space X with bounding box B. Then each lattice point $c\xi M_{A_3^*}$ with $\xi = (\xi_1, \xi_2, \xi_3) \in \mathbb{Z}^3$ is such that*

$$\left\lfloor \frac{m_i - M_3}{2c} \right\rfloor \leq \xi_i \leq \left\lceil \frac{M_i - m_3}{2c} \right\rceil,$$

for $i = 1, 2$ and

$$\left\lfloor \frac{m_3}{c} \right\rfloor \leq \xi_3 \leq \left\lceil \frac{M_3}{c} \right\rceil.$$

Proof. The generating matrix for A_3^* is

$$M_{A_3^*} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

so our inequalities for finding the appropriate ξ given $\vec{m} \leq \xi M \leq \vec{M}$ are going to be

$$\begin{aligned} m_1 &\leq 2\xi_1 + \xi_3 \leq M_1 \\ m_2 &\leq 2\xi_2 + \xi_3 \leq M_2 \\ m_3 &\leq \xi_3 \leq M_3 \end{aligned}$$

which implies that

$$\frac{1}{2}(m_i - M_3) \leq \xi_i \leq \frac{1}{2}(M_i - m_3)$$

for $i = 1, 2$ and

$$m_2 \leq \xi_3 \leq M_3.$$

To construct the lattice cover, we scale the generating matrix by c. With our scaling, each point in our lattice is of the form $c\xi M_{A_3^*}$ and so our inequalities for ξ are reduced to

$$\left\lfloor \frac{m_i - M_3}{2c} \right\rfloor \leq \xi_i \leq \left\lceil \frac{M_i - m_3}{2c} \right\rceil,$$

for $i = 1, 2$ and

$$\left\lfloor \frac{m_3}{c} \right\rfloor \leq \xi_3 \leq \left\lceil \frac{M_3}{c} \right\rceil.$$

$\square$

Again, with the above inequalities we compute that A_3^*-lattice cover will have at most

$$\left\lceil \frac{M_3 - m_3}{c} + 1 \right\rceil \prod_{i=1}^{2} \left\lceil \frac{(M_i - m_i) + (M_3 - m_3)}{2c} + 1 \right\rceil$$

many cover sets. This is on the order of k^3 cover sets and so it has similar efficiency to the cubical lattice and cubical covers.

Proposition 2.10. *Consider the A_n^*-lattice cover constructed from k-intervals and overlap fraction g over a compact topological space X with bounding box B. Then each lattice point $c\xi M_{A_n^*}$ with $\xi \in \mathbb{Z}^n$ is such that*

$$\left\lfloor \frac{m_{n+1} - M_{j+1}}{c} \right\rfloor \leq \xi_j \leq \left\lceil \frac{M_{n+1} - m_{j+1}}{c} \right\rceil,$$

for $1 \leq j \leq n-1$ and

$$\left\lfloor \frac{(n+1)m_{n+1}}{c} \right\rfloor \leq \xi_n \leq \left\lceil \frac{(n+1)M_{n+1}}{c} \right\rceil.$$

Proof. The generating matrix $M_{A_n^*}$ is of the form

$$M_{A_n^*} = \begin{bmatrix} 1 & -1 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & -1 & 0 & \cdots & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdots & \cdot & \cdot \\ 1 & 0 & 0 & 0 & \cdots & -1 & 0 \\ \frac{-n}{n+1} & \frac{1}{n+1} & \frac{1}{n+1} & \frac{1}{n+1} & \cdots & \frac{1}{n+1} & \frac{1}{n+1} \end{bmatrix},$$

where we can describe the rows v_i of $M_{A_n^*}$ in terms of the basis vectors e_i of $\mathbb{R}^{n+1}$ as $v_i = e_1 - e_{i+1}$ for $1 \leq i \leq n-1$ and $v_n = \left(\frac{-n}{n+1}, \frac{1}{n+1}, \frac{1}{n+1}, ..., \frac{1}{n+1}\right)$. To find optimal $\xi \in \mathbb{Z}^n$ we want to solve the set of inequalities

$$\vec{m} = \begin{bmatrix} m_1 \\ \vdots \\ m_n \end{bmatrix} \leq \xi M_{A_n^*} \leq \begin{bmatrix} M_1 \\ \vdots \\ M_n \end{bmatrix} = \vec{M}.$$

Alternatively, we can write these in terms of our basis so for each i we have

$$m_i \leq \sum_{k=1}^{n} \xi_k v_{ki} \leq M_i.$$

We notice that

$$v_{ki} = \begin{cases} 1, & i = 1, 1 \leq k \leq n-1 \\ \frac{-n}{n+1}, & i = 1, k = n \\ -1, & i = k+1, 1 \leq k \leq n-1 \\ \frac{1}{n+1}, & 2 \leq i \leq n+1, k = n \\ 0, & \text{otherwise} \end{cases}$$

so we can expand our above inequality for the following cases:

1. For $i = 1$:
$$m_1 \leq \xi_n \left(\frac{-n}{n+1} \right) + \sum_{k=1}^{n-1} \xi_k \leq M_1$$

2. For $2 \leq i \leq n$:
$$m_i \leq \xi_n \left(\frac{1}{n+1} \right) - \xi_{i-1} \leq M_i$$

3. For $i = n+1$:
$$m_{n+1} \leq \xi_n \left(\frac{1}{n+1} \right) \leq M_{n+1}.$$

Case 3 immediately yields the bounds

$$(n+1)m_{n+1} \leq \xi_n \leq (n+1)M_{n+1}.$$

Using this we can find bounds for $\xi_1, ..., \xi_{n-1}$ via Case 2 to get:

$$\left(\frac{1}{n+1} \right) \xi_n - M_{j+1} \leq \xi_j \leq \left(\frac{1}{n+1} \right) \xi_n - m_{j+1},$$

implying for $1 \leq j \leq n-1$,

$$m_{n+1} - M_{j+1} \leq \xi_j \leq M_{n+1} - m_{j+1}.$$

Therefore the A_n^*-lattice cover should consist of all points $\xi M_{A_n^*}$ such that $\xi \in \mathbb{Z}^n$ has the above bounds. Scaling by c, our lattice points are $c\xi M_{A_n^*}$ and so our inequalities for ξ are reduced to

$$\left\lfloor \frac{m_{n+1} - M_{j+1}}{c} \right\rfloor \leq \xi_j \leq \left\lceil \frac{M_{n+1} - m_{j+1}}{c} \right\rceil$$

$$\left\lfloor \frac{(n+1)m_{n+1}}{c} \right\rfloor \leq \xi_n \leq \left\lceil \frac{(n+1)M_{n+1}}{c} \right\rceil.$$

□

The A_n^*-lattice cover will then have at most

$$\left(\left\lceil \frac{(n+1)(M_{n+1} - m_{n+1})}{c} \right\rceil + 1 \right) \prod_{j=1}^{n-1} \left(\left\lceil \frac{M_{n+1} + M_{j+1} - m_{j+1} - m_{n+1}}{c} \right\rceil + 1 \right)$$

many cover sets. This is on the order of $2(n+1)k^n$ making it a larger cover than both the cubical lattice and cubical covers for any dimension n. This is due to the generating matrix being embedded as a hyperplane in $\mathbb{R}^{n+1}$, so it may not be the most efficient choice when computing 2-Mapper.

2.3 Analysis of 2-Mapper through Cover Choice

With our covers now defined we can compute a 2-Mapper complex across multiple distinct cover choices. Importantly, 2-Mapper allows us to compute Betti-1 so we created an initial experiment to see if there was any empirical evidence to identify an optimal cover choice. For this experiment we inquire as to which 2-Mapper complexes can spatially represent a *stochastic triangle*.

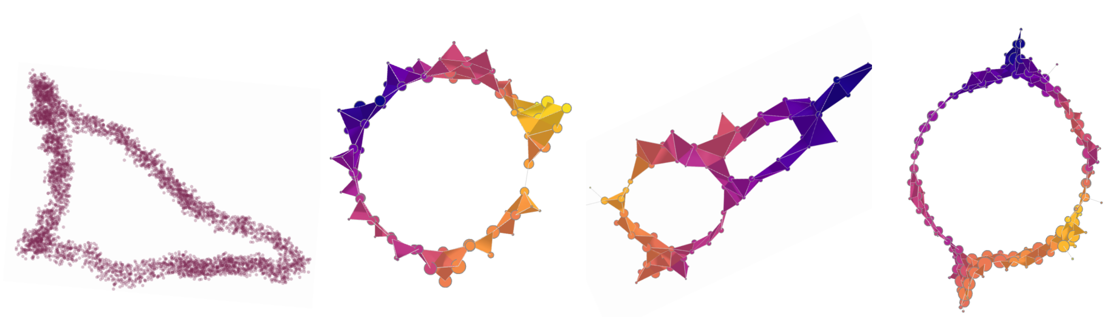

Figure 2.5. A stochastic triangle, simulated in $\mathbb{R}^8$, projected with PCA to $\mathbb{R}^2$ (left). We compute 2-Mapper complexes using a cubical cover (center left), cubical lattice cover (center right) and A_2^*-lattice cover (right). All covers were constructed with $k = 10$ intervals and overlap fraction $g = 0.3$.

The stochastic triangle data set consists of 5000 points sampled from a simulated random walk about a randomly generated triangle embedded in $\mathbb{R}^8$. We randomly choose three points in $[-100, 100]^8 \subset \mathbb{R}^8$, checking that the angle between each pair of vertices ranges from $0.35 \leq \theta \leq 2.4$ radians. We then randomly walk along each edge within a chosen standard deviation of 5, and then sample 5000 points with noise (uniformly distributed on [-5,5]) and get our stochastic triangle. We give an example of a stochastic triangle in Figure 2.5, but note that the two-dimensional projection is much more "filled in" than the eight-dimensional data set itself. Also in Figure 2.5 we provide 2-Mapper complexes for each cover type (cubical, cubical lattice, and A^*-lattice), and this shows that identical cover parameters do not necessarily produce 2-Mapper complexes with the same topological structure.

Our main aim in this experiment is to see if there is an optimal cover for computing Betti-1 for 2-Mapper for such a stochastic triangle, as representative possible structure in a larger data set. This experiment predates our work on Multiscale 2-Mapper and was indeed a strong motivation to pursue its development. The hopeful expectation would be to see Betti-1 predominantly equal to one for some choices of good covers. Generating one-hundred stochastic triangles in $\mathbb{R}^8$, we compute the Betti-1 of 2-Mapper complexes using multiple reference functions and cover types. In Figure 2.6 we plot histograms representing the Betti-1 value for these 2-Mappers. In the first row, we are considering projections to $\mathbb{R}^2$ and we notice that the accuracy between the cubical cover and cubical lattice cover are approximately the same, however PCA for the choice of reference function is about twice as accurate. These initial results suggest cubical cover as optimal, but we will see as we increase the dimension of our projection the histograms change quite drastically. In Figure 2.6C&E, we see that once we include more than two components for PCA, the cubical cover has a much more difficult time correctly estimating Betti-1 while the cubical lattice cover is far more stable.

When looking at the coordinate projection lens functions, Figure 2.6B&D&F, our results are relatively constant as we increase the dimension of the codomain. There is a slight increase in performance for all cover types when projecting to the first three coordinates, however this reduces for both the cubical cover and A^*-lattice cover when increasing the projection dimension to four. Overall, the A^*-lattice cover performed the worst in all cases. Both the cubical cover and cubical lattice cover performed similarly except when using PCA with three or four com-

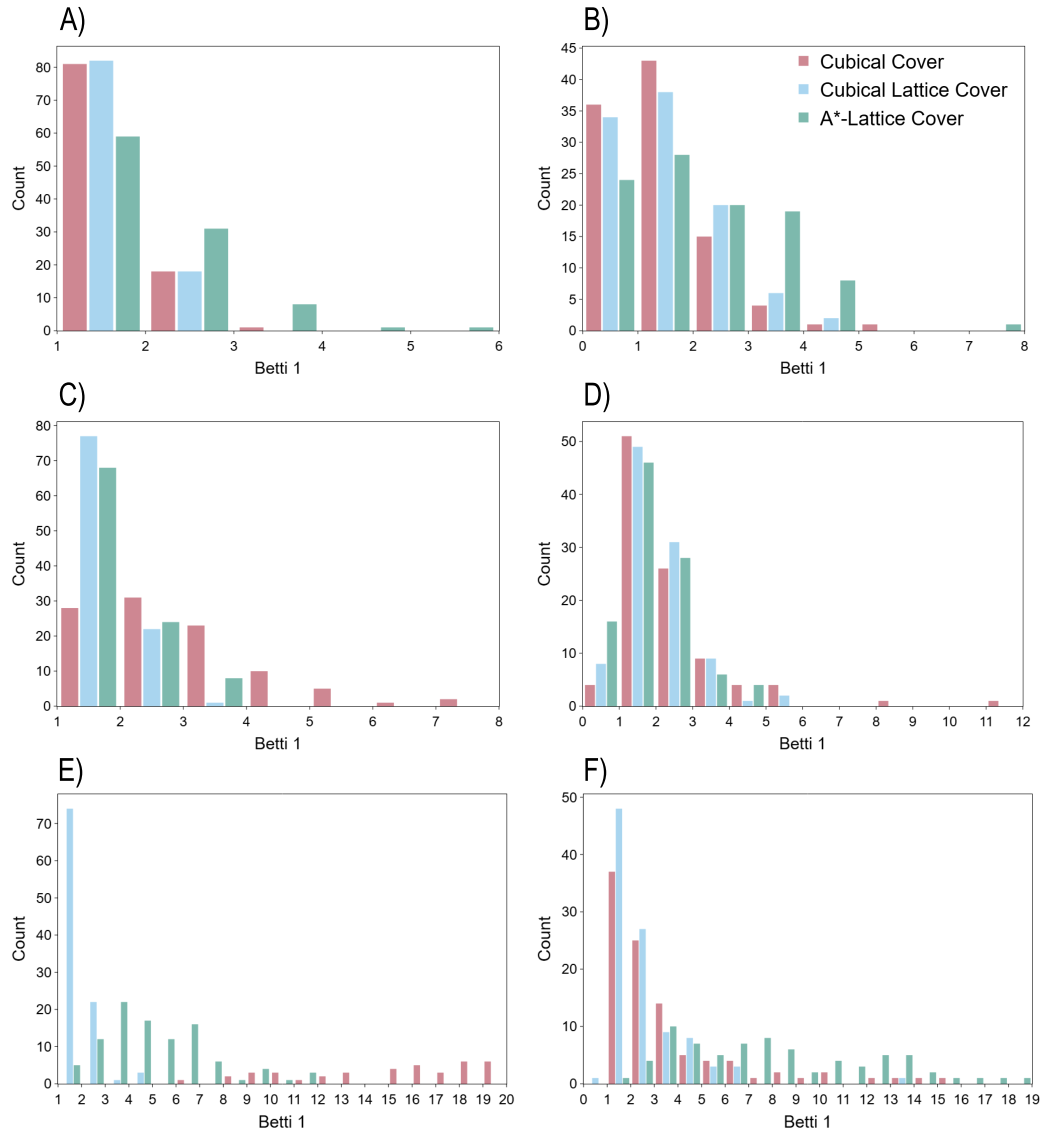

Figure 2.6. We compute Betti-1 for 2-Mapper complexes constructed from one-hundred stochastic triangles embedded in $\mathbb{R}^8$ with various choices for cover and reference function. We fix DBSCAN with default parameters as our clustering algorithm. For each subplot the 2-Mapper is constructed with a different lens function and each colored histogram represents a different cover type with cubical cover in red, cubical lattice cover in blue, and A^*-lattice cover in green. The first column uses PCA as the reference function with 2, 3, and 4 components for A, C, and E, respectively. The second column projects down to the first 2, 3, and 4 coordinates for B, D, and F, respectively. All covers are computed with 10 intervals and overlap fraction of 0.3.

ponents as a reference function. The cubical cover always subdivides each dimension of the co-domain into ten intervals of equal length, this likely leads to more noise than the cubical lattice covers, whose cover sets are uniform across all dimensions. We conjecture that this is the case, and is something we are looking into in future work. We do not however come away with concrete recommendations for either theoretical results or implementation suggestions for Mapper when comparing individual 2-Mapper complexes. Indeed, this motivated the results of Chapter 3, to more fully develop a robust set of cover choices.

CHAPTER 3

MULTISCALE 2-MAPPER

To understand 2-Mapper, as well as Mapper, across a range of cover choices, we implement *Multiscale 2-Mapper* for data. A basic challenge in employing Mapper is choosing parameters, since it is unstable. With Multiscale 2-Mapper, we can connect with persistent homology, and choose covers which are representative of persistent features.

In this chapter, we first formally define Multiscale 2-Mapper and then we extend our lattice covers to be used for Multiscale 2-Mapper. We prove properties of these covers needed for our computational setting. Finally, we discuss our implementation.

3.1 Multiscale Mapper for data with DBSCAN

While Dey, Mémoli, and Wang produced theoretical foundations for Multiscale Mapper in the setting of topological spaces and continuous functions, they did not adapt their ideas to be used with data. To emphasize this distinction, we define a computational Mapper graph to be one produced from a data set as in Definition 1.1. Recent work by Bungula and Darcy [BD24] makes needed progress to extend Multiscale Mapper to apply to data sets under some hyperparameter choices of the clustering algorithm DBSCAN. This means that it is possible to implement multiscale Mapper for the computational setting, though their implementation, created in R, is not yet available. We build on their results for our implementation of Multiscale Mapper in Python.

DBSCAN [Est+96] is a non-parametric density based clustering algorithm and is a common choice for the clustering algorithm Mapper. Unlike other clustering algorithms, DBSCAN can cluster points into non-linearly separable clusters and is robust to noise. It requires two parameters: the minimum number of points for each cluster MinPts $\in \mathbb{Z}^+$ and a search radius $\epsilon \in \mathbb{R}^+$. To cluster a data set (X, d_X), we consider ϵ-neighborhoods about each point p, denoted $N_\epsilon(p) = \{q \in X \colon d_X(p, q) \leq \epsilon\}$. We can then classify each point in X as one of the following:

- A *core point*: a point $p \in X$ such that $|N_\epsilon(p)| \geq$ MinPts.
- A *border point*: a point $q \in X$ such that $|N_\epsilon(q)| <$ MinPts and $q \in N_\epsilon(p)$ for a core point p.

- *Noise*: A point $r \in X$ that is neither a core point nor a border point.

From these three point labels we can decompose $X = C_1 \sqcup C_2 \sqcup \cdots \sqcup C_n \sqcup N$ such that each C_i is a cluster containing both core and border points, and N is the set of noise points. Points in N are not necessarily close in distance to one another. As N is the "cluster" of points which are neither a border nor core point, we consider N as a degenerate set. In fact, the results below for computational Multiscale Mapper removes the noise set N from the cluster decomposition.

A drawback to DBSCAN for our use is that it is not deterministic. Data points can belong to more than one cluster, and the determination of the cluster label for such points is dependent on the ordering of the data set. This occurs when a border point in X is equidistant to core points in two separate clusters. We call all such border points *free border points*. All the work of Bungula and Darcy assume that a data set is void of free border points, a reasonable assumption in many applications. They additionally show that X has no free border points when $\mathrm{MinPts} \leq 2$ [BD24, Corollary 1]. Hence we will keep this assumption and in our implementation fix $\mathrm{MinPts} \leq 2$.

For the following results, assume X is a data set with tower of covers $\mathcal{U} = \{U_\varepsilon\}_{\varepsilon \geq s}$, with resolution $s \in \mathbb{R}^+$, with each cover $U_\varepsilon = \{U_{\alpha,\varepsilon}\}_{\alpha \in A}$.

Lemma 3.1 ([BD24])**.** *Suppose there are no free-border points in the DBSCAN cluster decomposition of a data set X with hyperparameters ϵ and* MinPts*. Let $U_1 \subseteq U_2$. Denote by $C_{p,1}$ a cluster of U_1 determined by core point $p \in X$ and $C_{p,2}$ a cluster of U_2 also with core point p. Then $C_{p,1} \subseteq C_{p,2}$.*

Thus, as we traverse the tower of covers that $U_{\alpha,\varepsilon} \subseteq U_{\alpha,\delta}$ for $\varepsilon \leq \delta$, the clusters of $U_{\alpha,\varepsilon}$ are contained in clusters of $U_{\alpha,\delta}$ that contain the same core points. This allows us to define maps between clusters, creating a filtration as needed to construct Multiscale Mapper.

Theorem 3.2 ([BD24])**.** *Let X be a data set with tower of covers $\mathcal{U}$ and use DBSCAN to cluster X with fixed ϵ and* MinPts*. If no free-border points exist with respect to ϵ and* MinPts*, define $C_\varepsilon = \{C_{p_\alpha,\varepsilon} \colon p_\alpha \in U_{\alpha,\varepsilon} \text{ and } p_\alpha \text{ is a core point}\}$. Then these constitute a filtration of cluster covers, and in particular there are maps $c_{\varepsilon,\delta} : C_\varepsilon \to C_\delta$ for all $U_\varepsilon \subseteq U_\delta$ which are closed under composition.*

Corollary 3.3 ([BD24])**.** *Let DBSCAN be used to cluster X with fixed ϵ and* $\mathrm{MinPts} = 1, 2$*. Then, increasing cover set size constitutes a filtration of cluster covers.*

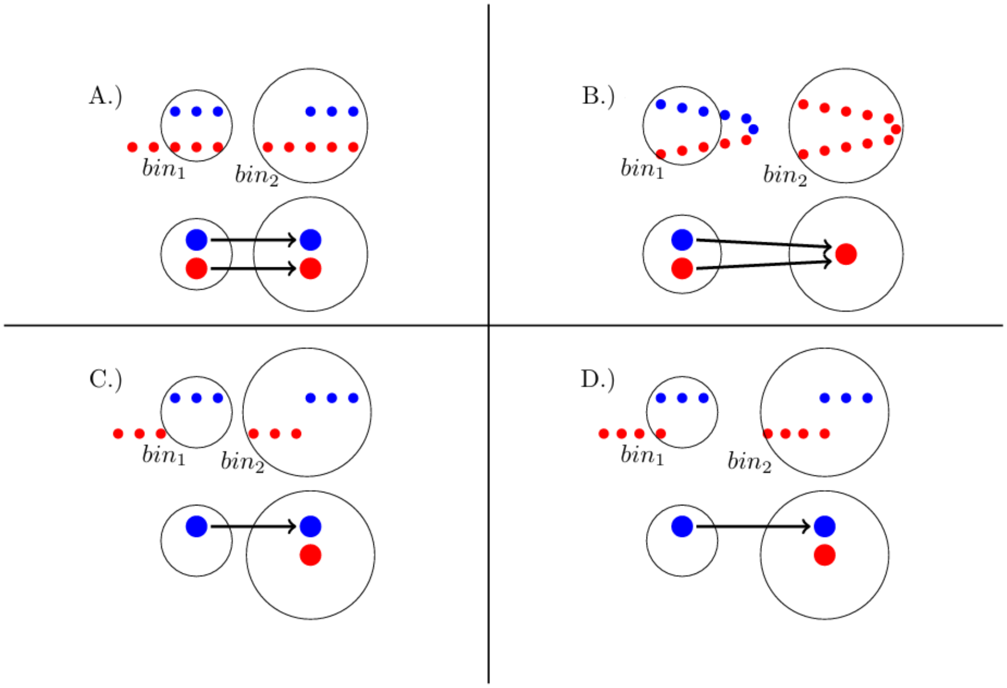

Figure 3.1. The following subfigures each show a case of cluster decomposition changing as cover set size increases [BD24, Figure 7]. For each subfigure, bin_1 and bin_2 refer to cover sets such that $bin_1 \subseteq bin_2$. The top row of each subfigure shows the data set as the cover set increases, while the bottom row shows the cluster decomposition of each cover set (colored to identify which data belongs to which cluster). As we increase the size of the cover set, we see that there are four possible cases which occur. Clusters will always map to a cluster with the same core point as bin size increases as in (A). However, it is possible that two clusters collapse as cover set size increases (B). Additionally, as the cover set as it increases it can contain more data, and this can create new clusters (C, D). In the case where the cover set contains noisy data (D), it is possible for that data to be included into a new created cluster as the cover set increases.

Theorem 3.4 ([BD24])**.** *Let X be a data set and assume DBSCAN is used to cluster X with fixed hyperparameters ϵ and* MinPts*. If no free-border points exist with respect to ϵ and* MinPts *then,*

- *There is a filtration of simplicial complexes $\phi_{\varepsilon,\delta} \colon \mathcal{N}(C_\varepsilon) \to \mathcal{N}(C_\delta)$ for all $U_\varepsilon \subseteq U_\delta$.*
- *There is a filtration of homology groups $f_{\varepsilon,\delta} \colon H_k(\mathcal{N}(C_\varepsilon)) \to H_k(\mathcal{N}(C_\delta))$ for all $U_\varepsilon \subseteq U_\delta$ and each degree $k \in \mathbb{N}$.*

Bungula and Darcy additionally show other filtrations of cluster covers across alternate hyperparameters including ϵ and MinPts. However, our work does not use these filtrations.

With a well defined filtration of cluster covers for a data set X with fixed DBSCAN parameters and tower of covers $\mathcal{U}$, we can extend to a computational Multiscale Mapper. The main technical point which needs to be addressed will be to augment the 2-Mapper complexes so that there are natural maps between them.

A key consideration is how the cluster decomposition for each cover set changes as the size of each cover set increases. When we apply DBSCAN to a data set X, we are applying it to each cover U_ε in the tower of covers. This decomposes each cover set into clusters $U_{\alpha,\varepsilon} = \bigsqcup C_{p_{\alpha,\varepsilon}}$. For $\delta \geq \varepsilon$, the cover set $U_{\alpha,\delta}$ can have a different cluster decomposition – see Figure 3.1. As the size of each cover set increases, we can include more data from X (Figure 3.1(C, D)). In the case of Figure 3.1(B), two clusters can collapse into one distinct cluster. We still have a filtration of cluster covers in each case of Figure 3.1, however the collapse of clusters can complicate the implementation of a Multiscale Mapper algorithm, as we discuss below.

3.2 Multiscale 2-Mapper

Recall Definition 1.12, where in the setting topological spaces, *Multiscale Mapper* is defined as a filtration of Mapper graphs constructed from a filtration of covers. Here we define Multiscale 2-Mapper as a Multiscale Mapper object consisting of two-dimensional nerves, but our results will then apply to classical Mapper graphs.

Definition 3.5. Let $Z \subset \mathbb{R}^n$ be a compact topological space. Then for a reference function $f\colon Z \to \mathbb{R}^m$ and tower of covers $\mathcal{U} = \{U_\varepsilon\}_{\varepsilon \geq s}$ over $f(Z)$, the *Multiscale 2-Mapper*, denoted $\mathrm{MM}_2(f, \mathcal{U})$, for a finite sequence $s \leq \varepsilon_1 \leq \cdots \leq \varepsilon_k$, is a filtration of 2-Mapper complexes

$$\mathcal{N}^2(f^*(U_s)) \to \mathcal{N}^2(f^*(U_{\varepsilon_1})) \to \cdots \to \mathcal{N}^2(f^*(U_{\varepsilon_k})).$$

We extend work from [DMW], by investigating towers of covers in the computational setting. We show results for particular parameters in our construction for towers of covers below. Our choices and results lead to an implementation, which we describe in Section 3.3.

We now extend our cover definitions from Section 2.2 to a tower of covers satisfying Definition 1.11. By construction, each of these will be a tower of covers with nested cover sets, so that Multiscale 2-Mapper is well-defined. This construction for each cover type is straightforward, each cover type can be expressed as an interval (cubical and cubical lattice covers) or ball (lattice cover) with a fixed center. Once we create an initial cover, later covers in the tower are the initial cover with scaled endpoints or radii, respectively. This means we can additionally think of this as a filtration of covers through the overlap fraction g – see Figure 3.2.

For the cubical cover, Definition 2.3, recall that each cover set can be expressed as an interval centered at a point c_i with radius $\frac{1}{2}l$ – see Equation (2.1). To construct a tower of cubical covers we initialize a cover with chosen parameters, fixing the centers of each cover set. As seen in Figure 3.2A, each subsequent cover in the tower is then defined with the same centers as the initial cover and we extend the radius by some $\varepsilon > 0$.

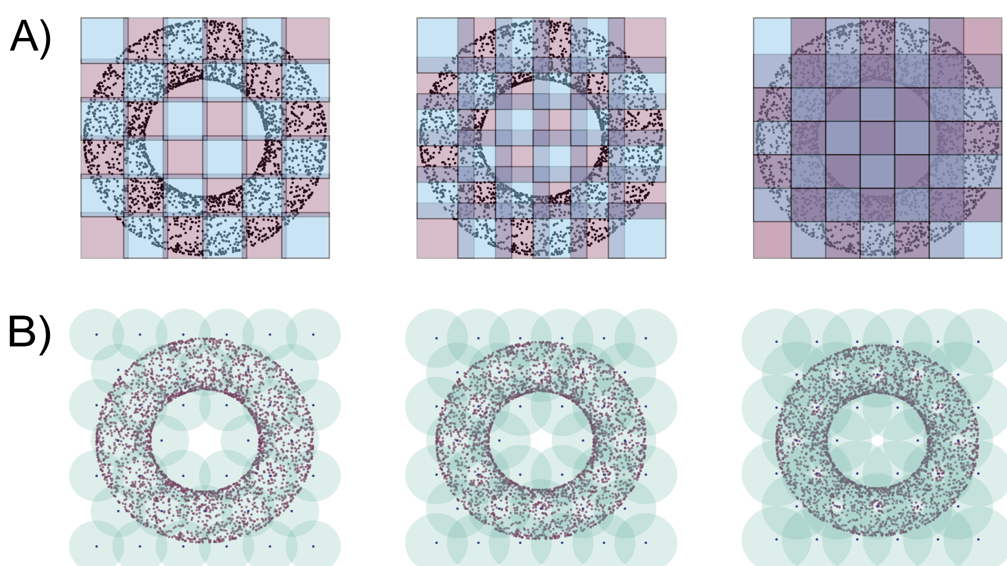

Figure 3.2. Two tower of covers over a torus point cloud in $\mathbb{R}^2$. Each tower consists of three covers each compute with $k = 6$ and $g = 0.1,\ 0.3,\ 0.5$ (left, center, right). A) The tower of cubical covers. B) The tower of A_2^*-lattice covers.

Definition 3.6 (Tower of Cubical Covers)**.** Let $U_s = \{U_{\alpha,s}\}_{\alpha\in A}$ be a cubical cover over a compact topological space $X \subset \mathbb{R}^n$ defined with k-intervals and overlap fraction g so that every cover set in U_s is of the form

$$U_{\alpha,s} = \prod_{i=1}^{n} \left[c_{\alpha,i} - \frac{l_i}{2}, c_{\alpha,i} + \frac{l_i}{2} \right],$$

for all $\alpha \in A$ where

$$c_{\alpha,i} = m_1 + (\alpha_i - 1)(1-g)l_i + \frac{1}{2}l_i, \quad \text{and} \quad l_i = \frac{M_i - m_i}{k-(k-1)g}.$$

The *tower of cubical covers* $\mathcal{U}$ on X is a tower of covers $\mathcal{U} = \{U_\varepsilon\}_{\varepsilon\geq s}$ with resolution $\mathrm{res}(\mathcal{U}) = s = \|(l_1, ..., l_n)\|_2$. For $\varepsilon \geq s$ we define the cover $\mathcal{U}_\varepsilon = \{U_{\alpha,\varepsilon}\}_{\alpha\in A}$ such that for each $\alpha \in A$

$$U_{\alpha,\varepsilon} = \prod_{i=1}^{n} \left[c_{\alpha,i} - \frac{1}{2}\left(l_i - \varepsilon'\right), c_{\alpha,i} + \frac{1}{2}\left(l_i + \varepsilon'\right) \right],$$

for some $\varepsilon' \geq 0$ so that

$$\mathrm{diam}(U_{\alpha,\varepsilon}) = ||(l_1 + \varepsilon', ..., l_n + \varepsilon')||_2 = \varepsilon.$$

For any $s \leq \varepsilon \leq \delta$ we have canonical cover maps $u_{\varepsilon,\delta} \colon U_\varepsilon \to U_\delta$ where $U_{\alpha,\varepsilon} \hookrightarrow U_{\alpha,\delta}$ for all $\alpha \in A$.

We note that the resolution of a tower of covers is the diameter of each cover set $U_{\alpha,s} \in \mathcal{U}_s$.

The construction is similar for a tower of covers derived from a lattice cover. Recall that each point in the lattice cover construction is given by $c\xi M$ for some $\xi \in \mathbb{Z}^n$ – see Definitions 2.5 and 2.6. The cover sets in a (cubical) lattice cover are defined as (intervals) balls centered at each of these lattice points. For subsequent covers in the tower of covers, we define each cover set to still be centered at these lattice points and we scale the radius of the (interval) ball by some $\varepsilon > 0$. We see an example of a tower of lattice covers in Figure 3.2B.

Definition 3.7 (Tower of Cubical Lattice Covers)**.** Let $U_s = \{U_{\alpha,s}\}_{\alpha\in A}$ be a cubical lattice cover over $X \subseteq \mathbb{R}^n$ defined with k-intervals and overlap fraction g so that every cover set in U_s is of the form

$$U_{\alpha,s} = \prod_{i=1}^{n} \left[\left(c\xi_{\alpha,i} - \frac{c}{2(1-g)} \right), c\left(c\xi_{\alpha,i} + \frac{c}{2(1-g)} \right) \right].$$

The *tower of cubical lattice covers* $\mathcal{U}$ is a tower of covers $\mathcal{U} = \{U_\varepsilon\}_{\varepsilon \geq s}$ with resolution $\text{res}(\mathcal{U}) = s$. For $\varepsilon \geq s$ we define the cover $U_\varepsilon = \{U_{\alpha,\varepsilon}\}_{\alpha \in A}$ such that for each $\alpha \in A$

$$U_{\alpha,\varepsilon} = \prod_{i=1}^{n} \left[c\xi_{\alpha,i} - \frac{1}{2}\left(\frac{c}{(1-g)} - \varepsilon'\right), c\xi_{\alpha,i} + \frac{1}{2}\left(\frac{c}{(1-g)} + \varepsilon'\right) \right],$$

for some $\varepsilon' \geq 0$ so that

$$\text{diam}(U_{\alpha,\varepsilon}) = \left\| \left(\frac{c}{1-g} + \varepsilon', ..., \frac{c}{1-g} + \varepsilon' \right) \right\|_2 = \varepsilon.$$

For any $s \leq \varepsilon \leq \delta$ we have canonical cover maps $u_{\varepsilon,\delta} \colon U_\varepsilon \to U_\delta$ where $U_{\alpha,\varepsilon} \to U_{\alpha,\delta}$ for all $\alpha \in A$.

Again we define the resolution s to be the diameter of each cover set $U_{\alpha,s} \in \mathcal{U}_s$. The length of each side of the cover set is $\frac{c}{1-g}$, so this is directly computed as $s = \frac{c\sqrt{n}}{1-g}$.

Definition 3.8 (Tower of Lattice Covers)**.** Let Λ be a lattice with generating matrix M and covering radius R, and let $U_s = \left\{B\left(c\xi M, \frac{1}{2}s\right)\right\}_{c\xi M \in B \cap \mathbb{Z}^n}$ be a Λ-lattice cover over $X \subseteq \mathbb{R}^n$ defined with k-intervals and overlap fraction g.

The *tower of* Λ*-lattice covers* $\mathcal{U}$ is a tower of covers $\mathcal{U} = \{U_\varepsilon\}_{\varepsilon \geq s}$ with $\text{res}(\mathcal{U}) = s = 2cR(1+g)$. For $\varepsilon \geq s$, we define the cover $U_\varepsilon = \left\{B\left(c\xi M, \frac{1}{2}\varepsilon\right)\right\}_{c\xi M \in B \cap \mathbb{Z}^n}$ with $\varepsilon = s + \varepsilon'$ for some $\varepsilon' > 0$. For any $s \leq \varepsilon \leq \delta$ we have canonical cover maps $u_{\varepsilon,\delta} \colon U_\varepsilon \to U_\delta$ where $B\left(c\xi M, \frac{1}{2}\varepsilon\right) \to B\left(c\xi M, \frac{1}{2}\delta\right)$ for all $c\xi M \in B \cap \mathbb{Z}^n$.

Stability for Computable Towers of Covers

Using a tower of covers, we can construct the Multiscale 2-Mapper using Algorithm 1. Previous theoretical work for Multiscale Mapper shows the stability for (c,s)-*good covers*, and show that covers defined on lattices are $(3,s)$-good. Recall Definition 1.15 that for $c \geq 1$ and $s > 0$ a finite tower of covers $\mathcal{U} = \{U_\varepsilon\}_{\varepsilon \geq s}$ with resolution s is (c,s)-good if

(i). $\text{res}(\mathcal{U}) = s$ and $s \leq \text{diam}(Z)$,

(ii). $\text{diam}(U) \leq \varepsilon$ for all $U \in U_\varepsilon$ and all $\varepsilon \geq s$, and

(iii). for all $O \subseteq Z$ with $\text{diam}(O) \geq s$ there exists $U \in U_{c \cdot \text{diam}(O)}$ such that $O \subseteq U$.

Towers of (c, s)-good covers interleave with one another – see Theorem 1.16 – and they also interleave with with the inferred Čech complex [DMW, Theorem 7.7]. We prove these results under certain hyperparameter choices for computable covers. We begin with the following lemma.

Lemma 3.9. *Let $B = \prod_{i=1}^{n}[m_i, M_i]$ be the bounding box of $Z \subseteq \mathbb{R}^n$. Then $\max_{i=1,\ldots,n} M_i - m_i \leq \mathrm{diam}(Z)$.*

Proof. For all $1 \leq i \leq n$, $M_i - m_i$ is the length of the bounding box parallel to the i-th coordinate axes in $\mathbb{R}^n$. The diameter of Z is maximum length of a line segment endpoints in Z. By our construction $m_i, M_i \in Z$ for all $1 \leq i \leq n$, and so our lemma follows by definition. □

This is a key property we use for proving the first condition of a (c, s)-good cover. We first show that the tower of cubical covers is a (c, s)-good tower of covers.

Theorem 3.10. *A tower of cubical covers $\mathcal{U}$ over a compact metric space $Z \subset \mathbb{R}^n$ with resolution $\mathrm{res}(\mathcal{U}) = s$ constructed with k-intervals such that $k \geq \sqrt{n}$ and overlap fraction g is a $(3, s)$-good tower of covers.*

Proof. Let $B = \prod_{i=1}^{n}[m_i, M_i]$ be the bounding box for Z. Without loss of generality, let $M_1 - m_1 = \max_{i=1,..,n} M_i - m_i$, and define $\vec{M} = (M_1, ..., M_n)$ and $\vec{m} = (m_1, ..., m_n)$. For condition (i), we can rewrite the resolution of $\mathrm{res}(\mathcal{U}) = s$ as

$$\begin{aligned} s = ||l||_2 &= \left(\sum_{i=1}^{n} \left(\frac{M_i - m_i}{k - (k-1)g} \right)^2 \right)^{\frac{1}{2}} \\ &= \frac{1}{k(1-g)+g} \left(\sum_{i=1}^{n} (M_i - m_i)^2 \right)^{\frac{1}{2}} \\ &= \frac{1}{k(1-g)+g} ||\vec{M} - \vec{m}||_2, \end{aligned}$$

which implies that, for $k \geq \sqrt{n}$,

$$\begin{aligned} s &\leq \frac{1}{k} ||\vec{M} - \vec{m}||_2 \leq \frac{1}{k} \left(\sum_{i=1}^{n} (M_1 - m_1)^2 \right)^{\frac{1}{2}} \\ &\leq \frac{\sqrt{n}}{k}(M_1 - m_1) \leq \mathrm{diam}(Z). \end{aligned}$$

For condition (ii) let $\varepsilon \geq s$ and consider the cover $U_\varepsilon \in \mathcal{U}$. Then for some $\varepsilon' > 0$ each cover set $U_{\alpha,\varepsilon}$ is a hypercube with dimensions $l_i^{\varepsilon'} = \frac{M_i - m_i}{k-(k-1)g} + \varepsilon'$, for all $\alpha \in A$. Choose $\varepsilon' = -\frac{1}{n}||l||_1 + \sqrt{\frac{1}{n^2}||l||_1^2 + \frac{1}{n}(\varepsilon^2 - s^2)}$. This choice of ε' is well-defined and non-negative as the discriminant is positive, and

$$\begin{aligned}
\varepsilon' = -\frac{1}{n}||l||_1 + \sqrt{\frac{1}{n}||l||_1^2 - \frac{1}{n}s^2 + \frac{1}{n}\varepsilon^2} > 0 \\
\Rightarrow \frac{1}{n}||l||_1^2 + \frac{\varepsilon^2 - s^2}{n} > \frac{1}{n^2}||l||_1^2 \\
\Rightarrow \frac{\varepsilon^2 - s^2}{n} > \frac{1-n}{n^2}||l||_1^2 \\
\Rightarrow \varepsilon^2 - s^2 > \left(\frac{1}{n} - 1\right)||l||_1^2,
\end{aligned}$$

where the final inequality is true for all $n \geq 1$. Then the diameter of each cover set $U_{\alpha,\varepsilon}$ is

$$\begin{aligned}
\operatorname{diam}(U_{\alpha,\varepsilon})^2 = ||l^{\varepsilon'}||_2^2 &= \sum_{i=1}^{n}(l_i + \varepsilon')^2 \\
&= \sum_{i=1}^{n} l_i^2 + 2l_i\varepsilon' + (\varepsilon')^2 \\
&= s^2 + 2||l||_1\varepsilon' + n(\varepsilon')^2 \\
&= \varepsilon^2,
\end{aligned}$$

yielding $\operatorname{diam}(U_{\alpha,\varepsilon}) = \varepsilon$. For condition (iii), let $O \subseteq Z$ such that $\operatorname{diam}(O) > s$. Let $U_{\alpha,s} \in U_s$ such that $U_{\alpha,s} \cap O \neq \emptyset$. Then for $x \in U_{\alpha,s} \cap O$ and center point $c_\alpha \in U_{\alpha,s}$, we have for any $o \in O$ that

$$||c_\alpha - o||_2 \leq ||c_\alpha - x||_2 + ||x - o||_2 \leq \frac{1}{2}s + \operatorname{diam}(O) \leq \frac{3}{2}\operatorname{diam}(O), \tag{3.1}$$

meaning that $O \subset U_{\alpha, 3\operatorname{diam}(O)} \in U_{3\operatorname{diam}(O)}$. Hence $\mathcal{U}$ is a $(3, s)$-good cover. □

This is a new result for cubical covers which are not lattice covers. While constructed from uniform hypercubes, these hypercubes are not uniformly positioned on the integer lattice, as each basic edge length of the hypercube is independently scaled.

We have similar results for the lattice based tower of covers. We continue to prove these for the other covers. Proofs for these are similar that of Theorem 3.10.

Theorem 3.11. *The tower of cubical lattice covers with resolution s, defined with k-intervals and overlap fraction g is $(3, s)$-good for $k \geq \frac{\sqrt{n}}{1-g}$.*

Proof. Let $B = \prod_{i=1}^{n}[m_i, M_i]$ be the bounding box for Z. Without loss of generality, let $M_1 - m_1 = \max_{i=1,..,n} M_i - m_i$, and define $\vec{M} = (M_1, ..., M_n)$ and $\vec{m} = (m_1, ..., m_n)$. For condition (i), we can rewrite the resolution of $\mathcal{U} = s$ as

$$\begin{aligned} s = ||l||_2 &= \left(\sum_{i=1}^{n}\left(\frac{c}{(1-g)}\right)^2\right)^{\frac{1}{2}} \\ &= \left(n\left(\frac{c}{(1-g)}\right)^2\right)^{\frac{1}{2}} = \sqrt{n}\frac{c}{(1-g)} \\ &= \frac{\sqrt{n}}{k(1-g)}(M_1 - m_1) \leq M_1 - m_1 \leq \mathrm{diam}(Z) \end{aligned}$$

For condition (ii), choose $\varepsilon' = \frac{\varepsilon}{\sqrt{n}} - l$. Note that this choice of ε' is non-negative as

$$\varepsilon > l\sqrt{n} = s \quad \Longrightarrow \quad \varepsilon' > 0.$$

Then

$$||U_{\alpha,\varepsilon}||_2 = ||l + \varepsilon'||_2 = \left(\sum_{i=1}^{n}\left(l + \left(\frac{\varepsilon}{\sqrt{n}} - l\right)\right)^2\right)^{\frac{1}{2}} = \left(\sum_{i=1}^{n}\left(\frac{\varepsilon^2}{n}\right)\right)^{\frac{1}{2}} = \left(\varepsilon^2\right)^{\frac{1}{2}} = \varepsilon.$$

For condition (iii), the proof is identical to the proof in Theorem 3.10, using Equation 3.1. □

We now turn our attention to the tower of lattice covers whose cover sets are balls centered at lattice points. Each lattice cover is defined with a unique generating matrix, and thus each tower of lattice covers has different parameter restrictions to be (c, s)-good.

Theorem 3.12. *Consider the A_2^* lattice with generating matrix $M_{A_2^*}$. A tower of A_2^*-lattice covers defined with k-intervals and overlap fraction g is $(3, s)$-good for $k \geq \frac{2}{\sqrt{3}}(1 + g)$.*

Proof. Let $\mathcal{U}$ be a finite tower of triangular lattice covers over compact $Z \subset \mathbb{R}^2$ with resolution s. Consider the bounding box $Z \subseteq [m_1, M_1] \times [m_2, M_2]$ where $m_i = \min \pi_i(Z)$ and $M_i = \max \pi_i(Z)$ for $i = 1, 2$. Suppose U_s is the A_2^*-lattice cover created with k-intervals and g overlap fraction and generating matrix $M_{A_2^*}$. Then $s = 2cR(1+g)$ where $c = \max_{i=1,2} \frac{M_i - m_i}{k}$ and $R = \frac{1}{\sqrt{3}}$. Assume that $k \geq \frac{2}{\sqrt{3}}(1+g)$, $g \in (0,1)$ and that $c = \frac{M_1 - m_1}{k}$.

For condition (i) we can rewrite s to verify

$$s = 2cR(1+g) = \frac{2(1+g)(M_1 - m_1)}{k\sqrt{3}} \leq M_1 - m_1 \leq \operatorname{diam}(Z).$$

Condition (ii) is straightforward. Let $\varepsilon \geq s$. Then U_ε is a cover with cover sets $U_{\xi,\varepsilon} = B(c\xi M_{A_2^*}, r)$ where $r = \frac{1}{2}\varepsilon$. We can simply choose $\varepsilon' = \varepsilon - s$ so that Condition (ii) follows by definition of the tower of lattice cover. Lastly, condition (iii) follows from the same argument as Equation (3.1). $\square$

Corollary 3.13. *A tower of A_2^*-lattice covers defined with k-intervals, overlap fraction g and generating matrix $M_{A_2^*}$ is $(3,s)$-good if $k \geq 3$.*

Theorem 3.14. *Consider the A_3^* lattice with generating matrix $M_{A_3^*}$. A tower of A_3^*-lattice covers $\mathcal{U}$ over compact topological space Z defined with k-intervals and overlap fraction g is $(3,s)$-good for $k \geq (1+g)\sqrt{5}$ and $g \in (0,1)$.*

Proof. Let $B = \prod_{i=1}^{3}[m_i, M_i]$ be a bounding box on Z. Then a tower of A_3^*-lattice covers is defined with $c = \max_{i=1,2,3} \frac{M_i - m_i}{k}$ and without loss of generality suppose $c = \frac{M_1 - m_1}{k}$. Then for $k \geq (1+g)\sqrt{5}$, we observe that

$$\operatorname{res}\mathcal{U} = s = 2cR(1+g) = \frac{2\sqrt{5}(1+g)(M_1 - m_1)}{2k} \leq M_1 - m_1 \leq \operatorname{diam}(Z),$$

resolving condition (i). For conditions (ii) and (iii), the proofs are identical to the proof of Theorem 3.12. $\square$

Corollary 3.15. *The tower of A_3^*-lattice covers defined with k-intervals, overlap fraction g and generating matrix $M_{A_3^*}$ is $(3,s)$-good for $k \geq 5$.*

Theorem 3.16. *Consider the A_n^* lattice with generating matrix $M_{A_n^*}$. A tower of A_n^*-lattice covers over a compact topological space Z, defined with k-intervals and overlap fraction g is $(3,s)$-good for $k \geq 4R$ and $g \in (0,1)$ where R is the covering radius of A_n^*.*

Proof. Let B be a bounding box for Z, and without loss of generality suppose $c = \max_{1 \leq i \leq n} M_i - m_i = M_1 - m_1$. Then for $k \geq 4R$ we have

$$\operatorname{res}(\mathcal{U}) = s = 2cR(1+g) = \frac{2R(1+g)(M_1 - m_1)}{k} \leq M_1 - m_1 \leq \operatorname{diam}(Z).$$

This proves condition (i) and conditions (ii) and (iii) follow similarly to Theorem 3.12. $\square$

It is not unexpected that the tower of A_n^*-lattice covers are (c, s)-good. Their construction, however, was completely general. Our results above yield bounds on parameters for computable covers, so in the experimental setting we will have stability if choosing within the correct range of parameters.

As we now have ranges for these computable covers, we extend our results further to verify that cluster covers that arise from using the clustering algorithm DBSCAN with a (c, s)-good cover is (c, s)-good.

Theorem 3.17. *Let $\mathcal{U} = \{U_\varepsilon\}_{\varepsilon \geq s}$ be one of the $(3, s)$-good covers defined above (cubical, cubical lattice, or lattice) with resolution $\operatorname{res}(\mathcal{U}) = s$ over a data set X. Define the cluster cover $\mathcal{C} = \{C_\varepsilon\}_{\varepsilon \geq s'}$ by clustering X subordinate to the indicated cover using DBSCAN with fixed parameters ϵ and MinPts. Then if there are no free border points when clustering X, $\mathcal{C}$ is a $(4, s')$-good cover where $s' \leq s$.*

Proof. This proof follows easily from the fact that $\mathcal{U}$ is a (c, s)-good cover. For condition (i), note the cluster cover $C_s = \{C_{p_\alpha,s}\}$ has cover sets $C_{p_\alpha,s} \subseteq U_{\alpha,p}$ for all $\alpha \in A$. This means $\operatorname{diam}(C_{p_\alpha,s}) \leq \operatorname{diam}(U_{\alpha,s}) = s$ for all $\alpha \in A$. By definition the resolution of $\mathcal{C}$ is $\operatorname{res}(\mathcal{C}) = \sup_{\alpha \in A} \operatorname{diam}(C_{p_\alpha,s}) = s' \leq s$. The tower $\mathcal{U}$ is (c, s)-good so we can conclude that $\operatorname{res}(\mathcal{C}) \leq \operatorname{res}(\mathcal{U}) = s \leq \operatorname{diam}(X)$.

Each cluster cover $C_\varepsilon = \{C_{p_\alpha,\varepsilon}\}$ for $\varepsilon \geq s$ contains cover sets $C_{p_\alpha,\varepsilon} \subseteq U_{\alpha,\varepsilon}$ implying that $\operatorname{diam}(C_{p_\alpha,\varepsilon}) \leq \operatorname{diam}(U_{\alpha,\varepsilon}) = \varepsilon$. This proves condition (ii).

Lastly for condition (iii), suppose that $O \subseteq X$ such that $\operatorname{diam}(O) \geq s'$. Let $C_{p_\alpha,s} \in C_s$ such that $C_{p_\alpha,s} \cap O \neq \emptyset$. Then for $x \in C_{p_\alpha,s} \cap O$ we have for any $o \in O$ that

$$||p_\alpha - o||_2 \leq ||p_\alpha - x||_2 + ||x - o||_2 \leq s' + \operatorname{diam}(O) \leq s + \operatorname{diam}(O) \leq 2\operatorname{diam}(O)$$

meaning that $O \subset C_{p_\alpha, 4\operatorname{diam}(O)} \in C_{4\operatorname{diam}(O)}$. Hence $\mathcal{C}$ is a $(4, s')$-good cover. $\square$

Our previous proofs relied on the fact that our cover sets were of uniform shape. Each $U_{\alpha,\varepsilon}$ was either a hypercube or ball with a center point c_α. However, for each cluster cover set $C_{p_\alpha,\varepsilon}$, the point p_α is a core point within the cluster, but there is not an obvious center point meaning we can only achieve the bound $||p_\alpha - x||_2 \leq s'$, which is different from the bound of Equation (3.1).

We conclude that an implementation for Multiscale 2-Mapper with DBSCAN clustering and any of the above tower of covers is topologically stable with respect to persistent homology.

Theorem 3.18. *Given a data set $X \subseteq \mathbb{R}^m$, let $f\colon X \to \mathbb{R}^n$ be continuous and let $\mathcal{U}$ and $\mathcal{W}$ be two towers of (cubical, cubical lattice or lattice) covers. Cluster X with DBSCAN using fixed parameters* MinPts *and ϵ. Then the persistence diagrams computed from the Multiscale 2-Mappers $D_k(\mathrm{MM}_2(f, \mathcal{C}_{\mathcal{U}}))$ and $D_k(\mathrm{MM}_2(f, \mathcal{C}_{\mathcal{W}}))$ are* 4*-multiplicatively interleaved.*

Proof. The result follows from Theorems 1.16 and 3.17. □

Achieving this stability is important, as it is rarely if ever the case that an optimal cover can be understood apriori. With well-defined persistent homology, users can choose parameters for 2-Mapper complexes which exhibit persistent topological features.

3.3 Algorithm for Multiscale 2-Mapper

While Multiscale Mapper as developed by Dey, Mémoli, and Wang [DMW] was an important theoretical advance, their results did not apply to data sets and thus have not previously not been implemented. The results of Bungula and Darcy [BD24] do apply to finite data sets, with DBSCAN as a choice of clustering function, though there was no implementation given with their manuscript. Motivated by the opportunity to enhance Mapper algorithms by choosing persistent features, we have now implemented these inspiring ideas.

There are many efficient libraries in Python for Topological Data Analysis, and we will build on these for our Multiscale 2-Mapper implementation. Our algorithm builds Multiscale 2-Mapper as a simplex tree [BM14], an object in `gudhi` which contains simplices σ at a filtration time t. Simplex trees are the main construction for simplicial filtrations in `gudhi`, where it is used for virtually all persistent homology implementations. Due to the structure of a simplex tree, one must modify the definition of Multiscale 2-Mapper so that the information across 2-Mapper complexes in the filtration are properly stored. Once a simplex is added to a simplex tree, it will persist for the rest of the filtration. However, each 2-Mapper complex in the filtration will have a different vertex structure based on the clustering algorithm DBSCAN. As seen in Figure 3.1, clusters can be created or collapse, and so to maintain a persistence in the Multiscale 2-Mapper we need to include some functionality across the 2-Mapper complexes in the filtration. We do this by constructing maps between vertex sets that map each vertex to it's most similar (based on cover set and data) counterpart in the next 2-Mapper complex. In certain cases, we additionally add simplices into the next 2-Mapper to maintain its homotopy type based on the constructed maps.

Let $X \subseteq \mathbb{R}^D$ be a data set with reference function $f\colon X \to \mathbb{R}^d$ and tower of covers $\mathcal{U}$. Cluster the preimage of the cover of X with DBSCAN. If there are no free border points, then there is a tower of cluster covers $\mathcal{C}$ on X induced by $\mathcal{U}$, and for a finite sequence $s = \varepsilon_0 < \varepsilon_1 < \cdots < \varepsilon_t$ we can write the Multiscale 2-Mapper $\mathrm{MM}_2(f, \mathcal{C})$ as the sequence,

$$M_2(f, \mathcal{C}_{\varepsilon_0}) \xrightarrow{c^*_{0,1}} M_2(f, \mathcal{C}_{\varepsilon_1}) \xrightarrow{c^*_{2,1}} \cdots \xrightarrow{c^*_{t,t-1}} M_2(f, \mathcal{C}_{\varepsilon_t}),$$

where each $M_2(f, \mathcal{C}_{\varepsilon_j})$ is the 2-Mapper complex constructed with cluster cover $\mathcal{C}_{\varepsilon_j}$. The maps $c^*_{j+1,j}$ are induced by the maps $c_{j+1,j}$ between cluster covers as arranged for by our choices of nested covers.

For the 2-Mapper complex $M_2(f, \mathcal{U}_{\varepsilon_j})$, DBSCAN clusters each cover set so that $f^*(U_{\alpha,\varepsilon_j}) = \bigcup_{i=0}^{m} C_{p_\alpha,i} \cup N_\alpha$, for each $\alpha \in A$. The classified clusters determined by DBSCAN are labeled as $C_{p_\alpha,i}$ and are elements of the cluster cover $\mathcal{C}_{\varepsilon_j}$. Points which categorized as noise are grouped into a noise cluster N_α. As seen in Figure 3.1(D), elements of a noise cluster can later be contained in a labeled cluster $C_{p_\alpha,l}$. Bungula and Darcy ignore the noise clusters in their results for Multiscale Mapper with DBSCAN, and as in Section 3.2 we will remove all noise clusters from our construction. Additionally, to guarantee that no free border points exist when clustering, we will fix MinPts $= 2$.

The key data in Multiscale 2-Mapper are the maps between simplicial complexes within the filtration. Attributes of each 2-Mapper complex vary throughout the filtration. In particular, the cluster structure of each cover set in $\mathcal{U}$ and the data contained in each cluster changes across each 2-Mapper complex in the filtration. Consider the 2-Mapper complex $M_2(f, \mathcal{C}_{\varepsilon_j})$ and let V_j be the its vertex set. Each node $n \in V_j$ is one cluster $C_{p_\alpha,\varepsilon_j}$, and we encode n as the tuple $n = (X_n, \alpha, p_\alpha)$ where $X_n \subseteq X$ is the subset of the data set in n, α is the $\mathcal{U}$ cover set label, and p_α is the cluster label for $C_{p_\alpha,\varepsilon_j}$. Define node maps $\varphi_j : V_j \to V_{j+1}$ where we map a node $n \in V_j$ to its best match in V_{j+1} defined by $\varphi_j((X_n, \alpha, p_\alpha)) = (X_m, \alpha, q_\alpha)$ such that $\frac{|X_n \cap X_m|}{|X_n \cup X_m|}$ is maximized.

One concern with these node maps is that they are not necessarily injective. Injectivity fails when there are two clusters $n_1, n_2 \in V_j$ that collapse into a single cluster, say n_1 as in Figure 3.1(B). In this case, this means that the 2-Mapper complex M_{j+1} does not contain a copy of n_2. Simplex trees, however, preserve vertices across an entire filtration, meaning n_2 will still remain in the simplex tree while not represented in M_{j+1}. To model this vertex collapse we instead insert the collapsed node n_2 into M_{j+1} creating a new simplicial complex M'_{j+1}. To preserve the homotopy type of M_{j+1} we also insert edges and 2-simplices depending on the adjacency of nodes with n_1 in the 2-Mapper complex M_j. This leads to two possible cases in our implementation for node collapse.

The first case, seen on the left of Figure 3.1, is when there is a single node collapse between 2-Mapper complexes. Clusters $(0, \alpha, p), (1, \alpha, q) \in M_j$ which belong to the same pullback cover set U_α are both connected to $(2, \beta, x) \in M_j$, a cluster of pullback cover set U_β. The map $c^*_{j+1,j} \colon M_j \to M_{j+1}$ shows a collapse of the U_α clusters. Algorithm 1 preserves the collapsed node $(1, \alpha, q) \in M_j$ by inserting it into M'_{j+1}. Because the node index can change, we insert this node as $(2, \alpha, q) \in M'_{j+1}$. To retain the homology of $H_*(M_{j+1})$, we insert an edge between $(0, \alpha, p)$ and $(2, \alpha, q)$ and insert a 2-simplex σ for every node adjacent to $(0, \alpha, p)$. In this case, $\sigma = (0, 1, 2) \in M'_{j+1}$ is included and we see that $M_{j+1} \sim M'_{j+1}$. The map $\varphi_j \colon V_j \to V'_{j+1}$ is defined as $\{0, 1, 2\} \mapsto \{0, 2, 1\}$.

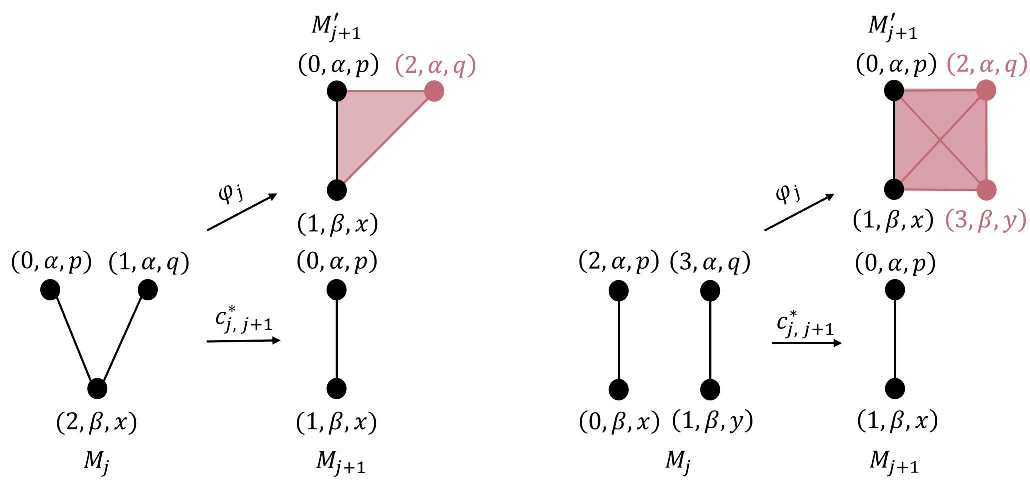

Figure 3.3. Cases for cluster collapse in Algorithm 1. Left: a cluster collapse for a single cover set. Right: a two cluster collapses on connected cover sets; also known as a double collapse.

The second case, seen on the right of Figure 3.1, occurs when two pullback cover sets U_α and U_β simultaneously collapse. For this case, we apply the single collapse procedure above, adding nodes $(2, \alpha, q), (3, \beta, y) \in M'_{j+1}$ and 2-simplices $(0, 1, 3), (1, 2, 3) \in M'_{j+1}$. Since node $(1, \beta, y)$ is adjacent to $(3, \alpha, q)$ in M_j, this adjacency is preserved in the simplex tree. This leaves two empty cycles $(0, 2, 3)$ and $(1, 2, 3)$ in M'_{j+1}. To retain the right homotopy type, we insert 2-simplices at these cycles so that $M_{j+1} \sim M'_{j+1}$ and define $\varphi_j \colon M_j \to M'_{j+1}$ as $\{0, 1, 2, 3\} \mapsto \{1, 3, 0, 2\}$.

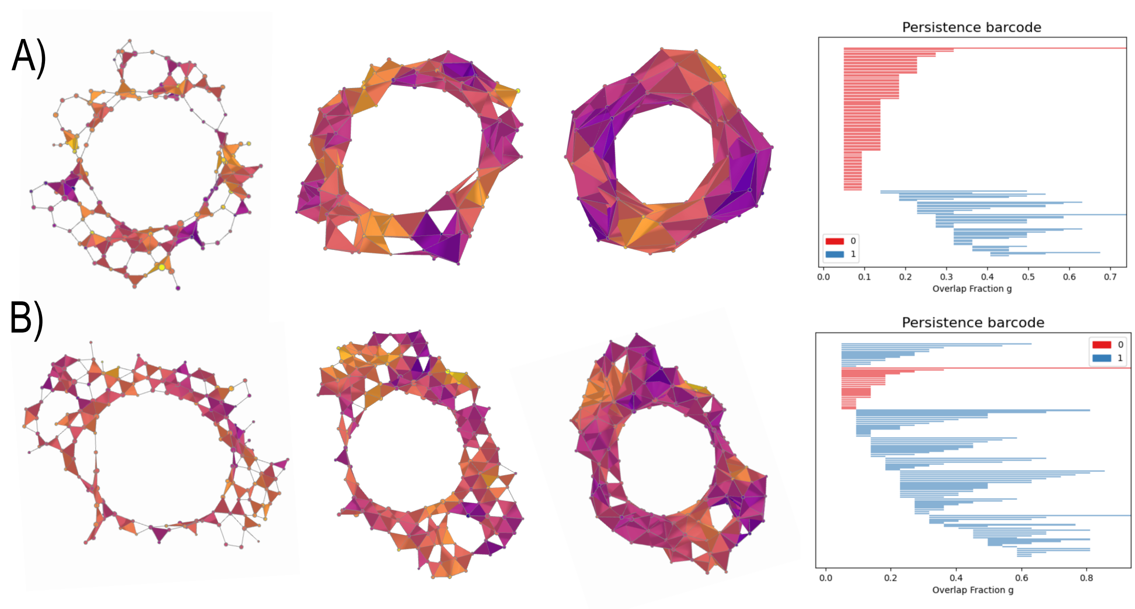

Figure 3.4. We compute two Multiscale 2-Mapper complexes and produce their resulting barcodes for a torus point cloud constructed from 500 points. Each Multiscale 2-Mapper is computed with a filtration of twenty 2-Mapper complexes. The lens function is coordinate projection to $\mathbb{R}^2$, and clustered with DBSCAN with MinPts $= 2$ and $\epsilon = 0.5$. Both towers of covers are initialized with 15 intervals and range the overlap fraction from 0.05 to 0.9. Shown are the 2-Mapper complexes in each filtration with overlap fractions 0.32, 0.54 and 0.72 (left to right). A) Multiscale 2-Mapper constructed from a tower of cubical covers. B) Multiscale 2-Mapper constructed from a tower of A_2^*-lattice covers.

Algorithm 1 constructs the maps φ_j and new 2-Mapper complexes M'_{j+1} with the `AlignMappers` function. To create the simplex tree we apply Algorithm 1 recursively across every consecutive pair of 2-Mapper complexes (M_j, M_{j+1}). The computational complexity for this algorithm is approximately $O((n-1)N^2)$ where n is the number of 2-Mapper complexes in the filtration of the Multiscale 2-Mapper and N is the average number of vertices per 2-Mapper complex. With this construction we can now compute persistence barcodes and persistence diagrams for a filtration of 2-Mapper complexes using methods found in `gudhi`. In the future we intend to improve the efficiency of this implementation.

As an example, we compute the persistence barcodes using the tower of cubical covers (Figure 3.4A) and tower of A_2^*-lattice covers (Figure 3.4B) for a sparse torus point cloud. Each filtration is initialized with the same parameters, and we see their resulting barcodes and 2-Mapper complexes across each Multiscale 2-Mapper. The A_2^*-lattice Multiscale 2-Mapper has many more cycles in its persistent barcode, and the cubical cover produces an accurate picture at high levels of overlap. But the overall structure of the A_2^* and cubical cover Multiscale 2-Mappers are similar. Theorem 3.18 asserts that these two Multiscale 2-Mappers are close with respect to the bottleneck distance, and when computed, the botteneck distance between these two persistence barcodes is 0.0894 in degree zero and 0.1789 in degree one.

Algorithm 1: Given a sequence of 2-Mapper complexes, without noise, $(M_1, ..., M_n)$ computed from a tower of covers $\mathcal{U}$, and clustering algorithm DBSCAN(MinPts=2, ϵ) on a data set X, compute the Multiscale 2-Mapper.

def `AlignMappers` *(M_i, M_{i+1})*:
    Let V_i, V_{i+1} be vertex sets of M_i and M_{i+1}, respectively.
    Let J be an empty $|V_i| \times |V_{i+1}|$ matrix.
    for $(n, m) \in V_i \times V_{i+1}$**:**
        Let $n = (X_n, \alpha, p_\alpha)$ and $m = (X_m, \beta, q_\beta)$.
        if $\alpha = \beta$**:**
            $J_{n,m} = \frac{|X_n \cap X_m|}{|X_n \cup X_m|}$
    Let $\varphi_i(n) = \text{argmax}_m J_{n,m}$.
    for $m \in V_{i+1}$**:**
        if $|\varphi_i^{-1}(m)| \geq 2$**:**
            Let $n_M = \text{argmax}_{n \in \varphi^{-1}(m)} |X_n|$.
            for $n \in \varphi_i^{-1}(m)$, with $n \neq n_M$**:**
                Add vertex $n \in V_{i+1}$, and set $\varphi_i(n) = n$.
                Add edge $(n, m) \in M_{i+1}$.
                for vertices $o \in V_i$ adjacent to n**:**
                    Add simplex $(o, n, m) \in M_{i+1}$.
                for vertex pairs (o_1, o_2) adjacent to n**:**
                    Add simplex $(o_1, o_2, n) \in M_{i+1}$.
    for node pairs $(n_1, n_2) \in V_i \times V_i$**:**
        if there is an edge between n_1 and n_2 and $n_1, n_2 \in M_i$ "collapse", ie. $n_1, n_2 \notin M_{i+1}$.**:**
            Let n_1' and n_2' be nodes in V_{i+1} that n_1 and n_2 collapse to, respectively.
            Add simplices $(n_1', n_2', n_1), (n_1', n_2', n_2) \in M_{i+1}$.
    return φ_i, M_{i+1}

CHAPTER 4

APPLICATIONS

We put 2-Mapper and Multiscale 2-Mapper to use as new tools for understanding data sets through 2-Mapper complexes. Our main strategy is to use the persistence barcode for Multiscale 2-Mapper to infer choices of cover informed by Betti-1 values. We first study the shape analysis capabilities of 2-Mapper in Section 4.1 to understand if there is an optimal cover choice to reproduce Betti-1 for various point clouds sampled from simple manifolds. Next in Section 4.2 we apply our methods to point clouds generated from dynamical systems that describe weather regimes, and we create a new method called *Trajectory Mapper* which can approximate common trajectories that arise in these systems.

4.1 Shape Analysis

We first apply our methods to synthetic data sets generated to simulate manifolds. We apply Multiscale 2-Mapper to two data sets (stochastic triangle and Klein bottle data sets) and present its persistence barcode and selected 2-Mapper complexes from the filtration. Our experiments reveal that one can obtain the correct Betti-1 for these simulated data sets.

Stochastic Triangle

In contrast with the (static) 2-Mapper calculations for stochastic triangles from Section 2.3, we can compute the Multiscale 2-Mapper for a randomly chosen stochastic triangle, seen in Figure 4.1. We project this data into $\mathbb{R}^2$ using PCA, and choose DBSCAN with MinPts $= 2$ and $\epsilon = 0.5$. For cover we choose the cubical tower of covers containing a sequence of ten covers, with 10 intervals, overlap fraction $0.1 \leq g \leq 0.7$. With these parameters we compute ten individual 2-Mapper complexes with the overlap fraction increasing by 0.06 across the filtration. In Figure 4.1B and C we see as g increases we can see the number of 1-cycles decrease. This makes intuitive sense, since the overlap fraction increases there is a higher possibility of a 2-simplex to appear in any give empty cycle. Once $g \geq 0.3$, all of these smaller 1-cycles disappear and we are left with the one prominent circle cycle, resembling the shape of the stochastic triangle. Therefore, through persistence we can identify that correct Betti-1 will be inferred when $g \geq 0.3$. Due to Theorem 3.18, this also means we should be able to infer the correct Betti-1 for any cover type.

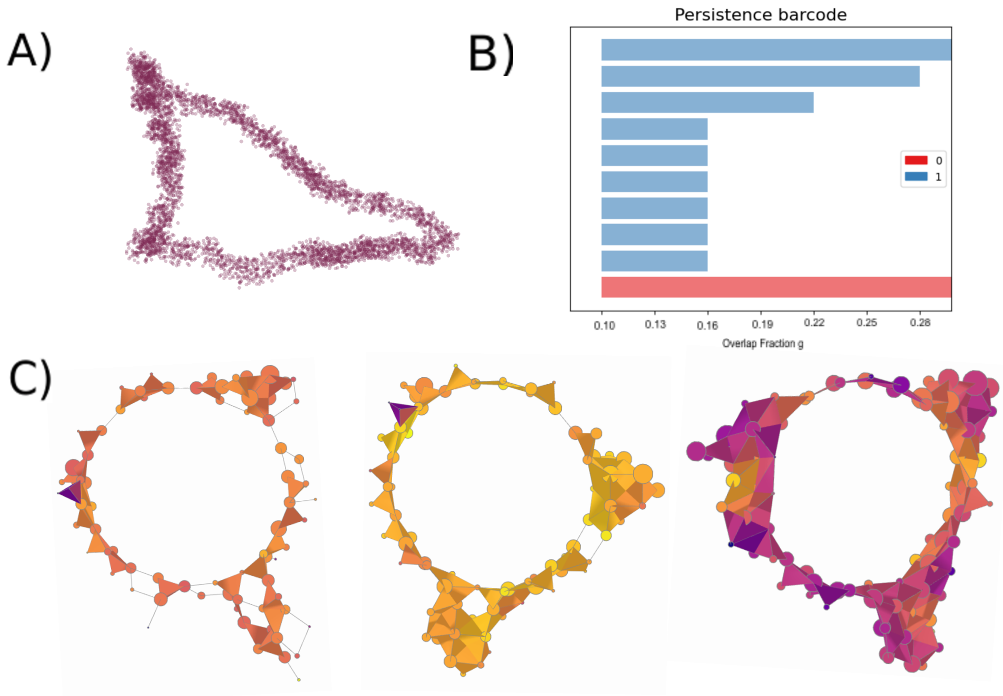

Figure 4.1. Multiscale 2-Mapper of a Stochastic Triangle. A) Point cloud of a stochastic triangle projected to $\mathbb{R}^2$. B) The persistence barcode of the multiscale 2-Mapper. C) 2-Mapper complexes in the filtration of the Multiscale 2-Mapper. From left to right we have 2-Mapper complexes with overlap fractions 0.1, 0.22, and 0.34.

Klein Bottle

The Klein bottle data set, obtained from Gudhi [Mar+14], consists of 15,875 points sampled from a Klein bottle embedded in $\mathbb{R}^5$. We filter this data set using projection onto the first two coordinates and cluster using DBSCAN with $\epsilon = 0.5$ and MinPts $= 2$. For cover, we choose the cubical tower of covers containing a sequence of ten covers with 17 intervals, overlap fraction $0.1 \leq g \leq 0.6$. With these parameters we construct a Multiscale 2-Mapper filtered over g where g increases by 0.05 between 2-Mapper complexes along the filtration. In Figure 4.2C, as g increases the 2-Mapper complex better resembles the Klein bottle, and there are some 1-cycles that persist for the entirety of the filtration.

For small values of g ($g = 0.15$), the 2-Mapper complex is not annular and does not even resemble the shape of the Klein bottle. However, once the overlap fraction exceeds 0.35 the Betti-1 value is 1, and is not representative of Betti-1 for a Klein bottle. These two observations we see are likely due to the twist in the data set. Small overlap fractions result in smaller portions of the cover with (at least) 3-fold intersections. This leads to less edges and 2-simplices and hence a more sparse 2-Mapper complex. Contrarily, large overlap fractions can remove geometric nuance in the data set though the generation of too many 2-simplices. This results in the death of 1-cycles and the reduction of Betti-1. In particular, this occurs in the twist of the Klein bottle, seen the the top left of Figure 4.2A. Because of the Klein bottle's self intersection, 2-Mapper constructs extra simplices in this area and this kills one of the Betti-1 representatives. By using the persistence barcode in Figure 4.2, we would choose $0.3 \leq g \leq 0.35$ in our filtration as the best representative 2-Mapper complex for the Klein bottle.

4.2 Climate Regimes and Dynamical Systems

Climate regimes and their mathematical modeling largely involve chaotic dynamical systems. Pseudo-periodic trajectories often occur, but they can be difficult to find analytically. Recently, scientists have turned to TDA to analyze these complex systems [Str+23]. Dynamical systems which model climate regimes are usually chaotic, and their visualization can be crucial to identifying a system's behavior. Previous research has used Mapper to identify changes in the state of a dynamical system representing neural signaling [Sag+18], and we intend to use similar techniques geared toward climate regimes. If we construct a Mapper graph from dynamical system data – for example, the Charney-Devore system – the graph would contain loops identifying some cyclical behavior in the system but could also include extraneous loops arising from spatial properties of the point-cloud. We developed *Trajectory Mapper*, an enhancement of 2-Mapper, to distinguish temporally relevant cycles in this setting, and to analyze time-series data more generally.

Definition 4.1. We say a data set is *temporal* if it is a subset of some $Z \times \mathbb{R}$, with an ordering convention that for points (x_i, t_i) we have $t_i < t_{i+1}$ for all i.

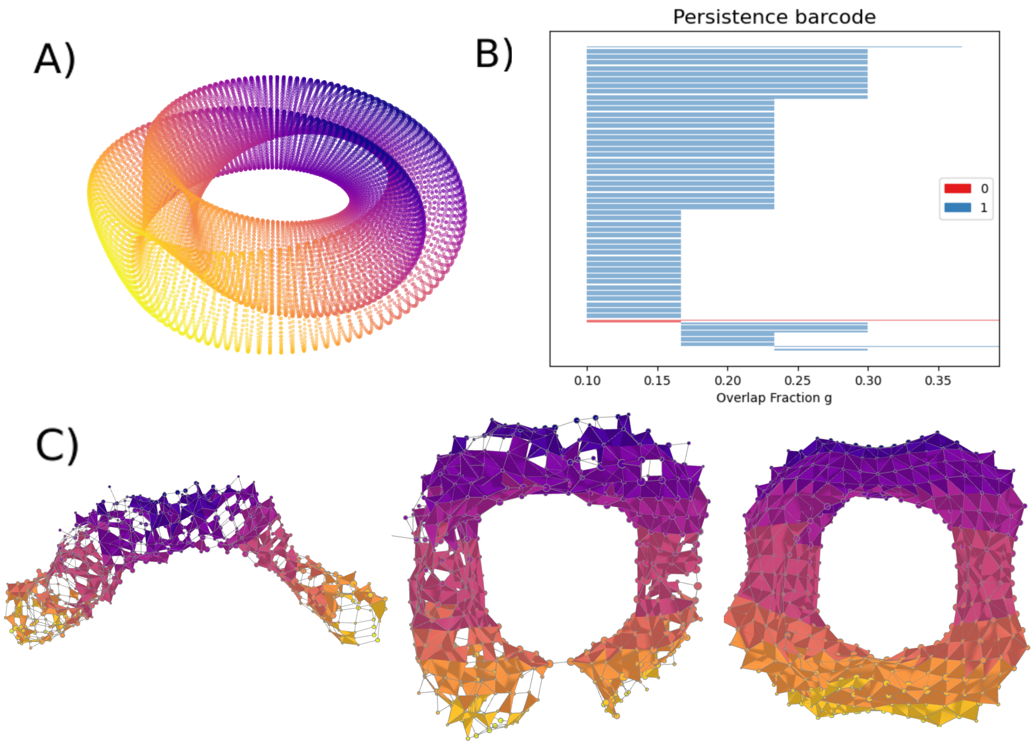

Figure 4.2. Multiscale 2-Mapper of the Klein bottle. A) The point cloud of the Klein bottle projected to $\mathbb{R}^3$. B) The persistence barcode of the multiscale 2-Mapper. C) 2-Mapper complexes in the filtration of the multiscale 2-Mapper. From left to right are the 2-Mapper complexes with overlap fractions 0.15, 0.2 and 0.3.

Let $M(f,\mathcal{U}) = (V, E)$ be a Mapper graph or 2-Mapper complex with vertices V and edges E of temporal data set $X \subseteq \mathbb{R}^n \times \mathbb{R}$. The *Trajectory Mapper graph* of X is the directed graph $TM(f,\mathcal{U}) = (\tilde{V}, \tilde{E})$ whose edges $\tilde{E} \subseteq E$ witness the temporal feature, in that for all $e = (v_1, v_2) \in \tilde{E}$ there exists a point $x_i \in v_1$ and $x_{i+1} \in v_2$ for some i.

The Trajectory Mapper graph is a subgraph of the original Mapper graph or 2-Mapper complex, consisting of edges that witness a temporal relationship between clusters. We encode Trajectory Mapper as a weighted directed graph where edges represent the direction with respect to time, and the weights for each edge are equal to the number of pairs of points in the edge witnesses. The weights help us compute path trajectories in the directed graph which are approximations of trajectories in the dynamical system, see Figures 4.4 and 4.6. As a novel technique, Trajectory Mapper has only been tested on classical dynamical systems – Lorentz model and Charney-deVore – used in climate science for benchmarking, though in current work we are applying it to data sets of current interest.

Lorentz '96

We first apply Trajectory Mapper to the Lorentz '96 Model, a chaotic dynamical system used to model atmospheric weather regimes [Str+23, Appendix B.1.3] which is a data set of 600 points in $\mathbb{R}^8$. In Figure 4.3A, we see the data set projected in $\mathbb{R}^3$ using principle component analysis [Hot36]. Here the points are connected by edges defined through the orbit structure of the dynamical system. We compute the Multiscale 2-Mapper using this PCA filter, a tower of cubical covers consisting of 8 covers, and use DBSCAN with MinPts = 2 and $\epsilon = 0.5$.

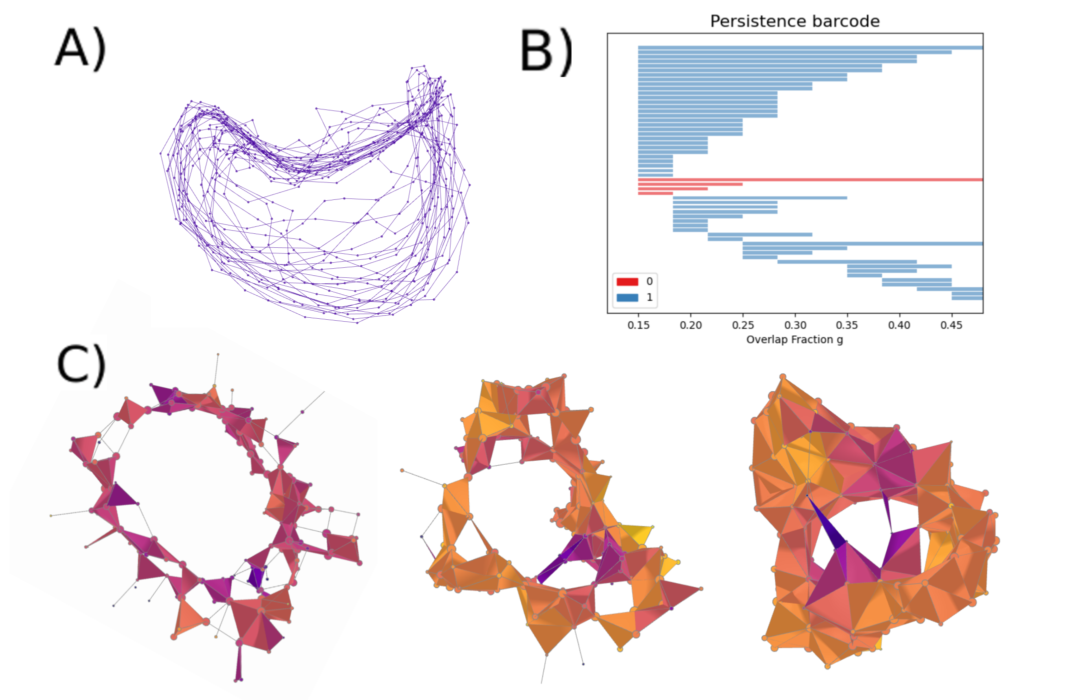

Figure 4.3. A) The Lorentz '96 dynamical system with $F = 5$ projected to $\mathbb{R}^3$ via PCA. We compute Multiscale 2-Mapper of the Lorentz model using a tower of cubical covers containing 8 covers with overlap fractions ranging from $g = 0.15$ to $g = 0.45$. B) The persistence barcode computed from this Multiscale 2-Mapper. C) We show the 2-Mapper complexes for $g = 0.21,\ 0.31,\ 0.45$ (left to right).

As we progress through the filtration, Figure 4.3C, we see the inclusion of more 2-simplices in the 2-Mapper complex. In the barcode, Figure 4.3B, there are some 1-cycles which persist for most of the filtration, and some moderately persistent cycles at different overlap values. We choose the representative from the Multiscale 2-Mapper with $g = 0.31$ (Figure 4.3C-center).

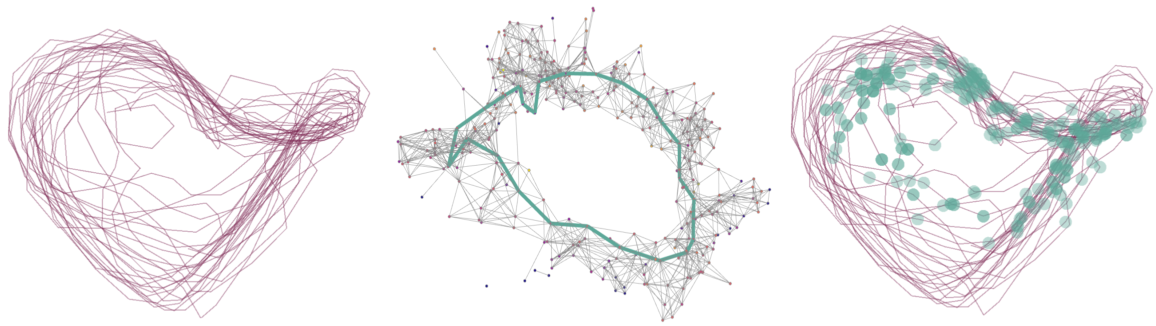

Figure 4.4. From the Lorentz '96 dynamical system (left) we compute the Trajectory Mapper. Contained in this directed graph is a single cycle (center) which we can extrapolate and visualize the consecutive times points from this cycle to see an "approximate" trajectory (right).

Now, with our chosen overlap fraction, we compute the Trajectory Mapper seen in Figure 4.4 which, as a directed graph, admits a single directed cycle, shown in the center sub-figure of Figure 4.4. Of course, given a directed graph there exist a multitude of directed paths between its vertices, however it an interesting result to only have one cycle admitted from this directed graph. Using this cycle, we can extrapolate the subset of data in the dynamical system that lies in this cycle, Figure 4.4C, by looking at the data in each vertex of the cycle in the Trajectory Mapper. With Trajectory Mapper, one can easily represent dynamical systems and visualize their trajectories. While the Lorentz '96 system is a trivial example, the identification of interesting trajectories is more apparent with Charney-deVore.

Charney-deVore

Next we apply Trajectory Mapper to the Charney-deVore dynamical system [Str+23, Appendix B.1.2]. This model describes barometric pressure and is a common benchmark in climate science. Within the point cloud – see Figure 4.5A or Figure 4.6(left) – an inner cyclonic region and larger cycles on the outside of the point cloud. Within the inner cylinder the system is low velocity and fairly stable. Once a point reaches the end of that cylinder the regime changes and it moves along the outside region at high velocity before re-entering the inner cylinder. The shape of the point cloud visually is difficult to interpret the existence of cyclic behavior, however we will see that we can visualize some trajectories with Trajectory Mapper.

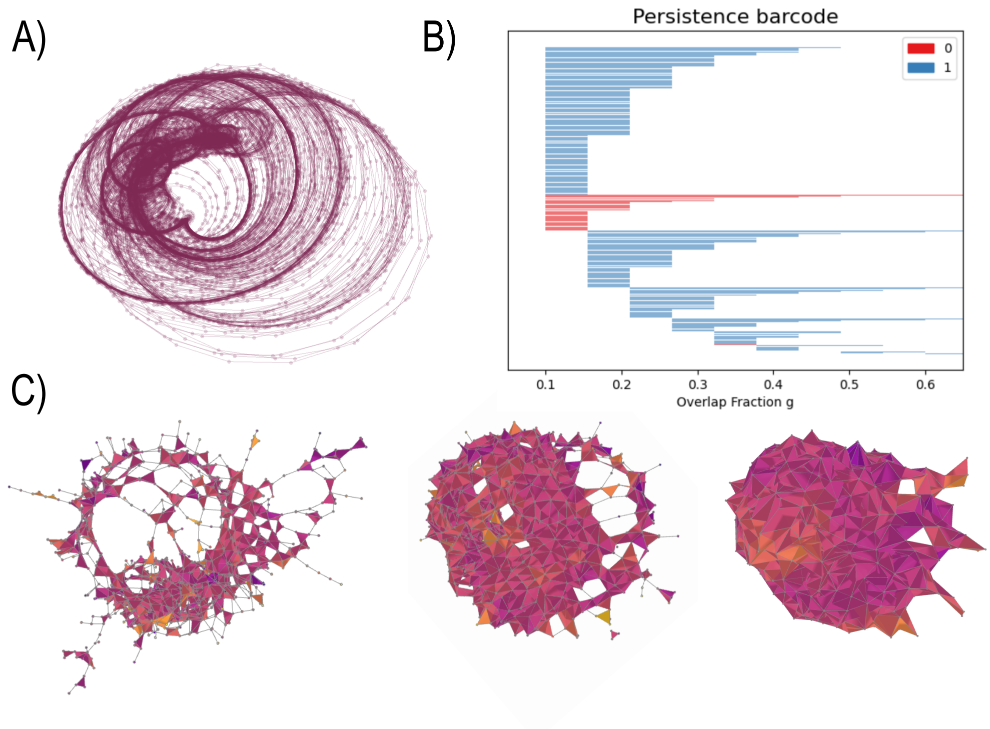

Figure 4.5. A) The Charney-deVore dynamical system projected to $\mathbb{R}^3$ via PCA. We compute Multiscale 2-Mapper using a tower of cubical covers containing ten covers with overlap fractions ranging from $g = 0.1$ to $g = 0.6$. B) The persistence barcode computed from the Multiscale 2-Mapper. C) We show 2-Mapper complexes for $g = 0.15, 0.32, 0.54$ (left to right).

First, we generate a 10,000 point data set of the Charney-deVore dynamical system. We then compute the Multiscale 2-Mapper of this system to identify a proper 2-Mapper complex to compute an approximate trajectory using Trajectory Mapper. This produces a barcode, seen in Figure 4.5B, which we see is quite noisy. For point cloud there is an overall separation in the data between the high and low velocity regimes, and we want to choose a 2-Mapper complex which visualizes this separation. Looking at the selected 2-Mapper complexes with the Multiscale

2-Mapper, in Figure 4.5C, the 2-Mapper complex computed with $g = 0.32$ best visualizes this while still being highly connected. We choose to use this 2-Mapper complex for computing approximate trajectories.

In Figure 4.6 we see the Trajectory Mapper highlighted in green within the 1-skeleton of the 2-Mapper complex. From that we observe the a path within the Trajectory Mapper graph (in purple). We note that this path does in fact follow the helicoidal movements from the original dynamical system, and this is more apparent when viewing the data which lies in this trajectory.

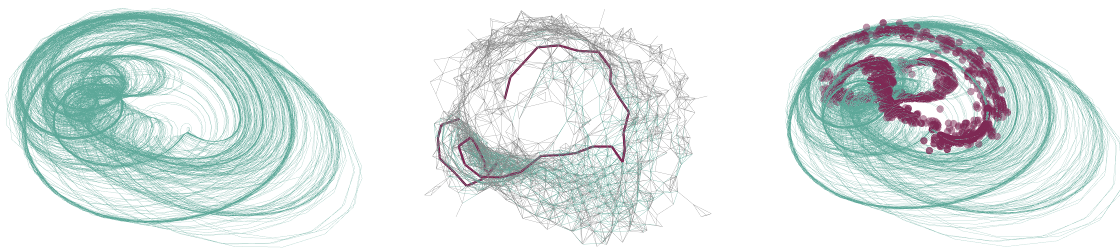

Figure 4.6. The Charney-deVore dynamical system (left). We compute its Trajectory Mapper and find a directed path in purple (center). From this path we can approximate a well traversed trajectory within the original dynamical system (right).

APPENDIX A

LATTICES

Definition A.1 (Lattice)**.** Given n linearly independent vectors $v_1, ..., v_n \in \mathbb{R}^m$, we define the *lattice* generated by this set of vectors as $\Lambda = \{\sum_{i=1}^n a_i v_i \; : \; a_i \in \mathbb{Z}\}$. We call $\{v_1, ..., v_n\}$ the *basis* of the lattice.

Given a basis of a lattice Λ, we define the *generating matrix* M of Λ as

$$M = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = \begin{bmatrix} v_{11} & \cdots & v_{1m} \\ v_{21} & \cdots & v_{2m} \\ \vdots & \ddots & \vdots \\ v_{n1} & \cdots & v_{nm} \end{bmatrix}.$$

Every vector $w \in \Lambda$ can be written as $w = \xi M$ for some $\xi \in \mathbb{Z}^n$. The matrix $A = MM^T$ is called the *Gram Matrix* of Λ. We define the *determinant* of Λ as

$$\det \Lambda \equiv \det := \det A.$$

The determinant describes the *size* of the lattice. If we take *Voronoi cells* at each point P_i in the lattice, defined

$$V(P_i) = \{x \in \mathbb{R}^n \mid d(x, P_i) \leq d(x, P_j) \; \forall j\},$$

these cells would be congruent and have volume equal to $\sqrt{\det}$.

An interesting problem about lattices is the **sphere covering problem**.

Question. *For spheres centered at lattice points, on average, what is the smallest number of spheres that contain a fixed point in space?*

The answer to this question is determined by the *thickness* Θ of a lattice Λ where

$$\Theta = \frac{V_n R^n}{\sqrt{\det}}.$$

The thickness is determined by the *covering radius* R of Λ, defined

$$R = \sup_{x \in \mathbb{R}^n} \inf_{P \in \Lambda} d(x, P),$$

and V_n is the volume of the n-unit ball

$$V_n = \begin{cases} \frac{\pi^{2k}}{k!}, & 2k = n \\ \frac{2(k!)(4\pi)^k}{(2k+1)!}, & 2k+1 = n \end{cases}.$$

Note that spheres of radius R are the smallest such balls that will cover $\mathbb{R}^n$.

With the basic definitions for lattices, we can now define dual lattices.

Definition A.2 (Dual Lattice)**.** Let Λ be an n-dimensional lattice with generating matrix M and Gram matrix A. We define the *dual lattice*, $\Lambda^* = \mathrm{Hom}(\Lambda, \mathbb{Z}) = \{x \in \mathbb{R}^n \mid x \cdot u \in \mathbb{Z},\ u \in L_n\}$.

If Λ has basis $v_1, ..., v_n$ we can choose canonical basis $v_1^*, ..., v_n^*$ for Λ^* such that $v_i^*(v_j) = \delta_{ij}$. The dual basis for Λ^* will also construct the generating matrix $(M^{-1})^T$ and Gram matrix A^{-1}. Thus the determinant of the dual lattice is $\det(\Lambda^*) = \det(\Lambda)^{-1}$.

Lastly, we can scale any lattice Λ by some $c \in \mathbb{R}$. The table below shows how scaling affects is properties.

Property	Λ	$c\Lambda$
Generator Matrix	M	cM
Gram Matrix	A	$c^2 A$
Determinant	det	c^{2n} det
Covering Radius	R	cR
Thickness	Θ	Θ
Dual Lattice	Λ^*	$c^{-1}\Lambda^*$

Table A.1. The effects of scaling a lattice Λ.

The primary lattice we are most concerned with is A_n^*. For $n \leq 5$, it is proven that A_n^* is the thinnest lattice covering. Additionally, A_n^* is the thinnest known lattice covering for $n \leq 23$ [Con+13, Chapter 2, Section 1.3]. To understand A_n^*, we will first define the root lattice A_n.

$$A_n = \left\{ (x_0, ..., x_n) \in \mathbb{Z}^{n+1} \;\middle|\; \sum_{i=0}^{n} x_i = 0 \right\},$$

which uses $n + 1$ coordinates to define an n-dimensional lattice. Thinking of A_n lying in the hyperplane $\sum x_i = 0$ in $\mathbb{R}^{n+1}$ yields a nice representation of the lattice where the i-th node is the basis vector $-e_i + e_{i+1}$ for $0 \leq i \leq n$. This gives us a generating matrix

$$M_{A_n} = \begin{bmatrix} -1 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & -1 & 1 & \cdots & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdots & \cdot & \cdot \\ 0 & 0 & 0 & 0 & \cdots & -1 & 1 \end{bmatrix}.$$

The dual lattice to A_n is

$$A_n^* = \bigcup_{i=0}^{n} [i] + A_n,$$

where

$$[i] = \left(\frac{i}{n+1}, ..., \frac{i}{n+1}, \frac{-j}{n+1}, ..., \frac{-j}{n+1} \right).$$

We note that $\frac{i}{n+1}$ occurs j times and $\frac{-j}{n+1}$ occurs i times with $i + j = n + 1$. A_n^* has a generating matrix

$$M_{A_n^*} = \begin{bmatrix} 1 & -1 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & -1 & 0 & \cdots & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdots & \cdot & \cdot \\ 1 & 0 & 0 & 0 & \cdots & -1 & 0 \\ \frac{-n}{n+1} & \frac{1}{n+1} & \frac{1}{n+1} & \frac{1}{n+1} & \cdots & \frac{1}{n+1} & \frac{1}{n+1} \end{bmatrix},$$

derived as the inverse the generating matrix of A_n above. However, A_n^* has two nice representations for dimensions $n \leq 3$. This is because $A_1^* \cong A_1 \cong \mathbb{Z}$, $A_2^* \cong A_2$ and $A_3^* \cong D_3^*$. Due to these special equivalencies, we have alternative generating matrices for A_2^* and A_3^*, respectively:

$$M_{A_2^*} = \begin{bmatrix} 1 & 0 \\ \frac{-1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}, \quad \text{and} \quad M_{A_3^*} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 1 & 1 & 1 \end{bmatrix}.$$

We then have the following properties for each of the lattices with their respective generating matrices.

Property	A_n^*	A_2^*	A_3^*
Generating Matrix	$M_{A_n^*}$	$M_{A_2^*}$	$M_{A_3^*}$
Determinant	$\frac{1}{n+1}$	$\frac{3}{4}$	16
Covering radius	$\sqrt{\frac{n(n+2)}{12(n+1)}}$	$\frac{1}{\sqrt{3}}$	$\frac{\sqrt{5}}{2}$
Thickness	$V_n\sqrt{n+1}\left(\frac{n(n+2)}{12(n+1)}\right)^{n/2}$	$\frac{2\pi}{3\sqrt{3}}$	$\frac{5\pi\sqrt{5}}{24}$

Table A.2. Attributes for A_n^* lattices.

APPENDIX B

2-MAPPER IMPLEMENTATION

We compute 2-Mapper using Algorithm 2. With the standard Mapper graph expressed as a tuple of vertices V and edges E, $M(f,\mathcal{U}) = (V, E)$, we can find all triples of nodes $(n_1, n_2, n_3) \in V$ whose subgraph is complete – that is, the triangles in M. Each node n_i corresponds to a cluster of data $X_{n_i} \subseteq X$, and so we can identify if we should insert a 2-simplex into the triangle (n_1, n_2, n_3), so long as the intersection of data $\bigcap_{i=1}^{3} X_{n_i}$ is non-empty. We find all such 2-simplices, thus computing the 2-dimensional nerve, and use them to complete the 2-Mapper complex $M_2(f,\mathcal{U})$.

We implement by building on `giotto-tda`, whose plotting functions use `plotly` [Inc15]. The Mapper graph M is a `3dScatter` plot, and we insert 2-simplices using the `3dMesh` plot. Finding and inserting 2-simplices in our implementation is approximately $O((d^2+1)n)$ where d is the average vertex degree in the 2-Mapper complex and n is the number of vertices. We color each 2-simplex as the average color between it's colored nodes. Additionally, we can weight the opacity of the simplex, or the thickness of each edge, by number of data points in the intersection using the boolean parameters `fancy_simplices` or `fancy_edges`, see Figure B.1.

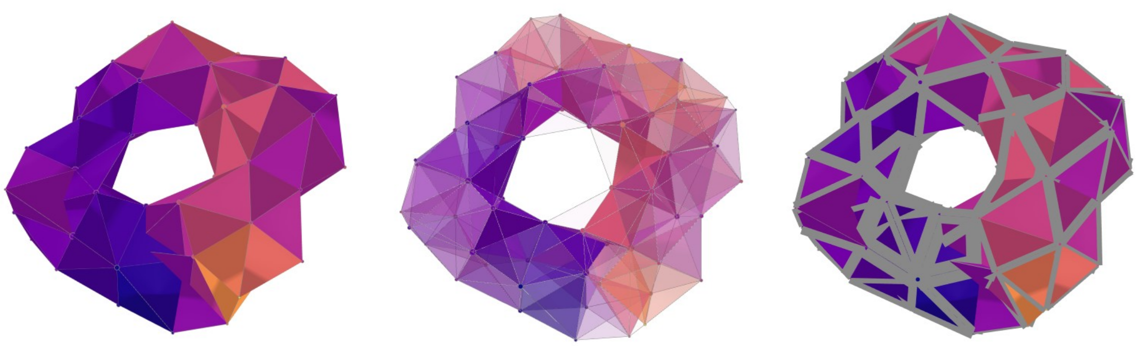

Figure B.1. 2-Mappers constructed from a torus point cloud using no additional parameters (left) and the parameters `fancy_simplices` (center) and `fancy_edges` (right).

Algorithm 2: For a data set X and a choice of given a filter function f, cover $\mathcal{U}$ of $f(X)$, and clustering algorithm, assume the Mapper graph $M = (V, E)$ has been constructed, based on clusters $X_v \subset X$, over $v \in V$. We additionally provide information on the plotting a 2-Mapper with boolean parameters *fancy edges* and *fancy simplices*.

def `list_2simplices` *(M)*:
    Let $V_\Delta \subset V \times V \times V$ be the subset of triangles in M.
    Let S denote the intended list of 2-simplices.
    Let S_X denote the list of data contained in each 2-simplex.
    Let $S_{|X|}$ denote the list of simplex weights.
    for $(n_1, n_2, n_3) \in V_\Delta$**:**
        if $X_{n_1} \cap X_{n_2} \cap X_{n_3} \neq \emptyset$**:**
            $(n_1, n_2, n_3) \in S$.
            $X_{n_1} \cap X_{n_2} \cap X_{n_3} = X_{(n_1,n_2,n_3)} \in S_X$.
            $|X_{(n_1,n_2,n_3)}| \in S_{|X|}$.
    return $S, S_X, S_{|X|}$

def `plot_2mapper` *(M,* `fancy_edges`*,* `fancy_simplices`*)*:
    Let S, S_X, $S_{|X|}$ = `list_2simplices` (M)
    Let E, $E_{|X|}$ be the set of edges and edge weights of M.
    Let P_M be the figure of Mapper graph M.
    if `fancy_edges`**:**
        for $w \in E_{|X|}$**:**
            Create `Scatter3d` plot with edge thickness equal to w.
            Add this plot to P_M.
    if `fancy_simplices`**:**
        for $w \in S_{|X|}$**:**
            Create `3dMesh` plot with triangle opacity set to
            $o = \frac{w - \min S_{|X|}}{\max S_{|X|} - \min S_{|X|}}$, for entries in S_X with weight w.
            Add this plot to P_M.
    if *not* `fancy_simplices`**:**
        Create `3dMesh` plot with entries in S_X.
        Add this plot to P_M.
    return P_M

REFERENCES CITED

[Alv+24] E. Alvarado et al., *G-Mapper: Learning a Cover in the Mapper Construction*, arXiv: 2309.06634 `[cs.LG]`, 2024.

[Amé+23] E. Amézquita et al., *Genomics data analysis via spectral shape and topology*, PLOS ONE **18** (2023), e0284820, DOI: 10.1371/journal.pone.0284820.

[BD24] W. Bungula and I. Darcy, *Bi-Filtration and Stability of TDA Mapper for Point Cloud Data*, arXiv: 2409.17360 `[math.AT]`, 2024.

[BM14] J.-D. Boissonnat and C. Maria, *The Simplex Tree: An Efficient Data Structure for General Simplicial Complexes*, Algorithmica **70** (2014), no. 3, 406–427, DOI: 10.1007/s00453-014-9887-3.

[Bor48] K. Borsuk, *On the imbedding of systems of compacta in simplicial complexes* (ENG), Fund. Math. **35** (1948), no. 1, 217–234.

[Bui+20] Q.-T. Bui et al., *F-Mapper: A Fuzzy Mapper clustering algorithm*, Knowledge-Based Systems **189** (2020), 105107, DOI: https://doi.org/10.1016/j.knosys.2019.105107.

[CAR+05] G. CARLSSON et al., *PERSISTENCE BARCODES FOR SHAPES*, International Journal of Shape Modeling **11** (2005), no. 02, 149–187, DOI: 10.1142/S0218654305000761, eprint: https://doi.org/10.1142/S0218654305000761.

[Car09] G. Carlsson, *Topology and Data*, Bulletin of The American Mathematical Society - BULL AMER MATH SOC **46** (2009), 255–308, DOI: 10.1090/S0273-0979-09-01249-X.

[CEH07] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer, *Stability of Persistence Diagrams*, Discrete & Computational Geometry **37** (2007), no. 1, 103–120, DOI: 10.1007/s00454-006-1276-5.

[Cha+08] F. Chazal et al., *The Stability of Persistence Diagrams Revisited*, 2008.

[CM21] M. Carrière and B. Michel, *Statistical analysis of Mapper for stochastic and multivariate filters*, arXiv: 1912.10742 `[math.AT]`, 2021.

[CMO17] M. Carrière, B. Michel, and S. Oudot, *Statistical Analysis and Parameter Selection for Mapper*, arXiv: 1706.00204 `[cs.CG]`, 2017.

[Con+13] J. Conway et al., *Sphere Packings, Lattices and Groups*, Grundlehren der mathematischen Wissenschaften, Springer New York, 2013.

[CV21a] G. Carlsson and M. Vejdemo-Johansson, *Topological Data Analysis with Applications*, Cambridge University Press, 2021, DOI: 10.1017/9781108975704.

[CV21b] Y. Chen and I. Volić, *Topological data analysis model for the spread of the coronavirus*, 2021.

[CZW21] N. Chalapathi, Y. Zhou, and B. Wang, *Adaptive Covers for Mapper Graphs Using Information Criteria*, 2021 IEEE International Conference on Big Data (Big Data), 2021, 3789–3800, DOI: 10.1109/BigData52589.2021.9671324.

[Del34] B. Delaunay, *Sur la sphère vide. A la mémoire de Georges Voronoı*, Izv. Math. (1934), no. 6.

[DMW] T. K. Dey, F. Mémoli, and Y. Wang, *Multiscale Mapper: Topological Summarization via Codomain Covers*, Proceedings of the 2016 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 997–1013, DOI: 10.1137/1.9781611974331.ch71, eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9781611974331.ch71.

[DMW17] ———, *Topological Analysis of Nerves, Reeb Spaces, Mappers, and Multiscale Mappers*, CoRR **abs/1703.07387** (2017), arXiv: 1703.07387.

[ELZ02] Edelsbrunner, Letscher, and Zomorodian, *Topological Persistence and Simplification*, Discrete & Computational Geometry **28** (2002), no. 4, 511–533, DOI: 10.1007/s00454-002-2885-2.

[Est+96] M. Ester et al., *A density-based algorithm for discovering clusters in large spatial databases with noise*, Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD'96, AAAI Press, Portland, Oregon, 1996, 226–231.

[Fri+25a] H. Fritze et al., *A forest is more than its trees: haplotypes and ancestral recombination graphs*, Submitted to Genetics, Under 2nd Review (2025).

[Fri+25b] H. Fritze et al., *Faithful Reeb Graph Reconstruction of a Tectonic Subduction Zone from Earthquake Hypocenters*, Accepted to ATMCS11. To be published in LaMatematica (2025).

[Fri25] H. Fritze, *Multiscale 2-Mapper – Exploratory Data Analysis Guided by the First Betti Number*, Accepted to ATMCS11. To be published in LaMatematica (2025).

[Hot36] H. Hotelling, *Relations Between Two Sets of Variates*, Biometrika **28** (1936), no. 3/4, 321–377.

[Hus+22] A. Husain et al., *Mappering Mecklenburg County: Exploring Census Data for Potential Communities of Interest*, Research in Computational Topology 2, Springer International Publishing, Cham, 2022, 245–264, DOI: 10.1007/978-3-030-95519-9_11.

[Inc15] P. T. Inc., *Collaborative data science*, Plotly Technologies Inc., Montreal, QC, 2015.

[Iqb+21] S. Iqbal et al., *Classification of COVID-19 via Homology of CT-SCAN*, arXiv preprint arXiv:2102.10593 (2021).

[Kan+16] L. Kanari et al., *Quantifying topological invariants of neuronal morphologies*, arXiv: 1603.08432 `[q-bio.NC]`, 2016.

[Kan+17] L. Kanari et al., *A Topological Representation of Branching Neuronal Morphologies*, Neuroinformatics **16** (2017), 3–13.

[Lee+17] Y. Lee et al., *Quantifying similarity of pore-geometry in nanoporous materials*, Nature Communications **8** (2017), no. 1, 15396, DOI: 10.1038/ncomms15396.

[Lov+23] E. R. Love et al., *Topological Convolutional Layers for Deep Learning*, Journal of Machine Learning Research **24** (2023), no. 59, 1–35.

[Mar+14] C. Maria et al., *The Gudhi Library: Simplicial Complexes and Persistent Homology*, Mathematical Software – ICMS 2014, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014, 167–174.

[NLC11] M. Nicolau, A. J. Levine, and G. Carlsson, *Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival* (en), Proc. Natl. Acad. Sci. U. S. A. **108** (2011), no. 17, 7265–7270.

[Pro24] T. G. Project, *GUDHI User and Reference Manual*, 3.10.1, GUDHI Editorial Board, 2024.

[Pur+23] E. Purvine et al., *Experimental Observations of the Topology of Convolutional Neural Network Activations*, Proceedings of the AAAI Conference on Artificial Intelligence **37** (2023), no. 8, 9470–9479, DOI: 10.1609/aaai.v37i8.26134.

[Sag+18] M. Saggar et al., *Towards a new approach to reveal dynamical organization of the brain using topological data analysis*, Nature Communications **9** (2018), no. 1, 1399, DOI: 10.1038/s41467-018-03664-4.

[SBH20] K. Sasaki, D. Bruder, and E. A. Hernandez-Vargas, *Topological data analysis to model the shape of immune responses during co-infections* (en), Commun. Nonlinear Sci. Numer. Simul. **85** (2020), no. 105228, 105228.

[SC04] V. d. Silva and G. Carlsson, *Topological estimation using witness complexes*, SPBG'04 Symposium on Point - Based Graphics 2004, The Eurographics Association, 2004, DOI: /10.2312/SPBG/SPBG04/157-166.

[SMC07] G. Singh, F. Mémoli, and G. E. Carlsson, *Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object Recognition*, 4th Symposium on Point Based Graphics, PBG@Eurographics 2007, Prague, Czech Republic, September 2-3, 2007, Eurographics Association, 2007, 91–100, DOI: 10.2312/SPBG/SPBG07/091-100.

[SPM15] G. Singh, P. Pearson, and D. Muellner, *TDAmapper*, 1st ed., 2015.

[Str+23] K. Strommen et al., *A topological perspective on weather regimes*, Climate Dynamics **60** (2023), no. 5, 1415–1445, DOI: 10.1007/s00382-022-06395-x.

[Tak+17] A. Takiyama et al., *Persistent homology index as a robust quantitative measure of immunohistochemical scoring*, Scientific reports **7** (2017), no. 1, 14002.

[Tau+20] G. Tauzin et al., *giotto-tda: A Topological Data Analysis Toolkit for Machine Learning and Data Exploration*, arXiv: 2004.02551 `[cs.LG]`, 2020.

[Tho+21] A. Thomas et al., *Topological Data Analysis of C. elegans Locomotion and Behavior*, Frontiers in Artificial Intelligence **4** (2021), DOI: 10.3389/frai.2021.668395.

[Vee+20] H. J. van Veen et al., *Kepler Mapper: A flexible Python implementation of the Mapper algorithm*, Zenodo, 2020, DOI: 10.5281/zenodo.4077395.

[ZC05] A. Zomorodian and G. Carlsson, *Computing Persistent Homology*, Discrete & Computational Geometry **33** (2005), no. 2, 249–274, DOI: 10.1007/s00454-004-1146-y.

[ZCS23] M. Zhang, S. Chowdhury, and M. Saggar, *Temporal Mapper: Transition networks in simulated and real neural dynamics*, Network Neuroscience **7** (2023), no. 2, 431–460, DOI: 10.1162/netn_a_00301, eprint: https://direct.mit.edu/netn/article-pdf/7/2/431/2118400/netn_a_00301.pdf.

[Zhe+21] F. Zheng et al., *HiDeF: identifying persistent structures in multiscale 'omics data*, Genome biology **22** (2021), 1–15.

[Zho+21] Y. Zhou et al., *Mapper Interactive: A Scalable, Extendable, and Interactive Toolbox for the Visual Exploration of High-Dimensional Data*, 2021 IEEE 14th Pacific Visualization Symposium (PacificVis), 2021, 101–110, DOI: 10.1109/PacificVis52677.2021.00021.