

PARAMETERIZED TOPOLOGICAL DATA ANALYSIS

A DISSERTATION

SUBMITTED TO THE INSTITUTE FOR COMPUTATIONAL AND

MATHEMATICAL ENGINEERING

AND THE COMMITTEE ON GRADUATE STUDIES

OF STANFORD UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Bradley J. Nelson

June 2020

© 2020 by Bradley Jared Nelson. All Rights Reserved.

Re-distributed by Stanford University under license with the author.



This work is licensed under a Creative Commons Attribution-Noncommercial 3.0 United States License.

<http://creativecommons.org/licenses/by-nc/3.0/us/>

This dissertation is online at: <http://purl.stanford.edu/kn625zh9782>

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Gunnar Carlsson, Primary Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Jonathan Taylor, Co-Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Steven Kerckhoff

Approved for the Stanford University Committee on Graduate Studies.

Stacey F. Bent, Vice Provost for Graduate Education

This signature page was generated electronically upon submission of this dissertation in electronic format. An original signed hard copy of the signature page is on file in University Archives.

Abstract

Topological data analysis seeks to understand and utilize topological features of data, such as clusters and holes. One such problem is to characterize topological spaces from sampled data in order to build mathematical models of how the data was generated. However, many common constructions for this purpose become computationally intractable for even a moderate number of samples as the topological feature dimension increases. Unfortunately, many samples may be necessary to resolve higher-dimensional topological features, making it challenging to use standard techniques.

This dissertation investigates several ways in which topological data analysis can be made more digestible by structuring computations and models. First, we introduce a new method for computing algebraic invariants of diagrams of topological spaces using matrices associated with quiver representations. This computational framework allows for parallel algorithms to compute persistent and zigzag homology in the most general case, with arbitrary induced maps on homology. Next, we extend the classical techniques of acyclic carriers to the filtered setting and demonstrate how these tools can be used to construct interleavings to compare persistent homology of filtered spaces. We introduce a class of geometric complexes parameterized by a cover of a data set and use carriers to analyze the relationship between these complexes to the unparameterized geometric complexes. Finally, we investigate spaces of data generated from sampling small cubes of voxels (patches) from three-dimensional images. We show how to generalize a well-known Klein bottle model for two-dimensional patches to the higher-dimensional setting by using a map that captures the direction of the largest variation. We combine this model with computational tools to investigate the distribution of patches in different three-dimensional image data sets.

To my parents:
Jared and Beth

And my family:
Xiaotong and Joshua

Acknowledgements

As my advisor, Gunnar Carlsson has offered invaluable advice in many aspects of this work. At a high level, none of the vision for this thesis could have developed without his guidance over the past five years. I think it is safe to say that he has pointed me in a promising direction on numerous occasions. At a more detailed level, Gunnar first proposed a block-matrix algorithm for computing zigzag homology, which eventually evolved into the algorithm in Chapter 3. Much of my understanding of the Klein bottle model for image patches is due to many conversations with Gunnar over the years on his prior work on this topic. When I first ran some experiments on samples from a model \mathcal{K}^3 in Chapter 5 with different fields, Gunnar suggested using the Leray-Serre spectral sequence to compute the homology of the model space.

Many of the technical details in Chapter 3 were understood while working in close collaboration with Anjan Dwaraknath. Working with him on this topic was one of the highlights of my Ph.D.

Jonathan Taylor has been a wonderful co-advisor for the past several years. We have had many enjoyable discussions in that time, and his detailed questions always leave me with more to think about. He has given a lot of valuable feedback on exposition in this work, and caught numerous typos.

I'd like to thank Steve Kerckhoff for serving as a reader for this dissertation. Additionally, thanks to Rafe Mazzeo for serving as an oral examiner, and Johan Ugander for chairing my defense.

Several additional people deserve thanks for helping me along the way: Ryan Lewis for first introducing me to topological data analysis. Greg Henselman for inspiring me to start tinkering with persistent homology computations. Nathan Perlmutter for

a homotopy theory class that caught my imagination. Samir Choudhury for keeping a reading group lively. Rickard Brüel Gabrielsson, Primoz Skraba, Leonidas Guibas, and Rahul Sarkar for collaborations outside of this work.

I would also like to thank Kazu Terao and the Neutrino group at SLAC for including me and providing support for nine months of my Ph.D. - it was a great experience working with them and getting a taste for real scientific problems. I'd also like to thank Geoff Sanders at LLNL who was my mentor for two summer internships, and gave me a lot of freedom to explore ideas.

There are many people to thank for making Stanford, and ICME in particular, a welcoming place for me. Michael Saunders for running a linear algebra and optimization seminar where students were always welcome. Margot Gerritsen for helping me in many ways in her time as director of ICME, and afterward. Indira Choudhury for administrative assistance. Gianluca Iaccarino for his help as the new ICME director. And, of course, many students who I've enjoyed spending time with including Austin Benson, Anil Damle, Victor Minden, Nolan Skochdopole, Lan Nguyen, Ron Estrin, Eileen Martin, Alex Ioannidis, Arun Jambulapati, Tim Moon, Song Mei, Lei Lei, Zhengyu Huang, Carson Kent, Gabriel Maher, Shaked Regev, Ruohan Zhan, Ryan Aronson, Steven Brill, and Bazyli Klockiewicz. Outside of ICME, I'd like to thank Tom Robbins, Amy Stanesco, Laura Domine, Victor Pereyra, Alan Karp, and Joe Grear for making my time at Stanford memorable.

I am fortunate to have been supported by US taxpayers through a three-year National Defense Science and Engineering Graduate Fellowship from the Department of Defense and additional funding through the Department of Energy while at SLAC and LLNL. For the last stretch of my Ph.D. I have been supported by Unbox AI.

I'd like to thank my family for their support over the years: My parents, Jared and Beth, for teaching me to learn. My siblings, Josh, Andrea, Spencer, and Michael, for their companionship. Finally, my wife, Xiaotong, who I was lucky to meet my first year at Stanford, and who has brightened my days ever since.

Contents

Abstract	iv
Acknowledgements	vi
1 Introduction	1
1.1 Topological Data Analysis	2
1.1.1 Geometric Constructions	2
1.1.2 Persistent Homology at a Glance	5
1.1.3 Quiver Representations	8
1.2 The Need for Structured Methods	8
1.2.1 The Topology of Image Patches	10
1.2.2 Topological Parameterization	13
1.2.3 Calculation Tools in Algebraic Topology	16
1.3 Concerning Torsion	22
1.4 Roadmap & Contributions	26
2 Preliminaries	28
2.1 Algebra and Notation	28
2.2 Cell Complexes	34
2.2.1 Homotopy	35
2.3 Chain Complexes	36
2.3.1 Augmented Chain Complexes	38
2.3.2 Chain Homotopy	38
2.4 Homology	39

2.4.1	Homotopy Invariance	40
2.4.2	Acyclic Complexes	41
2.5	Computing Homology	42
2.5.1	The Reduction Algorithm	43
2.5.2	Induced Maps	48
2.6	Persistent Homology of Filtrations	51
2.7	Reduction Revisited	53
2.8	From Inclusion to General Maps	55
2.8.1	Cylinders to Filtrations	55
2.8.2	The Mapping Cylinder	56
2.8.3	Persistent Homology	57
3	Quiver Representations and Algorithms	59
3.1	Prior Work on Computing Zigzag Homology	60
3.2	Classification of Quiver Representations and Barcodes	62
3.2.1	Quiver Representations	62
3.2.2	From Topology to Quiver Representations	63
3.2.3	Type A Quiver Representations	65
3.2.4	The Graded Module Structure of Persistent Homology	71
3.2.5	Generic Quiver Computations	73
3.3	Algorithms for Canonical Forms of Type-A Quiver Representations	76
3.3.1	Triangular Factorizations	77
3.3.2	Shape Commutation Relations	80
3.3.3	Algorithm for Persistence-type Quivers	83
3.3.4	Alternating arrow directions	85
3.3.5	General Sequential Algorithm	87
3.3.6	Parallel Quiver Algorithm	89
3.3.7	Correctness and Uniqueness of the Barcode Factorization	94
3.4	Experiments	96
3.4.1	Subsets and Rips Complexes	96
3.4.2	Bivariate Nerve	98

3.4.3	Sierpinski Triangle	100
3.5	Conclusion & Future Directions	100
4	Extensions of Maps and Interleavings	103
4.1	Carriers and Extensions of Maps	104
4.1.1	Carriers	104
4.1.2	Algorithmic Extensions of Maps	109
4.1.3	From Relations to Maps	110
4.1.4	Composition of Carriers	112
4.1.5	From Acyclic Carriers to Equivalence	113
4.2	Interleavings	115
4.2.1	Filtrations, Posets, and Non-decreasing Maps	116
4.2.2	Persistence Vector Spaces and Interleavings	118
4.2.3	The Interleaving and Bottleneck Distances	120
4.3	Filtered Carriers and Interleavings	122
4.3.1	Preliminary Notions	122
4.3.2	Filtered Carriers to Interleavings	125
4.3.3	Stability of Flag Filtrations	126
4.3.4	Stability of Geometric Filtrations	128
4.4	Cover Complexes	135
4.4.1	A Generalized Nerve Theorem	136
4.4.2	Local Interleavings	138
4.4.3	Interleaving with Full Complex	140
4.4.4	Finding Covers	143
4.5	Future Directions	144
5	Models for Image Patch Data	146
5.1	Preliminaries	146
5.1.1	Processing Image Patches	146
5.1.2	A Klein bottle near image patches	149
5.1.3	Three-dimensional Images	149
5.1.4	Primary Spheres and Secondary Circles	152

5.2	The Harris Fibration	154
5.2.1	The Harris Map	154
5.2.2	A Fibration Structure	156
5.2.3	Real Projective Spaces	158
5.2.4	Local Coefficient Calculations	159
5.2.5	Homology of the Harris Fibration	161
5.2.6	Field Coefficient Calculations	165
5.3	Density in the Image and Fibers of the Harris Map	166
5.3.1	Two-dimensional Images	166
5.3.2	Three-dimensional Images	169
5.4	Topological Computations	171
5.4.1	2-dimensional Image Patches	172
5.4.2	3-dimensional Image Patches	172
5.4.3	Commentary	174
5.5	Conclusion & Future Directions	175
6	Conclusion	177
	Bibliography	178

List of Figures

1.1	Points sampled from the unit circle, with some noise.	3
1.2	Persistence barcode and diagram for the Rips filtration on the points in Figure 1.1. There is one connected component that persists, and one prominent H_1 generator, which corresponds to the hole in the circle. Left: Persistence barcode. Each bar represents a birth-death pair that persists for the length of the bar. Blue bars are finite, and red bars indicate a death at ∞ . Right: Persistence diagram. Each point is a birth-death pair. No points can appear below the dashed diagonal, and points that appear above the dashed red line have deaths at ∞	7
1.3	Primary circle in image patches. Patches are represented by idealized grayscale circles, which would be discretized on a grid in image data. The patches shown are a discrete sample of the continuous primary circle.	10
1.4	Secondary circles in image patches.	11
1.5	A Klein bottle for image patches. The primary circle lies in the two horizontal green bands. Two secondary circles are seen in the vertical blue bands. The horizontal red band is equivalent to $\mathbb{R}P^1$	12
1.6	CW structure for the Klein bottle.	18
1.7	Klein bottle covered by two Möbius bands, denoted A and B	19
2.1	Notation for different matrices, along with pictorial symbols	34
3.1	Pictorial representation of the shape commutation relationship in Proposition 3.3.1	82

3.2	The factorization to use for the first sweep.	88
3.3	The commutation to use for the second sweep.	88
3.4	Time to compute zigzag homology of a zigzag diagram computed on subsamples of a noisy circle. Normal noise with variance 0.1 is added to points sampled from a unit circle. 200 points are contained in each subsample, and $r = 0.35$. The horizontal axis indicates the number of samples used in the diagram. Left: time to compute zigzag homology for both BATS and Dionysus. Right: Speedup seen using BATS instead of Dionysus. At the right hand side, BATS is over 600x faster.	97
3.5	Left: persistence diagram for Rips filtration on 200 points sampled from the unit circle. The unoptimized reduction algorithm runs in 20 seconds in BATS. Right: An approximate persistence diagram created using the discrete Morozov zigzag construction in [85] using the suggested parameters. The zigzag computation takes approximately 0.5 seconds in BATS. While the birth and death times are not identical, both diagrams qualitatively display the same information, namely a single connected component and a robust H_1 class, agreeing with the homology of the circle.	98
3.6	Zigzag barcodes of bivariate Nerve diagram on 5 covers of 500 points on the unit circle. Covers are computed by selecting 20 random landmarks. Left: each point is assigned to closest 2 landmarks. Right: each point is assigned to closest 3 landmarks. Note that both diagrams have single long bars in dimensions 0 and 1, agreeing with the homology of the circle.	99
3.7	Persistence barcode showing 4 iterations of the sequence in Equation (3.21).	101
4.1	Data lies on a 20×10 grid on a flat torus. Left: Persistence diagram for Rips filtration. Right: Persistence diagram for Rips cover filtration. Cover is obtained by a pullback of a cover on the first circular coordinate.	143

5.1	Image 400 from the Van Hateren dataset [101]. An image patch is a $k \times k$ pixel sub-image, such as the one indicated in red.	147
5.2	Two slices of an MRI in the BRATS data set. Left: a slice in the second two coordinates. Right: a slice in the first two coordinates. . .	151
5.3	Two slices of the Penobscot data set. Left: the top third of an “inline” slice (the image continues vertically). Right: a horizontal slice.. . .	151
5.4	Primary circle in 7×7 patches, $k = 100$, $p = 0.2$, with a secondary circle beginning to fill in. Center: projection of data onto first two principal component directions. Left: first eigenpatch (patch maximally aligned with first principal component). Right: second eigenpatch.	152
5.5	Secondary circle in $5 \times 5 \times 5$ patches from Penobscot data, $k = 100$, $p = 0.4$. Center: projection of data onto first two principal component directions. Left: first eigenpatch. Right: second eigenpatch.	153
5.6	Primary great circle in $5 \times 5 \times 5$ patches from BRATS data, $k = 100$, $p = 0.2$. Projection of data onto first two principal component directions. Left: first eigenpatch. Right: second eigenpatch. Top: third eigenpatch.	153
5.7	Harris map h from a patch to $\mathbb{R}P^1$	156
5.8	Left: patch with odd function ($k(\pi/2, \pi/2)$). Right: patch with even function ($k(\pi/2, 0)$).	156
5.9	Local coefficient system on $\mathbb{R}P^1 = S^1$. T denotes the non-trivial \mathbb{Z} -automorphism $T : a \mapsto -a$	160
5.10	Left: 3×3 patches. Right: 7×7 patches. Top: histograms of the image of the Harris map applied to patches obtained from the Van Hateren data set. Bottom: histograms of points assigned to nearest neighbors in a model Klein bottle (yellow bins contain more patches, and blue contain fewer).	167

5.11	PCA embeddings of inverse images of open sets in $\mathbb{R}P^1$ covering the image of the Harris map applied to 7×7 patches, with 1st and 2nd eigenpatches (left and right). Only patches in the top 35% highest codensity ($k = 50$) are displayed as points. Top: open set with the largest number of patches. Bottom: open set with smallest number of patches.	168
5.12	Image of the Harris map applied to patches sampled from the BRATS data set. Affine spaces of $\mathbb{R}P^2$ colored by the \log_{10} number of patches in landmark neighborhoods. From left to right: stereographic projections from e_1 , e_2 , and e_3	169
5.13	Histogram of the number of nearest neighbors in the BRATS data to points in \mathcal{K}^3 . The xy plane is a stereographic projection of $\mathbb{R}P^2$ from the third coordinate, and the z axis is the fiber coordinate θ	170
5.14	Image of the Harris map applied to patches sampled from the Penobscot data set. Affine spaces of $\mathbb{R}P^2$ colored by the \log_{10} number of patches in landmark neighborhoods. From left to right: stereographic projections from e_1 , e_2 , and e_3	170
5.15	Histogram of the number of nearest neighbors in the Penobscot data to points in \mathcal{K}^3 . The xy plane is a stereographic projection of $\mathbb{R}P^2$ from the second coordinate, and the z axis is the fiber coordinate θ	171
5.16	Dowker persistence diagrams with 200 landmarks computed on a dataset sampled from \mathcal{K}^2 . Left: computed with \mathbb{F}_2 coefficients. Right: computed with rational coefficients.	172
5.17	7×7 patches from the Van Hateren data set, filtered with $k = 30$, $p = 0.2$. Left: histogram of density of projection onto a model \mathcal{K}^2 . Right: Persistence diagram for Dowker complex, 200 landmarks, computed with \mathbb{F}_2 coefficients.	173
5.18	Dowker persistence diagrams with 300 landmarks computed on a dataset sampled from \mathcal{K}^3 . Left: computed with \mathbb{F}_2 coefficients. Right: computed with rational coefficients.	173

5.19	Dowker persistence diagram with 100 landmarks computed with \mathbb{F}_2 coefficients on $5 \times 5 \times 5$ patches sampled from the Penobscot data. $k = 40, p = 0.4$	174
5.20	$5 \times 5 \times 5$ patches from the BRATS dataset. Codensity filter with $k = 100, p = 0.2$. Left: histogram of density of projection onto model \mathcal{K}^3 , with xy plane stereographic projection from the third coordinate.. Right: Persistent homology of Dowker filtration with 100 landmarks, \mathbb{F}_2 coefficients.	175

Chapter 1

Introduction

Topological data analysis is a term used to refer to a collection of techniques developed for data exploration and analysis in the early 21st century which trace their roots to mid-20th century algebraic topology. The power of these techniques lie in how little structure in data is required, which allows them to be adapted to many different situations and strengthened as structure is added. In contrast to versions of data analysis which view data as vectors of features stacked into matrices to form a data set, topological data analysis views data as points in some space, often endowed with a notion which encodes which points are similar and which are dissimilar. This viewpoint can be applied equally well to a single molecule, defined by how its constituent atoms are arranged, as well as to databases of molecules, which seek to group molecules with similar structure together.

This dissertation originated as an attempt to understand *why* a Klein bottle appeared in a data set derived from natural images [25], and then to make the discovery process systematic. The culmination of these efforts can be found in Chapter 5 which discusses how structure, specifically maps to a parameter space, can be used both to aid in topological modeling of data and additionally structure and simplify computations. In order to reach this destination, we will also visit other topics. Chapters 2 and 3 will cover algebraic invariants of diagrams of topological spaces, and introduce new algorithms for computing these invariants using quiver representations. Chapter 4 will then develop techniques to compare invariants across different constructions,

or when data is perturbed or approximated, and apply these techniques to understand complexes that use a cover to impose structure on a space. Together, this collection of work offers one version (certainly not unique) of an end-to-end pipeline for topological data analysis.

1.1 Topological Data Analysis

Topological data analysis is concerned with computing invariants of topological spaces derived from data, and understanding the stability of these invariants under perturbation and approximation. One set of applications seeks to understand the structure of data sets using these invariants [1, 2, 16, 25, 87]. A second set of applications seeks to produce topological features for each data point, which then can be used by other data analysis techniques [21, 89]. A third set of applications seeks to use topological features to regularize machine learning models (see [17] for a review of recent work).

Topological data analysis offers a set of tools that both extend and complement other techniques found in data science. Persistent homology can be viewed as an extension of single-linkage clustering to higher-dimensional features of data sets. The mapper algorithm [92] can be viewed as a topologically motivated version of nonlinear dimensionality reduction.

1.1.1 Geometric Constructions

A fundamental problem that we will study is how to compute topological invariants of point clouds. A set of points \mathbf{X} sampled near the unit circle S^1 in two dimensions, as seen in Figure 1.1. We would like to capture information about the sample \mathbf{X} that shows that it is “close” to the circle S^1 , for instance, encoding that the points are close to one cluster (connected component), but that there is a region in the middle of the cluster that is not sampled (a hole). However, as a topological space, the points in \mathbf{X} naively have the discrete topology. We need to augment topology with additional information, in this case using geometry (pairwise distance information) to capture the desired features of the space.

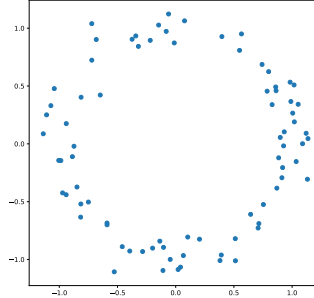


Figure 1.1: Points sampled from the unit circle, with some noise.

We will consider spaces built from simplices, called simplicial complexes. A k -simplex (x_0, \dots, x_k) is simply the convex hull of the points $\{x_0, \dots, x_k\}$. A 0-simplex is a point, a 1-simplex is an edge, a 2-simplex triangle, and higher order simplices are simply higher-order tetrahedra. Further detail will be found in Section 2.2. As a matter of notation, we will use standard capital letters such as X to refer to topological spaces (continuous or discrete), bold face letters such as \mathbf{X} to indicate a sample from a larger space X , and caligraphic letters such as \mathcal{X} to denote a simplicial complex (often with a vertex set of points in X or \mathbf{X}). All the following constructions will assume that a sample \mathbf{X} is equipped with a dissimilarity d_X (such a metric), often the restriction of a larger dissimilarity space (X, d_X) to the points $\mathbf{X} \subset X$.

Definition 1.1.1. *Let $\mathbf{X} \subseteq (X, d_X)$. The Čech complex at parameter r , denoted $\check{\mathcal{C}}(\mathbf{X}; r)$, is a simplicial complex on the vertex set \mathbf{X} , where the simplex $(x_0, \dots, x_k) \in \check{\mathcal{C}}(\mathbf{X}; r)$ if and only if there exists some $x \in X$ such that $d_X(x_i, x) \leq r$ for all $x_i \in \{x_0, \dots, x_k\}$.*

A related construction is the α -complex [44], which restricts the simplices of the Čech complex to the Delaunay triangulation of a point cloud.

Definition 1.1.2. *Let \mathbf{X} be a set with dissimilarity d_X . The Vietoris-Rips complex at parameter r , denoted $\mathcal{R}(\mathbf{X}; r)$ is a simplicial complex on the vertex set \mathbf{X} , where the simplex $(x_0, \dots, x_k) \in \mathcal{R}(\mathbf{X}; r)$ if and only if $d_X(x_i, x_j) \leq r$ for all $(x_i, x_j) \in \{x_0, \dots, x_k\}^2$.*

Both Čech and Rips complexes are built using the full data set \mathbf{X} as the vertex set of the complex, and for large parameters of r , will contain a very large number of simplices. One approach which can significantly reduce the size of the vertex set, and thus the maximum number of simplices, is by using a set of landmarks $\mathbf{L} \subset \mathbf{X}$, and witnesses $\mathbf{W} \subseteq \mathbf{X}$ (we will typically use $\mathbf{W} = \mathbf{X}$).

Definition 1.1.3. *Let \mathbf{L}, \mathbf{W} be sets, and $d : \mathbf{L} \times \mathbf{W} \rightarrow \mathbb{R}$ be any function. The Dowker complex [31, 42, 52] $\mathcal{D}(\mathbf{L}, \mathbf{W}; r)$ is a simplicial complex with vertex set \mathbf{L} . The simplex $(\ell_0, \dots, \ell_k) \in \mathcal{D}(\mathbf{L}, \mathbf{W}; r)$ if there exists some $w \in \mathbf{W}$ so that $d(\ell_i, w) \leq r$ for all $i = 0, \dots, k$.*

Note that for $\mathbf{X} \subseteq (X, d_X)$, that $\check{\mathcal{C}}(\mathbf{X}; r) = \mathcal{D}(\mathbf{X}, X; r)$. The advantage of Dowker complexes which use some sample \mathbf{W} instead of an ambient space X is that it is straightforward to determine when a simplex should be added without needing to compute whether metric balls intersect. Just as the Dowker complex offers a discrete version of the Čech complex, the *witness complex* [36, 37] offers a discrete version of the α -complex.

Definition 1.1.4. *Let \mathbf{L}, \mathbf{W} be sets and $d : \mathbf{L} \times \mathbf{W} \rightarrow \mathbb{R}$. The witness complex $\mathcal{W}(\mathbf{L}, \mathbf{W}; r)$ is a simplicial complex with vertex set \mathbf{L} . The simplex $(\ell_0, \dots, \ell_k) \in \mathcal{W}(\mathbf{L}, \mathbf{W}; r)$ if all its faces are also in $\mathcal{W}(\mathbf{L}, \mathbf{W}; r)$, and there exists some witness $w \in \mathbf{W}$ so that $d(\ell_i, w) \leq d(\ell, w) + r$ for all $i = 0, \dots, k$ and for all $\ell \in \mathbf{L} \setminus \{\ell_0, \dots, \ell_k\}$.*

There are several potential modifications to the definition of witness complexes which can be found in [37], and we use the present definition to coincide with the one used to obtain results in [31, 32]. The condition that a witness be closer to the landmarks in the witnessed simplex than other landmarks can make the complex sensitive to the choice of landmarks (see [31] for an example), which will make the Dowker complex easier to analyze.

Verifying conditions for high-dimensional simplices can be potentially difficult and computationally expensive. In practice, it can be easier to use complexes defined on lower dimensional skeleta. We define $\mathcal{D}_\infty(\mathbf{L}, \mathbf{W}; r)$ to be the standard (strict) Dowker complex. Now, fix some $k \geq 1$, and we define $\mathcal{D}_k(\mathbf{L}, \mathbf{W}; r)$ to be the maximal

simplicial complex on the k -skeleton of $\mathcal{D}_\infty(\mathbf{L}, \mathbf{W}; r)$. Similarly, we can define \mathcal{W}_k for Witness complexes, and $\check{\mathcal{C}}_k$ for Čech complexes (note that the Vietoris-Rips complex, \mathcal{R} , is already defined on its 1-skeleton). The case of \mathcal{D}_1 also appears as a version of the lazy witness complex in [37]. Note that if (X, d_X) is a geodesic metric, and $\mathbf{X} \subset X$, then $\check{\mathcal{C}}_1(\mathbf{X}; r) = \mathcal{R}(\mathbf{X}; 2r)$.

There is nothing canonical about any of these constructions – they each produce spaces on sampled points which will allow us to compute topological signatures of a point cloud, and each has its merits in different settings. The Čech complex is useful for relating a sample from a manifold to the manifold itself [82]. The Vietoris-Rips complex is computationally attractive because it only requires pairwise comparisons, and the α -complex can reduce the number of simplices [44]. Dowker and Witness complexes produce smaller complexes (also computationally attractive), but potentially lose some information.

1.1.2 Persistent Homology at a Glance

In Definitions 1.1.1, 1.1.2, 1.1.3 and 1.1.4, we defined complexes on point clouds that depend on a parameter r . In practice, choosing a single parameter that captures the desired topological features of the space may be an impossible task. Instead, we consider a *filtration* of spaces at *all* parameters, and consider invariants of these constructions.

Definition 1.1.5. A filtered (simplicial) complex \mathcal{X} over \mathbb{R}_+ is a nested family of complexes

$$\mathcal{X}^0 \subseteq \dots \subseteq \mathcal{X}^r \subseteq \dots$$

Note that simplicial complexes built on finite point cloud data such as those in Definitions 1.1.1, 1.1.2, 1.1.3 and 1.1.4, only add simplices at a finite number of parameters, we will only consider filtrations which have a finite number of *critical values*, where the complex actually changes. By reparameterizing the filtration index r , we can consider filtrations which take integer values $\mathcal{X}^0 \subset \mathcal{X}^1 \subset \dots$.

Homology of a simplicial complex \mathcal{X} with coefficients in a ring R is computed by first forming a chain complex $C_*(\mathcal{X})$, where $C_k(\mathcal{X})$ is a free R -module obtained by

considering formal R -linear sums of k -simplices in \mathcal{X} , with differential (boundary) operators $\partial_k : C_k(\mathcal{X}) \rightarrow C_{k-1}(\mathcal{X})$ that send a simplices to linear combinations of simplices in their boundaries. The differential maps ∂ have the property that $\partial_k \circ \partial_{k+1} = 0$. The homology quotient R -module in dimension k is defined to be

$$H_k(\mathcal{X}) = \ker \partial_k / \text{img } \partial_{k+1} \quad (1.1)$$

Details concerning this sequence of constructions will be found in Chapter 2. From a data analysis perspective, what is important is that the quotient modules $H_k(\mathcal{X})$ contain topological information about \mathcal{X} . The free rank of $H_0(\mathcal{X})$ is equal to the number of connected components in \mathcal{X} , $H_1(\mathcal{X})$ contains information about non-contractible “holes”, and higher-dimensional homology contains information about higher dimensional voids. Unless otherwise noted, we will consider the situation in which the ring R is a field \mathbb{F} , in which case modules become vector spaces – see Section 1.3 for further discussion.

Homology is *functorial*, meaning that it not only captures information about topological spaces, but also maps between topological spaces. For every map of simplicial complexes $f : \mathcal{X} \rightarrow \mathcal{Y}$, there is an induced linear map on homology $F_k : H_k(\mathcal{X}) \rightarrow H_k(\mathcal{Y})$ so that the following diagram commutes:

$$\begin{array}{ccc} \mathcal{X} & \xrightarrow{f} & \mathcal{Y} \\ \downarrow & & \downarrow \\ H_k(\mathcal{X}) & \xrightarrow{F_k} & H_k(\mathcal{Y}) \end{array} \quad (1.2)$$

Persistent homology is an algebraic invariant of diagrams of spaces where the maps go in a single direction

$$\begin{array}{ccccccc} \mathcal{X}^0 & \xrightarrow{f^0} & \mathcal{X}^1 & \xrightarrow{f^1} & \dots & & \\ \downarrow & & \downarrow & & & & \\ H_k(\mathcal{X}^0) & \xrightarrow{F_k^0} & H_k(\mathcal{X}^1) & \xrightarrow{F_k^1} & \dots & & \end{array} \quad (1.3)$$

In the case of filtrations, we can study this diagram where the maps f^i are inclusions $\mathcal{X}^i \subseteq \mathcal{X}^{i+1}$.

We will use $PH_k(\mathcal{X}^t)$ to refer to the full bottom row of Equation (1.3), which consists of homology vector spaces as well as the induced maps, and $PH_*(\mathcal{X}^t) = \{PH_k(\mathcal{X}^t)\}_{k \geq 0}$. The persistent homology of a filtered complex can be represented as a set of *birth-death pairs* for each homology dimension

$$PH_k(\mathcal{X}^t) = \{b_i, d_i\} \quad (1.4)$$

Each birth b_i corresponds to a new homology basis element that appears in the filtration at parameter b_i (meaning it is not in the image of $F_k^{b_i-1}$). This element *persists* through the maps induced by inclusion until it finally enters the kernel of a map at parameter d_i . Further interpretation, discussion, and algorithms for computing $\{b_i, d_i\}$ will be covered in Chapters 2 and 3. The collection of birth-death pairs for $PH_k(\mathcal{X}^t)$ is also known as a “barcode” or “persistence diagram” (when plotted as points in the plane), as seen in Figure 1.2.

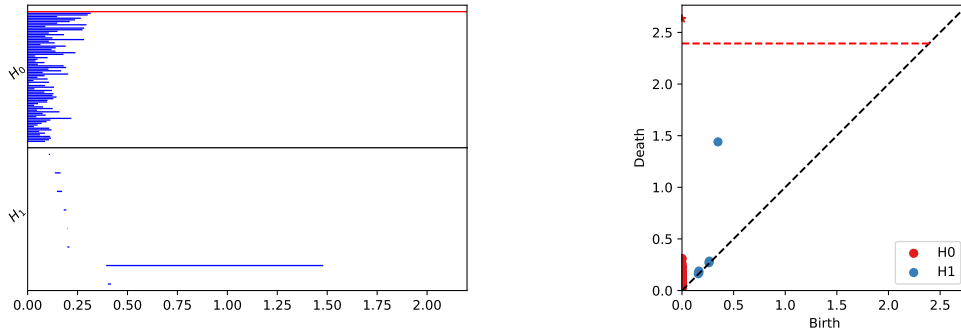


Figure 1.2: Persistence barcode and diagram for the Rips filtration on the points in Figure 1.1. There is one connected component that persists, and one prominent H_1 generator, which corresponds to the hole in the circle. Left: Persistence barcode. Each bar represents a birth-death pair that persists for the length of the bar. Blue bars are finite, and red bars indicate a death at ∞ . Right: Persistence diagram. Each point is a birth-death pair. No points can appear below the dashed diagonal, and points that appear above the dashed red line have deaths at ∞ .

1.1.3 Quiver Representations

In Equation (1.3) we considered the homology functor applied to a diagram of spaces with maps go in a single direction. More generally, we can consider diagrams of spaces

$$\mathcal{X}^0 \xleftrightarrow{f^0} \mathcal{X}^1 \xleftrightarrow{f^1} \mathcal{X}^2 \xleftrightarrow{f^2} \dots \xleftrightarrow{f^{N-1}} \mathcal{X}^n$$

where the bi-directional arrows can be given either direction, and the functions f^i can be arbitrary maps (not necessarily inclusions). Zigzag homology [22] studies the situation in which the direction of maps generally alternates, and we consider the most general case where any sequence of directions can be given. If we now look at induced maps on homology in dimension k we obtain the following diagram of vector spaces

$$H_k(\mathcal{X}^0) \xleftrightarrow{F_k^0} H_k(\mathcal{X}^1) \xleftrightarrow{F_k^1} H_k(\mathcal{X}^2) \xleftrightarrow{F_k^2} \dots \xleftrightarrow{F_k^{N-1}} H_k(\mathcal{X}^n)$$

This diagram of vector spaces is an example of a quiver representation of type A_{n+1} ($n + 1$ vector spaces and n linear transformations arranged on a line graph).

Like persistent homology, zigzag homology diagrams can be characterized by birth-death pairs (or barcodes). An important practical and theoretical question is whether or not persistence and zigzag barcodes depend on choices of bases for vector spaces in the above diagram. A theorem due to Gabriel [50] assures that this is not the case, by showing that type-A quiver representations have a finite indecomposable representation (although the proof is not constructive). In fact, bars in the barcodes of topological data analysis are exactly the indecomposables of the associated quiver representation, called interval indecomposables, and so are basis independent.

1.2 The Need for Structured Methods

The procedure described in Section 1.1.2 is straightforward to apply to point cloud data as well as other situations which produce filtrations. However, as the size of point clouds increase, the practitioner will quickly run into problems with the sheer size of the computations. The fundamental issue is that if \mathcal{X} is a simplicial complex

on a vertex set of size n , there can be up to $\binom{n}{k+2}$ $(k+1)$ -simplices, which are needed in order to compute $H_k(\mathcal{X}) = \ker \partial_k / \text{img } \partial_{k+1}$. In practice, all these simplices will be formed in geometric filtrations if we let the filtration parameter r grow until all points are connected. This makes it practically impossible to naively compute persistent homology in even moderately high dimensions on point clouds with more than several hundred points.

In practice, however, the situation is not so dire. Over the past decade, there has been tremendous progress in accelerating computation of persistent homology, through high quality implementations and improvements to fundamental algorithms [9, 11, 59, 72, 79, 105], as well as approximation schemes that can greatly reduce the number of simplices that are needed to be formed [15, 85, 91]. Many real-world datasets of interest have implicit low-dimensional structure that allow these approximation schemes to work reasonably well.

In contrast to methods that take advantage of implicit structure, we will be interested in developing methods that use *explicit* structure tailored to a particular data set. This explicit structure might come from known information about a data set, or might emerge as part of an iterative process in data analysis. For example, in Section 1.2.1 we'll see how topological structure was discovered in a data set derived from images, first by considering only the most dense regions of the space, and then adding more points to reveal additional structure. Methods that take advantage of such explicit structure have been very important in modern algebraic topology, often taking the form of various long exact sequences or spectral sequences, and have allowed topologists to compute the homology of fairly complex spaces "by hand". We will review several examples in Section 1.2.3.

We will seek to develop methods that use explicit structure in the persistent homology of data. This will allow for simplified computations, and also for an additional layer of interpretability when we perform topological modeling.

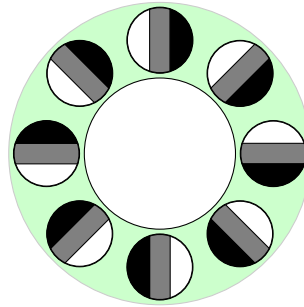


Figure 1.3: Primary circle in image patches. Patches are represented by idealized grayscale circles, which would be discretized on a grid in image data. The patches shown are a discrete sample of the continuous primary circle.

1.2.1 The Topology of Image Patches

In 2003, Lee et al [66] began the study of the distribution of 3×3 high-contrast pixel arrays (“patches”) derived from grayscale images of natural scenery in the van Hateren database [101]. Empirically, the most common behavior of these patches is to have little variation in pixel value (for example, a blue sky has little color variation). Attention is next naturally focused on high-contrast patches, that is patches in which pixel values have large variations. In the literature these patches are typically mean-centered and then normalized. Henceforth, we shall simply refer to the objects of study as “patches”, with the understanding that these patches have already undergone selection for their high-contrast behavior, have been mean-centered, and have had variance normalized.

This data was first investigated using topological data analysis techniques by de Silva and Carlsson in 2004 [37], who found a circle in a high-density region. This circle (see Figure 1.3) can be parameterized by all unit gradients in two dimensions, and was labeled the “primary circle”. Two additional (“secondary”) circles appeared for lower-density thresholds, which did not correspond to orientation, but a phase on the type of edges appearing in horizontal and vertical directions (see Figure 1.4). Collectively, these observations are referred to as the “three-circle model.”

In 2008, Carlsson et al. [25] proposed that the three-circle model lies near a Klein bottle (a 2-dimensional manifold) that includes the three circle model as a

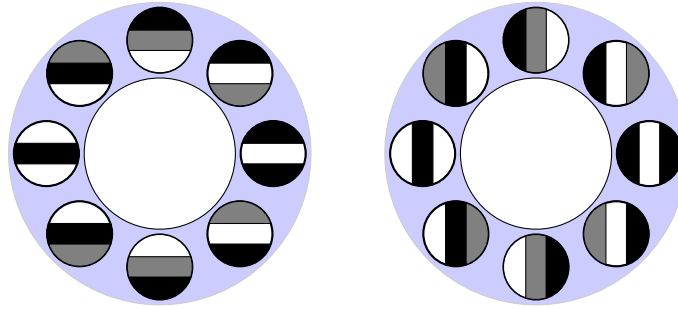


Figure 1.4: Secondary circles in image patches.

1-dimensional skeleton (Figure 1.5). This was supported by augmenting the three-circle data with the Klein bottle model, and observing that the Klein bottle homology was preserved. One interesting application of this model is in rotation-invariant texture recognition [86], which considers distributions of patches on the parameterized Klein bottle. This application also incorporated larger patch sizes into the model, as large as 11×11 . Analysis of range images extended to larger patch sizes (5×5 and 7×7) in the work of Adams and Carlsson [2], demonstrating the existence of a primary circle in these larger patch sizes. More recently, Adams et al. showed the existence of a torus in optical flow image data [1].

Ten years before the Klein bottle was proposed for image patches in the topological data analysis community, Tanaka proposed a Klein bottle to explain a orientation and phase-sensitive population of neurons in the primary visual cortex [97]. The correlation between natural image statistics and the sensitivities of the visual cortex was a primary motivation for the creation of the image patch dataset [66], which offers some explanation as to why similar topological structures appear in both the visual cortex and image patches. This has also inspired several attempts to improve image-based algorithms. Attempts to incorporate the Klein bottle into the first layer of convolutional neural networks in image recognition tasks have shown some promise [16], and a compression scheme for images that uses the Klein bottle [71] has been proposed.

The study of image patches in topological data analysis has laid bare a two challenges for the field. The first challenge is interpretability. The Klein bottle model

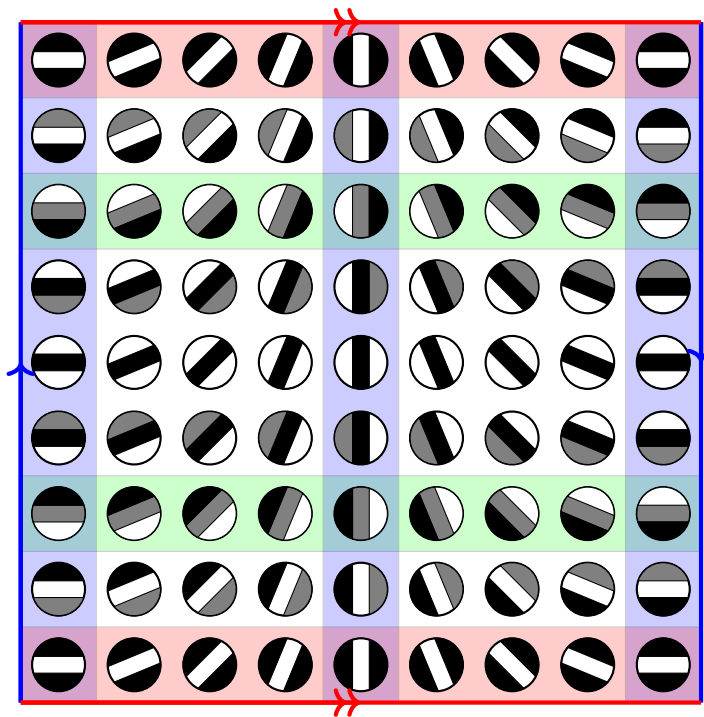


Figure 1.5: A Klein bottle for image patches. The primary circle lies in the two horizontal green bands. Two secondary circles are seen in the vertical blue bands. The horizontal red band is equivalent to $\mathbb{R}P^1$.

proposed in [25] is based on a family of degree-2 polynomials in two variables, and required several pages of analysis to show that the model space was indeed a Klein bottle. While aspects of the structure (such as orientation of patches) have been used in applications [86], to the best of our knowledge, this structure has not yet been used to explain *why* the Klein bottle appears in the first place. Related to this challenge is the fact that the proposed Klein bottle was used to *augment* the dataset, meaning that it covered regions that were never sampled (or removed by filtering). Can the Klein bottle be observed without this augmentation?

The second challenge is in general topological modeling. Modeling is the process of coming up with a description of a system that can be used to test hypotheses and generate new ones. It took several years to go from the discovery of the three-circle model [37] (2004), and the Klein bottle model [25] (2008), and one question is whether this could have been accelerated given better tools. Subsequent works investigating the topology of different types of imaging data [1, 2] would similarly benefit. In all these examples, an iterative process occurs in which high-density regions are first investigated, revealing some structure, for instance the primary circle. After this initial structure is discovered, the density threshold is lowered, and additional structure is revealed which must be incorporated into the initial model. As topological spaces become more complicated (and perhaps less intuitive), it can be difficult for a practitioner to do this process manually, and so the challenge is to develop tools that incorporate discovered structures *automatically*. Topological modeling is not confined to image patch data, and techniques that allow for these data-driven workflows could ease topological investigations in a variety of scientific applications.

1.2.2 Topological Parameterization

Parameterization refers to the process of controlling some output or quantity of interest by using a parameter from a space B . We'll consider parameterizations of a

space X through a surjective map $p : X \rightarrow B$, which we may write vertically as

$$\begin{array}{c} X \\ \downarrow p \\ B \end{array} \quad (1.5)$$

In this context, X is called the *total space*, and B is called the *base space*. p may be called the *projection* or *parameter* map. We'll be interested in *fibers* of p , meaning subspaces $p^{-1}(b) \subseteq X$ where $b \in B$.

Given a map $f : Y \rightarrow X$, the *homotopy lifting problem* asks whether a homotopy $\tilde{h} : I \times Y \rightarrow B$ can be lifted to a homotopy $h : I \times Y \rightarrow X$ so that the following diagram commutes

$$\begin{array}{ccc} 0 \times Y & \xrightarrow{f} & X \\ \downarrow & \nearrow h & \downarrow p \\ I \times Y & \xrightarrow{\tilde{h}} & B \end{array} \quad (1.6)$$

If homotopies can be lifted for all maps $f : Y \rightarrow X$, then $p : X \rightarrow B$ is called a *fibration*. This implies that all fibers $f^{-1}(b)$ are homotopy equivalent to some common space F , and we write a general fibration:

$$\begin{array}{ccc} F & \hookrightarrow & X \\ & & \downarrow p \\ & & B \end{array} \quad (1.7)$$

A related notion to a fibration is that of a *fiber (fibre) bundle* [62,95], where the British spelling is often used. These are important objects of study in differential topology, but the more general notion of fibration is all that is necessary for homotopy theory, and thus homology (fiber bundles are a special sub-class of fibrations).

In the case where the space X is a simplicial complex built from data \mathbf{X} , the situation is rarely so clean as it is in the case of continuous spaces. In practice, it is often easiest to break data into smaller subsets.

Definition 1.2.1. *Let X be a set. A cover of X is a collection of sets \mathcal{U} that contain X*

$$X \subseteq \bigcup_{U \in \mathcal{U}} U$$

We will only consider covers of X where each $U \in \mathcal{U}$ is a subset of X , implying $X = \bigcup_{U \in \mathcal{U}} U$ in Definition 1.2.1.

We can view a cover of X as a parameterization of the data as well. To take this view, we might take $p : X \rightarrow \mathcal{U}$ to be a choice of assignment from a point $x \in X$ to some set $U \in \mathcal{U}$ that contains the point. One might also consider pullbacks of a cover of B in Equation (1.5). When $U \subseteq B$ has some notion of locality, we'll also refer to $f^{-1}(U)$ as a fiber instead of the more general *inverse image*.

The viewpoint of treating covers as topological spaces in their own right is encoded in the *Nerve* of a cover:

Definition 1.2.2. *The Nerve of a cover \mathcal{U} , denoted $\mathcal{N}(\mathcal{U})$ is a simplicial complex with the sets in \mathcal{U} serving as the underlying vertex set. The simplex (U_0, \dots, U_k) is present if $\bigcap_{i=0}^k U_i \neq \emptyset$.*

For example, Čech and Dowker complexes both are explicitly defined as Nerves of covers of balls of radius r . Under certain conditions, the Nerve of a cover can actually give us complete topological information about the underlying set X .

Theorem 1.2.3. *(Nerve Theorem [14]) Let \mathcal{U} be a cover of a paracompact space X , where if $\bigcap U_i \neq \emptyset$, then $\bigcap U_i$ is contractible. Then $\mathcal{N}(\mathcal{U}) \simeq X$.*

A proof can be found in [57]. The mapper algorithm [92] is a prominent example of how the nerve of cover has been used in topological data analysis. Specifically, the Mapper construction takes a map $p : \mathbf{X} \rightarrow \mathbf{B}$, and constructs the nerve of the pullback of a cover of \mathbf{B} , where the inverse image of sets are broken up into connected components. We will investigate a generalization of these ideas to geometric complexes built on data in Chapter 4.

1.2.3 Calculation Tools in Algebraic Topology

For the most part, algorithms for topological data analysis have focused on making general-purpose persistent homology calculations very fast on a computer. This lies against a backdrop of calculations performed in pure algebraic topology, which for the better part of a century has developed techniques for calculations that are done “by hand.” In contrast to general purpose algorithms, these techniques typically exploit some structure in the problem in order to make these calculations tractable on paper. We’ll review how several techniques might be used to compute the integral homology of the Klein bottle.

Exact and Spectral Sequences

We’ll first provide algebraic definitions for certain types of diagrams that feature prominently in homological algebra.

Definition 1.2.4. *An exact sequence is a sequence of R -modules $\{A_i\}$ connected by maps $\{\phi_i : A_i \rightarrow A_{i+1}\}$*

$$\dots \longrightarrow A_i \xrightarrow{\phi_i} A_{i+1} \xrightarrow{\phi_{i+1}} A_{i+2} \longrightarrow \dots \quad (1.8)$$

with the property that $\ker \phi_{i+1} = \text{img } \phi_i$.

Exact sequences of indeterminate length as in Equation (1.8) are typically referred to as *long exact sequences*, whereas *short exact sequences* typically include at most three non-zero terms:

$$0 \longrightarrow A \longrightarrow B \longrightarrow C \longrightarrow 0 \quad (1.9)$$

The utility of exact sequences is that because $\ker \phi_{i+1} = \text{img } \phi_i$, only partial information in diagrams such as the one in Equation (1.9) is needed to determine the whole diagram. Examples will be found later in this section.

Spectral sequences are more complicated objects. First, we define a differential on a graded R -module as a collection of maps $\{\partial\}$ such that $\partial^2 = 0$. One such example

of this are the boundary maps ∂ in chain complexes (the grading is given by the dimension of the cells).

Definition 1.2.5. A spectral sequence (of homological type) is a collection of differential bigraded R -modules $\{E_{*,*}^r, \partial^r\}$, where $r = 1, 2, \dots$. The differentials ∂^r are of bidegree $(-r, r - 1)$, and for all p, q, r , $E_{p,q}^{r+1} = H_{p,q}(E_{*,*}^r, \partial^r)$.

The collection $E_{*,*}^r$ is known as the E^r page of the spectral sequence. The development of spectral sequences is discussed in much more depth than we will here in McCleary's book [75], as well as other sources in homological algebra [35, 51]. As a calculation tool, we will primarily be using the Leray-Serre spectral sequence [90], which is a first-quadrant spectral sequence defined beginning on the E^2 page (meaning $E_{p,q}^2 = 0$ if $p < 0$ or $q < 0$), so has differentials of bidegree $(-2, 1)$:

$$\begin{array}{ccccccc}
 0 & \longleftarrow & E_{0,2}^2 & \longleftarrow & E_{1,2}^2 & \longleftarrow & E_{2,2}^2 & E_{3,2}^2 \\
 & & & & & & & & \\
 0 & \longleftarrow & E_{0,1}^2 & \longleftarrow & E_{1,1}^2 & \longleftarrow & E_{2,1}^2 & E_{3,1}^2 \\
 & & & & & & & & \\
 0 & \longleftarrow & E_{0,0}^2 & \longleftarrow & E_{1,0}^2 & \longleftarrow & E_{2,0}^2 & E_{3,0}^2 \\
 & & & & & & & & \\
 0 & & 0 & & 0 & & 0 & & 0
 \end{array} \tag{1.10}$$

where we have only displayed arrows for the differentials which have both source and target displayed, and the diagram extends in all directions to include $E_{p,q}^2$ for all p, q .

Spectral sequences *collapse* at the E^r page when the $\partial^N = 0$ for all $N \geq r$. This means that $E_{p,q}^*$ terms cease to change for all subsequent pages, since then $\ker \partial_{p,q}^r = E_{p,q}^r$, and $\text{img } \partial_{p-r,q+r-1}^r = 0$, so $E_{p,q}^{r+1} = H_{p,q}(E_{*,*}^r, \partial^r) = E_{p,q}^r$. Typically the sparsity structure of the non-zero terms allows this condition to be verified. We will only encounter situations in which spectral sequences collapse at a finite page, in which case the colimit $E_{p,q}^\infty = \text{colim}_{r \rightarrow \infty} E_{p,q}^r$ is determined at the point at which the sequence collapses.

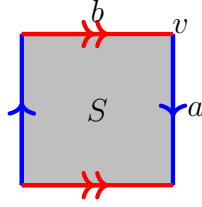


Figure 1.6: CW structure for the Klein bottle.

A spectral sequence $\{E_{*,*}^r, d^r\}$ converges to a graded R -module H_* if H_* has a filtration F_* , meaning

$$\cdots \subseteq F_p H_* \subseteq F_{p+1} H_* \subseteq \cdots \tag{1.11}$$

such that

$$E_{p,q}^\infty \cong F_p H_{p+q} / F_{p+1} H_{p+q} \tag{1.12}$$

this means that H_k is determined by $E_{i,k-i}^\infty$ up to an extension problem.

We'll now turn to actual computations.

Computation using a CW Complex

We'll begin with a description of the Klein bottle as a CW complex K . The red lines (b) are identified to form a cylinder, and the blue lines (a) are identified (note the twist) to form the Klein bottle. There is a single 0-cell v . There are two 1-cells, a and b , with $\partial_1 a = \partial_1 b = v - v = 0$, that is both a and b are cycles. There is a single 2-cell, S , with $\partial_2 S = b - a - b - a = -2a$. Now, we compute the homology groups of the chain complex, $H_i = \ker \partial_i / \text{img } \partial_{i+1}$

$$H_0(K) = \ker \partial_0 / \text{img } \partial_1 = \langle v \rangle / 0 = \mathbb{Z}$$

$$H_1(K) = \ker \partial_1 / \text{img } \partial_2 = \langle a, b \rangle / \langle 2a \rangle = \mathbb{Z} \oplus \mathbb{Z}_2$$

$$H_2(K) = \ker \partial_2 / \text{img } \partial_3 = 0 / 0 = 0$$

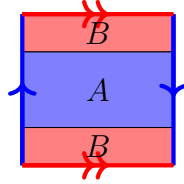


Figure 1.7: Klein bottle covered by two Möbius bands, denoted A and B .

where \mathbb{Z}_2 denotes the quotient ring $\mathbb{Z}/2\mathbb{Z}$. Since there are no higher-dimensional cells, $H_i(K) = 0$ for $i > 2$. Thus,

$$H_i(K) = \begin{cases} \mathbb{Z} & i = 0 \\ \mathbb{Z} \oplus \mathbb{Z}_2 & i = 1 \\ 0 & i \geq 2 \end{cases} \quad (1.13)$$

The Mayer-Vietoris Sequence

The Mayer-Vietoris sequence (c.f. [57] Chapter 2.2) computes the homology of a space X by using a cover with two sets A and B . Suppose that $A, B \subset X$, such that $X = A \cup B$. Then there is a long exact sequence of the form

$$\cdots \rightarrow H_n(A \cap B) \xrightarrow{\phi} H_n(A) \oplus H_n(B) \xrightarrow{\psi} H_n(X) \xrightarrow{\partial} H_{n-1}(A \cap B) \rightarrow \cdots \quad (1.14)$$

where $\phi(x) = (x, -x)$, and $\psi(x, y) = x + y$, and ∂ is determined by the conditions necessary to make the sequence exact. Now, the trick is to find a cover of K that lets us apply this sequence. We can create a Klein bottle by stitching together two Möbius bands along their boundary.

The homology of a Möbius band M is

$$H_i(M) = H_i(S^1) = \begin{cases} \mathbb{Z} & i = 0 \\ \mathbb{Z} & i = 1 \\ 0 & i \geq 2 \end{cases} \quad (1.15)$$

Since M retracts to the circle S^1 . Now, by covering K by two Möbius bands, we denote $A = M_1$, $B = M_2$, and $A \cap B = S^1$. First, we note that the LES looks like

$$\cdots \rightarrow 0 \rightarrow H_2(K) \rightarrow \mathbb{Z} \xrightarrow{\phi} \mathbb{Z} \oplus \mathbb{Z} \rightarrow \cdots$$

Where the maps $\phi : \mathbb{Z} \rightarrow \mathbb{Z} \oplus \mathbb{Z}$ is given by $1 \mapsto (2, -2)$, since the boundary wraps around the Möbius band twice. Since this map is injective, we must have $H_2(K) = 0$. Next, note that the H_0 portion of the sequence is

$$H_1(K) \rightarrow \mathbb{Z} \rightarrow \mathbb{Z} \oplus \mathbb{Z} \rightarrow \mathbb{Z} \rightarrow 0$$

This means that the map $H_1 \rightarrow \mathbb{Z}$ must be the zero map.

Now, since the sequence is exact, we have that $H_1(K)$ is $\mathbb{Z} \oplus \mathbb{Z}/(2, -2) \cong \mathbb{Z} \oplus \mathbb{Z}_2$, seen by using a basis for $\mathbb{Z} \oplus \mathbb{Z}$ as $(1, 0)$, $(1, -1)$.

The Mapping Torus Sequence

We'll now turn to another long exact sequence associated with the mapping torus (c.f. [57] Example 2.48). Suppose $f, g : X \rightarrow Y$, and define $Z = X \times I \sqcup Y / \sim$, where $(x, 0) \sim f(x)$, and $(x, 1) \sim g(x)$. In the case that f is the identity, this is known as the mapping torus of g , and can be expressed as $X \times I / \sim$, where $(x, 0) \sim (g(x), 1)$. The exact sequence has the form

$$\cdots \rightarrow H_n(X) \xrightarrow{f_* - g_*} H_n(Y) \xrightarrow{i_*} H_n(Z) \rightarrow H_{n-1}(X) \rightarrow \cdots$$

The Klein bottle is the mapping torus for the reflection $g : S^1 \rightarrow S^1$, $g : x \mapsto -x$. In this case, the interesting part of the sequence is

$$\begin{array}{ccccccc} 0 \rightarrow H_2(K) \rightarrow H_1(S^1) \xrightarrow{I-g_*} H_1(S^1) \rightarrow H_1(K) \rightarrow H_0(S^1) \xrightarrow{I-g_*} H_0(S^1) & & & & & & (1.16) \\ & & \parallel & & \parallel & & \\ & & \mathbb{Z} \xrightarrow{2} \mathbb{Z} & & \mathbb{Z} \xrightarrow{0} \mathbb{Z} & & \end{array}$$

Thus, $H_2(K) = 0$, and there is a short exact sequence

$$0 \rightarrow \mathbb{Z}_2 \rightarrow H_1(Z) \rightarrow \mathbb{Z} \rightarrow 0$$

Which splits, since \mathbb{Z} is free, so $H_1(Z) = \mathbb{Z}_2 \oplus \mathbb{Z}$.

The Leray-Serre Spectral Sequence

The Leray-Serre spectral sequence gives a way to compute the homology of a fibration. In the case of the Klein bottle, there is a fibration $p : K \rightarrow S^1$, with fibers also S^1 .

$$\begin{array}{ccc} S^1 & \hookrightarrow & K \\ & & \downarrow p \\ & & S^1 \end{array} \quad (1.17)$$

The fibration is not trivial, meaning there is a “twist” when connecting the fibers when following a non-trivial path around the base space. This structure can be seen in Figure 1.6, where the 1-cell b is the base space S^1 , and a fiber over v is the 1-cell a . We can see the twist in the identification of the left and right sides of the 2-cell.

The Serre spectral sequence lets us compute homology of K by computing homology of the base space S^1 with local coefficients in homology of the fiber, also S^1 . For a fibration $F \rightarrow E \rightarrow B$, and for a ring R , there is a first-quadrant spectral sequence converging to $H_*(E; R)$ where the terms on the E^2 page are given by

$$E_{p,q}^2 = H_p(B; \mathcal{H}_q(F))$$

Where $\mathcal{H}_q(F)$ is a local coefficient system over B in terms of the homology of the fiber F .

In the case of the Klein bottle, the non-trivial fibration creates a non-trivial local coefficient system over B . The fiber $F = S^1$, so $H_0(F) = H_1(F) = \mathbb{Z}$. For $\mathcal{H}_0(F)$, we have the trivial coefficient system $H_0(F) = \mathbb{Z}$. For $\mathcal{H}_1(F)$, we have a twisted coefficient system \mathbb{Z}_- , corresponding to the non-trivial \mathbb{Z} -automorphism $x \mapsto -x$ induced by the antipodal map $S^1 \rightarrow S^1$.

We will use that $H_i(S^1; \mathbb{Z}_-) = \mathbb{Z}_2$ for $i = 0$, and is 0 for all other i (see Section 5.2.4 for calculations). Thus, the E^2 page of the spectral sequence in the first quadrant looks like

$$\begin{array}{ccc} 0 & 0 & 0 \\ \mathbb{Z}_2 & 0 & 0 \\ \mathbb{Z} & \mathbb{Z} & 0 \end{array} \tag{1.18}$$

The spectral sequence collapses at the E^2 page, since all differentials either map to 0 or from 0 (see Equation (1.10) for illustration).

We can immediately see that $H_0(K) = \mathbb{Z}$ because $E_{0,0}^\infty = \mathbb{Z}$, and all other $E_{i,-i}^\infty = 0$. This agrees with the fact that the Klein bottle has one connected component. In the case of $H_1(K)$, the filtration across the first anti-diagonal includes \mathbb{Z}_2 and \mathbb{Z} . We have $F_1H_1(K) = \mathbb{Z}$ because preceding terms on the anti-diagonal are 0, and since there are two terms on the anti-diagonal, $F_2H_1(K) = H_1(K)$. Writing conditions for the convergence of the spectral sequence, we have $F_2H_1(K)/F_1H_1(K) = \mathbb{Z}_2$, so $H_1(K)/\mathbb{Z} = \mathbb{Z}_2$. It follows that $H_1(K) = \mathbb{Z} \oplus \mathbb{Z}_2$. Note that there are no non-zero terms on higher anti-diagonals of the E^∞ page, so $H_k(K) = 0$ for $k \geq 2$.

1.3 Concerning Torsion

In algebraic topology, homology is often computed with coefficients in the integer ring \mathbb{Z} (as in Section 1.2.3). In the context of persistent and zigzag homology, field coefficients \mathbb{F} are used, which not only makes computations a bit more straightforward, but allows classification results from quiver representation theory and modules over principal ideal domains to apply, as discussed in Section 3.2. We will briefly review how different choices of coefficient play out when torsion is present in integral homology, as this will play an important role for spaces, such as the Klein bottle, that are featured in Chapter 5. Material in this section is standard, and as references, we recommend [35, 57].

The field coefficients we will work with will be the integers modulo a prime p , $\mathbb{F} = \mathbb{Z}/p\mathbb{Z} = \mathbb{F}_p$, and the rationals, $\mathbb{F} = \mathbb{Q}$. We can obtain chain complexes with coefficients in \mathbb{F} from chain complexes with coefficients in \mathbb{Z} via the tensor product $\mathbb{F} \cong \mathbb{Z} \otimes \mathbb{F}$. Explicitly, because \otimes is a functor, when forming boundary matrices, we can simply just apply the map $a \mapsto a \otimes 1$ to each entry $a \in \mathbb{Z}$ of the matrix. In the case of the rationals \mathbb{Q} , this just sends $a \mapsto a$ since $\mathbb{Z} \subset \mathbb{Q}$, and for \mathbb{F}_p , this is the map $a \mapsto a \pmod{p}$.

We will use $H_k(C_*; R)$ to denote homology with coefficients in R , and $H_k(X; R)$ when C_* is obtained by applying the chain functor to a space X . It is perhaps easiest to first see things play out in an example chain complex obtained from the CW complex for the Klein bottle defined in Section 1.2.3

$$C_*(K; \mathbb{Z}) = 0 \longrightarrow \mathbb{Z} \xrightarrow{(\times 2) \times 0} \mathbb{Z} \oplus \mathbb{Z} \xrightarrow{0} \mathbb{Z} \longrightarrow 0 \quad (1.19)$$

In this case, one can easily compute, as in Section 1.2.3, $H_0(K; \mathbb{Z}) = \mathbb{Z}$, $H_1(K; \mathbb{Z}) = \mathbb{Z} \oplus \mathbb{Z}_2$, and $H_i(K; \mathbb{Z}) = 0$ for $i \geq 2$. In the case where $\mathbb{F} = \mathbb{F}_2 = \mathbb{Z}/2\mathbb{Z}$, the “ $\times 2$ ” map in Equation (1.19) becomes the zero map.

$$C_*(K; \mathbb{F}_2) = 0 \longrightarrow \mathbb{F}_2 \xrightarrow{0} \mathbb{F}_2 \oplus \mathbb{F}_2 \xrightarrow{0} \mathbb{F}_2 \longrightarrow 0 \quad (1.20)$$

and we can compute

$$H_k(K; \mathbb{F}_2) = \begin{cases} \mathbb{F}_2 & k = 0, 2 \\ \mathbb{F}_2 \oplus \mathbb{F}_2 & k = 1 \\ 0 & k > 2 \end{cases} \quad (1.21)$$

This has the effect of producing new nontrivial homology one dimension higher than the torsion. Using $\mathbb{F} = \mathbb{Q}$ or $\mathbb{F} = \mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$ for $p \neq 2$ has the effect of making the “ $\times 2$ ” map an isomorphism, denoted 1:

$$C_*(K; \mathbb{F}) = 0 \longrightarrow \mathbb{F} \xrightarrow{1 \times 0} \mathbb{F} \oplus \mathbb{F} \xrightarrow{0} \mathbb{F} \longrightarrow 0 \quad (1.22)$$

and we can compute

$$H_k(K; \mathbb{F}) = \begin{cases} \mathbb{F} & k = 0, 1 \\ 0 & k > 1 \end{cases} \quad (1.23)$$

This had the effect of “killing” the torsion. In general, tensoring with \mathbb{Q} will kill all torsion, whereas tensoring with \mathbb{F}_p will have effects depending on the the integral torsion submodule.

The more general way to obtain $H_k(C_*; R)$ is through the universal coefficient theorem for homology from the “universal” coefficient group \mathbb{Z} . This will be useful in situations where we don’t use explicit cellular methods to compute homology, as in Chapter 5.

Theorem 1.3.1. (c.f. [57], 3A.3) *If C_* is a chain complex of free abelian groups, there are natural short exact sequences*

$$0 \longrightarrow H_n(C_*; \mathbb{Z}) \otimes R \longrightarrow H_n(C_*; R) \longrightarrow \text{Tor}(H_{n-1}(C_*; \mathbb{Z}), R) \longrightarrow 0 \quad (1.24)$$

for all n and rings R , and these sequences split, though not naturally. (Tor is the $\text{Tor}_1^{\mathbb{Z}}$ functor).

A useful discussion of Tor can be found in Davis and Kirk [35], chapter 2. We can use the following proposition to compute homology with $\mathbb{F}_p = \mathbb{Z}_p$ coefficients:

Proposition 1.3.2. ([35], Prop 2.4.2) *Let M be a \mathbb{Z} -module, and $a \in \mathbb{Z}$. Then*

$$\text{Tor}(M, \mathbb{Z}_a) = {}_aM = \{m \in M \mid am = 0\} \quad (1.25)$$

Proof. We’ll use the fact that $\text{Tor}(M, \mathbb{Z}_a) \cong \text{Tor}(\mathbb{Z}_a, M)$ ([35], Cor 2.7). We have a short exact sequence

$$0 \longrightarrow \mathbb{Z} \xrightarrow{\times a} \mathbb{Z} \longrightarrow \mathbb{Z}/a\mathbb{Z} \longrightarrow 0 \quad (1.26)$$

Applying the right-exact functor $\otimes M$, we have the exact sequence

$$0 \longrightarrow \operatorname{Tor}(\mathbb{Z}_a, M) \longrightarrow \mathbb{Z} \otimes M \xrightarrow{\times a} \mathbb{Z} \otimes M \longrightarrow \mathbb{Z}_a \otimes M \longrightarrow 0 \quad (1.27)$$

The $\times a$ map in the middle has kernel $\{m \in M \mid am = 0\}$. □

As a corollary, $\operatorname{Tor}(\mathbb{Z}_a, \mathbb{Q}) = \operatorname{Tor}(\mathbb{Q}, \mathbb{Z}_a) = 0$ because \mathbb{Q} has no zero-divisors. We'll also use the fact that $\operatorname{Tor}(\mathbb{Z}, \mathbb{Q}) = 0$ (because \mathbb{Z} is free).

We'll now compute homology of the Klein bottle K with \mathbb{F}_2 coefficients starting with the integral homology (Equation (1.13)). For $H_0(K; \mathbb{F}_2)$, we have $H_0(K; \mathbb{Z}) \otimes \mathbb{F}_2 = \mathbb{F}_2$, and $H_{-1}(K; \mathbb{Z}) = 0$, so the short exact sequence in Equation (1.24) becomes

$$0 \rightarrow \mathbb{F}_2 \rightarrow H_0(K; \mathbb{F}_2) \rightarrow 0 \rightarrow 0 \quad (1.28)$$

from which we compute $H_0(K; \mathbb{F}_2) = \mathbb{F}_2$. For $H_1(K; \mathbb{F}_2)$, we use $(\mathbb{Z} \oplus \mathbb{Z}_2) \otimes \mathbb{F}_2 = \mathbb{F}_2 \oplus \mathbb{F}_2$, and $\operatorname{Tor}(\mathbb{Z}; \mathbb{F}_2) = 0$, so the short exact sequence becomes

$$0 \rightarrow \mathbb{F}_2 \oplus \mathbb{F}_2 \rightarrow H_1(K; \mathbb{F}_2) \rightarrow 0 \rightarrow 0 \quad (1.29)$$

from which we compute $H_1(K; \mathbb{F}_2) \cong \mathbb{F}_2 \oplus \mathbb{F}_2$. Finally, for $H_2(K; \mathbb{F}_2)$, we simply need that $\operatorname{Tor}(\mathbb{Z} \oplus \mathbb{Z}_2, \mathbb{F}_2) \cong \mathbb{F}_2$, from Proposition 1.3.2.

$$0 \rightarrow 0 \rightarrow H_2(K; \mathbb{F}_2) \rightarrow \mathbb{F}_2 \rightarrow 0 \quad (1.30)$$

from which, we have $H_2(K; \mathbb{F}_2) = \mathbb{F}_2$. At this point, we agree with Equation (1.21).

We'll now compute field coefficients for $\mathbb{F} = \mathbb{F}_p$, with $p > 2$ prime, or $\mathbb{F} = \mathbb{Q}$. Because $\operatorname{Tor}(H_k(K; \mathbb{Z}), \mathbb{F}) = 0$ for all these fields, we have that $H_k(K; \mathbb{F}) = H_k(K; \mathbb{Z}) \otimes \mathbb{F}$, which agrees with Equation (1.23).

1.4 Roadmap & Contributions

The main contributions of this dissertation are found in Chapters 3, 4 and 5, and detailed information about relevant prior work and contributions will be found in each of these chapters. These chapters are more or less independent of each other, so we will provide an overview to help an interested reader decide where to begin.

Chapter 2 contains technical background which explains the computation of ordinary and persistent homology, as well as induced maps on homology. The majority of this material is standard in algebraic topology and topological data analysis, but we provide explicit algorithms to compute induced maps on homology using the output of standard reduction algorithms in a way that, to the best of our knowledge, has not yet appeared in the literature.

Chapter 3 lays out a new computational approach to computing persistent and zigzag homology using matrix factorizations on quiver representations. In conjunction with Chapter 2, this provides a computational framework to compute persistent and zigzag barcodes on diagrams of spaces using arbitrary induced maps on homology. This capability is not present in existing topological data analysis packages, and our implementation opens up new potential applications of these constructions. In addition, our algorithms offer multiple opportunities for parallelization which, in combination, do not exist in other computational frameworks. This chapter consists of joint work with Gunnar Carlsson and Anjan Dwaraknath, and has appeared in a preprint along with portions of Chapter 2 [24]. As part of this collaboration, I surveyed the existing literature on the topic, worked closely with my co-authors in the development of our algorithms, wrote significant portions of the code and paper, and implemented the examples found at the end of the chapter.

Chapter 4 first extends the classical theory of carriers to the filtered setting, and shows how this can be used to compare persistent homology of different filtrations through the construction of interleavings. This is then used to compare persistent homology of various geometric constructions. Finally, we introduce a new family of

complexes parameterized by a covers of data, and prove results relating the construction to the nerve of the cover, showing stability to perturbation of data, and relating the construction to the non-parameterized construction.

Chapter 5 generalizes the Klein bottle in image patches introduced in Section 1.2.1 to higher-dimensional images. First, we introduce the Harris map, which sends patches to their direction of largest variation, and show that this map is a fibration on a model Klein bottle. We then generalize this map and construction to higher-dimensional patches and compute the homology of the total space using the Leray-Serre spectral sequence. Finally, we investigate two three-dimensional image data sets, and relate the topological structure of patches to subspaces of the model.

Chapter 2

Preliminaries

We will now introduce the algebraic and topological tools necessary for our algorithm. Specifically, we will need to compute homology of cell complexes, and compute induced maps on homology, which we will connect to existing methods for computing persistent homology. Finally, we will present an approach for using existing persistence algorithms for non-inclusion based maps using mapping cylinders.

While this section is reasonably self-contained for the purposes of setting up our algorithm and establishing notation, this material is fairly standard and can be found in a variety of locations. For a general reference for matrix computations, we defer to Golub and Van Loan's text [53]. For concepts in algebraic topology, we refer to Hatcher [57]. Early sources for computing persistent homology and zigzag homology are [46, 106] and [22, 23] respectively. Further background on computational topology can be found in [45] and for an overview of modern TDA software we recommend [83].

2.1 Algebra and Notation

We will use Householder notation for linear algebra [53, 61]. Upper case Greek or Latin letters such as A or Λ will refer to matrices, lower case Latin letters such as a will refer to vectors, and lower case Greek letters such as λ will refer to scalars. This is not always consistent with notation found in algebraic topology or the TDA literature, which does not have conventions as strong as Householder notation, but we

will attempt to use notation that is close to modern use. One special symbol, ∂ , will always refer to a boundary matrix. General topological spaces will be denoted with upper case Latin letters such as X , samples with bold face, such as \mathbf{X} , and simplicial and cell complexes with calligraphic font as in \mathcal{X} .

We will prefer to work with matrix factorizations when possible. Due to the abundance of subscripts in other contexts, we will use square brackets instead of subscripts for indexing. Vectors will be assumed to be column vectors, and v^T will denote the corresponding row vector. e_i will refer to the vector with 1 in the i -th entry and 0 elsewhere, with dimension determined by context. When we need to access elements in vectors, we will use the notation $x[i]$ to denote the scalar value $e_i^T x$, or the i -th entry in x . When we need to access elements of matrices, we will use the notation $A[i, j]$ to denote the scalar $e_i^T A e_j$, or the entry in the i -th row and j -th column of A . When we want to indicate columns of matrices, we will use the notation $A[j]$ to denote the vector $A e_j$, or the j -th column of A . As is standard in numerical linear algebra, indices will begin with 1, meaning the valid range of indexes for a vector $x \in \mathbb{F}^n$ is $1, \dots, n$. Asterisks indicate that a subscript or superscript runs over a range of values, determined by context. For example, F_* is often used to represent $F_k, k = 0, 1, \dots$.

Computations will be done in vector spaces over a fixed field \mathbb{F} . Computations in topological data analysis are typically done over finite fields, for example $\mathbb{F} = \mathbb{F}_2 = \mathbb{Z}/2\mathbb{Z}$, or the rationals \mathbb{Q} , because homology requires exact computation of kernels and images. Floating point arithmetic is typically avoided due to numerical issues. One exception is Hodge theory, which uses the more familiar fields \mathbb{R} or \mathbb{C} – for a numerical and application focused introduction see [70].

Rank-nullity theorem: Let $A : V \rightarrow W$, where V, W are finite-dimensional vector spaces. Then $\text{rank } A + \dim \ker A = \dim V$.

Change of Basis : Suppose that we have vector spaces V and W with bases $B^V = \{b_i^V\}$ and $B^W = \{b_i^W\}$ respectively and a linear transformation $T : V \rightarrow W$ represented by a matrix A in bases B^V and B^W . That is, $Tb_j^V = \sum_i A[i, j]b_i^W$. Now,

suppose that we have new bases $C^V = \{c_i^V\}$ and $C^W = \{c_i^W\}$, where $c_i^* = U^*b_i^*$. Then if we wish to write T in terms of bases C^V and C^W , we can write a new matrix

$$\hat{A} = (U^W)^{-1}AU^V \quad (2.1)$$

The matrix U^V first maps coefficients in C^V to coefficients in B^V , then the linear transformation T is applied, mapping to coefficients in B^W , which are then mapped to coefficients in C^W via $(U^W)^{-1}$.

Quotient Vector Spaces: Homology is an example of a quotient vector space. Simply, if V is a vector space, and $W \subseteq V$ is a subspace, then the quotient space V/W has elements (cosets) $[v]$, where if $v \in [v]$, then $v + w \in [v]$ for any $w \in W$. The set V/W is also endowed with a vector space structure, so is called a quotient vector space. The quotient operation is a linear transformation $V \rightarrow V/W$, meaning

$$[\alpha x + \beta y] = \alpha[x] + \beta[y] \quad (2.2)$$

We will often represent elements $[v] \in V/W$ using a *representative* $v \in [v]$. The choice of representative is not unique, as $v + w$ for any $w \in W$ may also be used as a representative.

Four Fundamental Subspaces: Let $T : V \rightarrow W$. There are four fundamental subspaces associated with T : the kernel,

$$\ker T \subseteq V = \{v \in V \mid T(v) = 0\}$$

the image

$$\text{img } T \subseteq W = \{w \in W \mid w = T(v), v \in V\}$$

the cokernel

$$\text{coker } T = W / \text{img } T$$

and the coimage

$$\text{coimg } T = V / \ker T.$$

The cokernel and coimage both are quotient vector spaces. Note that Tv is identical for all $v \in [v] \in \text{coimg } T$.

Triangular Matrices: Triangular matrices only contain non-zero elements on one side of a diagonal. Common triangular forms are lower and upper triangular matrices, where $L \in \mathbb{F}^{m \times n}$ is lower-triangular if $L[i, j] = 0$ if $j > i$, and $U \in \mathbb{F}^{m \times n}$ is upper triangular if $U[i, j] = 0$ if $j < i$. Triangular matrices are well extremely useful because solving linear systems using backward substitution [53] is as expensive as regular matrix multiplication. That is, solving $Tx = b$ for unknown x is as expensive as forming the matrix-vector product Tx when T is upper or lower triangular. Backward substitution can be used successfully even when a triangular matrix is not invertible, either because it is not square or because it is rank deficient, as long as b is in the column space of T (i.e. the linear system is consistent).

The J Matrix:

Definition 2.1.1. J is a square $n \times n$ matrix, such that

$$J[i, j] = \begin{cases} 1, & \text{if } i = n - j - 1 \\ 0, & \text{otherwise} \end{cases}$$

In other words, it is the anti-diagonal permutation matrix. Specifically, when applied to a matrix on the left, it reverses the row order, and when applied to a matrix on the right it reverses the column order. It is its own inverse:

$$J^{-1} = J.$$

A common operation is to conjugate with the J matrix, which reverses both row and column order and thus produces a reflection across the anti-diagonal. Note that this operation is distinct from taking the transpose of a matrix and cannot be

expressed in terms of it.

$$JAJ^{-1} = JAJ = A'$$

The following are useful commutation relations between the J matrix and other matrix shapes

$$JL = UJ \qquad \square \square = \square \square \qquad (2.3)$$

$$JU = LJ \qquad \square \square = \square \square \qquad (2.4)$$

$$JE_L = \hat{E}_U J \qquad \square \square = \square \square \qquad (2.5)$$

$$JE_U = \hat{E}_L J \qquad \square \square = \square \square \qquad (2.6)$$

where E matrices are defined below.

Pivot Matrices:

Definition 2.1.2. *A pivot matrix is a matrix in which every row and column has at most one non-zero element.*

In the context of matrix factorizations used in this work, the non-zero element will always be 1 (the multiplicative identity of the field \mathbb{F}) by convention. The term pivot matrix refers to its use in recording pivots (last non-zeros of rows or columns) when computing matrix factorizations. Pivot matrices are similar to permutation matrices in the sense that they map a single basis element to a single basis element, but contain the possibility that some rows and columns can be entirely zero. Thus they are not generally invertible.

Lemma 2.1.3. *The class of pivot matrices is closed under multiplication*

Proof. For a pivot matrix Q , define $i(j)$ to be the index of the non-zero row of column j if column j has a pivot, and $i(j) = \infty$ otherwise. Additionally, define $i(\infty) = \infty$. Thus, we can write columns of Q as

$$Q[j] = e_{i(j)}$$

where $e_\infty = 0$.

Let Q_1 and Q_2 be pivot matrices with compatible dimensions to form the product $A = Q_2Q_1$, then column j of A can be written as

$$Ae_j = Q_2(Q_1e_j) = Q_2e_{i_1(j)} = e_{i_2(i_1(j))}$$

So column $A[j]$ has at most one nonzero, with pivot $i_2(i_1(j))$. \square

Echelon Pivot Matrices: Echelon pivot matrices are pivot matrices with added structure. There are 4 types we consider

$$E_L = \begin{bmatrix} \times & & \\ & \times & \\ & & \times \end{bmatrix} \quad E_U = \begin{bmatrix} \times & & \\ & \times & \\ & & \times \end{bmatrix} \quad (2.7)$$

$$\hat{E}_L = \begin{bmatrix} \times & & \\ & \times & \\ & & \times \end{bmatrix} \quad \hat{E}_U = \begin{bmatrix} \times & & \\ & \times & \\ & & \times \end{bmatrix} \quad (2.8)$$

E_L and \hat{E}_U contain the pivots for variants of the column echelon form of a matrix, and E_U and \hat{E}_L contain the pivots for variants of the row echelon form of a matrix. The L and U subscript indicates whether the matrix is lower or upper triangular.

Definition 2.1.4. A matrix has the E_L shape if it is the sum of rank 1 matrices created from basis vectors

$$E_L = \sum_{(i,j) \in S} e_i e_j^T$$

The set $S \subset \{1, \dots, m\} \times \{1, \dots, n\}$ contains the locations of the pivots. Since E_L is a pivot matrix, for every j , there must be a unique i , therefore the pairs can be written as $(i(j), j)$. The function $i(j)$ is defined on the subset of columns that have a pivot, and must satisfy the following properties

1. $j_1 < j_2 \implies i(j_1) < i(j_2)$ on the domain of $i(j)$
2. For every j_1, j_2 s.t. $j_1 < j_2$ and $A[j_1] = 0 \implies A[j_2] = 0$

The other echelon pivot matrices can be defined in terms of the E_L shape.

Definition 2.1.5. A matrix is of shape E_U if its transpose is of shape E_L

$$(E_U)^T = E_L$$












Symbol	Meaning	Shape
A	Arbitrary matrix	
D	Diagonal matrix	
L	Lower-triangular	
U	Upper-triangular	
T	Any triangular form	
S	Schur Complement	
P	Permutations	
I	Identity	
J	Anti-diagonal permutation	
E	Echeleon-diagonal	
E_L	Echelon pivot lower	
\hat{E}_L	Echelon pivot lower	
E_U	Echelon pivot upper	
\hat{E}_U	Echelon pivot upper	
Q	Pivot matrix	

Figure 2.1: Notation for different matrices, along with pictorial symbols

Definition 2.1.6. A matrix is of shape \hat{E}_L if its J -Conjugate is of shape E_L

$$J\hat{E}_LJ = E_L$$

Definition 2.1.7. A matrix is of shape \hat{E}_U if its J -Conjugate is of shape E_U

$$J\hat{E}_UJ = E_U$$

2.2 Cell Complexes

We will now turn to topological notions. First, we need to know how we can construct topological spaces from basic building blocks such as cells or simplices. It turns out that many topological spaces of practical interest can be represented as cellular or simplicial complexes. See Hatcher [57] for a more complete discussion.

A *cell complex*, or CW complex \mathcal{X} can be built inductively by starting with a discrete set of points (0-cells) \mathcal{X}^0 called the 0-skeleton, and inductively forming the k -skeleton \mathcal{X}^k from \mathcal{X}^{k-1} by adding open k -dimensional balls along their boundary to \mathcal{X}^{k-1} .

Cell complexes offer a general way to encode spaces, but in many applications the need to specify all boundary maps can be onerous. Often it is easier to use simplicial or cubical complexes, both of which are special cases of cell complexes, for which the boundary maps come for free. We will focus on simplicial complexes, which are commonly used for the purposes of triangulating spaces. A k -simplex is simply the convex hull of $k + 1$ points in general position, denoted $s = (x_0, \dots, x_k)$. A k -simplex has $k + 1$ faces which are $(k - 1)$ -simplices in its boundary, each of which can be obtained by removing a single vertex $\partial(x_0, \dots, x_k) = \{(x_0, \dots, \hat{x}_i, \dots, x_k)\}$, where \hat{x}_i indicates that the i th point has been removed. A *simplicial complex* \mathcal{X} is a collection of simplices, where if a simplex $s \in \mathcal{X}$, its boundary is in \mathcal{X} : $\partial s \subset \mathcal{X}$.

Definition 2.2.1. *Let \mathcal{X} be a cell complex with zero-skeleton $\mathcal{X}^0 = X$, and let $A \subseteq X$. We define $\langle A \rangle \subseteq \mathcal{X}$ to be the maximal sub-complex of \mathcal{X} that has A as its zero-skeleton.*

We are also interested in maps between spaces. A map $f : \mathcal{X} \rightarrow \mathcal{Y}$ is *cellular* if $f(\mathcal{X}^k) \subset \mathcal{Y}^k$ for all k . All maps between topological spaces should be understood to be continuous. For simplicial complexes, we say a map $f : \mathcal{X} \rightarrow \mathcal{Y}$ is *simplicial* if each simplex of \mathcal{X} maps linearly to a simplex of \mathcal{Y} (possibly of lower dimension) by mapping vertices to vertices and extending to higher dimensional simplices. Explicitly, once a map on vertices $f^0 : \mathcal{X}^0 \rightarrow \mathcal{Y}^0$ has been specified, for higher order simplices we have

$$f^k : (x_0, \dots, x_k) \mapsto (f^0(x_0), \dots, f^0(x_k)) \quad (2.9)$$

2.2.1 Homotopy

Homotopies are continuous deformations of maps. Algebraic topology typically studies *homotopy* invariants, and so understanding when maps are homotopic is important for our purposes.

Definition 2.2.2. Let $f, g : \mathcal{X} \rightarrow \mathcal{Y}$ be maps. We say that f and g are homotopic, denoted $f \simeq g$, if there exists a map (homotopy) $m : \mathcal{X} \times [0, 1] \rightarrow \mathcal{Y}$ so that $m(\cdot, 0) = f(\cdot)$, and $m(\cdot, 1) = g(\cdot)$.

Because m is just another map, we can naturally talk about *cellular* or *simplicial* homotopies.

Homotopies can be used to describe deformations of spaces as well as maps. We say spaces \mathcal{X} and \mathcal{Y} are homotopic, denoted $\mathcal{X} \simeq \mathcal{Y}$ if there exist maps $f : \mathcal{X} \rightarrow \mathcal{Y}$ and $g : \mathcal{Y} \rightarrow \mathcal{X}$ so that $g \circ f \simeq \iota_{\mathcal{X}}$, and $f \circ g \simeq \iota_{\mathcal{Y}}$, where ι denotes an identity map. If $\mathcal{X} \subset \mathcal{Y}$ and $\mathcal{X} \simeq \mathcal{Y}$, we say that \mathcal{X} is a retraction of \mathcal{Y} .

2.3 Chain Complexes

A *chain complex* is a sequence of vector spaces $\{C_k\}$ $k = 0, 1, \dots$ with *boundary* maps $\partial_k : C_k \rightarrow C_{k-1}$ with the property that $\partial_{k-1} \circ \partial_k = 0$. While in general, k need not start at 0, we will use this convention unless otherwise noted, which implies $\partial_0 = 0$

$$0 \longleftarrow C_0 \xleftarrow{\partial_1} C_1 \xleftarrow{\partial_2} \dots \xleftarrow{\partial_k} C_k \xleftarrow{\partial_{k+1}} \dots$$

We'll use C_* to refer to the set of vector spaces as well as the maps in the chain complex. When considering more than one chain complex, for clarity we may use ∂^C to denote the boundaries in C_* . Elements of C_k are referred to as k -chains, elements of $\ker \partial_k$ are referred to as cycles, and elements of $\text{img } \partial_{k+1}$ are referred to as boundaries.

A *chain map* between chain complexes C_* and D_* is a set of maps $\{F_k : C_k \rightarrow D_k\}$, $k = 0, 1, \dots$ with the property $F_k \circ \partial_{k+1}^C = \partial_{k+1}^D \circ F_{k+1}$, i.e. the following diagram commutes

$$\begin{array}{ccccccc} 0 & \longleftarrow & C_0 & \xleftarrow{\partial_1^C} & C_1 & \xleftarrow{\partial_2^C} & \dots \xleftarrow{\partial_k^C} C_k \xleftarrow{\partial_{k+1}^C} \dots \\ & & \downarrow F_0 & & \downarrow F_1 & & \downarrow F_k \\ 0 & \longleftarrow & D_0 & \xleftarrow{\partial_1^D} & D_1 & \xleftarrow{\partial_2^D} & \dots \xleftarrow{\partial_k^D} D_k \xleftarrow{\partial_{k+1}^D} \dots \end{array} \quad (2.10)$$

We'll use $F_* : C_* \rightarrow D_*$ to denote a chain map.

There is a functor from the category of cell (simplicial) complexes to the category of chain complexes over a field \mathbb{F} , meaning that every cell (simplicial) complex \mathcal{X}

has an associated chain complex $C_*(\mathcal{X})$, and every cellular map $f : \mathcal{X} \rightarrow \mathcal{Y}$ has an associated chain map $F_* : C_*(\mathcal{X}) \rightarrow C_*(\mathcal{Y})$. We will consider the complex of cellular chains, where $C_k(\mathcal{X})$ is constructed as the free vector space with basis given by the k -cells of \mathcal{X} - in other words elements of $C_k(\mathcal{X})$ are formal \mathbb{F} -linear combinations of k -cells in \mathcal{X} . We will not distinguish between a cell $s \in \mathcal{X}$ and the basis vector generated by s in $C_*(\mathcal{X})$ unless it is not clear from context.

Boundary matrices ∂_k are obtained by examining how faces are attached to the oriented boundary of cells. For general cell complexes this is obtained from the specified attaching maps, but for simplicial complexes the formula is combinatorial

$$\partial_k(x_0, \dots, x_k) = \sum_{i=0}^k (-1)^i (x_0, \dots, \hat{x}_i, \dots, x_k)$$

For example, an edge (x_0, x_1) has boundary $(x_1) - (x_0)$.

Given a map $f : \mathcal{X} \rightarrow \mathcal{Y}$, one can also compute chain maps $F_* : C_*(\mathcal{X}) \rightarrow C_*(\mathcal{Y})$. Again, for cell complexes this may take some work to specify, but for simplicial maps, one need only worry about how the vertices map (since higher order simplices are extended linearly). For instance, we have

$$F_0 : (x) \rightarrow (f(x))$$

and for higher order simplices

$$F_k : (x_0, \dots, x_k) \rightarrow (\text{sgn } P)(f(x_0), \dots, f(x_k))$$

Where P is the permutation that sorts $f(x_0), \dots, f(x_k)$. If multiple vertices of x map to the same vertex in Y , then the simplex $(f(x_0), \dots, f(x_k))$ is *degenerate*, so is not in the chain basis, and the coefficient for (x_0, \dots, x_k) in the chain map is zero. This follows from considering the simplicial complex \mathcal{Y} as a simplicial set [51].

2.3.1 Augmented Chain Complexes

We will consider several situations in which *augmented chain complexes* will bring some clarity. We will follow [81] for the notions we need. If C_* is a chain complex, an *augmentation* is a surjection $\epsilon : C_0 \rightarrow \mathbb{F}$, such that $\epsilon \circ \partial_1 = 0$. The augmented chain complex (C_*, ϵ) is the chain complex obtained by appending $C_{-1} = \mathbb{F}$ to C_* , to obtain a chain complex

$$0 \longleftarrow \mathbb{F} \xleftarrow{\epsilon} C_0 \xleftarrow{\partial_1} C_1 \xleftarrow{\partial_2} \dots \quad (2.11)$$

An augmentation-preserving chain map $F_* : C_* \rightarrow D_*$ is a chain map on the augmented complexes $(C_*, \epsilon), (D_*, \epsilon')$, meaning $\epsilon' \circ F_0 = \epsilon$.

One way to augment chain complexes obtained from cell complexes via the chain functor is by defining $\epsilon : x \mapsto 1$ for all x in the cell basis for C_0 . The map is certainly surjective, and because the boundary of any 1-cell (edge) can be expressed as $\partial_1 e = x_i - x_j$, we see that $\epsilon \circ \partial_1 = 0$. Furthermore, any cellular map preserves this augmentation, because $f : \mathcal{X} \rightarrow \mathcal{Y}$ must map zero-cells to zero-cells, so $\epsilon Fx = 1 = \epsilon x$ for all zero-cells x .

Definition 2.3.1. *Let C_* be a chain complex, with a specified basis B_* . The canonical augmentation $\epsilon : C_* \rightarrow \mathbb{F}$ is the augmentation $x \mapsto 1$ for all $x \in B_0$.*

As above, we'll typically use the canonical augmentation in the cell basis.

2.3.2 Chain Homotopy

Cellular homotopies lead naturally to a notion of chain homotopy:

Definition 2.3.2. *We say two chain maps $F_*, G_* : C_* \rightarrow D_*$ are homotopic if there exists a chain homotopy consisting of maps $M_k : C_k \rightarrow D_{k+1}$ so that*

$$\partial_{k+1}^D M_k + M_{k-1} \partial_k^C = F_k - G_k$$

As in Section 2.2.1, we will use \simeq to denote homotopy in the category of chain complexes, and can talk about homotopic chain maps as well as homotopic chain complexes.

2.4 Homology

Given a chain complex C_* , the *homology* vector space in dimension k is defined as the quotient vector space $H_k(C_*) = \ker \partial_k / \text{img } \partial_{k+1}$. Because $\partial_k \circ \partial_{k+1} = 0$, we know that $\text{img } \partial_{k+1} \subseteq \ker \partial_k$, so the quotient vector space is defined.

A chain map $F_* : C_* \rightarrow D_*$ produces an induced map in homology $\tilde{F}_k : H_k(C_*) \rightarrow H_k(D_*)$ [57]. We'll make an argument using representatives. Let $x_k \in [x_k] \in H_k(C_*)$, and $x'_k \in \text{img } \partial_{k+1}^C$, meaning there is some $y \in C_{k+1}$ so that $x'_k = \partial_{k+1}^C(y)$, and that $[x_k + x'_k] = [x_k]$. Then

$$\begin{aligned} F_k(x_k + x'_k) &= F_k x_k + F_k \partial_{k+1}^C y \\ &= F_k x_k + \partial_{k+1}^D F_{k+1} y \end{aligned}$$

First, note that because x_k is a homology representative, $x_k \in \ker \partial_k^C$, and from the commutation property of chain maps Equation (2.10), $F_k x_k$ must be in $\ker \partial_k^D$, so is also a representative for some homology class in $H_k(D_k)$. Next, since $\partial_k^D \circ \partial_{k+1}^D = 0$, $\partial_{k+1}^D F_{k+1} y \in \ker \partial_k^D$, the homology class of $F_k x_k$ does not depend on the representative of $[x_k]$ chosen. Thus

$$\tilde{F}_k[x_k] = [F_k x_k]. \quad (2.12)$$

It follows that the identity map on chains, denoted I , induces an identity map on homology, \tilde{I} .

When homology is computed on a chain complex associated with a topological space \mathcal{X} , certain topological information can be extracted from the vector spaces $H_k(\mathcal{X}) = H_k(C_*(\mathcal{X}))$. The dimension of $H_0(\mathcal{X})$ is equal to the number of connected components in \mathcal{X} , the dimension of $H_1(\mathcal{X})$ counts the number of non-contractible

loops, and $H_k(\mathcal{X})$ generally counts the number of k -dimensional voids in \mathcal{X} . Representatives of vectors in H_k are also known as *generators*, and consist of linear combinations of cells in a subcomplex of \mathcal{X} , and since H_k is a quotient vector space, representatives are not unique. Because there are generally many choices of basis for $H_k(\mathcal{X})$, as well as many choices of representative, representatives need not be particularly interpretable, and may generally appear to be quite complex.

Induced maps on homology are often more revealing, and information about the kernel and image of a map can be used to understand what features in \mathcal{X} are collapsed by a map to another space \mathcal{Y} . The actual matrix representation is dependent on the bases chosen for homology.

2.4.1 Homotopy Invariance

A standard result in homological algebra is that if chain maps $F_*, G_* : C_* \rightarrow D_*$ are homotopic, then the induced maps on homology $\tilde{F}_*, \tilde{G}_* : H_*(C_*) \rightarrow H_*(D_*)$ are isomorphic [57]. When comparing homology in a restricted set of dimensions, we would like to construct something less than a full homotopy, so we will give a variant of this result

Lemma 2.4.1. *Let $F_*, G_* : C_* \rightarrow D_*$ be chain maps, and suppose $M_* : C_* \rightarrow D_{*+1}$ satisfies*

$$\partial_{k+1}^D M_k + M_{k-1} \partial_k^C = F_k - G_k \quad (2.13)$$

for some k . Then $\tilde{F}_k = \tilde{G}_k$

Proof. Let $[x] \in H_k(C_*)$. We want to show that $\tilde{F}_k[x] = \tilde{G}_k[x]$. It will suffice to show that $F_k x - G_k x \in [0]$, by showing $F_k x - G_k x \in \text{img } \partial_{k+1}^D$. First, note that because $[x]$ is a homology class, that $x \in \ker \partial_k^C$, so $M_{k-1} \partial_k^C x = 0$. Now, by assumption in Equation (2.13), we have

$$F_k x - G_k x = \partial_{k+1}^D M_k x$$

so $F_k x - G_k x \in \text{img } \partial_{k+1}^D$. □

Corollary 2.4.2. *Let $F_*, G_* : C_* \rightarrow D_*$ be chain maps, and $M_* : C_* \rightarrow D_{*+1}$ be a chain homotopy. Then $\tilde{F}_* = \tilde{G}_*$.*

Proof. This follows from Lemma 2.4.1, because Equation (2.13) is satisfied for all k . Alternatively, see [57]. \square

Another standard, but important result is

Corollary 2.4.3. *If $\mathcal{X} \simeq \mathcal{Y}$, then $H_*(\mathcal{X}) \cong H_*(\mathcal{Y})$.*

Proof. We have maps $f : \mathcal{X} \rightarrow \mathcal{Y}$ and $g : \mathcal{Y} \rightarrow \mathcal{X}$ that satisfy $g \circ f \simeq \iota_{\mathcal{X}}$, and $f \circ g \simeq \iota_{\mathcal{Y}}$. Passing through the homology functor, we have $\tilde{F}_* \tilde{G}_* = \tilde{I}_*$ (the identity map), and similarly $\tilde{G}_* \tilde{F}_* = \tilde{I}_*$. \square

2.4.2 Acyclic Complexes

First, we define homology of an augmented cell complex to be

$$\tilde{H}_k(C_*) = H_k((C_*, \epsilon)) \quad (2.14)$$

Note that regardless of the augmentation ϵ

$$H_k(C_*) = \begin{cases} \tilde{H}_0(C_*) \oplus \mathbb{F}, & k = 0 \\ \tilde{H}_k(C_*), & k > 0 \end{cases} \quad (2.15)$$

Definition 2.4.4. C_* is said to be acyclic if $\tilde{H}_k(C_*) = 0$ for all $k \geq 0$. We say a space \mathcal{X} is acyclic if $C_*(\mathcal{X})$ is acyclic.

In Chapter 4, we'll see that acyclic sub-complexes have nice properties that will allow us to extend chain maps. The following result is standard for a variety of different homology theories (referring to homology obtained via different chain functors from topological categories), to the extent that it was encoded as the “dimension axiom” in Eilenberg and Steenrod’s axiomatic treatment of homology [49].

Lemma 2.4.5. *Let $*$ denote the space with a single point. $*$ is acyclic.*

Proof. We'll show this for the cellular homology functor. We note that $*$ has only a single zero-cell. Thus, any augmented chain complex $(C_*(*), \epsilon)$ must take the form

$$0 \longleftarrow \mathbb{F} \xleftarrow{\cong} \mathbb{F} \longleftarrow 0 \dots \quad (2.16)$$

Computing the quotient vector spaces for the augmented chain complex confirms that $\tilde{H}_k(*) = 0$ for all $k \geq 0$, and so $*$ is acyclic. \square

Corollary 2.4.6. *Suppose $\mathcal{X} \simeq *$, meaning \mathcal{X} is contractible. Then \mathcal{X} is acyclic.*

Proof. This follows immediately from the homotopy invariance of homology. \square

2.5 Computing Homology

Given matrices ∂_k and ∂_{k+1} , with the property $\partial_k \partial_{k+1} = 0$. We seek to compute the quotient space $\ker \partial_k / \text{img } \partial_{k+1}$. This will require finding a basis for $\ker \partial_k$ (the k -cycles), and finding a sub-basis which is not in $\text{img } \partial_{k+1}$.

Definition 2.5.1. *A homology revealing basis for C_k is a pair (B_k, \mathcal{I}_k) , where B_k is a basis for C_k , and \mathcal{I}_k is an index set such that $\{b_i \in B_k\}_{i \in \mathcal{I}_k} \subseteq B_k$ generates a basis for $H_k(C_*)$. Explicitly, a basis for H_k is*

$$\{[b_i] \mid b_i \in B_k, i \in \mathcal{I}_k\}$$

In practice, a homology revealing basis for C_k is computed by first finding a basis for cycles $\ker \partial_k^C$, and then finding a sub-basis for cycles which are not boundaries $\text{img } \partial_{k+1}^C$. A homology-revealing basis is certainly not unique - there may be other choices of \mathcal{I}_k that would also give representatives that generate a basis for homology, or we can always modify the representatives by adding arbitrary boundaries from $\text{img } \partial_{k+1}$. Once we have chosen a basis B_k and a set \mathcal{I}_k , we will say a homology representative $x \in [x]$ is the preferred representative of $[x]$ if x is written as linear combination of cycles exclusively in the set \mathcal{I}_k .

Proposition 2.5.2. *Given a homology-revealing basis (B_k, \mathcal{I}_k) for C_k , every homology class $[x] \in H_k(C_*)$ has a unique preferred representative.*

Proof. Existence comes by definition, since (B_k, \mathcal{I}_k) generates a basis.

Now, suppose that a homology class has two preferred representatives $\sum_{i \in \mathcal{I}_k} \alpha_i b_i$ and $\sum_{i \in \mathcal{I}_k} \beta_i b_i$. Then using Equation (2.2), we have

$$\begin{aligned} \left[\sum_{i \in \mathcal{I}_k} \alpha_i b_i \right] &= \left[\sum_{i \in \mathcal{I}_k} \beta_i b_i \right] \\ \sum_{i \in \mathcal{I}_k} \alpha_i [b_i] &= \sum_{i \in \mathcal{I}_k} \beta_i [b_i] \end{aligned}$$

because $\{[b_i]\}$ is the generated basis for homology, we must have $\alpha_i = \beta_i$ for all $i \in \mathcal{I}_k$. \square

The advantage of working with homology revealing bases explicitly is that we can reason about vectors in $H_*(C_*)$ in the generated basis using preferred representatives in the chain complex.

2.5.1 The Reduction Algorithm

In this section we'll review a common approach for finding a homology-revealing basis known as the reduction algorithm [106], which has also has useful properties for computing persistent homology of filtrations. It involves putting the boundary matrices ∂_k in a reduced form $\partial_k U_k = R_k$, or written as a factorization $\partial_k = R_k U_k^{-1}$, and extracting homology information from U_k and R_k . We define the *pivot* of a column v to be the largest index i so that $v[i]$ that has a non-zero value.

$$\text{piv } v = \max\{i \mid v[i] \neq 0\}$$

If there are non non-zero values in v , we say $\text{piv } v = 0$.

Algorithm 1 [106] Reduction Algorithm with formation of basis U .

```

1: Input:  $m \times n$  matrix  $A$ 
2: Result: Factorization  $AU = R$ .
3:  $U = I_n$ 
4:  $R = A$ 
5: for  $j = 1, \dots, n$  do
6:   while  $\text{piv}(R[j]) > 0$  and  $j' < j$  exists so that  $i = \text{piv}(R[j]) = \text{piv}(R[j'])$  do
7:      $\alpha = R[i, j]/R[i, j']$ 
8:     Update  $R[j] = R[j] - \alpha R[j']$ 
9:     Update  $U[j] = U[j] - \alpha U[j']$ 
10:  end while
11: end for
12: return  $U, R$ 

```

In practice, pivots can be remembered using a data structure that permits fast look-up, and A can be modified in-place to form R . We will call the matrix R the *reduced matrix*. The rest of this section parallels the analysis found in [106], but we provide some additional care to explain why U is a homology-revealing basis in the context of applying the reduction algorithm to chain boundary matrices.

Proposition 2.5.3. *Algorithm 1 produces a valid factorization $AU = R$, where U is upper-triangular, and no two columns of R share the same non-zero pivot.*

Proof. At the beginning of the algorithm, we have the trivial identity $AU_0 = R_0$, where $U_0 = I$. We will count iterations over the columns using j , and iterations of the while-loop using ℓ .

Suppose that at step $\ell - 1$, we have maintained the invariant $AU_{\ell-1} = R_{\ell-1}$. Each iteration of the while loop performs a column operation that is equivalent to multiplying the matrices $R_{\ell-1}$ and $U_{\ell-1}$ on the right by a matrix $U'_\ell = I - e_j \alpha e_j$. Then we have $AU_{\ell-1}U'_\ell = R_{\ell-1}U'_\ell$. Writing $R_\ell = R_{\ell-1}U'_\ell$ and $U_\ell = U_{\ell-1}U'_\ell$, we see that we have $AU_\ell = R_\ell$. Note that these updates are done in-place in the matrices R and U . By induction, the invariant $AU_\ell = R_\ell$ is maintained throughout the algorithm.

Note that $U_0 = I$ is upper-triangular. Suppose that $U_{\ell-1}$ is upper-triangular. Note that $U'_\ell = I - e_{j'}\alpha e_j$ is upper-triangular as well, and because the class of upper triangular matrices is closed under multiplication, $U_\ell = U_{\ell-1}U'_\ell$ will also be upper triangular. By induction, U_ℓ is upper-triangular throughout the algorithm.

Now, we will show that we maintain the invariant that columns $j' \leq j$ in R have unique non-zero pivots. We consider an arbitrary column j in R . Note that the j -th column of R is only modified during the j -th iteration of the for-loop. Note that at each iteration of the while loop, if j shares a pivot with column $j' < j$, then that pivot of j is eliminated, and the pivot of column j decreases (because adding $R[j']$ can not introduce non-zeros after the pivot). This process continues until there are no columns $j' < j$ that share a pivot with j , or the pivot becomes zero. Thus, j shares no pivots with columns $j' < j$. Because this holds for all columns j , the final R can not have two columns with the same pivot (otherwise this principle would be violated for the column with larger index).

When the loops terminate, we take U to be the final state U_ℓ , and R to be the final state R_ℓ . Thus, $AU = R$, U is upper-triangular, and R has unique non-zero pivots.

□

Proposition 2.5.4. *The worst case run-time of Algorithm 1 is $O(m^2n)$.*

Proof. For a single column, the updates take worst case $O(m)$ time each (to loop over every entry), and we can iterate through at most m pivots, for at most $O(m^2)$ time for each column. For all n columns, the total time is $O(m^2n)$. □

Note that due to sparsity of the boundary matrices, the worst case runtime is generally pessimistic.

As a consequence of Proposition 2.5.3, the set of non-zero columns in R is linearly independent because they have distinct pivots. Finally, note that $\text{img } R = \text{img } A$ because U is just a change of basis in the columns.

Now, we'll explain how to extract information about homology assuming we have reduced boundary matrices for each dimension k as $\partial_k U_k = R_k$. First, note that we can extract a basis for cycles from U_k by examining which columns of R_k are

zero. Specifically, if $R_k e_j = 0$, then $R_k e_j = \partial_k U_k e_j = \partial_k U_k[j] = 0$, so $U_k[j] \in \ker \partial_k$. Because U_k is unit upper-triangular its columns are linearly independent, so the collection of cycles found in this way forms a basis for $\ker \partial_k$.

Now that we have a basis for cycles, we want to find a basis for homology by finding cycles that are not in $\text{img } \partial_{k+1}$.

Lemma 2.5.5. *Consider column $U_k[i]$. If there exists some j , such that $\text{piv } R_{k+1}[j] = i$, then $U_k[i]$ is a cycle.*

Proof. We know $\partial_k R_{k+1}[j] = \partial_k \partial_{k+1} U_{k+1}[j] = 0$. Thus,

$$\partial_k \left[R_{k+1}[i, j] e_i + \sum_{\ell < i} R_{k+1}[\ell, j] e_\ell \right] = 0.$$

This means that

$$\partial_k e_i = \sum_{\ell < i} (R_{k+1}[\ell, j] / R_{k+1}[i, j]) \partial_k e_\ell,$$

so $\partial_k e_i = \partial_k[i]$ can be written as a linear combination of columns $\partial_k e_\ell = \partial_k[\ell]$, $\ell < i$. This means that $R_k[i] = 0$, so $U_k[i]$ is a cycle. \square

Proposition 2.5.6. *The set of cycles in U_k whose column index do not appear as a pivot in R_{k+1} form a basis for $H_k = \ker \partial_k / \text{img } \partial_{k+1}$.*

Proof. We'll first show that if $U_k[i]$ is a cycle, and i does not appear as a pivot in R_{k+1} , that $U_k[i] \notin \text{img } R_{k+1} = \text{img } \partial_{k+1}$. Suppose $U_k[i] \in \text{img } R_{k+1}$. Because $U_k[i, i] = 1$, there must be some linear combination of columns of R that produces a 1 in the i -th entry.

$$1 = \sum_j \alpha_j R_{k+1}[i, j]$$

Let j' be the index column with largest pivot and $\alpha_{j'} \neq 0$. We know that $\text{piv } R_{k+1}[j'] > i$ because i does not appear as a pivot in R_{k+1} , and because $U_k[i, i] = 1$, some column with pivot greater than i must be used. Next, note that because only one column in R_{k+1} can have pivot $i' = \text{piv } R_{k+1}[j']$, we must have $U_k[i', i] = \alpha_{j'} R_{k+1}[i', j']$. However, since $i' > i$ and U_k is upper-triangular, we must have $U_k[i', i] = 0$, introducing a contradiction. Thus, $U_k[i]$ is not in $\text{img } R_{k+1}$.

Next, note that the set of such cycles is linearly independent since they are distinct columns in an upper-triangular matrix. Furthermore, by Lemma Lemma 2.5.5, each non-zero column in R_{k+1} is matched with a cycle in U_k .

Finally, we can count dimensions. Note that $\dim H_k = \dim \ker \partial_k - \dim \text{img } \partial_{k+1}$, and $\text{img } \partial_{k+1} = \text{img } R_{k+1}$. The number of cycles that do not appear as pivots in R_{k+1} is exactly $\dim \ker \partial_k - \dim \text{img } R_{k+1}$, and since these cycles are linearly independent and represent non-trivial homology classes, they must form a basis for H_k . \square

What we have shown is that the columns of U_k form a homology-revealing basis for the chain complex, where the subset of the basis that gives the homology basis is determined by looking at columns of R_k that have zero pivot, and which do not appear as a pivot in R_{k+1} . This produces an algorithm to get the indices \mathcal{I}_k for U_k that give the homology basis.

Algorithm 2 Extraction of homology-revealing bases from reduced boundary matrices.

```

1: Input: Matrices  $U_k, R_k$   $k = 0, 1, \dots$  from reduction algorithm.
2: Result: Index sets  $\mathcal{I}_k$  for homology bases.
3: for  $k = 0, 1, \dots$  do
4:    $\mathcal{I}_k \leftarrow \{\}$ 
5:    $n_k \leftarrow \dim C_k$ 
6:   for  $j = 1, \dots, n_k$  do
7:     if  $\text{piv}(R_k[j]) = 0$  and  $j$  is not a pivot in  $R_{k+1}$  then
8:        $\mathcal{I}_k \leftarrow \mathcal{I}_k \cup \{j\}$ 
9:     end if
10:  end for
11: end for
12: return  $\{\mathcal{I}_k\}$ 

```

Proposition 2.5.7. *Algorithm 2 takes $O(n_k)$ time for dimension k , where n_k is the number of columns in R_k .*

Proof. We assume that it takes $O(1)$ time to find the pivot of a column in R_k (e.g. this can be stored in an array while performing the reduction algorithm, or simply

computed if R_k is in a sparse format such as CSC or list of lists). We also assume that we can check if a column of R_{k+1} has a given pivot in $O(1)$ time, for instance this can be stored using an array or dictionary when running the reduction algorithm.

With these assumptions, it takes $O(1)$ time to check whether each column of R_k will be used to represent homology, for a total time of $O(n_k)$ in dimension k . \square

2.5.2 Induced Maps

A final ingredient we need is the ability to compute induced maps on homology. Assume we have a chain map $F_* : C_* \rightarrow D_*$, and that we've found homology-revealing bases represented by (U_*^C, \mathcal{I}_*^C) and (U_*^D, \mathcal{I}_*^D) for C_* and D_* . Note that we wish to use preferred representatives to obtain the induced map $\tilde{F}_k : H_k(C_*) \rightarrow H_k(D_*)$ in terms of the generated bases by using linearity of the quotient Equation (2.2)

$$\tilde{F}_k[x] = \left[\sum_{i \in \mathcal{I}_k^D} \alpha u_i \right] = \sum_{i \in \mathcal{I}_k^D} \alpha_i [u_i]. \quad (2.17)$$

We'll consider how a homology basis vector $[x] \in H_k(C_*)$, represented by a preferred representative $x = U_k^C[i], i \in \mathcal{I}_k^C$ passes through the chain map F_k . The image in the basis U_k^D is given by Equation (2.1) as $y = (U_k^D)^{-1} F_k x$. We know from Equation (2.12) that y is a representative for the induced map on homology of $\tilde{F}_k[x]$, but it may not be the preferred representative determined for the homology revealing basis (U_k^D, \mathcal{I}_k^D) . Because $H_k(D) = \ker \partial_k^D / \text{img } \partial_{k+1}^D$, we can arbitrarily add boundaries (elements of $\text{img } \partial_{k+1}^D$) to y without changing the homology class. Furthermore, by Proposition 2.5.2, we know there exists a unique linear combination $\sum_{i \in \mathcal{I}_k^D} \alpha_i e_i$ that is in the same homology class as y in $H_k(D)$, which can be obtained by adding an element of $\text{img } \partial_{k+1}^D$. If we have used the reduction algorithm to obtain $\partial_k^D = R_k^D (U_k^D)^{-1}$, we can write ∂_k^D in the bases U_{k-1}^D, U_k^D as $(U_{k-1}^D)^{-1} \partial_k U_k^D = (U_{k-1}^D)^{-1} R_k$. Combining these observations, the linear system

$$(U_k^D)^{-1} R_{k+1}^D [\bar{\mathcal{I}}_k^D, :] z = y [\bar{\mathcal{I}}_k^D] \quad (2.18)$$

is consistent and can be solved for z , and then $y_2 = y - (U_k^D)^{-1}R_{k+1}^D z$ will be in the same homology class as y , but will only have non-zero coefficients in the preferred basis \mathcal{I}_k^D . We then can obtain the induced map on homology by reading off the coefficients $y_2[\mathcal{I}_k^D]$.

When we use the reduction algorithm to obtain R_k^D and U_k^D , we can write down an explicit algorithm to obtain y_2 via the solution of Equation (2.18). First note that R_{k+1}^D can be made upper triangular via a column permutation, and that the application of the upper triangular $(U_k^D)^{-1}$ will not affect the pivots. Thus, we can perform a variant of backward-substitution for upper-triangular matrices.

Algorithm 3 Computation of induced map on homology.

- 1: **Input:** Homology representative $x = U_k^C[i]$ in $H_k(C_*)$, U_k^D, R_{k+1}^D , from reduction algorithm applied to ∂_*^D , with index set \mathcal{I}_k^D . Chain map F_k in original basis.
 - 2: **Result:** Induced map on homology, $\tilde{F}_k[x]$
 - 3: $y \leftarrow (U_k^D)^{-1}F_k x$
 - 4: $n \leftarrow \dim D_k$
 - 5: $\hat{\partial}_{k+1}^D \leftarrow (U_k^D)^{-1}R_{k+1}$
 - 6: **for** $j = n, n - 1, \dots, 1$ **do**
 - 7: **if** $y[j] \neq 0$ and j is a pivot of column i of $\hat{\partial}_{k+1}^D$ **then**
 - 8: $\alpha \leftarrow y[j]/\hat{\partial}_{k+1}^D[j, i]$
 - 9: $y \leftarrow y - \alpha\hat{\partial}_{k+1}^D[i]$
 - 10: **end if**
 - 11: **end for**
 - 12: **return** $y[\mathcal{I}_k^D]$
-

Proposition 2.5.8. *The homology class of y in Algorithm 3 is invariant.*

Proof. Only boundaries are added to y , as columns of $\hat{\partial}_{k+1}^D$, so the homology class is invariant. □

Proposition 2.5.9. *In Algorithm 3, at the end of the for-loop, y will be the preferred representative for its homology class with respect to the basis (U_k^D, \mathcal{I}_k^D) .*

Proof. This means that y will only have non-zeros in indices that are in the index set \mathcal{I}_k^D .

First, note that \mathcal{I}_k^D was constructed to be the indices of cycles in D_k that did not appear in pivots of R_k , and $\hat{\partial}_k^D$ has the same pivots as R_k . Second, note that because x is a cycle, and F_k is a chain map, y must be a linear combination of cycles in the basis given by U_k^D . Finally, the for-loop removes non-zero coefficients for any cycle that has a non-zero pivot in $\hat{\partial}_k^D$. Thus, y can only have non-zero coefficients for the cycles indexed by \mathcal{I}_k^D . \square

As a result, the coefficients returned by Algorithm 3 will be the coefficients for the induced map on homology as seen in Equation (2.17). We can construct a full matrix representing \tilde{F}_k in the bases generated by the homology revealing bases by applying this procedure for every preferred basis element in U_k^C given by the index set \mathcal{I}_k^C .

Proposition 2.5.10. *Let n_k^C and n_k^D denote the dimension of C_k , D_k respectively, and β_k^C , β_k^D denote the respective homology dimensions. Then Algorithm 3 runs in $O((n_k^D)^2 n_{k+1}^D + (n_k^C n_k^D + (n_k^D)^2) \beta_k^C)$ time.*

Proof. We'll assume that the matrix $\hat{\partial}_{k+1}^D$ is formed once for all representatives x , for a cost of $O((n_k^D)^2 n_{k+1}^D)$.

The application of the map F_k takes $O(n_k^C n_k^D)$ time, and the application of $(U_k^D)^{-1}$ takes $O((n_k^D)^2)$ time. The for-loop makes at most n_k^D iterations, and each update takes at most n_k^D time for an additional cost of $O((n_k^D)^2)$ time per representative. We then multiply the cost per representative by β_k^C representatives. \square

Example 2.5.11. *Let $X = \{(x_0), (x_1), (x_0, x_1)\}$ and $Y = \{(y_0), (y_1), (y_0, y_1)\}$ be simplicial complexes, and let $f : X \rightarrow Y$ be the simplicial map that sends $(x_0) \mapsto (y_1)$, $(x_1) \mapsto (y_0)$. The chain boundaries and chain map can be expressed in the cell basis as*

$$\partial_0^X = \partial_0^Y = \begin{bmatrix} 0 & 0 \end{bmatrix}, \quad \partial_1^X = \partial_1^Y = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \quad F_0 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad F_1 = \begin{bmatrix} 1 \end{bmatrix}$$

The reduction algorithm, Algorithm 1, will find the boundaries are already in reduced form, meaning $U_0^X = U_0^Y = I$, and we find $H_0(X) = H_0(Y) = \mathbb{F}$, with x_0 and y_0 selected as homology representatives.

Note that

$$F_0 x_0 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = y_1$$

is not a preferred representative for homology of $H_0(Y)$, so we need to use Algorithm 3 to find the induced map. Note that $\hat{\partial}_1^Y = \partial_1^Y$, which has a non-zero pivot in index 2. Thus, the algorithm will eliminate the non-zero entry for y_1 , and introduce a non-zero entry for y_0 :

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} - \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = y_0$$

which is a preferred representative of the homology class. Finally, we just read off the coefficient, to obtain $\tilde{F}_0 = [1]$.

In summary, the calculation confirms that the map f sends the single connected component of X to the single connected component of Y .

2.6 Persistent Homology of Filtrations

We will first consider persistent homology of filtrations before moving to the more general setting. This special case of persistent homology has been the focus of a lot of attention both for applications and algorithmic improvements in topological data analysis and we will focus on the special structure that aids in computation.

Recall that a filtration is a nested sequence of spaces

$$\mathcal{X}_0 \subseteq \mathcal{X}_1 \subseteq \cdots \subseteq \mathcal{X}_n \tag{2.19}$$

The persistent homology of the filtration studies how homology changes through the sequence of spaces.

$$H_k(\mathcal{X}_0) \rightarrow H_k(\mathcal{X}_1) \rightarrow \cdots \rightarrow H_k(\mathcal{X}_n) \tag{2.20}$$

Filtrations are not constrained to have integer parameters, for example the Rips and Čech filtrations both take real valued parameters. Our focus is on filtrations of finite cell complexes, in which case the filtration can be re-parameterized to take integer parameters with a distinct value for each real parameter that adds cells to the complex. For simplicity, we will consider filtrations where every cell is added one at a time. In general, multiple cells may appear at the same filtration value, in which case we can choose any arbitrary ordering that only ensures that a cell's boundary is present before a cell is added [93].

We can analyze what occurs at each step in this discrete filtration. The following result is standard in the literature [45].

Proposition 2.6.1. *Adding a k -dimensional cell x to a cell complex \mathcal{X} either creates homology in dimension k or destroys homology in dimension $k - 1$.*

Proof. The addition of a k -dimensional cell x appends a single column to ∂_k . Note that the only two subspaces that this can affect are $\ker \partial_k \subseteq C_k$ and $\text{img } \partial_k \subseteq C_{k-1}$. There are two possibilities for reduction:

1. $\partial_k x$ is already in $\text{img } \partial_k$, meaning that $\partial_k x = \partial_k y$ for some other chain $y \in C_k$. In this case, the chain $x - y$ will have boundary 0, so the dimension of $\ker \partial_k$ is incremented by 1. However, the dimension of $\text{img } \partial_{k+1}$ does not change, so the dimension of $H_k = \dim \ker \partial_k - \dim \text{img } \partial_{k+1}$ is also extended by 1. Because $\text{img } \partial_k$ does not change, H_{k-1} does not change.
2. $\partial_k x$ is not already in $\text{img } \partial_k$. In this case, the dimension of $\text{img } \partial_k$ increases by 1, but because ∂_{k-1} is the same, so is $\ker \partial_{k-1}$. Thus the dimension of H_{k-1} will be reduced by 1. Because x is not in $\ker \partial_k$, H_k will be unaffected. However, a new pivot will be added to I_k , so H_{k-1} will be reduced by one dimension.

□

Definition 2.6.2. *The birth of a homology class is the index in the filtration that the homology class first appears. The death of a homology class is the index in the filtration that the class disappears. Each homology class in the filtration has an associated birth-death pair (b, d) , where $d = \infty$ if the class is present at the end of the filtration.*

Definition 2.6.3. *The persistence barcode of a filtered complex \mathcal{X}_* is a multiset of birth-death pairs (b_i, d_i) , one for each homology class that appears at some point in the filtration.*

2.7 Reduction Revisited

The reduction algorithm, Algorithm 1, can be used un-modified to compute persistent homology. All that is necessary is a little more nuance in its interpretation. Recall that the chain complex $C_k(\mathcal{X})$ has as a basis all k -cells in \mathcal{X} . Henceforth, we will assume that this basis is ordered by the order of appearance of cells in the filtration, and that the boundary matrices ∂_k are given in this basis (i.e. the first column corresponds to the first k cell, and the first row corresponds to the first $k - 1$ cell in the filtration).

The key observation is that because the reduction of column j only looks to the left for pivots, the reduction for the first j columns for ∂_k^j and $\partial_k^{j_2}$, $j < j_2$ will proceed in exactly the same manner. This means that instead of computing a homology revealing basis for each filtration parameter and examining the maps on homology induced by inclusion we can simply run the reduction algorithm once for the final cell complex \mathcal{X}_n , and add an additional layer of analysis to the discussion in Section 2.5.1.

Suppose we have performed the reduction algorithm $\partial_k U_k = R_k$, $k = 0, 1, \dots$ for \mathcal{X}_n . Recall that if a column j of R_k is zero, then the j th column of U_k gives a cycle in $C_k(\mathcal{X}_n)$, and if j appears as a pivot in R_{k+1} , then that cycle is a boundary, so the index set \mathcal{I}_k for the homology revealing basis is found by identifying the zero columns of R_k that do not appear as pivot indices in R_{k+1} , following Algorithm 2. However, if the reduction had been performed when column j was first added, no corresponding pivot column in R_{k+1} would yet be added, so the column would generate homology (causing a birth at the filtration parameter for column j). Finally, at some filtration parameter a column would be added to ∂_{k+1} producing a pivot in row j , killing that homology class. If no such column ever appears in ∂_{k+1} , the homology class will be present in the final complex \mathcal{X}_n , and the corresponding death will be at ∞ .

The reason why the induced maps never need to be explicitly computed can be seen in the following proposition

Proposition 2.7.1. *Let (U_k^i, \mathcal{I}_k^i) and (U_k^j, \mathcal{I}_k^j) be homology revealing bases for a filtration $X_i \subseteq X_j$ computed using the ordered cell basis and the reduction algorithm. Then the induced map on homology $H_k(X_i) \rightarrow H_k(X_j)$ from inclusion either (a) sends a basis element $[x] \in H_k(X_i)$ to $[0]$, or (b) sends a basis element $[x] \in H_k(X_i)$ to exactly one other basis element $[x'] \in H_k(X_j)$. Furthermore, in case (b), the chain map sends the preferred representative of $[x]$ to the preferred representative of $[x']$.*

Proof. Let x be a preferred representative for $[x] \in H_k(\mathcal{X}_i)$. Note that because the reduction algorithm will produce the same result for the first n_i columns of ∂_k , the inclusion map $C_k(\mathcal{X}_i) \rightarrow C_k(\mathcal{X}_j)$ has the form

$$F_k = \begin{bmatrix} I \\ 0 \end{bmatrix}$$

where I denotes an identity and $(U_k^j)^{-1}F_kU_k^i = F_k$ takes the exact same form. We know that because x is a preferred representative, that it is a column of U_k^i , and the block identity in the chain map will map x to the corresponding column in U_k^j . Either that column is in \mathcal{I}_k^j , in which case it is a preferred representative for a basis element of $H_k(\mathcal{X}_j)$, or there must be a pivot with the corresponding index in R_{k+1}^j indicating that the column is a boundary, so is in $[0]$. \square

Corollary 2.7.2. *The induced map on homology $\tilde{F}_k : H_k(X_i) \rightarrow H_k(X_j)$ is in E_U form.*

Proof. This follows from using the ordering inherited on the basis for homology from the ordering on the basis for the cell complex. \square

2.8 From Inclusion to General Maps

2.8.1 Cylinders to Filtrations

Much of this section has focused on the use of persistent homology in the specific case where all maps between spaces are inclusions. As we saw in the introduction, the more general case that uses arbitrary maps is well-defined, even if it isn't popularly used. We'll briefly cover how one may re-purpose existing tools to compute the more general case before we see how quiver algorithms solve it more gracefully.

The key construction we'll use is based on the mapping cylinder between two spaces [57]. Given a map $f : \mathcal{X} \rightarrow \mathcal{Y}$, the *mapping cylinder* is defined as the quotient space

$$\text{Cyl } f = \mathcal{X} \times [0, 1] \sqcup \mathcal{Y} / [(x, 1) \sim f(x)]$$

where \sqcup is the disjoint union. That is, $\text{Cyl } f$ is obtained by taking the cylinder $\mathcal{X} \times [0, 1]$ and attaching to a copy of \mathcal{Y} by gluing $\mathcal{X} \times \{1\}$ to the image of f . The space is homotopic to \mathcal{Y} (there is a retraction of \mathcal{X} onto the image of f along the interval) so the space itself will have the same homology as \mathcal{Y} . However, if we add a filtration where $(\mathcal{X}, 0)$ appears at time 0, and the rest of the space appears at time 1, we see that homology in \mathcal{X} that is killed by f dies at time 1 in the filtration, and homology in \mathcal{X} in the image of f survives, while homology in \mathcal{Y} that is in the cokernel of f is born at time 1.

There are several potential difficulties that may be encountered when attempting to use mapping cylinders in existing TDA packages. In the case where the map $f : \mathcal{X} \rightarrow \mathcal{Y}$ is simplicial, the cylinder has the structure of a simplicial set, which has desirable combinatorial properties. However, simplicial sets are not generally supported in TDA packages, and so one must fall back to either using general cell complexes, or triangulating the space to form a simplicial complex. In the more general case, the mapping cylinder of a cellular map is a cell complex. This can be encoded explicitly if desired and passed to any persistent homology package that can take cell complexes as input.

General cell complexes require one to know the boundary of each cell. We can use the product rule $\partial(a \times (0, 1)) = -\partial a \times (0, 1) + a \times \partial(0, 1)$ to compute the boundary of cells in the cylinder. Given this information, one can either store the space as a general cell complex, or just form the boundary directly to pass to a solver. For a mapping cylinder $\text{Cyl } f : \mathcal{X} \rightarrow \mathcal{Y}$, the boundary in dimension k is

$$\partial_k^{\text{Cyl } f} = \begin{bmatrix} \partial_k^{\mathcal{X}} & -I \\ & \partial_k^{\mathcal{Y}} & F_{k-1} \\ & & -\partial_{k-1}^{\mathcal{X}} \end{bmatrix} \quad (2.21)$$

One can check explicitly that $\partial^2 = 0$. The utility of the mapping cylinder is that the map on homology induced by inclusion $\mathcal{X} \hookrightarrow \text{Cyl } f$ is equivalent to the map induced by $f : \mathcal{X} \rightarrow \mathcal{Y}$. Details will be found in Section 2.8.2. Thus, if we consider a filtration on $\text{Cyl } f$, at which $\mathcal{X} \times \{0\}$ appears at parameter t_0 , and the rest of the cylinder appears at parameter $t_1 > t_0$, in persistent homology the number of bars that survive from t_0 to t_1 will be equal to the rank of $\tilde{F}_k : H_k(\mathcal{X}) \rightarrow H_k(\mathcal{Y})$.

2.8.2 The Mapping Cylinder

The mapping cylinder can be viewed as the pushforward of the map $f : \mathcal{X} \rightarrow \mathcal{Y}$ and the inclusion of $\mathcal{X} \rightarrow \mathcal{X} \times I$.

$$\begin{array}{ccc} \mathcal{X} & \xrightarrow{f} & \mathcal{Y} \\ \downarrow \simeq & \searrow & \downarrow \simeq \\ \mathcal{X} \times I & \longrightarrow & \text{Cyl } f \end{array}$$

We'd like to show that the induced maps on homology do indeed commute. Because \mathcal{Y} is a deformation retract of $\text{Cyl } f$, this would imply that the induced map on homology from the inclusion $\mathcal{X} \hookrightarrow \text{Cyl } f$ is the same as the induced map on homology of $f : \mathcal{X} \rightarrow \mathcal{Y}$.

The inclusion $G_* : C_*(\mathcal{X}) \rightarrow C_*(\text{Cyl } f)$ traversing the left and bottom arrows of the diagram is given by $G(x) = x \times \{0\}$, and the map through \mathcal{Y} $F_* : C_*(\mathcal{X}) \rightarrow C_*(\text{Cyl } f)$ traversing the top and right arrows of the diagram is given by $F(x) = f(x) \times \{1\}$, where if $f(x) = \sum_i \alpha_i y_i$, then $f(x) \times \{1\} = \sum_i \alpha_i (y_i \times \{1\})$.

We now define maps $M_k : C_k(\mathcal{X}) \rightarrow C_{k+1}(\text{Cyl } f)$ mapping x to the image of $x \times [0, 1]$ in the cylinder. Explicitly, the basis element $x \in C_k(\mathcal{X})$ is mapped to the basis element $x \times [0, 1]/f \in C_{k+1}(\text{Cyl } f)$ with boundary $-(\partial x) \times [0, 1]/f + f(x) \times \{1\} - x \times \{0\}$, appearing in the left block of Equation (2.21). Now, we simply verify

$$\begin{aligned}
& \partial_k^{\text{Cyl } f} M_k(x) + M_{k-1} \partial_k^X(x) \\
&= -(\partial x) \times [0, 1]/f + f(x) \times \{1\} - x \times \{0\} + M_{k-1} \partial_k^X(x) \\
&= -(\partial x) \times [0, 1] + f(x) \times \{1\} - x \times \{0\} + (\partial x) \times [0, 1]/f \\
&= f(x) \times \{1\} - x \times \{0\} \\
&= F(x) - G(x)
\end{aligned}$$

so M_* is a chain homotopy. Thus the induced maps on homology are isomorphic: $\tilde{F}_* \simeq \tilde{G}_*$.

2.8.3 Persistent Homology

More generally, if we have a sequence of maps

$$X_0 \xrightarrow{f_0} \mathcal{X}_1 \xrightarrow{f_1} \dots \xrightarrow{f_{N-1}} \mathcal{X}_N$$

we can construct the *mapping telescope* [57]

$$\text{Tel } f_i = \left[(\mathcal{X}_N \times \{N\}) \sqcup \bigsqcup_{i=0}^{N-1} (\mathcal{X}_i \times [i, i+1]) \right] / [(x, i+1) \sim (f_i(x), i+1)]$$

And we can use a filtration that adds each subsequent cylinder at times $t = 1, 2, \dots, N$.

Chapter 3

Quiver Representations and Algorithms

In Section 1.1.3, we introduced persistent and zigzag homology barcodes as invariants of A -type quiver representations obtained through the homology functor. This chapter will introduce new algorithms for computing these barcodes using an explicit matrix factorization framework.

We will focus on classification of A -type quiver representations by putting an associated companion matrix into a canonical form. We consider a matrix A associated with the representation, which has a block structure of the directed adjacency matrix of the underlying graph, and each non-zero block contains the map of vector spaces along each edge (induced maps on homology). The goal is to find a factorization of the matrix

$$A = B\Lambda B^{-1} \tag{3.1}$$

In which B is an invertible block diagonal matrix, and Λ has the same block structure as A , but blocks are put in a canonical form which is unique up to permutation. Once we have a factorization $A = B\Lambda B^{-1}$, it is easy to read off the persistence or zigzag barcode from the matrix Λ , which we will cover in Section 3.2.

This is a powerful conceptual viewpoint. First, it shows that the barcodes used and studied in persistent and zigzag homology are as fundamental as matrix decompositions studied in other areas of linear algebra. Second, it lays the groundwork for a new set of algorithms for topological data analysis.

3.1 Prior Work on Computing Zigzag Homology

The study of persistent and zigzag homology as examples of type-A quiver representations was begun in [22], which is the theoretical basis for our algorithm. This has led a variety of interesting theoretical and applied work as surveyed in [84], but algorithmic implementations so far have not significantly leveraged this connection.

An algorithm for computing persistent homology was first described in [46] (for \mathbb{F}_2 coefficients) and was extended to general fields in [106], and an algorithm for computing zigzag homology was first described in [23]. These approaches operate on inclusion maps between spaces, and computations work directly on chain complex boundary matrices. While both persistent and zigzag homology are known to be computable in matrix-multiplication time [77], sparsity considerations are typically much more important in practice, and in order to compute on large data sets, several approaches have been pursued. First, there have been efforts to speed up computation of persistence through various optimizations [10, 33, 38] and high performance implementations [9, 11, 59, 72, 79, 105]. Second, there have been efforts to reduce the inherent size of computations using methods that preserve the homotopy type of a space while reducing the size of its combinatorial representation [13, 43, 78, 103]. Zigzag homology has received less attention than persistence, but similar efforts can be found in [73, 74]. The use of non-inclusion maps in persistent and zigzag homology has been somewhat limited in topological data analysis, although the case of simplicial maps has been investigated in [40, 63], based on a strategy that uses zigzag homology to compute a persistence barcode, and the implementation of zigzag homology in Javaplex [99] contains tools to compute induced maps for the bivariate witness construction [22, 98].

Our approach has several notable differences compared to existing computational approaches for persistent and zigzag homology. First, we consider a two step approach, where first induced maps on homology are computed to form a quiver representation, and then our algorithm extracts the barcode. In contrast, existing approaches work almost exclusively on the level of chain complexes, missing out on the abstraction and compression that induced maps on homology afford. We believe that the two approaches are complementary, and that many existing optimizations could be applied when computing homology and induced maps. Second, our approach works for general cell complexes, and general cell maps, whereas some existing approaches are focused on the simplicial (or cubical) complexes and simplicial maps. We also work with arbitrary field coefficients, when some existing approaches for induced maps are limited to \mathbb{F}_2 coefficients [40]. Third, our approach offers multiple opportunities for parallelization, whereas most existing approaches are sequential in nature. Computing induced maps on homology is trivially parallelizable, and our quiver algorithm also admits a divide and conquer parallelization scheme. The first divide and conquer approach for zigzag homology was described at a high level in [23]. The approach operates on chain complexes instead of induced maps on homology, and has not been implemented. Another approach observed the embarrassingly parallel computation of induced maps, and provided a divide and conquer scheme for induced maps based on pullbacks [94], but again, these observations were not implemented in any parallel framework. Greg Henselman has also had thoughts on an approach for divide and conquer [58]. A scheme to simplify complexes that is trivially parallelizable for reasons similar to the trivial parallelization of induced maps is found in [13]. Our parallelization scheme is different and complementary to existing efforts to use spectral sequences to parallelize homology calculations [69, 104]. Finally, we unify computations for persistent and zigzag homology to a degree that is not seen in existing algorithms, in the sense that modifications to handle arrows of different directions are trivial when stated in terms of matrix factorizations.

Our approach is closest in spirit to the original paper on zigzag homology [22], which started with induced maps on homology, and gave a constructive algebraic algorithm for the interval decomposables for A-type quivers. In Chapter 2, we addressed

the computational questions that are necessary for computing arbitrary induced maps on homology, and in this chapter, we will provide an explicit algorithm for computing interval indecomposables. In contrast to [22] and existing algorithms for zigzag homology [73, 74, 79, 99], our algorithm does not explicitly use right filtrations, and instead uses an approach that involves producing a matrix factorization. Available zigzag homology implementations can be found in Javaplex [99], which is based on the algebraic algorithm in [22], and Dionysus [79], which is based on the algorithm in [23], but neither of these implementations employs parallelization.

As a companion to this work, we have released a new open-source package for the algorithms we describe. Inspired by the basic linear algebra subprograms (BLAS) [65], we call it the basic applied topology subprograms (BATS). The library includes high-level C++ templates for the algorithms described, employing OpenMP for parallelization. The code is publicly available at <https://github.com/bnel/BATS>.

3.2 Classification of Quiver Representations and Barcodes

The application of quiver representations to persistent and zigzag homology has been of interest ever since it was used in the context of zigzag barcodes. A fairly complete survey of existing results and applications to topological data analysis can be found in [84]. In this section, we will introduce the necessary background to understand our algorithm.

3.2.1 Quiver Representations

A quiver is a mathematical term for what is known in computer science as a directed multi-graph. A quiver representation is a directed multi-graph $\mathcal{Q}(V, E)$ where every vertex $v_i \in V$ has an associated vector space V_i over a common field \mathbb{F} , and each directed edge $(v_i, v_j) \in E$ has an associated \mathbb{F} -linear transformation $A_{i,j} : V_i \rightarrow V_j$. Two quiver representations $\mathcal{Q}_1(V^1, E^1)$, $\mathcal{Q}_2(V^2, E^2)$ are said to be isomorphic if the underlying graphs are isomorphic and there are isomorphisms B_i for each V_i^2 so

that $A_{i,j}^1 = (B_j)^{-1}A_{i,j}^2B_i$. Quiver representation theory is concerned with classifying quivers up to isomorphism, a problem that originated with classification of Lie Algebras [39]. The ability to classify arbitrary quiver representations relies entirely on the underlying undirected graph, and not on the dimensions of the vector spaces or ranks of maps. Both persistent and zigzag homology are quiver representations of type A_n , for which the underlying graph is a line graph on n vertices, where n can be any finite positive integer.



A theorem due to Gabriel shows that the collection of underlying graphs of quiver representations that have a finite number of indecomposable representations are known as the Dynkin diagrams, which include A_n , as well as several other classes of graph [50]. We will refer to type A_n quivers where all arrows point in the same direction *persistence-type* quivers, and type A_n quivers where arrows alternate direction *zigzag-type* quivers.

3.2.2 From Topology to Quiver Representations

Quiver representations arise naturally from diagrams of topological spaces through the homology functor. In Section 2.4, we saw that homology (with coefficients in a field \mathbb{F}) is a functor that associates topological vector spaces X with vector spaces $H_k(X)$, and maps $f : X \rightarrow Y$ to linear transformations $\tilde{F}_k : H_k(X) \rightarrow H_k(Y)$. This means that the homology functor turns diagrams of topological spaces into diagrams of vector spaces (quiver representations)

Example 3.2.1. *The homology functor creates the following transformation of diagrams*

$$\begin{array}{ccc}
 X & \xrightarrow{f} & Y \\
 \downarrow g & \nearrow j & \\
 Z & & \\
 & \nwarrow i & \\
 & &
 \end{array}
 \Rightarrow
 \begin{array}{ccc}
 H_k(X) & \xrightarrow{\tilde{F}_k} & H_k(Y) \\
 \downarrow \tilde{G}_k & \nearrow \tilde{J}_k & \\
 J_k(Z) & & \\
 & \nwarrow \tilde{I}_k & \\
 & &
 \end{array}$$

where $\tilde{F}_k = H_k(f)$, $\tilde{G}_k = H_k(g)$, $\tilde{J}_k = H_k(j)$, and $\tilde{I}_k = H_k(i)$

Note that neither the diagram of topological spaces nor the diagram of vector spaces is required to commute.

Example 3.2.2. *Persistent homology studies diagrams of spaces*

$$X_0 \xrightarrow{f_0} X_1 \xrightarrow{f_1} \dots$$

which produces a quiver representation

$$H_k(X_0) \xrightarrow{(F_0)_k} H_k(X_1) \xrightarrow{(F_1)_k} \dots$$

which corresponds to a Dynkin diagram of type A .

Similarly, zigzag homology studies diagrams of type A .

The advantage of using quiver representations to study persistent and zigzag homology instead of using an approach such as the one introduced in Section 2.8 is that the application of the homology functor is *embarassingly parallelizable*. In fact, this can be applied to general diagrams of spaces.

When it comes to representing diagrams of topological spaces or vector spaces, we will think of using the same data structure: a directed multi-graph

Definition 3.2.3. A directed multi-graph (V, E) is a collection V of vertices (nodes), and a multiset E of edges in $V \times V$.

A directed multi-graph is a more general version of a *graph*, in that edges are directed, and multiple edges can share the same source and target. We will only consider directed multi-graphs consisting of a finite number of nodes and edges.

A diagram in a category is then just a directed multi-graph with an additional datum for each node and edge.

Example 3.2.4. A diagram in \mathbf{Top} is a directed multi-graph where every node $i \in V$ has an associated space X_i , and every edge $(i, j) \in E$ has an associated map $f_{i,j} : X_i \rightarrow X_j$.

Example 3.2.5. A diagram in $\mathbf{Vect}_{\mathbb{F}}$ is a directed multi-graph where every node $i \in V$ has an associated vector space V_i , and every edge $(i, j) \in E$ has an associated linear transformation $T_{i,j} : V_i \rightarrow V_j$. In other words, a quiver representation.

The homology functor turns a diagram in \mathbf{Top} to a diagram in $\mathbf{Vect}_{\mathbb{F}}$ which has the same underlying directed multi-graph structure, meaning the vertex and edge sets are the same, but the associated data is different.

Algorithm 4 Obtain a quiver representation from a diagram of spaces

- 1: **Input:** Directed multi-graph (V, E) with associated spaces X_i and maps $f_{i,j}$
 - 2: **Result:** Directed multi-graph (V, E) with associated vector spaces $H_k(X_i)$ and maps $\tilde{F}_{i,j} = H_k(f_{i,j})$
 - 3: **for** $i \in V$ **do**
 - 4: Obtain chain complex $C_*(X_i)$
 - 5: Obtain homology revealing bases (U_*^i, \mathcal{I}_*^i) as well as reduced boundaries R_*^i using Algorithm 1.
 - 6: **end for**
 - 7: **for** $(i, j) \in E$ **do**
 - 8: obtain $(\tilde{F}_{i,j})_* = H_*(f_{i,j})$ using Algorithm 3.
 - 9: **end for**
-

The important observation is that both for loops of Algorithm 4 have *completely independent iterations*. That is, computing homology of X_i can be done completely independently of computing homology for X_j . Similarly, computing the induced maps $\tilde{F}_{i,j}$ only requires the pre-computed information for the source and target of the associated edge, and is independent of the computation on any other edge. This means the algorithm is *embarrassingly parallelizable*, meaning that each for-loop iteration can be computed in parallel given enough processors. Furthermore, this is true for any diagram of spaces, meaning it will apply not only to the persistent and zigzag homology diagrams that we will study in this paper, but also other situations such as multiparameter persistence [26].

3.2.3 Type A Quiver Representations

We will now focus on classification of type A_n quiver representations, which appear for both persistent and zigzag homology. The indecomposable representations of these

quiver representations are known as interval indecomposables [22, 50, 84], and have the form

$$I[b, d] = \cdots \longrightarrow 0 \longrightarrow \mathbb{F} \longrightarrow \cdots \longrightarrow \mathbb{F} \longrightarrow 0 \longrightarrow \cdots$$

where b denotes the first index at which a copy of \mathbb{F} appears, and d denotes the final index where \mathbb{F} appears, all vector spaces with index $i \in [b, d]$ also have a copy of \mathbb{F} , with identity maps along all edges connecting two copies of \mathbb{F} and zero maps along all other edges. In other words, any quiver representation of type A_n is isomorphic to the direct sum of these indecomposables

$$\mathcal{Q} \cong \bigoplus_i I[b_i, d_i]$$

As a convention, we will use the lexicographical (total) order on \mathbb{Z}^2 when ordering interval indecomposables, using the parameters b, d . When the quiver representation is produced from induced maps on homology, the multiset $\{(b_i, d_i)\}$ is the barcode.

Definition 3.2.6. *The companion matrix of a quiver representation \mathcal{Q} is the block matrix which has non-zero blocks in the non-zero entries of the adjacency matrix of the underlying directed graph, where blocks are filled by the linear transformations along the corresponding edges. By necessity, the size of the i -th block must be the dimension of V_i in the quiver representation.*

These matrices act on the vector space $V = \bigoplus_i V_i$ by sending vectors to their images in each linear transformation in \mathcal{Q} . While general quiver representations may have multiple arrows between vector spaces, companion matrices can only represent those which have a underlying graph with at most one linear transformation on each directed edge – this will not limit our study of type A_n quiver representations which satisfy this property.

For example a persistence quiver $P_4 = \cdot \leftarrow \cdot \leftarrow \cdot \leftarrow \cdot$ will have a companion matrix of the form

$$\begin{bmatrix} 0 & A_1 & & \\ & 0 & A_2 & \\ & & 0 & A_4 \\ & & & 0 \end{bmatrix}$$

whereas a zigzag quiver $Z_4 = \cdot \rightarrow \cdot \leftarrow \cdot \rightarrow \cdot$ will have a companion matrix of the form

$$\begin{bmatrix} 0 & & & \\ A_1 & 0 & A_2 & \\ & & 0 & \\ & & A_3 & 0 \end{bmatrix}$$

Quiver representation isomorphism classes are maintained by conjugation of the companion matrix by block-diagonal change of bases matrices. For example two persistence quivers of type $P_3 = \cdot \leftarrow \cdot \leftarrow \cdot$ with companion matrices A and B respectively are isomorphic if there exists an invertible matrix $M = M_1 \oplus M_2 \oplus M_3$ acting on V such that

$$\begin{bmatrix} 0 & A_1 & \\ & 0 & A_2 \\ & & 0 \end{bmatrix} = \begin{bmatrix} M_1 & & \\ & M_2 & \\ & & M_3 \end{bmatrix} \begin{bmatrix} 0 & B_1 & \\ & 0 & B_2 \\ & & 0 \end{bmatrix} \begin{bmatrix} M_1^{-1} & & \\ & M_2^{-1} & \\ & & M_3^{-1} \end{bmatrix}$$

From this, it is clear that two quiver representations can not be isomorphic if they do not share the same underlying directed multi-graph, and the dimensions of the vector spaces are not identical.

An *indecomposable factorization* of the companion matrix A is a factorization $A = BTB^{-1}$, where B is an invertible (change of basis) matrix, and $T = \bigoplus_i I[b_i, d_i]$ is the matrix of indecomposables. We will allow for the indecomposable block to appear in any order but for exposition we will use the lexicographic partial order on \mathbb{Z}^2 to order the pairs (b_i, d_i) . For example, in the case where P_4 and Z_4 both have

Algorithm 5 Barcode Extraction

```

1: Input: Barcode matrix  $\Lambda$ , ranks of vector spaces  $V_i$  and directions of arrows in
   quiver.
2: Result: Barcode  $\mathcal{B}$ 
3: for  $i = 1, \dots, n$  do
4:   for  $j = 1, \dots, \text{rank } V_i$  do
5:     if  $V_{i-1} \rightarrow V_i$  then
6:       if Row  $j$  in block  $i$  contains a non-zero in column  $j'$  of block  $i - 1$  then
7:         Extend the bar at index  $j'$  of block  $i - 1$  to have index  $j$  in block  $i$ .
8:       else
9:         Begin a bar with index  $j$  in block  $i$ 
10:      end if
11:     else if  $V_{i-1} \leftarrow V_i$  then
12:       if Column  $j$  in block  $i$  contains a non-zero in row  $j'$  of block  $i - 1$  then
13:         Extend the bar at index  $j'$  of block  $i - 1$  to have index  $j$  in block  $i$ .
14:       else
15:         Begin a bar with index  $j$  in block  $i$ 
16:       end if
17:     else
18:       ( $i = 1$ )
19:       Begin bar with index  $j$  in block 1
20:     end if
21:   end for
22: end for
23: return  $\mathcal{B}$ 

```

has exactly one non-zero, or maps to zero (the bar ends at V), in which case the corresponding column of A_i is zero. Every basis vector $w \in W$ is either in the image of a basis vector of V , in which case the corresponding row of A_i has exactly one non-zero, or is the start of a new bar, in which case the row of A_i is zero. Thus A_i is a pivot matrix because it has at most one non-zero in each row and column. \square

3.2.4 The Graded Module Structure of Persistent Homology

For persistence-type quivers, there is a relationship between the indecomposable factorization and the Jordan normal form of a matrix, observed in equation Equation (3.2). Because interval indecomposables have the structure of a sub-graph of the directed graph underlying the quiver, in the case of persistence quivers they will all have the form of a Jordan zero block. That is the sub-matrix associated with $I[a, b]$ is equal to $J_{[1+b-a]}(0)$, where

$$J_i(\lambda) = \begin{bmatrix} \lambda & 1 & & \\ & \lambda & \ddots & \\ & & \ddots & 1 \\ & & & \lambda \end{bmatrix}$$

is an $i \times i$ matrix. This immediately shows that the characteristic polynomial of A , $\det(tI - A) = t^N$, where $N = \sum \dim V_i$ in the quiver representation, and that A is nilpotent with index corresponding to the length of the longest interval $1 + b - a$. While the Jordan form does not generally exist for non-algebraically closed fields, Greg Henselman showed in his thesis that nilpotent operators can always be decomposed into Jordan-zero blocks [60], and applied this along with the use of Schur complements to computing persistent homology of nested chain complexes. We also note that because homology of spaces starts with integer chain complexes, that we only need to use the field of fractions \mathbb{Q} for computation, and so $H_*(\cdot; \mathbb{R}) \cong H_*(\cdot; \mathbb{Q})$. In practice, we avoid any issues with finite precision observed when computing the Jordan form with real coefficients [54] through use of rational numbers.

Classification of persistent homology modules was established in [106] by showing that the persistent homology of inclusions has a $\mathbb{F}[T]$ module structure, where T acts by incrementing the filtration parameter. This first showed that the persistence barcode of a filtration is unique and computable. Explicitly, because $\mathbb{F}[T]$ modules are principal ideal domains, there is a unique description of the persistence module as

$$\bigoplus_{i \in \mathcal{I}} T^{b_i} \mathbb{F}[T] / T^{d_i} \oplus \bigoplus_{i \in \mathcal{J}} T^{b_i} \mathbb{F}[T]$$

where \mathcal{I} is the set of finite bars in a persistence diagram, and \mathcal{J} is the set of infinite bars. b_i and d_i correspond to the birth and death parameters in the discrete-time filtration.

In order to see the equivalence of the quiver representation and $\mathbb{F}[T]$ module viewpoints, we will consider infinite extensions of persistence quivers, where after a finite N all maps are isomorphisms

$$V_0 \xrightarrow{A_0} V_1 \xrightarrow{A_1} \dots \xrightarrow{A_{N-1}} V_N \xrightarrow{I} V_{N+1} \xrightarrow{I} \dots$$

In this case, any interval that is present at parameter N will extend infinitely to the right. Persistence quivers of this form appear naturally when considering homology of a filtered space

$$X_0 \subseteq X_1 \subseteq X_2 \subseteq \dots \subseteq X_N = X_{N+1} = \dots = X$$

The interval indecomposables will appear as transposed Jordan zero blocks, which are equivalent to the ordinary Jordan blocks via conjugation by the anti-diagonal matrix J

$$J^T \begin{bmatrix} 0 & & & & \\ 1 & 0 & & & \\ & \ddots & \ddots & & \\ & & & 1 & 0 \end{bmatrix} J = J_i(0) \tag{3.4}$$

The infinite extension of the quiver requires infinite indecomposables $I[b, \infty)$, represented by blocks of the form

$$\begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ & \ddots & \ddots & \\ & & & \ddots \end{bmatrix}$$

If we take $V = \bigoplus_i V_i$, and A to be the companion matrix of the quiver, there is $\mathbb{F}[A]$ module structure on V . The block structure of A shows that this is a graded module, graded by the index i of V_i , and the barcode factorization $A \cong \Lambda = \bigoplus_{i \in \mathcal{I}} I[b_i, d_i] \bigoplus_{j \in \mathcal{J}} I[b_j, \infty)$ shows gives a way to extract a basis for the module: $\{v_i\}_{i \in \mathcal{I}} \cup \{v_j\}_{j \in \mathcal{J}}$, where v_i is the vector with grade b_i that generates the subspace of the associated Jordan block. In the case of finite bars, these generate a torsion sub-module that disappears at grade d_i , so the sub-module is isomorphic to $T^{b_i}\mathbb{F}[T]/T^{d_i}$. In the case of infinite bars, the sub-module is free and so is isomorphic to $T^{b_i}\mathbb{F}[T]$.

We note that in practice, infinite extensions of quivers are not necessary for computation. We can simply take any bars that are present in the last vector space and extend them infinitely.

The module structure of persistent homology offers some insight into why the reduction algorithm works. The reason why only columns to the left can be used to eliminate pivots is that the grade of a basis element can not be altered under a change of basis. In other words, to maintain a $\mathbb{F}[T]$ module isomorphism it is valid to add elements with lower grades to elements with higher grades, but not *vice versa*.

3.2.5 Generic Quiver Computations

There is a correspondence between diagrams encoding quiver representations and block matrices encoding their companion matrices, and certain operations are easier to express using one notation or the other. In this section we establish some lemmas that apply to any quiver representation.

Lemma 3.2.8. *A change of basis (quiver representation isomorphism) at a single vertex via an invertible matrix M can be represented as*



Proof. This follows immediately from a change of basis on the central vector space in the diagram via Equation (2.1). □

If the quiver representation is representable by a companion matrix, this diagrammatically encodes the isomorphism

$$\begin{bmatrix} & B_0 & \dots & B_m \\ A_0 & & & \\ \vdots & & & \\ A_n & & & \end{bmatrix} = \begin{bmatrix} M & & & \\ & I & & \\ & & \ddots & \\ & & & I \end{bmatrix} \begin{bmatrix} & M^{-1}B_0 & \dots & M^{-1}B_m \\ A_0M & & & \\ \vdots & & & \\ A_nM & & & \end{bmatrix} \begin{bmatrix} M^{-1} & & & \\ & I & & \\ & & \ddots & \\ & & & I \end{bmatrix}$$

We see that this only affects linear transformations that have the center vertex as a source or target. For any vector spaces that do not have an arrow to or from the center vector space are multiplied by an identity on both left and right and are unaffected.

Lemma 3.2.8 implies the following two corollaries which set forth the rules for our *matrix passing algorithms*.

Corollary 3.2.9. *Passing an invertible matrix M through a target yields*



Corollary 3.2.10. *Passing an invertible matrix M through a source yields*



Notice that we draw arrows from right to left in the above diagrams. This is simply because the matrix M is closest to the vertex undergoing a change of basis. If we write arrows right to left we would have the correct, but less natural looking example

$$\cdot \xrightarrow{B_0} \cdot \xrightarrow{\tilde{A}_0 M} \cdot \quad \cong \quad \cdot \xrightarrow{M B_0} \cdot \xrightarrow{\tilde{A}_0} \cdot$$

One can use this to easily verify the change of basis formula for induced maps on homology Equation (2.12) by tracking the extraction of the cycle-revealing bases in a chain map $F_* : C_* \rightarrow D_*$, using factorizations $\partial_k = R_k U_k^{-1}$ and looking at the relevant sub-quiver:

$$\begin{array}{ccc} C_{k-1} & \xleftarrow{R_k^C (U_k^C)^{-1}} & C_k \\ \downarrow & & \downarrow F_k \\ D_{k-1} & \xleftarrow{R_k^D (U_k^D)^{-1}} & D_k \end{array} \cong \begin{array}{ccc} C_{k-1} & \xleftarrow{R_k^C} & C_k \\ \downarrow & & \downarrow (U_k^D)^{-1} F_k U_k^C \\ D_{k-1} & \xleftarrow{R_k^D} & D_k \end{array}$$

Generally, quiver representations encoding induced maps on homology can be derived from diagrams of chain maps.

The generic quiver representation computations are useful in representing a wide variety of computations. For example we can cast the problem of computing persistent homology from chain complex information as a factorization on the following grid quiver. Since computing the barcodes entails a change of basis, it can be expressed using matrix passing algorithms using the generic quiver computation.

$$\begin{array}{cccccc}
 \cdot & \xleftarrow{\partial_{1,0}} & \cdot & \xleftarrow{\partial_{1,1}} & \dots & \xleftarrow{\partial_{1,n-2}} & \cdot & \xleftarrow{\partial_{1,n-1}} & \cdot \\
 \downarrow \phi_{1,1} & & \downarrow \phi_{1,2} & & & & \downarrow \phi_{1,n-1} & & \downarrow \phi_{1,n} \\
 \cdot & \xleftarrow{\partial_{2,0}} & \cdot & \xleftarrow{\partial_{2,1}} & \dots & \xleftarrow{\partial_{2,n-2}} & \cdot & \xleftarrow{\partial_{2,n-1}} & \cdot \\
 \downarrow \phi_{2,1} & & \downarrow \phi_{2,2} & & & & \downarrow \phi_{2,n-1} & & \downarrow \phi_{2,n} \\
 \vdots & & \vdots & & \dots & & \vdots & & \vdots
 \end{array}$$

As another example, here is the standard type A quiver representation equipped with some extra identity maps going in and out of each vertex. When the factorization algorithm is applied to the type A sub-quiver, the identity maps will be modified and will record the basis change matrices required for the factorization. In this way we see that these quiver diagrams and associated quiver computations are a useful tool for expressing these algorithms.

$$\begin{array}{cccc}
 \cdot & & \cdot & \\
 \uparrow I & & \uparrow I & \\
 \cdot & \xleftarrow{A_0} & \cdot & \xleftarrow{A_1} \dots \xleftarrow{A_{n-2}} & \cdot & \xleftarrow{A_{n-1}} & \cdot \\
 \uparrow I & & \uparrow I & & \uparrow I & & \uparrow I \\
 \cdot & & \cdot & & \cdot & & \cdot
 \end{array}$$

$$\begin{array}{cccc}
 \cdot & & \cdot & \\
 \uparrow B_0 & & \uparrow B_1 & \\
 \cdot & \xleftarrow{E_0} & \cdot & \xleftarrow{E_1} \dots \xleftarrow{E_{n-2}} & \cdot & \xleftarrow{E_{n-1}} & \cdot \\
 \uparrow B_0^{-1} & & \uparrow B_1^{-1} & & \uparrow B_{n-1}^{-1} & & \uparrow B_n^{-1} \\
 \cdot & & \cdot & & \cdot & & \cdot
 \end{array}$$

3.3 Algorithms for Canonical Forms of Type-A Quiver Representations

In this section, we will describe our algorithm for computing the canonical form of a type-A quiver representation (Equation (3.1)). As we will see, the details of the

algorithm depends on the direction of the maps in the quiver, but there are core components which are the same.

There are two basic linear algebra operations we need as primitives for the algorithm.

1. Triangular factorizations
2. Shape commutation results with E-type matrices

We will first discuss these. Next we will consider the algorithm when all the the arrows point in the same direction. We will then show how this algorithm can be modified to the case of alternating arrow directions. Finally we will specify the full general algorithm.

3.3.1 Triangular Factorizations

In this section, we will discuss computing decompositions of a matrix A into triangular matrices with row or column pivoting. Specifically we will start with the LEUP decomposition. Variants include PLEU, UELP and PUEL form, which can be derived using the LEUP decomposition of either transposed or J-Conjugated versions of A . These factorizations are all variants of the standard LU decomposition [53], but we will need these specific forms for our algorithm.

Given a matrix A , we will describe an algorithm that will maintain a block invariant at each step i

$$A = \begin{bmatrix} L_{11} & \\ & I \end{bmatrix} \begin{bmatrix} E_{11} & \\ & \tilde{A} \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ & I \end{bmatrix} P \quad (3.5)$$

Proof of Correctness: We will show that the invariant Equation (3.5) is maintained at each step of the algorithm. Note that the loop increments i each iteration, so we use i as our index. For rows of A , as well as rows and columns of L , we will use the block index set $1 = \{0, \dots, i - 1\}$, and the block index set $2 = \{i + 1, \dots, m - 1\}$, and for columns of A , as well as rows and columns of U we will use the block index set $1 = \{0, \dots, j - 1\}$ and block index set $2 = \{j + 1, \dots, n - 1\}$.

Algorithm 6 LEUP factorization

```

1: Input:  $m \times n$  matrix  $A$ 
2: Result: Factorization  $A = LEUP$ 
3:  $L = I_m$ 
4:  $E = A$ 
5:  $U = I_n$ 
6:  $P = I_n$ 
7:  $i = 1, j = 1$ 
8: while  $i \leq m$  &  $j \leq n$  do
9:   if row  $i$  has a non-zero in column  $j' \geq j$  of  $E$  then
10:    Swap columns  $j, j'$  in  $E$ 
11:    Take Schur complement with respect to  $i, j$  entry
12:     $i = i + 1$ 
13:     $j = j + 1$ 
14:  else
15:     $i = i + 1$ 
16:  end if
17: end while

```

Assume the invariant is maintained at the beginning of iteration i . We can break up the matrices into

$$A = \begin{bmatrix} L_{11} & & \\ L_{i1} & 1 & \\ L_{21} & & I \end{bmatrix} \begin{bmatrix} E_{11} & & \\ & A_{ij} & A_{i2} \\ & A_{2j} & A_{22} \end{bmatrix} \begin{bmatrix} U_{11} & U_{1j} & U_{22} \\ & 1 & \\ & & I \end{bmatrix} P$$

In the case that there are no non-zero entries in A_{ij} or A_{i2} , we simply increment i (in the else clause of the while loop), and the invariant is trivially maintained.

In the case there is a non-zero entry in A_{ij} or A_{i2} , assume we have already permuted it to the A_{ij} position, and used the relation Equation (3.10) to move the permutation to the right. We can write the interior matrix as

$$\begin{bmatrix} E_{11} & & \\ & A_{ij} & A_{i2} \\ & A_{2j} & A_{22} \end{bmatrix} = \begin{bmatrix} I & & \\ & I & \\ & A_{2j}A_{ij}^{-1} & I \end{bmatrix} \begin{bmatrix} E_{11} & & \\ & A_{ij} & \\ & & S \end{bmatrix} \begin{bmatrix} I & & \\ & I & A_{ij}^{-1}A_{i2} \\ & & I \end{bmatrix}$$

where S is the Schur complement $S = A_{22} - A_{2j}A_{ij}^{-1}A_{i2}$. We can then pass off the left matrix to L and the right matrix to U , and now group A_{ij} with the echelon block E_{11} .

Note that because we may increment i without incrementing j , that the matrix E will be of type E_L .

Other Triangular Factorizations

$$A = LE_LUP \quad \blacksquare = \begin{matrix} \color{orange}\blacksquare & \color{blue}\diagdown & \color{green}\blacksquare & \color{blue}\boxplus \end{matrix} \quad (3.6)$$

$$A = PLE_UU \quad \blacksquare = \begin{matrix} \color{blue}\boxplus & \color{orange}\blacksquare & \color{blue}\diagdown & \color{green}\blacksquare \end{matrix} \quad (3.7)$$

$$A = U\hat{E}_U LP \quad \blacksquare = \begin{matrix} \color{green}\blacksquare & \color{blue}\diagdown & \color{orange}\blacksquare & \color{blue}\boxplus \end{matrix} \quad (3.8)$$

$$A = PU\hat{E}_L L \quad \blacksquare = \begin{matrix} \color{blue}\boxplus & \color{green}\blacksquare & \color{blue}\diagdown & \color{orange}\blacksquare \end{matrix} \quad (3.9)$$

All of the above factorizations can be shown to be equivalent to the LE_LUP factorization, using transposes and conjugation with J matrices.

For example consider the case,

$$A = (A^T)^T = (LE_LUP)^T = P^T U^T (E_L)^T L^T = \tilde{P} \tilde{L} E_U \tilde{U}$$

Thus the PLE_UU factorization is equivalent to the LE_LUP factorization of the transpose.

Similarly the $U\hat{E}_U LP$ can be expressed using J conjugation,

$$A = JJAJJ = J(JAJ)J = J\hat{A}J$$

Now we can replace \hat{A} with its LE_LUP factorization, and apply the commutation relations of J .

$$J\hat{A}J = JLE_LUPJ = \hat{U}JE_LUPJ = \hat{U}\hat{E}_UJUPJ = \hat{U}\hat{E}_U\hat{L}JPJ = \hat{U}\hat{E}_U\hat{L}\hat{P}$$

Thus we have found the $U\hat{E}_U LP$ factorization of A .

If we apply the PLE_UU factorization for \hat{A} instead, we get,

$$J\hat{A}J = JPLE_UUJ = \hat{P}JLE_UUJ = \hat{P}\hat{U}JE_UUJ = \hat{P}\hat{U}\hat{E}_LJUJ = \hat{P}\hat{U}\hat{E}_L\hat{L}$$

This gives us the $PU\hat{E}_LL$ factorization for the matrix A .

3.3.2 Shape Commutation Relations

Now, we'll consider shape commutation relations of echelon matrices with triangular matrices. We will first show the following commutation relationship, and derive others from this.

Proposition 3.3.1. *Given an echelon pivot matrix E_L and lower triangular matrix L , we can rewrite the product E_LL in the following way*

$$E_LL = \tilde{L}E_L \tag{3.10}$$

Where \tilde{L} matrix is also a lower triangular matrix.

Proof. In terms of matrix shapes this commutation looks like the following

$$\begin{array}{|c|} \hline \blacksquare \\ \hline \end{array} \begin{array}{|c|} \hline \blacksquare \\ \hline \end{array} = \begin{array}{|c|} \hline \blacksquare \\ \hline \end{array} \begin{array}{|c|} \hline \blacksquare \\ \hline \end{array}$$

Note that this is just the commutation of the shapes, In general L and \tilde{L} are not the same matrices, or even the same dimensions. We will first characterize left and right multiplication by the E_L matrix.

Consider the product of E_L with an arbitrary matrix A

$$(E_LA)[k, l] = \sum_p E_L[k, p]A[p, l]$$

From the definition of the shape of E_L , we know that its entries are 1 only when $p = j(k)$, thus we get

$$(E_L A)[k, l] = \sum_p E_L[k, p]A[p, l] = E_L[k, j(k)]A[j(k), l] = A[j(k), l]$$

Here we use the convention that if the entry for a particular k, l index pair is not assigned any value, it is by default 0. Similarly, if we are attempting to apply the function $j(\cdot)$ on an index not in its domain, the appropriate matrix element is 0. Multiplication on the right follows similarly

$$(A E_L)[k, l] = \sum_p A[k, p]E_L[p, l] = A[k, i(l)]E_L[i(l), l] = A[k, i(l)]$$

We will now show that \tilde{L} , constructed in the following way will satisfy the proposition.

$$\tilde{L}[k, l] = \begin{cases} L[j(k), j(l)], & \text{if } k \in \text{Domain}(j(\cdot)) \text{ and } l \in \text{Domain}(j(\cdot)) \\ 1, & \text{if } k = l \text{ and } k \notin \text{Domain}(j(\cdot)) \\ 0, & \text{otherwise} \end{cases}$$

Note that the above construction is not unique, we have opted to set the diagonal of otherwise zero columns to 1, this makes the matrix invertible, which will be useful later on. Next we will show that this construction is indeed lower triangular.

Suppose $k < l$, then we have from the definition of the shape of E_L that $j(k) < j(l)$ which implies that $L[j(k), j(l)] = 0$, as L is a lower triangular matrix. This shows that for $k < l$, $\tilde{L}[k, l] = 0$, hence \tilde{L} is lower triangular. Now we will compute $E_L L$ and $\tilde{L} E_L$ and show that they are equal.

$$(E_L L)[k, l] = L[j(k), l]$$

$$(\tilde{L} E_L)[k, l] = \tilde{L}[k, i(l)] = L[j(k), j(i(l))] = L[j(k), l]$$

Since $i(\cdot)$ is the inverse function of $j(\cdot)$, $i(l)$ will always be in the domain of $j(\cdot)$, thus

$$E_L L = \tilde{L} E_L$$

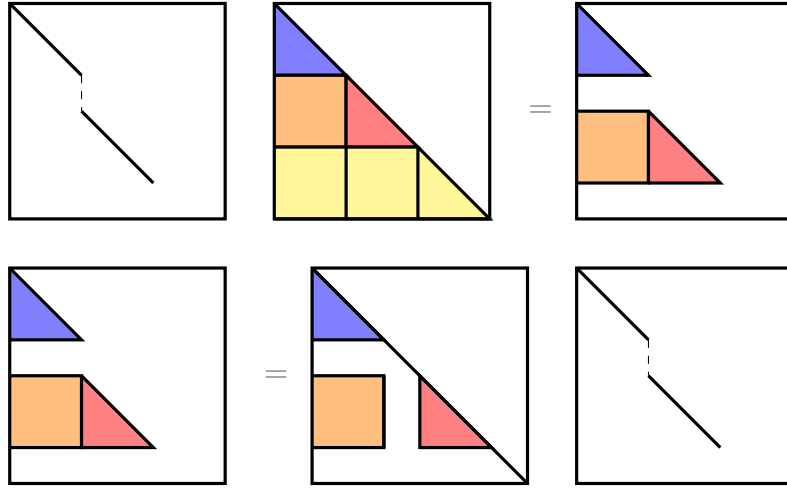


Figure 3.1: Pictorial representation of the shape commutation relationship in Proposition 3.3.1

□

Proposition 3.3.2. *The following shape commutation relations also hold*

$$L\hat{E}_L = \hat{E}_L\tilde{L} \quad \begin{array}{|c|} \hline \color{orange}{\square} \color{blue}{\square} \\ \hline \end{array} = \begin{array}{|c|} \hline \color{blue}{\square} \color{orange}{\square} \\ \hline \end{array} \quad (3.11)$$

$$UE_U = E_U\tilde{U} \quad \begin{array}{|c|} \hline \color{green}{\square} \color{blue}{\square} \\ \hline \end{array} = \begin{array}{|c|} \hline \color{blue}{\square} \color{green}{\square} \\ \hline \end{array} \quad (3.12)$$

$$\hat{E}_U U = \tilde{U} \hat{E}_U \quad \begin{array}{|c|} \hline \color{blue}{\square} \color{green}{\square} \\ \hline \end{array} = \begin{array}{|c|} \hline \color{green}{\square} \color{blue}{\square} \\ \hline \end{array} \quad (3.13)$$

Proof. Taking transpose on the commutation result for E_L

$$(E_L L)^T = (\tilde{L} E_L)^T$$

$$L^T E_L^T = E_L^T \tilde{L}^T$$

Rewriting to denote the shapes we get,

$$UE_U = E_U\tilde{U}$$

Taking the J-Conjugate of the E_L commutation result we have,

$$J(E_L L)J = J(\tilde{L}E_L)J$$

$$JE_L L J = \hat{E}_U J L J = \hat{E}_U U$$

$$J\tilde{L}E_L J = \tilde{U} J E_L J = \tilde{U} \hat{E}_U$$

We get the commutation result for \hat{E}_U

$$\hat{E}_U U = \tilde{U} \hat{E}_U$$

Taking the transpose of the above result, we get,

$$(\hat{E}_U U)^T = (\tilde{U} \hat{E}_U)^T$$

$$U^T \hat{E}_U^T = \hat{E}_U^T \tilde{U}^T$$

Rewriting to denote shapes,

$$L \hat{E}_L = \hat{E}_L \tilde{L}$$

□

3.3.3 Algorithm for Persistence-type Quivers

At a high level, the algorithm will put each matrix in a type-A quiver representation in pivot matrix form. This is accomplished in two passes - we will work from left to right on the first pass, and then right to left on the second pass. Note that we could equally work in the opposite order (see Figure 3.3). We will use diagrams to notate the steps of the algorithm, keeping in mind that we can keep track of the invertible basis transformation for each of the steps. We will first apply the $LE_L U P$ factorization to A_0

$$\begin{array}{ccccccc} \cdot & \xleftarrow{A_0} & \cdot & \xleftarrow{A_1} & \cdots & \xleftarrow{A_{n-1}} & \cdot \\ \cdot & \xleftarrow{L_0 E_0 U_0 P_0} & \cdot & \xleftarrow{A_1} & \cdots & \xleftarrow{A_{n-1}} & \cdot \end{array}$$

We can now use matrix passing to move both U_0 and P_0 matrices, as they are both invertible. We then multiply the matrices to get $\tilde{A}_1 = U_0 P_0 A_1$

$$\begin{array}{c} \cdot \xleftarrow{L_0 E_0} \cdot \xleftarrow{U_0 P_0 A_1} \cdots \xleftarrow{A_{n-1}} \cdot \\ \cdot \xleftarrow{L_0 E_0} \cdot \xleftarrow{\tilde{A}_1} \cdots \xleftarrow{A_{n-1}} \cdot \end{array}$$

We can now apply this procedure to \tilde{A}_1 and then iterate through the rest of the maps in the quiver representation

$$\begin{array}{c} \cdot \xleftarrow{L_0 E_0} \cdot \xleftarrow{L_1 E_1 U_1 P_1} \cdots \xleftarrow{A_{n-1}} \cdot \\ \quad \quad \quad \vdots \quad \quad \quad \vdots \\ \cdot \xleftarrow{L_0 E_0} \cdot \xleftarrow{L_1 E_1} \cdots \xleftarrow{L_{n-1} E_{n-1} U_{n-1} P_{n-1}} \cdot \end{array}$$

Applying matrix passing on the final edge, we can remove the factor $U_{n-1} P_{n-1}$

$$\begin{array}{c} \cdot \xleftarrow{L_0 E_0} \cdot \xleftarrow{L_1 E_1} \cdots \xleftarrow{L_{n-1} E_{n-1} U_{n-1} P_{n-1}} \cdot \\ \cdot \xleftarrow{L_0 E_0} \cdot \xleftarrow{L_1 E_1} \cdots \xleftarrow{L_{n-1} E_{n-1}} \cdot \end{array}$$

Next we can initiate the leftward pass by moving the lower triangular matrices leftward using the shape commutation relations at each step. We do so, as follows,

$$\begin{array}{c} \cdot \xleftarrow{L_0 E_0} \cdots \xleftarrow{L_{n-2} E_{n-2}} \cdot \xleftarrow{L_{n-1} E_{n-1}} \cdot \\ \cdot \xleftarrow{L_0 E_0} \cdots \xleftarrow{L_{n-2} E_{n-2} L_{n-1}} \cdot \xleftarrow{E_{n-1}} \cdot \\ \cdot \xleftarrow{L_0 E_0} \cdots \xleftarrow{\tilde{L}_{n-2} E_{n-2}} \cdot \xleftarrow{E_{n-1}} \cdot \end{array}$$

Here we have used the following shape commutation relation

$$L_{n-2} E_{n-2} L_{n-1} = \tilde{L}_{n-2} E_{n-2}$$

Applying iteratively the the rest of the edges in a right-to-left sweep, we obtain

$$\begin{array}{c} \cdot \xleftarrow{L_0 E_0} \cdots \xleftarrow{\tilde{L}_{n-2} E_{n-2}} \cdot \xleftarrow{E_{n-1}} \cdot \\ \quad \quad \quad \vdots \quad \quad \quad \vdots \\ \cdot \xleftarrow{\tilde{L}_0 E_0} \cdots \xleftarrow{E_{n-2}} \cdot \xleftarrow{E_{n-1}} \cdot \end{array}$$

Finally, we can remove the last factor \tilde{L}_0 , by matrix passing

$$\begin{array}{ccccccc} \cdot & \xleftarrow{\tilde{L}_0 E_0} & \cdots & \xleftarrow{E_{n-2}} & \cdot & \xleftarrow{E_{n-1}} & \cdot \\ \cdot & \xleftarrow{E_0} & \cdots & \xleftarrow{E_{n-2}} & \cdot & \xleftarrow{E_{n-1}} & \cdot \end{array}$$

We have reduced all the matrices to pivot matrix form.

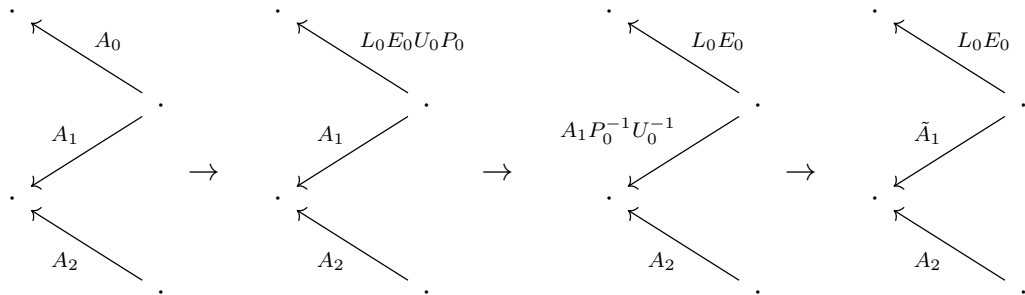
3.3.4 Alternating arrow directions

In the general A-type Quiver diagram, the arrows can be in either direction. So far we have seen an algorithm that works when all the arrows are in the same direction.

To illustrate how the algorithm would work with arbitrary arrow directions, consider the following zigzag quiver.

$$\begin{array}{ccccccc} \cdot & \xleftarrow{A_0} & \cdot & \xrightarrow{A_1} & \cdot & \xleftarrow{A_2} & \cdot \\ \cdot & \xleftarrow{L_0 E_0 U_0 P_0} & \cdot & \xrightarrow{A_1} & \cdot & \xleftarrow{A_2} & \cdot \\ \cdot & \xleftarrow{L_0 E_0} & \cdot & \xrightarrow{A_1 P_0^{-1} U_0^{-1}} & \cdot & \xleftarrow{A_2} & \cdot \\ \cdot & \xleftarrow{L_0 E_0} & \cdot & \xrightarrow{\tilde{A}_1} & \cdot & \xleftarrow{A_2} & \cdot \end{array}$$

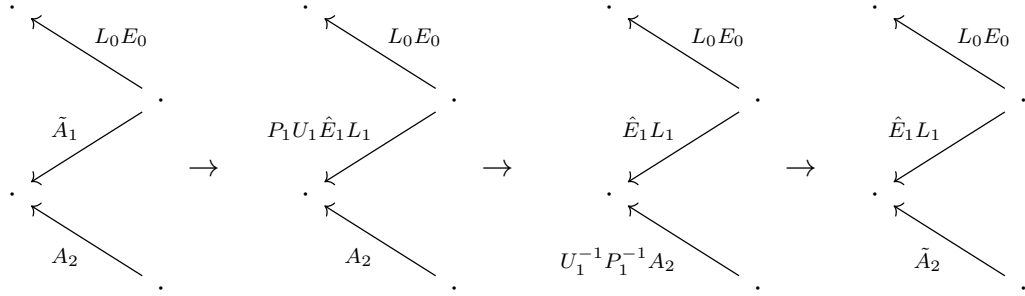
The above transformation is easier to see if the diagram is in the following form:



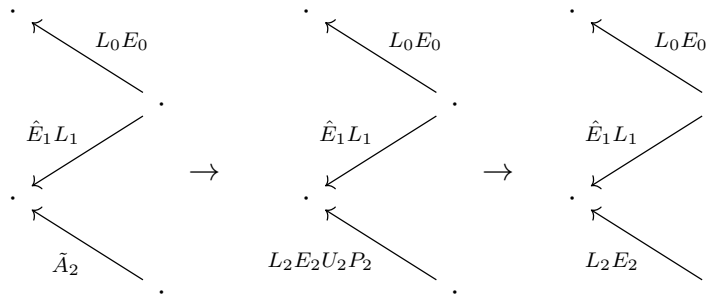
In the next step, as the arrow is reversed we cannot use the LE_LUP factorization. This would result in matrices that cannot be commuted during the second sweep. To handle this case, we use the $PU\hat{E}_L L$ factorization instead.

$$\tilde{A}_1 = P_1 U_1 \hat{E}_1 L_1$$

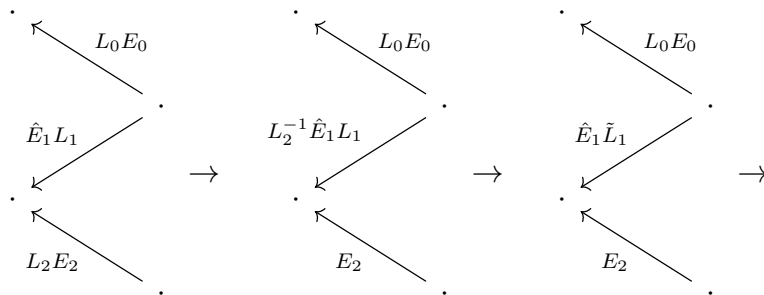
Applying matrix passing to move the factored matrices to the next edge, we get

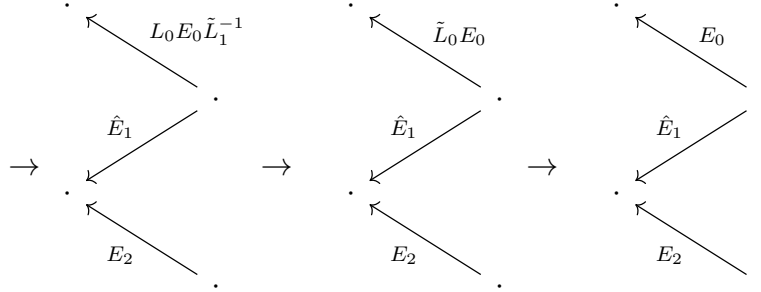


For the last edge, we can again apply the LE_LUP factorization,



Now we can perform the reverse sweep as follows,





During the reverse sweep we used the following commutation relations

$$L_2^{-1} \hat{E}_1 L_1 = \hat{E}_1 \tilde{L}_1$$

$$L_0 E_0 \tilde{L}_1^{-1} = \tilde{L}_0 E_0$$

Thus we have reduced all matrices to pivot matrices, the barcodes can be directly read of from them.

Here we had to use a different factorization depending on the arrow direction, we can now use this to generalize the algorithm to a type-A quiver with arbitrary arrow directions.

3.3.5 General Sequential Algorithm

For arbitrary directions of the arrows, as long as we use the correct factorization for each of the arrow directions, we can use the shape commutation relations in the reverse sweep to reduce all the matrices to echelon pivot form.

We can also initiate the first sweep from the right to the left, to obtain a leftward initial algorithm. All the examples shown above initiate the first sweep from the left. For a general initial sweep direction and arrow direction, The tables Figure 3.2 and Figure 3.3 specify the factorization and commutation relation to use.

We note that the commutation relations established in Section 3.3.2 do not change the nonzero structure of the E matrices. Thus, it is possible to extract the barcode

Algorithm 7 Obtain Barcode factorization of type A quiver: Rightward-initial

- 1: **Input:** Type A quiver representation.
 - 2: **Result:** Barcode form of quiver representation.
 - 3: **for** forward pass **do**
 - 4: **if** \leftarrow **then**
 - 5: Apply LE_LUP Factorization
 - 6: Matrix pass UP factors
 - 7: **else**
 - 8: Apply $PU\hat{E}_LL$ Factorization
 - 9: Matrix pass PU factors
 - 10: **end if**
 - 11: **end for**
 - 12: Matrix pass L factor on the last edge
 - 13: **for** backward pass **do**
 - 14: **if** \leftarrow **then**
 - 15: Commute $L_1E_LL_2 = \tilde{L}_1E_L$
 - 16: **else**
 - 17: Commute $L_1\hat{E}_LL_2 = \hat{E}_L\tilde{L}_2$
 - 18: **end if**
 - 19: **end for**
 - 20: Matrix pass the remaining L factor on the first edge.
-

	Rightward Initial	Leftward Initial
\leftarrow	LE_LUP	PLE_UU
\rightarrow	$PU\hat{E}_LL$	$U\hat{E}_U LP$

Figure 3.2: The factorization to use for the first sweep.

	Rightward Initial	Leftward Initial
\leftarrow	$U_1E_UU_2 = E_U\tilde{U}_2$	$L_1E_LL_2 = \tilde{L}_1E_L$
\rightarrow	$U_1\hat{E}_UU_2 = \tilde{U}_1\hat{E}_U$	$L_1\hat{E}_LL_2 = \hat{E}_L\tilde{L}_2$

Figure 3.3: The commutation to use for the second sweep.

without performing the backward pass of Algorithm 7 if one does not care to form the change of basis.

3.3.6 Parallel Quiver Algorithm

We can also parallelize the algorithm for computing the barcode factorization of a quiver representation using a divide and conquer approach. The protocol of matrix factorizations and matrix passing will be different.

LQU Factorization

The LQU factorization is different from the triangular factorizations introduced before. It does not perform any pivoting and therefore there is no auxiliary permutation matrix produced. Instead, we sacrifice structure in the pivot matrix and obtain a general pivot matrix Q as opposed to echelon pivot matrices.

Proof. In order to see that the above algorithm is correct, we first note that both the row operations and column operations are triangular, i.e. row i is used to eliminate rows at positions greater than i and column j is used to eliminate columns at positions greater than j . Thus the recorded L and U matrices are indeed lower and upper triangular respectively.

Now it is left to prove that the resultant matrix Q , has pivot structure. If we prove that the only non-zeros at the end of the algorithm are the pivots then we are done, as pivots are chosen such that no two of them share a row or column. We will argue by contradiction: suppose there is a non-zero that was not eliminated at the end of the algorithm. It has to be either in a non-pivot row or its column position is before a pivot in a pivot row, otherwise it would have been eliminated by the column operations. Such an element cannot exist as it should have been eliminated by row operations by a pivot above it in the same column. The pivot cannot be below as it would imply that we did not pick the first non-zero in a non-pivot row when choosing the pivot for this column. \square

Algorithm 8 LQU factorization

```
1: Input:  $m \times n$  matrix  $A$ 
2: Result: Factorization  $A = LQU$ 
3:  $L = I_m$ 
4:  $Q = A$ 
5:  $U = I_n$ 
6:  $j = 1$ 
7: while  $j \leq n$  do
8:   if column  $j$  has a non-zero in a non-pivot row, let the first such row be  $i$  then
9:     Zero out all elements in non-pivot rows in column  $j$  below  $i$ 
10:    Record row operations in  $L$ 
11:    Mark  $i$  as pivot row
12:     $j = j + 1$ 
13:  else
14:     $j = j + 1$ 
15:  end if
16: end while
17:  $i = 1$ 
18: while  $i \leq m$  do
19:   if row  $i$  is a pivot-row with pivot at  $j$  then
20:     Zero out all elements after  $j$  in row  $i$ .
21:     Record column operations in  $U$ 
22:      $i = i + 1$ 
23:   else
24:      $i = i + 1$ 
25:   end if
26: end while
```

E Matrix Transformations

We will now see how we can factorize any pivot matrix Q into a permutation and an echelon pivot matrix

Proposition 3.3.3. *Given any pivot matrix Q , we can rewrite it as the following*

$$Q = E_L P \tag{3.14}$$

$$Q = P E_U \tag{3.15}$$

$$Q = \hat{E}_U P \tag{3.16}$$

$$Q = P \hat{E}_L \tag{3.17}$$

$$\tag{3.18}$$

where P is an appropriate permutation matrix.

Proof. We apply the LE_LUP , PLE_UU , $U\hat{E}_ULP$ and $PU\hat{E}_LL$ factorizations to Q and note that the triangular matrices are just the identity matrices. This is because the triangular matrices are produced to eliminate one entry with another entry in the same row or column, but such a situation cannot occur in a pivot matrix Q , so only permutation operations are performed in the factorization, resulting in a permutation matrix and an echelon pivot matrix. \square

Divide and Conquer

Now, we will show how we can divide a type-A quiver into two parts and perform the computation in parallel for each of the parts. The results can then be combined to give the full barcode factorization of the entire quiver. Consider a quiver Q_γ with general arrow directions,

$$\cdot \xleftarrow{A_0} \cdot \xleftarrow{A_1} \cdot \dots \xleftarrow{A_{n-1}} \cdot$$

We will divide it into two parts at position m , to give us two sub-quivers Q_α and Q_β

$$\cdot \xleftarrow{A_0} \cdot \dots \xleftarrow{A_{m-1}} \cdot \xleftarrow{A_m} \cdot \dots \xleftarrow{A_{n-1}} \cdot$$

Next we will introduce three auxiliary edges containing identity maps to aide us in the computations by acting as a place holder for matrices.

$$\cdot \xleftarrow{I} \cdot \xleftarrow{A_0} \cdots \xleftarrow{A_{m-1}} \cdot \xleftarrow{I} \cdot \xleftarrow{A_m} \cdots \xleftarrow{A_{n-1}} \cdot \xleftarrow{I} \cdot$$

We can now perform two versions of the sequential algorithm in parallel. For quiver Q_α , we will use the rightward-initial algorithm (Algorithm 7). Notice that the terminal matrices are collected in the auxiliary edges, they will be important when we merge the results of the two sub-quivers.

$$\cdot \xleftarrow{L_\alpha} \cdot \xleftarrow{E_0} \cdots \xleftarrow{E_{m-1}} \cdot \xleftarrow{U_\alpha P_\alpha} \cdot \xleftarrow{A_m} \cdots \xleftarrow{A_{n-1}} \cdot \xleftarrow{I} \cdot$$

For the quiver Q_β , we will use the leftward-initial sequential algorithm. This allows us to collect both the permutation matrices in the center auxiliary edge.

$$\cdot \xleftarrow{L_\alpha} \cdot \xleftarrow{E_0} \cdots \xleftarrow{E_{m-1}} \cdot \xleftarrow{U_\alpha P_\alpha P_\beta L_\beta} \cdot \xleftarrow{E_m} \cdots \xleftarrow{E_{n-1}} \cdot \xleftarrow{U_\beta} \cdot$$

We can now multiply out the matrices in the center auxiliary edge and perform an LQU factorization using Algorithm 8

$$U_\alpha P_\alpha P_\beta L_\beta = C_\gamma$$

Since all the matrices $U_\alpha, P_\alpha, P_\beta, L_\beta$ are invertible, the pivot matrix Q_γ , will be full rank, and thus turn out to be a permutation matrix P_γ

$$C_\gamma = L_\gamma Q_\gamma U_\gamma = L_\gamma P_\gamma U_\gamma$$

$$\cdot \xleftarrow{L_\alpha} \cdot \xleftarrow{E_0} \cdots \xleftarrow{E_{m-1}} \cdot \xleftarrow{L_\gamma P_\gamma U_\gamma} \cdot \xleftarrow{E_m} \cdots \xleftarrow{E_{n-1}} \cdot \xleftarrow{U_\beta} \cdot$$

The matrices E_0 to E_{m-1} were produced by the rightward-intial algorithm, so they are of shape E_L or \hat{E}_L depending on the arrow directions. They can be used to commute the L_γ factor all the way to the left. This process is very similar to the second sweep of the rightward-initial algorithm. Similarly, the matrices E_m to E_{n-1} are of shape E_U or \hat{E}_U . These matrices can be used to commute the U_γ factor towards the right in a manner similar to the second sweep of the leftward-initial algorithm.

$$\cdot \xleftarrow{L_\alpha \hat{L}_\gamma} \cdot \xleftarrow{E_0} \cdots \xleftarrow{E_{m-1}} \cdot \xleftarrow{P_\gamma} \cdot \xleftarrow{E_m} \cdots \xleftarrow{E_{n-1}} \cdot \xleftarrow{\hat{U}_\gamma U_\beta} \cdot$$

These propagated factors can now be multiplied and we get one lower triangular factor \tilde{L}_γ on the left auxiliary edge and one upper triangular factor \tilde{U}_γ on the right auxiliary edge leaving a permutation matrix P_γ in the center auxiliary edge. At this stage we can think of the result as the ‘‘LQU’’ factorization of the quiver Q_γ

$$\cdot \xleftarrow{\tilde{L}_\gamma} \cdot \xleftarrow{E_0} \cdots \xleftarrow{E_{m-1}} \cdot \xleftarrow{P_\gamma} \cdot \xleftarrow{E_m} \cdots \xleftarrow{E_{n-1}} \cdot \xleftarrow{\tilde{U}_\gamma} \cdot$$

At this stage, we technically have a valid barcode factorization and if this is the entire quiver we could stop here, but since we want to apply this algorithm recursively, we want to convert this into the result of either a leftward-initial or rightward-initial algorithm. This can be done by propagating the permutation matrix P_γ either leftwards or rightwards using the E matrix transformations discussed in Section 3.3.6. If we wish to obtain the result of the rightward-initial algorithm, then we propagate right. Note this transforms the E_U and \hat{E}_U matrices in Q_β to E_L and \hat{E}_L . We will denote the transformed matrices by \tilde{E}_m to \tilde{E}_{n-1} .

$$\cdot \xleftarrow{\tilde{L}_\gamma} \cdot \xleftarrow{E_0} \cdots \xleftarrow{E_{m-1}} \cdot \xleftarrow{I} \cdot \xleftarrow{\tilde{E}_m} \cdots \xleftarrow{\tilde{E}_{n-1}} \cdot \xleftarrow{\tilde{P}_\gamma \tilde{U}_\gamma} \cdot$$

We can now drop the auxiliary identity map in the center, to obtain the rightward-initial merge.

$$\cdot \xleftarrow{\tilde{L}_\gamma} \cdot \xleftarrow{E_0} \cdots \xleftarrow{E_{m-1}} \cdot \xleftarrow{\tilde{E}_m} \cdots \xleftarrow{\tilde{E}_{n-1}} \cdot \xleftarrow{\tilde{P}_\gamma \tilde{U}_\gamma} \cdot$$

Alternatively, we can also propagate the permutation matrix leftwards, to obtain the result of the leftward initial algorithm. This would transform the E_L and \hat{E}_L matrices into E_U and \hat{E}_U in Q_α

$$\begin{aligned} &\cdot \xleftarrow{\tilde{L}_\gamma \tilde{P}_\gamma} \cdot \xleftarrow{\tilde{E}_0} \cdots \xleftarrow{\tilde{E}_{m-1}} \cdot \xleftarrow{I} \cdot \xleftarrow{E_m} \cdots \xleftarrow{E_{n-1}} \cdot \xleftarrow{\tilde{U}_\gamma} \cdot \\ &\cdot \xleftarrow{\tilde{L}_\gamma \tilde{P}_\gamma} \cdot \xleftarrow{\tilde{E}_0} \cdots \xleftarrow{\tilde{E}_{m-1}} \cdot \xleftarrow{E_m} \cdots \xleftarrow{E_{n-1}} \cdot \xleftarrow{\tilde{U}_\gamma} \cdot \end{aligned}$$

Note that the results are not exactly equal to the result you would obtain from applying either a rightward-initial or leftward-initial algorithm on the entire quiver.

While the echelon matrices are of the right shape, the terminal matrices $\tilde{P}_\gamma \tilde{U}_\gamma$ or $\tilde{L}_\gamma \tilde{P}_\gamma$ appear in reversed order. However this does not matter if this is used to merge barcode factorizations at a higher level. At a higher level of the recursion, we would have the following product in the center auxiliary edge,

$$\cdot \xleftarrow{L_\alpha} \cdot \xrightarrow{E_0} \cdots \xrightarrow{E_{m-1}} \cdot \xleftarrow{P_\alpha U_\alpha L_\beta P_\beta} \cdot \xrightarrow{E_m} \cdots \xrightarrow{E_{n-1}} \cdot \xleftarrow{U_\beta} \cdot$$

They can still be multiplied out and the LQU factorization can be performed without affecting the rest of the algorithm.

$$P_\alpha U_\alpha L_\beta P_\beta = C_\gamma$$

$$C_\gamma = L_\gamma P_\gamma U_\gamma$$

Thus we have seen how a divide and conquer algorithm can be used to compute the barcode factorization of a quiver representation in two parts. We can now apply this recursively until the size of the quiver is small enough that it is more efficient to apply one of the sequential algorithms.

3.3.7 Correctness and Uniqueness of the Barcode Factorization

We have presented sequential and divide and conquer parallel algorithms to produce a barcode matrix Λ from the companion matrix A of a finite dimensional quiver representation of type A_n . In this section we'll show the algorithm produces a quiver isomorphism $A \cong \Lambda$, and that Λ uniquely defines the quiver isomorphism class.

Proposition 3.3.4. *Every finite dimensional quiver representation of type A_n has a barcode form.*

Proof. This follows immediately from the existence of the LEUP and PLEU factorizations at each step, and the commutations established. \square

Proposition 3.3.5. *The barcode factorization $A = B\Lambda B^{-1}$ is a quiver isomorphism.*

Proof. We have shown that the factorizations at each step exist, and by Lemma 3.2.8 each matrix passing step is a quiver isomorphism. \square

The following theorem recasts the known fact that a barcode determines the isomorphism class of a zigzag or persistence module in terms of the barcode form.

Theorem 3.3.6. *The barcode form Λ of a quiver representation of type A_n uniquely determines its isomorphism class.*

Proof. This is known to be the case through Gabriel’s theorem [50] and the original work on zigzag homology [22] showing that the barcode is equivalent to interval indecomposables, but we will provide a self-contained proof in our notation for completeness.

If two quiver representations with companion matrices A, A' have the same barcode factorization, then $\Lambda = B^{-1}AB = (B')^{-1}A'B'$, so $A' = B'B^{-1}AB(B')^{-1}$. Thus, the quiver representations are isomorphic via the isomorphism represented by $B'B^{-1}$.

Now, suppose that two quiver representations with companion matrices A, A' are isomorphic, that is, $A' = MAM^{-1}$ for some block-diagonal invertible matrix M . In general, the barcode factorization algorithm will play out very differently on the two companion matrices, with different L, E, U , and P factors at each step, and it isn’t obvious that the intervals recovered by Section 3.2.3 will be the same.

Let $A = BAB^{-1}$ and $A' = B'\Lambda'(B')^{-1}$ be barcode factorizations. Then

$$\Lambda = B^{-1}M^{-1}B' \Lambda' (B')^{-1}MB$$

Let $v \in V_b$ be the basis element for an indecomposable $I[b, d]$ in A , and $v' = Mv$. Note that v' may be a mix of basis elements for indecomposables of A' . We can now trace v through the quiver.

$$\begin{array}{ccccc} V_i & \xleftarrow{A_i} & V_{i+1} & \xleftarrow{A_{i+1}} & \dots \\ \downarrow M_i & & \downarrow M_{i+1} & & \\ V'_i & \xleftarrow{A'_i} & V'_{i+1} & \xleftarrow{A'_i} & \dots \end{array}$$

There are two cases: either $V_i \rightarrow V_{i+1}$, in which case either both the image $A_i v$ and the image $A'_i v'$ are nonzero, or both images must be nonzero (otherwise the diagram

does not commute and M is not a quiver isomorphism). In the other case, $V_{i+1} \rightarrow V_i$, in which case either both v and v' are images of elements in V_{i+1}^* , or both are not images. Thus, since we can propagate v from V_b to V_d , its image v' must propagate from V'_b to V'_d , and if we consider v' in the indecomposable basis, there must be an element for the indecomposable $I[b, d]$. We choose to associate v' with this element. Now, suppose that w is also a basis element for an indecomposable $I[b, d]$ in A , and is associated with the same basis element for $I[b, d]$ in A' . Then $M(v - w)$ would have a zero coefficient for an element of $I[a, b]$, and could not be propagated the full length of the bar in A' , so we must have $v = w$ for the diagram to commute. Thus, the indecomposables of A map injectively to the indecomposables of A' .

We can then use an identical argument to see that the inverse map M^{-1} maps the indecomposables of A' injectively to indecomposables of A . Thus, there is a bijection between the indecomposables, and the barcode forms must be identical (up to permutation). \square

Theorem 3.3.6 is equivalent to the statement that the indecomposables of A_n -type quivers are interval indecomposables. Note that the uniqueness of the Jordan normal form for companion matrices is a special case of Theorem 3.3.6 that applies to persistence-type quivers, following Section 3.2.4.

3.4 Experiments

We'll now investigate several experiments which demonstrate the flexibility of our methodology. Several preliminary timing results are presented based on computations done on an Intel i7-6820HQ CPU capable of running 8 threads concurrently.

3.4.1 Subsets and Rips Complexes

Let $\mathbf{X}_0, \mathbf{X}_1, \dots \subseteq X$ be finite subsets of a space, and $r_0, r_1, \dots \in \mathbb{R}$ be parameters for the Rips construction. We can construct a zigzag through unions of subsets

$$\mathcal{R}(\mathbf{X}_0; r_0) \hookrightarrow \mathcal{R}(\mathbf{X}_0 \cup \mathbf{X}_1, \max\{r_0, r_1\}) \longleftarrow \mathcal{R}(\mathbf{X}_1; r_1) \hookrightarrow \dots \quad (3.19)$$

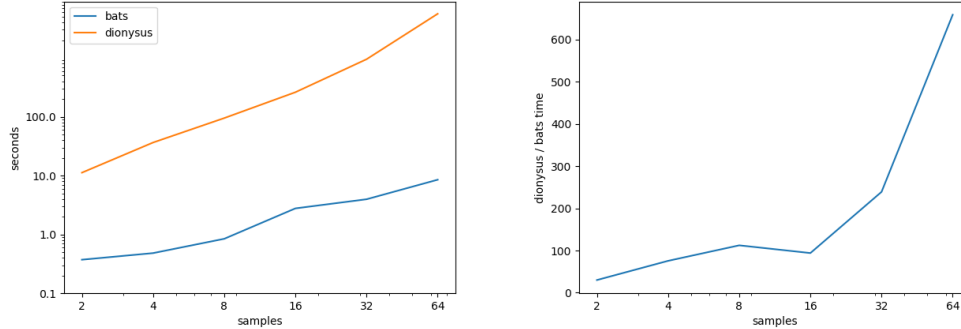


Figure 3.4: Time to compute zigzag homology of a zigzag diagram computed on subsamples of a noisy circle. Normal noise with variance 0.1 is added to points sampled from a unit circle. 200 points are contained in each subsample, and $r = 0.35$. The horizontal axis indicates the number of samples used in the diagram. Left: time to compute zigzag homology for both BATS and Dionysus. Right: Speedup seen using BATS instead of Dionysus. At the right hand side, BATS is over 600x faster.

The case where $r = r_0 = r_1 = \dots$ was proposed in [22] as a “topological bootstrapping” method to investigate the robustness of homological features for samples \mathbf{X}_i sampled uniformly from some larger point cloud \mathbf{X} , and initial investigations were performed by Tausz [98]. A single value of r is also used in the case of [23] where the \mathbf{X}_i come from thickened level sets of a map $f : \mathbf{X} \rightarrow \mathbb{R}$

$$\mathbf{X}_i = f^{-1}((a_i, b_i))$$

where $(a_i, b_i) \cap (a_{i+1}, b_{i+1}) \neq \emptyset$.

As seen in Figure 3.4, the implementation in BATS displays a large performance advantage over Dionysus for subsampled Rips constructions, especially as the length of the diagram grows.

Another application of a zigzag of this form is to approximate the persistent homology of the full Rips filtration. This was originally implemented by Morozov in his Dionysus software [79], and several variants with theoretical guarantees are investigated in [85]. In this situation, $\mathbf{X}_0 \subseteq \mathbf{X}_1 \subseteq \dots$, and $r_0 \geq r_1 \geq \dots$. An example can be found in Figure 3.5. At the time of this writing, the computational bottleneck in this experiment is in a dense linear algebra implementation of the quiver

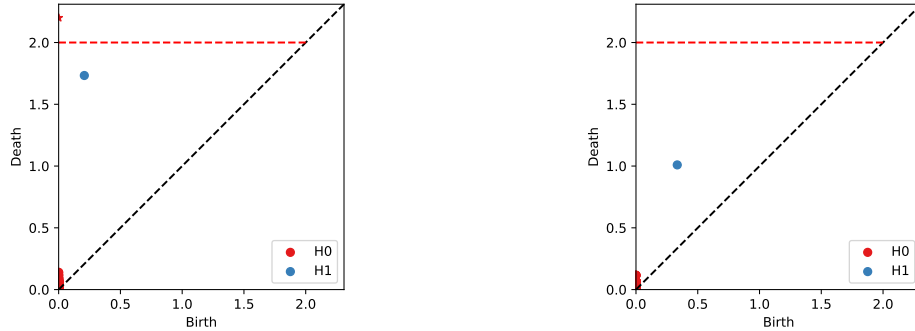


Figure 3.5: Left: persistence diagram for Rips filtration on 200 points sampled from the unit circle. The unoptimized reduction algorithm runs in 20 seconds in BATS. Right: An approximate persistence diagram created using the discrete Morozov zigzag construction in [85] using the suggested parameters. The zigzag computation takes approximately 0.5 seconds in BATS. While the birth and death times are not identical, both diagrams qualitatively display the same information, namely a single connected component and a robust H_1 class, agreeing with the homology of the circle.

algorithm, which grows cubically with the size of the point cloud. This is due to the large H_0 dimension at the end of the diagram, and we expect performance will improve significantly if replaced by a sparse computation.

3.4.2 Bivariate Nerve

Another application of zigzag homology is to investigate the relationship between algebraic features in nerves of two or more covers. This question arises naturally when generating covers algorithmically, where one might wonder how sensitive the Nerve is to choices that might be made or randomness in initialization. A related question is the stability of witness complexes to the choice of landmark set, and a *bivariate witness* complex was proposed in [22] and subsequently investigated by Tausz [98].

Definition 3.4.1. *Given two covers \mathcal{U}, \mathcal{V} of a space X , the bivariate cover is the fibered product $\mathcal{U} \times_X \mathcal{V} \subseteq \mathcal{U} \times \mathcal{V}$. Explicitly,*

$$\mathcal{U} \times_X \mathcal{V} = \{U \times V \mid U \in \mathcal{U}, V \in \mathcal{V}, U \cap V \neq \emptyset\}$$

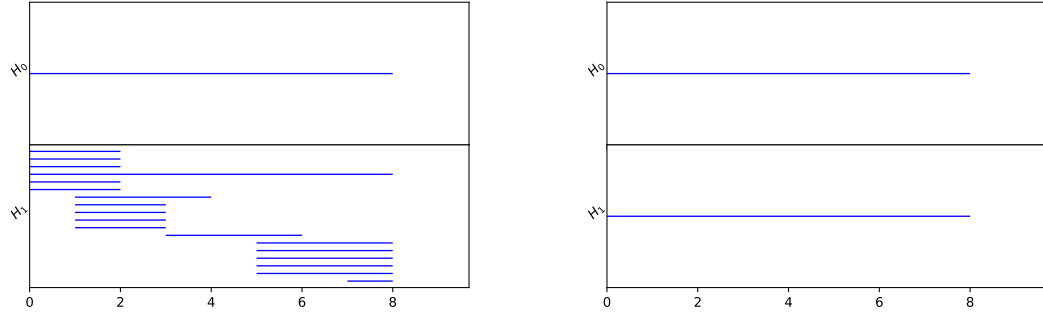


Figure 3.6: Zigzag barcodes of bivariate Nerve diagram on 5 covers of 500 points on the unit circle. Covers are computed by selecting 20 random landmarks. Left: each point is assigned to closest 2 landmarks. Right: each point is assigned to closest 3 landmarks. Note that both diagrams have single long bars in dimensions 0 and 1, agreeing with the homology of the circle.

and $\mathcal{U} \times \mathcal{V}$ can be identified as the intersection $U \cap V$ to form another cover of X . We will denote the nerve of $\mathcal{U} \times_X \mathcal{V}$ as $\mathcal{N}(\mathcal{U}, \mathcal{V})$.

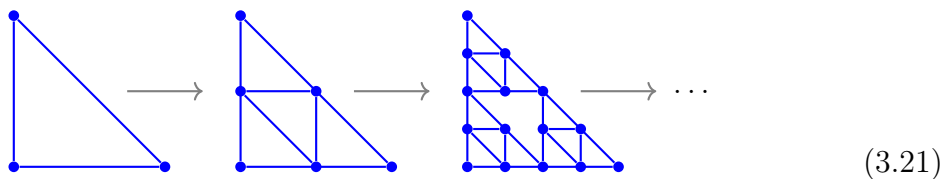
Due to the product structure on $\mathcal{U} \times \mathcal{V}$, there are projection maps $p_U : \mathcal{U} \times \mathcal{V} \mapsto \mathcal{U}$ and $p_V : \mathcal{U} \times \mathcal{V} \mapsto \mathcal{V}$. These maps extend to simplicial maps on the nerves:

$$\mathcal{N}(\mathcal{U}) \xleftarrow{p_U} \mathcal{N}(\mathcal{U}, \mathcal{V}) \xrightarrow{p_V} \mathcal{N}(\mathcal{V}) \tag{3.20}$$

We will consider covers based on landmark sets, as described in Section 4.4.4. Briefly, we choose a landmark set $\mathbf{L} \subset \mathbf{X}$, and assign each point in \mathbf{X} to the k -nearest landmarks. An example is performed in Figure 3.6, where covers of a noisy circle are created. We see that if each point belongs to only 2 sets that while there is a single long H_1 class, there are also many short-lived H_1 classes since there are no non-empty 3-way intersections to fill in small holes. However, if every point is assigned to 3 sets, then there is a single H_1 class that persists the length of the diagram.

3.4.3 Sierpinski Triangle

Our final application is an example of how persistent homology may be used with more general cell maps. We'll consider a sequence of spaces converging to a Sierpinski triangle:



Where each map above sends vertices to the vertex in the same location in the image to the right. Then each edge is subdivided in the image, introducing a new node and two new edges, and additional edges are introduced to fill in new interior triangles.

The maps are not simplicial because each edge is subdivided at each iteration. One could also consider a filtration of simplicial complexes based on the finest Sierpinski mesh, but the advantage of the mapping construction is that it is easy to add another iteration of the subdivision to the end of the sequence without recomputing every space. We compute the persistence barcode in Figure 3.7. There is a single connected component, and the k th iteration (starting at $k = 0$) adds another $3^k H_1$ classes.

This is a fairly simple example in which the maps are easy to specify. In general, it can be difficult to specify arbitrary cell maps, and in Section 4.1 we will see one potential algorithmic approach to generating cell maps through solving an extension problem which could potentially simplify the problem for practical applications. An implementation is beyond the scope of this work, but our methodology opens up the possibility of using these constructions and general cellular maps in future work.

3.5 Conclusion & Future Directions

We have presented a matrix factorization method for computing the indecomposables (barcodes) of finite type-A quiver representations. A natural question that arises is whether similar algorithms might exist for other types of quiver representations, which

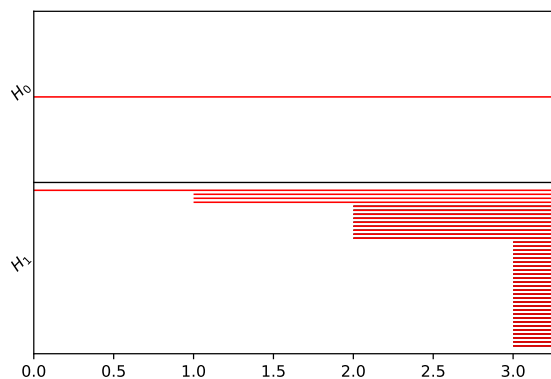


Figure 3.7: Persistence barcode showing 4 iterations of the sequence in Equation (3.21).

then might be used to extract information about different diagrams of spaces, as in Section 3.2.2. Representations on the other Dynkin diagrams almost certainly are amenable to similar techniques, as a result of Gabriel’s theorem [50], but it would be necessary to go beyond sequential application of LU factorizations. However, it is questionable whether type-D and type-E quiver representations would find applications in applied topology. A more promising direction is to develop general algorithms for type- \tilde{A} representations, whose underlying graphs discretize a circle. Burghelea [19] has shown topologically interesting applications, and in collaboration with Dey [20] developed an algorithm that works in certain limited situations.

It is generally much more difficult to compute invariants of wild-type quiver representations, as has been seen in multi-parameter persistence [26]. However, one might hope that for certain *specialized* situations, the problem becomes more tractable. This could be through choosing a specific field \mathbb{F} to work with, or by considering situations in which induced maps on homology satisfy certain properties.

We have considered using arbitrary induced maps on homology in our quiver representations. In the special case of inclusion maps, enormous compression can be achieved by working at the chain level, as seen in our analysis of Algorithm 1. One might ask whether there are other situations in which such compression is achievable.

Finally, we have worked exclusively with homology, whereas cohomology has additional structure in the form of various cohomology operations [80]. An interesting question is whether these cohomology operations could potentially aid in accelerating computations when computing barcodes for all dimensions simultaneously, both at the chain level, and at the level of induced maps.

Chapter 4

Extensions of Maps and Interleavings

So far, we have discussed a computational framework for computing algebraic invariants of diagrams of topological spaces. However, in order to effectively use this computational tool in practical situations we need to be able to

1. Create diagrams of spaces from data.
2. Analyze how the algebraic output is affected by perturbation or approximation.

These two topics are related through the notion of interleavings. In this chapter, we develop a general-purpose framework for generating maps between filtered spaces using carriers. Section 4.1 will introduce the classical carrier construction, and Section 4.2 will introduce the use of interleavings to compare persistent homology of two filtered spaces. In Section 4.3, we will extend the classical notion of carrier to the filtered setting and use this to prove several results on geometric complexes concerning *stability* of persistent homology to perturbations of underlying data, and comparing the persistent homology of different geometric constructions. Finally, in Section 4.4, we will introduce complexes parameterized by a cover, and use filtered carriers to examine stability of these complexes and compare these complexes to other geometric constructions.

4.1 Carriers and Extensions of Maps

We have seen how simplicial maps are completely defined by initial data on the vertex set in Equation 2.9. This makes simplicial maps easy to define and use in practice, but unfortunately does not meet all needs. Fortunately, a general-purpose tool exists, based on carriers, which allows for extensions of maps outside the simplicial setting, and which requires the same, or even weaker initial data.

The extension problem in topology can be stated as follows: given a subset $A \subset X$, and a map $f : A \rightarrow Y$, when can f be extended to a continuous map $f^* : X \rightarrow Y$ so that $f^*|_A = f$? The earliest general result in this direction is the Tietze extension theorem [100], which gives an affirmative answer for closed subsets A of a metric space X mapped to the real line. By 1940, a variety of results in this direction had been found, as discussed in [47], which gives the negative example: The identity mapping $S^n \hookrightarrow S^n$ has no extension to $B^{n+1} \rightarrow S^n$.

Carriers are a general-purpose tool in algebraic topology, which, for instance, are useful for establishing equivalence of different homology theories. See [49, 81] for introductions in this context, and [80] for an equivariant application. The notion of “carrying” cells dates back at least to the development of homology of simplicial complexes [4]. The use of carriers in algebraic topology has been largely superseded by acyclic models [48], but as we’ll see the concept of carrier will translate nicely to explicit algorithms, and to the filtered setting. This section will develop the classical theory of carriers from an algorithmic viewpoint, in the spirit of Chapter 2.

Because our applications are to finite simplicial complexes built from data, throughout this section we will assume that complexes are simplicial unless otherwise noted, and that filtrations can be described using a discrete subset of the real numbers.

4.1.1 Carriers

We’ll begin by defining parallel notions of ‘carrying’ for both cell complexes and chain complexes.

Definition 4.1.1. *Let $A \subseteq X$ be a sub-cell complex. A cell $x \in X$ is carried by A if $x \in A$.*

The similar notion for chain complexes is

Definition 4.1.2. Let $L_* \subseteq C_*$ be a sub-chain complex. A chain $x \in C_*$ is carried by L_* if x is zero everywhere outside of L_* in C_* .

Definition 4.1.3. A carrier $\mathcal{C} : \mathcal{X} \rightarrow \mathcal{Y}$ is a map between cells $x \in \mathcal{X}$ and sub-cell complexes $\mathcal{C}(x) \subseteq \mathcal{Y}$ so that if $z \subset x$ is a face, then $\mathcal{C}(z) \subseteq \mathcal{C}(x)$.

and for chain complexes,

Definition 4.1.4. A carrier $\mathcal{C} : C_* \rightarrow D_*$ between chain complexes is a map between elements of a (chosen) basis B_* of C_* and sub-chain complexes of D_* . We denote the subcomplex associated to a basis element $b \in B_*$ as $\mathcal{C}(b) \subseteq D_*$. The carrier must have the property that if b' has a nonzero coefficient in ∂b , then $\mathcal{C}(b') \subseteq \mathcal{C}(b)$. We say the carrier \mathcal{C} is proper with respect to a basis B_*^D of D_* if $\mathcal{C}(x)$ is generated by a sub-basis of B_*^D for each $x \in B_*$.

In this section, we will always use B_* to denote the basis of C_* used by the carrier. Our chain complexes will always arise from cell complexes, so we will take the basis B_* to be given by the cell basis. In this situation, the chain functor preserves carriers, so we will use \mathcal{C} to denote the carrier on cell complexes and the associated carrier on chain complexes interchangeably, with the meaning determined by context. Note that carriers of cell complexes always produce carriers of chain complexes that are proper with respect to the cell basis.

Definition 4.1.5. Let $f : \mathcal{X} \rightarrow \mathcal{Y}$ be a cellular map, and $\mathcal{C} : \mathcal{X} \rightarrow \mathcal{Y}$ be a carrier. We say f is carried by \mathcal{C} if $f(x) \subseteq \mathcal{C}(x)$ for all $x \in \mathcal{X}$.

Definition 4.1.6. Let C_*, D_* be chain complexes, $\mathcal{C} : C_* \rightarrow D_*$ be a carrier, and $F_k : C_k \rightarrow D_\ell$ be a map. We say F_k is carried by \mathcal{C} if $F_k(b) \in \mathcal{C}(b)$ for all $b \in B_k \subset C_k$.

Using this definition, we can talk about both chain maps and chain homotopies being carried by \mathcal{C} .

Definition 4.1.7. We say a carrier \mathcal{C} of chain complexes is acyclic if $\mathcal{C}(b)$ is acyclic (Definition 2.4.4) for every $b \in B_*$.

We say a carrier of cell complexes $\mathcal{C} : X \rightarrow Y$ is acyclic if the associated carrier of chain complexes is acyclic.

Theorem 4.1.8. (*Acyclic carrier theorem*) *If $\mathcal{C} : C_* \rightarrow D_*$ is acyclic, and $L_* \subset C_*$ is a sub-chain complex of C_* , then any chain map $\hat{F}_* : L_* \rightarrow D_*$ can be extended to a chain map $F_* : C_* \rightarrow D_*$. Furthermore, this extension is unique up to chain homotopy.*

Proofs can be found in [49, 80, 81]. A similar proof will be given for Proposition 4.1.10. Note that because the extension is unique up to chain homotopy, the induced maps on homology will be isomorphic.

To see why Theorem 4.1.8 is so useful from a practical perspective, we can take $L_* = C_0$ (i.e. the zero-chains), and extend to a full chain map, just as we are able to do for simplicial maps, but without restrictions on the map (or spaces) being simplicial. This is particularly useful in the context of point-cloud data, where we typically will begin with a map between points.

Proposition 4.1.9. *A carrier between cell complexes $\mathcal{C} : \mathcal{X} \rightarrow \mathcal{Y}$ is acyclic if $\mathcal{C}(x)$ is contractible for each $x \in \mathcal{X}$.*

This follows readily from the fact that contractible spaces have trivial homology, corollary 2.4.6..

Proposition 4.1.10. *Let C_*, D_* be chain complexes. Let $F_*, G_* : C_* \rightarrow D_*$ be chain maps carried by an acyclic carrier $\mathcal{C} : C_* \rightarrow D_*$, and $H : C_0 \rightarrow D_1$ be a homotopy of F_* and G_* on 0-chains carried by \mathcal{C} . That is $\partial_1 H(x) = F(x) - G(x)$ and $H(x) \in \mathcal{C}(x)$. Then H can be extended to a full chain homotopy H_* of F_* and G_* carried by \mathcal{C} .*

Proof. As usual, we will use B to denote the basis for the carrier. Proof is by induction on dimension of the map. Let $H_0 = H$. We have that $\partial_1 H_0 = F_0 - G_0$ (note $\partial_0 = 0$), so H is a homotopy of 0-chains.

Now, suppose that we have extended H to H_i , carried by \mathcal{C} so that

$$H_{i-1}\partial_i + \partial_{i+1}H_i = F_i - G_i$$

and $H_i(x) \in \mathcal{C}(x)$ for all $x \in B_i$. Note that

$$\begin{aligned}
\partial_{i+1}[F_{i+1} - G_{i+1} - H_i\partial_{i+1}] &= \partial_{i+1}F_{i+1} - \partial_{i+1}G_{i+1} - \partial_{i+1}H_i\partial_{i+1} \\
&= F_i\partial_{i+1} - G_i\partial_{i+1} - \partial_{i+1}H_i\partial_{i+1} \\
&= F_i\partial_{i+1} - G_i\partial_{i+1} - [F_i - G_i - H_{i-1}\partial_i]\partial_{i+1} \\
&= H_{i-1}\partial_i\partial_{i+1} \\
&= 0
\end{aligned}$$

Let $x \in B_{i+1}$ be a basis element, with boundary $\partial_{i+1}x = \sum_j \alpha_j y_j$ where $y_j \in B_i$ is in the carrier basis. Because F_{i+1}, G_{i+1} and H_i are all carried by \mathcal{C} , we have that $(F_{i+1} - G_{i+1} - H_i\partial_{i+1})(x) \in \mathcal{C}(x)$, and from the above $\partial_{i+1}(F_{i+1} - G_{i+1} - H_i\partial_{i+1})(x) = 0$, so it is a cycle. Because the carrier is acyclic, this cycle must be a boundary of an $(i+2)$ -chain in $\mathcal{C}(x)$, which we will denote $H_{i+1}(x)$. Thus,

$$\partial_{i+2}H_{i+1}(x) + H_i\partial_{i+1}(x) = F_{i+1}(x) - G_{i+1}(x)$$

By choosing such a chain for each $x \in B_{i+1}$, we have extended the chain homotopy to dimension $(i+1)$ in a way that is carried by \mathcal{C} . By induction on dimension, we construct a chain homotopy H_* between F_* and G_* . \square

Maps carried by the same acyclic carrier are typically homotopic to similar maps, as in the following example.

Proposition 4.1.11. *Suppose $F_*, G_* : C_* \rightarrow D_*$ are augmentation-preserving chain maps carried by an acyclic carrier \mathcal{C} . Then F_* and G_* are chain-homotopic, and thus produce identical induced maps on homology.*

Proof. For each basis element $x \in C_0$, $F_0(x), G_0(x) \in \mathcal{C}(x)$, and because F_* and G_* are augmentation preserving, $\epsilon x = \epsilon F(x) = \epsilon G(x)$, so $\epsilon(F(x) - G(x)) = 0$. Because \mathcal{C} is acyclic, $\ker \epsilon = \text{img } \partial_1$, so there must exist a 1-chain $H(x) \in \mathcal{C}(x)$ so that $\partial_1 H(x) = F(x) - G(x)$, which is a homotopy of zero-chains. We can then apply 4.1.10 to extend this to a homotopy $H_* : F_* \rightarrow G_*$. \square

Note, that as discussed in Section 2.3.1, chain maps obtained through the cellular chain functor are augmentation preserving. This leads to the following corollary:

Corollary 4.1.12. *Let $f, g : \mathcal{X} \rightarrow \mathcal{Y}$ be cellular maps carried by an acyclic carrier $\mathcal{C} : \mathcal{X} \rightarrow \mathcal{Y}$. Then f and g induce isomorphic maps on homology.*

Furthermore, just the existence of an acyclic carrier implies the existence of an augmentation-preserving chain map.

Proposition 4.1.13. *Let $\mathcal{C} : C_* \rightarrow D_*$ be an acyclic carrier that is proper with respect to a basis B_*^D of D_* . Then there exists a canonical augmentation-preserving chain map $F_* : C_* \rightarrow D_*$ carried by \mathcal{C} .*

Proof. For each $x \in B_0$, we simply assign $F_0(x) = y$ for some $y \in B_0^D \mid_{\mathcal{C}(x)}$. Such a y must exist because $\mathcal{C}(x)$ is non-empty and proper with respect to B_*^D . We can then extend this map to higher dimensions using Theorem 4.1.8. This map preserves the canonical augmentation (Definition 2.3.1) on both complexes because it sends basis elements of 0-chains to basis elements of 0-chains. \square

Proposition 4.1.11 shows that this map is unique up to chain homotopy. We'll now turn to specializations of carriers for simplicial complexes.

Definition 4.1.14. *Let \mathcal{X} and \mathcal{Y} be simplicial complexes. We say a carrier $\mathcal{C} : \mathcal{X} \rightarrow \mathcal{Y}$ is simplicial if $\mathcal{C}(x)$ is a simplex for all $x \in \mathcal{X}$.*

Note that simplicial carriers are acyclic because simplices are contractible.

Definition 4.1.15. *Let $f, g : \mathcal{X} \rightarrow \mathcal{Y}$ be simplicial maps. We say f and g are contiguous if for each simplex $(x_0, \dots, x_k) \in \mathcal{X}$, the (possibly degenerate) set of points*

$$f(x_0), \dots, f(x_k), g(x_0), \dots, g(x_k)$$

span a simplex in \mathcal{Y} .

Given contiguous simplicial maps $f, g : \mathcal{X} \rightarrow \mathcal{Y}$, we can define a simplicial carrier $\mathcal{C} : \mathcal{X} \rightarrow \mathcal{Y}$ via

$$\mathcal{C}(x_0, \dots, x_k) = \langle f(x_0), \dots, f(x_k), g(x_0), \dots, g(x_k) \rangle \quad (4.1)$$

The carrier is simplicial by definition of contiguous simplicial map. It follows immediately from corollary 4.1.12 that f and g induce isomorphic maps on homology. An alternative proof can be found in [81] 1.12.

4.1.2 Algorithmic Extensions of Maps

In the preceding section, we established some useful theoretical properties of acyclic carriers. In this section, we will provide explicit algorithms for extending maps using carriers in the flavor of Chapter 2.

The key observation is that if a carrier is acyclic, then all cycles in $\mathcal{C}(\sigma)$ are boundaries. This means that if a vector $y \in \mathcal{C}(\sigma)_k$ is a cycle, then the linear system

$$y = \partial_{k+1}^{\mathcal{C}(\sigma)} z \quad (4.2)$$

is consistent, meaning a solution z exists.

Algorithm 9 Extension of Map Using Carrier

- 1: **Input:** basis element $x \in C_{k+1}$. Chain map extended up to dimension k : F_k , and carrier $\mathcal{C}(x) \subseteq D_*$
 - 2: **Result:** Extension of $F_* : C_* \rightarrow D_*$ by computing $F_{k+1}x$
 - 3: compute $y = F_k \partial_{k+1}^{\mathcal{C}} x$
 - 4: compute $R_{k+1} U_{k+1} = \partial_{k+1}^{\mathcal{C}(x)}$
 - 5: solve $y = R_{k+1} w$
 - 6: solve $w = U_{k+1} z$
 - 7: return z
-

Note that because U_{k+1} is invertible, that the only place where the algorithm may fail is in Line 5, if y is not in the column space of R_{k+1} (i.e. y is not a boundary). If y is always a boundary, then the algorithm will terminate successfully.

Proposition 4.1.16. *If Algorithm 9 terminates successfully, then the resulting map is valid chain map.*

Proof. By construction, the maps commute with the boundary operators. □

Remark 4.1.17. *Algorithm 9 may succeed even when the carriers are not acyclic. If this is the case, the map is a valid chain map, but is not unique up to homotopy, so the induced map on homology will generally not be unique.*

Proposition 4.1.18. *If the carrier \mathcal{C} is acyclic, then Algorithm 9 will terminate successfully.*

Proof. Because $y = F_k \partial_{k+1}^{\mathcal{C}} x$ is a cycle, and \mathcal{C} is acyclic, the linear system in Line 5 is consistent. This holds for every $x \in B \subset C_{k+1}$. \square

Remark 4.1.19. *This means the construction of an acyclic carrier has two nice properties:*

1. *Algorithm 9 is guaranteed to successfully extend a chain map*
2. *That extension is unique up to homotopy, so while the algorithm makes a particular choice of extension, any other valid choice would produce the same induced map on homology.*

4.1.3 From Relations to Maps

We will now develop an approach to obtaining carriers (and therefore maps) from complexes built from data.

Definition 4.1.20. *A binary relation (or simply, relation) between two sets X and Y is a subset $\Omega \subseteq X \times Y$.*

With a slight abuse of notation, we will use the following shorthand

$$\Omega(x) = \{y \mid (x, y) \in \Omega\}, \quad \Omega(y) = \{x \mid (x, y) \in \Omega\} \quad (4.3)$$

And more generally,

$$\Omega(x_0, \dots, x_k) = \bigcup_{i=0}^k \Omega(x_i), \quad \Omega(y_0, \dots, y_\ell) = \bigcup_{j=0}^{\ell} \Omega(y_j) \quad (4.4)$$

Following [64], we'll say a relation is *left total* if for all $x \in X$, $\Omega(x)$ is not empty, and a relation is *right total* if for all $y \in Y$, $\Omega(y)$ is not empty.

Definition 4.1.21. *A binary relation $\Omega \subseteq X \times Y$ is a correspondence if it is both left-total and right-total. We will use $\mathcal{C}(X, Y)$ to denote the set of all correspondences on X and Y .*

In practice, binary relations will arise naturally by identifying points in X with points in Y in some way, for instance by taking all points within a certain distance or with high enough similarity.

Definition 4.1.22. *Let $f : X \times Y \rightarrow \mathbb{R}$. The sublevelset relation of f at level r is the relation*

$$\Omega_{f,r} = \{(x, y) \in X \times Y \mid f(x, y) \leq r\} \quad (4.5)$$

We can obtain carriers from relations in the following way

Definition 4.1.23. *Let $\Omega \subset X \times Y$ be a left-total binary relation. Let \mathcal{X} and \mathcal{Y} be cell complexes with zero-skeleta $\mathcal{X}_0 = X$ and $\mathcal{Y}_0 = Y$. Then we can define a carrier $\mathcal{C}_\Omega : \mathcal{X} \rightarrow \mathcal{Y}$ via*

$$\mathcal{C}_\Omega(\sigma) = \langle \Omega(\sigma_0) \rangle \quad (4.6)$$

where σ_0 is the zero-skeleton of the cell σ .

Note that \mathcal{C}_Ω satisfies Definition 4.1.3 because if $\tau \subset \sigma$ is a face, then $\tau_0 \subseteq \sigma_0$, and so $\mathcal{C}_\Omega(\tau) \subseteq \mathcal{C}_\Omega(\sigma)$. We'll say that the relation Ω is acyclic if \mathcal{C}_Ω is acyclic. We can similarly define a carrier $\mathcal{D}_\Omega : \mathcal{Y} \rightarrow \mathcal{X}$ if Ω is right-total.

A natural way to define a chain map $F_* : C_*(\mathcal{X}) \rightarrow C_*(\mathcal{Y})$ given a (left-total) relation Ω is to start with a map of points $F_* : x \mapsto y \in \Omega(x)$ that is carried by Ω , and then extending to higher-order cells using the carrier. Applying Proposition 4.1.11, we see that the choice of point-wise map does not matter when computing induced maps on homology when Ω is acyclic:

Corollary 4.1.24. *(to Proposition 4.1.11) Let Ω be an acyclic relation and F_*, G_* be extensions of vertex maps carried by \mathcal{C}_Ω . Then F_* and G_* are chain homotopic.*

Corollary 4.1.25. *Suppose that $X \subset Y$, and $\mathcal{X} \subseteq \mathcal{Y}$, and there is an acyclic relation Ω carrying the inclusion map, i.e. $x \in \Omega(x)$. Then any vertex map carried by \mathcal{C}_Ω is chain homotopic to the inclusion map.*

Proof. This is a special case of corollary 4.1.24, where $f = \iota$, the inclusion map. \square

4.1.4 Composition of Carriers

Carriers of cell complexes can be composed in a well-defined way. If $\mathcal{C} : \mathcal{X} \rightarrow \mathcal{Y}$ and $\mathcal{D} : \mathcal{Y} \rightarrow \mathcal{Z}$ are carriers, then we can define $\mathcal{D} \circ \mathcal{C} : \mathcal{X} \rightarrow \mathcal{Z}$

$$\mathcal{D} \circ \mathcal{C}(x) = \bigcup_{y \in \mathcal{C}(x)} \mathcal{D}(y) \quad (4.7)$$

Note that composition of carriers of chain complexes may not be well-defined, since if $\mathcal{C} : C_* \rightarrow D_*$ is a carrier the sub-chain complex $\mathcal{C}(b) \subseteq D_*$ need not admit a basis that is a sub-basis of D_* . In the case where $\mathcal{C}(b)$ proper, then Equation (4.7) can be applied. This is the case when the chain carriers are obtained from cell carriers.

Remark 4.1.26. *Composition of acyclic carriers need not be acyclic. For example, we can define a cell carrier from the cycle graph on 4 edges (the boundary of a square) to itself, sending a vertex to its star, and an edge to the union of the two stars of its boundary vertices. This carrier is acyclic, but if we compose the carrier with itself and apply to any cell, the composite carrier is the whole space, which has a cycle.*

In certain situations, acyclic carriers can have stronger properties.

Proposition 4.1.27. *Let $\mathcal{C} : \mathcal{X} \rightarrow \mathcal{Y}$ be a simplicial carrier, and $\mathcal{D} : \mathcal{Y} \rightarrow \mathcal{Z}$ be an acyclic carrier. Then $\mathcal{D} \circ \mathcal{C}$ is acyclic.*

Proof. Let $x \in \mathcal{X}$. Then $\mathcal{C}(x) = y$ is a simplex in \mathcal{Y} , whose carrier $\mathcal{D}(y)$ is acyclic. Thus, $\mathcal{D} \circ \mathcal{C}$ is acyclic for all $x \in \mathcal{X}$. \square

Corollary 4.1.28. *If $\mathcal{C} : \mathcal{X} \rightarrow \mathcal{Y}$ and $\mathcal{D} : \mathcal{Y} \rightarrow \mathcal{Z}$ are simplicial carriers (thus acyclic), then $\mathcal{D} \circ \mathcal{C}$ is simplicial (thus acyclic).*

Proof. Let $x \in \mathcal{X}$. Because \mathcal{C} is simplicial, $\mathcal{C}(x) = y$ is a simplex in \mathcal{Y} , and because \mathcal{D} is simplicial, $\mathcal{D}(y)$ is a simplex in \mathcal{Z} . Thus, $\mathcal{D} \circ \mathcal{C}(x) = \mathcal{D}(y)$ is a simplex in \mathcal{Z} , and because this holds for all simplices $x \in \mathcal{X}$, the composite $\mathcal{D} \circ \mathcal{C}$ is simplicial. \square

Corollary 4.1.29. *Let $\mathcal{C} : \mathcal{X} \rightarrow \mathcal{Y}$ and $\mathcal{D} : \mathcal{Y} \rightarrow \mathcal{X}$ be simplicial carriers, so that $\text{id}_{\mathcal{X}}$ is carried by $\mathcal{D} \circ \mathcal{C}$, and $\text{id}_{\mathcal{Y}}$ is carried by $\mathcal{C} \circ \mathcal{D}$. Then $H_*(\mathcal{X}) \cong H_*(\mathcal{Y})$.*

Proof. Because \mathcal{C} and \mathcal{D} are simplicial, $\mathcal{C} \circ \mathcal{D}$ and $\mathcal{D} \circ \mathcal{C}$ are both simplicial and thus acyclic. We can take any cellular map $f : \mathcal{X} \rightarrow \mathcal{Y}$ carried by \mathcal{C} , and any cellular map $g : \mathcal{Y} \rightarrow \mathcal{X}$ carried by \mathcal{D} . Applying the chain functor, because $g \circ f$ is carried by $\mathcal{D} \circ \mathcal{C}$, the corresponding chain map is homotopic to the chain map for $\text{id}_{\mathcal{X}}$. Similarly, $f \circ g$ is homotopic to $\text{id}_{\mathcal{Y}}$ at the chain level. \square

Maps of cell complexes can be used to generate carriers

Definition 4.1.30. *Let $f : \mathcal{X} \rightarrow \mathcal{Y}$ be a cellular map. We define the carrier $\mathcal{C}^f : \mathcal{X} \rightarrow \mathcal{Y}$ to be $\mathcal{C}^f(x) = f(x) \subseteq \mathcal{Y}$, for each cell $x \in \mathcal{X}$. When composing carriers, we will use the notation $\mathcal{D} \circ f$ to denote the composition $\mathcal{D} \circ \mathcal{C}^f$.*

Note that \mathcal{C}^f carries f by definition. In the case where f is a simplicial map, then the carrier \mathcal{C}^f is simplicial.

4.1.5 From Acyclic Carriers to Equivalence

We'll now investigate conditions under which we can use acyclic carriers to demonstrate that two spaces have identical homology.

Proposition 4.1.31. *Let \mathcal{X}, \mathcal{Y} be cell complexes, with carriers $\mathcal{C} : \mathcal{X} \rightarrow \mathcal{Y}$ acyclic; $\mathcal{D} : \mathcal{Y} \rightarrow \mathcal{X}$ acyclic; $\mathcal{A} : \mathcal{X} \rightarrow \mathcal{X}$ acyclic, carrying the identity, with $\mathcal{D} \circ \mathcal{C} \subseteq \mathcal{A}$; and $\mathcal{B} : \mathcal{Y} \rightarrow \mathcal{Y}$ acyclic, carrying the identity, with $\mathcal{C} \circ \mathcal{D} \subseteq \mathcal{B}$. Then $H_*(\mathcal{X}) \simeq H_*(\mathcal{Y})$.*

Proof. Because \mathcal{C} is acyclic, there exists an augmentation-preserving chain map $F_* : C_*(\mathcal{X}) \rightarrow C_*(\mathcal{Y})$ carried by \mathcal{C} , and because \mathcal{D} is acyclic, there exists an augmentation-preserving chain map $G_* : C_*(\mathcal{Y}) \rightarrow C_*(\mathcal{X})$ carried by \mathcal{D} . The composition $G_* \circ F_*$ is augmentation preserving and carried by $\mathcal{D} \circ \mathcal{C} \subseteq \mathcal{A}$, which is acyclic, so $G_* \circ F_*$ is chain homotopic to $\text{id}_{C_*(\mathcal{X})} : C_*(\mathcal{X}) \rightarrow C_*(\mathcal{X})$. Similarly, $F_* \circ G_* \simeq \text{id}_{C_*(\mathcal{Y})}$.

Thus the induced maps on homology are isomorphisms, so $H_*(\mathcal{X}) \simeq H_*(\mathcal{Y})$. \square

We'll now use this to show the equivalence of a variant of the Nerve of a cover to the Nerve itself.

Definition 4.1.32. *Let \mathcal{U} be a cover of a set X . The maximal self-refinement of \mathcal{U} , denoted $\bar{\mathcal{U}}$, is defined as*

$$\bar{\mathcal{U}} = \{V \mid V = \bigcap_k U_k \neq \emptyset, \{U_k\} \subseteq \mathcal{U}\} \quad (4.8)$$

Definition 4.1.33. *Let \mathcal{U} be a cover of a set X . The subdivision of the Nerve of \mathcal{U} , denoted $\mathcal{N}(\bar{\mathcal{U}})$ is a simplicial complex on the vertex set $\bar{\mathcal{U}}$, where $(V_0, \dots, V_k) \in \mathcal{N}(\bar{\mathcal{U}})$ if there is some permutation of the indices so that $V_{i_0} \subseteq V_{i_1} \subseteq \dots \subseteq V_{i_k}$. This definition coincides with the definition of the Nerve of $\bar{\mathcal{U}}$ considered as a poset (category).*

In the case that $\bar{\mathcal{U}}$ is non-degenerate, meaning that $\cap\{U_k\} \not\subseteq U_i$ for all $U_i \notin \{U_k\}$ for all non-empty intersections, $\mathcal{N}(\bar{\mathcal{U}})$ is simply the standard subdivision of $\mathcal{N}(\mathcal{U})$. This immediately implies that $\mathcal{N}(\bar{\mathcal{U}}) \simeq \mathcal{N}(\mathcal{U})$ [51]. We will prove equivalence on homology even with degeneracy.

Proposition 4.1.34. $H_*(\mathcal{N}(\mathcal{U})) \simeq H_*(\mathcal{N}(\bar{\mathcal{U}}))$.

Proof. We define a carrier $\mathcal{C} : \mathcal{N}(\mathcal{U}) \rightarrow \mathcal{N}(\bar{\mathcal{U}})$ via

$$\mathcal{C}(U_0, \dots, U_k) = \langle V = \cap\{U_i\} \mid \{U_i\} \subseteq \{U_0, \dots, U_k\} \rangle \quad (4.9)$$

This carrier is acyclic because it forms a cone with the vertex $V = \cap_{i=0}^k U_i$.

We define a carrier $\mathcal{D} : \mathcal{N}(\bar{\mathcal{U}}) \rightarrow \mathcal{N}(\mathcal{U})$ via

$$\mathcal{D}(V_0, \dots, V_k) = \langle U \mid U \supset V_i \in \{V_0, \dots, V_k\} \rangle \quad (4.10)$$

Note that if (V_0, \dots, V_k) is a simplex in $\mathcal{N}(\bar{\mathcal{U}})$, there is a smallest set V_{i_0} in the simplex, and so $\mathcal{D}(V_0, \dots, V_k) = \mathcal{D}(V_{i_0})$. This also implies that \mathcal{C} is simplicial, thus acyclic.

Now, we have that $\mathcal{D} \circ \mathcal{C}(U_0, \dots, U_k)$ is a the simplex $(U_0, \dots, U_k, U'_0, \dots)$ where the extra simplices U'_i are added if $\cap_{i=0}^k U_i \subseteq U'_i$, which can appear for degenerate \bar{U} . This carrier is simplicial, thus acyclic, and clearly carries the identity.

The composition $\mathcal{C} \circ \mathcal{D}$ is acyclic because $\mathcal{C} \circ \mathcal{D}(V_0, \dots, V_k)$ forms a cone with the vertex on the minimal element $V_{i_0} \in \{V_0, \dots, V_k\}$. This composite carrier also carries the identity map.

We can now apply Proposition 4.1.31 to obtain the result. \square

4.2 Interleavings

An important question that we have put off until now is how to compare the persistent homology of two constructions. For instance, we would like to understand how the persistent homology of a Rips filtration changes if we perturb the underlying point cloud, or if we compare two samples from the same distribution. Other important applications are in simplifying computation - for instance, Rips filtrations tend to have a large number of simplices at large distances. Is there a way to perform a computation on a simplified complex that captures the large features while perturbing the small features?

Because homology is a homotopy invariant, if \mathcal{X}, \mathcal{Y} are topological spaces and $f : \mathcal{X} \rightarrow \mathcal{Y}$, $g : \mathcal{Y} \rightarrow \mathcal{X}$ are maps such that $g \circ f \simeq \text{id}_{\mathcal{X}}$ and $f \circ g \simeq \text{id}_{\mathcal{Y}}$, then $H_*(\mathcal{X}) \cong H_*(\mathcal{Y})$. On the chain level, C_*, D_* have isomorphic homology if there exist maps $F_* : C_* \rightarrow D_*$ and $G_* : D_* \rightarrow C_*$, so that $G_* F_* \simeq I_*^C$, and $F_* G_* \simeq I_*^D$.

In the setting of persistent homology, if \mathcal{X}^T and \mathcal{Y}^T are filtrations then if $\mathcal{X}^t \simeq \mathcal{Y}^t$ for all $t \in T$, then $PH_*(\mathcal{X}^T) \simeq PH_*(\mathcal{Y}^T)$. However, the requirement that two filtered spaces be homotopic at each filtration value is too strong to work in the setting of perturbed point clouds, or approximations of complexes. In general, we care about relating homological features that persist for a long range of filtration parameters, but want the capacity to ignore features that are only present for a brief interval, since these are typically interpretable as artifacts of sampling/perturbation.

These issues are addressed through the notion of interleaving. The basic idea is to construct maps $\tilde{F}_k^t, \tilde{G}_k^t$, $t \in T$, that can increase the filtration parameter so that

the following diagram commutes:

$$\begin{array}{ccccc}
 H_k(\mathcal{X}^t) & \xrightarrow{\quad} & H_k(\mathcal{X}^{t+\epsilon}) & \xrightarrow{\quad} & H_k(\mathcal{X}^{t+2\epsilon}) \\
 & \searrow \tilde{F}_k^t & \nearrow \tilde{F}_k^{t+\epsilon} & & \\
 & & & \nearrow \tilde{F}_k^{t+\epsilon} & \\
 & \nearrow \tilde{G}_k^t & \searrow \tilde{G}_k^{t+\epsilon} & & \\
 H_k(\mathcal{Y}^t) & \longrightarrow & H_k(\mathcal{Y}^{t+\epsilon}) & \longrightarrow & H_k(\mathcal{Y}^{t+2\epsilon})
 \end{array} \tag{4.11}$$

which can be accomplished if we construct chain maps $F_k^t, G_k^t, t \in T$

$$\begin{array}{ccccc}
 C_k(\mathcal{X}^t) & \longrightarrow & C_k(\mathcal{X}^{t+\epsilon}) & \longrightarrow & C_k(\mathcal{X}^{t+2\epsilon}) \\
 & \searrow F_k^t & \nearrow F_k^{t+\epsilon} & & \\
 & & & \nearrow F_k^{t+\epsilon} & \\
 & \nearrow G_k^t & \searrow G_k^{t+\epsilon} & & \\
 C_k(\mathcal{Y}^t) & \longrightarrow & C_k(\mathcal{Y}^{t+\epsilon}) & \longrightarrow & C_k(\mathcal{Y}^{t+2\epsilon})
 \end{array} \tag{4.12}$$

so that $G_k^{t+\epsilon} \circ F_k^t \simeq I_k^{t, t+2\epsilon}$ and $F_k^{t+\epsilon} \circ G_k^t \simeq I_k^{t, t+2\epsilon}$ for all t , where I_k denote the relevant inclusion maps. In this situation, we say that the chain complexes are ϵ -interleaved.

The rest of Section 4.2 will give a more formal development of interleavings, introduce the interleaving distance, and discuss geometric interpretations of the interleaving on persistence diagrams using the bottleneck distance.

The notion of interleaving was first introduced in [28], and a thorough reference for 1-dimensional interleavings is [30]. Interleavings of multi-parameter persistence modules are discussed in [67, 68], and more general posets in [18]. A homotopy-theoretic version of interleaving is also discussed in [12, 68], but we will focus on the algebraic version.

4.2.1 Filtrations, Posets, and Non-decreasing Maps

We will primarily be interested in comparing real-valued filtrations using interleavings, but it can be instructive to consider generalizations. Recall a filtered topological space, or filtration, is a sequence of spaces

$$\mathcal{X}^0 \subset \dots \subseteq \mathcal{X}^t \subseteq \dots$$

Alternatively, we can define a filtration as a functor from the totally ordered set \mathbb{R}^+ to the category of topological spaces where all maps are inclusions.

More generally, we can consider functors from partially ordered sets (posets) to topological categories.

Definition 4.2.1. *A partially ordered set, or poset, is a small category T where there is a unique arrow $t_i \rightarrow t_j$ iff $t_i \leq t_j$*

The notion of filtration can be expanded to posets

Definition 4.2.2. *A filtration over a poset T is a functor from T to a topological category where all maps are inclusions.*

For our applications (real-valued filtrations), we will consider posets T which are finite strict total orderings, considered as discrete subsets of \mathbb{R} . In the case where $T \subseteq \mathbb{R}^d$, the study of the diagrams of induced maps on homology is known as multi-parameter persistence [67], and for more general posets is known as generalized persistence [18]. While our applications do not use these fully generalized notions, the language of posets offers a clean description of the objects we will study and can give insight into some possible extensions.

Definition 4.2.3. *A filtered basis over a poset T is a collection of bases B^t where $B^t \subseteq B^{t'}$ for all $t \leq t' \in T$.*

Definition 4.2.4. *A filtered chain complex over a poset T is a functor from T to the category of chain complexes where $C_*^t \subseteq C_*^{t'}$ for all $t \leq t'$.*

We note that the chain functor applied to a filtration of cell complexes produces a filtered basis (from the filtration on cells) for a filtered chain complex.

In order to discuss interleavings, we need a notion of *non-decreasing map*, which is encoded as a poset functor

Definition 4.2.5. *Let S, T be posets. A poset functor, or non-decreasing map, $\alpha : S \rightarrow T$ satisfies the condition $\alpha(s_i) \leq \alpha(s_j)$ whenever $s_i \leq s_j$.*

In the case of real-valued filtrations, we will simply refer to non-decreasing maps. There is a partial ordering on non-decreasing maps. We say $\alpha \leq \beta : S \rightarrow T$ if $\alpha(s) \leq \beta(s)$ for all $s \in S$.

Definition 4.2.6. *We will refer to the non-decreasing map $\epsilon : t \mapsto t + \epsilon$ as the ϵ -shift map.*

When dealing with filtered objects in a category, we can also talk about morphisms indexed by a poset functor.

Definition 4.2.7. *Let $\mathcal{X}^S, \mathcal{Y}^T$ be filtered objects in a category over posets S, T respectively. Let $\alpha : S \rightarrow T$ be a non-decreasing map. An α -graded map $f^\alpha : \mathcal{X}^S \rightarrow \mathcal{Y}^T$ is a collection of maps $f^s : \mathcal{X}^s \rightarrow \mathcal{Y}^{\alpha(s)}$ for each $s \in S$ so that the following diagram commutes.*

$$\begin{array}{ccc} \mathcal{X}^s & \longrightarrow & \mathcal{X}^{s'} \\ \downarrow f^s & & \downarrow f^{s'} \\ \mathcal{Y}^{\alpha(s)} & \longrightarrow & \mathcal{Y}^{\alpha(s')} \end{array} \quad (4.13)$$

For example, in Equation (4.11), the maps $\{F_k^t\}_{t \in T}, \{G_k^t\}_{t \in T}$ are ϵ -graded maps because they apply the ϵ -shift functor to the filtration indices.

Definition 4.2.8. *Let $F_*^\alpha, G_*^\alpha : C_*^S \rightarrow D_*^T$ be α -filtered maps of chain complexes. We say F^α, G^α are β -chain homotopic, where $\beta : T \rightarrow T$ is a non-decreasing map if there exists a collection of maps $H_k^s : C_k^s \rightarrow D_{k+1}^{\beta \circ \alpha(s)}$ $k = 0, 1, \dots$, and $s \in S$, so that*

$$\partial_{k+1}^D H_k^s + H_{k-1}^s \partial_k^C = G_k^s - F_k^s \quad (4.14)$$

4.2.2 Persistence Vector Spaces and Interleavings

Definition 4.2.9. *Let T be a poset, and fix a field \mathbb{F} . A persistence vector space over T , denoted V^T is a functor $T \rightarrow \mathbf{Vect}_{\mathbb{F}}$ (the category of vector spaces over \mathbb{F}).*

Note that this is different from the notion of a *filtered* vector space, because the maps need not be inclusions (note that maps on homology induced by inclusions are not necessarily inclusions of sub-spaces).

The information in a persistence vector space V^T can be encoded in a quiver representation \mathcal{Q} over the underlying graph $\text{DAG}(T)$. The information contained in a persistence vector space V^T is a vector space V^t for each $t \in T$, and maps $A^{t,t'} : V^t \rightarrow V^{t'}$ for each $t < t' \in T$ which are consistent via composition.

Definition 4.2.10. *Let V^S, W^T be persistence vector spaces over S, T respectively. An α -graded map $F^\alpha : V^S \rightarrow W^T$ for a nondecreasing map $\alpha : S \rightarrow T$ is a collection of linear maps*

$$F^s = F^{s, \alpha(s)} : V^s \rightarrow W^{\alpha(s)}$$

for each $s \in S$, which commute with the maps in V^S and W^T , meaning the following diagram commutes

$$\begin{array}{ccc} V^s & \xrightarrow{A^{s,s'}} & V^{s'} \\ \downarrow F^s & & \downarrow F^{s'} \\ W^{\alpha(s)} & \xrightarrow{A^{\alpha(s), \alpha(s')}} & W^{\alpha(s')} \end{array} \quad (4.15)$$

Clearly, one way to obtain an α -graded map of persistence vector spaces is to apply the homology functor to an α -graded map of filtered chain complexes.

Definition 4.2.11. *Let $\gamma : T \rightarrow T$ be a poset functor, and V^T be a persistence vector space over T . We define $I^\gamma : V^T \rightarrow V^T$ to be the γ -shift map where $I^{t, \gamma(t)} : V^t \rightarrow V^{\gamma(t)} = A^{t, \gamma(t)}$.*

We'll now return to the case of real-valued filtrations. Consider the diagram in Equation (4.11). Applying the homology functor, we obtain a diagram

$$\begin{array}{ccccc} H_k(\mathcal{X}^t) & \xrightarrow{\quad} & H_k(\mathcal{X}^{t+\epsilon}) & \xrightarrow{\quad} & H_k(\mathcal{X}^{t+2\epsilon}) \\ & \searrow \tilde{F}_k^\epsilon & \nearrow \tilde{G}_k^\epsilon & & \\ & & & \searrow \tilde{F}_k^\epsilon & \nearrow \tilde{G}_k^\epsilon \\ H_k(\mathcal{Y}^t) & \xrightarrow{\quad} & H_k(\mathcal{Y}^{t+\epsilon}) & \xrightarrow{\quad} & H_k(\mathcal{Y}^{t+2\epsilon}) \end{array} \quad (4.16)$$

The top row is a persistent vector space $PH_k(X^T)$, and the bottom row is a persistent vector space $PH_k(Y^T)$. The diagonal maps are ϵ -graded maps \tilde{F}_k^ϵ and \tilde{G}_k^ϵ , where ϵ is the ϵ -shift map in Definition 4.2.6, and the conditions on homotopy imply that $\tilde{F}^\epsilon \tilde{G}^\epsilon = \tilde{I}^{2\epsilon}$ on $H_k(\mathcal{X}^T)$ and $\tilde{G}^\epsilon \tilde{F}^\epsilon = \tilde{I}^{2\epsilon}$ on $H_k(\mathcal{Y}^T)$.

We are now ready to define the general notion of interleaving on persistence vector spaces.

Definition 4.2.12. An (α, β) -interleaving between V^S and W^T is a pair of graded maps $F^\alpha : V^S \rightarrow W^T, G^\beta : W^T \rightarrow V^S$ so so that $G^\beta \circ F^\alpha \cong I^{\beta \circ \alpha}$ and $F^\alpha \circ G^\beta \cong I^{\alpha \circ \beta}$.

If $S, T \subseteq \mathbb{R}$, and $\alpha, \beta = \epsilon$, then we refer to an (ϵ, ϵ) -interleaving as simply an ϵ -interleaving. If $\alpha \leq \alpha'$ and $\beta \leq \beta'$, then we can construct an (α', β') -interleaving from a (α, β) -interleaving by appending inclusion maps: $F^{\alpha'} = \{I^{\alpha(s), \alpha'(s)} \circ F^{s, \alpha(s)}\}$, and $G^{\beta'} = \{I^{\beta(t), \beta'(t)} \circ G^{t, \beta(t)}\}$.

Proposition 4.2.13. Suppose $f^\alpha : \mathcal{X}^S \rightarrow \mathcal{Y}^T$ and $g^\beta : \mathcal{Y}^T \rightarrow \mathcal{X}^S$ are graded maps, such that $f^\alpha \circ g^\beta \simeq \iota^{\alpha \circ \beta}$ on \mathcal{Y}^T and $g^\beta \circ f^\alpha \simeq \iota^{\beta \circ \alpha}$ on \mathcal{X}^S . Then $H_*(\mathcal{X}^S)$ and $H_*(\mathcal{Y}^T)$ are (α, β) -interleaved.

Proof. Applying the homology functor we obtain maps \tilde{F}^α and \tilde{G}^β . Because homology is homotopy invariant, because compositions of the graded maps are homotopic to the inclusions we have $\tilde{F}^\alpha \tilde{G}^\beta = I^{\alpha \circ \beta}$ on $H_*(\mathcal{Y}^T)$ and $\tilde{G}^\beta \tilde{F}^\alpha = I^{\beta \circ \alpha}$ on $H_*(\mathcal{X}^S)$. \square

Note that Proposition 4.2.13 applies equally well to homotopy in topological categories and chain complex categories.

4.2.3 The Interleaving and Bottleneck Distances

Definition 4.2.14. Let T be a poset equipped with a dissimilarity measure on objects $d : (t_i, t_j) \rightarrow \mathbb{R}_+$. The distortion of a functor $\alpha : T \rightarrow T$ is defined as

$$d(\alpha) = \sup_{t \in T} d(t, \alpha(t)) \quad (4.17)$$

The distortion of a persistence map, $d(F_\alpha)$, is the distortion of α , $d(\alpha)$.

For example, if we take the diagram in Equation (4.16), because both filtrations are over the real numbers, by taking $d(t, t') = |t' - t|$, we see that $d(\epsilon) = \epsilon$, and $d(F_\epsilon) = d(G_\epsilon) = \epsilon$, and $d(G_\epsilon F_\epsilon) = d(F_\epsilon G_\epsilon) = 2\epsilon$.

We will now only consider dissimilarity measures d on the objects of T which are metrics, as in the example above.

Definition 4.2.15. Let T be a poset, and d be a metric on the objects of T . The interleaving distance between two persistence vector spaces V^T, W^T over T , denoted $d_I(V^T, W^T)$ is the smallest distortion incurred by an interleaving between V^T and W^T .

$$d_I(V^T, W^T) = \inf\{\max(d(\alpha), d(\beta)) \mid \exists(\alpha, \beta)\text{-interleaving of } V^T, W^T\} \quad (4.18)$$

The interleaving distance can be checked to satisfy the definition of a metric. As an example, the diagram in Equation (4.11) bounds the interleaving distance between $PH_*(\mathcal{X}^T)$ and $PH_*(\mathcal{Y}^T)$ by ϵ .

In the special case of ordinary real-valued persistence, the interleaving distance between two persistence vector spaces is equivalent to a geometric notion of distance on persistence diagrams known as the bottleneck distance.

Definition 4.2.16. Let $D_1 = \{(b_i, d_i)\}$ and $D_2 = \{(b_j, d_j)\}$ be two persistence diagrams, augmented with an infinite number of zero-length pairs. The bottleneck distance is defined as

$$d_b(D_1, D_2) = \inf_{\gamma} \sup_{b_i, d_i} \|(b_i, d_i) - \gamma(b_i, d_i)\|_{\infty} \quad (4.19)$$

where γ is a bijection between D_1 and D_2

For 1-dimensional persistent homology, $d_I = d_B$ [67]. This means that if two persistence modules are ϵ -interleaved, their barcodes are closely related, and vice-versa.

The importance of the interleaving and bottleneck distances are that they give a way to characterize how sensitive persistent homology of a filtration is to perturbations of the data that was used to construct the filtration. Ideally, we would like to use constructions that are *stable*, meaning that the persistent homology of a filtration is not overly sensitive to the input data. In general this will be accomplished by bounding the interleaving distance by constructing a specific interleaving.

4.3 Filtered Carriers and Interleavings

In Section 4.1, we covered the classical notion of a carrier between cellular or chain complexes, and how carriers can be used to extend maps from initial data. In Section 4.2 we introduced how interleavings can be used to compare persistence vector spaces, and in the case of ordinary persistence, can be related to the bottleneck distance.

In this section, we will develop a notion of *filtered carrier*. This will provide a general-purpose tool for use in constructing explicit interleavings. We will then apply this tool to analyze the relationship between the persistent homology of a variety of filtered complexes on point cloud data.

4.3.1 Preliminary Notions

Definition 4.3.1. *A filtered carrier of chain complexes over a poset T , denoted $\mathcal{C}^T : C_*^S \rightarrow D_*^T$ is an assignment of basis vectors of C_*^S to filtered sub-complexes of D_*^T . In situations where T is understood, we will drop the superscript, and simply write $\mathcal{C} : C_*^S \rightarrow D_*^T$.*

Note that while a basis element $x \in C_*^S$ may appear at parameter $s \in S$, the carrier $\mathcal{C}^T(x)$ is filtered by T . The notion of filtered carrier can also be applied to filtrations of cell complexes.

Definition 4.3.2. *We say a filtered chain complex C_*^T is acyclic at t if C_*^t is acyclic.*

Definition 4.3.3. *We say a filtered chain complex C_*^T is α -acyclic if every cycle in C_*^t has a boundary in $C_*^{\alpha(t)}$.*

Note that Definition 4.3.2 is just a special case of Definition 4.3.3, where $\alpha \leq \text{id}$ for all $t' \geq t$.

Definition 4.3.4. *Let C_*^S, D_*^T be filtered chain complexes, and $\mathcal{C}^T : C_*^S \rightarrow D_*^T$ be a filtered carrier, and $\alpha : S \rightarrow T, \beta : T \rightarrow T$ be non-decreasing maps. We say \mathcal{C}^T is (α, β) -acyclic if $\mathcal{C}^T(x)$ is β -acyclic after $t = \alpha(s)$ for all $x \in C_*^s$ and for all $s \in S$. In the case where $\beta = \text{id}$, then we just say \mathcal{C}^T is α -acyclic.*

We can also extend the notion of carrying a map

Definition 4.3.5. Let $\mathcal{C}^T : C_*^S \rightarrow D_*^T$ be a filtered carrier, and F_*^α be an α -filtered chain map. We say that $F_*^\alpha : C_*^S \rightarrow D_*^T$ is carried by \mathcal{C}^T if $F^\alpha(x) \in \mathcal{C}^T(x)$ at parameter $\alpha(s)$ for all basis elements $x \in C_*^S$.

Theorem 4.3.6. [Filtered acyclic carrier theorem] Let $\mathcal{C}^T : C_*^S \rightarrow D_*^T$ be an (α, β) -acyclic carrier of filtered chain complexes, with S a strict total order with an initial object $0 \in S$. Let $L_*^S \subseteq C_*^S$ be a filtered sub-complex generated by a filtered sub-basis of C_*^S , and $\tilde{F}^\alpha : L_*^S \rightarrow D_*^T$ be an α -filtered chain map carried by \mathcal{C}^T . Then \tilde{F}^α extends to a filtered chain map $F^{\beta^k \circ \alpha} : C_*^S \rightarrow D_*^T$, where k is the maximal dimension of the chain map, and the extension is unique up to β -chain homotopy (Definition 4.2.8).

Proof. We will proceed by induction on the dimension k of the map, and on the total order on S . First, we start with $\tilde{F}_0^{0, \alpha(0)} : L_0^0 \rightarrow D_0^{\alpha(0)}$. From the acyclic carrier theorem, Theorem 4.1.8, we can extend to a chain map $F_0^{0, \alpha(0)} : C_0^0 \rightarrow D_0^{\alpha(0)}$.

Now, let $s > 0$. Assume that we have extended F_0^α for all $r < s$ so that if $r' < r$,

$$F_0^{r, \alpha(r)} \big|_{C_{*'}^{r'}} = F_0^{r', \alpha(r')} \quad (4.20)$$

Note that this is satisfied trivially for $s = 0$. Let $L_0^S = L_0^0 \cup \bigcup_{r < s} C_0^r$, and \tilde{F}_0^α denote the extended map up to all $r < s$. We can now apply Theorem 4.1.8 again to extend to $F_0^{s, \alpha(s)}$ to C_0^s . Since S is a strict total order, Equation (4.20) continues to be satisfied because the function is extended on each basis element exactly once. By induction, we can extend to a map of 0-chains $F^\alpha : C_0^S \rightarrow D_0^T$.

Now, we'll extend to higher-dimensional chains. Assume that we have extended to $F_k^{\beta^k \circ \alpha} : C_k^S \rightarrow D_k^T$. Again, we'll start with the initial object 0 of S . We take $L_{* \leq k+1}^0 = C_{* \leq k}^0 \cup L_{* \leq k}^0$. We have extended $F_{* \leq k}^{\beta^k \circ \alpha} : C_{* \leq k}^0 \rightarrow D_{* \leq k}^{\beta^k \circ \alpha(s)}$. Let $x \in B_{k+1}$ be a basis element that we must extend at filtration parameter $s = 0$. We need $\partial_{k+1} F_{k+1} x = F_k \partial_{k+1} x$. The image of the boundary $F_k \partial_{k+1} x$ lies in $D_k^{\beta^k \circ \alpha(0)}$, but since \mathcal{C} is (α, β) -acyclic, the cycle need not have a boundary until we increase the filtration parameter T by another factor of β . We can increase the grade on the map $F^{\beta^{k+1} \circ \alpha}$, taking $F^{\beta^{k+1} \circ \alpha} x = \iota^\beta F^{\beta^k \circ \alpha} x$ for $x \in L^0$, and then apply Theorem 4.1.8 to extend the map for $x \in C_{k+1}^0$.

Now, we'll extend to higher dimensional chains for $s > 0$. Assume that so far we have satisfied for $r' < r < s$

$$F_{k+1}^{r, \beta^{k+1} \circ \alpha(r)} \Big|_{C_k^{r'}} = F_{k+1}^{r, \beta^{k+1} \circ \alpha(r)} \quad (4.21)$$

and furthermore, that we have shifted the chain maps in lower dimensions via $F^{\beta^{k+1} \circ \alpha} = \iota^\beta F^{\beta^k \circ \alpha}$. Let $x \in B_{k+1}$ via a basis element that we must extend at filtration parameter s . The image of the boundary $F_k \partial_{k+1} x$ lies in $D_k^{\beta^k \circ \alpha(s)}$, and we have already shifted the grade to $\beta^{k+1} \circ \alpha(s)$ at which point the cycle is a boundary of some $y \in D_{k+1}^{\beta^{k+1} \circ \alpha(s)}$ in $\mathcal{C}(x)$. Thus, we can extend the map via $F_{k+1}^{\beta^{k+1} \circ \alpha} x = y$. Again, because S is a strict total order, the map is extended for every basis element exactly once, so Equation (4.21) is satisfied.

To see that the map is unique up to β -chain homotopy, we simply need to note that we need to increase grades by a factor of β to produce homotopies between maps as in Proposition 4.1.10. \square

Note that to compute induced maps in homology in dimension k , it is only necessary to extend maps up to dimension k . In many cases, β will be the identity id , in which case there is no additional penalty for extending to higher-dimensional chains.

Remark 4.3.7. *In Theorem 4.3.6 we used the strict total ordering on S to extend the initial map so that we guaranteed that Equation (4.20) is always satisfied. If S is not a strict total ordering, then additional restrictions on the extension are needed to satisfy this condition.*

Proposition 4.3.8. *Let $\mathcal{C} : C_*^S \rightarrow D_*^T$ be an (α, β) -acyclic carrier that is proper with respect to a T -filtered basis B_*^D of D_* . Then there exists a canonical augmentation-preserving chain map $F_0^\alpha : C_0^S \rightarrow D_0^T$ carried by \mathcal{C} .*

Proof. As in Proposition 4.1.13, we map 0-dimensional basis elements in C_0^S to 0-dimensional basis elements in D_0^T that are carried by \mathcal{C} . Such elements exist at level $\alpha(s)$ for 0-dimensional basis elements at parameter s in C_0^s , so the map is α -filtered. \square

Note that the map F_0^α in Proposition 4.3.8 can then be extended to $F_k^{\beta^k \circ \alpha}$ using Theorem 4.3.6.

We'll now extend Definition 4.1.14 to the filtered setting

Definition 4.3.9. *Let \mathcal{X}^S and \mathcal{Y}^T be filtered simplicial complexes over posets S, T , and $\alpha : S \rightarrow T$ be a non-decreasing map. We say $\mathcal{C}^T : \mathcal{X}^S \rightarrow \mathcal{Y}^T$ is α -simplicial if $\mathcal{C}^T(x)$ is a simplex at level $\alpha(s)$ for all $x \in \mathcal{X}^s$.*

An α -simplicial carrier is equivalent to an α -simplicial multi-valued map defined in [31]. Note that an α -simplicial carrier is α -acyclic (or (α, id) -acyclic). This allows for a proposition analogous to corollary 4.1.28

Proposition 4.3.10. *If $\mathcal{C}^T : \mathcal{X}^S \rightarrow \mathcal{Y}^T$ is α -simplicial and $\mathcal{D}^U : \mathcal{Y}^T \rightarrow \mathcal{Z}^U$ is β -simplicial, then $\mathcal{D}^U \circ \mathcal{C}^T$ is $(\beta \circ \alpha)$ -simplicial.*

Proof. Let $x \in \mathcal{X}^s$. Then $\mathcal{C}(x)$ is simplicial at level $\alpha(s)$. Let $y = \mathcal{C}^{\alpha(s)}(x)$ be the simplex in \mathcal{Y} . Since \mathcal{D} is β -simplicial, $\mathcal{D}(y)$ is simplicial at level $\beta \circ \alpha(s)$. \square

4.3.2 Filtered Carriers to Interleavings

We'll now turn to examining the conditions under which interleavings can be constructed from filtered carriers.

Proposition 4.3.11. *Let \mathcal{X}^S and \mathcal{Y}^T be filtered cell complexes, and suppose that $\mathcal{C} : \mathcal{X}^S \rightarrow \mathcal{Y}^T$ is an α -acyclic carrier, $\mathcal{D} : \mathcal{Y}^T \rightarrow \mathcal{X}^S$ is a β -acyclic carrier, $\mathcal{A} \supseteq \mathcal{D} \circ \mathcal{C}$ is a $(\beta \circ \alpha)$ -acyclic carrier that carries the inclusion map on \mathcal{Y}^T , and $\mathcal{B} \supseteq \mathcal{C} \circ \mathcal{D}$ is $(\alpha \circ \beta)$ -acyclic and carries the inclusion map on \mathcal{X}^S . Then $PH_*(\mathcal{X}^S)$ and $PH_*(\mathcal{Y}^T)$ are (α, β) -interleaved.*

Proof. First, we construct augmentation-preserving filtered maps $F^\alpha : C_*(\mathcal{X}^s) \rightarrow C_*(\mathcal{Y}^{\alpha(s)})$ and $G^\beta : C_*(\mathcal{Y}^t) \rightarrow C_*(\mathcal{X}^{\beta(t)})$ using Proposition 4.3.8 and Theorem 4.3.6. Now, note that $G^\beta \circ F^\alpha$ is augmentation preserving, and is carried by $\mathcal{D} \circ \mathcal{C} \subseteq \mathcal{A}$ which also carries the inclusion map, so $G^\beta \circ F^\alpha \simeq I^\mathcal{X}$. Similarly, $F^\alpha \circ G^\beta \simeq I^\mathcal{Y}$. Thus, the maps F^α and G^β give an (α, β) -interleaving. \square

In practice, more specific situations reduce the number of conditions that we need to satisfy. Often, we will find it convenient to take $\mathcal{A} = \mathcal{C} \circ \mathcal{D}$, and $\mathcal{B} = \mathcal{D} \circ \mathcal{C}$ when we can show that the composites are acyclic and carry inclusions.

Corollary 4.3.12. *Suppose $f^\alpha : \mathcal{X}^s \rightarrow \mathcal{Y}^{\alpha(s)}$ is a surjective simplicial map for every $s \in S$, and suppose $\mathcal{C} : \mathcal{Y}^T \rightarrow \mathcal{X}^S$, defined by $\mathcal{C}(y) = \langle f^{-1}(y) \rangle$ be a β -acyclic carrier. Then $PH_*(\mathcal{X}^S)$ and $PH_*(\mathcal{Y}^T)$ are (α, β) -interleaved.*

Proof. Because f^α is simplicial, the carrier \mathcal{C}^f is an α -acyclic carrier that carries f^α . Because f^α is a surjective simplicial map, $\mathcal{C}(y)$ is nonempty and maps to proper subcomplexes of \mathcal{X}^S for each $y \in \mathcal{Y}^T$, so is a well-defined filtered carrier. By definition, of \mathcal{C} , the composition $\mathcal{C} \circ f^\alpha$ carries the inclusion map $\iota^{\mathcal{X}}$. Additionally, $\mathcal{C} \circ f^\alpha$ is $(\beta \circ \alpha)$ -acyclic, because \mathcal{C} is β -acyclic for the simplex $f^\alpha(x)$ for each $x \in \mathcal{X}^S$ (a filtered version of Proposition 4.1.27). Because $\mathcal{C}(y) = \langle f^{-1}(y) \rangle$, $f^\alpha \circ \mathcal{C}(y) = \langle y \rangle$, which is a simplicial carrier and thus acyclic. Note that $y \in f^\alpha \circ \mathcal{C}(y)$, so $f^\alpha \circ \mathcal{C}$ carries $\iota^{\mathcal{Y}}$. We can now apply Proposition 4.3.11 to complete the proof. \square

In the case where carriers are obtained from a correspondence Ω , we need not satisfy that the composition of carriers is acyclic.

Lemma 4.3.13. *Let \mathcal{X}^S and \mathcal{Y}^T be simplicial complexes with underlying vertex sets X and Y respectively. Let $\Omega \subset X \times Y$ be a correspondence, with induced carriers $\mathcal{C}_\Omega : \mathcal{X}^S \rightarrow \mathcal{Y}^T$, and $\mathcal{D}_\Omega : \mathcal{Y}^T \rightarrow \mathcal{X}^S$. Then if $(y_0, \dots, y_\ell) \in \mathcal{C}_\Omega(x_0, \dots, x_k)$, then $(x_0, \dots, x_k) \in \mathcal{D}_\Omega(y_0, \dots, y_\ell)$.*

Proof. $(y_0, \dots, y_\ell) \in \mathcal{C}_\Omega(x_0, \dots, x_k)$ implies $y_0, \dots, y_\ell \in \Omega(x_0, \dots, x_k)$. Because Ω is a relation, this means $x_0, \dots, x_k \in \Omega(y_0, \dots, y_\ell)$, so $(x_0, \dots, x_k) \in \mathcal{D}_\Omega(y_0, \dots, y_\ell)$. \square

4.3.3 Stability of Flag Filtrations

We'll now examine a construction that is commonly used when producing filtrations on data.

Definition 4.3.14. *A simplicial complex is a flag complex if it is the maximal simplicial complex on its 1-skeleton. A flag filtration is a filtration of flag complexes, and is completely determined by a filtration on its edges.*

Flag complex constructions are often easy to work with because they only require pairwise comparisons. We'll consider examples where all edges eventually appear, such as in Vietoris-Rips filtrations and \mathcal{D}_1 , and $\check{\mathcal{C}}_1$ filtrations.

Let X and Y be finite sets of points, and let $f : X \times X \rightarrow \mathbb{R}$, $g : Y \times Y \rightarrow \mathbb{R}$. We'll denote the sub-levelset flag filtrations on X and Y as \mathcal{X}^f , \mathcal{Y}^g respectively, with $\mathcal{X}^f(r)$, $\mathcal{Y}^g(r)$ denoting the complexes at parameter $r \in \mathbb{R}$.

$$\mathcal{X}^f(r) = \langle (x_i, x_j) \mid f(x_i, x_j) \leq r \rangle \quad (4.22)$$

We will assume that $f(x, x) \leq f(x, x')$ for all $x' \in X$, which will allow us to assume that the zero-simplex (x) appears at parameter $f(x, x)$ in \mathcal{X}^f .

Let $\Omega \subseteq X \times Y$ be a binary relation between the sets X and Y . Let $d(f, g; \Omega)$ denote the distortion of the functions f and g over Ω :

$$d(f, g; \Omega) = \sup_{(x, y), (x', y') \in \Omega} |f(x, x') - g(y, y')| \quad (4.23)$$

We'll consider situations in which f and g are dissimilarities (or metrics) on X and Y respectively. When X and Y are safely assumed to be equipped with f and g respectively, we'll use the notation $d(\Omega) = d(f, g; \Omega)$. It follows immediately that if $f(x, x) = 0$ for $x \in X$, then for any $y, y' \in \Omega(x)$, $g(y, y') \leq d(\Omega)$. A similar statement may be made if $g(y, y) = 0$. Note that these conditions generally hold for dissimilarities.

Proposition 4.3.15. *Suppose $\Omega \subset X \times Y$ is a left-total binary relation. Then the carrier $\mathcal{C}_\Omega : \mathcal{X}^f \rightarrow \mathcal{Y}^g$ (Definition 4.1.23) is $d(\Omega)$ -simplicial.*

Proof. Let $(x_0, \dots, x_k) \in \mathcal{X}^f(r)$. We have that

$$\mathcal{C}_\Omega(x_0, \dots, x_k) = \langle \Omega(x_0, \dots, x_k) \rangle$$

Suppose $y, y' \in \Omega(\{x_0, \dots, x_k\})$. Then there is some $x, x' \in \{x_0, \dots, x_k\}$ such that $y \in \Omega(x)$, $y' \in \Omega(x')$. Since $(x_0, \dots, x_k) \in \mathcal{X}^f(r)$, $f(x, x') \leq r$, and $|f(x, x') - g(y, y')| \leq d(\Omega)$, we obtain $g(y, y') \leq r + d(\Omega)$. Thus, $(y, y') \in \mathcal{Y}^g(r + d(\Omega))$. Because this holds

for every pair $y, y' \in \mathcal{C}_\Omega(x_0, \dots, x_k)$, all possible edges are in the carrier, forming a clique. Because \mathcal{Y}^g is a flag complex, $\mathcal{C}_\Omega(x_0, \dots, x_k)$ is a simplex of dimension $|\Omega(x_0, \dots, x_k)|$. This holds for all simplices in $\mathcal{X}^f(r)$, so the carrier is simplicial after applying a $d(\Omega)$ -shift to the filtration index. \square

In the case that $f(x, x) = 0$ for all $x \in X$ and $g(y, y) = 0$ for all $y \in Y$, and a binary relation $\Omega \subset X \times Y$ is both left-total and right-total, we obtain $d(\Omega)$ -simplicial carriers $\mathcal{C}_\Omega : \mathcal{X}^f \rightarrow \mathcal{Y}^g$, and $\mathcal{D}_\Omega : \mathcal{Y}^g \rightarrow \mathcal{X}^f$.

As a result of proposition 4.3.10, the composition of carriers $\mathcal{D}_\Omega \circ \mathcal{C}_\Omega : \mathcal{X}^f \rightarrow \mathcal{X}^f$ is $2d(\Omega)$ -simplicial. Furthermore, because $y \in \Omega(x)$ implies $x \in \Omega(y)$, the inclusion map $\mathcal{X}^f(r) \rightarrow \mathcal{X}^f(r + 2d(\Omega))$ is carried by $\mathcal{D}_\Omega \circ \mathcal{C}_\Omega$. We can thus apply Proposition 4.3.11 to obtain the following interleaving stability theorem for flag filtrations:

Theorem 4.3.16. *Let \mathcal{X}^f and \mathcal{Y}^g be sub-levelset flag filtrations with vertex sets X and Y respectively, based on the edge filtration functions $f : X \times X \rightarrow \mathbb{R}$ and $g : Y \times Y \rightarrow \mathbb{R}$. Let $\Omega \in \mathcal{C}(X, Y)$ be a correspondence. Then there exists a $d(\Omega)$ -interleaving of $PH_*(\mathcal{X}^f)$ and $PH_*(\mathcal{Y}^g)$.*

Proof. We apply Proposition 4.3.11 to the above discussion. \square

4.3.4 Stability of Geometric Filtrations

Theorem 4.3.16 encompasses a variety of results that have been proven for flag filtrations in topological data analysis. The first such result was a bound on the bottleneck distance between persistence diagrams of Vietoris Rips filtrations, found in [29]. Theorem 4.3.16 was established in its full generality in [31], which uses contiguous simplicial maps in its proof technique. Our approach using carriers is equivalent, as a result of the construction found in Equation (4.1). This section will review several applications of Theorem 4.3.16 as found in [31] which we will use in Section 4.4. In addition, we will provide an interleaving on Dowker and Rips filtrations in Theorem 4.3.27, which to the best of our knowledge is the first such result explicitly relating the two constructions.

Distances

First, we want to define several notions of distances on spaces and sets that we will use in the rest of the chapter.

If (X, d_X) is a metric space, then there are two notions of induced distance on sets $U, V \subseteq X$ we will use:

Definition 4.3.17. *The Hausdorff distance between two sets $U, V \subseteq (X, d_X)$ is*

$$d_H(U, V) = \max\left(\sup_{x \in U} \inf_{y \in V} d_X(x, y), \sup_{y \in V} \inf_{x \in U} d_X(x, y)\right) \quad (4.24)$$

There is also a premetric

Definition 4.3.18. *The set premetric on $U, V \subseteq X$ is defined as*

$$d_p(U, V) = \inf_{x \in U, y \in V} d_X(x, y) \quad (4.25)$$

Note that Definition 4.3.18 does not satisfy the identity of discernibles or triangle inequality conditions of a metric.

We also want to be able to compare different metric spaces:

Definition 4.3.19. *The Gromov-Hausdorff distance between two compact metric spaces (X, d_X) and (Y, d_Y) is*

$$d_{GH}((X, d_X), (Y, d_Y)) = \inf_{\Omega \in \mathcal{C}(X, Y)} \sup_{(x, y), (x', y') \in \Omega} |d_X(x, x') - d_Y(y, y')|$$

We think of d_{GH} as measuring the amount unavoidable distortion incurred if we seek to form a correspondence between X and Y .

Vietoris-Rips Stability

An immediate application of theorem 4.3.16 is the Gromov-Hausdorff distance bound for Vietoris-Rips persistence.

Theorem 4.3.20. [29] For any finite metric spaces (X, d_X) and (Y, d_Y) , for any $k \in \mathbb{N}$,

$$d_I(PH_k(\mathcal{R}(X, d_X)), PH_k(\mathcal{R}(Y, d_Y))) \leq d_{GH}((X, d_X), (Y, d_Y))$$

Proof. This follows immediately from Theorem 4.3.16 by using $f = d_X$, $g = d_Y$ and using the correspondence in $\mathcal{C}(X, Y)$ that realizes the Gromov-Hausdorff distance. \square

Lazy Dowker Stability

We now turn to the use of Theorem 4.3.16 in the investigation of the stability of the lazy Dowker complex \mathcal{D}_1 , also known as the lazy Witness complex [37]. Let (\mathbf{L}, \mathbf{X}) be a landmark-space pairs in an ambient space (X, d_X) . We define the witnessed dissimilarity on L as

$$d_{WX}(\ell_i, \ell_j) = \inf_r \{r \mid \exists x \in \mathbf{X} \text{ s.t. } d_X(\ell_i, x), d_X(\ell_j, x) \leq r\}$$

Note that $d_{WX}(\ell_i, \ell_i)$ may be strictly greater than zero. If $d_{WX}(\ell_i, \ell_i) = 0$ for all $\ell_i \in \mathbf{L}$ (for instance, if $\mathbf{L} \subseteq \mathbf{X}$), then the Rips filtration $\mathcal{R}(\mathbf{L}, d_{WX}; r)$ is the same as the lazy dowker filtration $\mathcal{D}_1(\mathbf{L}, \mathbf{X}; r)$.

Proposition 4.3.21. Suppose that $d_H(\mathbf{X}, \mathbf{X}') \leq \epsilon_X$, and $d_H(\mathbf{L}, \mathbf{L}') \leq \epsilon_L$ in an ambient metric space with metric d_X . Let $\Omega \in \mathcal{C}(\mathbf{L}, \mathbf{L}')$ be the sublevel-set correspondence $\Omega = \Omega(d_X, \epsilon_L)$. Then $d(d_{WX}, d_{WX'}; \Omega) \leq \epsilon_X + \epsilon_L$

Proof. We need to show that for any $(\ell_i, \ell'_i), (\ell_j, \ell'_j) \in \Omega$ that

$$|d_{WX}(\ell_i, \ell_j) - d_{WX'}(\ell'_i, \ell'_j)| \leq \epsilon_X + \epsilon_L$$

Let $\ell \in L$, and $\ell' \in \Omega(\ell)$. Because d_X is a metric, by triangle inequality we have $d_X(\ell', x') \leq d_X(\ell, x') + \epsilon_L$ for any $x' \in \mathbf{X}'$. Additionally, because $d_H(\mathbf{X}, \mathbf{X}') \leq \epsilon_X$, for any $x' \in \mathbf{X}'$, there exists some $x \in \mathbf{X}$ such that $d_X(x, x') \leq \epsilon_X$. Thus, for every $x' \in \mathbf{X}'$, there is some $x \in \mathbf{X}$ so that

$$d_X(\ell', x') \leq d_X(\ell, x) + \epsilon_L + \epsilon_X$$

The choice of ℓ and ℓ' was arbitrary, and holds for any pair.

Now, let $(\ell_i, \ell'_i), (\ell_j, \ell'_j) \in \Omega$. Using the above, we have

$$\begin{aligned}
d_{WX'}(\ell'_i, \ell'_j) &= \inf_r \{r \mid \exists x' \in \mathbf{X}' \text{ s.t. } d_{\mathcal{X}}(\ell'_i, x'), d_{\mathcal{X}}(\ell'_j, x') \leq r\} \\
&\leq \inf_r \{r \mid \exists x \in \mathbf{X} \text{ s.t. } d_{\mathcal{X}}(\ell'_i, x)d_{\mathcal{X}}(\ell'_j, x) \leq r + \epsilon_X\} \\
&\leq \inf_r \{r \mid \exists x \in \mathbf{X} \text{ s.t. } d_{\mathcal{X}}(\ell_i, x)d_{\mathcal{X}}(\ell_j, x) \leq r + \epsilon_X + \epsilon_L\} \\
&\leq \inf_r \{r \mid \exists x \in \mathbf{X} \text{ s.t. } d_{\mathcal{X}}(\ell_i, x), d_{\mathcal{X}}(\ell_j, x) \leq r\} + \epsilon_X + \epsilon_L \\
&\leq d_{WX}(\ell_i, \ell_j) + \epsilon_X + \epsilon_L
\end{aligned}$$

By reversing the roles of ℓ, ℓ' above, we obtain a bound

$$d_{WX}(\ell_i, \ell_j) \leq d_{WX'}(\ell'_i, \ell'_j) + \epsilon_X + \epsilon_L$$

Combining the two bounds yields the desired result

$$d_{\Omega}((L, d_{WX}), (L', d_{WX'})) = \sup_{(\ell_i, \ell'_i), (\ell_j, \ell'_j) \in \Omega} |d_{WX}(\ell_i, \ell_j) - d_{WX'}(\ell'_i, \ell'_j)| \leq \epsilon_X + \epsilon_L$$

□

Note that the above proof relied on the triangle inequality, so does not generally hold if $d_{\mathcal{X}}$ is not a metric.

Corollary 4.3.22. *Suppose that $d_H(X, X') \leq \epsilon_X$, and $d_H(L, L') \leq \epsilon_L$ in an ambient space with metric $d_{\mathcal{X}}$. Then*

$$d_I(PH_*(\mathcal{D}_1(L, X; r)), PH_*(\mathcal{D}_1(L', X'; r))) \leq \epsilon_L + \epsilon_X$$

Proof. This follows immediately from Proposition 4.3.21 applied to Theorem 4.3.16.

□

Corollary 4.3.22 states that the persistence diagram of the lazy Dowker filtration is stable both to perturbations of X and perturbations of the set L .

Dowker-Rips Interleaving

A fundamental question regarding the Dowker filtration on the pair (\mathbf{L}, \mathbf{X}) is how it can be related to a “full” Vietoris-Rips computation on the set \mathbf{X} . In this section, we answer this question via an interleaving between $\mathcal{D}_k(\mathbf{L}, \mathbf{X}; r)$ and $\mathcal{R}(\mathbf{X}; r)$.

One trivial interleaving (related to one presented for witness complexes in [37]) is based on the inclusions (assuming $\mathbf{L} \subset \mathbf{X}$)

$$\mathcal{D}_k(\mathbf{L}, \mathbf{X}; r) \subseteq \mathcal{R}(\mathbf{L}; 2r) \subseteq \mathcal{D}_k(\mathbf{L}, \mathbf{X}; 2r) \quad (4.26)$$

Unfortunately, this interleaving may have little to say about $\mathcal{R}(\mathbf{X}; r)$ if landmarks are not sampled in a representative way. For this reason, it is difficult to tell if the features in the witness complex capture features in the ambient space from Equation (4.26).

Suppose $A, B \subseteq X$ where X is a metric space with metric d_X . We use the notation $d_X(x, B) = \inf_{b \in B} d_X(x, b)$. We define $d_S(A, B) = \sup_{a \in A} d_X(a, B)$. Note that d_S is not symmetric, so it is a quasimetric. Note that the Hausdorff distance is $d_H(A, B) = \max\{d_S(A, B), d_S(B, A)\}$.

Let (\mathbf{L}, \mathbf{X}) be a landmark-space pair, let $\epsilon_1 = d_S(\mathbf{L}, \mathbf{X})$ and $\epsilon_2 = d_S(\mathbf{X}, \mathbf{L})$. Note that if $\mathbf{L} \subseteq \mathbf{X}$ then $\epsilon_1 = 0$, and generally $\epsilon_2 > 0$ if $\mathbf{L} \subsetneq \mathbf{X}$.

Lemma 4.3.23. *Let $(x_0, \dots, x_p) \in \mathcal{R}(\mathbf{X}, r)$. Then the carrier from the Rips complex to witness complex $\mathcal{D}_k(\mathbf{L}, \mathbf{X}; r)$ given by*

$$\mathcal{C}_{RW} : (x_0, \dots, x_p) \mapsto \left\langle \left\{ l \in \mathbf{L} \mid d_X(l, \{x_0, \dots, x_p\}) \leq \epsilon_2 \right\} \right\rangle$$

is ϵ_2 -simplicial.

Proof. By definition of ϵ_2 , \mathcal{C}_{RW} is not empty. Let $l \in \mathcal{C}(x_0, \dots, x_p)$. Then $d(l, x_i) \leq \epsilon_2$ for some $x_i \in (x_0, \dots, x_p)$. $d_X(x_i, x_j) \leq r$ for $x_i, x_j \in (x_0, \dots, x_p)$, because the simplex is in $\mathcal{R}(X, r)$. By triangle inequality, we have that $d(l, x_j) \leq r + \epsilon_2$ for any $x_j \in (x_0, \dots, x_p)$.

This means that any q -simplex $(l_0, \dots, l_q) \in \mathcal{C}(x_0, \dots, x_p)$ is witnessed by every vertex in (x_0, \dots, x_p) at level $r + \epsilon_2$, so the simplex is in $\mathcal{D}_k(\mathbf{L}, \mathbf{X}; r + \epsilon_2)$. Since this

holds for every q -simplex $q = 0, 1, \dots, k$, and the k -skeleton defines \mathcal{D}_k , the carrier forms the maximal simplex on the vertices of $\mathcal{C}(x_0, \dots, x_p)$, and thus the carrier is ϵ_2 -simplicial. \square

Lemma 4.3.24. *Let $(l_0, \dots, l_p) \in \mathcal{D}_k(\mathbf{L}, \mathbf{X}; r)$. Then the carrier to $\mathcal{R}(\mathbf{X}, r)$ given by*

$$\mathcal{C}_{WR} : (l_0, \dots, l_p) \mapsto \left\langle \{x \in \mathbf{X} \mid d_X(x, \{l_0, \dots, l_p\}) \leq \epsilon_1\} \right\rangle$$

is α -simplicial, where $\alpha : r \rightarrow 2r + 2\epsilon_1$.

Proof. Suppose (l_0, \dots, l_p) appears at parameter r . Let $l_i, l_j \in \{l_0, \dots, l_p\}$. Then there exists some $x \in \mathbf{X}$ so that $d_X(l_i, x) \leq r$ and $d_X(l_j, x) \leq r$, which implies $d_X(l_i, l_j) \leq 2r$ by triangle inequality. Now, let $x_i, x_j \in \mathcal{C}_{WR}(l_0, \dots, l_p)$. Then $d_X(x_i, l_i) \leq \epsilon_1$ and $d_X(x_j, l_j) \leq \epsilon_1$ for some $l_i, l_j \in \{l_0, \dots, l_p\}$. By triangle inequality, this means that

$$d_X(x_i, x_j) \leq d_X(l_i, l_j) + 2\epsilon_1 = 2r + 2\epsilon_1.$$

Because this holds for all $x_i, x_j \in \mathcal{C}_{WR}(l_0, \dots, l_p)$, the carrier is a simplex at level $\alpha(r) = 2r + 2\epsilon_1$. Thus, the carrier is α -simplicial. \square

Unfortunately, the compositions of these carriers do not carry the inclusion maps on one of the filtrations (needed for Proposition 4.3.11), unless $\epsilon_1 = \epsilon_2$. Thus, we need to construct acyclic carriers to show that the composition of augmentation-preserving maps between $\mathcal{D}_k(L, X; r)$ and $\mathcal{R}(X; r)$ are homotopic to the relevant inclusion maps.

Lemma 4.3.25. *Let $(\ell_0, \dots, \ell_p) \in \mathcal{D}_k(\mathbf{L}, \mathbf{X}; r)$. Then the carrier to $\mathcal{D}(\mathbf{L}, \mathbf{X}; r)$ given by*

$$\mathcal{C}_{WW} : (\ell_0, \dots, \ell_p) \mapsto \left\langle \{\ell \in L \mid d_X(\ell, \{\ell_0, \dots, \ell_p\}) \leq \epsilon_1 + \epsilon_2\} \right\rangle$$

is $(\alpha \circ \epsilon_2)$ -simplicial.

Proof. Note $\alpha \circ \epsilon_2 : r \mapsto 2r + 2\epsilon_1 + 2\epsilon_2$. Suppose that $m_0, \dots, m_1 \in \mathcal{C}_{WW}(\ell_0, \dots, \ell_p)$. Then there exists some $\ell_0, \dots, \ell_1 \in \ell_0, \dots, \ell_p$ such that $d(\ell_i, m_i) \leq \epsilon_1 + \epsilon_2$, $i = 0, \dots, q$. Because $(\ell_0, \dots, \ell_q) \in \mathcal{D}_k(\mathbf{L}, \mathbf{X}, r)$, there exists an x such that $d_X(x, \ell_i) \leq r$, $i = 0, \dots, q$. By triangle inequality, we have $d_X(x, m_i) \leq r + \epsilon_1 + \epsilon_2 \leq 2r + 2\epsilon_1 + 2\epsilon_2$.

Thus, (m_0, \dots, m_q) is witnessed by x at level $(\alpha \circ \epsilon_2)(r)$, and is in $\mathcal{D}_k(\mathbf{L}, \mathbf{X}; \alpha \circ \epsilon_2(r))$. Since this holds for all q -simplices $q = 0, \dots, k$ the carrier is $(\alpha \circ \epsilon_2)$ -simplicial. \square

Lemma 4.3.26. *Let $(x_0, \dots, x_p) \in \mathcal{R}(\mathbf{X}; r)$. Then the carrier to $\mathcal{R}(\mathbf{X}; r)$ given by*

$$\mathcal{C}_{RR} : (x_0, \dots, x_p) \mapsto \left\langle \{x \in X \mid d_X(x, \{x_0, \dots, x_p\}) \leq \epsilon_1 + \epsilon_2\} \right\rangle$$

is $(\epsilon_2 \circ \alpha)$ -acyclic.

Proof. Note $\epsilon_2 \circ \alpha : r \mapsto 2r + 2\epsilon_1 + \epsilon_2$. Fix $x \in \{x_0, \dots, x_p\}$. Suppose that $y \in \mathcal{C}_{RR}(x_0, \dots, x_p)$. Then there exists some $x' \in \{x_0, \dots, x_p\}$ such that $d(x', y) \leq \epsilon_1 + \epsilon_2$. Because $(x', x) \in \mathcal{R}(X; r)$, $d(x', x) \leq r$, and by triangle inequality, $d_X(y, x) \leq r + \epsilon_1 + \epsilon_2$. Thus $\mathcal{C}_{RR}(x_0, \dots, x_p)$ forms a cone with x at level $r + \epsilon_1 + \epsilon_2 \leq 2r + 2\epsilon_1 + \epsilon_2$, and so is $(\epsilon_2 \circ \alpha)$ -acyclic. \square

Theorem 4.3.27. *$PH_*(\mathcal{D}_k(\mathbf{L}, \mathbf{X}; r))$ and $PH_*(\mathcal{R}(\mathbf{X}; r))$ are (ϵ_2, α) -interleaved.*

Proof. First, note that $\mathcal{C}_{WR} \circ \mathcal{C}_{RW} \subseteq \mathcal{C}_{WW}$, and $\mathcal{C}_{RW} \circ \mathcal{C}_{WR} \subseteq \mathcal{C}_{RR}$, and furthermore \mathcal{C}_{WW} and \mathcal{C}_{RR} both carry the inclusion maps on $\mathcal{D}_k(\mathbf{L}, \mathbf{X}; r)$ and $\mathcal{R}(\mathbf{X}; r)$ respectively.

Now, because \mathcal{C}_{WR} is ϵ_2 -simplicial (thus ϵ_2 -acyclic), we can construct an augmentation-preserving map $F^{\epsilon_2} : C_*(\mathcal{D}_k(\mathbf{L}, \mathbf{X}; r)) \rightarrow C_*(\mathcal{R}(\mathbf{X}; r))$, and because \mathcal{C}_{RW} is α -simplicial (thus α -acyclic), we can construct an augmentation preserving map $G^\alpha : C_*(\mathcal{R}(\mathbf{X}; r)) \rightarrow C_*(\mathcal{D}_k(\mathbf{L}, \mathbf{X}; r))$. The composition $G^\alpha \circ F^{\epsilon_2}$ is carried by \mathcal{C}_{WW} , which is $(\alpha \circ \epsilon_2)$ -simplicial and carries the identity, so $G^\alpha \circ F^{\epsilon_2} \simeq I^{\alpha \circ \epsilon_2}$. Similarly, $F^{\epsilon_2} \circ G^\alpha$ is carried by \mathcal{C}_{RR} , which is $(\epsilon_2 \circ \alpha)$ -acyclic, and carries the identity so $F^{\epsilon_2} \circ G^\alpha \simeq I^{\epsilon_2 \circ \alpha}$. \square

In practice, this bound may be quite conservative, although triangle inequality bounds admit examples in which the factors are tight in Lemmas 4.3.23 and 4.3.24. This interleaving offers theoretical justification for the greedy landmarking procedure employed in [37], as it attempts to minimize the ϵ_2 term. Note that if we have $\epsilon_2 = \epsilon_1 = 0$ (i.e. if $d_H(\mathbf{L}, \mathbf{X}) = 0$), we obtain the Čech-Rips multiplicative 2 interleaving, as we expect, since Dowker complexes are motivated as a data-driven versions of the Čech complex.

4.4 Cover Complexes

In this section, we will introduce families of complexes that use a cover to constrain the appearance of simplices.

Definition 4.4.1. *Let \mathcal{X}^T be a filtered cell complex over a poset T , with vertex set $\mathcal{X}_0^T = X$, and let \mathcal{U} be a cover of X . We define the cover complex $\mathcal{X}^T(\mathcal{U})$ to be the restriction of \mathcal{X}^T to cells whose 0-skeleton lies in some $U \in \mathcal{U}$.*

We will specifically consider cover complexes on geometric constructions. We will denote the Vietoris-Rips cover complex as $\mathcal{R}(\mathbf{X}, \mathcal{U}; r)$, and Čech cover complex as $\check{\mathcal{C}}(\mathbf{X}, \mathcal{U}; r)$.

In particular, we will focus on Vietoris-Rips cover complexes $\mathcal{R}(\mathbf{X}, \mathcal{U}; r)$, and seek to answer the following questions:

1. For a fixed cover \mathcal{U} , how sensitive is $\mathcal{R}(\mathbf{X}, \mathcal{U}; r)$ to perturbations of the underlying data X ?
2. For a fixed dataset \mathbf{X} , how sensitive is $\mathcal{R}(\mathbf{X}, \mathcal{U}; r)$ to the choice of cover \mathcal{U} ?
3. How does $\mathcal{R}(\mathbf{X}, \mathcal{U}; r)$ relate to the full Vietoris-Rips complex $\mathcal{R}(\mathbf{X}; r)$?

While we will focus on Vietoris-Rips cover complexes, similar results may be obtained for other geometric constructions.

Our definition of cover complex (Definition 4.4.1) coincides with a similar definition which appeared in an early pre-print of [55], but which was abandoned in subsequent versions. The goal of [55], as well as associated literature [27, 32] is to understand when a filtered nerve can effectively be used to approximate a larger computation, a question which we will address for cover complexes in Section 4.4.1. In contrast, we will seek to use the actual cover complex in computations in situations where the complex restricted to each set is not necessarily close to acyclic, which we will investigate in Sections 4.4.2 and 4.4.3. For instance, in Chapter 5, we will see a situation in which fibers may lie near a circle.

Another related definition is the *Rips system* found in Yoon's 2018 dissertation [104] which is used for distributed computation of persistent homology of Rips

complexes via cellular (co)-sheaves. Yoon shows that if the Nerve is 1-dimensional, and the system covers the full Rips complex, that the Rips system can be used to obtain the Homology of the full complex, and develops a distributed algorithm for computation. We will consider more general coverings, and characterize regimes where the cover complex and full complex are interleaved, but not identical. Distribution schemes for computing persistent homology of cover complexes in their full generality are beyond the scope of this work.

Finally, there are some similarities between the complexes we will see and the multiscale mapper construction [41]. This also uses inverse images of sets in covers, but applies this to simplicial complexes generated using the mapper algorithm [92], and is only measures connected components in the inverse image of sets.

4.4.1 A Generalized Nerve Theorem

We'll now prove a version of the Nerve theorem for cover complexes. This result can be viewed as a special case of of the approximate nerve theorems in [27, 55]. While our proof is narrower in scope than the aforementioned results, the use of carriers will considerably simplify the proof, compared to [55] which used the Mayer-Vietoris spectral sequence, and [27] which used a construction using the blowup complex.

Theorem 4.4.2. *[an α -Acyclic Nerve Theorem] Let \mathcal{U} be a cover of a vertex set X , and let $\mathcal{X}^T(\mathcal{U})$ be a simplicial cover complex, with T a strict order with initial object 0 . If $\mathcal{X}^T(V)$ is α -acyclic for every $V \in \bar{\mathcal{U}}$, then $H_k(\mathcal{N}(\mathcal{U}))$ and $H_k(\mathcal{X}^T(\mathcal{U}))$ are $(\alpha^{k+1}, \text{id})$ -interleaved.*

Proof. We'll construct an interleaving with $\mathcal{N}(\bar{\mathcal{U}})$, which has isomorphic homology to $\mathcal{N}(\mathcal{U})$ by Proposition 4.1.34.

We'll first define a carrier $\mathcal{D} : \mathcal{N}(\bar{\mathcal{U}}) \rightarrow \mathcal{X}^T$. We take $\mathcal{D}(V) = \mathcal{X}^T(V)$, and $\mathcal{D}(V_0, \dots, V_k) = \mathcal{X}^T(V_0 \cup \dots \cup V_k) = \mathcal{X}^T(V_{i_k})$, where V_{i_k} is the maximal set in $\{V_0, \dots, V_k\}$. This forms a $(0, \alpha)$ -acyclic carrier by assumption, where 0 denotes the map to the initial object of T .

Now, we define a carrier $\mathcal{C} : \mathcal{X}^T(\mathcal{U}) \rightarrow \mathcal{N}(\bar{\mathcal{U}})$. We take

$$\mathcal{C}(x_0, \dots, x_k) = \left\langle \left\{ \bigcap_{V \supseteq S} V \right\}_{S \subseteq \{x_0, \dots, x_k\}} \right\rangle \quad (4.27)$$

Let $V' = \bigcap_{V \supseteq \{x_0, \dots, x_k\}} V$. The carrier above forms a cone with V' , so is acyclic.

$\mathcal{D} \circ \mathcal{C}$ carries the identity because Equation (4.27) ensures that some V' for which $\{x_0, \dots, x_k\} \subseteq V'$ is included in $\mathcal{C}(x_0, \dots, x_k)$, and $\mathcal{D}(V) \ni (x_0, \dots, x_k)$ for that V . Because all other sets in Equation (4.27) are contained in V' , $\mathcal{D} \circ \mathcal{C}(x_0, \dots, x_k) = \mathcal{D}(V')$, which is $(0, \alpha)$ -acyclic by assumption.

Any $(x_0, \dots, x_k) \in \mathcal{D}(V)$, implies $\{x_0, \dots, x_k\} \subseteq V$. Thus, every V_i generating the carrier in Equation (4.27) satisfies $V_i \subseteq \mathcal{C}(x_0, \dots, x_k)$. We can define $\mathcal{A}(V)$ to be the star of V inside $\mathcal{N}(\bar{\mathcal{U}})$. This carrier is acyclic because it forms a cone with the vertex for V , and contains $\mathcal{C} \circ \mathcal{D}(V)$. For $(V_0, \dots, V_k) \in \mathcal{N}(\bar{\mathcal{U}})$, we take $\mathcal{A}(V_0, \dots, V_k) = \mathcal{A}(V_{i_k})$, where V_{i_k} is the maximal set in the simplex. Again, this carrier is acyclic and carries the identity.

We have now constructed carriers for maps in the following diagram

$$\begin{array}{ccc} \mathcal{X}^T(\mathcal{U}) & \xleftrightarrow{\quad} & \mathcal{X}^T(\mathcal{U}) \\ \downarrow \mathcal{C} & \nearrow \mathcal{D} & \downarrow \mathcal{C} \\ \mathcal{N}(\bar{\mathcal{U}}) & \xrightarrow{\mathcal{A}} & \mathcal{N}(\bar{\mathcal{U}}) \end{array} \quad (4.28)$$

We can now construct a map $P_* : C_*(\mathcal{X}^T(\mathcal{U})) \rightarrow C_*(\mathcal{N}(\bar{\mathcal{U}}))$ carried by \mathcal{C} by applying Proposition 4.1.13. We can also construct maps $F_i^{\alpha^i} : C_i(\mathcal{N}(\bar{\mathcal{U}})) \rightarrow C_i(\mathcal{X}^{\alpha^i(0)})$ using Theorem 4.3.6, where $\partial_i F_i^{\alpha^i} x = F_{i-1}^{\alpha^{i-1}} \partial_i x$, which we need to construct for $i = 0, \dots, k+1$. Because $\mathcal{D} \circ \mathcal{C}$ carries the inclusion, we can construct a homotopy, but only after increasing the grade by an extra factor of α in each dimension i , $I^\alpha F_i^{\alpha^i} \circ P_i \simeq I_i^{\alpha^{i+1}}$. In order to compute induced maps on homology for H_i , we only need to extend the chain homotopy up to dimension i (see Lemma 2.4.1). On homology, we have $\tilde{I}^\alpha \tilde{F}_k^{\alpha^k} \tilde{P}_k \cong \tilde{I}_k^{\alpha^{k+1}}$.

Finally, because \mathcal{A} is acyclic and carries $P_* \circ F_*$ as well as the inclusion, we have $\tilde{P}_k \circ I^\alpha \tilde{F}^{\alpha^k} \simeq I$, we have constructed a $(\alpha^{k+1}, \text{id})$ -interleaving. \square

Note that for Vietoris-Rips cover complexes as well as other geometric complexes, that there will be some parameter $t \in T$ at which $\mathcal{X}^T(V)$ will be acyclic for all V , when $\mathcal{X}^T(V)$ forms the maximal simplex on its vertex set. At this point, the cover complex and nerve are homotopic by the standard nerve theorem (Theorem 1.2.3).

Corollary 4.4.3. *Let \mathcal{U} be a cover of X , where $\mathcal{X}^T(\mathcal{U})$ satisfies the conditions of Theorem 4.4.2. Then if $\mathcal{N}(\mathcal{U})$ is acyclic, $H_k(\mathcal{X}^T(\mathcal{U}))$ is (α^{k+1}) -acyclic.*

Proof. This follows because if $\mathcal{N}(\mathcal{U})$ is acyclic, then the interleaving implies that $\mathcal{X}^T(\mathcal{U})$ is α^{k+1} -acyclic. \square

4.4.2 Local Interleavings

We now consider the more general situation, in which cover complexes are not necessarily close to acyclic when restricted to inverse images of open sets in the cover.

Definition 4.4.4. *Let $\mathcal{X}^S(\mathcal{U})$, $\mathcal{Y}^T(\mathcal{U})$ be cover complexes over a cover \mathcal{U} . A system of carriers $\mathcal{C}(\mathcal{U}) : \mathcal{X}^S(\mathcal{U}) \rightarrow \mathcal{Y}^T(\mathcal{U})$ consists of carriers $\mathcal{C}(U) : \mathcal{X}^S(U) \rightarrow \mathcal{Y}^T(U)$ for each $U \in \mathcal{U}$. We say the system of carriers is compatible if $\cap U_k \neq \emptyset$ implies $\mathcal{C}(U_i) |_{\cap U_k} = \mathcal{C}(U_j) |_{\cap U_k}$ for all $U_i, U_j \in \{U_k\}$.*

In general, $U \in \mathcal{U}$ need not cover the same points in X and Y (denoting the vertex sets of \mathcal{X} , \mathcal{Y} respectively). We can alternatively think of it as an identification of sets in covers of each vertex set, or a set in a cover of the disjoint union $X \sqcup Y$.

When the system of carriers is compatible, we can extend the defined carriers to intersections via $\mathcal{C}(\cap U_k) = \mathcal{C}(U_i) |_{\cap U_k}$ for $U_i \in \{U_k\}$. We'll say a compatible system of carriers is α -acyclic if $\mathcal{C}(\cap U_k)$ is α -acyclic for all $\{U_k\} \subset \mathcal{U}$ where $\cap U_k \neq \emptyset$.

Note that we can define a carrier $\mathcal{C}_{\mathcal{U}} : \mathcal{X}^T(\mathcal{U}) \rightarrow \mathcal{Y}^S(\mathcal{U})$ from a compatible system of carriers via $\mathcal{C}_{\mathcal{U}}(x) = \mathcal{C}(\cap\{U \ni x\})(x)$. When a compatible system of carriers $\mathcal{C}(\mathcal{U})$ is α -acyclic, $\mathcal{C}_{\mathcal{U}}$ is also α -acyclic through application of the definition. The advantage of using a compatible system of carriers $\mathcal{C}(\mathcal{U})$ instead of the global carrier $\mathcal{C}_{\mathcal{U}}$ is that we only need to check conditions locally in the cover.

Proposition 4.4.5. *Let \mathcal{U} be a finite cover. Suppose $\mathcal{C}(\mathcal{U}) : \mathcal{X}^S(\mathcal{U}) \rightarrow \mathcal{Y}^T(\mathcal{U})$ is an α -acyclic compatible system of carriers, and $\mathcal{D}(\mathcal{U}) : \mathcal{Y}^T(\mathcal{U}) \rightarrow \mathcal{X}^S(\mathcal{U})$ is a β -acyclic*

compatible system of carriers. Furthermore suppose that for each $V = \cap U_k \neq \emptyset$, that $\mathcal{C}(V) \circ \mathcal{D}(V)$ is $(\alpha \circ \beta)$ -acyclic and carries the identity, and $\mathcal{D}(V) \circ \mathcal{C}(V)$ is $(\beta \circ \alpha)$ -acyclic and carries the identity. Then there exists an (α, β) -interleaving of \mathcal{X}^S and \mathcal{Y}^T which is locally carried by each carrier in the system.

Proof. This follows by constructing the global carriers $\mathcal{C}_{\mathcal{U}} : \mathcal{X}^S(\mathcal{U}) \rightarrow \mathcal{Y}^T(\mathcal{U})$ and $\mathcal{D}_{\mathcal{U}} : \mathcal{Y}^T(\mathcal{U}) \rightarrow \mathcal{X}^S(\mathcal{U})$, and noting that because the composite $\mathcal{C}_{\mathcal{U}} \circ \mathcal{D}_{\mathcal{U}}$ is $(\alpha \circ \beta)$ -acyclic locally and carries the identity locally, it satisfies these properties globally. Similarly, $\mathcal{D}_{\mathcal{U}} \circ \mathcal{C}_{\mathcal{U}}$ is $(\beta \circ \alpha)$ -acyclic and carries the identity. We can then apply Proposition 4.3.11 to obtain the result. \square

Proposition 4.4.5 can be used to extend the results in Section 4.3.4 to cover complexes. We will focus on how cover complexes behave with respect to perturbations of the data

Corollary 4.4.6. *Let \mathbf{X}, \mathbf{Y} be samples from a metric space (X, d_X) , and let \mathcal{U} be a cover of $\mathbf{X} \sqcup \mathbf{Y}$ such that $d_H(V|_{\mathbf{X}}, V|_{\mathbf{Y}}) \leq \epsilon$ for all $V \in \bar{\mathcal{U}}$. Then $\mathcal{R}(\mathbf{X}, \mathcal{U}; r)$ and $\mathcal{R}(\mathbf{Y}, \mathcal{U}; r)$ are 2ϵ -interleaved.*

Proof. Let $V_x = \bigcap \{U \in \mathcal{U} \mid U \ni x\}$. By assumption $d_H(V_x|_{\mathbf{X}}, V_x|_{\mathbf{Y}}) \leq \epsilon$, so there must exist some $y \in V_x$ so that $d_X(x, y) \leq \epsilon$. Let $\Omega \subseteq X \times Y$ be the left-total relation $\Omega(x) = \{y \in V_x \cap Y \mid d_X(x, y) \leq \epsilon\}$. Then the induced carrier $\mathcal{C}_{\Omega} : \mathcal{R}(\mathbf{X}, \mathcal{U}; r) \rightarrow \mathcal{R}(\mathbf{Y}, \mathcal{U}; r)$ is 2ϵ -simplicial. Similarly, using the right-total relation $\Psi \subseteq X \times Y$, with $\Psi(y) = \{x \in V_y \cap X \mid d_X(x, y) \leq \epsilon\}$, we obtain a 2ϵ -simplicial carrier $\mathcal{D}_{\Psi} : \mathcal{R}(\mathbf{Y}, \mathcal{U}; r) \rightarrow \mathcal{R}(\mathbf{X}, \mathcal{U}; r)$.

Now, note that the composite carrier $\mathcal{D}_{\Psi} \circ \mathcal{C}_{\Omega}$ need not carry the identity, because $y \in V_x$ does not imply $x \in V_y$. However, $y \in V_x$ implies does imply that $V_y \subseteq V_x$ which combined with the Hausdorff distance bound implies there must exist some $x' \in V_y \cap \mathbf{X}$ such that $d_X(x', y) \leq \epsilon$, which implies $d_X(x', x) \leq 2\epsilon$ by triangle inequality. We can define a left-total relation $\Omega' \subseteq \mathbf{X} \times \mathbf{X}$, with $\Omega' = \{(x, x') \mid d(x, x') \leq 2\epsilon, x' \in V_x\}$, which is nonempty, and 4ϵ -simplicial by triangle inequality. Furthermore, the carrier $\mathcal{A}_{\Omega'}$ contains the composite $\mathcal{D}_{\Psi} \circ \mathcal{C}_{\Omega}$ and carries the identity. Similarly, we can define a relation $\Psi' \subseteq \mathbf{Y} \times \mathbf{Y}$ with $\Psi' = \{(y, y') \mid d_X(y, y') \leq 2\epsilon\}$ which produces a 4ϵ -simplicial carrier $\mathcal{B}_{\Psi'}$ which contains the composite $\mathcal{C}_{\Omega} \circ \mathcal{D}_{\Psi}$ and carries the identity.

We can now apply Proposition 4.3.11 to obtain the result. \square

4.4.3 Interleaving with Full Complex

We will now turn to relating the persistent homology of $\mathcal{R}(\mathbf{X}, \mathcal{U}; r)$ to the persistent homology of $\mathcal{R}(\mathbf{X}; r)$. At large r parameters, Vietoris-Rips complexes become acyclic, so we see from Theorem 4.4.2 that $PH_*(\mathcal{R}(\mathbf{X}, \mathcal{U}; r))$ will eventually converge to $H_*(\mathcal{N}(\mathcal{U}))$. This means that unless $\mathcal{N}(\mathcal{U})$ is acyclic, $PH_*(\mathcal{R}(\mathbf{X}, \mathcal{U}; r))$ and $PH_*(\mathcal{R}(\mathbf{X}; r))$ can not possibly interleave for sufficiently large r parameters. However, in situations in which the pullbacks of open sets in the cover have non-trivial structure, we would like to understand how this structure relates to the full filtration $\mathcal{R}(\mathbf{X}; r)$, particularly for small values of r .

Definition 4.4.7. *Let (X, d) be a dissimilarity space. We define*

$$d(x_0, \dots, x_k) = \max_{0 \leq i < j \leq k} d(x_i, x_j) \quad (4.29)$$

Because the Rips filtration is a flag filtration, the simplex (x_0, \dots, x_k) appears at parameter $d(x_0, \dots, x_k)$.

Proposition 4.4.8. *Let \mathcal{U} be a cover of \mathbf{X} with dissimilarity d_X . Suppose that for some $R \geq 0$, if $d_X(x_0, \dots, x_k) \leq R$, then there exists some $U \in \mathcal{U}$ so that $x_0, \dots, x_k \in U$. Then $\mathcal{R}(\mathbf{X}, \mathcal{U}; r) = \mathcal{R}(\mathbf{X}; r)$ for all $r \leq R$.*

Proof. This follows because if $(x_0, \dots, x_k) \in \mathcal{R}(\mathbf{X}; r)$, then $d(x_0, \dots, x_k) \leq r \leq R$, so $(x_0, \dots, x_k) \in \mathcal{R}(\mathbf{X}, U; r) \subseteq \mathcal{R}(\mathbf{X}, \mathcal{U}; r)$ for some $U \in \mathcal{U}$. Thus $\mathcal{R}(\mathbf{X}; r) \subseteq \mathcal{R}(\mathbf{X}, \mathcal{U}; r)$, and we already know $\mathcal{R}(\mathbf{X}, \mathcal{U}; r) \subseteq \mathcal{R}(\mathbf{X}; r)$, giving equality. \square

This means that covers \mathcal{U} that encode some notion of locality will produce cover complexes which are identical to the full Rips complex at the beginning of the filtration. When the condition of Proposition 4.4.8 ceases to hold, we can still hope to construct an interleaving even when we can not have equality of the complexes.

Because there are inclusions $\mathcal{R}(\mathbf{X}, \mathcal{U}; r) \hookrightarrow \mathcal{R}(\mathbf{X}; r)$, it suffices to study under what conditions we can extend a map f_α in the diagram

$$\begin{array}{ccc}
 \mathcal{R}(\mathbf{X}, \mathcal{U}; r) & \hookrightarrow & \mathcal{R}(\mathbf{X}, \mathcal{U}; \alpha(r)) \\
 \downarrow & \nearrow f_\alpha & \downarrow \\
 \mathcal{R}(\mathbf{X}; r) & \hookrightarrow & \mathcal{R}(\mathbf{X}; \alpha(r))
 \end{array} \tag{4.30}$$

We will focus on a carrier generated from witness sets

$$\mathbf{X}(x_0, \dots, x_k) = \{y \in \mathbf{X} \mid d(y, x_i) \leq d(x_0, \dots, x_k) \forall i = 0, \dots, k\} \tag{4.31}$$

and their union, denoted

$$\bar{\mathbf{X}}(x_0, \dots, x_k) = \bigcup_{S \in \mathcal{P}(\{x_0, \dots, x_k\})} \mathbf{X}(S) \tag{4.32}$$

where \mathcal{P} denotes the power set. We define the carrier $\mathcal{C} : \mathcal{R}(\mathbf{X}; r) \rightarrow \mathcal{R}(\mathbf{X}, \mathcal{U}; r)$ via

$$\mathcal{C} : (x_0, \dots, x_k) \mapsto \langle \bar{\mathbf{X}}(x_0, \dots, x_k) \rangle \tag{4.33}$$

We'll let α denote the non-decreasing map for which \mathcal{C} is α -acyclic.

Let $\iota : \mathcal{R}(\mathbf{X}, \mathcal{U}; r) \rightarrow \mathcal{R}(\mathbf{X}; r)$ denote the canonical inclusion, seen in Equation (4.30). Clearly, $\mathcal{C} \circ \iota$ carries the inclusion $\mathcal{R}(\mathbf{X}, \mathcal{U}; r) \rightarrow \mathcal{R}(\mathbf{X}, \mathcal{U}; \alpha(r))$.

However, the carrier $\iota \circ \mathcal{C}$ does not carry the inclusion for any simplices in $\mathcal{R}(\mathbf{X}; r)$ that are not in the cover complex $\mathcal{R}(\mathbf{X}, \mathcal{U}; r)$. We need to find another carrier which does carry the inclusion which also contains this carrier. Consider

$$\mathcal{D} : (x_0, \dots, x_k) \mapsto \langle \bar{\mathbf{X}}(x_0, \dots, x_k) \rangle \tag{4.34}$$

The difference between \mathcal{C} and \mathcal{D} , despite the similarity of their definitions is that they map to different complexes. \mathcal{C} maps to subcomplexes of $\mathcal{R}(\mathbf{X}, \mathcal{U}; r)$, and \mathcal{D} maps to subcomplexes of $\mathcal{R}(\mathbf{X}; r)$. Note that \mathcal{D} does carry $\iota \circ \mathcal{C}$.

If \mathcal{D} is also α -acyclic, we could apply Proposition 4.3.11 to construct the interleaving. The remainder of this section will describe conditions that will allow us to bound the non-decreasing map α .

Proposition 4.4.9. *If (\mathbf{X}, d) is a metric space, then \mathcal{D} is acyclic for $\alpha : r \mapsto 2r$.*

Proof. Consider $\mathcal{D}(x_0, \dots, x_k)$, and let $r = d_X(x_0, \dots, x_k)$. Without loss of generality, consider distances to x_0 . Let $y \in \mathcal{D}(x_0, \dots, x_k)$. By definition of \mathcal{D} , either $d(y, x_0) \leq r$, or $d(y, x_i) \leq r$ for some $x_i \in \{x_1, \dots, x_k\}$. Because $d_X(x_0, x_i) \leq r$, by triangle inequality, $d(y, x_0) \leq 2r$. Because the Vietoris-Rips complex is a flag complex, this implies $\mathcal{D}(x_0, \dots, x_k)$ forms a cone with x_0 , and so is acyclic. \square

The more difficult carrier to analyze is \mathcal{C} . We'll consider the restriction of the cover to the carrier. Let

$$\bar{\mathcal{U}}(x_0, \dots, x_k) = \{V \cap \bar{\mathbf{X}}(x_0, \dots, x_k) \mid V \in \bar{\mathcal{U}}, \bar{\mathbf{X}}(x_0, \dots, x_k) \cap V \neq \emptyset\} \quad (4.35)$$

which covers $\mathcal{C}(x_0, \dots, x_k)$. If $\mathcal{N}(\bar{\mathcal{U}}(x_0, \dots, x_k))$ is acyclic for each $(x_0, \dots, x_k) \in \mathcal{R}(\mathbf{X}; r)$, and $\mathcal{R}(\bar{\mathbf{X}}(x_0, \dots, x_k), V; r)$ is α -acyclic, then we can apply corollary 4.4.3 to see that \mathcal{C} is α -acyclic.

Proposition 4.4.10. *Let $r = d_X(x_0, \dots, x_k)$. For each $V \in \bar{\mathcal{U}}(x_0, \dots, x_k)$, $\mathcal{R}(V; 3r)$ is acyclic.*

Proof. Let $y, y' \in V$. Then there is some $x, x' \in \{x_0, \dots, x_k\}$ for which $d(y, x), d(y', x') \leq r$. Because $d(x, x') \leq r$, by triangle inequality $d(y, y') \leq 3r$. Thus, $\mathcal{R}(V; 3r)$ forms a simplex, so is acyclic. \square

In general, the bound in Proposition 4.4.10 will be fairly pessimistic. For instance,

Proposition 4.4.11. *Let $r = d_X(x_0, \dots, x_k)$, and suppose that $\mathbf{X}(x_0, \dots, x_k) \cap V$ is non-empty for each $V \in \bar{\mathcal{U}}(x_0, \dots, x_k)$. Then $\mathcal{R}(V; 2r)$ is acyclic.*

Proof. Fix $V \in \bar{\mathcal{U}}$. By assumption, there is some $y \in V$ so that $d(y, x_i) \leq r$ for all $i = 0, \dots, k$. For some other $y' \in V$, we have $d(y, x_i) \leq r$ for some $i = 0, \dots, k$. By triangle inequality, $d(y, y') \leq 2r$. Since this holds for all $y' \in V$, $\mathcal{R}(V; 2r)$ forms a cone with y , and is thus acyclic. \square

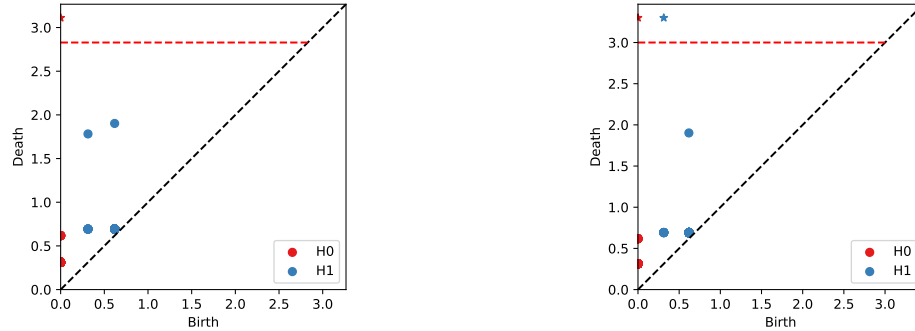


Figure 4.1: Data lies on a 20×10 grid on a flat torus. Left: Persistence diagram for Rips filtration. Right: Persistence diagram for Rips cover filtration. Cover is obtained by a pullback of a cover on the first circular coordinate.

The preceding discussion can be summarized in the following theorem:

Theorem 4.4.12. *Suppose that for some R , if $d(x_0, \dots, x_k) \leq R$ then $\mathcal{N}(\bar{\mathcal{U}}(x_0, \dots, x_k))$ is acyclic. Then $\mathcal{R}(\mathbf{X}, \mathcal{U}; r)$ and $\mathcal{R}(\mathbf{X}; r)$ are $(\text{id}, 3r)$ -interleaved.*

Note that the sets in the covers \mathcal{U} do not need to be acyclic at the levels prescribed, but rather their restriction to points within a certain distance of each simplex. This means there can be a fair amount of structure in the open sets of the cover.

In Figure 4.1, we see persistence diagrams for both a Rips filtration and a Rips cover filtration on a flat torus, where the cover is pulled back from a cover of the first circular coordinate. In contrast to the Rips filtration, the cover filtration has a persistent H_1 class, due to the Nerve of the cover. For parameters before the death of this H_1 class in the Rips complex, the two are tightly interleaved. The cover complex was able to significantly reduce the number of simplices formed: only 23,500 simplices were formed in the cover filtration, compared to 1.3 million in the full Rips filtration.

4.4.4 Finding Covers

There are any number of ways to come up with covers of a data set. We are primarily interested in using covers in which sets encode some notion of neighborhood with respect to a similarity (see Proposition 4.4.8). In continuous spaces, these covers

might take the form of metric balls or Voronoi cells. We will use an approach to obtaining covers based on a landmark set, which works in arbitrary dissimilarity spaces, Algorithm 10.

Algorithm 10 Obtain a cover of \mathbf{X} using landmark set \mathbf{L} .

```

1: Input: data set  $\mathbf{X} \subset X$ , landmark set  $\mathbf{L} \subset X$ , dissimilarity  $d$  defined on  $\mathcal{X}$ , and
   integer parameter  $k$ .
2: Result: Cover  $\mathcal{U}_{\mathbf{L}} = \{U_{\ell}\}$  of  $\mathbf{X}$  in which each point appears in  $k$  open sets.
3: for  $\ell \in \mathbf{L}$  do
4:    $U_{\ell} \leftarrow \emptyset$ 
5: end for
6: for  $x \in X$  do
7:   for  $k$  closest  $\ell \in \mathbf{L}$  to  $x$  do
8:      $U_{\ell} \leftarrow U_{\ell} \cup \{x\}$ 
9:   end for
10: end for

```

Of course, the topological quality of a cover obtained from Algorithm 10 will depend on the choice of landmark set \mathbf{L} , and the choice of k . As with witness and Dowker complexes, the choice of \mathbf{L} may drastically affect the outcome. If we assume that \mathbf{X} is sampled near a low-dimensional space \mathcal{X} , one desirable property would be to have $\mathcal{N}(\mathcal{U}_{\mathbf{L}}) \simeq \mathcal{X}$. Heuristically, we would expect that \mathbf{L} should landmark different regions of \mathbf{X} , and that if \mathcal{X} is d -dimensional, that we should choose $k = d + 1$ to ensure that we have d -dimensional simplices in the nerve.

4.5 Future Directions

In this section, we developed a filtered version of the acyclic carrier theorem, which allowed us to construct interleavings between different geometric constructions. We have presented a few results, and we anticipate that the use of filtered carriers has broad potential as a technique to construct interleavings in situations that we have not yet considered. In this chapter, we have focused on algebraic interleavings, and many

of these results could potentially be extended to homotopy interleavings [12] given additional care when constructing carriers of cell complexes. Another interesting line of future investigation would be to use the algorithmic construction of maps from carriers (Algorithm 9) in data analysis. This could potentially be used, for instance, in constructing low dimensional embeddings of data that minimize the interleaving distance between a filtration on the higher-dimensional point cloud and the embedded point cloud.

One of the potential appeals of cover complexes, beside their ability to reduce the number of simplices, is their potential use for distributed computation. As discussed previously, a limited version of this was explored in [104], and our interleaving results expand the potential use of cover complexes to more general settings. We also believe that the interleaving bounds we derive are likely pessimistic in many situations where data has additional structure. Analyses of these situations may help tighten our bounds considerably.

Chapter 5

Models for Image Patch Data

We'll now return to the topic which was introduced in Section 1.2. Contributions of this chapter include an explanation of *why* a Klein bottle appeared to be a good model for image patches in [25] in terms of a fibration. We will then describe the topology of models of other types of image patch data in terms of similar fibrations, including d -dimensional voxel cubes. We will also propose a framework for data analysis, particularly in building complexes on point cloud data, that takes into account the structure of a map, and apply this to patch-based examples we develop.

5.1 Preliminaries

5.1.1 Processing Image Patches

We will first review how a dataset of image patches was created from a dataset of natural images [101], first in [66], and subsequently in the topological analyses [25,37].

The image data set we will discuss was produced by van Hateren and van der Schaff [101], which consists of 4212 digital (1024×1536 pixel) black and white natural images, meaning images of natural scenes, not digitally generated or altered aside from camera lens calibration via an inverse filtering procedure. The pixel values are approximately proportional to scene luminance, discretized into 8-bit bins. This data set was used in [101] to study independent component analysis of 18×18 patches, in



Figure 5.1: Image 400 from the Van Hateren dataset [101]. An image patch is a $k \times k$ pixel sub-image, such as the one indicated in red.

order to compare with the sensitivity of simple cells in the visual cortex. This data set was then used by Lee, Pedersen, and Mumford [66], which studied 3×3 image patches in order to compute empirical probability densities of regions of a 7-sphere obtained by considering the subspace spanned by the eight non-constant eigenvectors of the 3×3 discrete cosine transform equipped with the angular distance. One of the conclusions of [66] was that these patches appeared to concentrate around a low-dimensional subspace of the 7-sphere, specifically an annulus thickening what would later be called the primary circle. The method of producing image patches in [66] was subsequently used in [25, 37], which produced a data set $\mathbf{X}(n, \ell, q) \subset \mathbb{R}^{n \times |\ell|}$ via the following procedure

1. transform images by taking the logarithm of the pixel luminance value, in order to normalize the effect of large numeric fluctuations in high intensity regions
2. sample n patches of size $\ell = (\ell_1, \ell_2, \dots)$ uniformly at random from the transformed images. Each patch is considered as a vector $x \in \mathbb{R}^{|\ell|}$, where $|\ell| = \prod_i \ell_i$.
3. mean-center each patch: $x \mapsto x - \sum x_i$
4. take the top q percent (typically $q = 20\%$) of patches with highest L -norm (Equation (5.1)).
5. normalize remaining patches by their L -norm $x \mapsto x / \|x\|_L$

We define the L -(semi-)norm on an image patch $x \in \mathbb{R}^k$ as

$$\|x\|_L = \sqrt{\sum_{i \sim j} (x_i - x_j)^2} = \sqrt{x^T L x} \quad (5.1)$$

where $i \sim j$ if index i and index j are adjacent (horizontally or vertically) on the $k_1 \times k_2$ pixel grid, and L is the (un-normalized) graph Laplacian of the graph G defined by this adjacency relation. The matrix L is called D in [25, 37, 66], but we will continue to use L to highlight the connection with the graph Laplacian. We illustrate the graph G defined by the adjacency relation for patches $x \in \mathbb{R}^9$, taken as flattening a 3×3 patch in column-major order in Equation (5.2).

$$\begin{array}{ccccc} x_1 & \text{---} & x_4 & \text{---} & x_7 \\ | & & | & & | \\ x_2 & \text{---} & x_5 & \text{---} & x_8 \\ | & & | & & | \\ x_3 & \text{---} & x_6 & \text{---} & x_9 \end{array} \quad (5.2)$$

It is well known that graph Laplacians are positive semi-definite operators, with an eigenspace with zero-dimensional eigenvalue spanned by constant functions (vectors) on connected components on the graph [34]. This makes $\|\cdot\|_L$ a semi-norm, where $\|x\|_L = 0$ for constant $x \neq 0$. Because G is the cartesian product of two line-graphs on k_1 and k_2 nodes respectively, eigenvectors v of L can be expressed in terms of the Kronecker product of eigenvectors on the line graphs $v_{i,j}^G = v_i^{k_1} \otimes v_j^{k_2}$. Transformation of a vector x into this basis (assuming eigenvectors are chosen to have unit norm) on $k_1 \times k_2$ images is known as the discrete cosine transform [3].

Topological investigations into the space of image patches [25, 37] both produced a data set of 3×3 image patches as in [66], but added additional processing steps to focus on regions of high density in $\mathbf{X}(n, \ell, q)$. The result was a data set $\mathbf{X}(n, \ell, q, k, p) \subset \mathbf{X}(n, \ell, q)$ obtained through the following steps:

5. Compute distances to k -nearest neighbors in the remaining set of patches in $\mathbf{X}(n, \ell, q)$.

6. Take the bottom p percent of patches sorted by increasing distance to the ℓ -th nearest neighbor.

5.1.2 A Klein bottle near image patches

In [37] $\mathbf{X}(n, \ell, q)$ was produced by randomly subsampling a large data set with $\ell = (3, 3)$, and $q = 20$ to produce a data set size of 5×10^4 . Further filtering was applied by varying $k = 15, 100, 300$ and $p = 10, 20, 30$, and witness complexes were constructed which revealed the primary and secondary circles in Figures 1.3 and 1.4, stitched together in a *three-circle* model.

In [25], a model Klein bottle was proposed which included the three-circle model as a 1-dimensional skeleton. While the homology of the Klein bottle (see Section 1.2.3) was not observed in the raw data, samples from this model were used to augment the data, and the Klein bottle homology was preserved for a range of persistence parameters, implying that the data retracted onto a space generated by the augmented data points. The model Klein bottle was generated from a family of polynomials P in two variables (x, y) parameterized by the torus $S^1 \times S^1$

$$P : (a, b, c, d) \mapsto c(ax + by)^2 + d(ax + by) \quad (5.3)$$

where $(a, b) \in S^1 \subset \mathbb{R}^2$, $(c, d) \in S^1 \subset \mathbb{R}^2$ lie on unit circles. These polynomials were evaluated on a 3×3 pixel index grid $(x, y) \in \{-1, 0, 1\}^2$ (note the scale is arbitrary due to the normalization procedure applied to patches). It was shown that there was an identification of points $P(a, b, c, d) = P(-a, -b, c, -d)$ corresponding to an identification on the torus that obtains the Klein bottle.

5.1.3 Three-dimensional Images

In the following sections, we will consider spaces of higher dimensional image patches. Our experiments will focus on three-dimensional images, given the practical difficulty of obtaining images in higher dimensions. We will focus on dense three-dimensional images, obtained through medical and seismic imaging which reveal the whole volume,

as opposed to range images or three-dimensional meshes, which can be used to construct (two-dimensional) animations, but which typically do not represent anything inside an object that would not be visible.

We will look at two sources of three dimensional images. First, images from the BRATS MRI database [5–7, 76], which consists of MRI images of human brains. Second, we will use the Penobscot seismic dataset [8], which consists of a single image of a subsurface volume collected off the coast of Nova Scotia.

In contrast to two-dimensional images, which are easy to obtain through single exposures of film or a digital sensor, three-dimensional images typically have to be constructed through multiple observations either by multiple sensors at the same time, or by a single sensor at multiple times. Instead of capturing exposure of pixels to light, typically the sensor measures some sort of wave (acoustic or electromagnetic), and an inverse problem is solved to reconstruct density or impedance in the interior of the object being measured.

Depending on the constraints imposed on the position of the sensors, resolution or accuracy of voxels in a three-dimensional image may not be uniform, and can cause artifacts in images. For instance, in the BRATS data set, one of the dimensions is not as well-aligned as the other two (see Figure 5.2), likely an artifact of the preference to view MRI images as two dimensional slices in a certain direction so reconstruction efforts are focused on creating high-quality visualizations for these slices. In the Penobscot data set, horizontal and vertical slices are qualitatively very different because of the vertical deposition of sedimentation (see Figure 5.3). Furthermore, the deeper slices are potentially of lower quality due to the position of the sensors near the surface.

This anisotropy in three-dimensional image data is fairly common, either as an artifact of image reconstruction as in the BRATS data set, or due to actual anisotropy in the object being imaged as in the Penobscot data set. Just as two-dimensional image patches sampled from the Van Hateren data set revealed certain structures at different densities, it will be important to understand that samples of three-dimensional patches will not be sampled uniformly from the space of all possible patches. The distribution of patches on model spaces from a given data set or image can be thought

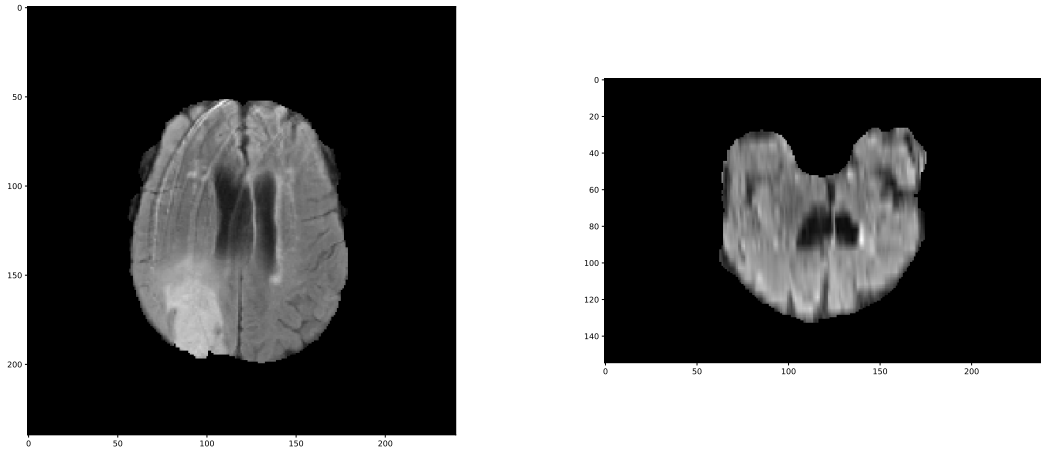


Figure 5.2: Two slices of an MRI in the BRATS data set. Left: a slice in the second two coordinates. Right: a slice in the first two coordinates.

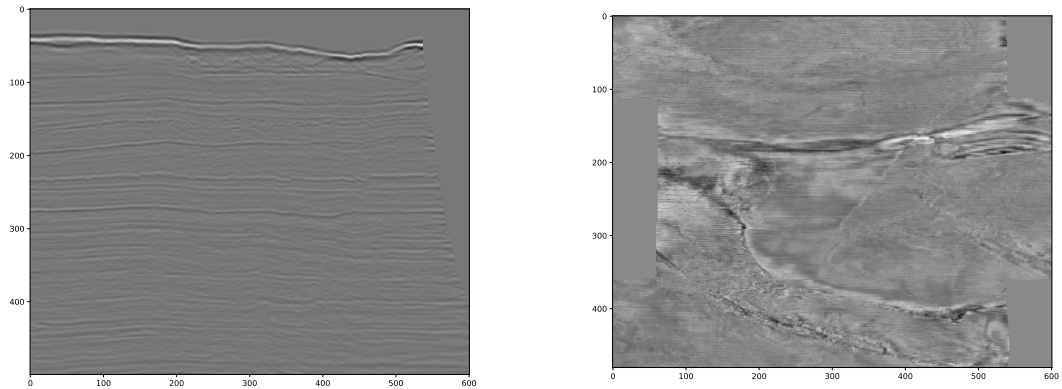


Figure 5.3: Two slices of the Penobscot data set. Left: the top third of an “inline” slice (the image continues vertically). Right: a horizontal slice..

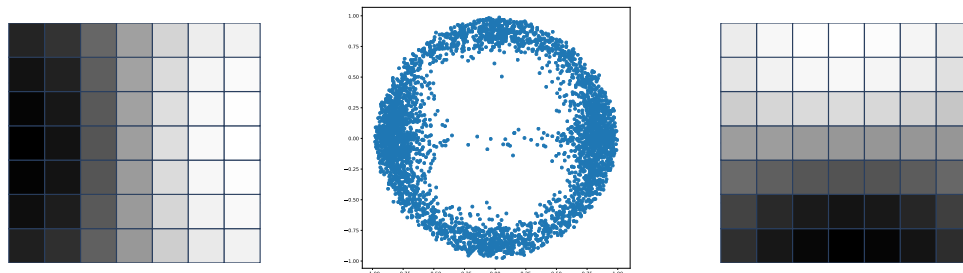


Figure 5.4: Primary circle in 7×7 patches, $k = 100$, $p = 0.2$, with a secondary circle beginning to fill in. Center: projection of data onto first two principal component directions. Left: first eigenpatch (patch maximally aligned with first principal component). Right: second eigenpatch.

of as a feature of that particular data set or image, and understanding such distributions is an important first step in beginning to leverage patch-based techniques for three dimensional data, as has been done for two-dimensional images in [16, 71, 86].

5.1.4 Primary Spheres and Secondary Circles

For certain cuts of patches for the Van Hateren data set, we see a “primary circle”. This appears in the data in Figure 5.4 – further exploration of different codensity cuts can be found in [37]. We’ll refer to the patch maximally aligned with a principal component as an *eigenpatch*.

As discussed in Section 5.1.3, three dimensional data often has directional artifacts. A codensity cut of the Penobscot data set can be seen in Figure 5.5, where the 2-dimensional PCA embedding shows a clear circle, but the eigenpatches (patches maximally aligned with the principal components) show that this circle corresponds to a “secondary” circle. Note that the first prin

The first eigenpatch for the Penobscot data corresponds to a higher-dimensional patch that we find on the primary circle for two-dimensional patches (i.e. a patch with a linear gradient). For higher-dimensional images, the generalization of the primary circle is the primary $(d - 1)$ -sphere, consisting of patches with all possible linear gradients.

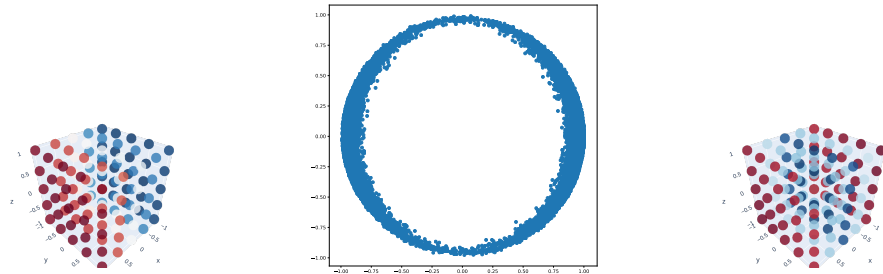


Figure 5.5: Secondary circle in $5 \times 5 \times 5$ patches from Penobscot data, $k = 100$, $p = 0.4$. Center: projection of data onto first two principal component directions. Left: first eigenpatch. Right: second eigenpatch.

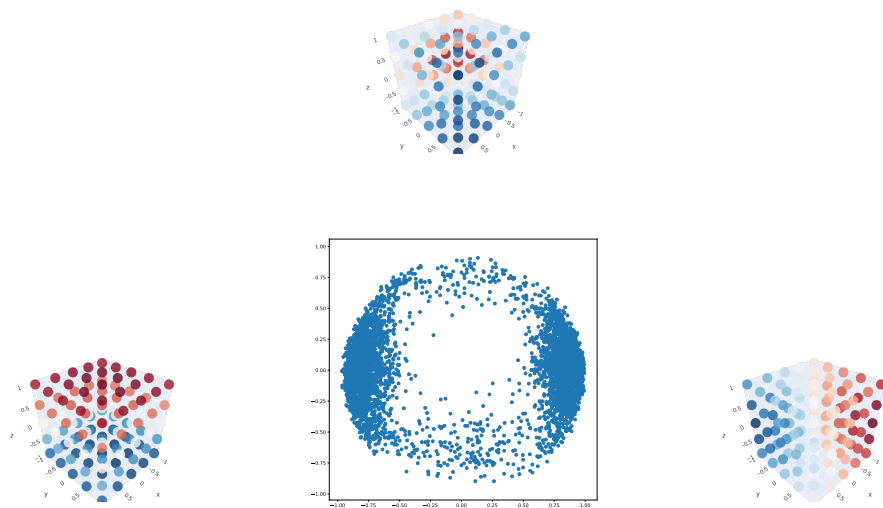


Figure 5.6: Primary great circle in $5 \times 5 \times 5$ patches from BRATS data, $k = 100$, $p = 0.2$. Projection of data onto first two principal component directions. Left: first eigenpatch. Right: second eigenpatch. Top: third eigenpatch.

In contrast to the Penobscot Data, the top two eigenpatches of the BRATS data lie on a primary *great circle*, corresponding to the equator of the primary 2-sphere, as seen in Figure 5.6. As seen, the top three eigenpatches all lie on the primary 2-sphere, although the sphere is not filled in.

As we see in Figures 5.4, 5.5 and 5.6, primary spheres and secondary circles appear in image patch data in a variety of contexts. We will now develop a model space that can be used to explain and understand these structures, and for which the behaviors seen above can be interpreted as sub-spaces contained in the model.

5.2 The Harris Fibration

Both Lee, Pedersen and Mumford [66] and Carlsson and de Silva [37] found that a version of the primary circle was an important structure in 3×3 image patch data. In the 2011 dissertation of Jose Perea [88] and 2014 paper of Perea and Carlsson [86], an explicit map based on the Harris edge detector [56] was used to find coordinate locations of patches on the Klein bottle as part of a pipeline to classify textures.

In this section, we'll define the Harris map on d -dimensional image patches (or voxel cubes)

$$h : \mathbf{X} \rightarrow \mathbb{R}P^{d-1} \tag{5.4}$$

and discuss how this map is a fibration on classes of models for images patches, including the Klein bottle.

5.2.1 The Harris Map

In 1988, Harris and Stephens [56] proposed a combined corner and edge detector for use in computer vision problems. The basic object of analysis was a matrix defined for each patch $x \in \mathbb{R}^{k_1 \times k_2}$

$$M(x) = \begin{bmatrix} I_{1,1}(x) & I_{1,2}(x) \\ I_{2,1}(x) & I_{2,2}(x) \end{bmatrix} \tag{5.5}$$

where

$$\begin{aligned} I_{1,1}(x) &= \frac{1}{k_1 k_2} \sum_{i,j} \Delta_1(x_{i,j})^2 \\ I_{2,2}(x) &= \frac{1}{k_1 k_2} \sum_{i,j} \Delta_2(x_{i,j})^2 \\ I_{1,2}(x) = I_{2,1}(x) &= \frac{1}{k_1 k_2} \sum_{i,j} \Delta_1(x_{i,j}) \Delta_2(x_{i,j}) \end{aligned}$$

and Δ_i is a finite-difference operator in the i -th coordinate direction. Alternatively, we can write

$$M(x) = \frac{1}{k_1 k_2} \sum_x \Delta(x) \Delta(x)^T \quad (5.6)$$

where Δ is a finite-difference gradient. Effectively, $M(x)$ is the matrix of covariances of finite differences over the patch. The eigenvalues and eigenvectors of $M(x)$ were used in [56] to decide if a patch contained an edge or corner – if one eigenvalue was large and the other small, the patch was labeled an edge; if both were large, the patch was labeled a corner; if both were small, the patch was neither an edge nor a corner. We define the map $h : x \rightarrow \mathbb{R}P^1$ as the map $x \rightarrow \text{MaxEigSpace } M(x)$ (the maximum eigenspace of $M(x)$) in a manner that parallels [86]. We can think of $h(x)$ as being represented by a unit eigenvector, which must be identified with $-h(x)$ because of sign and scaling ambiguity of eigenvectors. We will assume that the maximum eigenspace is 1-dimensional for every patch x (the set of matrices $M(x)$ for which this is not the case has measure zero, so we assume that this is valid with high probability). By the Courant-Fisher Minimax Theorem [53] (the variational characterization of eigenvectors and eigenvalues), this eigenvector is interpretable the direction of largest variation in the finite differences of x .

While the matrix $M(x)$ in Equation (5.5) was originally defined for two-dimensional images, there is no reason why its definition can not be extended to patches (cubes) in d -dimensional images, where we form a $d \times d$ covariance matrix of finite differences

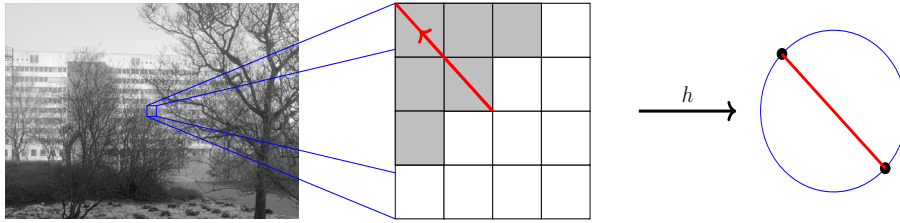


Figure 5.7: Harris map h from a patch to $\mathbb{R}P^1$

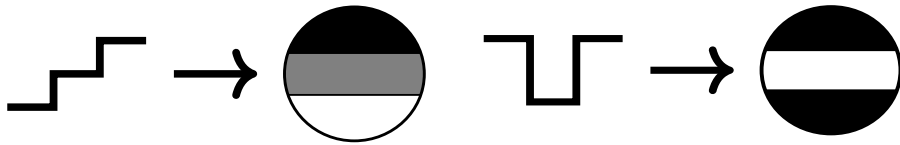


Figure 5.8: Left: patch with odd function $(k(\pi/2, \pi/2))$. Right: patch with even function $(k(\pi/2, 0))$.

in every dimension. In this case, we will obtain a map

$$h : \mathbf{X} \rightarrow \mathbb{R}P^{d-1} \tag{5.7}$$

following the same arguments.

5.2.2 A Fibration Structure

As seen in Section 5.1.4, primary spheres and secondary circles are important features of image patch data. We'll now describe a model for patches that captures these structures, and show that when restricted to patches of this form, the Harris map is a fibration. For simplicity, we'll consider image patches on a continuous domain, which will allow us to ignore issues of discretization when choosing values for pixels or voxels.

Let $D \subset \mathbb{R}^d$ denote either the unit d -dimensional disk or unit d -dimensional cube centered at the origin (generally, any d -dimensional symmetric bounded domain centered at the origin). Let $f_e, f_o : \mathbb{R} \rightarrow \mathbb{R}$ be functions, where $f_e(x) = f_e(-x)$ is an even function, and $f_o(-x) = -f_o(x)$ is an odd function. Let $v_\phi \in S^{d-1}$ denote a unit vector parameterized by angle ϕ . We define a family of functions $k(\phi, \theta; x) :$

$S^{d-1} \times S^1 \times D \rightarrow \mathbb{R}$ via

$$\mathcal{K}^d(f_e, f_o) = \{k(\phi, \theta; x) = \cos(\theta)f_e(v_\phi^T x) + \sin(\theta)f_o(v_\phi^T x) \mid v_\phi \in S^{d-1}, \theta \in [0, 2\pi)\} \quad (5.8)$$

See Figure 5.8 for an example of how even and odd functions define two-dimensional patches. If we fix ϕ , and vary θ , we obtain a secondary circle, as in Figure 1.4. If we fix $\theta = \pi/2$ or $3\pi/2$ (so only the odd component is present), we obtain the primary circle for two-dimensional patches (Figure 1.3), and a primary sphere S^{d-1} generally.

Fixing ϕ and θ , each $k(\phi, \theta; x)$ is a continuous function in x if f_e, f_o are both continuous. Let $\mathcal{L}_2(D)$ denote the space of square-integrable functions on D . If we let ϕ, θ vary, $k(\phi, \theta; x)$ is a continuous map from $S^{d-1} \times S^1$ to $\mathcal{L}_2(D)$ if f_e, f_o are both square-integrable. If $k(\phi, \theta; x)$ were one-to-one, this would imply that $\mathcal{K}^d(f_e, f_o)$ has the same topology as the parameter space, $S^{d-1} \times S^1$. However, the map is not one-to-one, because there is an identification of parameters

$$k(-\phi, -\theta; x) = \cos(-\theta)f_e(-v_\phi^T x) + \sin(-\theta)f_o(-v_\phi^T x) \quad (5.9)$$

$$= \cos(\theta)f_e(v_\phi^T x) + \sin(\theta)f_o(v_\phi^T x) \quad (5.10)$$

$$= k(\phi, \theta; x) \quad (5.11)$$

where we use $-\phi$ to denote the antipodal coordinate of S^{d-1} , so $v_{-\phi} = -v_\phi$ (for S^1 , $-\phi = \phi + \pi$), and $-\theta$ in the usual sense of reflection of S^1 across the horizontal axis.

We can use this structure to deduce that \mathcal{K}^d is a fiber bundle over $\mathbb{R}P^{d-1}$, by identifying $k(\phi, \theta; x)$ with a preferred representative of ϕ in $\mathbb{R}P^d$. The identification in Equation (5.9) defines a free \mathbb{Z}_2 action on the space $S^{d-1} \times S^1$, and the space \mathcal{K}^d is fiber bundle identified with the orbits of this action. Note that this fiber bundle is non-trivial, because the identification requires us to apply the \mathbb{Z}_2 -action $\theta \mapsto -\theta$ when crossing the boundary of the hemisphere on $\mathbb{R}P^d$.

Using this fiber bundle interpretation, we see that \mathcal{K}^2 is a Klein bottle, and \mathcal{K}^d is generally a d -dimensional manifold that generalizes the twisted structure of the Klein bottle using higher-dimensional base spaces.

If we take the continuous limit of the finite difference gradients Δ used to define the Harris matrix Equation (5.6), we obtain

$$M(k(\phi, \theta; x)) = \frac{1}{|D|} dx \int_D \nabla_x(k) \nabla_x^T(k) \quad (5.12)$$

If f_e, f_o are differentiable, we obtain the formula

$$\begin{aligned} \nabla_x k(\phi, \theta; x) &= \cos(\theta) f_e(v_\phi^T x) + \sin(\theta) f_o(v_\phi^T x) \\ &= (\cos(\theta) f_e'(v_\phi^T x) + \sin(\theta) f_o'(v_\phi^T x)) v_\phi \\ &= k'(\phi, \theta; x) v_\phi \end{aligned}$$

in which case, the matrix $M(x)$ is rank-1 with non-zero eigenvector v_ϕ . Thus, we see that the Harris map is a fibration for the model space \mathcal{K}^d , mapping each patch to its coordinate in $\mathbb{R}P^{d-1}$. Note that for discrete patches that there is some distortion in the map due to discretization.

5.2.3 Real Projective Spaces

Because the Harris map has the real projective space $\mathbb{R}P^{d-1}$ as its target, we will review some facts about these spaces which will be useful for us in calculations. Material in this section can be found in standard texts, such as [35, 57, 102].

First, we recall that $\mathbb{R}P^n$ is defined as a quotient space of $\mathbb{R}^{n+1} - \{0\} \simeq S^n$, defined by identifying points via the relation $x \sim \alpha x$ for all $x \in \mathbb{R}^{n+1} - \{0\}$, and all $\alpha \in \mathbb{R} - \{0\}$. We'll find it convenient to work with the antipodal identification on S^n , where we only need to use the relation $x \sim -x$, which makes it clear that S^n is a double cover of $\mathbb{R}P^n$. There is a relationship between an explicit cell structure on $\mathbb{R}P^n$ and a hemispherical cell structure on S^n , which for example can be seen in Figure 5.9, where $\mathbb{R}P^n$ is obtained by taking $e_i \sim Te_i$. This cell structure on $\mathbb{R}P^n$ has a cell in each dimension $\mathbb{R}P^n = e_0 \cup e_1 \cup \dots \cup e_n$, where ∂e_i wraps around e_{i-1}

twice. Applying the integral chain complex functor, we obtain

$$\partial e_i = e_{i-1} + (-1)^i e_{i-1} = \begin{cases} 0 & i \text{ odd} \\ 2e_{i-1} & i \text{ even} \end{cases} \quad (5.13)$$

This produces the chain complex

$$0 \longleftarrow \langle e_0 \rangle \xleftarrow{0} \langle e_1 \rangle \xleftarrow{\times 2} \langle e_2 \rangle \xleftarrow{0} \dots \quad (5.14)$$

Computing homology, we obtain

$$H_k(\mathbb{R}P^n; \mathbb{Z}) = \begin{cases} \mathbb{Z} & k = 0 \\ \mathbb{Z}_2 & k \text{ odd}, 0 < k < n \\ \mathbb{Z} & k \text{ odd}, k = n \\ 0 & \text{otherwise} \end{cases} \quad (5.15)$$

Finally, we need know the fundamental group of $\mathbb{R}P^n$, in order to understand how fibrations over $\mathbb{R}P^n$ can twist. Because $\mathbb{R}P^n, n \geq 1$ is path connected, we ignore the choice of base point. For $\mathbb{R}P^1 \simeq S^1$, $\pi_1(S^1) = \mathbb{Z}$. For $\mathbb{R}P^n, n \geq 2$, we have $\pi_1(\mathbb{R}P^n) \simeq \mathbb{Z}_2$ [57].

5.2.4 Local Coefficient Calculations

We'll now perform some calculations that will be necessary to compute homology of the fibration \mathcal{K}^d using the Lerray-Serre spectral sequence. Recall from Section 5.2.2, that ϕ can be thought of as an element of $\mathbb{R}P^{d-1}$ where antipodal points of S^{d-1} are identified. There is an antipodal action of $\pi_1(\mathbb{R}P^{d-1})$ on the secondary circles, which turns into an action on the homology of fibers, specifically $\rho : H_1(S^1) \rightarrow H_1(S^1)$, where $\rho : a \mapsto -a$. This “twisting” of the fiber has interesting effects on homology of the total space of the fibration, which we will capture using *local coefficients*. Homology with local coefficients was introduced by Steenrod [96], and we will follow [35] for exposition.

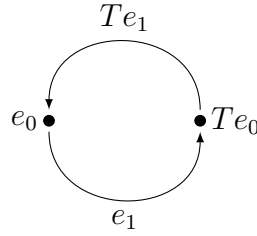


Figure 5.9: Local coefficient system on $\mathbb{R}P^1 = S^1$. T denotes the non-trivial \mathbb{Z} -automorphism $T : a \mapsto -a$.

A local coefficient system is thought of as a fiber bundle over a base space B . Let A be a discrete abelian group (we will consider $A = H_1(S^1) = \mathbb{Z}$), and let G be a subgroup of $\text{Aut}(A)$ (we will consider $G = \mathbb{Z}_2 = \text{Aut}(\mathbb{Z})$). Any fiber bundle E over B with fiber A and structure group G is called a *system of local coefficients on B* .

We'll now compute homology of $\mathbb{R}P^n$ with local coefficients given by the spherical double cover. In Figure 5.9 we have the local coefficient system for $\mathbb{R}P^1$, which lies on the double cover of the circle. For $\mathbb{R}P^n$, the double cover is the sphere S^n , and $\mathbb{R}P^1$ constitutes a special situation where $\mathbb{R}P^1 \simeq S^1$. We will first create a chain complex of S^n with coefficients in $\mathbb{Z}[\mathbb{Z}_2]$, considered as the module $\mathbb{Z}[T]/(T^2 - 1)$ using the hemispherical cell complex $\cup_{i=0}^n e_i \cup Te_i$. The cell complex is freely generated in each dimension by e_i , and the boundaries can be computed, similar to Equation (5.13) as

$$\partial e_i = e_{i-1} + (-1)^i T e_{i-1} = \begin{cases} (1 - T)e_{i-1} & i \text{ odd} \\ (1 + T)e_{i-1} & i \text{ even} \end{cases} \quad (5.16)$$

This produces the chain complex

$$0 \longleftarrow \langle e_0 \rangle \xleftarrow{1-T} \langle e_1 \rangle \xleftarrow{1+T} \langle e_2 \rangle \xleftarrow{1-T} \dots \quad (5.17)$$

We can then compute homology of $\mathbb{R}P^n$ with twisted \mathbb{Z} coefficients by replacing T with an element of $\text{Aut}(\mathbb{Z})$. In the case where we take the trivial automorphism $T = \text{id} = 1$ (that is, non-twisted coefficients), we obtain the chain complex Equation (5.14). In the case where we take the non-trivial automorphism $T = -\text{id} = -1$ (coefficients

with a twist), we obtain the chain complex

$$0 \longleftarrow \mathbb{Z} \xleftarrow{\times 2} \mathbb{Z} \xleftarrow{0} \mathbb{Z} \xleftarrow{\times 2} \dots \quad (5.18)$$

In this case, we can compute homology by induction to obtain

$$H_k(\mathbb{R}P^n; \mathbb{Z}_-) = \begin{cases} \mathbb{Z}_2 & k \text{ even, } k < n \\ \mathbb{Z} & k \text{ even, } k = n \\ 0 & \text{otherwise} \end{cases} \quad (5.19)$$

5.2.5 Homology of the Harris Fibration

We're now ready to compute homology of the total space of the Harris Fibration, \mathcal{K}^d , defined in Section 5.2.2 using the Leray-Serre spectral sequence. We already completed this computation for the case of 2-dimensional image patches in Section 1.2.3, in which case \mathcal{K}^2 was the Klein bottle. Recall that the Leray-Serre spectral sequence starts on the E^2 page, with

$$E_{p,q}^2 = H_p(B; \mathcal{H}_q(F))$$

In this case, $B = \mathbb{R}P^{d-1}$, and because $F = S^1$, there are non trivial entries for $q = 0, 1$. In the case of $q = 0$, we have $\mathcal{H}_0(F) = \mathbb{Z}$ (there is no action on $H_*(F)$ by $\pi_1(\mathbb{R}P^{d-1})$), and in the case of $q = 1$, we have the twisted coefficient system $\mathcal{H}_1(F) = \mathbb{Z}_-$ because $\pi_1(\mathbb{R}P^{d-1})$ acts on S^1 via the antipodal map. Thus, $E_{p,0}^2 = H_p(\mathbb{R}P^{d-1}; \mathbb{Z})$ (Equation (5.15)), and $E_{p,1}^2 = H_p(\mathbb{R}P^{d-1}; \mathbb{Z}_-)$ (Equation (5.19)). For \mathcal{K}^3 on 3-dimensional patches, we have a fibration over $\mathbb{R}P^2$, where the E^2 page of the

Leray-Serre Spectral sequence is

$$\begin{array}{ccccc}
 & 0 & & 0 & & 0 \\
 & & & & & \\
 & \mathbb{Z}_2 & & 0 & & \mathbb{Z} \\
 & & & & & \\
 & \mathbb{Z} & & \mathbb{Z}_2 & & 0
 \end{array} \tag{5.20}$$

And for the 4-dimensional patch space \mathcal{K}^4 , the E^2 page of the Leray-Serre spectral sequence is

$$\begin{array}{cccc}
 0 & 0 & 0 & 0 \\
 \mathbb{Z}_2 & 0 & \mathbb{Z}_2 & 0 \\
 \mathbb{Z} & \mathbb{Z}_2 & 0 & \mathbb{Z}
 \end{array} \tag{5.21}$$

We can already see some patterns that hold as the dimension d of the patch increases.

Proposition 5.2.1. *The Leray-Serre spectral sequence of the Harris fibration collapses at the E^2 page for any dimension d of patches.*

Proof. since the differential ∂^2 is of bi-degree $(-2, 1)$, the only possible non-trivial maps are $\partial_{p,0}^2 : E_{p,0}^2 \rightarrow E_{p-2,1}^2$, $p \geq 2$. If p is even, then $E_{p,0}^2 = H_p(\mathbb{R}P^{d-1}; \mathbb{Z}) = 0$, and if p is odd, then $E_{p-2,1}^2 = H_p(\mathbb{R}P^{d-1}, \mathbb{Z}_-) = 0$, so $\partial^2 = 0$. Thus $\partial^2 = 0$, since every map has either 0 as a source or target. For the E^r page, $r > 2$, this same condition is trivially verifiable from the $(-r, r - 1)$ bi-degree of the differential, and the observation that non-zeros only occur on the first two rows. \square

Proposition 5.2.2. *Let \mathcal{K}^d denote the total space of the Harris fibration on d -dimensional patches. Then $H_k(\mathcal{K}^d) = 0$ for even k , $k \geq 2$, and also for $k > d$.*

Proof. Suppose $k \geq 2$ is even or $k > d$. Because the spectral sequence collapses on the E^2 page, we have $E_{0,k}^\infty = H_k(\mathbb{R}P^{d-1}; \mathbb{Z}) = 0$, since $k \geq 2$ is even, or $k > d$, by

Equation (5.15), and $E_{1,k-1}^\infty = H_{k-1}(\mathbb{R}P^{d-1}; \mathbb{Z}_-) = 0$, since $k-1 \geq 1$ is odd, or $k > d$, by Equation (5.19). Thus, the filtration on $H_k(\mathcal{K}^d)$ given by the spectral sequence is trivial, and so $H_k(\mathcal{K}^d) = 0$. \square

Proposition 5.2.3. $H_d(\mathcal{K}^d) = \mathbb{Z}$ if d is odd.

Proof. See Equation (5.20) for an example. Because the spectral sequence collapses on the E^2 page, we have $E_{0,d}^\infty = H_d(\mathbb{R}P^{d-1}; \mathbb{Z}) = 0$ since $d-1$ is even by Equation (5.15), and $E_{1,d-2}^\infty = H_{d-1}(\mathbb{R}P^{d-1}; \mathbb{Z}_-) = \mathbb{Z}$ since $d-1$ is even by Equation (5.19). This gives a filtration on $H_d(\mathcal{K}^d)$ for which the only possible solution is $H_d(\mathcal{K}^d) = \mathbb{Z}$. \square

Proposition 5.2.4. $H_{d-1}(\mathcal{K}^d) = \mathbb{Z} \oplus \mathbb{Z}_2$ if d is even.

Proof. See Equation (5.21) for an example. Because the spectral sequence collapses on the E^2 page, we have $E_{0,d-1}^\infty = \mathbb{Z}$, and $E_{1,d-2}^\infty = \mathbb{Z}_2$, which gives a filtration on $H_{d-1}(\mathcal{K}^d)$. The only possible \mathbb{Z} module with this filtration problem is $H_{d-1}d(K) = \mathbb{Z} \oplus \mathbb{Z}_2$. \square

The most challenging statement involves $H_k(\mathcal{K}^d)$ for $k < d-1$ odd, since the extension problem on the filtration is non-trivial.

Proposition 5.2.5. $H_k(\mathcal{K}^d) = \mathbb{Z}_2 \oplus \mathbb{Z}_2$ if $0 < k < d-1$ is odd.

Proof. For examples, see the first super-diagonal in Equations (5.20) and (5.21). Because the spectral sequence collapses on the E^2 page, we have $E_{0,k}^\infty = H_k(\mathbb{R}P^{d-1}; \mathbb{Z}) = \mathbb{Z}_2$, since $k < d-1$ is odd by Equation (5.15), and $E_{1,k-1}^\infty = H_{k-1}(\mathbb{R}P^{d-1}; \mathbb{Z}_-) = \mathbb{Z}_2$, since $k-1$ is even by Equation (5.19).

This gives a filtration on $H_k(\mathcal{K}^d)$ as $F_1 H_k(\mathcal{K}^d) = \mathbb{Z}_2$, and $F_2 H_k(\mathcal{K}^d)/F_1 H_k(\mathcal{K}^d) = \mathbb{Z}_2$. There are two possible solutions to this extension problem - either $H_k(\mathcal{K}^d) = \mathbb{Z}_2 \oplus \mathbb{Z}_2$, or $H_k(\mathcal{K}^d) = \mathbb{Z}_4$. In order to determine which case actually occurs, we need to go beyond the statement of the Leray-Serre spectral sequence, and look under the hood of how the spectral sequence is constructed.

One way of constructing the Leray-Serre spectral sequence uses a filtration on a cell structure for \mathcal{K}^d (see [75] Chapter 6), where the i -skeleta are defined via $\mathcal{K}_i^d = \bigcup_{j=0}^i F_j \times B_{i-j}$, where attaching maps are determined via a combination of the

attaching maps on F and B and the action of $\pi_1(B)$. We will use the cell structure on $B = \mathbb{R}P^{d-1}$ given in Section 5.2.3, and the (same) cell structure on $F = S^1$ with a single cell in dimensions 0 and 1. Passing through the chain functor, we see that there is a sub-chain complex generated by $\mathbb{R}P^{d-1} \times e_0 \simeq \mathbb{R}P^{d-1}$. In particular, because $\partial_*^F = 0$, the kernel of this sub-chain complex is not in the image of any chains generated by $\mathbb{R}P^{d-1} \times e_1$, so $[v] \in H_k(\mathbb{R}P^{d-1})$ will generate homology via $[v \times e_0] \in H_k(K)$ with the same torsion. Specifically, this implies that the \mathbb{Z}_2 homology class in $H_k(\mathbb{R}P^{d-1})$ becomes a \mathbb{Z}_2 homology class in $H_k(\mathcal{K}^d)$. In conjunction with information from the Leray-Serre spectral sequence, we must have $H_k(\mathcal{K}^d) = \mathbb{Z}_2 \oplus \mathbb{Z}_2$. \square

All the preceding propositions can be combined into a single statement

Theorem 5.2.6. *Let \mathcal{K}^d denote the total space of the Harris fibration on d -dimensional patches. Then*

$$H_k(\mathcal{K}^d) = \begin{cases} \mathbb{Z} & k = 0 \\ \mathbb{Z}_2 \oplus \mathbb{Z}_2 & 0 < k < d - 1, k \text{ odd} \\ \mathbb{Z} & k = d, d \text{ odd} \\ \mathbb{Z} \oplus \mathbb{Z}_2 & k = d - 1, d \text{ even} \\ 0 & \text{otherwise} \end{cases} \quad (5.22)$$

For example, for 2-dimensional patches, we have a Klein bottle

$$H_*(\mathcal{K}^2; \mathbb{Z}) = \mathbb{Z}, \mathbb{Z} \oplus \mathbb{Z}_2, 0 \quad (5.23)$$

and for 3-dimensional patches, we have

$$H_*(\mathcal{K}^3; \mathbb{Z}) = \mathbb{Z}, \mathbb{Z}_2 \oplus \mathbb{Z}_2, 0, \mathbb{Z} \quad (5.24)$$

and for 4-dimensional patches, we have

$$H_*(\mathcal{K}^4; \mathbb{Z}) = \mathbb{Z}, \mathbb{Z}_2 \oplus \mathbb{Z}_2, 0, \mathbb{Z} \oplus \mathbb{Z}_2 \quad (5.25)$$

5.2.6 Field Coefficient Calculations

Since we will work with field coefficients when computing persistent homology, we will now apply the Universal Coefficient Theorem (Theorem 1.3.1) to state the homology of the Harris fibration with different field coefficients.

First, we would like to have results for the real projective plane.

$$H_k(\mathbb{R}P^n; \mathbb{F}_2) = \begin{cases} \mathbb{F}_2 & k \leq n \\ 0 & k > n \end{cases} \quad (5.26)$$

and for $\mathbb{F} = \mathbb{F}_p$, $p > 2$, or $\mathbb{F} = \mathbb{Q}$, we have

$$H_k(\mathbb{R}P^n; \mathbb{F}) = \begin{cases} \mathbb{F} & k = 0, n \text{ odd} \\ 0 & \text{otherwise} \end{cases} \quad (5.27)$$

Results for the total space \mathcal{K}^d of the Harris fibration for d -dimensional patches can be obtained either through the universal coefficient theorem (Theorem 1.3.1), or by computing the Leray-Serre spectral sequence with field coefficients.

$$H_k(\mathcal{K}^d; \mathbb{F}_2) = \begin{cases} \mathbb{F}_2 & k = 0, d \text{ odd} \\ \mathbb{F}_2 \oplus \mathbb{F}_2 & 0 < k < d \\ \mathbb{F}_2 & k = d \\ 0 & \text{otherwise} \end{cases} \quad (5.28)$$

and for $\mathbb{F} = \mathbb{F}_p$, $p > 2$, or $\mathbb{F} = \mathbb{Q}$, we have

$$H_k(\mathcal{K}^d; \mathbb{F}) = \begin{cases} \mathbb{F} & k = 0 \\ \mathbb{F} & k = d - 1, d \text{ even} \\ \mathbb{F} & k = d, d \text{ odd} \\ 0 & \text{otherwise} \end{cases} \quad (5.29)$$

For example, on 3-dimensional patches we have

$$H_*(\mathcal{K}^3; \mathbb{F}_2) = \mathbb{F}_2, \mathbb{F}_2 \oplus \mathbb{F}_2, \mathbb{F}_2 \oplus \mathbb{F}_2, \mathbb{F}_2 \quad (5.30)$$

$$H_*(\mathcal{K}^3; \mathbb{F}) = \mathbb{F}, 0, 0, \mathbb{F} \quad (5.31)$$

and for 4-dimensional patches we have

$$H_*(\mathcal{K}^4; \mathbb{F}_2) = \mathbb{F}_2, \mathbb{F}_2 \oplus \mathbb{F}_2, \mathbb{F}_2 \oplus \mathbb{F}_2, \mathbb{F}_2 \oplus \mathbb{F}_2, \mathbb{F}_2 \quad (5.32)$$

$$H_*(\mathcal{K}^4; \mathbb{F}) = \mathbb{F}, 0, 0, \mathbb{F} \quad (5.33)$$

5.3 Density in the Image and Fibers of the Harris Map

As discussed in Sections 1.2.1 and 5.1.3, the distribution of patches is not uniform in either two- or three-dimensional data. In this section, we investigate the distribution of patches in $\mathbb{R}P^{d-1}$, as well as the distribution of patches in pullbacks of open sets in the image of the Harris Map.

5.3.1 Two-dimensional Images

In Figure 5.10, we visualize the density of image patches sampled from the Van Hateren data set both in the image of the Harris map and in terms of the nearest neighbor on a model Klein bottle described in Section 5.2.2, using $f_e = x^2$ and $f_o = x$, sampled on a uniform grid. Note that no codensity filter was applied, so some patches may be removed in subsequent analyses. We see that patches do not occur with equal frequency in terms of orientation, and see that concentration occurs with higher frequency near $\phi \approx \pi \sim 0$ (variation is in horizontal direction), with a smaller bump near $\phi \approx \pi/2$ (variation in a vertical direction). In the Klein bottle densities, the vertical lines are “secondary circles”, and the two horizontal bands, especially visible for 3×3 patches, are the “primary circle” appearing as the double cover of

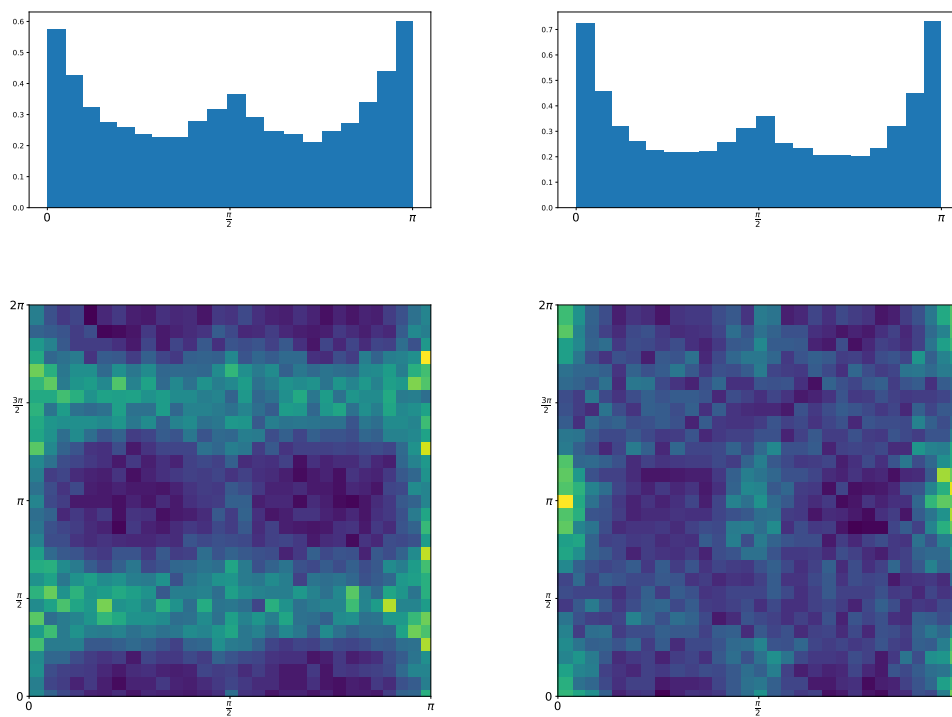


Figure 5.10: Left: 3×3 patches. Right: 7×7 patches. Top: histograms of the image of the Harris map applied to patches obtained from the Van Hateren data set. Bottom: histograms of points assigned to nearest neighbors in a model Klein bottle (yellow bins contain more patches, and blue contain fewer).

$\mathbb{R}P^1$ (note the Klein bottle is obtained by an antipodal identification on the left and right sides of the histogram).

In Figure 5.11, we inverse images of open sets of the Harris map applied to image patches sampled from the Van Hateren data set. We see a concentration of patches in two clusters along the first eigenpatch direction, corresponding to the primary circle as a double cover of $\mathbb{R}P^1$. As the codensity of patches decreases, we see secondary circles appear, with the second eigenpatch direction appearing as a line. Note that in Figure 5.11 that if no codensity threshold is applied to the PCA plots, the circles would fill in.

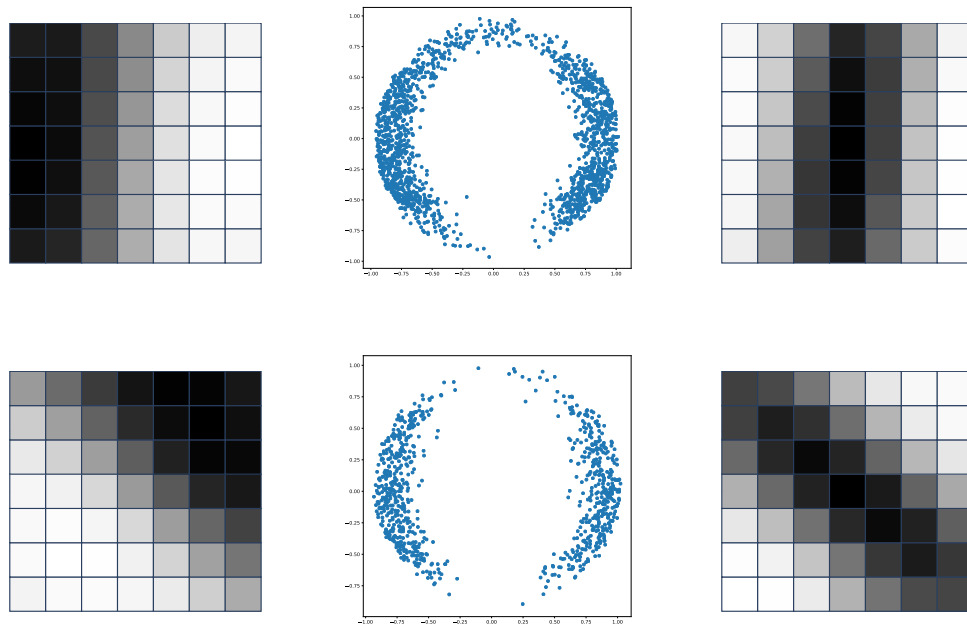


Figure 5.11: PCA embeddings of inverse images of open sets in $\mathbb{R}P^1$ covering the image of the Harris map applied to 7×7 patches, with 1st and 2nd eigenpatches (left and right). Only patches in the top 35% highest codensity ($k = 50$) are displayed as points. Top: open set with the largest number of patches. Bottom: open set with smallest number of patches.

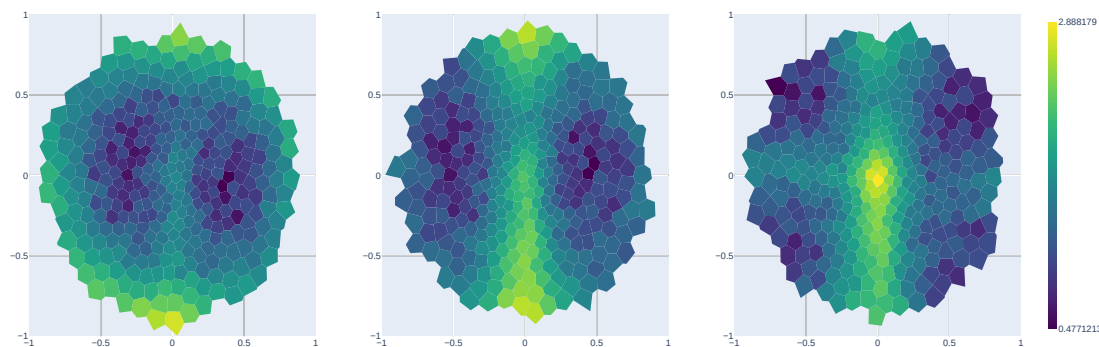


Figure 5.12: Image of the Harris map applied to patches sampled from the BRATS data set. Affine spaces of $\mathbb{R}P^2$ colored by the \log_{10} number of patches in landmark neighborhoods. From left to right: stereographic projections from e_1 , e_2 , and e_3 .

5.3.2 Three-dimensional Images

We can apply similar techniques to visualize the distribution of three-dimensional patches in both the BRATS and Penobscot data sets. In Figures 5.12 and 5.14, we visualize the density of the image of the Harris map by landmarking $\mathbb{R}P^2$ and then assigning patches to the closest landmark in the image of the Harris map. Cells in the images are the Voronoi cells of the landmarks in several stereographic projections, and are colored by the number of points assigned. In the BRATS data set, we see concentrations of patches along the three great circles spanned by two of the three canonical coordinates. In the Penobscot data set, we see a large concentration of patches that primarily vary in the second coordinate.

In Figures 5.13 and 5.15, we project patches onto a model \mathcal{K}^3 , and visualize the distribution using a 3D heatmap. Again, we employ a stereographic projection for two of the axes, and the third axis is given by the coordinate θ , cut at $0 \sim 2\pi$. In the BRATS data, we see two sections corresponding to the primary 2-sphere, although some fill-in can be seen in the secondary circles aligned with the third coordinate. In the Penobscot data, we see a strong secondary circle over the second coordinate.

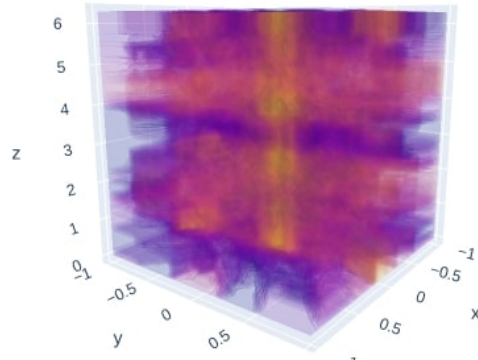


Figure 5.13: Histogram of the number of nearest neighbors in the BRATS data to points in \mathcal{K}^3 . The xy plane is a stereographic projection of $\mathbb{R}P^2$ from the third coordinate, and the z axis is the fiber coordinate θ .

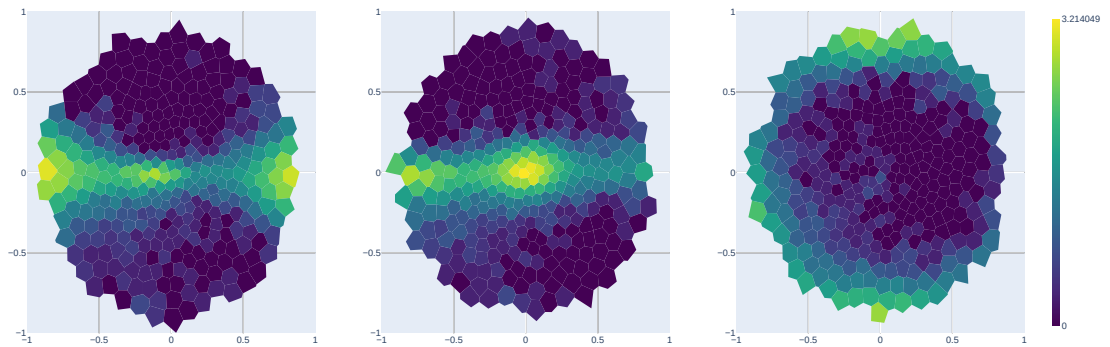


Figure 5.14: Image of the Harris map applied to patches sampled from the Penobscot data set. Affine spaces of $\mathbb{R}P^2$ colored by the \log_{10} number of patches in landmark neighborhoods. From left to right: stereographic projections from e_1 , e_2 , and e_3 .

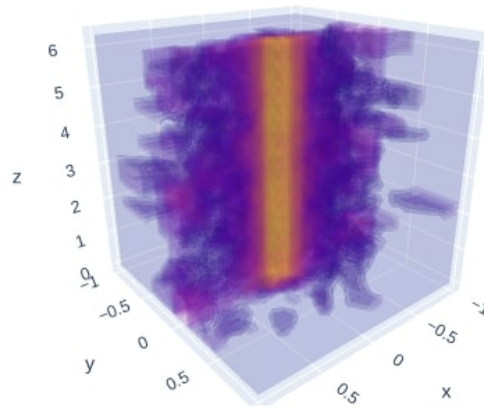


Figure 5.15: Histogram of the number of nearest neighbors in the Penobscot data to points in \mathcal{K}^3 . The xy plane is a stereographic projection of $\mathbb{R}P^2$ from the second coordinate, and the z axis is the fiber coordinate θ .

In contrast to the two-dimensional Van Hateren data, both the BRATS and Penobscot data display a large degree of anisotropy. In the BRATS data, the third coordinate has a larger step size than the first and second coordinate, explaining the larger variation seen in patches that vary primarily in the third coordinate. In the Penobscot data, patches primarily vary in the second coordinate, which is the vertical direction (physically) in the data. This can be attributed to the fact that sediment layers primarily vary in this direction. We also see increased variation along the third coordinate, compared with the first coordinate. This is an artifact of how the three dimensional volume is constructed as a stack of two-dimensional slices which are not always perfectly aligned.

5.4 Topological Computations

In Section 5.3, we visualized the distribution of patches in the BRATS and Penobscot data sets. Note that these visualizations effectively reduced patches from a 125-dimensional space to the three-dimensional space \mathcal{K}^3 . We will now compute persistent homology of sub-spaces of the data, filtered by codensity, and relate what we see to the visualizations.

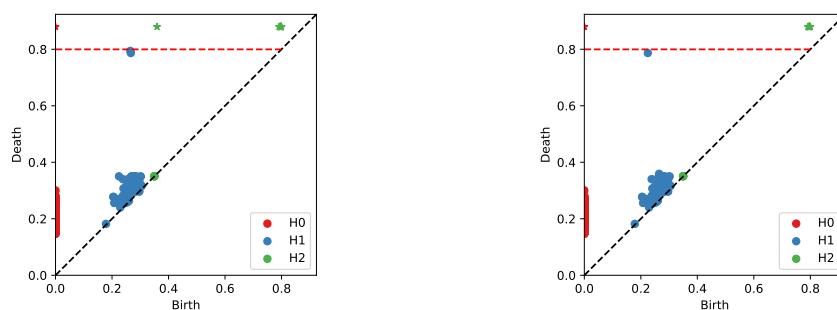


Figure 5.16: Dowker persistence diagrams with 200 landmarks computed on a dataset sampled from \mathcal{K}^2 . Left: computed with \mathbb{F}_2 coefficients. Right: computed with rational coefficients.

5.4.1 2-dimensional Image Patches

In Figure 5.16, we compute persistence diagrams for Dowker filtrations on patches sampled from a model Klein bottle, \mathcal{K}^2 . As we expect from our computations in Section 5.2.6, we see a prominent H_2 class when computing with \mathbb{F}_2 coefficients, which disappears along with one of the two prominent H_1 classes when we compute with rational coefficients instead.

In Figure 5.17, we see a space obtained by applying the $k = 100$, $p = 0.2$ co-density filter to 7×7 patches in the Van Hateren data. We see three prominent H_1 classes, which can be attributed to a model consisting of the primary circle with two chords at $\phi = 0$ and $\phi = \pi/2$, passing through $\theta = \pi$, seen in the histogram on the left side of the figure.

5.4.2 3-dimensional Image Patches

In Figure 5.18, we compute persistence diagrams for Dowker filtrations on patches sampled from a model space \mathcal{K}^3 . Again, our computations agree with our computations in Section 5.2.6, we see two prominent H_2 classes and two prominent H_1 classes when computing with \mathbb{F}_2 coefficients, all of which disappear when computing with rational coefficients.

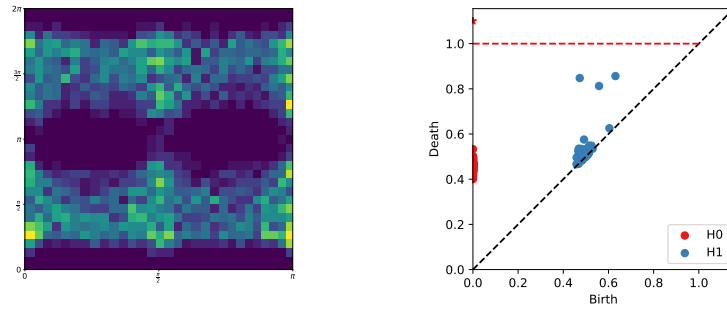


Figure 5.17: 7×7 patches from the Van Hateren data set, filtered with $k = 30$, $p = 0.2$. Left: histogram of density of projection onto a model \mathcal{K}^2 . Right: Persistence diagram for Dowker complex, 200 landmarks, computed with \mathbb{F}_2 coefficients.

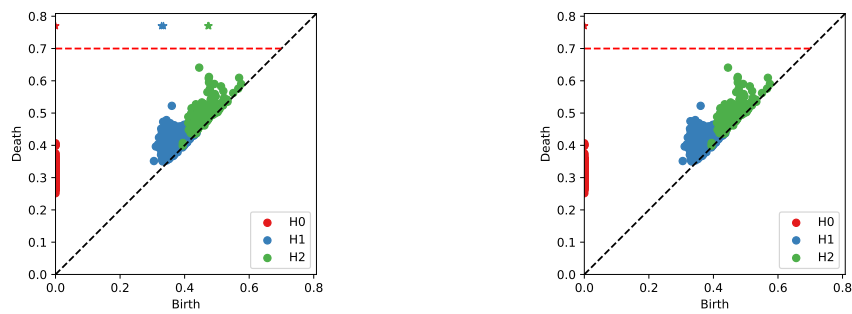


Figure 5.18: Dowker persistence diagrams with 300 landmarks computed on a dataset sampled from \mathcal{K}^3 . Left: computed with \mathbb{F}_2 coefficients. Right: computed with rational coefficients.

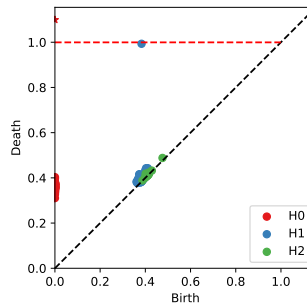


Figure 5.19: Dowker persistence diagram with 100 landmarks computed with \mathbb{F}_2 coefficients on $5 \times 5 \times 5$ patches sampled from the Penobscot data. $k = 40$, $p = 0.4$.

In Figure 5.19, we compute a Dowker persistence diagram for patches sampled from the Penobscot data. We see what we expect from Figure 5.15, which is a strong secondary circle producing a prominent H_1 class.

Finally, we filter patches in the BRATS data set to a high-density region. In Figure 5.20, we see a prominent H_2 class, corresponding to the primary two-sphere appearing as the double cover of the horizontal plane in the histogram on the left. The prominent H_1 class comes from the high-density fiber aligned with the third coordinate in the middle of the histogram. Thus, the histogram and persistence diagram can be explained by a primary 2-sphere with the north and south poles connected by a secondary circle.

5.4.3 Commentary

One of the striking results of this section is the utility of the the visualizations in Section 5.3 to explain what we see in persistence diagrams in high-density regions of the patch data. The fact that we can see the same structure in persistence diagrams as we visualize in the model implies that the projection onto the model space does not produce overly large and unnatural topological distortions on these high-density sub-spaces. Furthermore, the structure seen in the model spaces is capable of explaining the prominent features of the persistence diagrams, meaning we aren't missing anything topologically important in the model.

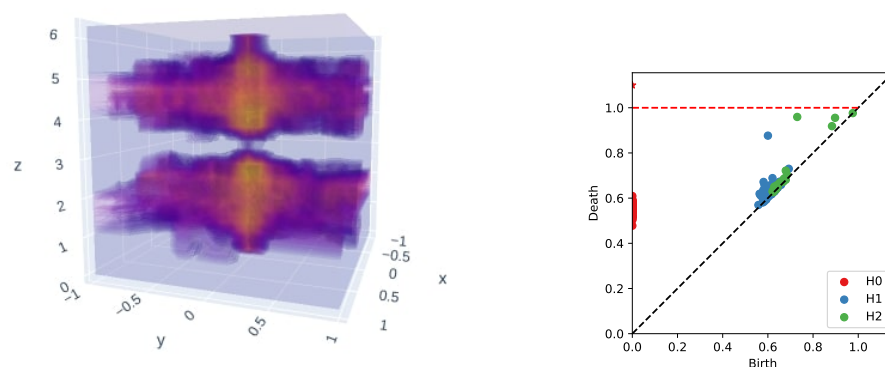


Figure 5.20: $5 \times 5 \times 5$ patches from the BRATS dataset. Codensity filter with $k = 100$, $p = 0.2$. Left: histogram of density of projection onto model \mathcal{K}^3 , with xy plane stereographic projection from the third coordinate. Right: Persistent homology of Dowker filtration with 100 landmarks, \mathbb{F}_2 coefficients.

All together, the projection of 125-dimensional patches onto the model subspace \mathcal{K}^3 appears to capture much of the interesting topology in high-density regions of the data. Note that there still might be important structure that *contracts* onto this space which is not visible in the persistence diagrams.

5.5 Conclusion & Future Directions

In this chapter, we investigated the use of parameterized models to help in the investigation of complex, yet structured, spaces of data. We used a model space for which the Harris map is a fibration in order to investigate the distribution of image patches in 2-dimensional and 3-dimensional images. We also used Dowker complexes, which we analyzed in Chapter 4 to perform computations to investigate topology of the data.

The structure that we discovered in these data sets has a variety of potential applications. For instance, the model space introduced in Section 5.2 could be used to extend applications of the Klein bottle for 2-dimensional image patches, such as in designing compression schemes for higher-dimensional images [71], or in augmenting and understanding convolutional neural networks in higher-dimensional images [16].

The density analysis we conducted on higher-dimensional images could also be useful in 3-dimensional texture analysis, extending the work of [86, 88]. Additionally, our investigation of anisotropy in the image of the Harris map may help guide denoising and reconstruction efforts for 3-dimensional images, particularly when we expect an isotropic ground truth.

Finally, the parameterized methods that we used to investigate spaces of image patches could be used in entirely different contexts. Natural candidates include data that is sampled from other fibrations, or spaces that come equipped with maps to a base space. One topic to investigate would be dynamic point clouds, where there is a natural map to the time parameter. More generally, topology that is governed by a control system (perhaps with a complicated state space) would also be a candidate.

Chapter 6

Conclusion

In this dissertation, we first developed new matrix-factorization based algorithms for computing persistent and zigzag homology, then developed the use of filtered carriers to investigate the relationship between cover complexes and other geometric complexes, and finally proposed a fibration model for image patches and used this to investigate data sets of patches sampled from 3-dimensional images.

Overall, we believe that the incorporation of problem structure into topological data analysis will present many opportunities to aid computation and data analysis. The conclusions of Chapters 3, 4 and 5 outline various directions in which future work may build on the ideas presented. The use of maps to a base space has been a fruitful point of view in pure topology, for instance in the study of fiber bundles, and has the potential to reveal richer information about a space than can be captured using ordinary (persistent) homology. Development of parameterized methods for topological data analysis has the potential to enrich the field as well. This may take many forms, including some found in this work.

Bibliography

- [1] H. Adams, J. Bush, B. Carr, L. Kassab, and J. Mirth. A torus model for optical flow. *Pattern Recognition Letters*, 129:304–310, 2020.
- [2] H. Adams and G. Carlsson. On the nonlinear statistics of range image patches. *SIAM J. Imaging Sciences*, 2(1):110–117, 2009.
- [3] N. Ahmed, T. Natarajan, and K. R. Rao. Discrete cosine transform. *IEEE Transactions on Computers*, C-23(1):90–93, 1974.
- [4] J. W. Alexander. Combinatorial analysis situs. *Transactions of the American Mathematical Society*, 28(2):301–329, 1926.
- [5] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J. Kirby, J. Freymann, K. Farahani, and C. Davatzikos. Segmentation labels for the pre-operative scans of the tcga-gbm collection. The Cancer Imaging Archive, 2017.
- [6] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J. S. Kirby, J. B. Freymann, K. Farahani, and C. Davatzikos. Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features. *Scientific Data*, 4:170–117, 2017.
- [7] S. Bakas et al. Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the BRATS challenge. Preprint <https://arxiv.org/abs/1811.02629>, 2019.

- [8] L. Baroni, R. M. Silva, R. S. Ferreira, D. Chevitarese, D. Szwarcman, and E. Vital Brazil. Penobscot interpretation dataset. Available at <https://doi.org/10.5281/zenodo.1341774>, July 2018.
- [9] U. Bauer. Ripser: efficient computation of vietoris-rips persistence barcodes. Preprint: <https://arxiv.org/abs/1908.02518>, Aug. 2019.
- [10] U. Bauer, M. Kerber, and J. Reininghaus. Clear and compress: Computing persistent homology in chunks. In P.-T. Bremer, I. Hotz, V. Pascucci, and R. Peikert, editors, *Topological Methods in Data Analysis and Visualization III*, pages 103–117. Springer International Publishing, 2014.
- [11] U. Bauer, M. Kerber, J. Reininghaus, and H. Wagner. Phat – persistent homology algorithms toolbox. *Journal of Symbolic Computation*, 78:76–90, 2017.
- [12] A. J. Blumberg and M. Lesnick. Universality of the homotopy interleaving distance. Preprint: <http://arxiv.org/abs/1705.01690>, 2017.
- [13] J.-D. Boissonnat, S. Pritam, and D. Pareek. Strong collapse for persistence. In *26th Annual European Symposium on Algorithms, ESA 2018*, pages 67:1–67:13, 2018.
- [14] K. Borsuk. On the imbedding of systems of compacta in simplicial complexes. *Fundamenta Mathematicae*, 35:217–234, 1948.
- [15] M. Brun and N. Blaser. Sparse dowker nerves. *Journal of Applied and Computational Topology*, 3:1–28, 2019.
- [16] R. Brüel Gabrielsson and G. Carlsson. Exposition and interpretation of the topology of neural networks. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pages 1069–1076, 2019.
- [17] R. Brüel-Gabrielsson, B. J. Nelson, A. Dwaraknath, P. Skraba, L. J. Guibas, and G. Carlsson. A topology layer for machine learning. In *The 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020. To appear.

- [18] P. Bubenkik, V. de Silva, and J. Scott. Metrics for generalized persistence modules. *Foundations of Computational Mathematics*, 15:1501–1531, 2015.
- [19] D. Burghelea. *New topological invariants for real- and angle-valued maps: an alternative to Morse-Novikov theory*. World Scientific, 2018.
- [20] D. Burghelea and T. K. Dey. Topological persistence for circle-valued maps. *Discrete & Computational Geometry*, 50(1):69–98, 2013.
- [21] Z. Cang and G.-W. Wei. TopologyNet: Topology based deep convolutional and multi-task neural networks for biomolecular property predictions. *PLOS Computational Biology*, 13(7):e1005690, 2017.
- [22] G. Carlsson and V. de Silva. Zigzag persistence. *Foundations of Computational Mathematics*, 10(4):367–405, 2010.
- [23] G. Carlsson, V. de Silva, and D. Morozov. Zigzag persistent homology and real-valued functions. In *SCG '09: Proceedings of the twenty-fifth annual symposium on Computational geometry*, page 247. ACM Press, 2009.
- [24] G. Carlsson, A. Dwaraknath, and B. J. Nelson. Persistent and Zigzag Homology: A Matrix Factorization Viewpoint. Preprint: <https://arxiv.org/abs/1911.10693>, 2019.
- [25] G. Carlsson, T. Ishkhanov, V. de Silva, and A. Zomorodian. On the local behavior of spaces of natural images. *International Journal of Computer Vision*, 76(1):1–12, 2008.
- [26] G. Carlsson and A. Zomorodian. The theory of multidimensional persistence. *Discrete & Computational Geometry*, 42(1):71–93, 2009.
- [27] N. J. Cavanna and D. R. Sheehy. The generalized persistent nerve theorem. Preprint: <http://arxiv.org/abs/1807.07920>, 2018.
- [28] F. Chazal, D. Cohen-Steiner, M. Glisse, L. J. Guibas, and S. Y. Oudot. Proximity of persistence modules and their diagrams. In *Proceedings of the 25th annual symposium on Computational geometry - SCG '09*, page 237. ACM Press, 2009.

- [29] F. Chazal, D. Cohen-Steiner, L. J. Guibas, F. Mémoli, and S. Y. Oudot. Gromov-hausdorff stable signatures for shapes using persistence. *Computer Graphics Forum*, 28(5):1393–1403, 2009.
- [30] F. Chazal, V. de Silva, M. Glisse, and S. Oudot. *The Structure and Stability of Persistence Modules*. SpringerBriefs in Mathematics. Springer International Publishing, 2016.
- [31] F. Chazal, V. de Silva, and S. Oudot. Persistence stability for geometric complexes. *Geometriae Dedicata*, 173:193–214, 2014.
- [32] F. Chazal and S. Y. Oudot. Towards persistence-based reconstruction in euclidean spaces. In *Proceedings of the twenty-fourth annual symposium on Computational geometry - SCG '08*, page 232. ACM Press, 2008.
- [33] C. Chen and M. Kerber. Persistent homology computation with a twist. In *27th European Workshop on Computational Geometry*, pages 197–200, 2011.
- [34] F. Chung. *Spectral Graph Theory*, volume 92 of *CBMS Regional Conference Series in Mathematics*. American Mathematical Society, 1997. ISSN: 0160-7642, 2380-5668.
- [35] J. F. Davis and P. Kirk. *Lecture Notes in Algebraic Topology*. Graduate Studies in Mathematics 35. AMS, 2001.
- [36] V. de Silva. A weak characterisation of the delaunay triangulation. *Geometriae Dedicata*, 135(1):39–64, 2008.
- [37] V. de Silva and G. Carlsson. Topological estimation using witness complexes. In *Proc. Sympos. Point-Based Graphics*, pages 538–552, 2004.
- [38] V. de Silva, D. Morozov, and M. Vejdemo-Johansson. Dualities in persistent (co)homology. *Inverse Problems*, 27(12):124003, 2011.
- [39] H. Derksen and J. Weyman. Quiver representations. *Notices of the American Mathematical Society*, 52(2), 2005.

- [40] T. K. Dey, F. Fan, and Y. Wang. Computing topological persistence for simplicial maps. In *Proceedings of the Thirtieth Annual Symposium on Computational Geometry*, SOCG'14, pages 345:345–345:354. ACM, 2014.
- [41] T. K. Dey, F. Mémoli, and Y. Wang. Multiscale mapper: Topological summarization via codomain covers. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on discrete algorithms*, pages 997–1013. SIAM, 2016.
- [42] C. H. Dowker. Homology groups of relations. *Annals of Mathematics*, 56(1):84–95, 1952.
- [43] P. Dłotko and H. Wagner. Simplification of complexes of persistent homology computations. *Homology, Homotopy and Applications*, 16(1):49–63, 2014.
- [44] H. Edelsbrunner. The union of balls and its dual shape. *Discrete & Computational Geometry*, 13(3):415–440, 1995.
- [45] H. Edelsbrunner and J. Harer. *Computational Topology - an Introduction*. American Mathematical Society, 2010.
- [46] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 454–463. IEEE, 2000.
- [47] S. Eilenberg. Extensions and classification of continuous mappings. In R. L. Wilder and W. L. Ayres, editors, *Lectures in Topology. The University of Michigan Conference of 1940*, pages 57–99. The University of Michigan Press, 1941.
- [48] S. Eilenberg and S. MacLane. Acyclic models. *American Journal of Mathematics*, 75(1):189–199, 1953.
- [49] S. Eilenberg and N. E. Steenrod. *Foundations of Algebraic Topology*. Princeton Mathematical Series. Princeton University Press, 1952.
- [50] P. Gabriel. Unzerlegbare darstellungen I. *Manuscripta Mathematica*, 6:71–103, 1972.

- [51] S. Gelfand and Y. Manin. *Methods of Homological Algebra*. Springer, 2 edition, 2003.
- [52] R. Ghrist. *Elementary Applied Topology*. Createspace, 2014.
- [53] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, 4th edition, 2013.
- [54] G. H. Golub and J. H. Wilkinson. Ill-conditioned eigensystems and the computation of the jordan canonical form. *SIAM Review*, 18(4):578–619, 1976.
- [55] D. Govc and P. Skraba. An approximate nerve theorem. *Foundations of Computational Mathematics*, 18:1245–1297, 2017.
- [56] C. Harris and M. Stephens. A Combined Corner and Edge Detector. In *Alvey Vision Conference*, pages 23.1–23.6, 1988.
- [57] A. Hatcher. *Algebraic Topology*. Cambridge University Press, 2002.
- [58] G. Henselman. Combinatorial foundations in TDA: A morse-theoretic algorithm to compute persistent homology with generators. Talk at the SIAM minisymposium in Applied and Computational Topology, 2017.
- [59] G. Henselman and R. Ghrist. Matroid Filtrations and Computational Persistent Homology. Preprint: <https://arxiv.org/abs/1606.00199>, 2016.
- [60] G. F. Henselman. *Matroids and Canonical Forms: Theory and Applications*. Ph.D. dissertation, University of Pennsylvania, 2017.
- [61] A. S. Householder. *The Theory of Matrices in Numerical Analysis*. Blaisdell Publishing Company, 1964.
- [62] D. Husemöller. *Fibre Bundles*. Graduate Texts in Mathematics 20. Springer New York, 2013.
- [63] M. Kerber and H. Schreiber. Barcodes of towers and a streaming algorithm for persistent homology. *Discrete & Computational Geometry*, 61(4):852–879, 2019.

- [64] M. Kilp, U. Knauer, and A. V. Mikhalev. *Monoids, Acts and Categories: With Applications to Wreath Products and Graphs. A Handbook for Students and Researchers*. De Gruyter, 2011.
- [65] C. L. Lawson, R. J. Hanson, D. R. Kincaid, and F. T. Krogh. Basic linear algebra subprograms for fortran usage. *ACM Trans. Math. Softw.*, 5(3):308–323, 1979.
- [66] A. B. Lee, K. S. Pedersen, and D. B. Mumford. The nonlinear statistics of high-contrast patches in natural images. *International Journal of Computer Vision*, 54:83–103, 2003.
- [67] M. Lesnick. The theory of the interleaving distance on multidimensional persistence modules. *Foundations of Computational Mathematics*, 15:613–650, 2015.
- [68] M. P. Lesnick. *Multidimensional interleavings and applications to topological inference*. Ph.D. dissertation, Stanford University, 2012.
- [69] R. Lewis and D. Morozov. Parallel computation of persistent homology using the blowup complex. In *Proceedings of the 27th ACM symposium on Parallelism in Algorithms and Architectures*, pages 323–331. ACM Press, 2015.
- [70] L.-H. Lim. Hodge laplacians on graphs. *Siam Review*, 2020. To appear.
- [71] A. Maleki, M. Shahram, and G. Carlsson. A near optimal coder for image geometry with adaptive partitioning. In *2008 15th IEEE International Conference on Image Processing*, pages 1061–1064. IEEE, 2008.
- [72] C. Maria, J.-D. Boissonnat, M. Glisse, and M. Yvinec. The gudhi library: Simplicial complexes and persistent homology. In H. Hong and C. Yap, editors, *Mathematical Software – ICMS 2014*, Lecture Notes in Computer Science, pages 167–174. Springer, 2014.
- [73] C. Maria and S. Y. Oudot. Zigzag persistence via reflections and transpositions. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete*

- Algorithms*, Proceedings, pages 181–199. Society for Industrial and Applied Mathematics, 2014.
- [74] C. Maria and H. Schreiber. Discrete morse theory for computing zigzag persistence. In Z. Friggstad, J.-R. Sack, and M. R. Salavatipour, editors, *Algorithms and Data Structures*, Lecture Notes in Computer Science, pages 538–552. Springer International Publishing, 2019.
- [75] J. McCleary. *A User’s Guide to Spectral Sequences*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2001.
- [76] B. H. Menze et al. The multimodal brain tumor image segmentation benchmark (BRATS). *IEEE transactions on medical imaging*, 34(10):1993–2024, 2015.
- [77] N. Milosavljević, D. Morozov, and P. Skraba. Zigzag persistent homology in matrix multiplication time. In *Proceedings of the 27th annual ACM symposium on Computational geometry - SoCG ’11*, page 216. ACM Press, 2011.
- [78] K. Mischaikow and V. Nanda. Morse theory for filtrations and efficient computation of persistent homology. *Discrete & Computational Geometry*, 50(2):330–353, 2013.
- [79] D. Morozov. Dionysus2. Software available at <https://www.mrzv.org/software/dionysus2/>.
- [80] R. Mosher and M. Tangora. *Cohomology Operations and Applications in Homotopy Theory*. Harper’s Series in Modern Mathematics. Harper & Row, 1968.
- [81] J. R. Munkres. *Elements of Algebraic Topology*. Addison-Wesley Publishing Company, 1984.
- [82] P. Niyogi, S. Smale, and S. Weinberger. Finding the homology of submanifolds with high confidence from random samples. *Discrete & Computational Geometry*, 39(1):419–441, 2008.

- [83] N. Otter, M. A. Porter, U. Tillmann, P. Grindrod, and H. A. Harrington. A roadmap for the computation of persistent homology. *EPJ Data Science*, 6(1), 2017.
- [84] S. Y. Oudot. *Persistence Theory: From Quiver Representations to Data Analysis*, volume 209 of *Mathematical Surveys and Monographs*. American Mathematical Society, 2015.
- [85] S. Y. Oudot and D. R. Sheehy. Zigzag zoology: Rips zigzags for homology inference. *Foundations of Computational Mathematics*, 15(5):1151–1186, 2015.
- [86] J. Perea and G. Carlsson. A klein-bottle-based dictionary for texture representation. *International Journal of Computer Vision*, 107(1):75–97, 2014.
- [87] J. A. Perea and J. Harer. Sliding windows and persistence: An application of topological methods to signal analysis. *Foundations of Computational Mathematics*, 15(3):799–838, 2015.
- [88] J. A. Perea Benitez. *Topology of spaces of micro-images, and an application to texture discrimination*. Ph.D. dissertation, Stanford University, 2011.
- [89] R. Sarkar and B. J. Nelson. Texture based classification of seismic image patches using topological data analysis. In *81st EAGE Conference and Exhibition 2019*, pages 1–5. European Association of Geoscientists & Engineers, 2019.
- [90] J.-P. Serre. Homologie singuliere des espaces fibres. *The Annals of Mathematics*, 54(3):425, 1951.
- [91] D. R. Sheehy. Linear-size approximations to the vietoris–rips filtration. *Discrete & Computational Geometry*, 49(4):778–796, 2013.
- [92] G. Singh, F. Memoli, and G. Carlsson. Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object Recognition. In M. Botsch, R. Pajarola, B. Chen, and M. Zwicker, editors, *Eurographics Symposium on Point-Based Graphics*. The Eurographics Association, 2007.

- [93] P. Skraba, G. Thoppe, and D. Yogeshwaran. Randomly weighted d -complexes: Minimal spanning acycles and persistence diagrams. *The Electronic Journal of Combinatorics*, 27(2), 2020.
- [94] P. Skraba and M. Vejdemo-Johansson. Parallel & scalable zig-zag persistent homology. In *Neural Information Processing Systems (NIPS)*, 2012.
- [95] N. Steenrod. *The Topology of Fibre Bundles*. Princeton University Press, 1951.
- [96] N. E. Steenrod. Homology with local coefficients. *Annals of Mathematics*, 44(4):610–627, 1943.
- [97] S. Tanaka. Topological analysis of point singularities in stimulus preference maps of the primary visual cortex. *Proceedings: Biological Sciences*, 261:81–88, 1995.
- [98] A. Tausz. *Extensions and Applications of Persistence Based Algorithms in Computational Topology*. Ph.D. dissertation, Stanford University, 2012.
- [99] A. Tausz, M. Vejdemo-Johansson, and H. Adams. JavaPlex: A research software package for persistent (co)homology. In H. Hong and C. Yap, editors, *Proceedings of ICMS 2014*, Lecture Notes in Computer Science 8592, pages 129–136, 2014. Software available at <http://appliedtopology.github.io/javaplex/>.
- [100] H. Tietze. Über funktionen, die auf einer abgeschlossenen menge stetig sind. *Journal für die reine und angewandte Mathematik*, 145:9–14, 1914.
- [101] J. van Hateren and A. van der Schaaf. Independent component filters of natural images compared with simple cells in primary visual cortex. *Proceedings of the Royal Society of London, Biological Sciences*, 265:359–366, 1998.
- [102] J. W. Vick. *Homology Theory*, volume 145 of *Graduate Texts in Mathematics*. Springer New York, 1994.

- [103] A. C. Wilkerson, H. Chintakunta, and H. Krim. Computing persistent features in big data: A distributed dimension reduction approach. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11–15, 2014.
- [104] H. R. Yoon. *Cellular Sheaves And Cosheaves For Distributed Topological Data Analysis*. Ph.D. dissertation, University of Pennsylvania, 2018.
- [105] S. Zhang, M. Xiao, C. Guo, L. Geng, H. Wang, and X. Zhang. HYPHA: a framework based on separation of parallelisms to accelerate persistent homology matrix reduction. In *Proceedings of the ACM International Conference on Supercomputing - ICS '19*, pages 69–81. ACM Press, 2019.
- [106] A. Zomorodian and G. Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33(2):249–274, 2005.