

ABSTRACT

Title of dissertation: TOPOLOGICAL STRUCTURE OF
 SPATIALLY-DISTRIBUTED NETWORK
 CODED INFORMATION

Brenton Walker, Doctor of Philosophy, 2014

Dissertation directed by: Professor Jonathan Rosenberg
 Department of Mathematics

In this paper we generalize work using topological methods for testing wireless/sensor network coverage to the problem of covering a geographically-distributed wireless network with linear network coded data. We define the coverage complex, a new type of simplicial complex built on the nodes of the network which captures properties of the data coverage, and use tools from algebraic topology, persistent homology, and matroid theory to study it. The coverage complex shares properties with the Rips complex, however it also suffers from a more diverse variety of potential failures. We extend the standard coverage criteria to account for some of these situations using persistent homology, multi-sheeted localized covers of the space, and Mayer-Vietoris sequences. We also investigate the combinatorial properties of the coverage complex, determining the correspondence between it and the lattice of linear subspaces of a vector space. Finally we present algorithms for computing coverage complexes, present a software package designed to compute and experiment with coverage complexes, and provide a summary of ongoing and future work.

TOPOLOGICAL STRUCTURE OF SPATIALLY-DISTRIBUTED
NETWORK CODED INFORMATION

by

Brenton Walker

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2014

Advisory Committee:
Professor Jonathan Rosenberg, Chair/Advisor
Professor Wojciech Czaja
Professor William Goldman
Professor Richard La
Professor James Schafer

© Copyright by
Brenton Walker
2014

Acknowledgments

I would like to thank everyone who supported me on the way to this point. My advisor, Jonathan Rosenberg, and his supreme patience during my meandering path to a thesis topic. He has been available whenever I had a problem to discuss, and willingly joined my adventure into computational algebraic topology. Thanks to my committee members, all of whom have been great teachers and/or collaborators during my time in College Park. Thanks to Michael Postol and Luke, who have been like proxy advisors, always available and ready with advice, support, and new ideas. Thanks to my fellow grad students, Ryan Hoban and Angela Hennessy, for their friendship and countless hours of studying together and mathematical discussions. Thanks to Kathy Dolney and Barb Martinko for leading me to really appreciate math, and realize that I was not that bad at it. Thanks especially to my family and friends who have encouraged and supported me through everything.

Table of Contents

List of Figures	v
1 Network and Erasure Coding	1
1.1 Conventional Model of (Linear) Network Coding	1
1.2 Network and Erasure Coding in Distributed Wireless Networks	7
2 Network Coverage Problems and Topology	16
2.1 Technical Problems and Solutions	20
2.1.1 Defects of the Rips Complex	21
2.1.2 Network Boundary Identification Problems	23
2.1.3 Disconnected Domain Boundaries	25
2.2 Proving the Coverage Criterion	25
2.3 Shadows, Lifts, and Chambers’s Isomorphism	27
2.3.1 The Rips Shadow	27
2.3.2 Shadow Paths and Rips Paths	30
2.3.3 Lifting Shadow Paths	32
2.4 Extending to Network Code Coverage	34
3 The Network Coded Coverage Problem	38
3.1 The Coverage Complex $\mathcal{R}^C(U)$	40
3.1.1 Simple Examples of \mathcal{R}^C	41
3.1.2 Basic Properties of \mathcal{R}^C	43
3.1.3 Basic Data Coverage Results For \mathcal{R}^C	46
3.2 Enumerating the Coverage Complexes	57
3.2.1 Single-Vector Resolvability	69
3.3 Failure of Chambers’s Proof for $\mathcal{R}^C(U)$	71
3.4 Geometric Connections Between \mathcal{X} and $\mathcal{R}^C(U, \mathcal{X})$	75
3.4.1 Layered Loops in $\mathcal{R}^C(U)$	78
3.4.2 Layered 2-simplices in $\mathcal{R}^{C1}(U)$	80
3.4.3 Layered Loops and Resolvability	82
3.5 Computing $\mathcal{R}^C(U)$ and $\mathcal{R}^{C1}(U)$	83

4	Persistent Homology and Discounting Holes in \mathcal{S}	89
4.1	Persistent Homology	93
4.1.1	Applications of Persistent Homology	98
4.2	Structure and Computation of Persistent Homology	102
4.2.1	Persistence Computation Based On Pairing of Simplices	103
4.2.2	Persistence Modules	106
4.2.3	Persistence Computation Based On Persistence Modules	107
4.3	Discounting Holes in \mathcal{S}	113
4.3.1	Allowing changes to $\mathcal{R}^C(U)$	117
5	Using Persistent Homology To Recover Spatial Information From Encounter Traces	119
5.1	The Encounter Complex Metric	122
5.2	Building a Witness Complex	125
5.3	Graph-Based Experiments	127
5.3.1	A Note on Higher Degree Vertices	132
5.3.2	Dynamic Graphs	132
5.4	Two-Dimensional Experiments	134
5.4.1	Detecting Changes in a 2D Space	134
5.4.2	2D Grid With Boundaries vs. a Torus	135
5.5	Experiments With Real Encounter Data	138
6	Patching False Holes in the Coverage Complex	141
6.1	Bubble Boxes and the Mayer-Vietoris Sequence	142
6.2	Local Bases and Local Coverage	147
6.3	Bubble Baubles and Multi-Fold Intersections	152
6.3.1	The Mayer-Vietoris Spectral Sequence	153
6.3.2	$(4n)$ -vertex Bubble Baubles	156
6.3.3	$(4n + 2)$ -vertex Bubble Baubles	158
6.3.4	The Spectral Sequence for the Triple Bubble Box	162
6.4	The Wormhole Complex	164
7	Examples, Software, and Future Work	168
7.1	The CoverageSim Java Package	168
7.1.1	The Non-Lifting	172
7.1.2	The Bubble Box	173
7.1.3	The Layered Loop and Layered Simplex	174
7.1.4	Resolved Hex Points and Resolved Quad Points	175
7.1.5	The Wormhole Complex	176
7.1.6	The Coverage Grid and Fenced Coverage Square	176
7.2	Ongoing and Future Work	180
7.2.1	Traversing the $\mathcal{R}_r^C(\mathcal{X}, U)$ Lattice With Zig-Zag Persistence	180
7.2.2	Filtering $\mathcal{R}_r^C(\mathcal{X}, U)$ by Rips Radius	181
7.2.3	Alternative Coverage Complexes	182
7.2.4	Optimizing Data Arrangements	182

List of Figures

1.1	The butterfly network	4
1.2	A simple index coding example	7
1.3	The network-coded data dissemination scenario	8
1.4	The r -local rank in a region at two hour intervals during an experiment	12
1.5	Plot of the minimum spanning radius during an experiment	13
2.1	Model of a region covered by sensors	17
2.2	Communication radius and coverage radius in sensor networks	18
2.3	The constraint on the ratio between communication and coverage radii	20
2.4	The weakly-fenced sensor coverage model	23
2.5	The relationship between points, Rips complex, and Rips shadow . . .	29
2.6	Example of data coverage with disks	36
3.1	The situation in Lemma 3.31	75
3.2	The situation in Prop 3.33	78
3.3	An example of a layered loop	79
3.4	An example of a layered simplex	80
4.1	Classic example of a filtered simplicial complex and its persistent homology	96
4.2	An example of a sublevel filtration of a manifold	97
4.3	The correspondence between critical values of $f(x)$ and points in the persistence diagram of the sublevel filtration.	98
4.4	An example of a filtered Rips complex built on point cloud data . . .	100
4.5	The persistence barcode for a point cloud filtration	101
5.1	A comparison of encounter patterns possible in different spaces	120
5.2	The witness complex and persistence barcode for an experiment in the two-loop graph space	128
5.3	The persistence barcodes for an experiment in a three-loop space . . .	130
5.4	The witness complex obtained from the expanding/contracting loop experiment	133

5.5	The witness complex obtained for the repulsion phase of the 2D random walk experiment	135
5.6	The persistence barcodes for the torus and bounded rectangle experiments	136
5.7	The persistence barcodes of the Haggie Cambridge Computer Lab experiment	140
6.1	A data arrangement containing three bubble boxes	148
6.2	The three sheets of a triple bubble box	149
6.3	A data arrangement for a bubble octagon over $V = \mathbf{GF}(2)^4$	157
6.4	A data arrangement for a bubble hexagon over $V = \mathbf{GF}(2)^3$	159
6.5	Labeled elements of a bubble hexagon	159
6.6	Labeled elements of a triple bubble box	162
6.7	The essential structure of the wormhole complex	165
6.8	The full wormhole complex	166
7.1	The non-lifting example	172
7.2	The triple bubble box with generators	173
7.3	A grid of bubble boxes	173
7.4	A layered loop	174
7.5	A layered simplex	174
7.6	An example of resolvable quad points	175
7.7	The wormhole complex	176
7.8	Some examples of Coverage Grids	177
7.9	Some examples of Fenced Coverage Squares	179

Chapter 1: Network and Erasure Coding

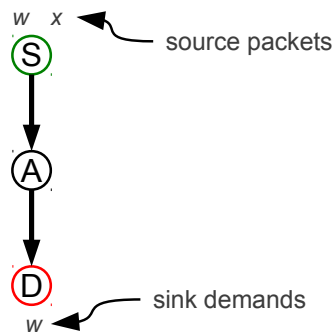
In this section we will provide an introduction to network coding and network coded data distribution which motivates the results in this paper. We will also summarize prior work on topological testing of wireless/sensor network coverage. Network coding is a huge and active field, and our goal here is not to give a survey, but just some context.

1.1 Conventional Model of (Linear) Network Coding

Network coding is a more robust and efficient generalization of traditional routing in data networks. In traditional routing, *packets* of data are forwarded from one node to another across network links. In network coded routing, nodes are allowed to create and transmit linear combinations of the packets. Interest in network coding has exploded since the publication of several seminal papers on the topic around 2000 [1, 2].

Information theorists use a simplified and abstract model of data networks to study network coding. A network is a directed graph consisting of a set of nodes, P , and links $\mathcal{L} \subseteq P \times P$. Some nodes in the network are designated as *source nodes* at which data packets will originate, and some others are designated as *sink nodes* which

have a demand for some packets coming from the source nodes. Each link connects two nodes and has some (integer) capacity value, specifying how many packets it can carry. Typically the capacity will be indicated by drawing multiple links between nodes (essentially making the network a directed multi-graph). A network coding problem consists of some such network, with the data packets initially held by source nodes and the demands of the sink nodes labeled as in the diagram below.



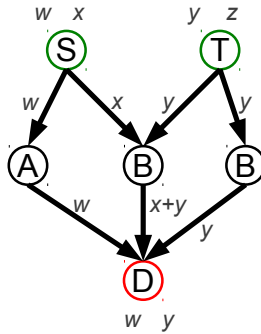
A solution to the network coding problem consists of an assignment of packets to each edge of the network so that:

- The number of items assigned to each link does not exceed its capacity.
- The packets on the outgoing links from any node must be a linear combination of the packets on the incoming links.
- The packets on the incoming links to the sink must have the sink's demands within their linear span.

Solutions may or may not be unique, depending on the network, source packets, and sink demands. A solution to the example above would be:



A more complicated example might involve multiple sources, multiple routes, and multiple demands.



The canonical example of a network which can be solved with network coding but not with traditional routing is the butterfly network shown in figure 1.1. The source, S holds two packets, x and y , and both sink nodes, E and F demand each of those packets. Each intermediate node except for C has only one input, and therefore no choice over what to put on its outgoing links. On the outgoing link from C we must choose to send either x or y . Either choice will result in failing to meet the demands of either E or F . Figure 1.1a depicts the choice of x , which leads to E receiving two copies of x and no copies of y .

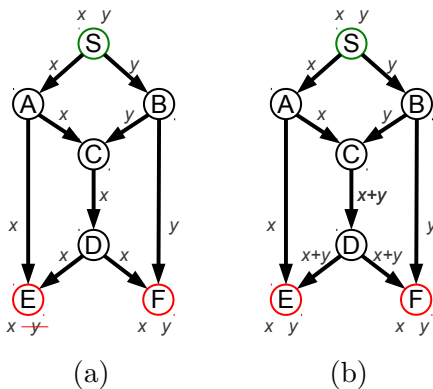


Figure 1.1: The butterfly network. In (a) with just traditional routing the demands of the sink nodes cannot be met. In (b) the network is solved using network coding, sending the linear combination of two packets on the link from C to D .

In figure 1.1b network coding is used; the linear combination $x + y$ is sent on the outgoing link from C , and then on both outgoing links from D . In this case E receives a copy of x , and can solve for $(x + y) + x = y$. Similarly F can recover x .

The number of unique source packets in the network is called the *dimension* of the network, because the linear combination of packets on any outgoing link can be specified as a vector of that dimension. The butterfly network example, for instance, has dimension 2. In the simplest examples of network coding each packet can be considered as a single bit (an element of $GF(2)$) and the linear combinations will have coefficients in that same field. In more generality the packets and linear coefficients can be elements of any finite field. In fact not all networks are solvable over all fields. Furthermore, the technique described here is sometimes called *scalar network coding* to distinguish it from *vector network coding* where messages are strings of elements from a finite field, and they can be fragmented into smaller packets, which are then subject to coding. In fact it turns out that some networks are vector linear solvable, but not scalar linear solvable [3].

Solvability deals with the ability to satisfy the demands of sink nodes under the constraints of the network; however, much network coding research focuses on showing that certain network codes are optimal for certain networks. Optimality in these cases deals with the set of sources and sinks for which the network satisfies the max flow-min cut bound.

Matroid theory has naturally been tied into network coding. [4, 5] introduced the notion of *matroidal networks* and an algorithm for deriving a matroidal network from any matroid. The abstract notion of independence and dependence in matroid theory naturally capture the requirement that every node's out edges are dependent on its in edges. The authors prove that any scalar linear solvable network is matroidal over a representable matroid. Then by applying their matroid-to-network construction to non-representable matroids, they derive a series of networks which are not scalar linear solvable.

Recently sheaf cohomology has been connected to network coding [6]. Given a network code the authors derive an *NC sheaf*, and by studying its cohomology are able to prove the max flow-min cut bound, as well as state a criterion for the network to be robust against link failure.

Another popular, and certainly more practical, research area is random linear network coding. In random linear coding the linear combinations generated at each node are random. This type of work takes advantage of the following fact about finite dimensional vector spaces over finite fields.

Proposition 1.1. *Let $\mathbb{F} = GF(q)$ be a finite field, and let $V = \mathbb{F}^D$ be a D -dimensional vector space. Let $\mathcal{M}_1 = \{v_1, \dots, v_n\} \subset V$ and $\mathcal{M}_2 = \{w_1, \dots, w_m\} \subset V$ be two collections of vectors in V , such that $\langle v_1, \dots, v_n \rangle \not\subseteq \langle w_1, \dots, w_m \rangle$. If a_1, \dots, a_n are chosen uniformly randomly from \mathbb{F} , then $a_1v_1 + \dots + a_nv_n \notin \langle w_1, \dots, w_m \rangle$ with probability at least $1 - 1/q$.*

This means that if we randomly choose a linear combination from one set of vectors, there is a high probability that it will be outside the span of another set. It happens that random linear coding tends to produce very simple algorithms that are extremely efficient at disseminating data through a network.

Somewhat closer to our application is the problem of index coding. In an index coding problem a central node has copies of several messages, and is able to broadcast to a set of clients. Each client has a list of messages it demands, but may also have some preexisting *side information*; that is, some number of messages or linear combinations of messages it has already collected. The objective is for the central server to satisfy all the clients' demands with the minimum number of broadcasts. A simple example of the index coding problem is depicted in figure 1.2.

The index coding problem turns out to be much more complex than one might expect. In fact it is shown in [7] that index coding is at least as difficult as network coding in the sense that any network coding problem can be reduced to an index coding problem. However index coding problems only have one broadcast domain, and therefore the geometry of the node arrangement is not an issue. When multiple server nodes are involved, this index coding begins to intersect with the problem

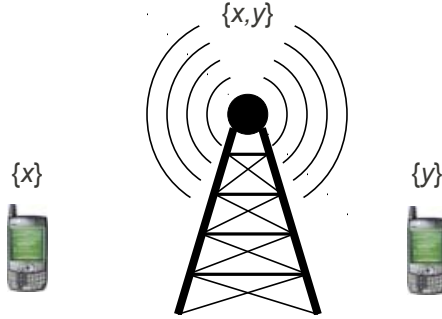


Figure 1.2: A simple example of the index coding problem. Both clients demand both packets, x and y , and each has one packet of side information. The central server can transmit x , then transmit y , or it can satisfy both demands with a single transmission of $x + y$.

of interference alignment, which aims to find ways to avoid interference between proximal transmitters.

1.2 Network and Erasure Coding in Distributed Wireless Networks

We are interested in an application of network coding for data dissemination in distributed wireless networks. This is just for the sake of introduction and motivation, so we will describe the problem for random linear binary codes, but it easily generalizes to linear coding over larger finite fields. Suppose we have a collection of (stationary) wireless nodes in a region of the plane, $\mathbf{X} = \{X_1, \dots, X_N\} \subset \mathbb{R}^2$, and a large bundle of L bits of data. A mobile source node travels through the network distributing pieces of the bundle to the stationary nodes as shown in figure 1.3. Then a mobile destination node travels through the network attempting to collect the data bundle. Because the source and destination nodes are mobile and the connections are opportunistic, it isn't possible to transfer the data bundle in a single session. The naive solution is to chop the bundle into $M = \lceil \frac{L}{k} \rceil$ fragments of

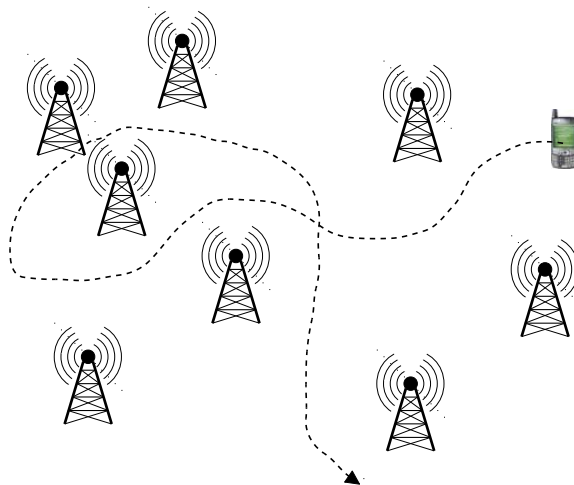


Figure 1.3: The network-coded data dissemination scenario. A mobile node travels through a region covered by stationary servers, distributing fragments of a large data bundle. A destination node also travels through the region, trying to collect the pieces and reconstruct the data bundle.

k bits each, where k is small enough that at least one fragment can be transmitted during each contact. The source node could then distribute random fragments to the stationary nodes, and the destination node could pick them up in the same way.

The classic problem with plain fragmentation is the coupon collector's problem. If there are M fragments and the destination has collected $M - 1$ of them, the probability that the next one it finds will be the missing piece is $\frac{1}{M}$. One elegant solution to the coupon collector's problem is to use erasure coding. In erasure coding the source node distributes k -bit blocks of data that are encoded in such a way that the destination can collect *any* M of them and recover the original data bundle. The encoded blocks can be mathematically structured to ensure that any M of them are enough to reconstruct the bundle, or random linear combinations, in which case it may be necessary to collect slightly more than M encoded fragments. By prop 1.1, with high probability it suffices to collect only slightly more than M .

Using erasure coding in opportunistic networks was first proposed around 2005. In [8] the authors propose using erasure coding in conjunction with existing opportunistic routing methods, and [9] proposes a route selection method based on portfolio theory to balance risk of sending encoded fragments over different unreliable routes. The traditional erasure coding used in these early efforts only allowed encoded fragments to be generated at the source node; intermediate nodes simply forward and exchange the encoded fragments verbatim. However, in the case of random linear coding, adding together encoded fragments produces a new linear combination of the original fragments, so it is reasonable to allow intermediate nodes to create new linear combinations from old ones. [10] presented and mathematically modeled a random linear coding scheme for effective epidemic routing wherein all nodes constantly generate new linear combinations of the encoded fragments that they carry, and [11] presents simulation results showing that allowing such re-encoding can be beneficial. [12] gives details of an implementation of an opportunistic bundle router based on this random linear coding algorithm.

In this paper we will only consider linear codes. In this motivating example in particular, we will focus on linear binary coding. We define some terminology to make the motivation more clear.

Data Bundle: a large block of L bits of data that is to be distributed.

Fragment f_i : a data bundle of length L is split into $M = \lceil \frac{L}{k} \rceil$ fragments of k bits, which can be thought of as k -long binary vectors, $f_i \in \text{GF}(2)^k$.

Coefficient vector \mathbf{c} : each encoded fragment has associated with it a vector $\mathbf{c} = [c_1, c_2, \dots, c_M]$, where $c_i \in \text{GF}(2)$ and $i \in [1, M]$, specifies which fragments were summed (**xor**) to create that encoded fragment.

Encoded fragment $w_{\mathbf{c}}$: an encoded bundle made up of some linear combination of fragments such that $w_{\mathbf{c}} = \sum_{i=1}^M c_i f_i$ for some coefficient vector \mathbf{c} .

Encoding Inventory \mathcal{M}_X : for a given node, X , its encoding inventory is the collection of coefficient vectors of the encoded fragments it has stored.

Encoding Matrix \mathbf{E}_X : for a given node, X , a matrix whose rows contain the coefficient vectors of a node's encoding inventory.

If a node, X , collects an encoding inventory such that its encoding matrix, \mathbf{E}_X , has full rank, then it can invert \mathbf{E}_X to recover the original fragments and thereby the original data bundle. Specifically, assume \mathbf{E}_X is an $M \times M$ full rank encoding matrix (if \mathbf{E}_X has rank M , but more than M rows, we can discard any rows that are linearly dependent). Say the rows of \mathbf{E}_X are $\mathbf{c}_1, \dots, \mathbf{c}_M$, corresponding to encoded fragments $w_{\mathbf{c}_1}, \dots, w_{\mathbf{c}_M}$. Let $\mathbf{d}_1, \dots, \mathbf{d}_M$ be the rows of \mathbf{E}_X^{-1} . Then since

$$w_{\mathbf{c}_i} = \sum_{j=1}^M c_{i,j} f_j \tag{1.1}$$

we know

$$f_i = \sum_{j=1}^M d_{i,j} w_{\mathbf{c}_j} \tag{1.2}$$

With respect to a given node, X , with encoding inventory \mathcal{M}_X and encoding matrix \mathbf{E}_X , an encoded fragment is **innovative** if adding it to X 's inventory increases the rank of \mathbf{E}_X . A complementary notion of *knowledge* is coined in [13]. A node, X , **knows about** coefficient vector \mathbf{c} if \mathbf{c} is not orthogonal to X 's existing inventory; that is, if $\mathbf{E}_X \mathbf{c} \neq 0$. Note that in a vector space over a finite field it is possible for vectors to be self-orthogonal, that is, $\mathbf{c} \neq 0$ but $\mathbf{c} \cdot \mathbf{c} = 0$. However the notion of *knowledge* is still useful because of the following fact (Lemma 4.2 in [13]):

Lemma 1.2. *If node X knows about coefficient vector \mathbf{c} and transmits a new encoded fragment, generated by a random linear combination of its inventory, to node Y , then Y knows about \mathbf{c} with probability at least $\frac{1}{2}$. If the encoding is done over finite field $\text{GF}(q)$, then Y knows about \mathbf{c} with probability at least $1 - \frac{1}{q}$. Furthermore \mathbf{E}_X has full rank iff X knows about every non-zero vector in $\text{GF}(q)^M$.*

Conventionally in opportunistic network research, the special case where only the source node generates encoded fragments is referred to as *Erasur Coding* (EC), and the more general case where in-network coding is allowed, and all nodes may generate new linear combinations, is called *Network Coding* (NC). In [11] the authors simulate and compare random linear NC and EC, and in [14] experiments with them in several scenarios using the implementation described in [12].

The experiments in [14] are one of the motivations for this work. The authors ran experiments with the mobile scenario described above, and attempt to quantify how well the source node is distributing information to the stationary nodes. One way to quantify the quality of code distribution at a point $p \in \mathbb{R}^2$ is to draw a circle

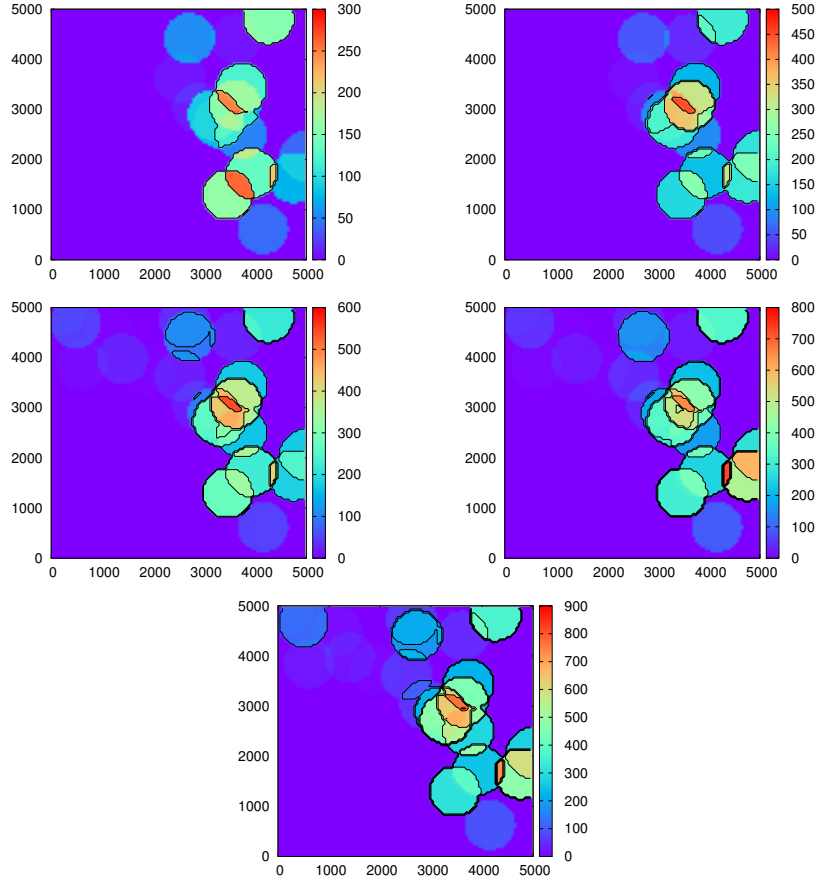


Figure 1.4: The r -local rank in a region at two hour intervals during an experiment. The x and y axes are physical coordinates, and the color gradient labels the r -local rank at each point in the region. The maximum rank in this experiment is 1000.

of radius r around p , and consider the combined encoding inventories of the nodes enclosed by that circle. We call this quantity the **r -local rank**, and denote it by $\text{Rk}_r(p)$. Figure 1.4 shows the evolution of the r -local rank at a sequence of times during an experiment.

The r -local rank gives some quantification of the quality of a distribution of encoded fragments. Turning this metric around, however we note that the destination node's goal is to collect enough encoded fragments to reconstruct the entire data bundle. Therefore we want to quantify the amount of *work* necessary to do

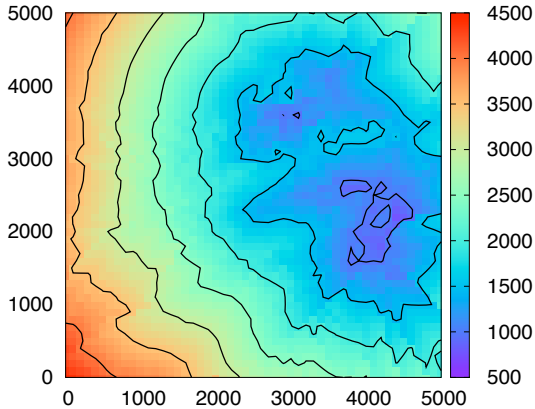


Figure 1.5: Contour plot of the minimum spanning radius in a $5\text{km} \times 5\text{km}$ area at the end of a 10-hour 50-node experiment. Blue indicates small MSR, and therefore a high concentration of code diversity.

that at different points in the region. We define the **minimum spanning radius** at point $p \in \mathbb{R}^2$ as the radius of the smallest circle centered at p that encloses nodes whose encoding inventories are sufficient to reconstruct the data bundle.

$$R_{\text{span}}(p) = \arg \min_r [R_r^t(x, y) = M] \quad (1.3)$$

where M is the length of the coefficient vector (*i.e.* full rank). For this metric, small values indicate a high concentration of code diversity. A contour plot of the minimum spanning radius for one bundle at the end of an experiment is shown in Figure 1.5. As one would expect, the destination tends to collect a full basis more quickly when it starts out in areas of higher code diversity [14].

These metrics characterize coverage entirely in terms of rank, and omit any information about the actual vector spaces spanned. In order to diagnose problems in information coverage in a network, or just generally to understand the mathe-

mathematical structure of the problem, it would be useful to know what subspaces are *not* spanned in regions of sparse code coverage. The minimum spanning radius can be generalized to characterize the coverage of specific subspace. Let $V = \text{GF}(2)^M$ be the vector space of all coefficient vectors and let $U \subseteq V$ be a linear subspace. Define the **neighborhood encoding inventory** of radius r at the point p to be:

$$\mathcal{M}_r(p) = \bigcup_{X \in \mathbf{X} \text{ s.t. } |X-p| \leq r} \mathcal{M}_X \quad (1.4)$$

Then we can define the **minimum spanning radius for U** at point p as:

$$R_{\text{span}}(p, U) = \arg \min_r [U \subseteq \langle \mathcal{M}_r(p) \rangle] \quad (1.5)$$

For a fixed radius r_0 and a subspace $U \subseteq V$, this allows us to define a region of the plane where the work necessary to collect a set of encoded fragments spanning U is (in some sense) bounded by a circular region of radius r_0 .

$$\mathcal{C}_{r_0}^{\text{Cov}}(U) = \{p \in \mathbb{R}^2 \mid R_{\text{span}}(p, U) \leq r_0\} \quad (1.6)$$

On the other hand it allows us to define a region where any node confined within a circular region of radius r_0 can never collect the encoded fragments necessary to span U .

$$\mathcal{C}_{r_0}^{\text{NCov}}(U) = \{p \in \mathbb{R}^2 \mid R_{\text{span}}(p, U) > r_0\} \quad (1.7)$$

Thus we can define regions where there are lower and upper bounds on the work necessary to reconstruct the data bundle.

Unfortunately exactly computing these regions requires detailed location information about the nodes, and even with that information is a difficult computational geometry problem. We will take inspiration from prior work on homological testing of sensor network coverage to try and derive more practical and computationally tractable criteria for testing information coverage in a geographically distributed wireless network.

Chapter 2: Network Coverage Problems and Topology

The premise of a sensor network is that a large number of low-cost low-power wireless nodes with some sort of sensor attached are scattered around a domain $\mathcal{D} \subseteq \mathbb{R}^2$. For example sensors could be collecting temperature or air quality readings, or they could be seismic or motion-sensitive sensors that detect intruders in an area. A simple model of such a sensor network assumes that each sensor has a circular coverage area with a fixed radius within which its measurements are effective or valid. Naturally one would like to distribute sensors in such a way that every point in the domain is within range of at least one sensor; that is, the domain is *covered* by sensors, as depicted in figure 2.1. A more stringent and robust criterion is to ensure that every point is *k-covered*, that is, every point in the domain is within the range of at least k sensors.

Algebraic topology has been employed as a tool for testing the geographic coverage of a domain by radios and sensors with uniform circular ranges [15–18]. If $\mathbf{X} = \{X_1, \dots, X_N\} \subseteq \mathbb{R}^2$ are the locations of the sensors, and their sensing radius is r_s , then the criterion we want to test is that $\mathcal{D} \subseteq \cup_{i=1}^N B(X_i, r_s)$. For now assume that \mathcal{D} is compact Hausdorff and simply connected. In this case a reasonable homological criterion for coverage would be that:

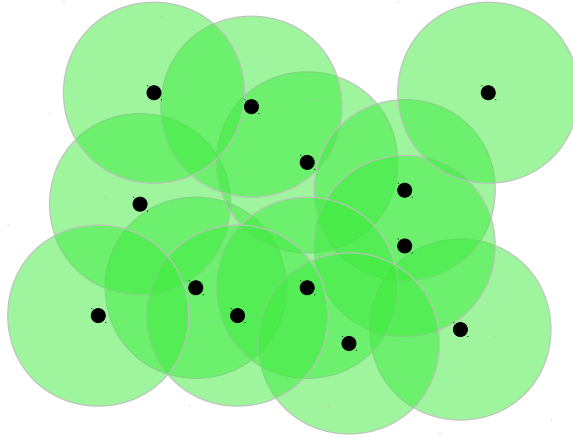


Figure 2.1: In the sensor network coverage problem the goal is to cover a domain, $\mathcal{D} \subseteq \mathbb{R}^2$ with balls of a certain radius, with no gaps or holes.

- the domain's boundary is covered, $\partial\mathcal{D} \subseteq \bigcup_{i=1}^N \text{B}(X_i, r_s)$
- the covered domain is connected, $\tilde{H}_0(\bigcup_{i=1}^N \text{B}(X_i, r_s)) = 0$
- the covered domain has no non-trivial 1-cycles, $H_1(\bigcup_{i=1}^N \text{B}(X_i, r_s)) = 0$

In order to compute homology groups in practice we need something more discrete than a union of balls. To make the problem more computationally tractable we would use the Čech complex, or nerve, of the coverage balls, which we'll denote by $\mathcal{C}_{r_s}(\mathbf{X})$. The Čech complex is a simplicial complex whose 0-simplices correspond to the individual balls $\{\text{B}(X_i, r_s)\}_{X_i \in \mathbf{X}}$, and whose k -simplices correspond to k -fold intersections of the balls. The Čech theorem states that the Čech complex has the same homotopy type as the union of the balls, and therefore

$$H_k \left(\bigcup_{i=1}^N \text{B}(X_i, r_s) \right) = H_k(\mathcal{C}_{r_s}(\mathbf{X}))$$

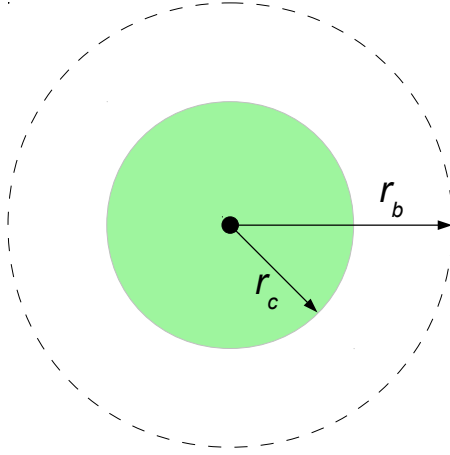


Figure 2.2: In the sensor network model each node has a sensor coverage radius, r_c , and a broadcast radius, r_b , within which it can detect and communicate with its neighbors.

Unfortunately the Čech complex isn't accessible to us in practice. Since GPS is expensive and energy-intensive, and generally requires either a large antenna or a clear view of the sky, it is reasonable to expect that the low-cost low-power nodes in a sensor network do not know their absolute locations or orientations. The only local information they can gather is the identities of their immediate neighbors, and possibly some rough distance indication. Even if the location information were available, computing all k -fold intersections of a collection of balls is a surprisingly complex problem, and it is desirable to build coverage criteria requiring only local information.

Therefore we fall back to the Rips complex, which is the clique complex on the connectivity graph of the network.

Definition 2.1. Given a collection of points $\mathbb{X} = \{X_1, \dots, X_N\}$ in a metric space with metric $d(\cdot, \cdot)$, the **Rips complex**, $\mathcal{R}_r(\mathbb{X})$ for radius r is the simplicial complex defined by:

- $[X_i] \in \mathcal{R}_r(\mathbb{X})$
- $[X_i X_j] \in \mathcal{R}_r(\mathbb{X}) \Leftrightarrow d(X_i, X_j) \leq r$
- $[X_{i_1} \dots X_{i_k}] \in \mathcal{R}_r(\mathbb{X}) \Leftrightarrow [X_s X_t] \in \mathcal{R}_r(\mathbb{X}) \forall s, t \in \{i_1, \dots, i_k\}$

When the vertex set \mathbb{X} is clear from the context we will simply write \mathcal{R}_r .

Part of the premise of the sensor network is that the sensor nodes have radii for communication and that they can detect and communicate with their neighbors within a *broadcast radius* $r_b > r_c$. This allows the nodes to build a connectivity graph, which becomes the 1-skeleton of a Rips complex.

Since the goal is to cover the domain \mathcal{D} with balls of radius r_c , but we are assumed to have a Rips complex with radius r_b , certain constraints the ratio of the two are required to ensure that homological properties of the Rips complex imply coverage by the sensors. The criterion for planar points is that $r_c \geq r_b/\sqrt{3}$, which is exactly the ratio required to cover an equilateral triangle with sides of length r_b with balls of radius r_c as shown in figure 2.3. Essentially this ensures that if a simplex $\sigma = [X_0 \dots X_k] \in \mathcal{R}_{r_b}$, then the convex hull $\text{conv}(X_0, \dots, X_k)$ is covered by corresponding balls of radius r_c .

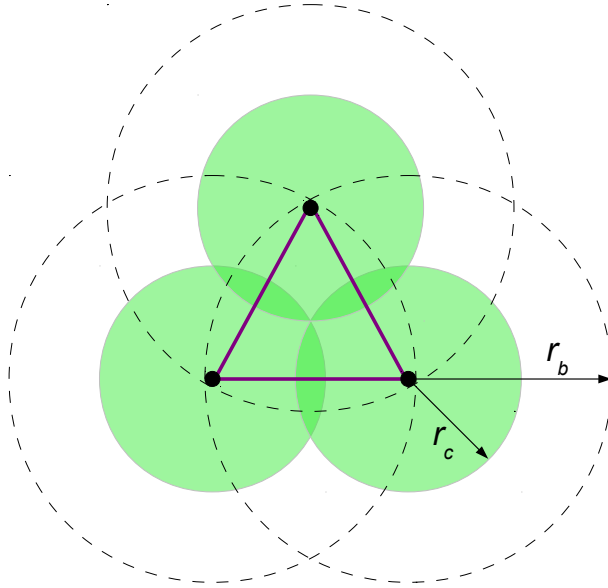


Figure 2.3: The ratio between r_c and r_b is constrained by the worst-case of an equilateral triangle.

2.1 Technical Problems and Solutions

There are three main complications that come up in homological sensor network coverage criteria. First are the geometric defects of the Rips complex; the Čech theorem does not apply to the Rips complex, and the Rips complex does not generally have the homotopy type of the union of the coverage areas. The relationship between the coverage radius and broadcast radius complicates this further. Second is the problem of dealing with the boundaries of the network. The network coverage criteria that solve the first problem require identifying a sub-complex of the Rips complex that roughly, geometrically, corresponds to the geographic boundaries of the domain, \mathcal{D} . Third is the possibility of \mathcal{D} itself having holes; that is, if $\partial\mathcal{D}$ is not connected. Because of the first problem, all the homological tests for network coverage can be inconclusive in the negative case. The tests may detect a flaw in

the network, even when the coverage is complete. On the other hand, a positive result is conclusive; when the homological coverage tests succeed, it implies that the coverage is complete.

2.1.1 Defects of the Rips Complex

The Cech theorem does not apply to the Rips complex, and the Rips complex does not generally have the homotopy type of the union of the coverage areas. Non-trivial (and high dimensional) homological features can appear in the Rips complex $\mathcal{R}_r(\mathbf{X})$, even when the corresponding Cech complex $\mathcal{C}_{r/2}(\mathbf{X})$ is contractible. A key theorem proved by de Silva and Ghrist captures the homology of the Cech complex between the homologies of two Rips complexes [15, 16]:

Theorem 2.2. *Given a collection of points $\mathbf{X} = \{X_1, \dots, X_N\} \subseteq \mathbb{R}^2$ and given two positive radii $r > r'$ such that $\frac{r}{r'} \geq \frac{2}{\sqrt{3}}$, then*

$$\mathcal{R}_{r'}(\mathbf{X}) \subseteq \mathcal{C}_{r/2}(\mathbf{X}) \subseteq \mathcal{R}_r(\mathbf{X}) \tag{2.1}$$

This means that the map induced by the inclusion

$$i_* : H_k(\mathcal{R}_{r'}) \rightarrow H_k(\mathcal{R}_r) \tag{2.2}$$

passes through the homology of the Cech complex.

$$\begin{array}{ccc}
H_k(\mathcal{R}_{r'}) & \xrightarrow{i_*} & H_k(\mathcal{R}_r) \\
& \searrow & \nearrow \\
& H_k(\mathcal{C}_{r/2}) &
\end{array} \tag{2.3}$$

Therefore if $0 \neq [\alpha] \in H_k(\mathcal{R}_{r'})$ and if $i_*([\alpha]) \neq 0$, then $[\alpha]$ corresponds to a non-zero element of $H_k(\mathcal{C}_{r/2})$, and therefore a non-trivial feature in the actual sensor coverage area.

Using this result requires a complication of the assumptions of the sensor network model. In [16] de Silva and Ghrist further assume that each node has a sensor with coverage radius r_c , and a radio that can communicate with and detect neighbors within a *weak signal radius* r_w , and can also detect when its neighbors are within a *strong signal radius* $r_s < r_w$ as shown in figure 2.4. The geometric constraint on these radii in the planar case is $r_w \geq r_s \frac{2}{\sqrt{3}}$.

A similar tool is used by Chambers to deal with quasi-Rips complexes of planar point sets [19, 20]. Testing the longevity of generators of homology groups across a filtered complex can be done by computing the persistent homology of the filtration.

This theorem gives a sufficient, but not necessary, criterion for identifying non-zero elements of $H_k(\mathcal{C}_{r/2})$. Therefore it is only useful for coverage criteria where the existence of such a non-zero element implies coverage. It isn't useful for criteria based on $H_1(\mathcal{R})$ and $H_1(\mathcal{C})$, where the existence of a non-zero element would indicate a coverage defect.

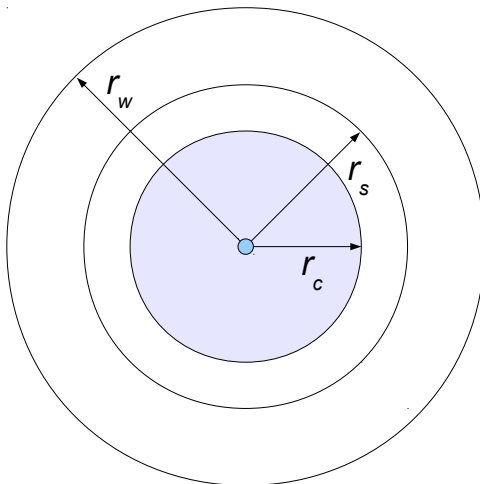


Figure 2.4: In the weakly-fenced sensor network model each node has a sensor coverage radius, r_c , and two communication radii. It can distinguish between neighbors within a strong communication radius, r_s , and a (longer) weak communication radius, r_w .

2.1.2 Network Boundary Identification Problems

A second challenge for homological coverage testing is identifying the network boundaries. Considering a collection $\mathbb{X} = \{X_1, \dots, X_N\}$ of nodes in the plane, in general the notion of the boundary of the network is not well defined. In general it isn't possible to deduce, based on connectivity information alone, which nodes are at the edge of the network. The simplest solution, taken by de Silva and Ghrist in [21], is to assume prior knowledge of the network boundary; that the boundary of the domain, $\partial\mathcal{D}$, is connected and piecewise linear, with specially identified *fence nodes* at the vertices, and that neighboring fence nodes are within communication range of each other. They call this subcomplex the *fence cycle*, $\mathcal{F} \subset \mathcal{R}_{r_b}$.

Other papers by de Silva and Ghrist make weaker assumptions about having prior information about the fence cycle. In [15, 18] they assume that the nodes can

detect when they are within a certain distance, the *fence radius* r_f , of the domain boundary $\partial\mathcal{D}$, and define this subset of nodes to be the *fence nodes*. They define the *fence complex*, \mathcal{F}_r , for a given radius r , to be the Rips complex on this subset of points. They define a sub-domain $\mathcal{C} \subseteq \mathcal{D}$ called the *collar* which are the areas within $r_f + \frac{1}{2}r_s$ of $\partial\mathcal{D}$, and concentrate on coverage tests for the interior, $\mathcal{D} - \mathcal{C}$.

This more complicated criterion requires additional assumptions about the curvature and shape of the boundary. The boundary cannot be too undulating, and the interior of the domain $\mathcal{D} - \mathcal{C}$ must be connected.

In fact, the problem of identifying boundary cycles in a sensor network without location information is a serious research topic in itself [22]. The main practical methods proposed for network boundary detection involve either finding nodes with lower degrees of connectivity (statistical considerations), or detecting certain topological structures or discontinuities and extremities in the shortest paths through the network (routing considerations) [22, 23]. Neither of these seems to provide the strong geometric guarantees required in de Silva and Ghrist’s work. Logistically the the assumption of prior knowledge of the fence cycle seems just as reasonable, and much simpler, than the fence detection assumptions.

The concept and the metaphor is intuitive, but de Silva and Ghrist and others use the term “fence” to mean different things in different settings. We will give a a more precise definition in section 3.1.3 before we use it formally.

2.1.3 Disconnected Domain Boundaries

A third challenge for homological criteria for network coverage is the possibility of internal boundaries, and the shape of the coverage domain, \mathcal{D} . If the domain \mathcal{D} itself contains holes, then any criterion based *solely* on testing H_1 of the Rips complex will be insufficient. Instead of looking for holes in coverage by testing $H_1(\mathcal{R}_{r_b})$, de Silva and Ghrist consider the second relative homology $H_2(\mathcal{R}_{r_b}, \mathcal{F})$. If this is non-zero, then they can *almost* conclude that the domain \mathcal{D} is fully covered by sensors. It is also necessary to exclude features of H_2 that arise because of Rips complex complications; either using lemma 2.2 or by explicitly requiring a generator $[\alpha] \in H_2(\mathcal{R}_{r_b}, \mathcal{F})$ such that $\partial\alpha \neq 0$. This criterion is effective even if the boundary of the domain, $\partial\mathcal{D}$ is disconnected, as long as the outer and inner boundaries are distinguished. In chapter 3.1.3 we will generalize these results slightly for our purposes and then build on them in chapter 4.

2.2 Proving the Coverage Criterion

This summarizes de Silva and Ghrist's proof for the simplest case, where the domain \mathcal{D} is simply connected and the Rips complex has a well-identified subcomplex \mathcal{F} such that the projection $p : \mathcal{F} \rightarrow \partial\mathcal{D} \cong S^1$ is a homeomorphism. We will deal with this situation and some extensions more formally in chapter 3.1.3.

The link between the Rips complexes and the geometry comes from the projection map $p : \mathcal{R}_r(\mathbf{X}) \rightarrow \mathbb{R}^2$ which maps each sensor vertex $X_i \in \mathbf{X}$ to its location in the plane, and maps each simplex $\sigma = [X_0 \dots X_k] \in \mathcal{R}_r(\mathbf{X})$ to its convex hull

in the plane, $p([X_0 \dots X_k]) = \text{conv}(X_0, \dots, X_k)$. This induces maps between the long exact sequences for the pairs $(\mathcal{R}_{r_b}, \mathcal{F})$ and $(\mathbb{R}^2, \partial\mathcal{D})$, giving the commutative square:

$$\begin{array}{ccc} H_2(\mathcal{R}_{r_b}, \mathcal{F}) & \xrightarrow{\delta_*} & H_1(\mathcal{F}) \\ \downarrow p_* & & \downarrow p_* \\ H_2(\mathbb{R}^2, \partial\mathcal{D}) & \xrightarrow{\delta_*} & H_1(\partial\mathcal{D}) \end{array} \quad (2.4)$$

By our assumptions the map $p_* : H_1(\mathcal{F}) \rightarrow H_1(\partial\mathcal{D})$ on the right side of this box is an isomorphism. Assume that we have $[\alpha] \in H_2(\mathcal{R}_{r_b}, \mathcal{F})$ with $\partial\alpha \neq 0$. Then $p_*\delta_*[\alpha] = p_*[\partial\alpha] \neq 0$. Therefore $\delta_*p_*[\alpha] \neq 0$, so $p_*[\alpha] \neq 0$.

Let $\mathcal{U} = \mathcal{D} \cap \left(\bigcup_{i=1}^N B_{r_c}(X_i) \right)$ be the subset of the domain, \mathcal{D} , actually covered by sensors. Suppose there is some point in \mathcal{D} not covered, $x \in \mathcal{D} - \mathcal{U}$. Then $p : (\mathcal{R}_{r_b}, \mathcal{F}) \rightarrow (\mathbb{R}^2 - x, \partial\mathcal{D})$ and we can factor the left side of the box:

$$\begin{array}{ccccc} & & H_2(\mathcal{R}_{r_b}, \mathcal{F}) & \xrightarrow{\delta_*} & H_1(\mathcal{F}) \\ & \swarrow p_* & \downarrow p_* & & \downarrow p_* \\ H_2(\mathbb{R}^2 - x, \partial\mathcal{D}) & & & & \\ & \searrow i_* & H_2(\mathbb{R}^2, \partial\mathcal{D}) & \xrightarrow{\delta_*} & H_1(\partial\mathcal{D}) \end{array} \quad (2.5)$$

But $H_2(\mathbb{R}^2 - x, \partial\mathcal{D}) = 0$, so $p_*[\alpha] = 0$, a contradiction. So there can be no point $x \in \mathcal{D}$ not covered by \mathcal{U} .

In the case with dual communication radii, $r_w > r_s > r_c$, the proof is essentially the same, but the details get much more complicated.

2.3 Shadows, Lifts, and Chambers's Isomorphism

Some of the interesting structure in the sensor network coverage problem comes from the fact that de Silva and Ghrist's model assumes a sensing radius that is less than the communication radius. If one looks at the coverage provided by balls with the same radius as the Rips radius, and restricting the Rips vertices to \mathbb{R}^2 , then the result is much simpler and stronger. The fundamental group of the Rips complex is isomorphic to that of the *shadow* of the Rips complex. The Rips complex may still have exotic topological features in higher dimensions, but the one-dimensional features are well behaved. This was proved by Chambers et al. [19,20]. Here we will summarize parts of that work. We will use similar notions to define data coverage, and it will be informative to see how the Rips complex and coverage complex diverge in the context of Chambers's proof.

Throughout this section, we fix the Rips radius $r = 1$.

2.3.1 The Rips Shadow

Given a set of points in the plane $\mathbb{X} = \{X_1, \dots, X_n\} \subset \mathbb{R}^2$ and a simplicial complex Σ whose 0-simplices are in 1-1 correspondence with the points of \mathbb{X} , there is a natural projection

$$p : \Sigma \longrightarrow \mathbb{R}^2 \tag{2.6}$$

defined by

$$\begin{aligned} [X_i] &\mapsto X_i \\ [X_{i_0} \cdots X_{i_k}] &\mapsto \text{conv}(X_{i_0}, \dots, X_{i_k}) \end{aligned}$$

Where $\text{conv}(X_1, \dots, X_k)$ denotes the convex hull of the specified points.

In the case where Σ is a Rips complex, the image of this map is called the *Rips shadow* of \mathbb{X} , or simply the *shadow*.

Definition 2.3. For any finite set of points in the plane $\mathbb{X} \subseteq \mathbb{R}^2$ with Rips complex $\mathcal{R}_r(\mathbb{X})$, the **Rips shadow**, $\mathcal{S}_r(\mathbb{X})$, is the image of the projection map:

$$\mathcal{S}_r(\mathbb{X}) := p(\mathcal{R}_r(\mathbb{X})) \tag{2.7}$$

Often the Rips radius and the set of points will be fixed and we will simply write $\mathcal{S} = p(\mathcal{R})$.

If $\mathcal{R}_r(\mathbb{X})$ is a Rips complex, then $\mathcal{S}_r(\mathbb{X})$ is a natural approximation to the union of balls $\bigcup_{i=1}^n B(X_i, r)$ for the following obvious reason.

Lemma 2.4. *For any simplex $\sigma = [X_0 \dots X_k] \in \mathcal{R}$, and any point $x \in p(\sigma) = \text{conv}(X_0, \dots, X_k)$, $|x - X_i| \leq r$ for each X_0, \dots, X_k .*

Corollary 2.5. *For any simplex $\sigma = [X_0 \dots X_k] \in \mathcal{R}$, $p(\sigma) \subseteq \bigcup_{i=1}^k B(X_i, r)$.*

Caratheodory's theorem says that if a point $x \in \mathbb{R}^d$ lies in the convex hull of some set of points in \mathbb{R}^d , then it lies in the convex hull of some subset of $d + 1$ of them. Therefore in our case, for the purpose of investigating π_1 we only need to consider the 2-skeleton of \mathcal{R} . Combining this with Corollary 2.5 and extending it

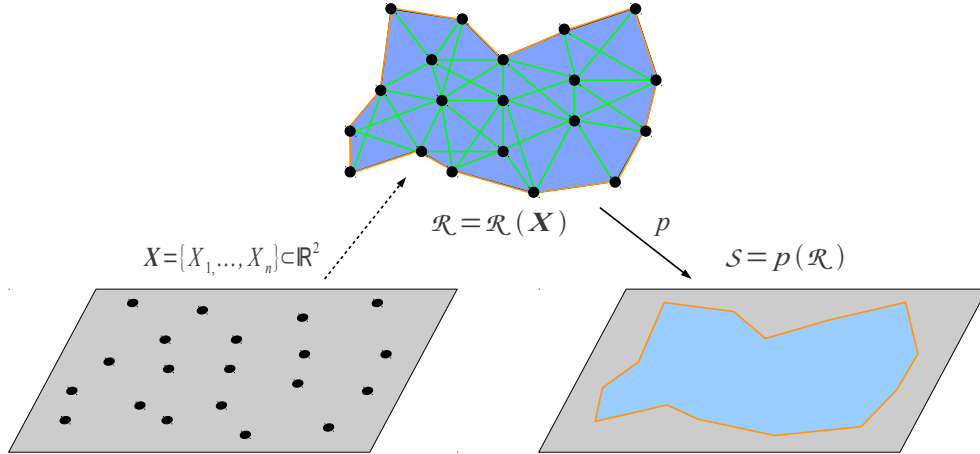


Figure 2.5: The Rips complex of points in the plane projects naturally back down to the plane. Each vertex maps to itself, and each simplex maps to the convex hull of its vertices. The image of this projection is called the Rips shadow, \mathcal{S} .

to all simplices in \mathcal{R} we get

$$\mathcal{S}_r(\mathbb{X}) = \bigcup_{\sigma \in \mathcal{R}_r(\mathbb{X})} p(\sigma) \tag{2.8}$$

$$= \bigcup_{k \in \mathbb{Z}^+} \bigcup_{[X_0 \dots X_k] \in \mathcal{R}} \text{conv}(X_0, \dots, X_k) \tag{2.9}$$

$$= \bigcup_{k=0}^2 \bigcup_{[X_0 \dots X_k] \in \mathcal{R}} \text{conv}(X_0, \dots, X_k) \tag{2.10}$$

Now we can state Chambers's main result.

Theorem 2.6. *For any finite set of distinct points in the plane $\mathbb{X} \subseteq \mathbb{R}^2$ and positive radius r , the projection map $p : \mathcal{R}_r(\mathbb{X}) \rightarrow \mathcal{S}_r(\mathbb{X})$ induces an isomorphism between fundamental groups.*

$$p_* : \pi_1(\mathcal{R}_r(\mathbb{X})) \xrightarrow{\cong} \pi_1(\mathcal{S}_r(\mathbb{X})) \tag{2.11}$$

2.3.2 Shadow Paths and Rips Paths

Chambers's proof relies on the concept of lifting paths from the shadow to the Rips complex. Explaining this requires some more definitions. Recall that for any simplicial complex Σ , $\Sigma^{(k)}$ denotes the k -skeleton of Σ .

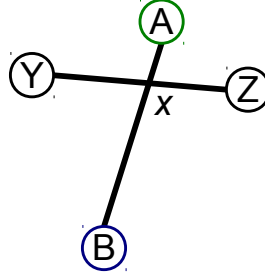
$$\Sigma^{(k)} = \{\sigma \in \Sigma \mid |\sigma| \leq k + 1\} \quad (2.12)$$

Definition 2.7. Given a finite set of points $\mathbb{X} \subseteq \mathbb{R}^2$ with Rips complex $\mathcal{R} = \mathcal{R}_r(\mathbb{X})$:

- A **Rips vertex** is any 0-simplex $[X] \in \mathcal{R}$.
- A **Rips edge** is any 1-simplex $[X Y] \in \mathcal{R}$
- A **shadow vertex** is the projected image of a Rips vertex or a transverse intersection of the projection of two Rips edges. The set of shadow vertices will be denoted $\mathcal{S}^{(0)}$.
- A **shadow edge** \overline{xy} is the closure of a connected component in $p(\mathcal{R}^{(1)}) \setminus \mathcal{S}^{(0)}$. The set of shadow edges will be denoted $\mathcal{S}^{(1)}$.

The idea of shadow vertex and shadow edge could use some more explanation. Initially one would expect to define a shadow vertex to be the projected image of a 0-simplex of \mathcal{R} , and a shadow edge to be projected image of a 1-simplex in \mathcal{R} . However that definition would be insufficient because the projection of two Rips edges can intersect. p is not a bijection and these points of intersection will be used to capture the effects of that.

For example, in the figure below $[A]$, $[B]$, $[Y]$, $[Z]$ are Rips vertices, and $[A B]$, $[Y Z]$ are Rips edges. Their projections intersect at the point x , so A, B, Y, Z, x are all shadow vertices. The shadow edges in this case are \overline{Ax} , \overline{xB} , \overline{Yx} , \overline{xZ} .



Generally, a **path** in \mathbb{R}^n is a continuous map $\gamma : [0, 1] \rightarrow \mathbb{R}^n$. A **cycle** (or loop) is a continuous map $\gamma : S^1 \rightarrow \mathbb{R}^n$. Any path, $\gamma : [0, 1] \rightarrow \mathcal{S}^{(1)}$ with $\gamma(0), \gamma(1) \in \mathcal{S}^{(0)}$ is homotopic rel its endpoints to a concatenation of paths along individual shadow edges; $\tilde{\gamma} = \gamma_0 * \dots * \gamma_m$ such that $\gamma_i([0, 1]) = \overline{t_i h_i}$, $\gamma_i(0) = t_i$, $\gamma_i(1) = h_i$ for some sequence $\overline{t_0 h_0}, \overline{t_1 h_1}, \dots, \overline{t_m h_m}$ of oriented shadow edges, such that the head of the i th edge is the tail of the $(i + 1)$ th edge; $h_i = t_{i+1}$. Similarly, any shadow cycle $\gamma : [0, 1] \rightarrow \mathcal{S}^{(1)}$ with $\gamma(0), \gamma(1) \in \mathcal{S}^{(0)}$ is homotopic rel its endpoints to a concatenation of paths along individual shadow edges; $\tilde{\gamma} = \gamma_0 * \dots * \gamma_m$ such that $\gamma_i([0, 1]) = \overline{t_i h_i}$, $\gamma_i(0) = t_i$, $\gamma_i(1) = h_i$ for some sequence $\overline{t_0 h_0}, \overline{t_1 h_1}, \dots, \overline{t_m h_m}$ of oriented shadow edges, such that the head of the i th edge is the tail of the $(i + 1)$ th edge; $h_i = t_{i+1}$ and $h_m = t_0$.

By **shadow path** and **shadow cycle** we will be referring to paths and cycles of this type. To simplify notation, shadow paths/cycles will be denoted by the sequence of shadow vertices they pass through. For example $\overline{t_0 h_0}, \overline{t_1 h_1}, \dots, \overline{t_m h_m}$ would be denoted $t_0 t_1 \dots t_m$. Paths and cycles can also be defined in the Rips complex. A

Rips path is an oriented sequence of 1-simplices $\Gamma = [A_0 B_0], \dots, [A_m B_m]$ such that $B_i = A_{i+1}$. A **Rips cycle** is a Rips path with $B_m = A_0$.

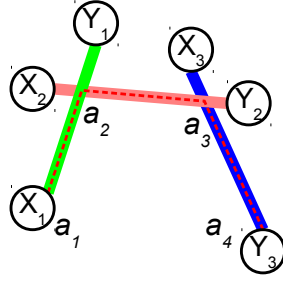
2.3.3 Lifting Shadow Paths

Lifting here refers to taking a shadow path, α , in \mathcal{S} and chaining together a sequence of Rips edges into a Rips path $\hat{\alpha}$ that covers α . It is not immediately obvious that such a lift will exist. In general it is not unique. The important property, in the case of Rips complexes, is that for any shadow path, lifting and projecting the lift back to \mathcal{S} is homotopic (in \mathcal{S}) rel the endpoints of the path to the original shadow path.

In general if we have a shadow path $\alpha = a_1 a_2 \dots a_n$, each edge, $a_i a_{i+1}$, lies in the image of some Rips edge $[X_i Y_i]$. Therefore, given α , we will always have a corresponding sequence of Rips edges $[X_1 Y_1], [X_2 Y_2], \dots, [X_{n-1} Y_{n-1}]$ where $a_i a_{i+1} \subseteq p([X_i Y_i])$. However this sequence is not necessarily a Rips path because successive segments may not share an endpoint.

The figure below shows an example. The shadow path $\alpha = a_1 a_2 a_3 a_4$ has three segments, but the Rips edges that cover them do not share any endpoints. In order for a true lift to a Rips path to exist there must be some additional Rips edges available in \mathcal{R} which we can insert into the sequence to *chain* these Rips edges together.

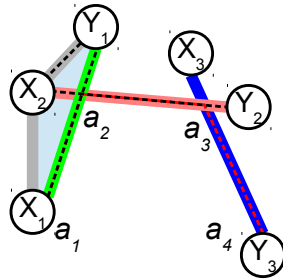
The following lemma follows from the triangle inequality, and essentially shows that lifts of shadow paths to Rips paths exist, and can be used to show that any



two lifts of a shadow path are homotopic, rel the endpoints, in \mathcal{R} .

Lemma 2.8. *Let $\{A, B, Y, Z\} \subseteq \mathbb{X} \subseteq \mathbb{R}^2$. Suppose $[A B], [Y Z] \in \mathcal{R}(\mathbb{X})$ and $AB \cap YZ = \{x\}$, and that Ax is the shortest of the four resulting shadow edges. Then $[A Y Z] \in \mathcal{R}$. In particular $\mathcal{R}_r(\{A, B, Y, Z\})$ is a cone over $[A]$.*

In the example above, for instance, considering adjacent shadow edges a_1a_2, a_2a_3 , the Rips complex on the four Rips vertices $\mathcal{R}(\{X_1, Y_1, X_2, Y_2\})$ is a cone over one of those Rips vertices. No matter which one, there is a sequence of Rips edges in $\mathcal{R}(\{X_1, Y_1, X_2, Y_2\})$ that can be inserted between $[X_1 Y_1]$ and $[X_2 Y_2]$ to give a true Rips path from $[X_1]$ to $[Y_2]$, and any such sequence is homotopic to all the possible ones.



The following lemma says that lifts to \mathcal{R} are well-behaved, and is key to proving theorem 2.6.

Lemma 2.9. *Given a shadow path α with endpoints in $p(\mathcal{R}^{(0)})$, for any lift $\hat{\alpha}$, $p(\hat{\alpha}) \simeq \alpha$ (in \mathcal{S}) rel the endpoints.*

The proof of theorem 2.6 centers on showing that p_* is injective, and is done by showing that the boundary of a shadow face must lift to a contractible (in \mathcal{R}) Rips cycle.

We will cite theorem 2.6 several times in this paper, and it will be illuminating to see where the coverage complexes we define will diverge from the nice behavior of the Rips complex. For example for the type of simplicial complexes we investigate lifts may not exist; or if they do they may not project back down to something homotopic to the original shadow path.

2.4 Extending to Network Code Coverage

In the case of sensor networks, the questions of coverage are entirely geometric. There is some fixed underlying domain \mathcal{D} which must be covered by sensors, and each node has two associated radii: r_b is the radio range, or the range within which two nodes can detect each others' presence, and $r_c < r_b$, the sensor coverage radius. One has to derive conclusions about the coverage of balls of radius r_c from connectivity information at radius r_b , or the Rips complex \mathcal{R}_{r_b} . In extending this approach to network coded information coverage we can simplify the problem by assuming the the coverage radius and communication radius are the same. In principle any server node can communicate just as well with any other server node as it can with any client node. A practical argument could be made for a case where server nodes

are more powerful, or mounted with higher visibility, and therefore their ability to communicate with each other is better than their ability to communicate with a mobile client, but the problem of network code coverage will introduce enough complications without adding the geometric issues of multiple communication radii. Besides that, the geometric constraint of having the coverage radius smaller than the radius used to construct the Rips complex has been thoroughly studied by de Silva and Ghrist.

Regarding network boundaries: in the case of network coded information coverage problems we would like to take the network/node arrangement for what it is. That is, given an arrangement of server nodes, we can never hope to distribute coded information in a way that covers an area larger than the limits of the network. Since our only window into the node geometry is the connectivity graph, the natural domain to aim for is the Rips shadow, $\mathcal{S} \subset \mathbb{R}^2$, discussed in the last section. Unfortunately in general it is impossible to identify the boundary $\partial\mathcal{S}$ based only on connectivity information. $\partial\mathcal{S}$ is piecewise linear, but the vertices of $\partial\mathcal{S}$ need not correspond to vertices of the Rips complex. To make the problem of network coded information coverage approachable we will make assumptions similar to de Silva and Ghrist in [21]: that we have prior knowledge of a subcomplex $\mathcal{F} \subseteq \mathcal{R}$ such that $p : \mathcal{F} \rightarrow \partial\mathcal{S}$ is a homeomorphism.

While the sensor network coverage work is geared toward identifying holes in coverage so nodes can be added or rearranged to patch those holes, we will be focused on distributing network coded data amongst the nodes so that any point in the coverage area of the network, whatever that may be, a mobile user can recover

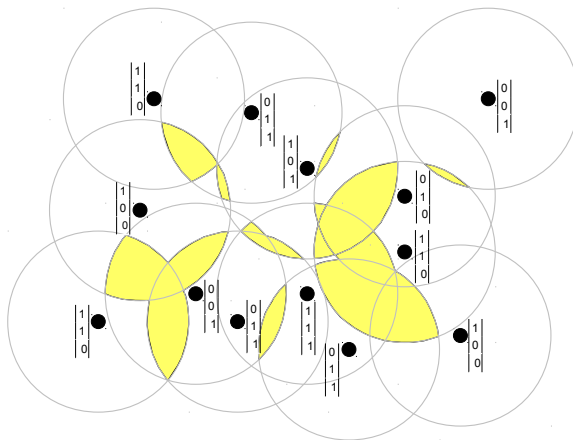


Figure 2.6: The network coded data coverage problem is more complicated. Each node has a certain inventory of vectors in a vector space. Only the intersection of a collection of balls whose inventories span the entire vector space are considered covered. Covered areas are the places where a user is within radius r of the data necessary to recover the original message.

the complete data bundle with some bounded amount of work. It is important to decide what our actual coverage goal will be; that is, what we mean by “coverage area of the network”, and how we will characterize successful coverage.

Figure 2.6 shows the same node arrangement as in figure 2.1 but with one vector distributed to each node. The yellow shaded areas are areas in range of nodes collectively holding a full basis for the full vector space $V = \text{GF}(2)^3$. In this figure we are confronted with one major difficulty of network code coverage problems. There is no clear analogue of the Čech complex for network code coverage. That is, there is no natural simplicial complex whose vertices correspond to the nodes which will capture the homotopy type of the covered areas. This is because, while intersections of convex shapes have a natural hereditary property (if three balls intersect, then each pair of two of them also intersects), information overlaps are just the opposite. If $X, Y, Z \in \mathbb{R}^2$, and the points inside $B_r(X) \cap B_r(Y) \cap B_r(Z)$ are within range of

a full basis for V , that is no guarantee that any of the two-fold intersections span anything. The best we will have are a variety of subcomplexes of the Rips complex whose shadows are contained in the covered areas.

On the other hand, there is a universal recipe for success in the network code coverage problem. If all nodes carry a full basis, then the entire network is covered. Everything else is an optimization on that extreme case, or a diagnosis of its failure.

Chapter 3: The Network Coded Coverage Problem

Here we will state a mathematically-abstracted version of the network coded coverage problem and begin building and studying topological structures based on it. First we define an abstract mathematical representation of a geographically distributed collection of network coded data.

Definition 3.1. Let \mathbb{F} be a finite field of characteristic q , and let $V = \mathbb{F}^D$ be a finite dimensional vector space over \mathbb{F} . Then a **finite data arrangement over V** is a collection of pairs $\mathcal{X} = \{(X_1, \mathcal{M}_{X_1}), \dots, (X_N, \mathcal{M}_{X_N})\}$ where X_i are points in a metric space, and $\mathcal{M}_{X_i} = \{\vec{v}_{i,1}, \dots, \vec{v}_{i,n_i}\}$ are collections of vectors in V . \mathcal{M}_{X_i} is called the **inventory of X_i** . If $\{X_1, \dots, X_N\} \subseteq \mathbb{R}^2$ then we will say that \mathcal{X} is a **finite planar data arrangement**.

Note that \mathcal{M}_{X_i} can be viewed as a vector matroid.

The goal in network coded data dissemination is to arrange the inventories of the points of \mathbb{X} so that every point in the range of the network is in range of a full basis for V . We will need to clarify what we mean by “in the range of the network” and “in range of a full basis for V ”. In order to simplify the exposition, we make the following notational definitions.

Definition 3.2. Given a collection of vectors $\mathcal{M} = \{\vec{v}_1, \dots, \vec{v}_k\}$, $\vec{v}_i \in V = \mathbb{F}^D$ we let $\langle \mathcal{M} \rangle = \langle \vec{v}_1, \dots, \vec{v}_k \rangle$ denote the linear span of the vectors in that collection.

$$\langle \mathcal{M} \rangle = \left\{ \vec{v} \in V \mid \exists a_i \in \mathbb{F} \text{ with } \vec{v} = \sum_{i=1}^k a_i \vec{v}_i \right\} \quad (3.1)$$

Given collections of vectors $\mathcal{M}_1, \dots, \mathcal{M}_k$, $\mathcal{M}_i = \{\vec{v}_{i,1}, \dots, \vec{v}_{i,l_i}\}$ in $V = \mathbb{F}^D$ we let $\langle \mathcal{M}_1, \dots, \mathcal{M}_k \rangle$ denote the collective span of the vectors in those sets.

$$\langle \mathcal{M}_1, \dots, \mathcal{M}_k \rangle = \langle \mathcal{M}_1 \cup \dots \cup \mathcal{M}_k \rangle \quad (3.2)$$

For convenience we extend this notation to define the linear span of points in \mathcal{X} and simplices on those points.

Definition 3.3. For a point $(X, \mathcal{M}_X) \in \mathcal{X}$ the **span of X** , denoted $\langle X \rangle$, is the subspace of V spanned by the vectors of \mathcal{M}_X

$$\langle X \rangle = \langle \mathcal{M}_X \rangle \quad (3.3)$$

For a simplex $\sigma = [X_0 \dots X_k]$ the span of the simplex is defined as the span of the union of the inventories of its vertices:

$$\langle \sigma \rangle = \langle \mathcal{M}_{X_0} \cup \dots \cup \mathcal{M}_{X_k} \rangle \quad (3.4)$$

Finally, given two or more simplices $\sigma_1, \dots, \sigma_n$ we define

$$\langle \sigma_1, \dots, \sigma_n \rangle = \langle \sigma_1^{(0)} \cup \dots \cup \sigma_n^{(0)} \rangle \quad (3.5)$$

3.1 The Coverage Complex $\mathcal{R}^C(U)$

Let $\mathcal{X} = \{(X_1, \mathcal{M}_{X_1}), \dots, (X_N, \mathcal{M}_{X_N})\}$ be a finite planar data arrangement.

For such a data arrangement we can define the Rips complex of radius r in the obvious way, based on the points in the arrangement.

$$\mathcal{R}_r(\mathcal{X}) = \mathcal{R}_r(\{X_1, \dots, X_N\}) \quad (3.6)$$

Definition 3.4. Let $V = \mathbb{F}^D$ and let \mathcal{X} be a finite planar data arrangement over V with Rips complex $\mathcal{R}_r(\mathcal{X})$. Let $U \subseteq V$ be a linear subspace. Then the **coverage complex for U** , $\mathcal{R}_r^C(\mathcal{X}, U) \subseteq \mathcal{R}_r(\mathcal{X})$, is the subcomplex of the Rips complex defined by

$$\sigma \in \mathcal{R}_r^C(\mathcal{X}, U) \Leftrightarrow \exists \tau_1, \dots, \tau_l \in \mathcal{R}_r(\mathcal{X}) \text{ such that}$$

$$\sigma \subseteq \tau_i \quad \forall i = 1 \dots l \text{ and } U \subseteq \langle \tau_1, \dots, \tau_l \rangle$$

In other words given a Rips simplex, $\sigma \in \mathcal{R}_r(\mathcal{X})$, we have $\sigma \in \mathcal{R}_r^C(\mathcal{X}, U)$ if σ is a face in the union of some collection of Rips simplices whose combined span contains U (under the convention that a simplex is a face of itself).

We could equivalently take the τ_i to be maximal in the definition above.

In much of this paper the data arrangement \mathcal{X} and the radius r will be fixed, so we will tend to leave them off the notation and simply write $\mathcal{R}^C(U)$ for the coverage complex for U and \mathcal{R} for the Rips complex.

The following fact is obvious, but worth noting.

Lemma 3.5. $\mathcal{R}^C(U)$ is a simplicial complex.

Proof. $\mathcal{R}^C(U)$ is a collection of simplices in \mathcal{R} . We just need to verify that it satisfies the hereditary property. That is, if a simplex σ is in $\mathcal{R}^C(U)$, then all of its faces are as well.

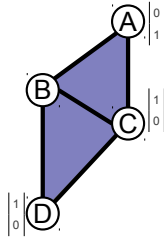
Suppose $\sigma \in \mathcal{R}^C(U)$. Then there are some (maximal) simplices τ_1, \dots, τ_n such that $\sigma \subseteq \tau_i$ for $i = 1, \dots, n$ and $U \subseteq \langle \tau_1, \dots, \tau_n \rangle$. Let σ' be a face of σ . Then $\sigma' \subseteq \tau_i$ for $i = 1, \dots, n$, as well, and therefore $\sigma' \in \mathcal{R}^C(U)$ □

We also note the following.

Lemma 3.6. $\mathcal{R}^C(0) = \mathcal{R}$.

3.1.1 Simple Examples of \mathcal{R}^C

Consider an arrangement of four points, $A, B, C, D \in \mathbb{R}^2$, with the Rips complex \mathcal{R} and vector inventories from $V = \text{GF}(2)^2$ pictured below.

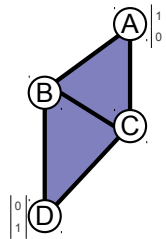


Then $V = \text{GF}(2)^2$ has three non-zero vectors and $3 + 3 + 1$ non-zero linear subspaces (including V itself). The coverage complexes for the dimension 1 subspaces are:

$\mathcal{R}^C(\begin{pmatrix} 1 \\ 0 \end{pmatrix})$	$\mathcal{R}^C(\begin{pmatrix} 0 \\ 1 \end{pmatrix})$	$\mathcal{R}^C(\begin{pmatrix} 1 \\ 1 \end{pmatrix})$

In the example above there are two maximal simplices, $[A B C]$ and $[B C D]$. Since the combined inventories of A and C span all of V , $\langle [A B C] \rangle = V$, and $[A B C]$ is in all the coverage complexes. On the other hand $\langle [B C D] \rangle = \langle \begin{pmatrix} 1 \\ 0 \end{pmatrix} \rangle$, so $[B C D]$ is only in $\mathcal{R}^C(\begin{pmatrix} 1 \\ 0 \end{pmatrix})$.

Next consider the same four points, with the vector inventories pictured below.



The coverage complexes for the 1D subspaces are shown below. Obviously $\langle [A B C] \rangle = \langle \begin{pmatrix} 1 \\ 0 \end{pmatrix} \rangle$ and $\langle [B C D] \rangle = \langle \begin{pmatrix} 0 \\ 1 \end{pmatrix} \rangle$. However the 1-simplex $[B C]$ is a face of both maximal simplices, which together span all of V . Therefore, by the definition of the coverage complex, $[B C] \subseteq [A B C]$ and $[B C] \subseteq [B C D]$, and $\begin{pmatrix} 1 \\ 1 \end{pmatrix} \in \langle [A B C], [B C D] \rangle = V$, so $[B C] \in \mathcal{R}^C(\begin{pmatrix} 1 \\ 1 \end{pmatrix})$.

$\mathcal{R}^C\left(\begin{pmatrix} 1 \\ 0 \end{pmatrix}\right)$	$\mathcal{R}^C\left(\begin{pmatrix} 0 \\ 1 \end{pmatrix}\right)$	$\mathcal{R}^C\left(\begin{pmatrix} 1 \\ 1 \end{pmatrix}\right)$

The property illustrated here, that the span of a simplex is the combined span of all the maximal simplices that it is a face of, leads us to sometimes call the coverage complex the **multi-sponsored coverage complex**. This is in contrast to the single-sponsored coverage complex we will investigate later. The multi-sponsored coverage complex makes sense geometrically, and is reasonable to compute, but it is more difficult to deal with in some proofs.

3.1.2 Basic Properties of \mathcal{R}^C

A lattice is a partially ordered set, (\mathcal{L}, \leq) , with binary operations \wedge (**meet**) and \vee (**join**). A lattice is bounded if it has a unique minimal element $\hat{0}$ which is less than everything else in the lattice, and a unique maximal element which is greater than everything else in the lattice.

The collection of subcomplexes of \mathcal{R} forms a bounded lattice, $(\mathcal{L}_{\mathcal{R}}, \subseteq)$, ordered by inclusion with:

- minimal element \emptyset
- maximal element \mathcal{R}
- meet operation is set intersection $\Sigma_1 \wedge \Sigma_2 = \Sigma_1 \cap \Sigma_2$

- join operation is set union $\Sigma_1 \vee \Sigma_2 = \Sigma_1 \cup \Sigma_2$

The collection of linear subspaces of V also forms a lattice, $(\mathcal{L}_V, \subseteq)$, ordered by inclusion, with:

- minimal element $\{\vec{0}\}$
- maximal element V
- meet operation is set intersection $W_1 \wedge W_2 = W_1 \cap W_2$
- join operation is the span $W_1 \vee W_2 = \langle W_1, W_2 \rangle$

Lemma 3.7. *The coverage complex operation:*

$$\mathcal{R}^C(\cdot) : (\mathcal{L}_V, \subseteq) \longrightarrow (\mathcal{L}_\mathcal{R}, \subseteq) \quad (3.7)$$

reverses the poset order. That is:

$$U \subseteq W \subseteq V \Rightarrow \mathcal{R}^C(W) \subseteq \mathcal{R}^C(U) \quad (3.8)$$

Proof. Suppose $U \subseteq W \subseteq V$, and $\sigma \in \mathcal{R}^C(W)$. Then there is a $\tau_1, \dots, \tau_l \in \mathcal{R}$ such that $\sigma \subseteq \tau_i \forall i$, and $W \subseteq \langle \tau_1, \dots, \tau_l \rangle$.

But $U \subseteq W \subseteq \langle \tau_1, \dots, \tau_l \rangle$, so $\sigma \in \mathcal{R}^C(U)$. □

It is natural to wonder if there is a complete correspondence between the two lattices. Do joins in \mathcal{L}_V correspond to meets in $\mathcal{L}_\mathcal{R}$, and vice-versa? It turns out only one of these is true in general.

Lemma 3.8. $\mathcal{R}^C(\langle U, W \rangle) = \mathcal{R}^C(U) \cap \mathcal{R}^C(W)$

Proof. $U \subseteq \langle U, W \rangle$ and $W \subseteq \langle U, W \rangle$, so by the last lemma $\mathcal{R}^C(\langle U, W \rangle) \subseteq \mathcal{R}^C(U)$ and $\mathcal{R}^C(\langle U, W \rangle) \subseteq \mathcal{R}^C(W)$. Therefore $\mathcal{R}^C(\langle U, W \rangle) \subseteq \mathcal{R}^C(U) \cap \mathcal{R}^C(W)$

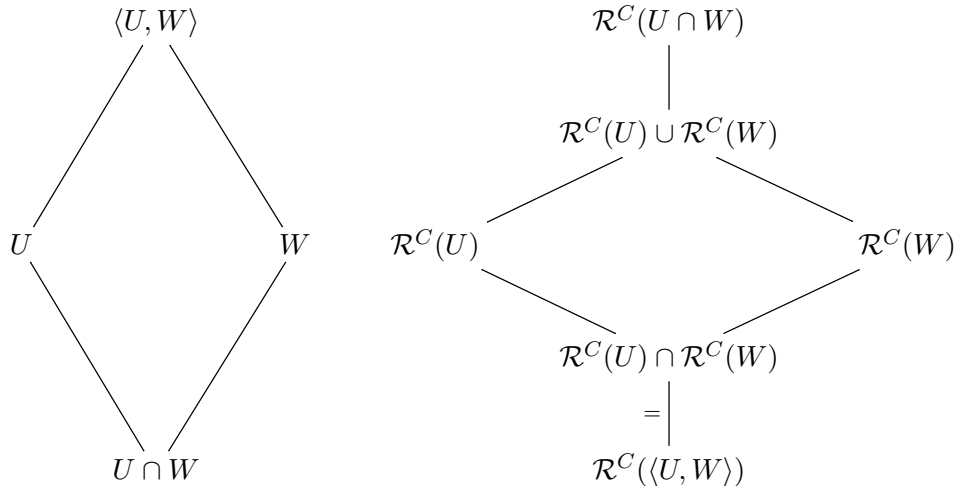
Take $\sigma \in \mathcal{R}^C(U) \cap \mathcal{R}^C(W)$. Then there exist $\tau_1, \dots, \tau_l \in \mathcal{R}$ such that $\sigma \subseteq \tau_i$ for each τ_i , and $U \subseteq \langle \tau_1, \dots, \tau_l \rangle$.

Similarly there exist $\tau'_1, \dots, \tau'_k \in \mathcal{R}$ such that $\sigma \subseteq \tau'_i$ for each τ'_i , and $W \subseteq \langle \tau'_1, \dots, \tau'_k \rangle$. Then $\langle U, W \rangle \subseteq \langle \tau_1, \dots, \tau_l, \tau'_1, \dots, \tau'_k \rangle$. Therefore $\sigma \in \mathcal{R}^C(\langle U, W \rangle)$, and generally $\mathcal{R}^C(\langle U, W \rangle) \supseteq \mathcal{R}^C(U) \cap \mathcal{R}^C(W)$ \square

Corollary 3.9. *If $\{\vec{v}_1, \vec{v}_2, \dots, \vec{v}_d\}$ is any set of vectors spanning V , then*

$$\mathcal{R}^C(V) = \bigcap_{i=1}^d \mathcal{R}^C(\langle \vec{v}_i \rangle)$$

The following Hasse diagrams summarize the poset reversal property of the coverage complex.



3.1.3 Basic Data Coverage Results For \mathcal{R}^C

The coverage complex is a geometrically sensible object to use to study network code coverage because, as a subcomplex of the Rips complex, lemma 2.4 holds: for any $\sigma \in \mathcal{R}$, every point $x \in p(\sigma)$ is within distance r of each of the vertices of σ . Therefore if $\langle \sigma \rangle = U \subseteq V$ it is clear that any point $x \in p(\sigma) = \text{conv}(X_0, \dots, X_k)$ is within distance r of a set of points whose collective inventories span U . We could say that every point in $p(\sigma)$ is *covered by* U . This is the notion of coverage we want to capture.

Definition 3.10. Let $\mathcal{X} = \{(X_1, \mathcal{M}_{X_1}), \dots, (X_N, \mathcal{M}_{X_N})\}$ be a finite planar data arrangement over $V = \mathbb{F}^D$, and let $r > 0$ be some radius. Then for any point $x \in \mathbb{R}^2$ define the **ambient vector matroid of radius r at x** , $\mathcal{M}_r(x)$, to be the collection of vectors in the inventories of all points of \mathcal{X} within radius r of x .

$$\mathcal{M}_r(x) = \bigcup_{|x-X_i| \leq r} \mathcal{M}_{X_i} \quad (3.9)$$

For any linear subspace $U \subseteq V$ and domain $\mathcal{D} \subseteq \mathbb{R}^2$ we say that \mathcal{D} is **covered by** U if $\forall x \in \mathcal{D}, U \subseteq \langle \mathcal{M}_r(x) \rangle$.

Lemma 3.11. *If $\sigma = [X_0 \dots X_k] \in \mathcal{R}^C(U)$ then $p(\sigma) = \text{conv}(X_0, \dots, X_k)$ is covered by U .*

Proof. Let $\tau_1, \dots, \tau_n \in \mathcal{R}$ be all maximal cofaces of σ in \mathcal{R} , and let $\mathcal{Y} = \{Y_1, \dots, Y_n\}$

be the collection of all their vertices. Since $\sigma \subseteq \tau_1, \dots, \tau_n \in \mathcal{R}$ every point $x \in \text{conv}(X_0, \dots, X_k)$ is within distance r of each vertex in \mathcal{Y} . Therefore $\mathcal{M}_{Y_1}, \dots, \mathcal{M}_{Y_n} \subseteq \mathcal{M}_r(x)$. Then, since $\sigma \in \mathcal{R}^C(U)$, by the definition of $\mathcal{R}^C(U)$ we know

$$\begin{aligned} U &\subseteq \langle \tau_1, \dots, \tau_n \rangle \\ &= \langle \mathcal{M}_{Y_1}, \dots, \mathcal{M}_{Y_n} \rangle \\ &\subseteq \langle \mathcal{M}_r(x) \rangle \end{aligned}$$

Therefore $p(\sigma)$ is covered by U . □

We would also like to define an analogue of the Rips shadow for coverage complexes.

Definition 3.12. Let $U \subseteq V$ be a linear subspace of V , and let $\mathcal{R}_r^C(\mathcal{X}, U)$ be a coverage complex. The **coverage shadow** $\mathcal{S}_r^C(\mathcal{X}, U)$ for radius r is the image of $\mathcal{R}_r^C(\mathcal{X}, U)$ under the Rips shadow projection map.

$$\mathcal{S}_r^C(\mathcal{X}, U) = p(\mathcal{R}_r^C(\mathcal{X}, U)) \subseteq \mathbb{R}^2 \tag{3.10}$$

Just as with the coverage complex, in contexts where the radius r and the data arrangement \mathcal{X} are clear, we will omit them from the notation and simply write $\mathcal{S}^C(U)$.

Then lemma 3.11 implies that the coverage shadow is covered.

Corollary 3.13. *For a given data arrangement \mathcal{X} , radius $r > 0$, linear subspace $U \subseteq V$, $\mathcal{S}^C(U)$ is covered by U .*

So \mathcal{R}^C gives us a sufficient (but not necessary) criterion for ensuring that the convex hull of a set of points is covered by a certain vector space. Furthermore we will see that if there are no “holes” in the coverage complex for U , then we can conclude that there are no “holes” in the region with access to a basis for U .

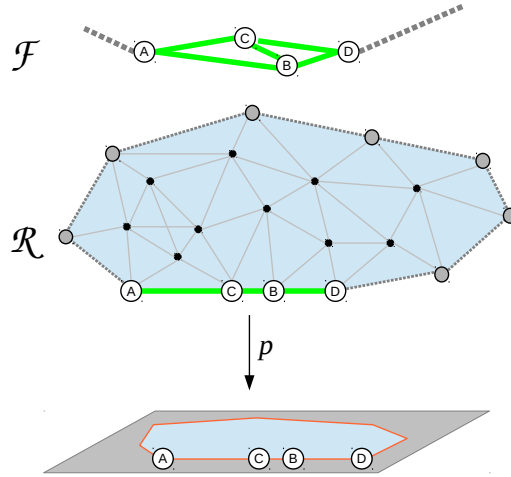
The specific properties of the boundary of the coverage domain and its correspondence with the Rips and coverage complexes is a key detail. It is worth defining exactly what we mean when we talk about a “fence subcomplex”.

Definition 3.14. Given a finite set of points $\mathbb{X} \subseteq \mathbb{R}^2$ with Rips complex $\mathcal{R} = \mathcal{R}(\mathbb{X})$ and a domain $\mathcal{D} \subseteq \mathbb{R}^2$, we say that a subcomplex $\mathcal{F} \subseteq \mathcal{R}$ is a **fence subcomplex** for \mathcal{D} if

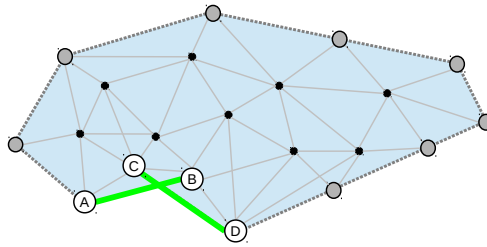
- $p : \mathcal{F} \xrightarrow{\cong} \partial\mathcal{D}$ is a homeomorphism.
- $\mathcal{F} = \mathcal{F}^+ \sqcup \mathcal{F}^-$ where \mathcal{F}^+ projects to $\partial^+\mathcal{D}$, the outer boundary of \mathcal{D} , and \mathcal{F}^- projects to $\partial^-\mathcal{D}$, the inner boundary components of \mathcal{D} .

Note that having a fence subcomplex implies that $\partial\mathcal{D}$ is piecewise linear and that all the 0-simplices of \mathcal{F} are on the boundary $\partial\mathcal{D}$. The reason we require that $p : \mathcal{F} \xrightarrow{\cong} \partial\mathcal{D}$ be a homeomorphism, rather than $p(\mathcal{F}) = \partial\mathcal{D}$, is a technicality, to avoid cases where the projection map on the fence subcomplex would not be 1-to-1. For example for the points shown below, we could have several closely-spaced colinear points on the boundary of \mathcal{D} that lead to several 1-simplices in the Rips

complex whose images under p overlap. This would make it possible to choose a subcomplex $\mathcal{F} \subseteq \mathcal{R}$ such that $p(\mathcal{F}) = \partial\mathcal{D}$ but with extra cycles that are degenerate upon projection to \mathbb{R}^2 .

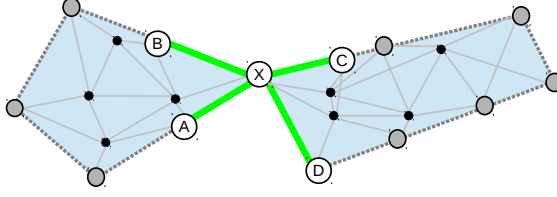


Another possible boundary complication that is prohibited under this definition is the case where the projection of two 1-simplices of \mathcal{F} cross transversally in their interiors, as shown below. In such a case $p(\mathcal{F}) \neq \partial\mathcal{S}$.



A final complication that *is* allowed under this definition, but which can often be discounted, is the case where the outer boundary $\partial^+\mathcal{S}$ does not have the homotopy type of a circle, as in the figure below. This is possible when the outer boundary intersects itself at a Rips vertex.

In this case the outer boundary is homeomorphic to a wedge of circles. In



most coverage criteria we have to either explicitly assume that this does not occur, or deal with the individual regions enclosed by the sub-cycles of $\partial^+ \mathcal{S}$ separately.

The following theorem is a slight generalization of that proved by de Silva and Ghrist in [21]. Essentially every simplex in the Rips complex has the geometric property of lemma 3.11, so therefore the theorem applies to any subcomplex of the Rips complex.

Theorem 3.15. *Let $\mathcal{R} = \mathcal{R}_r(\mathbb{X})$ be a Rips complex for a finite set of points $\mathbb{X} \subseteq \mathbb{R}^2$, and let $\mathcal{S} = p(\mathcal{R})$ be its Rips shadow. Let $\Sigma \subseteq \mathcal{R}$ be any subcomplex such that*

- *There is a fence subcomplex for \mathcal{S} , and $\mathcal{F} \subseteq \Sigma \subseteq \mathcal{R}$*
- *$\partial \mathcal{S} \simeq S^1$*

If there exists a 2-chain $\alpha \in C_2(\Sigma)$ defining a non-zero homology class $0 \neq [\alpha] \in H_2(\Sigma, \mathcal{F})$ such that $0 \neq \partial \alpha \in C_1(\Sigma)$, then $p(\Sigma) = \mathcal{S}$.

Note that for a 2-chain $\alpha \in C_2(\Sigma)$ to define a homology class $[\alpha] \in H_2(\Sigma, \mathcal{F})$ it must be a relative 2-cycle. That is, $\partial \alpha = 0$ in $C_1(\Sigma, \mathcal{F})$, or in other words $\partial \alpha \in C_1(\mathcal{F})$. This would include the possibility that $\partial \alpha = 0$ in $C_1(\Sigma)$. The requirement that $0 \neq \partial \alpha \in C_1(\Sigma)$ explicitly rules that out.

Proof. The proof here works exactly the same as that outlined in section 2.2.

Since $\Sigma \subseteq \mathcal{R}$, it is always the case that $p(\Sigma) \subseteq \mathcal{S}$. We will be proving the reverse inclusion, that the Rips shadow is contained in $p(\Sigma)$.

Just as before the projection map $p : \Sigma \rightarrow \mathbb{R}^2$ induces maps between the long exact sequences for the pairs (Σ, \mathcal{F}) and $(\mathbb{R}^2, \partial\mathcal{S})$, giving the commutative square:

$$\begin{array}{ccccc} H_2(\Sigma) & \xrightarrow{j_*} & H_2(\Sigma, \mathcal{F}) & \xrightarrow{\delta_*} & H_1(\mathcal{F}) \\ & & \downarrow p_* & & \downarrow p_* \cong \\ & & H_2(\mathbb{R}^2, \partial\mathcal{S}) & \xrightarrow{\delta_*} & H_1(\partial\mathcal{S}) \end{array} \quad (3.11)$$

We assumed $\partial\alpha \neq 0$; in other words α is not a cycle in $C_2(\Sigma)$, and $[\alpha] \notin H_2(\Sigma)$. Therefore $[\alpha] \notin \text{im}(j_*)$ in the diagram above, and by exactness of the top row $\delta_*[\alpha] \neq 0$.

Since we assumed that \mathcal{F} is a fence subcomplex (that $p : \mathcal{F} \rightarrow \partial\mathcal{D}$ is a homeomorphism), the right side of this box is an isomorphism, so we have $p_*\delta_*[\alpha] = p_*[\partial\alpha] \neq 0$. By commutativity $\delta_*p_*[\alpha] \neq 0$, in particular $p_*[\alpha] \neq 0$ in $H_2(\mathbb{R}^2, \partial\mathcal{S})$.

Suppose there is some point in \mathcal{S} not covered by U , $x \in \mathcal{S} - p(\Sigma)$. Then $p : (\Sigma, \mathcal{F}) \rightarrow (\mathbb{R}^2 - x, \partial\mathcal{S})$ and we can factor the left side of the box:

$$\begin{array}{ccccc} & & H_2(\Sigma, \mathcal{F}) & \xrightarrow{\delta_*} & H_1(\mathcal{F}) \\ & \swarrow p_* & \downarrow p_* & & \downarrow p_* \cong \\ H_2(\mathbb{R}^2 - x, \partial\mathcal{S}) & & H_2(\mathbb{R}^2, \partial\mathcal{S}) & \xrightarrow{\delta_*} & H_1(\partial\mathcal{S}) \\ & \searrow i_* & & & \end{array} \quad (3.12)$$

But $H_2(\mathbb{R}^2 - x, \partial\mathcal{S}) = 0$, so $p_*[\alpha] = 0$, a contradiction. So there can be no point $x \in \mathcal{S}$ not covered by $p(\Sigma)$, hence $\mathcal{S} \subseteq p(\Sigma)$. \square

Corollary 3.16. *Let \mathcal{X} be a finite planar data arrangement over $V = \mathbb{F}^D$, and let $U \subseteq V$ be a linear subspace. Let $r > 0$ be some finite radius, and let $\mathcal{R} = \mathcal{R}_r(\mathcal{X})$ be the corresponding Rips complex and $\mathcal{S} = p(\mathcal{R})$ the Rips shadow. Let $\mathcal{R}^C(U)$ be the corresponding coverage complex for vector space U , and $\mathcal{S}^C(U) = p(\mathcal{R}^C(U))$ the coverage shadow. Suppose:*

- *There is a fence subcomplex for \mathcal{S} , and $\mathcal{F} \subseteq \mathcal{R}^C(U) \subseteq \mathcal{R}$*
- *$\partial\mathcal{S} \simeq S^1$*

If there exists a 2-chain $\alpha \in C_2(\mathcal{R}^C(U))$ defining a non-zero homology class $0 \neq [\alpha] \in H_2(\mathcal{R}^C(U), \mathcal{F})$ such that $0 \neq \partial\alpha \in C_1(\mathcal{R}^C(U))$, then $\mathcal{S}^C(U) = \mathcal{S}$.

Though not immediately amenable to applying with theorem 2.2, it is worth translating this to give the obvious coverage criterion based on first homology.

Corollary 3.17. *Let $\mathcal{R} = \mathcal{R}_r(\mathbb{X})$ be a Rips complex for a finite set of points $\mathbb{X} \subseteq \mathbb{R}^2$, and let $\mathcal{S} = p(\mathcal{R})$ be its Rips shadow. Let $\Sigma \subseteq \mathcal{R}$ be any subcomplex such that*

- *There is a fence subcomplex for \mathcal{S} , and $\mathcal{F} \subseteq \Sigma \subseteq \mathcal{R}$*
- *$\partial\mathcal{S} \simeq S^1$*

If $\exists [\gamma] \neq 0$ in $H_1(\mathcal{F})$ such that the image $i_[\gamma] = 0$ in $H_1(\Sigma)$, then $p(\Sigma) = \mathcal{S}$.*

Proof. \mathcal{F} has no 2-simplices so $C_2(\mathcal{F}) = 0$, and $H_1(\mathcal{F}) = Z_1(\mathcal{F})$. Therefore $[\gamma] \neq 0$ in $H_1(\mathcal{F})$.

Consider the relevant section of the long exact sequence for (Σ, \mathcal{F}) .

$$\cdots \rightarrow H_2(\Sigma, \mathcal{F}) \xrightarrow{\delta_*} H_1(\mathcal{F}) \xrightarrow{i_*} H_1(\Sigma) \rightarrow \cdots$$

Our assumption is that $i_*[\gamma] = 0$ in $H_1(\Sigma)$. By exactness this means that there is some $[\alpha] \in H_2(\Sigma, \mathcal{F})$ such that $\delta_*[\alpha] = [\gamma] \neq 0$. Therefore $\partial\alpha \neq 0$ and by theorem 3.15, $p(\Sigma) = \mathcal{S}$. \square

Applying this to data arrangements we conclude:

Corollary 3.18. *Under the hypotheses of corollary 3.16, if $\exists [\gamma] \neq 0$ in $H_1(\mathcal{F})$ such that $i_*[\gamma] = 0$ in $H_1(\mathcal{R}^C(U))$, then $\mathcal{S}^C(U) = \mathcal{S}$.*

These results can be generalized to Rips complexes with (known) inner boundaries and to general cycles in the Rips complex, just as with de Silva and Ghrist's criterion. This means loosening the hypothesis that $\partial\mathcal{S} \simeq S^1$. The following theorem is a slight generalization of another result of de Silva and Ghrist in [21]. The proof uses the long exact sequence of a triple. Recall that if we have a triple of topological spaces (X, Y, Z) such that $Z \subseteq Y \subseteq X$, then there is an exact sequence of chain groups

$$0 \rightarrow C_\bullet(Y, Z) \xrightarrow{i} C_\bullet(X, Z) \xrightarrow{j} C_\bullet(X, Y) \rightarrow 0 \quad (3.13)$$

which gives a long exact sequence on homology.

$$\cdots \rightarrow H_k(Y, Z) \xrightarrow{i_*} H_k(X, Z) \xrightarrow{j_*} H_k(X, Y) \xrightarrow{\delta_*} H_{k-1}(Y, Z) \rightarrow \cdots \quad (3.14)$$

Theorem 3.19. Let $\mathcal{R} = \mathcal{R}_r(\mathbb{X})$ be a Rips complex for a finite set of points $\mathbb{X} \subseteq \mathbb{R}^2$, and let $\mathcal{S} = p(\mathcal{R})$ be its Rips shadow. Let $\Sigma \subseteq \mathcal{R}$ be any subcomplex such that

- There is a fence subcomplex for \mathcal{S} , $\mathcal{F} = \mathcal{F}^+ \sqcup \mathcal{F}^-$, and $\mathcal{F} \subseteq \Sigma \subseteq \mathcal{R}$
- $\partial^+ \mathcal{S} \simeq S^1$

If there exists a 2-chain $\alpha \in C_2(\Sigma)$ defining a non-zero homology class $0 \neq [\alpha] \in H_2(\Sigma, \mathcal{F})$ such that $0 \neq \partial\alpha \in H_1(\mathcal{F}^+)$, then $p(\Sigma) = \mathcal{S}$.

Proof. The projection p maps between the triples $(\Sigma, \mathcal{F}, \mathcal{F}^-) \xrightarrow{p_*} (\mathbb{R}^2, \partial\mathcal{S}, \partial^- \mathcal{S})$, inducing maps between the long exact sequences

$$\begin{array}{ccccccc} \cdots & \longrightarrow & H_2(\mathcal{F}, \mathcal{F}^-) & \xrightarrow{i_*} & H_2(\Sigma, \mathcal{F}^-) & \xrightarrow{j_*} & H_2(\Sigma, \mathcal{F}) & \xrightarrow{\delta_*} & H_1(\mathcal{F}, \mathcal{F}^-) & \longrightarrow & \cdots \\ & & \downarrow p_* & & \downarrow p_* & & \downarrow p_* & & \downarrow p_*, \cong & & \\ \cdots & \longrightarrow & H_2(\partial\mathcal{S}, \partial^- \mathcal{S}) & \xrightarrow{i_*} & H_2(\mathbb{R}^2, \partial^- \mathcal{S}) & \xrightarrow{j_*} & H_2(\mathbb{R}^2, \partial\mathcal{S}) & \xrightarrow{\delta_*} & H_1(\partial\mathcal{S}, \partial^- \mathcal{S}) & \longrightarrow & \cdots \end{array}$$

Suppose we have $[\alpha] \in H_2(\Sigma, \mathcal{F})$ such that $\partial\alpha$ is non-zero in $C_1(\mathcal{F}^+)$. Tracing through the snake lemma for the upper long exact sequence, one can see that this is equivalent to $\delta_*[\alpha] \neq 0$ in $H_1(\mathcal{F}, \mathcal{F}^-)$. By our assumption that \mathcal{F} is a fence subcomplex, $p_* : H_1(\mathcal{F}, \mathcal{F}^-) \rightarrow H_1(\partial\mathcal{S}, \partial^- \mathcal{S})$ is an isomorphism, and we have $p_*\delta_*[\alpha] \neq 0$. By commutativity $\delta_*p_*[\alpha] \neq 0$.

Suppose that $p(\Sigma) \neq \mathcal{S}$. Then there must be some $x \in \mathcal{S} \setminus p(\Sigma)$, and we can

factor one of these projections through $H_2(\mathbb{R}^2 - x, \partial\mathcal{S})$:

$$\begin{array}{ccc}
 & H_2(\Sigma, \mathcal{F}) & \xrightarrow{\delta_*} & H_1(\mathcal{F}, \mathcal{F}^-) & \\
 & \swarrow p_* & \downarrow p_* & \downarrow p_*, \cong & \\
 H_2(\mathbb{R}^2 - x, \partial\mathcal{S}) & & & & \\
 & \searrow i_* & \downarrow p_* & \downarrow p_*, \cong & \\
 & H_2(\mathbb{R}^2, \partial\mathcal{S}) & \xrightarrow{\delta_*} & H_1(\partial\mathcal{S}, \partial^-\mathcal{S}) &
 \end{array} \tag{3.15}$$

The key observation is that the composite map $\delta_* i_* : H_2(\mathbb{R}^2 - x, \partial\mathcal{S}) \rightarrow H_2(\partial\mathcal{S}, \partial^-\mathcal{S})$ is the zero map.

First consider the long exact sequence for the triple $(\mathbb{R}^2 - x, \partial\mathcal{S}, \partial^-\mathcal{S})$ and note that the composite $\delta_* i_*$ is the same as the connecting homomorphism $\delta'_* : H_2(\mathbb{R}^2 - x, \partial\mathcal{S}) \rightarrow H_1(\partial\mathcal{S}, \partial^-\mathcal{S})$ from dim 2 to dim 1 on this LES. In order to fully see this one can trace through the snake lemma for the triples $(\mathbb{R}^2, \partial\mathcal{S}, \partial^-\mathcal{S})$ and $(\mathbb{R}^2 - x, \partial\mathcal{S}, \partial^-\mathcal{S})$. The key points to notice are that the boundaries $\partial\mathcal{S}$ and $\partial^-\mathcal{S}$ can be ignored in $C_2(\cdot, \cdot)$, and that $C_1(\partial\mathcal{S}, \partial^-\mathcal{S}) \cong C_1(\partial^+\mathcal{S})$. Then both $\delta_* i_*$ and δ'_* take an element $[\alpha] \in H_2(\mathbb{R}^2 - x, \partial\mathcal{S})$ and map it to the homology class of the part of $\partial\alpha$ that lies in $C_1(\partial^+\mathcal{S})$.

Given this observation we look at the long exact sequence where δ' appears:

$$\cdots \rightarrow H_2(\mathbb{R}^2 - x, \partial\mathcal{S}) \xrightarrow{\delta'_*} H_1(\partial\mathcal{S}, \partial^-\mathcal{S}) \xrightarrow{j_*} H_1(\mathbb{R}^2 - x, \partial^-\mathcal{S}) \rightarrow \cdots \tag{3.16}$$

and see that it will be enough to show that $j_* : H_1(\mathbb{R}^2 - x, \partial^-\mathcal{S}) \rightarrow H_1(\partial\mathcal{S}, \partial^-\mathcal{S})$ is injective. Geometrically this is fairly obvious, since the outer boundary of the shadow, $\partial^+\mathcal{S}$, encloses x , but none of the inner boundary cycles in $\partial^-\mathcal{S}$ enclose x .

This shows that $\delta_* i_*$ is the zero map, contradicting the fact that $\delta_* p_*[\alpha] \neq 0$.

Thus there can be no such $x \in \mathcal{S} \setminus p(\Sigma)$. \square

This result can also be translated in terms of H_1 .

Corollary 3.20. *Let $\mathcal{R} = \mathcal{R}_r(\mathbb{X})$ be a Rips complex for a finite set of points $\mathbb{X} \subseteq \mathbb{R}^2$, and let $\mathcal{S} = p(\mathcal{R})$ be its Rips shadow. Let $\Sigma \subseteq \mathcal{R}$ be any subcomplex such that*

- *There is a fence subcomplex for \mathcal{S} , $\mathcal{F} = \mathcal{F}^+ \sqcup \mathcal{F}^-$, and $\mathcal{F} \subseteq \Sigma \subseteq \mathcal{R}$*
- *$\partial^+ \mathcal{S} \simeq S^1$*

If there exists $[\gamma] \neq 0$ in $H_1(\mathcal{F}^+)$ such that the image $[\gamma] = 0$ in $H_1(\Sigma, \mathcal{F}^-)$, then $p(\Sigma) = \mathcal{S}$.

Proof. Consider the long exact sequence of the triple $(\Sigma, \mathcal{F}, \mathcal{F}^-)$.

$$\cdots \rightarrow H_2(\Sigma, \mathcal{F}^-) \rightarrow H_2(\Sigma, \mathcal{F}) \xrightarrow{\delta_*} H_1(\mathcal{F}, \mathcal{F}^-) \xrightarrow{i_*} H_1(\Sigma, \mathcal{F}^-) \rightarrow \cdots$$

By excision $H_1(\mathcal{F}, \mathcal{F}^-) \cong H_1(\mathcal{F}^+)$. So our assumption that $0 \neq [\gamma] \in H_1(\Sigma, \mathcal{F}^-)$ means that $[\gamma] \neq 0$ in $H_1(\mathcal{F}, \mathcal{F}^-)$. The second part of our assumption is that $i_*[\gamma] = 0$. By exactness there is $[\alpha] \in H_2(\Sigma, \mathcal{F})$ such that $\delta_*[\alpha] = [\gamma] \neq 0$. By theorem 3.19 $p(\Sigma) = \mathcal{S}$. \square

So with practically no new work we have a coverage criterion for network coded data distribution. However the variety of cases where the hypotheses of these theorems fail despite having complete data coverage are much more extensive than those arising in the sensor network coverage world from the geometric defects of the

Rips complex. We will spend considerable work examining some of those *false holes*, and refining the hypotheses of this theorem to account for them.

3.2 Enumerating the Coverage Complexes

Given a finite planar data arrangement \mathcal{X} and radius $r > 0$, for any linear subspace $W \subseteq V$ there is a coverage complex $\mathcal{R}^C(W) \subseteq \mathcal{R}$. On the other hand, not every subcomplex $\Sigma \subseteq \mathcal{R}$ can be expressed as a coverage complex. Consider the simple case with a single 1-simplex: $\mathcal{R} = \{ [A], [B], [A B] \}$, and let $\Sigma = \{ [A], [B] \}$. If Σ were a coverage complex, it would necessarily also contain $[A B]$, since in this case $\langle [A] \rangle = \langle [B] \rangle = \langle [A B] \rangle = \langle \mathcal{M}_A, \mathcal{M}_B \rangle$. This is because A and B only belong to one maximal simplex in \mathcal{R} . However for simplices that are faces of multiple maximal simplices this becomes more complicated. Consideration of cases like this lead us to investigate a slightly restricted version of the coverage complex.

Definition 3.21. Let $V = \mathbb{F}^D$ and let \mathcal{X} be a finite planar data arrangement over V with Rips complex $\mathcal{R}_r(\mathcal{X})$. Let $U \subseteq V$ be a linear subspace. Then the **single-sponsored coverage complex** or **$C1$ -coverage complex** for U , $\mathcal{R}_r^{C1}(\mathcal{X}, U) \subseteq \mathcal{R}_r(\mathcal{X})$, is the subcomplex of the Rips complex defined by

$$\sigma \in \mathcal{R}_r^{C1}(\mathcal{X}, U) \Leftrightarrow \exists \tau \in \mathcal{R}_r(\mathcal{X}) \text{ such that } \sigma \subseteq \tau \text{ and } U \subseteq \langle \tau \rangle$$

Note that

$$\mathcal{R}_r^{C1}(\mathcal{X}, U) \subseteq \mathcal{R}_r^C(\mathcal{X}, U) \subseteq \mathcal{R}_r(\mathcal{X}) \quad (3.17)$$

We would like to characterize the subcomplexes of the Rips complex which can be obtained as a single-sponsored coverage complex for some vector space V , data arrangement \mathcal{X} over V , and subspace U . The following definitions are useful.

Definition 3.22. Given a set of points $\{X_1, \dots, X_N\} \subseteq \mathbb{R}^2$ with Rips complex \mathcal{R} , a subcomplex $\Sigma \subseteq \mathcal{R}$ is **resolvable** if there exists a vector space $V = \mathbb{F}^D$, a data arrangement \mathcal{X} , and a subspace $U \subseteq V$ such that $\Sigma = \mathcal{R}^C(\mathcal{X}, U)$.

We say $\Sigma \subseteq \mathcal{R}$ is **$C1$ -resolvable** if there exists a vector space $V = \mathbb{F}^D$, a data arrangement \mathcal{X} , and a subspace $U \subseteq V$ such that $\Sigma = \mathcal{R}^{C1}(\mathcal{X}, U)$.

Then $C1$ -resolvability is easily characterized. In fact, for a given Rips complex \mathcal{R} there is a sort of *universal* vector space and data arrangement that can yield every possible single-sponsored coverage complex by taking the correct 1-dimensional subspace of V .

The following definition is a technicality, but necessary to state the results correctly.

Definition 3.23. For a collection of simplices, Σ , the **closure** of Σ , $\bar{\Sigma}$ is the set of all simplices σ such that $\sigma \subseteq \tau \in \Sigma$. That is, $\bar{\Sigma}$ is the smallest possible simplicial complex containing Σ .

When $\Sigma = \{\tau\}$ contains a single simplex we will simply write $\bar{\tau}$ instead of $\overline{\{\tau\}}$.

Proposition 3.24. *Given a set of points $\{X_1, \dots, X_N\} \subseteq \mathbb{R}^2$ with Rips complex \mathcal{R} , a subcomplex $\Sigma \subseteq \mathcal{R}$ is **C1-resolvable** if and only if it is the closure of a union of maximal simplices of \mathcal{R} .*

Furthermore if \mathcal{R} has M maximal simplices, then there is a data arrangement \mathcal{X} over the vector space $V = \mathbb{F}^M$ such that every C1-resolvable subcomplex of \mathcal{R} can be obtained as $\mathcal{R}^{C1}(\mathcal{X}, \langle \vec{v} \rangle)$ for some $\vec{v} \in V$. The C1-resolvable subcomplexes of \mathcal{R} are in 1-1 correspondence with the 1-dimensional subspaces of $\text{GF}(2)^M$.

Proof. Let $\mathbb{X} = \{X_1, \dots, X_N\} \subseteq \mathbb{R}^2$ be the set of points and let \mathcal{R} be their Rips complex. Let τ_1, \dots, τ_M be the maximal simplices of \mathcal{R} .

First suppose that $\mathcal{X} = \{(X_1, \mathcal{M}_{X_1}), \dots, (X_N, \mathcal{M}_{X_N})\}$ is a data arrangement over some vector space V , and suppose $\Sigma = \mathcal{R}^{C1}(U)$ for some $U \subseteq V$. Suppose that Σ is not the closure of a union of \mathcal{R} -maximal simplices. That is, there exists $\sigma \in \Sigma$ such that σ is maximal in Σ , but not maximal in \mathcal{R} . So there is at least one $\tau \in \mathcal{R} \setminus \mathcal{R}^{C1}(U)$ such that $\sigma \subsetneq \tau$. But then by definition $\langle \sigma \rangle \subseteq \langle \tau \rangle$, and by the definition of $\mathcal{R}^{C1}(U)$, $U \subseteq \langle \sigma \rangle$, so $\tau \in \mathcal{R}^{C1}(U)$; a contradiction.

Therefore $\mathcal{R}^{C1}(U)$ must be a union of \mathcal{R} -maximal simplices.

We prove the other direction constructively. Let \mathbb{F} be a finite field, and let $V = \mathbb{F}^M$ and fix a basis $\mathcal{B} = \{\vec{v}_1, \dots, \vec{v}_M\}$.

For each point $X \in \mathbb{X}$ build an inventory:

$$\mathcal{M}_X = \{\vec{v}_j \in \mathcal{B} \mid X \notin \tau_j\}$$

Clearly this ensures that $\vec{v}_j \notin \langle \tau_j \rangle$.

On the other hand, suppose $\vec{v}_j \notin \langle \tau_k \rangle$ for some $\tau_k \neq \tau_j$. By the definition of $\langle \cdot \rangle$ for simplices and the specification of \mathcal{M}_X above this means that for each $X \in \tau_k$, $X \in \tau_j$, and therefore $\tau_k \subseteq \tau_j$; a contradiction since the τ_i are maximal. Therefore $\vec{v}_j \in \langle \tau_k \rangle \forall k \neq j$.

Thus, for any subset $S = \{i_1, \dots, i_n\} \subseteq \{1, \dots, M\}$

$$\mathcal{R}^{C1}(\langle \vec{v}_{i_1}, \dots, \vec{v}_{i_n} \rangle) = \bigcup_{j \in \{1, \dots, M\} \setminus S} \bar{\tau}_j$$

so each subcomplex $\Sigma \subseteq \mathcal{R}$ consisting of a union of maximal simplices of \mathcal{R} is resolvable. This includes $\mathcal{R}^{C1}(0) = \mathcal{R}$ and $\mathcal{R}^{C1}(V) = \emptyset$.

Furthermore, because \mathcal{B} is a basis, note that for any \vec{v}_i and corresponding τ_i , if $\vec{w} \in \langle \mathcal{B} \setminus \vec{v}_i \rangle$, then $\vec{w} \in \langle \tau_i \rangle$ but $\vec{w} + \vec{v}_i \notin \langle \tau_i \rangle$, and therefore $\tau_i \in \mathcal{R}^{C1}(\langle \vec{w} \rangle)$ but $\tau_i \notin \mathcal{R}^{C1}(\langle \vec{w} + \vec{v}_i \rangle)$.

Therefore for any $\Sigma = \bigcup_{j \in S} \bar{\tau}_j$ for some $S = \{i_1, \dots, i_n\} \subseteq \{1, \dots, M\}$, we can take a single vector $v_\Sigma = \sum_{j \notin S} \vec{v}_j$ and by our construction

$$\mathcal{R}^{C1}(\langle v_\Sigma \rangle) = \mathcal{R}^{C1}(\langle \sum_{j \notin S} \vec{v}_j \rangle) = \bigcup_{j \in S} \bar{\tau}_j = \Sigma$$

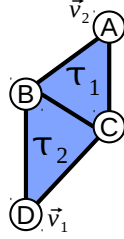
Finally if we take $\mathbb{F} = \mathbf{GF}(2)$ the choices for $S \subseteq \{1, \dots, M\}$ correspond exactly to 1-dimensional linear subspaces of $V = \mathbf{GF}(2)^M$, and therefore the $C1$ -resolvable subcomplexes of \mathcal{R} are in 1-1 correspondence with the 1-dimensional linear subspaces of V . □

Corollary 3.25. *For a Rips complex \mathcal{R} with maximal simplices τ_1, \dots, τ_M , there are exactly 2^M C1-resolvable subcomplexes.*

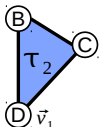
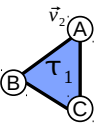
Note that corollary 3.25 includes the C1-resolvable simplex $\mathcal{R} = \mathcal{R}^{C1}(0)$ in the count.

Of course the number of maximal simplices in \mathcal{R} depends on the set of points $\mathbb{X} = \{X_1, \dots, X_N\}$. The collection of maximal simplices of \mathcal{R} is a Sperner family in the power set of \mathbb{X} ; it has the property that no two elements are subsets of each other. Therefore by Sperner's Theorem $M \leq \binom{N}{\lfloor N/2 \rfloor}$ [24]. However \mathcal{R} is not an arbitrary subset of the power set of \mathbb{X} . On a purely combinatorial level it is a flag complex, therefore it is entirely characterized by the cliques in its 1-skeleton. The problem of counting and producing all maximal cliques of a graph is well known in graph theory and computer science. A famous theorem of Moon and Moser [25] says that $M \leq 3^{N/3}$. The Rips complex also has geometric constraints on it. However the number of maximal simplices can still be exponential in extreme cases. The example that yields an arbitrary-dimensional cross-polytope in the Rips complex also happens to have $2^{N/2}$ maximal simplices [26]. So in that case there are $2^{2^{N/2}}$ possible single-sponsored coverage complexes.

The figure below shows an example of the universal data arrangement used in the proof of prop 3.24. The complex has two maximal simplices τ_1, τ_2 , and therefore requires a 2D vector space with two basis vectors $\mathcal{B} = \{\vec{v}_1, \vec{v}_2\}$. The vertices B and C are faces of both τ_1 and τ_2 , so their inventories are empty. A is a face of τ_1 so its inventory contains only \vec{v}_2 , and D is a face of τ_2 so its inventory contains only \vec{v}_1 .



Then $V = \text{GF}(2)^2$ has three non-zero vectors and three 1D linear subspaces. The single-sponsored coverage complexes for the 1D subspaces are as desired. Including \vec{v}_1 eliminates the simplex τ_1 , and including \vec{v}_2 eliminates τ_2 . The vector $\vec{v}_1 + \vec{v}_2$ is outside the span of either maximal simplex, and therefore eliminates the entire complex.

$\mathcal{R}^{C1}(\vec{v}_1)$	$\mathcal{R}^{C1}(\vec{v}_2)$	$\mathcal{R}^C(\vec{v}_1 + \vec{v}_2)$
		\emptyset

The situation for (multi-sponsored) coverage complexes $\mathcal{R}^C(U)$ is similar but more complicated. As we observed in the example in section 3.1.1 non-maximal simplices can be resolvable. It turns out that a condition for resolvability can be stated recursively; a simplex is resolvable if it is maximal or the intersection of two resolvable simplices. The construction is almost the same as that used to prove $C1$ -resolvability.

Proposition 3.26. *Given a set of points $\{X_1, \dots, X_N\} \subseteq \mathbb{R}^2$ with Rips complex \mathcal{R} , a simplex $\sigma \in \mathcal{R}$ is resolvable if and only if it is the intersection of maximal simplices in \mathcal{R} . A subcomplex $\Sigma \subseteq \mathcal{R}$ is resolvable if and only if it is the closure of a union of resolvable simplices.*

Furthermore if \mathcal{R} has M maximal simplices with an intersection poset of size P , then there is a data arrangement \mathcal{X} over $V = \mathbb{F}^P$ such that every resolvable proper subcomplex of \mathcal{R} can be obtained as $\mathcal{R}^C(\mathcal{X}, \langle \vec{v} \rangle)$ for some $\vec{v} \in V$.

Proof. Let $\mathbb{X} = \{X_1, \dots, X_N\} \subseteq \mathbb{R}^2$ be the set of points and $\mathcal{R} = \mathcal{R}(\mathbb{X})$.

Just as before, suppose that $\mathcal{X} = \{(X_1, \mathcal{M}_{X_1}), \dots, (X_N, \mathcal{M}_{X_N})\}$ is a data arrangement over some vector space V , and suppose $\Sigma = \mathcal{R}^C(U)$ for some $U \subseteq V$. Suppose that Σ is not a union of intersections of closures of \mathcal{R} -maximal simplices. That is, there exists $\sigma \in \Sigma$ such that σ is maximal in Σ , but not equal to the intersection of maximal simplices in \mathcal{R} . Then σ must be properly contained in at least one maximal simplex of \mathcal{R} . Let τ_1, \dots, τ_k be all the \mathcal{R} -maximal simplices containing σ , so $\sigma \subsetneq \sigma' = \tau_1 \cap \dots \cap \tau_k$. But then $U \subseteq \langle \sigma \rangle \subseteq \langle \tau_j \rangle$ for each of the τ_1, \dots, τ_k , so $U \subseteq \langle \sigma' \rangle$, and therefore $\sigma' \in \mathcal{R}^C(U)$ contradicting the maximality of σ . So all resolvable simplices must be the intersection of maximal simplices.

The other direction is similar as well except we take $V = \mathbb{F}^P$ and fix a basis $\mathcal{B} = \{\vec{v}_1, \dots, \vec{v}_P\}$. Let $\gamma_1, \dots, \gamma_P \in \mathcal{R}$ be the simplices that are intersections of maximal simplices of \mathcal{R} . This includes the maximal simplices themselves. For each point $X \in \mathbb{X}$ build an inventory that ensures that $\vec{v}_j \notin \langle \tau \rangle$ for any maximal τ containing γ_j :

$$\begin{aligned} \mathcal{M}_X &= \{\vec{v}_j \in \mathcal{B} \setminus \{\vec{v}_P\} \mid \{X\} \cup \gamma_j \notin \mathcal{R}\} \\ &= \{\vec{v}_j \in \mathcal{B} \setminus \{\vec{v}_P\} \mid \gamma_j \notin \text{ClSt } X\} \\ &= \{\vec{v}_j \in \mathcal{B} \setminus \{\vec{v}_P\} \mid \{X\} \notin \text{ClSt } \gamma_j\} \end{aligned}$$

and let $\mathcal{X} = \{(X_1, \mathcal{M}_{X_1}), \dots, (X_N, \mathcal{M}_{X_N})\}$. This ensures that $\vec{v}_j \notin \langle \gamma_j \rangle$, and further, that $\vec{v}_j \notin \langle \sigma \rangle$ for any σ containing γ_j

Claim: $\gamma_i \in \mathcal{R}^C(\langle \vec{v}_j \rangle) \iff \gamma_j \not\subseteq \gamma_i$

(\Rightarrow) Suppose $\sigma \in \mathcal{R}^C(\langle \vec{v}_j \rangle)$. Let $\gamma_{i_1}, \dots, \gamma_{i_k}$ be the maximal simplices and intersections of maximal simplices containing σ . That is, $\sigma \subseteq \gamma_{i_1} \cap \dots \cap \gamma_{i_k}$. By definition this means that $\vec{v}_j \in \langle \gamma_{i_1}, \dots, \gamma_{i_k} \rangle$. By our construction of \mathcal{X} (constructing all inventories from a single basis) this implies that $\vec{v}_j \in \langle \gamma_{i_k} \rangle$ for at least one of them, and that there is an $X \in \gamma_{i_k}$ such that $\vec{v}_j \in \mathcal{M}_X$. Therefore by our construction, $\{X\} \cup \gamma_j \notin \mathcal{R}$. Now suppose $\gamma_j \subseteq \sigma$. Then $\gamma_j \subseteq \sigma \cap \gamma_{i_k}$, and $X \in \gamma_{i_k}$, so $\gamma \cup \{X\} \subseteq \gamma_{i_k} \in \mathcal{R}$, a contradiction.

So $\sigma \in \mathcal{R}^C(\langle \vec{v}_j \rangle) \Rightarrow \gamma_j \not\subseteq \sigma$. In particular $\gamma_i \in \mathcal{R}^C(\langle \vec{v}_j \rangle) \Rightarrow \gamma_j \not\subseteq \gamma_i$.

(\Leftarrow) Suppose $\gamma_j \not\subseteq \gamma_i$. Say that $\gamma_j = \tau_1 \cap \dots \cap \tau_n$ for some \mathcal{R} -maximal $\{\tau_*\}$, and that $\gamma_i = \sigma_1 \cap \dots \cap \sigma_m$ for some \mathcal{R} -maximal $\{\sigma_*\}$.

$\gamma_j \not\subseteq \gamma_i$ implies that there is at least one maximal σ such that $\gamma_i \subseteq \sigma$, but that $\gamma_j \not\subseteq \sigma$.

If $\{X\} \cup \gamma_j \in \mathcal{R} \forall X \in \sigma$ then $\gamma_j \subseteq \sigma$ by the maximality of σ , a contradiction. So $\exists X \in \sigma$ such that $\{X\} \cup \gamma_j \notin \mathcal{R}$. Then by our construction $\vec{v}_j \in \mathcal{M}_X$, and therefore $\vec{v}_j \in \langle \sigma \rangle$. Since $\gamma_i \subseteq \sigma$ by definition $\gamma_i \in \mathcal{R}^C(\langle \vec{v}_j \rangle)$.

This proves the claim.

Now suppose $\gamma = \tau_1 \cap \dots \cap \tau_n$ for some maximal set of τ_* . Construct the vector

space $V_\gamma = \langle \vec{v}_j | \gamma_j \not\subseteq \gamma \rangle$. Then by the claim proved above and corollary 3.9:

$$\begin{aligned} \mathcal{R}^C(V_\gamma) &= \bigcap_{\gamma_j \not\subseteq \gamma} \mathcal{R}^C(\vec{v}_j) \\ &= \{ \bar{\gamma}_i \mid \gamma_i \subseteq \gamma \} = \bar{\gamma} \end{aligned} \tag{3.18}$$

Then, because all inventories are chosen from the same basis we can choose a single vector $\vec{v}_\gamma = \sum_{\gamma_j \not\subseteq \gamma} \vec{v}_j$ to get

$$\mathcal{R}^C(\langle \vec{v}_\gamma \rangle) = \mathcal{R}^C(V_\gamma) = \bar{\gamma} \tag{3.19}$$

For any set $S \subseteq \{1, \dots, P\}$, if we have $\sigma = \bigcup_{i \in S} \bar{\gamma}_i$ a union of closures of intersections of maximal simplices, we can take $V_\Sigma = \langle \vec{v}_j | \gamma_j \notin \Sigma \rangle$, and then similarly:

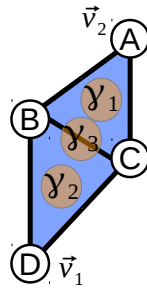
$$\begin{aligned} \mathcal{R}^C(V_\Sigma) &= \bigcap_{\gamma_j \notin \Sigma} \mathcal{R}^C(\vec{v}_j) \\ &= \{ \bar{\gamma}_i \mid \gamma_i \in \Sigma \} \\ &= \Sigma \end{aligned} \tag{3.20}$$

And finally, because all inventories are chosen from the same basis we can choose a single vector $\vec{v}_\Sigma = \sum_{\gamma_j \notin \Sigma} \vec{v}_j$ to get

$$\mathcal{R}^C(\langle \vec{v}_\Sigma \rangle) = \mathcal{R}^C(V_\Sigma) = \Sigma \tag{3.21}$$

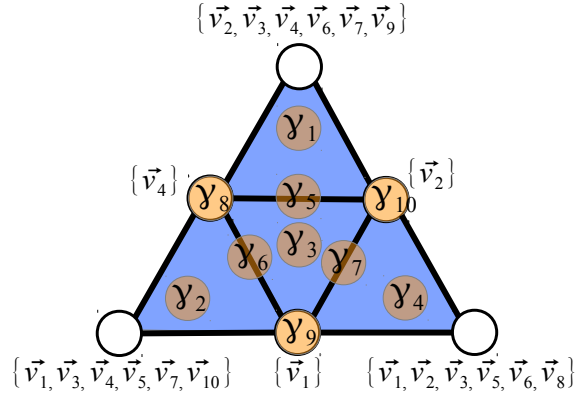
□

The figure below shows our simple example Rips complex with the data arrangement construction used in the proof. For this example the resulting data arrangement is almost the same as that used to prove $C1$ -resolvability. The complex has two maximal simplices γ_1, γ_2 , with intersection $\gamma_3 = \gamma_1 \cap \gamma_2$ and therefore requires a 3 dimensional vector space with three basis vectors $\mathcal{B} = \{\vec{v}_1, \vec{v}_2, \vec{v}_3\}$. The vertices B and C are faces of both γ_1 and γ_2 , and both share a maximal simplex with γ_3 , so their inventories are empty. A is a face of γ_1 so its inventory contains only \vec{v}_2 , and D is a face of γ_2 so its inventory contains only \vec{v}_1 . Note that \vec{v}_3 does not appear in any inventories.



Then $V = \text{GF}(2)^3$ has seven non-zero vectors. The resolvable subcomplexes for the 1D subspaces are as expected. Including \vec{v}_1 eliminates the simplex γ_1 , and including \vec{v}_2 eliminates γ_2 . The vector $\vec{v}_1 + \vec{v}_2$ is outside the span of either maximal simplex alone, but is in their collective span $\vec{v}_1 + \vec{v}_2 \in \langle \gamma_1, \gamma_2 \rangle$, so the corresponding coverage complex is their intersection, γ_3 . The final basis vector \vec{v}_3 is not in any inventories, so eliminates the entire complex.

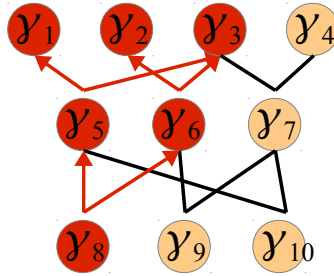
$\mathcal{R}^C(\vec{v}_1)$	$\mathcal{R}^C(\vec{v}_2)$	$\mathcal{R}^C(\vec{v}_1 + \vec{v}_2)$	$\mathcal{R}^C(\vec{v}_3)$
			\emptyset



The figure above shows a more interesting example. The complex has four maximal simplices $\gamma_1, \gamma_2, \gamma_3, \gamma_4$, three 2-fold intersections, $\gamma_5, \gamma_6, \gamma_7$, and three 3-fold intersections, $\gamma_8, \gamma_9, \gamma_{10}$. So a data arrangement over a 10-dimensional vector space with basis $\mathcal{B} = \{\vec{v}_1, \dots, \vec{v}_{10}\}$ suffices to resolve every subcomplex that is resolvable. We show some illustrative examples of $\mathcal{R}^C(U)$ in the table below.

$\mathcal{R}^C(\vec{v}_1)$	$\mathcal{R}^C(\vec{v}_2)$	$\mathcal{R}^C(\vec{v}_4)$
$\mathcal{R}^C(\vec{v}_3)$	$\mathcal{R}^C(\vec{v}_1 + \vec{v}_2 + \vec{v}_4)$	$\mathcal{R}^C(\vec{v}_8 + \vec{v}_9 + \vec{v}_{10})$
		\emptyset
$\mathcal{R}^C(\vec{v}_5)$	$\mathcal{R}^C(\vec{v}_6)$	$\mathcal{R}^C(\vec{v}_7)$
$\mathcal{R}^C(\vec{v}_8)$	$\mathcal{R}^C(\vec{v}_9)$	$\mathcal{R}^C(\vec{v}_{10})$

Another way of viewing this is as a poset. Let $\Gamma = \{\gamma \in \mathcal{R} \mid \gamma \text{ is resolvable}\}$. Then (Γ, \subseteq) is a poset ordered by inclusion. Since $\mathcal{R}^C(U)$ is a simplicial complex, and because including the vector \vec{v}_j in U has the effect of removing γ_j from the complex, it also has the effect of removing all simplices that contain γ_j . This is sometimes called the **upset** of $\{\gamma_j\}$, written $\uparrow\{\gamma_j\}$. For example, including \vec{v}_8 in U eliminates γ_8 from $\mathcal{R}^C(U)$, and also five other resolvable simplices, pictured below.



So in this construction if $\mathbb{F} = \text{GF}(2)$ there are 2^P vectors in V and they resolve all resolvable subcomplexes, but the correspondence with coverage complexes is not 1-1. In general if $S, T \subseteq \{1, \dots, P\}$ and $\vec{v} = \sum_{i \in S} \vec{v}_i$ and $\vec{w} = \sum_{i \in T} \vec{v}_i$ then $\mathcal{R}^C(\vec{v}) = \mathcal{R}^C(\vec{w})$ if and only if $\uparrow\{\gamma_i\}_{i \in S} = \uparrow\{\gamma_i\}_{i \in T}$. So the resolvable subcomplexes of \mathcal{R} correspond to the unique upsets of (Γ, \subseteq) . As \mathcal{R} is finite, the upsets are characterized entirely by their minimal elements. In general for a finite poset the upsets are in 1-1 correspondence with the anti-chains. Therefore the resolvable subcomplexes are in 1-1 correspondence with the anti-chains of (Γ, \subseteq) . The anti-chains of (Γ, \subseteq) can be studied using Sperner's theorem and Dilworth's theorem [27, 28].

The Rips complex $N = 2n$ -vertex cross-polytope example provides an extreme case of resolvability. Every simplex is resolvable, and so every subcomplex is resolvable. The maximum width of the face poset (the maximum anti-chain size) is at

dimension $N/2$ with width $\binom{N}{N/2}$. The depth of the poset is of course $N - 1$, so the number of resolvable subcomplexes is on the order of $N^{\binom{N}{N/2}}$.

3.2.1 Single-Vector Resolvability

In this section we have implicitly used the following fairly obvious fact.

Lemma 3.27. *Let \mathcal{X} be a data arrangement over a vector space $V = \text{GF}(q)^D$ where all inventories \mathcal{M}_i are drawn from the same fixed basis $\mathcal{B} = \{\vec{v}_1, \dots, \vec{v}_D\}$ for V . Then for any $U \subseteq V$ there is a vector $\vec{v} \in V$ such that $\mathcal{R}^C(U) = \mathcal{R}^C(\vec{v})$. The same is true for \mathcal{R}^{C1} .*

Proof. Let V , \mathcal{B} , and \mathcal{X} be as in the statement of the lemma. Let $U \subseteq V$ be a linear subspace, and let $\langle \vec{w}_1, \dots, \vec{w}_m \rangle = U$ be a basis for U . Since \mathcal{B} is a basis each \vec{w}_i has a unique expression as a sum of elements of \mathcal{B} . Let $\mathcal{B}_U = \{\vec{v}_{i_1}, \dots, \vec{v}_{i_k}\} \subseteq \mathcal{B}$ be all of the vectors which are summands for at least one of the \vec{w}_i . Then \mathcal{B}_U is the smallest subset of \mathcal{B} that spans U .

Claim: $\mathcal{R}^{C1}(U) = \mathcal{R}^{C1}(\vec{v}_{i_1} + \dots + \vec{v}_{i_k})$.

(\subseteq)

Suppose $\sigma \in \mathcal{R}^C(U)$. Let $\tau_1, \dots, \tau_n \in \mathcal{R}$ be the \mathcal{R} -maximal cofaces of σ . Then by the definition of \mathcal{R}^C , $U \subseteq \langle \tau_1, \dots, \tau_n \rangle$. Let $\mathcal{B}_\sigma = \mathcal{M}_{\tau_1} \cup \dots \cup \mathcal{M}_{\tau_n}$. Then we know $U \subseteq \langle \mathcal{B}_\sigma \rangle$. Also, since all the inventories in \mathcal{X} are drawn from the same basis we know $\mathcal{B}_\sigma \subseteq \mathcal{B}$. But \mathcal{B}_U was the smallest subset of \mathcal{B} spanning U , so $\mathcal{B}_U \subseteq \mathcal{B}_\sigma$, and therefore $\vec{v}_{i_1} + \dots + \vec{v}_{i_k} \in \langle \mathcal{B}_\sigma \rangle$, and $\sigma \in \mathcal{R}^C(\vec{v}_{i_1} + \dots + \vec{v}_{i_k})$.

(\supseteq)

Suppose $\sigma \notin \mathcal{R}^C(U)$. Let τ_1, \dots, τ_n be all of the \mathcal{R} -maximal cofaces σ , and let $\mathcal{B}_\sigma = \mathcal{M}_{\tau_1} \cup \dots \cup \mathcal{M}_{\tau_n} \subseteq \mathcal{B}$. Then there must be $\vec{v}_0 \in U$ such that $\vec{v}_0 \notin \langle \tau_1, \dots, \tau_n \rangle = \langle \mathcal{B}_\sigma \rangle$. Since \mathcal{B} is a basis there is a unique set of coefficients such that $\vec{v}_0 = a_1 \vec{v}_{i_1} + \dots + a_k \vec{v}_{i_k}$. But since $\vec{v}_0 \notin \langle \mathcal{B}_\sigma \rangle$ there must be at least one \vec{v}_{i_x} in this sum such that $\vec{v}_{i_x} \notin \mathcal{B}_\sigma$. Therefore $\vec{v}_{i_1} + \dots + \vec{v}_{i_k} \notin \langle \mathcal{B}_\sigma \rangle$, and hence $\sigma \notin \mathcal{R}^C(\vec{v}_{i_1} + \dots + \vec{v}_{i_k})$. \square

If we remove the requirement that all inventories be drawn from a single fixed basis this is not true in general. It is almost true in the case where $V = \text{GF}(q)^2$, because all proper subspaces of V are rank 1, and are spanned by a single vector. However the example constructed later in figure 6.2 shows a case where no single vector has $\mathcal{R}^C(\vec{v}) = \mathcal{R}^C(V)$.

A more interesting counterexample uses rank 2 subspaces instead of individual vectors to resolve the simplices. Consider a data arrangement over $V = \text{GF}(2)^3$ whose Rips complex consists of four isolated 0-simplices, $[A], [B], [C], [D]$, with rank 2 inventories as pictured below.

$$\begin{array}{cccc}
 U_1 = \begin{array}{|c|} \hline 1 \\ \hline 0 \\ \hline 0 \\ \hline \end{array} & U_2 = \begin{array}{|c|} \hline 0 \\ \hline 1 \\ \hline 0 \\ \hline \end{array} & U_3 = \begin{array}{|c|} \hline 1 \\ \hline 1 \\ \hline 0 \\ \hline \end{array} & U_4 = \begin{array}{|c|} \hline 1 \\ \hline 0 \\ \hline 0 \\ \hline \end{array} \\
 \text{\textcircled{A}} & \text{\textcircled{B}} & \text{\textcircled{C}} & \text{\textcircled{D}}
 \end{array}$$

Each 0-simplex is maximal, and since none of the U_i are the same, they are each individually resolved under this data arrangement.

$$\begin{aligned}\mathcal{R}^C(U_1) &= \{[A]\} & \mathcal{R}^C(U_2) &= \{[B]\} \\ \mathcal{R}^C(U_3) &= \{[C]\} & \mathcal{R}^C(U_4) &= \{[D]\}\end{aligned}$$

However for any single vector $\vec{v} \in V$, $\mathcal{R}^C(\vec{v})$ contains either two of the points or no points. None of the 0-simplices are resolvable by a single vector (under this data arrangement).

3.3 Failure of Chambers's Proof for $\mathcal{R}^C(U)$

In the next sections and throughout this paper we will see several counterexamples to the conjecture that the coverage complex satisfies an analogue of theorem 2.6. It is natural to wonder what are the essential features that allow $\mathcal{R}^C(U)$ to fail here, and if there are any restrictions that can be put on \mathbb{X} or $\mathcal{R}(\mathbb{X})$ that would allow us to prove an analogue of theorem 2.6. In the latter case we do not have any satisfying result. In the former case, we have investigated the degree to which the coverage complex satisfies the preliminary geometric lemmas used by Chambers et al. It turns out that even for $D = 2$ dimensional vector spaces there are exceptional arrangements of points and data where lemma 2.8 fails. It is possible to have two edges in $\mathcal{R}^C(U)$ whose shadows intersect, but with none of the possible 2-simplices on the endpoints contained in $\mathcal{R}^C(U)$. The simplest example of this will be constructed in section 7.1.1.

The complication for the coverage complex is that we cannot generally look at

the induced complex on a subset of the points in isolation. With the Rips complex, given two points $X, Y \in \mathbb{X}$, $[X Y] \in \mathcal{R}(\mathbb{X})$ iff $[X Y] \in \mathcal{R}(\{X, Y\})$. Membership in $\mathcal{R}^C(U)$ depends on the collective inventories of maximal simplices. This is captured by the observation that for $\{X_0, \dots, X_k\} \subseteq \mathbb{X}$:

$$\mathcal{R}(\{X_0, \dots, X_k\}) = \mathcal{R}(\mathbb{X}) \cap \overline{[X_0 \cdots X_k]} \quad (3.22)$$

but in general, for $\{(X_0, \mathcal{M}_{X_0}), \dots, (X_k, \mathcal{M}_{X_k})\} \subseteq \mathcal{X}$:

$$\mathcal{R}^C(\{(X_0, \mathcal{M}_{X_0}), \dots, (X_k, \mathcal{M}_{X_k})\}, U) \neq \mathcal{R}^C(\mathcal{X}, U) \cap \overline{[X_0 \cdots X_k]} \quad (3.23)$$

The following is a (very restricted) analogue of lemma 2.8 for $\mathcal{R}^C(\mathcal{X}, U)$.

Lemma 3.28. *Let $\mathcal{X} = \{(A, \mathcal{M}_A), (B, \mathcal{M}_B), (Y, \mathcal{M}_Y), (Z, \mathcal{M}_Z)\}$ a finite planar data arrangement over a finite vector space $V = \mathbb{F}^D$. Suppose $[A B], [Y Z] \in \mathcal{R}^C(\mathcal{X}, U)$ for some $U \subseteq V$, and $AB \cap YZ = \{x\}$. Then $\mathcal{R}^C(\mathcal{X}, U)$ is a cone.*

Proof. Since AB and YZ intersect at x , at least one of the shadow edges must be shortest. Without loss of generality, assume Ax is the shortest shadow edge. Then by lemma 2.8 $[A Y Z] \in \mathcal{R}$. The crux of this proof is to look at all the ways $[A B]$ and $[Y Z]$ could get admitted to $\mathcal{R}^C(U)$.

To start, suppose $U \subseteq \langle A, B \rangle$ and $U \subseteq \langle Y, Z \rangle$. Then $U \subseteq \langle A, Y, Z \rangle$ and since $[A Y Z] \in \mathcal{R}$, we have $[A Y Z] \in \mathcal{R}^C(U)$. The only thing that could go wrong is if $[B Y]$ or $[B Z]$ is in $\mathcal{R}^C(U)$. Suppose $[B Y] \in \mathcal{R}^C(U)$. In that case $[A B], [B Y], [A Y] \in \mathcal{R}$, and therefore $[A B Y] \in \mathcal{R}$. Since $U \subseteq \langle A, B \rangle \subseteq \langle A, B, Y \rangle$,

$[A B Y] \in \mathcal{R}^C(U)$. By the same argument, if $[B Z] \in \mathcal{R}^C(U)$, then $[A B Z] \in \mathcal{R}^C(U)$. In any of these cases $\mathcal{R}^C(U)$ is a cone over A .

On the other hand suppose $U \not\subseteq \langle A, B \rangle$. Since $[A B] \in \mathcal{R}^C(\mathcal{X}, U)$ it must be contained in some set of maximal simplices whose inventories together span U .

Say $[A B Y] \in \mathcal{R}$ and $U \subseteq \langle A, B, Y \rangle$. Then $[A B Y] \in \mathcal{R}^C(U)$, and if $[B Y] \in \mathcal{R}^C(U)$ then $[A B Y Z] \in \mathcal{R}$, and hence $[A B Y Z] \in \mathcal{R}^C(U)$. The same argument applies if $[A B Y] \in \mathcal{R}$ and $U \subseteq \langle A, B, Y \rangle$. On the other hand if $U \not\subseteq \langle A, B, Y \rangle$ and $U \not\subseteq \langle A, B, Z \rangle$. Then in order to get $[A B] \in \mathcal{R}^C(U)$ we must have $U \subseteq \langle A, B, Y, Z \rangle$, and either $[A B Y Z] \in \mathcal{R}$, or $[A B Y], [A B Z] \in \mathcal{R}$ (which implies $[A B Y Z] \in \mathcal{R}$). So in either case $[A B Y Z] \in \mathcal{R}^C(U)$, and $\mathcal{R}^C(U)$ is a cone over each vertex. \square

This example worked because \mathcal{X} consisted of only four points. When even one more point is involved it becomes very difficult to draw conclusions. When two more points are involved the analogue to lemma 2.8 fails, even for \mathcal{R}^{C1} .

When more points are involved, verifying an analogue of lemma 2.8 splits up into many cases, indexed by the linear dependences between the inventories of the points. We will give an example of one of those cases for the five-point situation. The investigation will require another lemma that captures some geometry of the triangle inequality. This is Lemma 4.2 from another paper by Chambers et al. [29].

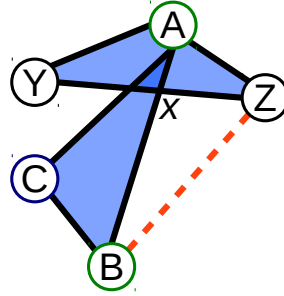
Lemma 3.29. *Let $\{A, B, Y, Z\} \subseteq \mathbb{X} \subseteq \mathbb{R}^2$. Suppose $[A B], [Y Z] \in \mathcal{R}(\mathbb{X})$ and $AB \cap YZ = \{x\}$. Then either $[A Y] \in \mathcal{R}(\mathbb{X})$ or $[B Z] \in \mathcal{R}(\mathbb{X})$.*

Proposition 3.30. *Let $V = \mathbb{F}^D$ be a vector space and let*

$\mathcal{X} = \{(A, \mathcal{M}_A), (B, \mathcal{M}_B), (C, \mathcal{M}_C), (Y, \mathcal{M}_Y), (Z, \mathcal{M}_Z)\}$ be a finite planar data arrangement over V , and $U \subseteq V$ a linear subspace. Suppose $[A B], [Y Z] \in \mathcal{R}^C(\mathcal{X}, U)$ for some $U \subseteq V$, $AB \cap YZ = \{x\}$, and that Ax is the shortest of the four shadow edges impinging on x . Suppose further, that $U \subseteq \langle A, B, C \rangle$, $U \subseteq \langle Y, Z \rangle$, and $[A B C] \in \mathcal{R}$. Then $\mathcal{R}^C(\mathcal{X}, U) \cap \overline{[A B Y Z]}$ is a cone over $[A]$.

Proof. By lemma 2.8 we have $[A Y Z] \in \mathcal{R}$, and since we assumed $U \subseteq \langle Y, Z \rangle$ we have $U \subseteq \langle A, Y, Z \rangle$, so $[A Y Z] \in \mathcal{R}^C(U)$.

The possible problem we face here is if $[B Z] \in \mathcal{R}^C(U)$ but $[A B Z] \notin \mathcal{R}^C(U)$, or similarly for $[B Y]$. Because of the symmetry in this arrangement we need only consider the $[B Z]$ case.



Suppose $[B Z] \in \mathcal{R}^C(U)$. Note first that if $[B Y] \in \mathcal{R}$ then $[A B Y Z] \in \mathcal{R}$ and since $U \subseteq \langle Y, Z \rangle \subseteq \langle A, B, Y, Z \rangle$ we have $[A B Y Z] \in \mathcal{R}^C(U)$. Then $\mathcal{R}^C(\mathcal{X}, U) \cap \overline{[A B Y Z]}$ is still a cone over $[A]$.

So we may assume that $[Y Z] \notin \mathcal{R}$.

Since $[B Z] \in \mathcal{R}^C(U)$ it must be in the intersection of some maximal simplices of \mathcal{R} whose collective inventories span U . Fortunately, without $[B Y]$ there aren't many possible maximal simplices left.

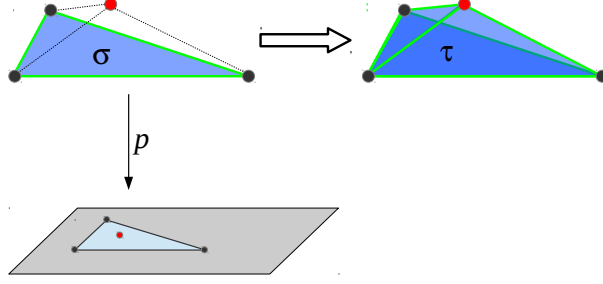


Figure 3.1: The situation in Lemma 3.31. If a vertex X lies inside the projection in \mathbb{R}^2 of a simplex $\sigma \in \mathcal{R}^C(U)$, then there is a simplex $\tau \in \mathcal{R}^C(U)$ containing both σ and X .

Note that now $[A B Z] \in \mathcal{R}$. If $U \subseteq \langle B, Z \rangle$ or $U \subseteq \langle A, B, Z \rangle$ then $[A B Z] \in \mathcal{R}^C(U)$ and $\mathcal{R}^C(\mathcal{X}, U) \cap \overline{[A B Y Z]}$ is still a cone over $[A]$.

The only remaining possibility is if $U \subseteq \langle B, C, Z \rangle$. But then $[A B C Z] \in \mathcal{R}$, and $U \subseteq \langle A, B, C, Z \rangle$, so $[A B C Z] \in \mathcal{R}^C(U) \Rightarrow [A B Z] \in \mathcal{R}^C(U)$, and therefore $\mathcal{R}^C(\mathcal{X}, U) \cap \overline{[A B Y Z]}$ is still a cone over $[A]$. \square

3.4 Geometric Connections Between \mathcal{X} and $\mathcal{R}^C(U, \mathcal{X})$

In this section fix the Rips radius $r = 1$. Let $V = \mathbb{F}^D$ and let \mathcal{X} be a finite planar data arrangement over V with Rips complex $\mathcal{R} = \mathcal{R}_r(\mathcal{X})$ and Rips shadow $\mathcal{S} = p(\mathcal{R})$. Let $U \subseteq V$ be a linear subspace, and let $\mathcal{R}^C(U) \subseteq \mathcal{R}$ be the coverage complex for U , with coverage shadow $\mathcal{S}^C(U) = p(\mathcal{R}^C(U))$. We will look at both the coverage complex, $\mathcal{R}^C(U)$ and the single-sponsored coverage complex, $\mathcal{R}^{C1}(U)$.

Lemma 3.31. *If $(X, \mathcal{M}) \in \mathcal{X}$ and $X \in p(\sigma)$ for some simplex $\sigma = [Y_0 Y_1 \cdots Y_k] \in \mathcal{R}^C(U, \mathcal{X})$, then $[Y_0 Y_1 \cdots Y_k X] \in \mathcal{R}^C(U)$.*

Similarly if $X \in p(\sigma)$ for some $\sigma = [Y_0 Y_1 \cdots Y_k] \in \mathcal{R}^{C1}(U, \mathcal{X})$, then $[X Y_0 Y_1 \cdots Y_k] \in \mathcal{R}^{C1}(U)$.

Proof. $X \in p(\sigma) = \text{conv}(Y_0, Y_1, \dots, Y_k)$, so by lemma 2.4 for any maximal $\tau \in \mathcal{R}$ with $\sigma \subseteq \tau$ we have $X \in \tau$.

In the case of $\mathcal{R}^{C^1}(U)$, if $\sigma = [Y_0 Y_1 \dots Y_k] \in \mathcal{R}^{C^1}(U)$ then there is $\tau \in \mathcal{R}$, $\sigma \subseteq \tau$, such that $U \subseteq \langle \tau \rangle$. By the observation above $[X Y_0 Y_1 \dots Y_k] \subseteq \tau$, so $[X Y_0 Y_1 \dots Y_k] \in \mathcal{R}^{C^1}(U)$.

In the case of $\mathcal{R}^C(U)$, if $\sigma = [Y_0 Y_1 \dots Y_k] \in \mathcal{R}^C(U)$ then there are $\tau_1, \dots, \tau_m \in \mathcal{R}$, $\sigma \subseteq \tau_i \ \forall i = 1, \dots, m$, such that $U \subseteq \langle \tau_1, \dots, \tau_m \rangle$. By the observation above $[X Y_0 Y_1 \dots Y_k] \subseteq \tau_i \ \forall i = 1, \dots, m$ as well, so $[X Y_0 Y_1 \dots Y_k] \in \mathcal{R}^C(U)$. \square

Lemma 3.32. *Let $\Sigma \subseteq \mathcal{R}^C(U) \subseteq \mathcal{R}$ be such that $p(\Sigma) = \mathcal{S}$. Let $\tilde{\Sigma}$ be the closure of the set of maximal simplices in $\mathcal{R}^C(U)$ which have a face in Σ . Then all of the vertices in \mathcal{R} are also in $\tilde{\Sigma}$. That is, $\tilde{\Sigma}^{(0)} = \mathcal{R}^C(U)^{(0)} = \mathcal{R}^{(0)}$.*

The same is true of $\mathcal{R}^{C^1}(U)$.

Proof. We already know $\tilde{\Sigma}^{(0)} \subseteq \mathcal{R}^C(U)^{(0)} \subseteq \mathcal{R}^{(0)}$. We must show $\mathcal{R}^{(0)} \subseteq \tilde{\Sigma}^{(0)}$.

Let $[X] \in \mathcal{R}^{(0)}$. Since $X \in \mathcal{S} = p(\Sigma)$ there must be some $\sigma = [Y_0 \dots Y_k] \in \Sigma$ such that $X \in p(\sigma)$. Then by lemma 3.31 $[Y_0 \dots Y_k X] \in \mathcal{R}^C(U)$, which has a face in Σ , as does any $\mathcal{R}^C(U)$ -maximal simplex containing $[Y_0 \dots Y_k X]$. Therefore $[X] \in \tilde{\Sigma}^{(0)}$. The same proof applies for $\mathcal{R}^{C^1}(U)$. \square

If $\mathcal{R}^C(U)$ has a contractible fence subcomplex, then every point of \mathcal{X} is in the coverage complex, and there are no isolated/disconnected points (besides any that are in \mathcal{R}). The same is true for $\mathcal{R}^{C^1}(U)$.

Proposition 3.33. *Assume there is a fence subcomplex for \mathcal{S} , $\mathcal{F} \subseteq \mathcal{R}^{C^1}(U) \subseteq \mathcal{R}^C(U) \subseteq \mathcal{R}$ with $\mathcal{F}^+ \simeq S^1$. Let $0 \neq [\gamma] \in H_1(\mathcal{F}^+)$.*

If $[\gamma] = 0$ in $H_1(\mathcal{R}^C(U))$ then $\mathcal{R}^C(U)^{(0)} = \mathcal{R}^{(0)}$ and $H_0(\mathcal{R}^C(U)) = H_0(\mathcal{R})$.

If $[\gamma] = 0$ in $H_1(\mathcal{R}^{C^1}(U))$ then $\mathcal{R}^{C^1}(U)^{(0)} = \mathcal{R}^{(0)}$ and $H_0(\mathcal{R}^{C^1}(U)) = H_0(\mathcal{R})$.

Proof. For finite simplicial complexes and their shadows, connectedness and path connectedness coincide. The following proof also works for \mathcal{R}^{C^1} .

Since $[\gamma] = 0$ in $H_1(\mathcal{R}^C(U))$ there must be a 2-chain $\alpha \in C_2(\mathcal{R}^C(U))$ such that $\partial\alpha = \gamma$. Let $\Sigma_\alpha \subseteq \mathcal{R}^C(U)$ be the closure of the set of 2-simplices with non-zero coefficients in α . Then by Corollary 3.17, $p(\Sigma_\alpha) = \mathcal{S}$. Therefore by lemma 3.32 there is a $\mathcal{R}^C(U)$ -maximal simplex $\sigma \in \mathcal{R}^C(U)$ such that $X \in \sigma$. In particular $[X] \in \mathcal{R}^C(U)$. Since X was any point in \mathcal{X} we conclude that every point in \mathcal{X} is in $\mathcal{R}^C(U)$. So $\mathcal{R}^C(U)^{(0)} = \mathcal{R}^{(0)}$. Furthermore X is connected to every vertex of σ , and therefore connected to Σ_α by at least one 1-simplex.

Also, because $p(\mathcal{F}) = \partial\mathcal{S}$, we know $p_* : H_0(\mathcal{R}^C(U)) \rightarrow H_0(\mathcal{S})$ is surjective. Suppose we have two vertices $X, Y \in \mathcal{X}$ that lie in the same connected component of \mathcal{S} , namely $[X] = [Y]$ in $H_0(\mathcal{S})$. By the argument above X and Y are each connected to vertices X', Y' in Σ_α by at least one 1-simplex. Since X' and Y' both lie in the same component of \mathcal{S} we can draw a path between them in \mathcal{S} . This path is homotopic rel its endpoints to a path lying entirely in the shadow of the 1-skeleton of Σ_α . This path can be lifted to Σ_α , giving a path from X' to Y' in $\mathcal{R}^C(U)$, and therefore a 1-chain $\rho \in C_1(\mathcal{R}^C(U))$ such that $\partial\rho = [X] - [Y]$. Therefore $[X] = [Y]$ in $H_0(\mathcal{R}^C(U))$. \square

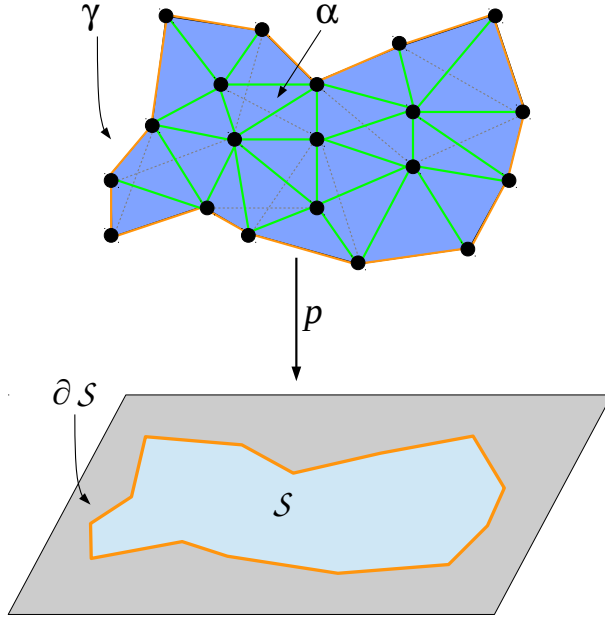


Figure 3.2: The situation in Prop 3.33. $C_1(\mathcal{R}^C(U))$ contains a 1-cycle γ that projects to the boundary of the Rips shadow, $\partial\mathcal{S}$. We assume $[\gamma] = 0$ in $H_1(\mathcal{R}^C(U))$, so there is a 2-chain $\alpha \in C_2(\mathcal{R}^C(U))$ with $\partial\alpha = \gamma$.

Another intuitive fact about the Rips complex is that in the situation of prop 3.33, where the outer fence subcomplex \mathcal{F}^+ is null-homologous, then we must have $H_1(\mathcal{R}) = 0$. This follows from Chambers’s isomorphism, theorem 2.6. This turns out not to be true in general for both $\mathcal{R}^C(U)$ and $\mathcal{R}^{C^1}(U)$. Even in the special case with a contractible fence subcomplex we do not always have $H_1(\mathcal{R}^C(U)) \cong H_1(\mathcal{S}^C(U))$ or $H_1(\mathcal{R}^{C^1}(U)) \cong H_1(\mathcal{S}^{C^1}(U))$. The following sections give examples of two distinct ways this can fail for $\mathcal{R}^C(U)$.

3.4.1 Layered Loops in $\mathcal{R}^C(U)$

In $\mathcal{R}^C(U)$ it is possible to have a null-homologous outer fence subcomplex, $\mathcal{F}^+ \subseteq \mathcal{R}^C(U)$ such that $H_1(\mathcal{F}^+) \xrightarrow{i_*} H_1(\mathcal{R}^C(U))$ is the zero map, and still have $H_1(\mathcal{R}^C(U)) \neq 0$. Figure 3.3 shows a “layered loop” example with $V = \text{GF}(2)^2$. The

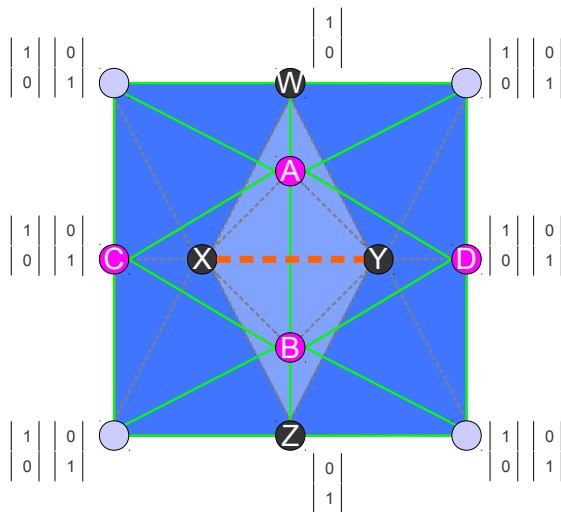


Figure 3.3: The example used in section 3.4.1 in which the outer fence cycle $[\gamma] \in H_1(\mathcal{F}^+)$ is null-homologous in $H_1(\mathcal{R}^C(V))$, but $H_1(\mathcal{R}^C(V)) \neq 0$. The green edges highlight a 2-chain with γ as its boundary. $[X Y]$ is a maximal 1-simplex.

solid green edges show the 1-skeleton of a 2-chain α with $p(\partial\alpha) = \partial\mathcal{S}$. The gray dashed edges show other 1-simplices included in $\mathcal{R}^C(U)$. The orange dashed edge $[X Y]$ is a maximal 1-simplex in $\mathcal{R}^C(V)$.

We see that both of the 2-simplices $[A B C]$ and $[A B D]$ are in $\mathcal{R}^C(V)$. Therefore by lemma 3.33 $[A B C X], [A B D Y] \in \mathcal{R}^C(V)$. Also $[X Y W] \in \mathcal{R}^C(\binom{1}{0})$ and $[X Y Z] \in \mathcal{R}^C(\binom{0}{1})$, but neither is in $\mathcal{R}^C(V)$. However $[X Y] \in \mathcal{R}^C(\binom{1}{0}) \cap \mathcal{R}^C(\binom{0}{1}) = \mathcal{R}^C(V)$. So $[X]$ and $[Y]$ are contained in different maximal simplices with faces in α , but $[X Y]$ is not contained in any 2-simplex. Therefore $[X Y]$ is part of a 1-cycle, but it is not bounding to any 2-chain, and so there is a 1-cycle $\rho \in C_1(\mathcal{R}^C(V))$ containing $[X Y]$ such that $[\rho] \neq 0$ in $H_1(\mathcal{R}^C(V))$.

Prop 3.33 basically ensures that this sort of layered loop cannot be more than one hop long.

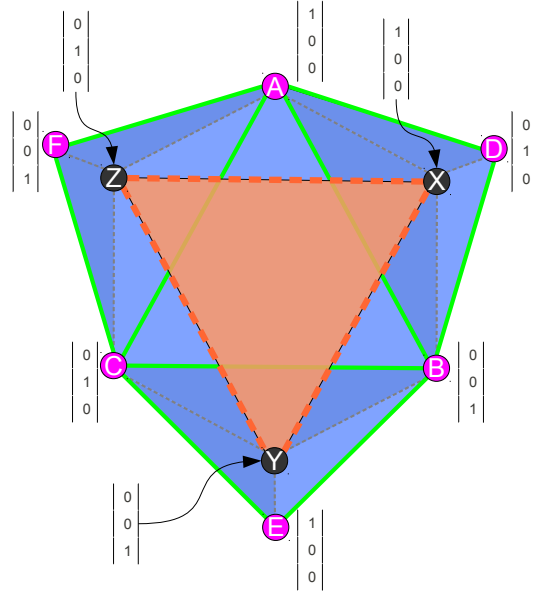


Figure 3.4: The example used in section 3.4.2 in which the outer fence cycle $[\gamma] \in H_1(\mathcal{F}^+)$ is null-homologous in $H_1(\mathcal{R}^C(V))$, but $H_1(\mathcal{R}^C(V)) \neq 0$. In this case there is a maximal (in $\mathcal{R}^C(V)$) 2-simplex, $[X Y Z]$, that only intersects the contractible 2-chain at its vertices.

3.4.2 Layered 2-simplices in $\mathcal{R}^{C1}(U)$

The example in section 3.4.1 is not a problem for the single-sponsored coverage complex. The problematic 1-simplex, $[X Y]$, is admitted to $\mathcal{R}^C(V)$ because it is in the intersection of two maximal simplices that together span V . In the case of the single-sponsored coverage complex, a simplex must be contained in a *single* maximal simplex that spans V , so in the example in figure 3.3, $[X Y] \notin \mathcal{R}^{C1}(V)$.

It is still possible in $\mathcal{R}^{C1}(U)$ to have a null-homologous outer fence subcomplex, $\mathcal{F}^+ \subseteq \mathcal{R}^{C1}(U)$ such that $H_1(\mathcal{F}^+) \xrightarrow{i_*} H_1(\mathcal{R}^{C1}(U))$ is the zero map, and still have $H_1(\mathcal{R}^{C1}(U)) \neq 0$. In fact it is possible create a data arrangement that has a $\mathcal{R}^{C1}(U)$ -maximal 2-simplex $\sigma \in \mathcal{R}^{C1}(U)$ that only intersects the rest of $\mathcal{R}^{C1}(U)$ at its vertices. More generally it is (at least) possible to construct a data arrangement

that contains a maximal $(3k - 1)$ -simplex that only intersects the rest of $\mathcal{R}^{C1}(U)$ at three $(k - 1)$ simplices. This type of example works in $\mathcal{R}^C(U)$ as well.

An example of a layered 2-simplex is shown in figure 3.4. Just as before, fix the Rips radius at $r = 1$, let $V = \mathbf{GF}(2)^3$, and let $\vec{v}_1, \vec{v}_2, \vec{v}_3 \in V$ be the three standard basis vectors (or any basis) for V . To produce a \mathcal{R}^{C1} complex with a covering 2-chain we place three points, A, B, C evenly spaced at 120° to each other around a circle of radius $R_1 = \sqrt{3}/3$ centered at the origin. Each of these three points is given a single basis vector so that $[A B C] \in \mathcal{R}^{C1}(V)$. We place another three points, D, E, F , around a concentric circle of radius R_2 , where $\sqrt{3}/3 < R_2 \leq 2\sqrt{3}/3$, rotated 60° relative to the points on the inner circle. Each of these three points is within distance 1 of two of the points on the inner circle, and is given a single basis vector such that it spans V when combined with those two inner-circle points. This ensures $[A F C], [B D A], [C E B] \in \mathcal{R}^{C1}(V)$. Finally we place three final points, X, Y, Z , on the inner circle of radius R_1 with the same 60° offset as the points on the outer circle. Each of these points is given the same basis vector as one of the two inner-circle points that it is adjacent to, and such that $\langle X, Y, Z \rangle = V$. This ensures that $[X Y Z] \in \mathcal{R}^{C1}(V)$.

However consider the points A, X, Z . The simplex $[A X Z]$ is maximal in \mathcal{R} , but because $\langle X \rangle = \langle A \rangle$ we have $V \not\subseteq \langle A, X, Z \rangle$, and therefore $[A X Z] \notin \mathcal{R}^{C1}(V)$. Likewise with $[B X Y]$ and $[C Y Z]$. We are left with $[X Y Z]$ maximal in $\mathcal{R}^{C1}(V)$, and only intersecting the rest of the complex at its vertices. $H_1(\mathcal{R}^{C1}(V))$ has rank 2.

It is possible to repeat this pattern to tile the plane, however (unless we extend the vector space) the missing simplices that make this example interesting end up

in $\mathcal{R}^C(V)$ except on the edges of the data arrangement. It is only on the *corners* of the data arrangement that we can have two missing simplices under the same layered 2-simplex to give a non-trivial feature in H_1 .

3.4.3 Layered Loops and Resolvability

Given what we know about resolvability of simplices in $\mathcal{R}^C(U)$, the examples in sections 3.4.1 and 3.4.2 should not be that surprising. Any time we have a 1-simplex, $\sigma \in \mathcal{R}$, such that $\sigma = \tau_1 \cap \dots \cap \tau_n$ for some maximal simplices $\tau_1, \dots, \tau_n \in \mathcal{R}$, there is a vector space V , a data arrangement \mathcal{X} over V , and a vector $\vec{v} \in V$ such that $\sigma \in \mathcal{R}^C(\vec{v})$ but σ is not contained in the boundary of any 2-simplex in $\mathcal{R}^C(\vec{v})$. The interesting part about the example in section 3.4.1 is that it is possible to eliminate all of the maximal cofaces of σ but still retain a 2-chain bounded by γ .

The examples with layered 2-simplices require a resolvable (or $C1$ -resolvable, i.e. Rips-maximal) 2-simplex. Notice that the example in section 3.4.2 contains the same arrangement of points used to demonstrate the existence of dimension 3 cross-polytopes in the Rips complex of planar points.

Looking at the example in figure 3.4 one naturally wonders if it would be possible to cover \mathcal{S} with two “sheets” of 2-simplices such that no 1-simplices are homotopic between sheets in $\mathcal{R}^C(U)$. Of course this turns out not to be possible. Considering figure 3.4, $[X]$ lies in the shadow of $[A B D]$ giving rise to 3-simplex $[A B D X]$. If $[X Y Z]$ were part of a second sheet covering \mathcal{S} then each of A , B , and D would lie inside the shadow of some 2-simplex in that sheet and this leads to

common 1-simplices between sheets. Continuing this on all sides ends up producing a non-trivial 2-cycle between the two sheets.

We can however tile the plane by repeating the pattern in figure 3.3. Since each copy of $[X Y Z]$ is still resolvable we can produce arbitrarily many of these 2-cycles over a sufficiently large vector space. The fact that we produced the example in figure 3.3 over $\text{GF}(2)^3$ instead of $\text{GF}(2)^8$ is interesting though. Trying to extend the pattern while restricting V to rank 3 leads to more interesting structure that we do not intend to exposit here.

3.5 Computing $\mathcal{R}^C(U)$ and $\mathcal{R}^{C^1}(U)$

Here we will give a rough algorithm for computing the coverage complex. We would like for the algorithm to allow each node to deduce which simplices it is a vertex of based on local information. Then collecting these simplex lists from all the nodes would constitute the entire coverage complex. On the other hand if the simplex structure is known locally, a distributed homology computation could be done.

Note however that if each node computes all of the simplices that it is a vertex of, there will be a great deal of duplicated effort. Each k -simplex will be added to the complex $k + 1$ times. When testing the membership of a simplex in $\mathcal{R}^C(U)$ requires matrix reductions this amounts to a lot of wasted computation. This algorithm avoids that duplication by assuming each node has a unique sortable ID value, and stipulating that in a given simplex σ , the node with the smallest ID value is the

owner of σ , responsible for testing its membership in $\mathcal{R}^C(U)$. This algorithm is written from the perspective of a single node, X , but could be used in a centralized way by iterating through all nodes and applying it to each.

We assume that each node has the following information about itself.

ID	the identity of the node
\mathcal{N}	the set of all immediate neighbors of the node
\mathcal{N}^+	the set of all immediate neighbors of the node with an ID value greater than its own
\mathcal{M}	the vector inventory of the node

We also assume that each node has the following information for each immediate neighbor, Y :

$Y.ID$	the identity of the neighbor, Y
\mathcal{N}_Y	the list of neighbors of Y
\mathcal{N}_Y^+	the list of neighbors of Y that have an ID greater than $Y.ID$ (this is computable from $Y.ID$ and \mathcal{N}_Y).
\mathcal{M}_Y	the vector inventory of Y

This means that each node, X , is required to maintain a list of its neighbors, and share with its immediate neighbors:

- its ID
- the list of its neighbors
- its inventory vectors, \mathcal{M}_X

In the pseudocode below the recursive procedure `findCoverageCofaces()` takes a simplex $\sigma = [X_0 X_1 \cdots X_i]$ with the property that $X_0.\text{ID} < X_1.\text{ID} < \cdots < X_i.\text{ID}$, and finds all cofaces $\tau = [X_0 \cdots X_j]$, $j \geq i$, of σ (including σ itself) which are in $\mathcal{R}^C(U)$ up to dimension k , such that $X_0.\text{ID} < X_1.\text{ID} < \cdots < X_j.\text{ID}$. As a minor optimization we also pass in a set \mathcal{N}_{com} containing all of the neighbors that are common to the nodes in σ .

Algorithm 1 Compute the simplices of $\mathcal{R}^C(U)$ owned by a node X

```

1: procedure COMPUTE_COVERAGE_COMPLEX_SIMPLICES( $k, U$ )
2:   mySimplices.clear()  $\triangleright$  initialize the set of simplices computed
3:   findCoverageCofaces([X],  $\mathcal{N}$ ,  $k, U$ )  $\triangleright$  call a recursive helper function
4: end procedure
5:
6: procedure FIND_COVERAGE_COFACES( $\sigma, \mathcal{N}_{\text{com}}, k, U$ )
7:    $\mathcal{M} = \bigcup_{Y \in \mathcal{N}_{\text{com}}} \mathcal{M}_Y$ 
8:   if rank( $\mathcal{M} \cup U$ ) > rank( $\mathcal{M}$ ) then  $\triangleright$  test if  $\sigma \in \mathcal{R}^C(U)$ 
9:     return
10:  end if
11:  mySimplices.add( $\sigma$ )
12:  if size( $\sigma$ ) > k then
13:    return
14:  end if
15:   $Z = \text{lastVertex}(\sigma)$ 
16:  for all  $Y \in \mathcal{N}_Z^+$  do  $\triangleright$  Test higher-dim cofaces of  $\sigma$ 
17:    findCoverageCofaces( $\sigma \cup [Y]$ ,  $\mathcal{N}_{\text{com}} \cap \mathcal{N}_Y$ ,  $k, U$ )
18:  end for
19: end procedure

```

After executing this procedure on a node, X , the set `mySimplices` will contain exactly the simplices of $\mathcal{R}^C(U)$ owned by X . First note that throughout this procedure every node added to the simplex σ is a neighbor of X , so the iteration on line 16, where we iterate over the elements of \mathcal{N}_Z^+ , is reasonable because Z is a vertex of σ . Recall that \mathcal{N}_{com} contains all the neighbors that are common to all

of the nodes in σ . Therefore on line 7 we are effectively computing $\langle \tau_1, \dots, \tau_n \rangle$ for the maximal simplices $\tau_i \supseteq \sigma$. Then on line 8 we test whether or not the span of these vectors contains U . It is clear, then, that this procedure only adds simplices to `mySimplices` if they are actually in $\mathcal{R}^C(U)$. On the other hand, every simplex $\sigma \in \mathcal{R}^C(U)$ can have its vertices reordered to give a simplex $\tilde{\sigma} = [X_0 \ X_1 \ \dots \ X_i]$ such that $X_0.\text{ID} < X_1.\text{ID} < \dots < X_i.\text{ID}$. If $\sigma \in \mathcal{R}^C(U)$, then all of its faces are in $\mathcal{R}^C(U)$ as well. In particular $\tilde{\sigma}^- = [X_0 \ X_1 \ \dots \ X_{i-1}] \in \mathcal{R}^C(U)$. If $\tilde{\sigma}^-$ is added to `mySimplices`, then the procedure will continue to call `findCoverageCofaces()` for $\tilde{\sigma}$, and $\tilde{\sigma}$ will be added to `mySimplices`. By induction we conclude that every simplex in $\mathcal{R}^C(U)$ owned by X will be added to `mySimplices`.

Also note that no simplex is added to `mySimplices` more than once. To see this notice that in order to reach a point where we call `findCoverageCofaces()` with a simplex $\sigma = [X_0 \ X_1 \ \dots \ X_i]$ we must have $X_0.\text{ID} < X_1.\text{ID} < \dots < X_i.\text{ID}$, and therefore the chain of recursive calls to `findCoverageCofaces()` is unique.

This procedure can be used in a centralized way by having `computeCoverageComplexSimplices()` iterate over the list of all nodes. In the distributed setting this algorithm will tend to require nodes with low-numbered IDs to do more work. For example the node in the network with the smallest ID will own every simplex it is part of, and the node with the largest ID will not own any simplices. This unfair work distribution could be a problem in a situation where the nodes are battery-powered, because lower-ID nodes would drain their battery more quickly. It is also an issue for performance overall because computing all of $\mathcal{R}^C(U)$ is limited by the slowest node. Since our implementation is centralized it is not a big issue,

but optimizing this algorithm is a good area for future research.

Computing $\mathcal{R}^{C^1}(U)$, on the other hand, is a bit more difficult, even if we only want to compute the k -skeleton. Suppose we only want to compute the 2-skeleton, $\mathcal{R}^{C^1}(U)^{(2)}$, and suppose we have a 2-simplex $\sigma \in \mathcal{R}$. In order to determine if $\sigma \in \mathcal{R}^{C^1}(U)$ we need to find at least one maximal $\tau \in \mathcal{R}$ such that $\sigma \subseteq \tau$ and $U \subseteq \langle \tau \rangle$. If $\sigma \notin \mathcal{R}^{C^1}(U)$ we may have to iterate through all of the Rips-maximal cofaces of σ . In general there is one optimization based on the fact that $\mathcal{R}^{C^1}(U) \subseteq \mathcal{R}^C(U)$. Namely if in our search we come across a simplex such that $\sigma \notin \mathcal{R}^C(U)$ then we do not need to pursue any of its maximal cofaces. We can write a simple modification to the $\mathcal{R}^C(U)$ algorithm incorporating this optimization.

In the algorithm below the procedure `findMaximalCoverageCofaces()` fills the set `mySimplices` with all of the Rips-maximal simplices in $\mathcal{R}^C(U)$. Then by taking all faces of these simplices up to dimension k we obtain the k -skeleton of the single-sponsored coverage complex. The implementation of the procedure `addAllFaces(σ , k)` is not spelled out, but it is a simple enumeration of the subsets of σ of size up to $k + 1$.

The `findMaximalCoverageCofaces()` only adds each admissible Rips-maximal simplex to `myMaximalSimplices` once. However the individual k -faces of these simplices may appear in more than one maximal simplex, and therefore may be added to `mySimplices` multiple times.

Another natural optimization to investigate is to deal efficiently with changes in the data arrangement. Suppose we compute the coverage complex, compute its homology, and find a defect. We will want to add more vectors to some nodes' inven-

ories to fix the defect and continue testing for coverage as the changes take place. On the other hand, there may be too much redundancy in the data arrangement and we want to remove vectors from some nodes' inventories while maintaining coverage. There are interesting optimizations possible to cache certain data structures on the nodes so that it is not necessary to repeat this entire computation from scratch after each change. These optimizations are independent of optimizations to the homology computation that can handle changing simplicial complexes, such as the vineyard algorithm or zig-zag persistent homology, and we will not elaborate on them here.

Algorithm 2 Compute the simplices of $\mathcal{R}^{C^1}(U)$ owned by a node X

```

1: procedure COMPUTESSCOVERAGECOMPLEXSIMPLICES( $k, U$ )
2:   mySimplices.clear()           ▷ initialize the set of simplices computed
3:   myMaximalSimplices.clear()   ▷ initialize the set of maximal simplices
   computed
4:   findMaximalCoverageCofaces( $[X], \mathcal{N}, U$ )   ▷ call a recursive helper
   function
5:   for all  $\sigma \in$  myMaximalSimplices do
6:     addAllFaces( $\sigma, k$ )
7:   end for
8: end procedure
9:
10: procedure FINDMAXIMALCOVERAGECOFACES( $\sigma, \mathcal{N}_{\text{com}}, U$ )
11:    $\mathcal{M} = \bigcup_{Y \in \mathcal{N}_{\text{com}}} \mathcal{M}_Y$ 
12:   if  $\text{rank}(\mathcal{M} \cup U) > \text{rank}(\mathcal{M})$  then           ▷ test if  $\sigma \in \mathcal{R}^C(U)$ 
13:     return
14:   end if
15:   if  $\text{size}(\sigma) = \text{size}(\mathcal{N}_{\text{com}})$  then           ▷ Check if  $\sigma$  is maximal
16:     myMaximalSimplices.add( $\sigma$ )
17:     return
18:   end if
19:    $Z = \text{lastVertex}(\sigma)$ 
20:   for all  $Y \in \mathcal{N}_Z^+$  do           ▷ Test higher-dim cofaces of  $\sigma$ 
21:     findMaximalCoverageCofaces( $\sigma \cup [Y], \mathcal{N}_{\text{com}} \cap \mathcal{N}_Y, U$ )
22:   end for
23: end procedure

```

Chapter 4: Persistent Homology and Discounting Holes in \mathcal{S}

When distributing vectors amongst the points in a data arrangement, we cannot expect to verify coverage of an area larger than the Rips shadow. We would prefer not to assume too much about the shape of the Rips shadow. Theorem 3.19 and corollary 3.20 already gave us a criterion for testing if the Rips shadow is covered even when it suffers from topological holes. However that criterion relies on computing a homology or relative homology group from three different spaces, and requires that the outer boundary of the shadow be a simple closed curve. It also requires *knowing* the identity and decomposition of the fence subcomplex. It turns out that we can build on this and relate the results of chapter 3.1.3 to a persistent homology computation that makes testing coverage of the Rips shadow simpler and more intuitive, and will only require that the fence subcomplex exist. We do this by studying the inclusion map

$$i : \mathcal{R}^C(U) \hookrightarrow \mathcal{R} \tag{4.1}$$

and the associated long exact sequence.

We will still assume that there is a fence subcomplex $\mathcal{F} \subseteq \mathcal{R}^C(U) \subseteq \mathcal{R}$. Given this assumption, the cycles of \mathcal{F}^- give all the generators of $H_1(\mathcal{R})$ (by the

Chambers isomorphism). Therefore if $0 \neq [\alpha] \in H_1(\mathcal{R})$, since $\mathcal{F} \subseteq \mathcal{R}^C(U)$, there is an $[\tilde{\alpha}] \in H_1(\mathcal{R}^C(U))$ such that $i_*[\tilde{\alpha}] = [\alpha]$. Namely α is homologous to some sum of fence cycles, and those fence cycles are all contained in $C_1(\mathcal{R}^C(U))$.

This means that if we are looking for non-zero elements of $H_1(\mathcal{R}^C(U))$ to indicate flaws in the data coverage, we want to discount any such elements which do not map to zero under i_* . That is, a cycle $[\tilde{\alpha}] \in H_1(\mathcal{R}^C(U))$ is only problematic if $[\tilde{\alpha}] \in \ker(i_*)$. So if $\ker(H_1(\mathcal{R}^C(U)) \xrightarrow{i_*} H_1(\mathcal{R})) = 0$ then every point in the shadow of \mathcal{R} is also in the shadow of $\mathcal{R}^C(U)$, i.e. $\mathcal{S} = \mathcal{S}^C(U)$.

Looking at the long exact sequence for the pair $(\mathcal{R}, \mathcal{R}^C(U))$

$$\cdots \rightarrow H_2(\mathcal{R}, \mathcal{R}^C(U)) \xrightarrow{\delta_*} H_1(\mathcal{R}^C(U)) \xrightarrow{i_*} H_1(\mathcal{R}) \rightarrow H_1(\mathcal{R}, \mathcal{R}^C(U)) \rightarrow \cdots \quad (4.2)$$

we note that $\ker(i_*) = \text{im}(\delta_*)$, so checking that either one is zero gives us a more general coverage criterion based on H_1 .

Proposition 4.1. *Let $\mathcal{R} = \mathcal{R}(\mathbb{X})$ be a Rips complex for a finite set of points $\mathbb{X} \subseteq \mathbb{R}^2$, and let $\mathcal{S} = p(\mathcal{R})$ be its Rips shadow. Let Σ be a subcomplex containing a fence subcomplex $\mathcal{F} = \mathcal{F}^+ \sqcup \mathcal{F}^- \subseteq \Sigma \subseteq \mathcal{R}$.*

If $\ker(H_1(\Sigma) \xrightarrow{i_} H_1(\mathcal{R})) = 0$ then $p(\Sigma) = \mathcal{S}$.*

Proof. We will assume that $\mathcal{F}^+ \simeq S^1$, and therefore that \mathcal{S} is connected. If there are multiple components this proof can be applied to each of them separately. If the outer boundary has the homotopy type of a wedge of circles, then the points of intersection must be Rips vertices (by lemma 2.8), and this proof can be applied to the component contained inside each sub-cycle of \mathcal{F}^+ separately.

First we will show that:

$$\ker(H_1(\Sigma) \xrightarrow{i_*} H_1(\mathcal{R})) = 0 \quad \Rightarrow \quad \ker(H_1(\Sigma, \mathcal{F}^-) \xrightarrow{i_*} H_1(\mathcal{R}, \mathcal{F}^-)) = 0$$

The inclusion map on pairs $i : (\Sigma, \mathcal{F}^-) \hookrightarrow (\mathcal{R}, \mathcal{F}^-)$ induces maps between the long exact sequences. We assumed $i_* : H_1(\Sigma) \rightarrow H_1(\mathcal{R})$ is injective.

$$\begin{array}{ccccccc} H_1(\mathcal{F}^-) & \xrightarrow{\iota_*} & H_1(\Sigma) & \xrightarrow{j_*} & H_1(\Sigma, \mathcal{F}^-) & \xrightarrow{\delta_*} & H_0(\mathcal{F}^-) \\ \parallel & & \downarrow i_* & & \downarrow i_* & & \parallel \\ H_1(\mathcal{F}^-) & \xrightarrow{\iota_*} & H_1(\mathcal{R}) & \xrightarrow{j_*} & H_1(\mathcal{R}, \mathcal{F}^-) & \xrightarrow{\delta_*} & H_0(\mathcal{F}^-) \end{array}$$

Suppose we have $0 \neq [\alpha] \in H_1(\Sigma, \mathcal{F}^-)$. If $\delta_*[\alpha] \neq 0$ then by commutativity $\delta_* i_*[\alpha] \neq 0$, and therefore $i_*[\alpha] \neq 0$. On the other hand, if $\delta_*[\alpha] = 0$ then by exactness there is $[\beta] \in H_1(\Sigma)$ such that $j_*[\beta] = [\alpha]$. By our assumption of injectivity $i_*[\beta] \neq 0$. If $j_* i_*[\beta] = 0$ then there must be a $[\gamma] \in H_1(\mathcal{F}^-)$ such that $\iota_*[\gamma] = [\beta]$. But this $[\gamma]$ is in $H_1(\mathcal{F}^-)$ in the top row too, so we have $\iota_*[\gamma] = [\beta]$ in $H_1(\Sigma)$, which contradicts the exactness of the top sequence. Therefore $j_* i_*[\beta] = i_* j_*[\beta] = i_*[\alpha] \neq 0$. We conclude that $i_* : H_1(\Sigma, \mathcal{F}^-) \rightarrow H_1(\mathcal{R}, \mathcal{F}^-)$ is injective.

To apply corollary 3.20 we need $0 \neq [\gamma] \in H_1(\mathcal{F}^+)$ such that $[\gamma] = 0$ in $H_1(\Sigma, \mathcal{F}^-)$. We assumed that $\mathcal{F}^+ \simeq S^1$, so we can take $[\gamma]$ to be a generator for $H_1(\mathcal{F}^+)$. Note that by excision $H_1(\mathcal{S}, \partial^- \mathcal{S}) = 0$ and so by Chambers's isomorphism $H_1(\mathcal{R}, \mathcal{F}^-) = 0$. Since $i_* : H_1(\Sigma, \mathcal{F}^-) \rightarrow H_1(\mathcal{R}, \mathcal{F}^-)$ is injective we must have $H_1(\Sigma, \mathcal{F}^-) = 0$, and therefore $[\gamma] = 0$ in $H_1(\Sigma, \mathcal{F}^-)$.

By corollary 3.20 $p(\Sigma) = \mathcal{S}$. □

Just to be explicit we can apply this to coverage complexes.

Corollary 4.2. *Let \mathcal{X} be a finite planar data arrangement over $V = \mathbb{F}^D$, and let $U \subseteq V$ be a linear subspace. Let $r > 0$ be some finite radius, and let $\mathcal{R} = \mathcal{R}_r(\mathcal{X})$ be the corresponding Rips complex and $\mathcal{S} = p(\mathcal{R})$ the Rips shadow. Let $\mathcal{R}^C(U)$ be the coverage complex for vector space U , and $\mathcal{S}^C(U) = p(\mathcal{R}^C(U))$ the coverage shadow. Suppose $\mathcal{F} = \mathcal{F}^+ \sqcup \mathcal{F}^- \subseteq \mathcal{R}^C(U) \subseteq \mathcal{R}$ is a fence subcomplex.*

If $\ker(H_1(\mathcal{R}^C(U)) \xrightarrow{i_} H_1(\mathcal{R})) = 0$ then $\mathcal{S}^C(U) = \mathcal{S}$.*

In order to apply this refined criterion we need to test either $\ker(H_1(\mathcal{R}^C(U)) \xrightarrow{i_*} H_1(\mathcal{R})) = 0$ or $\text{im}(H_2(\mathcal{R}, \mathcal{R}^C(U)) \xrightarrow{\delta_*} H_1(\mathcal{R}^C(U))) = 0$. Testing the kernel seems more straightforward.

Our assumptions about the fence subcomplex being contained in $\mathcal{R}^C(U)$ essentially ensure that $H_1(\mathcal{R}) \subseteq H_1(\mathcal{R}^C(U))$. Therefore one approach is simply to compute both $H_1(\mathcal{R})$ and $H_1(\mathcal{R}^C(U))$ and compare their ranks. If $H_1(\mathcal{R}^C(U))$ has a higher rank, then i_* must have non-zero kernel. If they have the same rank then $\ker(i_*) = 0$. A drawback of this is that there is not much room for optimization; we must compute two homology groups no matter what, and no work is shared even though the boundary operators involved share a lot of their structure. Another problem is that just comparing ranks does not identify which of the basis elements of $H_1(\mathcal{R}^C(U))$ corresponds to a new hole in coverage.

For a more comprehensive test we can look at the basis elements of $H_1(\mathcal{R}^C(U))$ themselves. To test if the kernel is zero we can start by computing a basis β_1, \dots, β_n for $H_1(\mathcal{R}^C(U))$. Each of these is still a cycle when viewed in $C_1(\mathcal{R})$. To see if they

are part of $\ker(i_\star)$ we only need to test if they become bounding when included into \mathcal{R} . The boundary map in question is

$$C_2(\mathcal{R}) \xrightarrow{\partial_1} C_1(\mathcal{R})$$

∂_1 for \mathcal{R} can be naturally expressed as a boundary matrix, M_1 , relative to a basis compatible with that for $\mathcal{R}^C(U)$; for example by representing all 1-simplices in \mathcal{R} but not $\mathcal{R}^C(U)$ in the bottom rows of the matrix. Then we would need to test if $\beta_i \in \text{im}(M_1)$ for each β_i ; or in other words, if $\beta_i \in \text{span}(\text{columns of } M_1)$. If that is the case for any β_i we have found a non-zero element of $\ker(i_\star)$. This test could be carried out by Gaussian elimination.

This approach is also not particularly optimized. It turns out that testing the kernel of i_\star could be computed more efficiently and discussed more elegantly using the tools of persistent homology. With $\mathcal{R}^C(U) \subseteq \mathcal{R}$ we have a two-step filtration and we want to check that all non-empty persistence intervals are infinite; that is, there is no torsion in the H_1 persistence module. We will devote the next few sections to background on persistent homology and an example of its use.

4.1 Persistent Homology

Persistent homology is a tool for tracking changes in homology throughout a filtration of topological spaces. The extremely useful things about persistent homology are the existence of efficient algorithms to effectively compute homology across the entire filtration with a single homology computation over a polynomial

ring, and the fact that it effectively tracks a generator for each feature from one stage of the filtration to another.

Persistent homology was first introduced as a computational tool by Edelsbrunner et al. [30], and later algebraically reframed by Zomorodian and Carlsson in [31], and expanded upon in [32]. Some good surveys are in [33–36].

The input to the persistence algorithm is a filtration of a topological space X .

$$X_0 \subseteq X_1 \subseteq \cdots \subseteq X_N = X \tag{4.3}$$

These inclusions induce maps on homology. Given a generator $[\alpha] \in H_k(X_i)$ we might want to ask if $[\alpha]$ has a preimage, $[\tilde{\alpha}] \in H_k(X_b)$, $b < i$, such that $i_*^{i-b}[\tilde{\alpha}] = [\alpha]$

$$\begin{array}{ccccccc} \cdots & \xrightarrow{i_*} & H_k(X_b) & \xrightarrow{i_*} & H_k(X_{b+1}) & \xrightarrow{i_*} & \cdots \xrightarrow{i_*} & H_k(X_i) \\ & & [\tilde{\alpha}] & \mapsto & i_*[\tilde{\alpha}] & \mapsto & \cdots \mapsto & [\alpha] \end{array}$$

or if after repeated applications of i_* , $[\alpha]$ eventually maps to zero.

$$\begin{array}{ccccccc} H_k(X_i) & \xrightarrow{i_*} & H_k(X_{i+1}) & \xrightarrow{i_*} & \cdots \xrightarrow{i_*} & H_k(X_{d-1}) & \xrightarrow{i_*} & H_k(X_d) \\ [\alpha] & \mapsto & i_*[\alpha] & \mapsto & \cdots \mapsto & i_* \cdots i_*[\alpha] & \mapsto & 0 \end{array}$$

If a generator $[\alpha] \in H_k(X_b)$ has no preimage in $H_k(X_{b-1})$ people typically say $[\alpha]$ is “born at time b ”, or “born at stage b of the filtration”. Similarly if $[\alpha]$ is born at time b and $i_*^{d-b-1}[\alpha] \neq 0$ but $i_*^{d-b}[\alpha] = 0 \in H_k(X_d)$ we say that $[\alpha]$ “dies at time d ” or “dies at stage d of the filtration”. Looking at this algebraically, one might

worry that there is potential for convergence in the lineage of the generators as they move up the filtration. I.e., there could be cases where $[\alpha] \neq [\beta] \in H_k(X_t)$, but $i_\star[\alpha] = i_\star[\beta]$ in $H_k(X_{t+1})$. However it works out that both the algorithms and the algebra avoid this complication by favoring the older generators. If $[\alpha]$ is born at time b_1 and $[\beta]$ is born at time b_2 , $b_1 < b_2$, then if $[\alpha]$ and $[\beta]$ become homologous at stage d in the filtration, $i_\star^{d-b_1}[\alpha] = i_\star^{d-b_2}[\beta]$, then this would be the death time of $[\beta]$.

Therefore we can unambiguously characterize homological generators in a filtration by their birth and death times. The **persistence diagram** of a filtration consists of a multi-set of pairs $\{(b_i, d_i)\}_{i \in \mathbb{I}}$. The individual birth/death pairs are called **persistence intervals**. The canonical example of a filtered simplicial complex (from [31]) is shown in figure 4.1, along with the two popular ways of plotting the persistence diagram. In dimension 0 at time $t = 0$ there are two connected components, A and B . At $t = 1$ A and B have joined the same component, so one of the persistence intervals is $(0, 1)$. It doesn't matter which generator is retained because they are indistinguishable. At $t = 1$ a new component, C also appears, and then is connected to the rest of the complex at $t = 2$, giving another persistence interval $(1, 2)$. The remaining persistence interval is infinite in the sense that it survives to the final stage of the filtration. In normal (not reduced homology) persistence there will always be at least one infinite interval in dimension zero.

In dimension 1 the first non-trivial 1-cycle with generator $[ABCD]$ appears at time $t = 2$. At $t = 3$ a second 1-cycle is born. Note that at this stage there are two choices for the generator for this second 1-cycle, either $[ABC]$ or $[ACD]$. At time

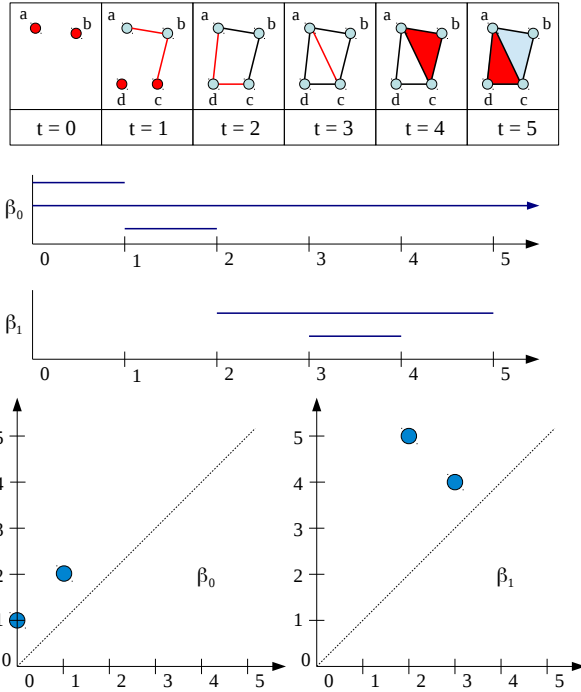


Figure 4.1: A classic example of a filtered simplicial complex and its persistent homology plotted in two different types of persistence diagrams.

$t = 4$ the cycle ABC becomes bounding and the generator that was born at $t = 3$, the younger generator, dies. The generator that was born at $t = 2$ dies at $t = 5$ when the cycle ACD is filled by a 2-simplex.

The first type of persistence diagram shown in figure 4.1 is sometimes called a **Betti barcode**. Each non-empty persistence interval, (b_i, d_i) , in dimension k is shown as a horizontal line running from b_1 to d_i in the diagram for β_k . Persistence intervals that survive through the final step in the filtration are called infinite intervals ($d_i = \infty$) and are shown as lines running from their birth time to the end of the diagram, with a small arrow indicating that they continue past the end of the filtration. In this type of diagram longer intervals indicate more prominent topological features.

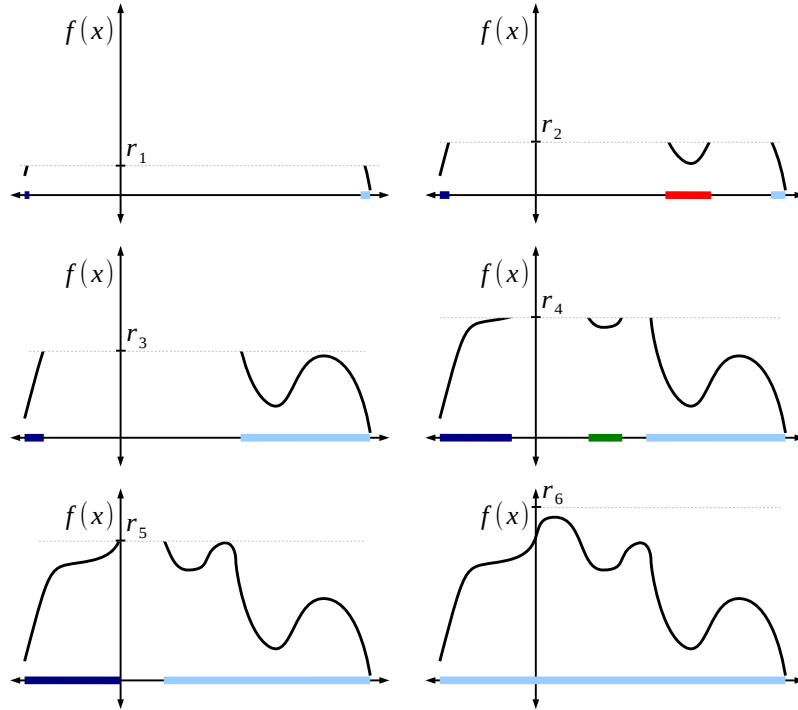


Figure 4.2: An example of a sublevel filtration of a manifold (an interval of \mathbb{R} in this case).

The second type of persistence diagram is more compact and generally used in connection with sublevel filtrations of height functions. Each persistence interval (b_i, d_i) is plotted as a point in the Cartesian plane. Since $b_i < d_i$ all the points will be above the diagonal, and points on the diagonal correspond to empty intervals. In this type of diagram points further from the diagonal correspond to more prominent topological features. This type of diagram does not represent infinite intervals, but when studying sublevel filtrations of a space based on height functions, there is no reason to have infinite intervals.

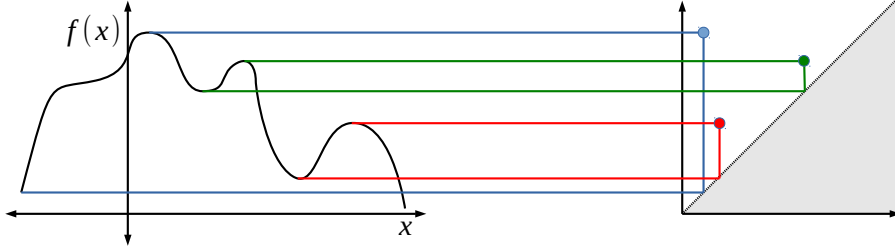


Figure 4.3: The correspondence between critical values of $f(x)$ and points in the persistence diagram of the sublevel filtration.

4.1.1 Applications of Persistent Homology

All of the meaning that can be extracted from a data set using persistent homology depends on the method by which one derives a filtered topological space. Very broadly speaking there are two popular ways to do this. The first is as a sub-level filtration of a manifold based on a real-valued function, such as a Morse height function or a probability density on the manifold. The second, and much more common in actual applications, is by building a distance based filtration of a simplicial complex constructed from a *point cloud* in a metric space.

When computing the persistence of a sublevel filtration of a manifold based on a real-valued function, the final stage of the filtration is known and tractable, namely it is the manifold, or a finite simplicial representation of the manifold. In some cases the manifold is familiar, such as a subset of Euclidean space or a sphere. Consider the example in figure 4.2. We have a continuous function $f : \mathbb{R} \rightarrow \mathbb{R}$ with finitely many critical points. Then for any $r \in \mathbb{R}$ we can define $X_r = \{x \in \mathbb{R} | f(x) \leq r\}$. Then $X_r \subseteq X_{r+\epsilon}$ and we have a filtration. In this case the underlying manifold is a line segment, so the only interesting homology is H_0 . The values of r where the induced inclusion $H_0(X_r) \xrightarrow{i_*} H_0(X_{r+\epsilon})$ is not the identity map are *critical values*,

and correspond to the birth and death times of the persistence intervals. The x -axis in figure 4.2 is highlighted to show the parts included in the filtration for different values of r .

Figure 4.3 shows a popular way of visualizing the correspondence between critical values and points in the persistence diagram. In the figure we see that small bumps in the function lead to points in the persistence diagram that are close to the diagonal, such as the one highlighted in green.

This type of filtration can be used to study the topography of a surface, literally or metaphorically [37] or density functions on higher dimensional manifolds. It has also been used in image analysis, for example attempting to identify indications of autism from cortical thickness measured from MRI scans [38]. Given certain constraints on the function f , there are stability results relating perturbations in f to changes in the persistence diagram [39,40]. Based partly on these stability results, the sublevel filtration approach has also been generalized to dynamic functions, or a homotopy of f . In that case, as f changes continuously, the points in the persistence diagram sweep out a path, called a *vine*. The collection of all such paths for a given homotopy is called a *persistence vineyard* [41,42]. In the paper where they are introduced, vineyards are used to analyze a time-varying two-dimensional surface based on protein folding data.

The approach of constructing a filtered simplicial complex based on point cloud data in a metric space seems more common in practice. In this approach we start with a collection of data points $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ with a distance function $d : \mathbf{X} \times \mathbf{X} \rightarrow \mathbb{R}$, and build a filtered simplicial complex with the points of \mathbf{X} as the

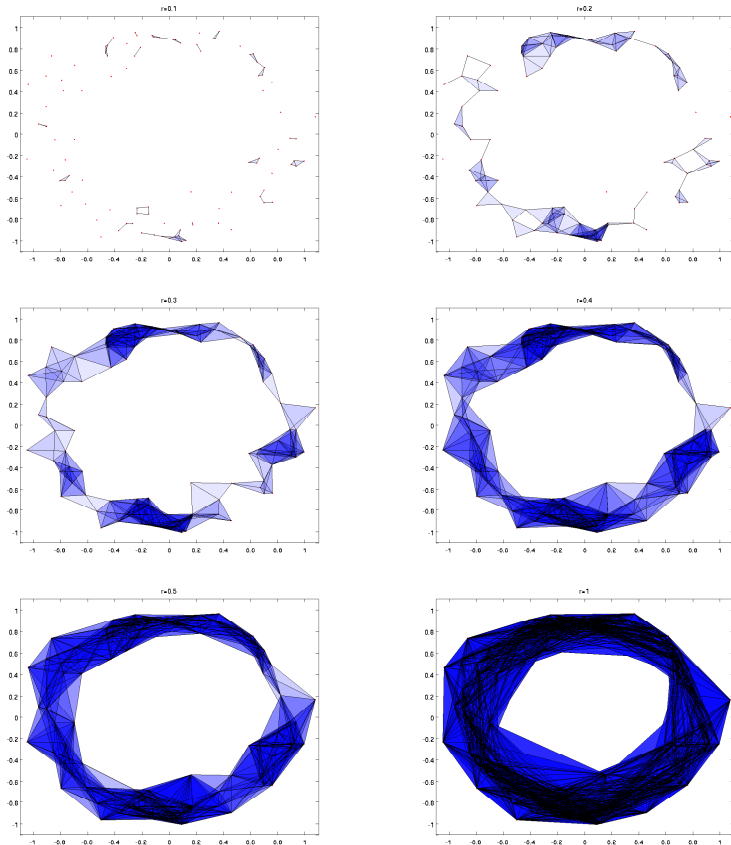


Figure 4.4: An example of a filtered Rips complex built on point cloud data

vertices. The Rips complex is the prototypical derivation. Others include the alpha complex [30, 43], and various types of witness complex [44, 45]. All of these have the property of forming simplices between nearby points.

Many data sets can be expected to have topological structure, but because the data points themselves can lie in very high dimensional space, discovering that structure is a challenge. Constructing something like the Rips complex on high dimensional metric data and computing homology is simple enough, though. The reason persistent homology is needed is that data can be incomplete, irregular, and noisy. Constructing something like the Rips complex requires the choice of a Rips radius, and choosing wrong can lead to nonsensical results.

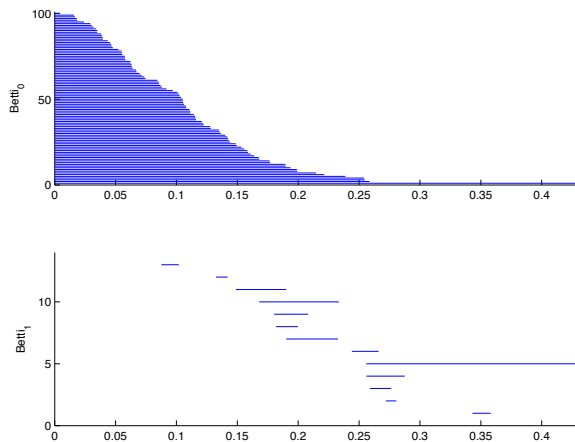


Figure 4.5: The persistence barcode for the filtration in figure 4.4. Several short-lived features appear as the Rips radius, r , increases. The generator corresponding to the prominent circle in the data is finally recovered when $r \approx 0.25$.

Figure 4.4 illustrates this. The figure shows a sequence of Rips complexes built on a set of 100 data points that lie near a circle, but with uniformly random angular placement and Gaussian noise added. For Rips radii that are too small, many points are disconnected and the topology of the circle is not recovered. Also for many radii small loops form amongst the points. These spurious loops could be detected by a homology computation on a single Rips complex and lead to the false conclusion that the data has the topological structure of several circles. Computing persistent homology allows us to pick out the topological features that persist across many scales of distance. The persistence barcode for this example is shown in figure 4.5.

One of the most famous examples of topological analysis of point cloud data is in [46] where the structure of a Klein bottle is discovered in data points derived from black and white digital photos. Section 5 gives details on an example from [47] that uses persistent homology to recover spatial information from encounter traces. Another uses persistent homology in natural language processing, attempting to

identify circular patterns in essays [48].

One technical obstacle to using persistent homology for point cloud analysis is that the size of the simplicial complex grows extremely quickly once the filtration reaches a certain point. For example, in the Rips complex on a set of N points, as the Rips radius increases the complex will eventually become an N simplex, with $\binom{N}{k}$ k -simplices. The witness complex is intended to mitigate this problem. It is similar to the Rips complex, but built on a subset of the data points, and tends to recover topological features at lower filtration values. Another approach to dealing with the complexity of point cloud-derived simplicial complexes is to allow simplifications of the complex while preserving topological features. The `CHomP` software has long used this sort of optimization for static simplicial complexes [49, 50]. An algorithm that excludes simplices that are likely to be topologically redundant from the Rips filtration has also been proposed [51].

4.2 Structure and Computation of Persistent Homology

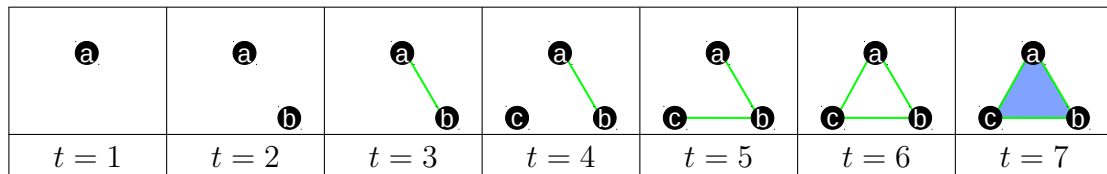
There are two commonly implemented algorithms for computing persistent homology: the pairing-based algorithm originally introduced by Edelsbrunner et al. [30], and the algorithm based on the persistence module correspondence introduced in [31]. Mathematically they must be equivalent on some level, but they have different implementations, see for instance the implementations of both in `JavaPlex` [52]. Other algorithms with new optimizations have been developed and implemented as well [53].

4.2.1 Persistence Computation Based On Pairing of Simplices

The original persistence algorithm, sometimes called **pHcol**, computes a pairing between “positive” simplices whose appearance creates a topological feature, and “negative” simplices, whose appearance destroys a topological feature. This pairing yields the persistence intervals, and the process also produces generators for the homology groups. In principle the algorithm can compute persistent homology over any finite field, but is typically presented only over $\text{GF}(2)$, which we will use here. The matrix looks cleaner without minus signs.

The premise of the algorithm is to start with a boundary matrix with the basis elements ordered by the filtration, and to perform the minimum reductions necessary to deduce the pairing. Since the computation is done over a finite field, torsion in the homology module is impossible, and it is not necessary to reduce completely to the Smith normal form.

Consider the filtration of a 2-simplex below. There are three 0-simplices, three 1-simplices, and one 2-simplex. Each appears at a different stage of the filtration.



The filtration order of the simplices is **a**, **b**, **ab**, **c**, **bc**, **ac**, **abc**. We can write a single boundary matrix, D , that contains all of the chain group information and has its basis elements ordered by this filtration. The corresponding basis elements are indicated at the head of the corresponding row or column.

$$D = \begin{array}{c} a \\ b \\ ab \\ c \\ bc \\ ac \\ abc \end{array} \begin{bmatrix} a & b & ab & c & bc & ac & abc \\ & & 1 & & & 1 & \\ & & 1 & & 1 & & \\ & & & & & & 1 \\ & & & & 1 & 1 & \\ & & & & & & 1 \\ & & & & & & 1 \end{bmatrix}$$

For a matrix, M , let $\mathbf{low}_M(j)$ be the row index (or corresponding basis element) of the last non-zero entry in column j . A matrix is considered “reduced” when $\mathbf{low}_M(j) \neq \mathbf{low}_M(j')$ for $j \neq j'$. The goal of the algorithm is to reduce D using only column operations, with the added constraint that columns can only be added up the filtration; that is, a column can only be added to a column to its right. These column operations can be represented by right-multiplication by an upper-triangular matrix, V , to give the reduced matrix.

$$R = DV \tag{4.4}$$

In the example above we have $\mathbf{low}_D(bc) = \mathbf{low}_D(ac)$, so we add column bc to column ac to get

$$R_1 = DV_1 = \begin{array}{c} a \\ b \\ ab \\ c \\ bc \\ ac \\ abc \end{array} \begin{bmatrix} a & b & ab & c & bc & ac+bc & abc \\ & & 1 & & & 1 & \\ & & 1 & & 1 & 1 & \\ & & & & & & 1 \\ & & & & 1 & & \\ & & & & & & 1 \\ & & & & & & 1 \end{bmatrix}$$

This leaves us with $\mathbf{low}_{R_1}(ab) = \mathbf{low}_{R_1}(ac + bc)$ so we add column ab to column

$ac + bc$ to get the final reduced matrix:

$$R = DV = \begin{matrix} & a & b & ab & c & bc & ac+bc+ab & abc \\ a & & & 1 & & & & \\ b & & & 1 & & 1 & & \\ ab & & & & & & & 1 \\ c & & & & & 1 & & \\ bc & & & & & & & 1 \\ ac & & & & & & & 1 \\ abc & & & & & & & \end{matrix}$$

The upper-triangular matrix V in this case looks like:

$$V = \begin{matrix} & a & b & ab & c & bc & ac & abc \\ a & 1 & & & & & & \\ b & & 1 & & & & & \\ ab & & & 1 & & & 1 & \\ c & & & & 1 & & & \\ bc & & & & & 1 & 1 & \\ ac & & & & & & 1 & \\ abc & & & & & & & 1 \end{matrix}$$

The pairing of positive and negative simplices is read off from the reduced matrix as (σ_i, σ_j) where $i = \text{low}_R(j)$. The generators for the homology classes can be read off of the columns of V . For this example we have:

(σ_i, σ_j)	interval	generator
(b, ab)	$(2, 3)$	b
(c, bc)	$(4, 5)$	c
(ac, abc)	$(6, 7)$	$ac + bc + ab$

4.2.2 Persistence Modules

The notion of persistence modules and using them to compute persistent homology groups was introduced by Zomorodian and Carlsson in [31].

Definition 4.3. Let R be a commutative PID. A **persistence module**, \mathcal{M} , is an indexed collection of R -modules, M^i , together with homomorphisms $\phi^i : M^i \rightarrow M^{i+1}$

$$\mathcal{M} = \{M^i, \phi^i\}$$

A persistence module is of **finite type** if the M^i are finitely generated and there exists $N \in \mathbb{N}$ such that $\phi^i : M^i \cong M^{i+1}$ for all $i \geq N$.

The main theorem contained in [31] is that the category of R -persistence modules of finite type is equivalent to the category of finitely generated non-negatively graded $R[t]$ modules.

Theorem 4.4. *Let $\mathcal{M} = \{M^i, \phi^i\}$ be a persistence module of finite type. Define the functor Υ taking persistence modules of finite type to finitely generated non-negatively graded $R[t]$ -modules by*

$$\Upsilon(\mathcal{M}) = \bigoplus_{i=0}^{\infty} M^i$$

with the action of t given by

$$t \cdot (m^0, m^1, m^2, \dots) = (0, \phi^0(m^0), \phi^1(m^1), \phi^2(m^2), \dots)$$

Then Υ defines an equivalence of categories.

This means that as long as $R = \mathbb{F}$ is a field, all R -persistence modules will have a tractable structure consisting of a free part and a torsion part:

$$\left(\bigoplus_{i=1}^n \Sigma^{a_i} \mathbb{F}[t] \right) \oplus \left(\bigoplus_{j=1}^m \Sigma^{b_j} (\mathbb{F}[t]/(t^{n_j})) \right)$$

where Σ^a is a shift upward in the grading by a degrees.

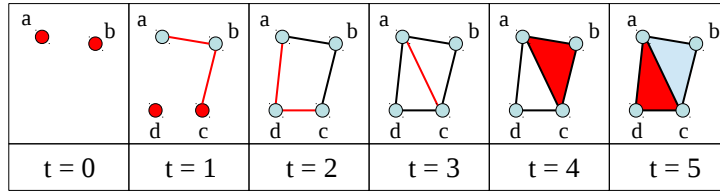
Clearly the persistent homology groups, over a field \mathbb{F} , for a filtration constitute an \mathbb{F} -persistence module with the ϕ^i being the induced inclusion maps. In fact the filtered chain complex $C_\bullet(X_0) \subseteq C_\bullet(X_1) \subseteq \dots \subseteq C_\bullet(X_n)$ is a persistence module, and we can use the structure of the graded polynomial ring to compute its homology as a persistence module.

4.2.3 Persistence Computation Based On Persistence Modules

Suppose we have a filtered simplicial complex $X_0 \subseteq X_1 \subseteq \dots \subseteq X_n = X$. In the persistence module-based algorithm each simplex in X has a corresponding degree (as an element of the $\mathbb{F}[t]$ -module) equal to the stage it first appears in the filtration. For example if $\sigma \in X_0$ then $\deg(\sigma) = 0$. On the other hand $\deg(t^2\sigma) = 2$, as $t^2\sigma$ corresponds to $\sigma \in X_2$.

We can write the boundary operators for $C_\bullet(X)$ in a way that captures all the filtration information. This gives us a chain complex over $\mathbb{F}[t]$. Theorem 4.4 says that computing the homology of this chain complex over $\mathbb{F}[t]$ effectively computes the persistent homology. Infinite persistence intervals correspond to free generators, and finite intervals correspond to torsion terms.

However, working over a field, certain simplifications to the reduction are possible. It isn't necessary to reduce the matrices to Smith Normal Form. We will describe the simplified algorithm by going through the same example as is used in [31]. The point of going through it here is to clarify the process, explain the relationship with the pairing algorithm, and to have a point of reference to refer to when explaining modifications to it.



For reference, the table below lists the degree of each of the simplices, according to the correspondence between the filtration and the $\mathbb{F}[t]$ module.

a	b	c	d	ab	bc	cd	ad	ac	abc	acd
0	0	1	1	1	1	2	2	3	4	5

Consider the boundary operator $\partial_1 : C_1 \rightarrow C_0$ operating on the basis element ad . Over the integers, and disregarding the filtration, we would have $\partial_1(ad) = d - a$. However in our $\mathbb{F}[t]$ -module, ad has degree 2, so $\partial_1(ad)$ must have degree 2. But a has degree 0 and d has degree 1, so we get $\partial_1(ad) = t \cdot d - t^2 \cdot a$ which is homogeneous

of degree 2. When we write the boundary operator ∂_k as a matrix M_k relative to the standard basis, the entries will be elements of $\mathbb{F}[t]$ satisfying

$$\deg \sigma_i + \deg M_k(i, j) = \deg \tau_j \quad (4.5)$$

where σ_i and τ_j are basis elements of C_{k-1} and C_k , respectively.

To compute H_1 , we start by writing the boundary operators for the complex in terms of the standard basis for C_0 , C_1 , and C_2 .

$$M_1 = \begin{matrix} & ab & bc & cd & ad & ac \\ d & & & t & t & \\ c & & 1 & -t & & t^2 \\ b & t & -t & & & \\ a & -t & & -t^2 & -t^3 & \end{matrix} \quad M_2 = \begin{matrix} & abc & acd \\ ab & t^3 & \\ bc & t^3 & \\ cd & & t^3 \\ ad & & -t^3 \\ ac & -t & t^2 \end{matrix} \quad (4.6)$$

Note that the basis elements in the rows of M_1 have been sorted in reverse filtration order; the opposite of how they were arranged in the **pHcol** algorithm. This ensures that the degree of the entries is strictly non-decreasing down the column. Therefore we only need to reduce the matrix M_k to column echelon form, and not completely to Smith Normal Form, to obtain the pivot entries. Once it is in column-echelon form, because of the increasing degree down each column, the entries below the pivots *could* be eliminated by row operations without changing the pivots. So the reason for arranging the matrix in this way is to more clearly relate the algorithm to underlying algebra and the Smith Normal Form. The process could equivalently be performed with the rows in opposite order, taking the bottom-most non-zero entry

as the pivot, as the **pHcol** algorithm does.

The first step in the algorithm is to reduce M_1 to column-echelon form using elementary column operations. It is helpful to see what happens in the steps of the reduction. The important point is that if we have a product of two matrices XY , we can perform an elementary column operation on X and a corresponding row operation on Y without changing the product. The column operations can be achieved by right-multiplication by a matrix P , and the row operations by left-multiplication by P^{-1} , so that

$$XY = (XP)(P^{-1}Y) \tag{4.7}$$

For a column operation

$$X(\cdot, i) \rightarrow X(\cdot, i) + qX(\cdot, j) \tag{4.8}$$

the corresponding row operation will be

$$Y(j, \cdot) \rightarrow Y(j, \cdot) - qY(i, \cdot) \tag{4.9}$$

We start by re-ordering the C_1 basis elements and proceed with the reduction.

$$\begin{aligned}
M_1 M_2 &= \begin{array}{c} d \\ c \\ b \\ a \end{array} \begin{bmatrix} cd & bc & ab & ad & ac \\ t & 0 & 0 & t & 0 \\ -t & 1 & 0 & 0 & t^2 \\ 0 & -t & t & 0 & 0 \\ 0 & 0 & -t & -t^2 & -t^3 \end{bmatrix} \begin{array}{c} cd \\ bc \\ ab \\ ad \\ ac \end{array} \begin{bmatrix} abc & acd \\ 0 & t^3 \\ t^3 & 0 \\ t^3 & 0 \\ 0 & -t^3 \\ -t & t^2 \end{bmatrix} \\
&= \begin{array}{c} d \\ c \\ b \\ a \end{array} \begin{bmatrix} cd & bc & ab & ad-cd & ac-t^2 \cdot bc \\ t & 0 & 0 & 0 & 0 \\ -t & 1 & 0 & t & 0 \\ 0 & -t & t & 0 & t^3 \\ 0 & 0 & -t & -t^2 & -t^3 \end{bmatrix} \begin{array}{c} cd+ad \\ bc+t^2 \cdot ac \\ ab \\ ad \\ ac \end{array} \begin{bmatrix} abc & acd \\ 0 & 0 \\ 0 & t^4 \\ t^3 & 0 \\ 0 & -t^3 \\ -t & t^2 \end{bmatrix} \\
&= \begin{array}{c} d \\ c \\ b \\ a \end{array} \begin{bmatrix} cd & bc & ab & ad-cd-t \cdot bc & ac-t^2 \cdot bc-t^2 \cdot ab \\ t & 0 & 0 & 0 & 0 \\ -t & 1 & 0 & 0 & 0 \\ 0 & -t & t & t^2 & 0 \\ 0 & 0 & -t & -t^2 & 0 \end{bmatrix} \begin{array}{c} cd+ad \\ bc+t^2 \cdot ac+t \cdot ad \\ ab+t^2 \cdot ac \\ ad \\ ac \end{array} \begin{bmatrix} abc & acd \\ 0 & 0 \\ 0 & 0 \\ 0 & -t^3 \\ 0 & -t^3 \\ -t & t^2 \end{bmatrix} \\
&= \begin{array}{c} d \\ c \\ b \\ a \end{array} \begin{bmatrix} cd & bc & ab & ad-cd-t \cdot bc-t \cdot ab & ac-t^2 \cdot bc-t^2 \cdot ab \\ t & 0 & 0 & 0 & 0 \\ -t & 1 & 0 & 0 & 0 \\ 0 & -t & t & 0 & 0 \\ 0 & 0 & -t & 0 & 0 \end{bmatrix} \begin{array}{c} cd+ad \\ bc+t^2 \cdot ac+t \cdot ad \\ ab+t^2 \cdot ac+t \cdot ad \\ ad \\ ac \end{array} \begin{bmatrix} abc & acd \\ 0 & 0 \\ 0 & 0 \\ 0 & -t^3 \\ -t & t^2 \end{bmatrix}
\end{aligned}$$

So we are left with:

$$\begin{aligned}
\widetilde{M}_1 &= \begin{array}{c} d \\ c \\ b \\ a \end{array} \begin{bmatrix} cd & bc & ab & ad-cd-t \cdot bc-t \cdot ab & ac-t^2 \cdot bc-t^2 \cdot ab \\ t & 0 & 0 & 0 & 0 \\ -t & 1 & 0 & 0 & 0 \\ 0 & -t & t & 0 & 0 \\ 0 & 0 & -t & 0 & 0 \end{bmatrix} \widehat{M}_2 = \begin{array}{c} cd+ad \\ bc+t^2 \cdot ac+t \cdot ad \\ ab+t^2 \cdot ac+t \cdot ad \\ ad \\ ac \end{array} \begin{bmatrix} abc & acd \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & -t^3 \\ -t & t^2 \end{bmatrix}
\end{aligned}$$

What we can read from \widetilde{M}_1 is that $Z_1 = \langle ad-t \cdot ab-t \cdot bc-cd, ac-t^2 \cdot ab-t \cdot bc \rangle$

is a basis for cycles of C_1 . Also the top non-zero entries in the first three columns

would become the diagonal entries of the Smith Normal Form.

Understanding the relationship with M_2 and obtaining B_1 is a little more elusive. Note that in our example, applying the row operations to M_2 caused all the rows corresponding to pivot columns in \widetilde{M}_1 to be zeroed out. This must *always* happen based on the fact that $\widetilde{M}_1 \widehat{M}_2 = M_1 M_2 = 0$. Therefore the rows of \widehat{M}_2 that correspond to pivot columns of \widetilde{M}_1 do not contribute to the 1-boundaries, B_1 . Furthermore the rows of M_2 corresponding to zero columns of \widetilde{M}_1 are not modified. This must also *always* be the case, since they are not pivot columns, there is not reason to add them to another column during the reduction.

Therefore there is no need to carry out any row operations on M_2 . We can simply discard the rows corresponding to the pivot columns of \widetilde{M}_1 and label the remaining rows with the corresponding basis element computed in the reduction of M_1 . In the case above we get:

$$\begin{array}{cc} & \begin{array}{cc} abc & acd \end{array} \\ \begin{array}{l} ad-cd-t \cdot bc-t \cdot ab \\ ac-t^2 \cdot bc-t^2 \cdot ab \end{array} & \begin{bmatrix} 0 & -t^3 \\ -t & t^2 \end{bmatrix} \end{array}$$

Swapping the columns puts it in column echelon form.

$$\check{M}_2 = \begin{array}{cc} & \begin{array}{cc} abc & acd \end{array} \\ \begin{array}{l} ad-cd-t \cdot bc-t \cdot ab \\ ac-t^2 \cdot bc-t^2 \cdot ab \end{array} & \begin{bmatrix} -t^3 & 0 \\ t^2 & -t \end{bmatrix} \end{array}$$

Sorting the rows by reverse filtration value and reducing to column echelon form we obtain:

$$\widetilde{M}_2 = \begin{matrix} & & abc & acd+t \cdot abd \\ ac-t^2 \cdot bc-t^2 \cdot ab & -t & 0 \\ ad-cd-t \cdot bc-t \cdot ab & 0 & t^3 \end{matrix} \begin{bmatrix} -t & 0 \\ 0 & t^3 \end{bmatrix}$$

The first row basis element has degree 3, and its pivot has degree 1. This contributes a term $\Sigma^3\mathbb{F}[t]/(t)$. This is a persistence interval (3, 4). The second row basis element has degree 2 and its pivot has degree 3. This contributes a term $\Sigma^2\mathbb{F}[t]/(t^3)$. This is a persistence interval (2, 5).

4.3 Discounting Holes in \mathcal{S}

Recall that we are interested in identifying coverage holes in $\mathcal{R}^C(U)$ that do not persist under the inclusion into \mathcal{R} . These are holes that could be fixed by adding or rearranging vectors in the inventories of the points. Algebraically we are looking at:

$$\ker(i_\star : H_1(\mathcal{R}^C(U)) \longrightarrow H_1(\mathcal{R})) \tag{4.10}$$

If we treat $\mathcal{R}^C(U) \subseteq \mathcal{R}$ as a two-stage filtration, then this kernel corresponds exactly with the non-infinite 1-dimensional persistence intervals.

Let us consider how this looks in the pairing-based persistence algorithm. Since we are only considering H_1 , assume we are dealing only with the 2-skeleta of the complexes. In a two-stage filtration we essentially have two sets of simplices: $\mathcal{R}^C(U)$ and $\mathcal{R} \setminus \mathcal{R}^C(U)$. For convenience call them $\Sigma = \{\sigma_1, \dots, \sigma_n\}$ and $T = \{\tau_1, \dots, \tau_m\}$,

respectively (note \mathbb{T} is not generally a simplicial complex). We order the basis elements by the filtration, but since all of the simplices fall into one of the two stages, and because the ordering of simplices within a stage of the filtration is arbitrary, we can divide the matrix D into blocks.

$$D = \begin{array}{c} \sigma_1 \\ \vdots \\ \sigma_n \\ \tau_1 \\ \vdots \\ \tau_m \end{array} \left[\begin{array}{ccc|ccc} \sigma_1 & \cdots & \sigma_n & \tau_1 & \cdots & \tau_m \\ \hline & & D^{1,1} & & & D^{1,2} \\ \hline & & 0 & & & D^{2,2} \end{array} \right] \quad (4.11)$$

Then this reduces to

$$R = DV = \begin{array}{c} \sigma_1 \\ \vdots \\ \sigma_n \\ \tau_1 \\ \vdots \\ \tau_m \end{array} \left[\begin{array}{ccc|ccc} \sigma_1 & \cdots & \sigma_n & \tau_1 & \cdots & \tau_m \\ \hline & & R^{1,1} & & & R^{1,2} \\ \hline & & 0 & & & R^{2,2} \end{array} \right] \quad (4.12)$$

Recall that a simplex is negative if its corresponding column in the reduced matrix is non-zero, and positive if its column is zero. For a positive simplex σ_i it is unpaired if there is no column σ_j such that $\text{low}_R(j) = i$, and paired (with finite death time) if there *is* a column σ_j with $\text{low}_R(j) = i$. For the purpose of discounting holes in \mathcal{S} we only care about positive simplices in Σ , i.e. columns that are zero in $R^{1,1}$. Furthermore we only care about positive simplices in Σ that are paired with

a simplex in T , i.e. rows that have $\text{low}_R(\tau_j) = \sigma_i$ for some column τ_j in $R^{1,2}$.

The reduction necessary to check the first condition can be optimized over the normal persistence algorithm. It will be clearer to break D up into dimension-specific boundary operators.

Let $\Sigma_k = \{\sigma_1^k, \dots, \sigma_{n_k}^k\}$ be the k -simplices in Σ , and $T_k = \{\tau_1^k, \dots, \tau_{m_k}^k\}$ be the k -simplices in T . Then we can write

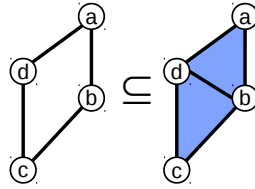
$$D_k = \begin{array}{c} \sigma_1^{k-1} \\ \vdots \\ \sigma_{n_k}^{k-1} \\ \tau_1^{k-1} \\ \vdots \\ \tau_{m_k}^{k-1} \end{array} \left[\begin{array}{ccc|ccc} \sigma_1^k & \dots & \sigma_{n_k}^k & \tau_1^k & \dots & \tau_{m_k}^k \\ & & D_k^{1,1} & & & D_k^{1,2} \\ \hline & & 0 & & & D_k^{2,2} \end{array} \right] \quad (4.13)$$

Then the matrices D_1 and D_2 contain all the information we need to compute the H_1 persistence. Since we are only interested in finding positive 1-simplices it is only necessary to reduce $D_1^{1,1}$. We can ignore the rest of the 1-simplices that appear only in T_1 . To see this, recall that to respect the filtration during reduction, we may only add a column to a column to its right. Therefore the columns in $D_1^{1,2}$ cannot affect the reduction of the columns of $D_1^{1,1}$.

Once we identify the positive 1-simplices in Σ_1 , we want to check if they are paired with a 2-simplex in T_2 . Unfortunately we cannot optimize the reduction of R_2 at all. In order to determine low_R for all of the τ_j^2 columns the lower block $D_k^{2,2}$ must get reduced. In order to reduce the upper right block $D_2^{1,2}$ properly, the upper left block $D_2^{1,1}$ must be reduced.

Thus, the limited scale of this filtration and our limited interest in certain types of persistence intervals allows us to optimize the computation somewhat, but the overall worst-case complexity will stay the same because the reduction of D^2 cannot be broken down into blocks.

As an example, consider two-stage the filtration shown below. This is just an example. There is no data arrangement that would give this as a coverage complex.



Then the dimension 1 and 2 boundary matrices would look like:

$$D_1 = \begin{array}{c} a \\ b \\ c \\ d \end{array} \left[\begin{array}{cccc|c} ab & bc & cd & ad & bd \\ -1 & & & -1 & \\ 1 & -1 & & & -1 \\ & 1 & -1 & & \\ & & 1 & 1 & 1 \end{array} \right] \quad D_2 = \begin{array}{c} ab \\ bc \\ cd \\ ad \\ bd \end{array} \left[\begin{array}{c|cc} & abd & bcd \\ 1 & & \\ & 1 & \\ & & 1 \\ -1 & & \\ \hline 1 & & -1 \end{array} \right]$$

The reduced matrices would be:

$$R_1 = \begin{array}{c} a \\ b \\ c \\ d \end{array} \left[\begin{array}{cccc|c} ab & bc & cd & ad-cd-bc-ab & bd-cd-bc \\ -1 & & & & \\ 1 & -1 & & & \\ & 1 & -1 & & \\ & & 1 & & \end{array} \right] \quad R_2 = \begin{array}{c} ab \\ bc \\ cd \\ ad \\ bd \end{array} \left[\begin{array}{c|cc} & abd & bcd+abd \\ 1 & 1 & \\ & 1 & \\ & & 1 \\ -1 & & -1 \\ \hline 1 & & \end{array} \right]$$

Looking at R_1 we see that the reduced column corresponding to ad is zero, and therefore ad is a positive simplex in the first stage of the filtration. Discovering

this did not require reducing the right part of the boundary matrix (though we have reduced it in the equation above). Looking at R_2 we see that $ad = \mathbf{low}_{R_2}(bcd)$, so ad is paired with bcd which appears in the second stage of the filtration. Therefore the cycle $[ad - cd - bc - ad]$ is in the kernel of the induced inclusion map on H_1 , and would be considered a problematic 1-cycle.

4.3.1 Allowing changes to $\mathcal{R}^C(U)$

Another natural algorithmic question to ask is “How much of this computation can be reused if we change $\mathcal{R}^C(U)$?” The idea is, if we are expanding the data arrangement to try to patch holes we want to check if we are successful. If we are removing vectors from nodes’ inventories in order to save storage space we want to check that we have not created new holes. On the other hand we may have a fixed data arrangement and we are traversing the lattice of linear subspaces of $U \subseteq V$ to try to isolate imperfections in the coverage. The complex at the top of the filtration is fixed in our case, so changing $\mathcal{R}^C(U)$ simply amounts to moving simplices between Σ and T (using the terminology of the last section). This is the same situation as in persistence vineyards [41, 42].

Persistence vineyards are interesting and useful partly because there is an efficient linear-time algorithm which can update the reduced matrix as the filtration changes. It is conceivable that we could use the same algorithm to update our pairing when a simplex appears or disappears from $\mathcal{R}^C U$. One difference is in the interpretation of the output. Because of the sublevel filtration setting vineyards

are developed in, stability results ensure that the points in the persistence diagram sweep out continuous paths in the vineyard. This interpretation makes no sense in our context. For us the persistence diagram is either empty or not, and there is no continuity between diagrams as we change the filtration. However it is desirable to track equivalent generators as the filtration changes.

Another obstacle is that algorithm used to repair the $R = DV$ decomposition only deals with the exchange of adjacent elements in the filtration. That is, it can only transpose adjacent rows and columns. Moving a simplex from Σ to T or from T to Σ will generally not involve the simplices at the boundary, σ_n and τ_1 . In our case any simplex could switch from one stage of the filtration to the other. Repairing the reduced matrices after a row/column exchange could trigger a new cascade of reduction steps, which would not be linear time. Optimizations along the lines of vineyards could be used, but not directly. Investigating this sort of optimization could be future work.

Chapter 5: Using Persistent Homology To Recover Spatial Information From Encounter Traces

This section is a detour from the study of coverage complexes to give an example of how persistent homology can be used to extract unexpected structure from the encounter data of mobile wireless networks [47].

The performance of wireless networks with mobile nodes is influenced by the mobility patterns of the nodes. In Mobile Ad-Hoc Networks (MANETs) which assume the existence of end-to-end paths and use synchronous protocols, the mobility of the nodes will affect the feasibility of certain routing algorithms. In Delay-Tolerant Networks (DTNs) where nodes are only intermittently connected, the mobility of the nodes is the medium used to move data around the network. In such networks understanding the nodes' mobility is crucial to being able to analyze network performance and design routing algorithms.

Unfortunately node mobility is fantastically complicated. Researchers have focused instead on the encounter patterns that the nodes' mobility produces. To this end there have been several experiments which tag people or animals with wireless motes that record which other motes they come into contact with and when. These experiments produce **encounter traces**; a series of data points, each consisting of

the encounter time and the IDs of the two nodes involved. Other data sets have also been repurposed for this type of analysis.

Some statistics that have been computed from encounter traces include distributions of node inter-contact times, distributions of node degree, and global encounter rates. These statistics focus on isolated local phenomena between individual pairs of nodes. Persistent homology allows us to tie many encounters together and extract more global structure from the encounter pattern. For example, in experiments where the nodes are restricted to a 1-dimensional space (a graph), the tools of persistent homology can sometimes recover the 1-dimensional topology of the space and detect certain changes in the dimensions of the space. We demonstrate that the same sort of technique can be used to deduce topological information when the nodes are moving in 2-dimensional space.

Efforts to gather mobile node encounter traces became popular in the mid 2000's as researchers realized the need to evaluate MANET and DTN routing algorithms against realistic patterns of node mobility. Experiments involving social

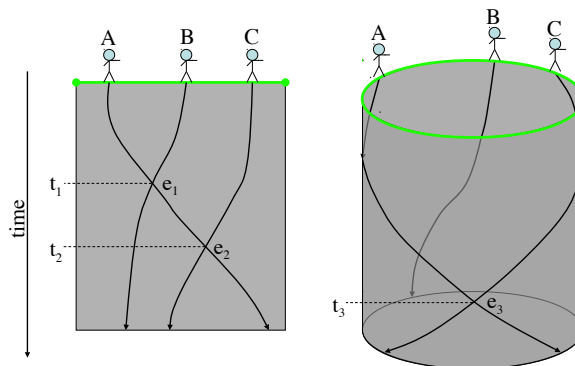


Figure 5.1: The topology of the space affects the type of encounter patterns that are possible. If the space is like a line, node A cannot encounter node C without one of them encountering node B. If the space is like a loop, node A and C can encounter each other without encountering node B.

associations go back to at least Milgram’s famous small-world experiment [54], but mobile wireless networks such as MANETs and DTNs require actual node proximity to transfer information, so the encounter patterns produced by people and animals have become a salient topic for engineers to study.

Encounter trace experiments include the famous Huggle project experiments [55]. Wireless LAN traces such the MIT trace [56], the UCSD trace [57], and the Dartmouth trace [58] are also commonly repurposed for use as encounter traces [59]. The initial Huggle project analysis focused on the distribution of node inter-encounter times. That is, given an encounter between a particular pair of nodes, how long before they come into contact again? The initial conclusion was that the distribution of inter-encounter times follows a power-law. This was an important discovery because most simulators produce, and most mathematical models assume, an exponential inter-encounter time distribution. Since then it has been observed that the experimental inter-encounter time distributions appear to undergo a transition from power-law to exponential on a timescale of about one half day [60], and the reasons behind the observed exponentially distributed inter-encounter times in simulations using random walk mobility have been thoroughly examined [61]. Other analyses have involved the uncensoring of trace data [62], and node degree distribution.

Several other researchers have analyzed such data in a social context, attempting to identify relationships and communities. For example the MIT Reality Mining Project analyzes encounter data in conjunction with information volunteered by test subjects to build models which recognize various social relationships between node

carriers [63]. In addition to developing and analyzing distributed community detection algorithms, [64] have demonstrated some vibrant tools for visualizing encounter traces.

We are not aware of any studies that attempt to recover spatial information from encounter traces. In wireless LAN traces the locations of the access points are fixed and are usually known to the researchers. Knowing that certain nodes are stationary and that encounters can only occur near the stationary nodes makes the topology reconstruction problem (but not the geographic reconstruction problem) trivial. In [65] the researchers do attempt to make deductions about the shapes of routes and the presence of popular paths between certain access points.

5.1 The Encounter Complex Metric

A crucial step in any application of persistent homology is to derive a filtered simplicial complex from the raw data that captures the desired structure. In this example we build a weighted graph on the set of encounters, and define a metric on that graph. This will be different from the euclidean metric associated with the physical space of the experiment.

We assume that we are given an encounter trace containing N data points of the form:

$$\begin{aligned} e_1 &= (t_1, \text{nodeA}_1, \text{nodeB}_1) \\ e_2 &= (t_2, \text{nodeA}_2, \text{nodeB}_2) \\ &\vdots \\ e_N &= (t_N, \text{nodeA}_N, \text{nodeB}_N) \end{aligned}$$

Here e_i represents the i th encounter, consisting of t_i , the time the encounter took place, and nodeA_i and nodeB_i , the two nodes involved in the encounter. We do not assume that any particular nodes are, or are not, stationary. We do not assume that the nodes all have the same speed, or that the speed of any particular node is constant, but we do assume that there is some maximum velocity v_{\max} .

We will construct a weighted graph, \mathcal{G} , in which the vertices correspond to the encounters $\{e_i\}_{i=1..N}$. Suppose two encounters, e_i and e_j , have a node in common. These encounters occurred at two particular points in space, \mathbf{x}_i and \mathbf{x}_j , respectively. We do not know \mathbf{x}_i or \mathbf{x}_j , but we can make the deduction:

$$|t_i - t_j| < T \implies |\mathbf{x}_i - \mathbf{x}_j| < T \cdot v_{\max} \quad (5.1)$$

Therefore, if two encounters, e_i and e_j have a node in common, we connect them with an edge with weight $|t_i - t_j|$.

This allows us to define a metric on the set of encounters

$$d(e_i, e_j) = \begin{cases} d_{\mathcal{G}}(e_i, e_j) & \text{if } e_i \text{ and } e_j \text{ are connected in } \mathcal{G} \\ \infty & \text{otherwise} \end{cases} \quad (5.2)$$

where $d_{\mathcal{G}}(e_i, e_j)$ is the minimum distance between the vertices e_i and e_j in the weighted graph \mathcal{G} . We call the Rips complex built on the vertices of \mathcal{G} with metric $d(\cdot, \cdot)$ an **encounter complex**.

Note that the reverse implication of (5.1) is not valid, even if the speed of the nodes is identical and constant. That is why a simplicial complex built based on this metric will be different from a Rips complex based on Euclidean distances, and why one might not expect persistent homology methods to work very well.

Because of this, it is better to think of the space we are studying as the physical space crossed with time. This idea is visualized in Figure 5.1. For example, if the nodes are moving in a space $X \subseteq \mathbb{R}^n$, then each encounter takes place at a point in the space $X \times \mathbb{R}^+ \subseteq \mathbb{R}^{n+1}$. More concretely, if the nodes live on a circle $X = S^1 \subset \mathbb{R}^2$ with coordinates (x, y) , then the space whose topology we hope to reconstruct is a cylinder with coordinates (x, y, t) . \mathbb{R}^+ is contractible; in fact X is a strong deformation retract of $X \times \mathbb{R}^+$. Therefore $X \times \mathbb{R}^+$ has the same homotopy type as X and the Betti numbers of the product space we are reconstructing will be the same as the Betti numbers of X .

A variation on our construction is to use the metric derived from the minimum number of hops through the encounter graph, instead of the metric derived from the

encounter complex. This is equivalent to restricting the connections in the encounter complex to only the two preceding and two succeeding encounters and setting all edge weights to 1.0. This technique seems to give more stable results for static one-dimensional spaces. However it fails badly with the two-dimensional experiments, and will be invariant under scaling of the size of the space and speed of the nodes. Therefore we focus only on the metric induced by the full encounter complex.

5.2 Building a Witness Complex

We now have a set of data points $\{e_i\}_{i=1..N}$ with a metric on them. There are a variety of ways to build a filtered simplicial complex from these data, but building the witness complex in the manner of de Silva and Carlsson [45] seems to give the best results. They compare several variations, so let us outline the particular method we used.

The first step to building a witness complex is to reduce the number of vertices by identifying a subset of the data points as **landmarks**. As in [45] we choose these landmarks using the maxmin algorithm.

- Choose the first landmark randomly
- Choose the next landmark from the remaining data points so as to maximize the minimum distance to all previously chosen landmarks. - Repeat.

The witness complex will be a simplicial complex whose 0-simplices are the set of landmarks.

The concept of the witness complex is based on the Delaunay triangulation. Given discrete set of points $A \subset \mathbb{R}^n$ in general position, the k -simplex $\sigma = [a_{i_0} a_{i_1} \cdots a_{i_k}]$ is part of the Delaunay triangulation of A iff there is a point $x \in \mathbb{R}^n$ such that x has $\{a_{i_0}, a_{i_1}, \dots, a_{i_k}\}$ among its nearest neighbors in A and $|x - a_{i_0}| = |x - a_{i_1}| = \dots = |x - a_{i_k}|$. de Silva calls such a point a **strong witness** for the simplex σ [44]. Given a finite data set, such as our set of encounters, the probability of finding strong witnesses is effectively zero. A **weak witnesses** for $[a_{j_0} a_{j_1} \cdots a_{j_{k'}}]$ is a point, x , which merely has $\{a_{j_0}, a_{j_1}, \dots, a_{j_{k'}}\}$ among its nearest neighbors. de Silva showed that it suffices to find a weak witness for every subsimplex $[a_{j_0} a_{j_1} \cdots a_{j_{k'}}] < [a_{i_0} a_{i_1} \cdots a_{i_k}]$ [44].

Let $E = \{e_i\}_{i=1..N}$ be the set of encounter data and let $L \subset E$ be the set of landmarks. For each $e_i \in E$, let m_i be the distance to the second closest landmark to e_i . Then for all pairs of 0-simplices (landmarks) $a, b \in L$, we add the 1-simplex $[a b]$ iff there exists an $e_i \in E$ such that

$$\max(d(e_i, a), d(e_i, b)) \leq m_i + T \tag{5.3}$$

for some threshold $T \geq 0$. Repeating this for increasing values of T gives a filtered simplicial 1-complex.

This procedure can be generalized to build witness complexes with higher dimensional simplices, but we only use it to build a 1-skeleton and then fill in all possible higher dimensional simplices up to the dimension of interest. This is commonly called the “lazy” witness complex.

5.3 Graph-Based Experiments

Our first experiments are simulations of nodes doing random walks on a graph. When a node arrives at a vertex it chooses its next destination randomly from among the adjacent vertices, and its velocity uniformly from the range $[v_{\min}, v_{\max}]$. The node departs immediately for its next destination. Encounters are recorded the moment two nodes pass each other on an edge or at a vertex. In our simulations all edges have the same length, but longer arcs can be created by attaching several edges in series. We built a simple event-driven simulator to generate these data.

As a practical matter our simulator also outputs the coordinates of each encounter as well as the time of the encounter and the IDs of the nodes involved. The physical coordinates are not used in the computation, but are indispensable for visualizing and validating the results. All figures showing simplicial complexes were plotted using the real coordinates, though the coordinates were not used to build the simplicial complex.

In all experiments in this section edges have length 1.0km, and the speed distribution is uniform with $v_{\min} = 0.3\text{km}/\text{min}$, $v_{\max} = 1.0\text{km}/\text{min}$. The units are completely arbitrary. For each experiment we took 1000 consecutive encounters from the middle of a longer encounter trace, built an encounter complex, and from that a filtered witness complex. We found that using maxmin to select 20% of the data points as landmarks gave good results, in agreement with [45].

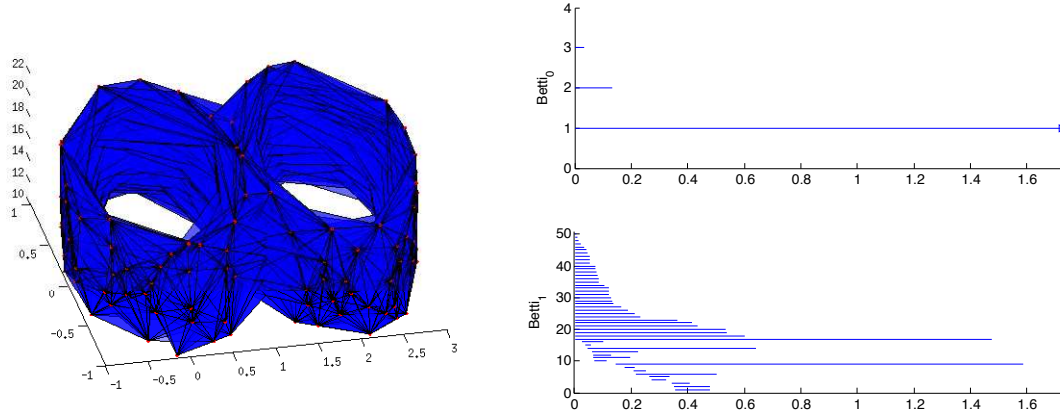


Figure 5.2: The witness complex and persistence barcode for an experiment in the two-loop graph space. The x and y coordinates in the witness complex plot correspond to the physical location of the encounters and are used to visualize the results, but are not used in the computation. The z-axis corresponds to the encounter time.

We compare three types of graph-based experiments.

- A linear graph
- A single loop
- A multi-loop

In the linear graph experiment 50 nodes followed discrete random walks on a graph consisting of 10 vertices connected by edges in a linear chain. Topologically this graph has one connected component and no higher-dimensional topological features, so $\beta_0 = 1$, and $\beta_k = 0$ for $k > 0$. Since the fully contractible topology is always results from setting the witness complex threshold high enough, it is difficult to quantify how well the method is working for this type of graph.

In the single loop experiment 50 nodes followed discrete random walks on a graph consisting of 10 vertices arranged on a circle and connected with edges to form a single loop. This graph also has one connected component, so $\beta_0 = 1$, but it

has a single non-trivial 1-cycle so $\beta_1 = 1$. Building a filtered witness complex and computing the persistent homology we recover the correct Betti numbers 100% of the time, though the results are not always as stable as we would like.

In the multi-loop experiments we find that adding 50 nodes per extra loop tends to generate enough encounters to reconstruct the spatial topology. The node mobility is the same as before. For the line, and one and two-loop experiments we chose 1000 consecutive data points to process. In the three-loop experiments the space was large enough that we had to use 1500 encounters to reliably cover the space. Each loop has 10 vertices arranged around a circle and one vertex is shared between adjacent loops. The correct Betti numbers are $\beta_0 = 1$ since there is one connected component, and $\beta_1 = l$, where l is the number of loops. The same technique as before correctly recovers this information for all two and three loop examples we tried.

Looking at a persistence diagram is a fairly qualitative method for evaluating our results. In [45] de Silva and Carlsson present and use some more objective performance metrics, and we replicate those as best we can. In some cases the large data sizes we need to use makes computation of exactly the same statistics intractable.

Let $R \in \mathbb{R}^+$ be the parameter enumerating the filtration. Define R_0, R_1 to be such that the correct Betti numbers are achieved for $R \in [R_0, R_1)$, and the correct Betti numbers are not achieved for $R < R_0$ or $R \geq R_1$. This is called the **success interval**. An experiment is considered a success if the success interval is non-empty. Define K_0 to be the point in the filtration where the Betti profile changes

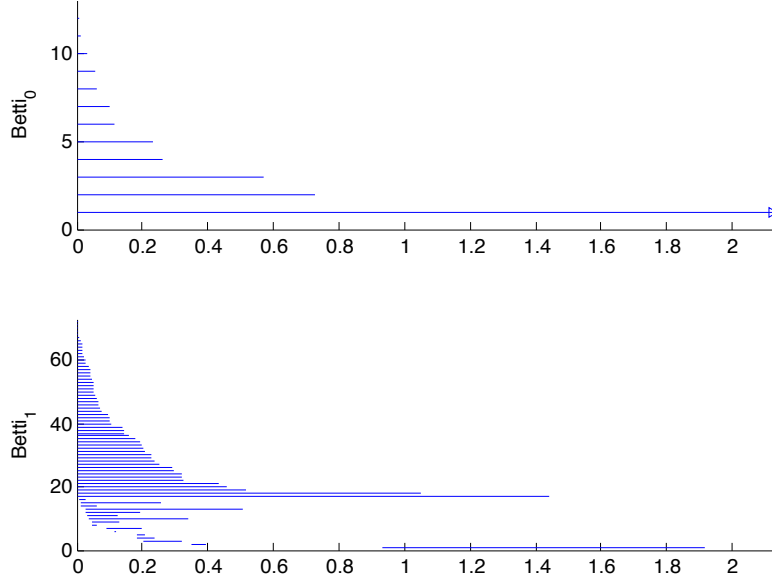


Figure 5.3: The persistence barcodes for an experiment in a three-loop space. The three persistent 1-cycles are quite prominent, but the intervals only overlap briefly, leading to a small “success interval”.

permanently to $\beta_0 = 1$, $\beta_1 = 0$. This is slightly simplified from the criterion used in [45]. Finally, define K_1 to be the point in the filtration where the witness complex becomes a single simplex on the landmark vertices (i.e. a complete graph).

The **relative dominance** is defined to be $(R_1 - R_0)/K_0$. This quantity measures the size of the success interval relative to the region over which the complex is homologically non-trivial. The **absolute dominance** is defined to be $(R_1 - R_0)/K_1$. This quantity measures the size of the success interval relative to the scale of the most distant data points. Both quantities will be between 0 (success interval is empty) and 1. Larger values indicate better performance.

Results for the one, two, and three loop experiments are given in the table.

Number of loops	1	2	3
med. rel. dominance	0.4208	0.4946	0.4045
med. abs. dominance	0.1096	0.1251	0.0683
med. num. cells	203384	182304	357836

The median relative dominance results are fairly consistent as we vary the number of loops in the graph. Since each of the loops is the same size and has the same probability of being the site of an encounter this is what we should expect. The relative dominance measures to what extent the correct homology stands out from the noise in the persistence diagram. Our results are about half that reported for the sphere experiments in [45]. This means that our topological features stand out quite a bit less, despite the fact that we are studying a simpler structure and using much more data. This highlights how difficult things become when the metric on the data is only probabilistically related to the metric in the underlying physical space.

The absolute and relative dominance are reasonable statistics to evaluate our results, but there are some situations where they are overly pessimistic. Consider for example Figure 5.3. This is a persistence barcode from a three-loop experiment. Most people would visually pick out the three long-lived features in the $Betti_1$ plot and conclude that there are three fairly persistent 1-cycles in the space. But because the persistence intervals only overlap briefly the success interval is small and the relative and absolute dominance appear very poor. It seems that sometimes there is no substitute for manually examining the results.

5.3.1 A Note on Higher Degree Vertices

In a graph, an **essential vertex** is a vertex of degree three or more. In the first two examples, the graph the nodes traveled on had no essential vertices. In such a graph there is no way for the nodes to change their relative arrangement without encountering each other. On a multi-loop, such as a figure-eight, there are vertices of degree four, making it possible for the nodes to effectively pass each other without ever having an encounter. In fact, Abrams has shown that in a connected graph with an essential vertex there is always a way to move the nodes from any physical arrangement to any other physical arrangement without any encounters [66]. It is reasonable to worry that essential vertices will cause holes in the simplicial complex which the filtration cannot fill. Though it is theoretically possible, the random mobility of the nodes makes problems like this extremely unlikely.

5.3.2 Dynamic Graphs

We have found that persistent homology can detect certain types of changes in a space. For a first simple example we look at a single loop that starts out small, enlarges to a certain extent, then shrinks down again. Assuming the time for a node to circumnavigate the shrunken loop is comparable to the times between encounters on the expanded loop, the shrunken loop will appear contractible. The persistent homology of the encounter complex should approximate that of a sphere. That is, $\beta_0 = 1$, $\beta_1 = 0$, and $\beta_2 = 1$.

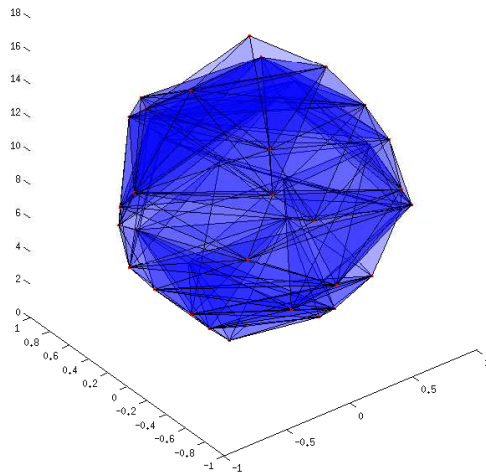


Figure 5.4: The witness complex obtained from for the expanding/contracting loop experiment. We were able to recover the homology of a sphere, demonstrating that persistent homology can detect certain types of changes in the physical space.

We performed this experiment using the same simulator as before. The graph that the nodes traveled on was a single loop with 8 vertices arranged in a circle. At time t_{start} the graph was scaled down by a factor of 10 relative to the regular size, while the node velocities were the same as usual. Then the graph was scaled up at a constant rate, reaching normal size (edge length 1) at time t_{mid} . Finally the graph was scaled back down to the initial size at the opposite rate that it was expanded ending at t_{end} .

The persistent homology algorithm does recover the correct Betti numbers. Figure 5.4 shows the witness complex recovered from this experiment. Since we were attempting to recover 2nd homology information it was necessary to fill in 3-simplices in the witness complex. The number of simplices in a complex tends

to increase rapidly with dimension, and that was the case here. We had 56 0-cells, 1074 1-cells, 11703 2-cells and 86568 3-cells.

Though our experiment changed the size of the space, one could equivalently change the speed of the nodes. Since changing speeds seems more realistic than changing the physical space in most situations, perhaps we should point out that persistent homology would detect that phenomenon in exactly the same way. We use the changing space example here because it is easier to understand visually.

5.4 Two-Dimensional Experiments

The dynamic graph experiment was rather contrived, and restricting the nodes to a one dimensional space like a graph is somewhat unrealistic. It is important to see if we can recover topological information from a more general space. This experiment will show that it is possible to recover topological information and detect certain spatial or behavioral changes from an encounter trace generated by nodes moving randomly in a two dimensional grid.

5.4.1 Detecting Changes in a 2D Space

We performed an experiment in which 50 nodes performed discrete random walks on a 2-dimensional grid with reflecting walls. After 5000 simulation steps the nodes' mobility model is changed so that they are repelled by the center of the grid. This causes them to congregate near the boundary. Then, after another 5000 steps the repulsive force is removed and the nodes randomly fill up the grid again.

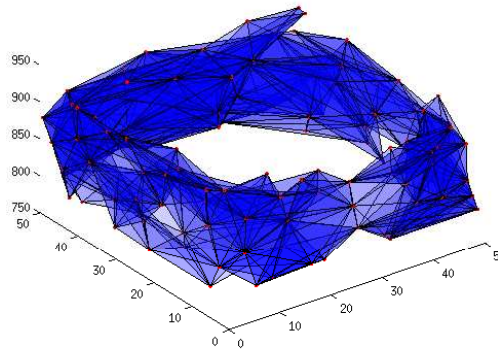


Figure 5.5: The witness complex obtained for the repulsion phase of the 2D random walk experiment. The number of simplices made it intractable to compute the homology of the entire data set at once.

One would expect the encounter complex during the random walk phase of the experiment to have no real topological features. On the other hand, the encounters during the middle phase of the experiment, when the nodes are repelled by the center of the grid, should have the homology of a loop. Amazingly we were able to recover exactly these features for both phases. Combining all three phases into one computation should yield non-trivial 2nd homology for the same reasons as the expanding/contracting loop experiment. Unfortunately the number of simplices involved made the combined computation intractable to do with the MATLAB modules.

5.4.2 2D Grid With Boundaries vs. a Torus

Network simulations are commonly done in finite simulated spaces with synthetic mobility models. The same is commonly assumed in mathematical models of wireless networks. One way to avoid the complications and non-uniformity caused

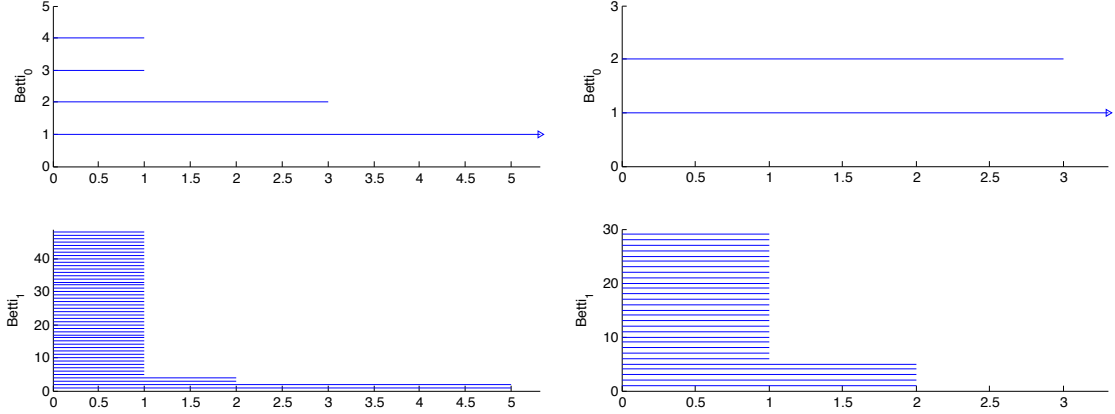


Figure 5.6: The persistence barcodes for the torus (left) and bounded rectangle (right) experiments. The torus experiment shows two prominent 1-cycles, in agreement with the expected topology of a toroidal space. In the bounded rectangle results no 1-cycles particularly stand out from the noise.

by the edges of the finite space is to assume that opposite edges wrap around, thereby eliminating the boundary of the space. Topologically a rectangular area with opposite edges identified is a torus, homeomorphic to $S^1 \times S^1$.

It is not surprising that the encounter patterns of nodes on a torus would somehow be different from the encounter patterns of nodes in a bounded domain, but it is difficult to say precisely how they differ. We have found that the encounter patterns in these two types of spaces are different in a way that can be detected by the persistent homology of the encounter complex. Specifically, we can differentiate the two experimentally by their first persistent Betti numbers. For the torus we have $\beta_1 = 2$ but for a bounded rectangular region we have $\beta_1 = 0$. Presumably we could use the second persistent Betti numbers also, but we have found that filling in 3-simplices is too computationally intensive. In the toroidal case the object we are reconstructing is $S^1 \times S^1 \times \mathbb{R}^+$, which cannot be visualized in the same manner as the previous spaces.

In this experiment we used the same discrete-time grid-based simulator as above. We let 1000 nodes do discrete random walks on a 50×50 grid for 500 simulation steps. Two nodes registered an encounter if they coincided on the same grid cell at the same time. In the torus experiment a node could walk off the edge of the grid and appear on the opposite edge. In the bounded rectangle experiment nodes were not allowed to go past the edge of the grid.

Since we used such a large number of nodes it was necessary to include a large number of encounters in the persistence computation to keep from having several disconnected components. In these experiments we used 5000 encounter data points. The torus experiment took longer to compute since it had more connections than the bounded rectangle.

Number of k -simplices	$k = 0$	$k = 1$	$k = 2$
bounded rectangle	320	11766	196682
torus	303	15232	297375

We chose the landmarks for these experiments according to the same process as before, but instead of choosing 20% of the data points as landmarks, we chose landmarks until we could not choose another one without it being within distance 10.0 of a previously chosen landmark. Setting a threshold like that requires some familiarity with the scale of the data, but seems to produce more efficient results.

5.5 Experiments With Real Encounter Data

The Huggle Project encounter data includes data from three experiments. We applied the same methods used in the previous sections to analyze the data from the Cambridge Computer Lab experiment.

According to the documentation, the experiment was conducted over seven days in January 2005 at the University of Cambridge Computer Lab. Nineteen iMotes were carried by graduate students from the System Research Group. Only twelve of the mobile motes yielded usable data, and an additional 210 external Bluetooth devices appeared in the traces. For our analysis we filtered out encounters with external devices and sorted the data by encounter time. We trusted that the iMote clocks were sufficiently well synchronized that this sorting made sense.

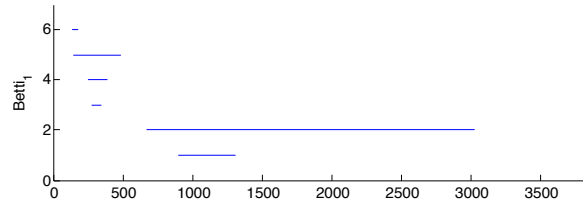
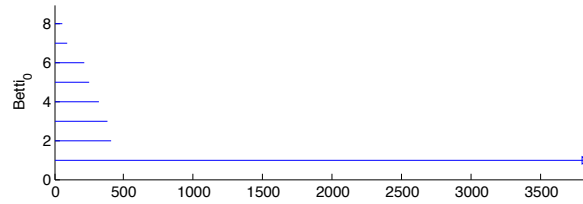
Plotting a histogram of encounter frequencies one can identify the separate days in the experiment. We extracted three data sets corresponding to the three busiest days, and excluded the off-hours during which very few encounters took place. Since the experimenters included the duration of an encounter, we added a filler encounter data point every 500 seconds during long encounters. Since we did not know much about the scale of the data we used the reliable method of selecting 20% of the data points as landmarks.

We computed persistent Betti numbers for dimensions 0 and 1 for each day's data individually. The persistence diagrams for the three days are shown in figure 5.7. We observed a single fairly persistent 1-cycle on the scale of 50-65 minutes in the results for each day. The cycle was particularly prominent on day 1, but similar

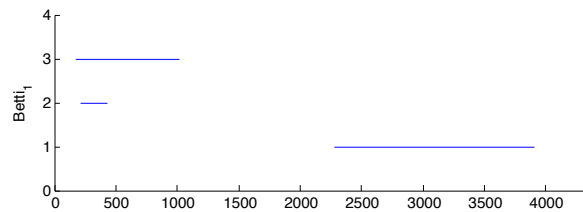
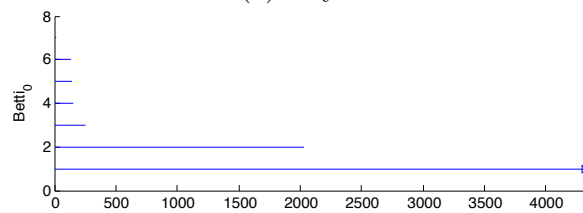
features appear in days 2 and 3. Comparing the persistence diagrams for β_0 and β_1 for days 2 and 3, it appears that two disconnected components merge to form the prominent 1-cycle.

In this case we doubt that the observed features are due to the space the experiment was conducted in, though we cannot rule it out either. Based on our simulation experiments, we expect it would require many more mobile nodes to reliably identify a spatial cycle. The most likely explanation we have for these results is some sort of scheduling. For example a group of mote carriers may encounter another group at a meeting or class, then another encounter may take place at lunch. Unfortunately not enough is known about the particular experiment to draw a definitive conclusion. We believe it would be worth performing a similar experiment which records more details, and is perhaps more tightly controlled in order to see how different behaviors effect the persistent homology results.

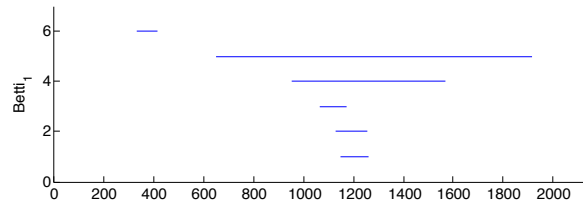
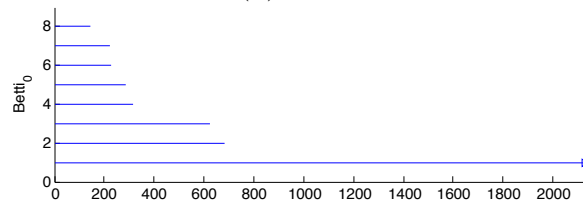
These results show that topological analysis methods, such as persistent homology, can find structure in real encounter data that may not have been accessible via traditional statistical methods.



(a) Day 1



(b) Day 2



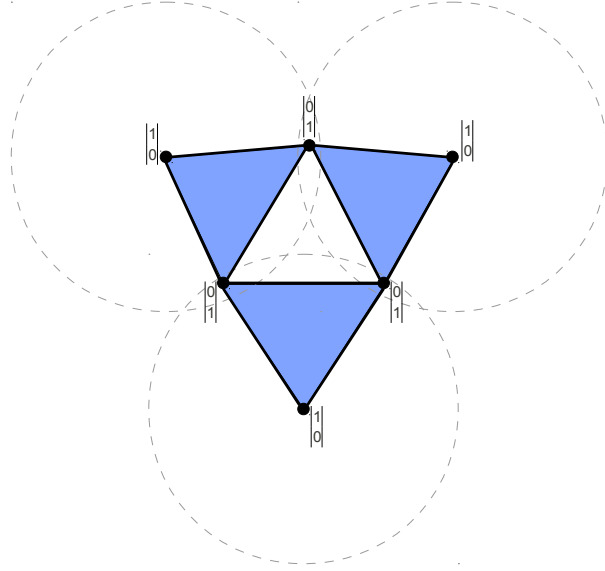
(c) Day 3

Figure 5.7: The persistence barcodes based on the filtered encounter traces from three days of the Haggie Cambridge Computer Lab experiment. In each day's data we observe a fairly persistent 1-cycle on the scale of 50-65 minutes. We cannot conclusively explain these results, but they do demonstrate how persistent homology can reveal topological structure in real encounter trace data.

Chapter 6: Patching False Holes in the Coverage Complex

Theorem 3.15 and corollary 3.16 already gave us a reasonable criterion for testing network coded information coverage. We know that because of the geometric defects of the Rips complex that this criterion cannot be bi-directional; if the criterion holds it guarantees coverage, but it can fail even when coverage is complete. In the case of coverage complexes, we can actually have false negatives for other reasons. Here we will give some examples, and attempt to refine the coverage criteria to discount these false holes.

One first defect we note, \mathcal{R}^C is not a Rips complex. It is not even a flag complex, as the example below demonstrates. This example shows $\mathcal{R}^C(V)$ for a six-point data arrangement over $V = \mathbf{GF}(2)^2$. Each of the three outer 2-simplices has access to a full basis for V , but the inner 2-simplex only has three copies of $\binom{0}{1}$. Therefore the central 2-simplex is part of $\mathcal{R}^C(\binom{0}{1})$ but not in $\mathcal{R}^C(V)$, even though its 1-skeleton is.

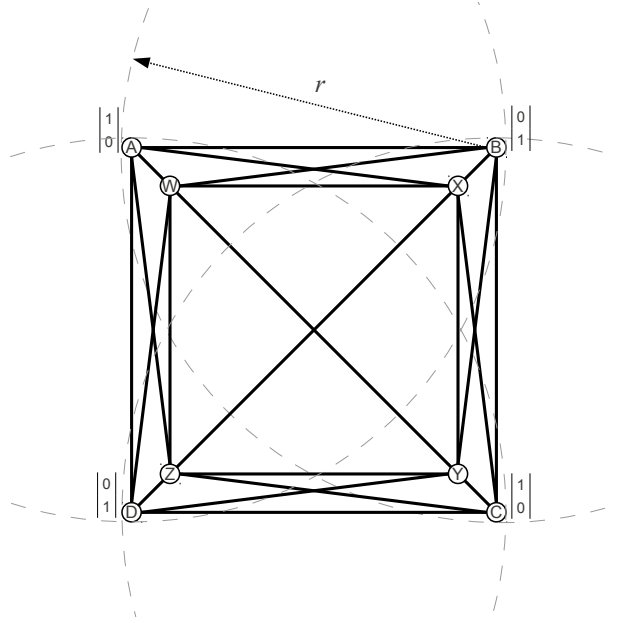


6.1 Bubble Boxes and the Mayer-Vietoris Sequence

Consider the eight-point data arrangement over $V = \text{GF}(2)^2$ on the points shown below with their Rips 1-skeleton. Points A, B, C, D are placed at the corners of a square of side length r . Points W, X, Y, Z are placed at the corners of a slightly smaller concentric square of side length r' with $1 > \frac{r'}{r} > (\sqrt{2} - 1)$. This ensures that

$$\frac{r - r'}{\sqrt{2}} + r'\sqrt{2} = \frac{r + r'}{\sqrt{2}} > r \quad (6.1)$$

or essentially that the distance from a corner of the outer box to the opposite corner of the inner box is more than r . As a result, the diagonal edges $[A Y], [B Z], [C W], [D X]$ are not in the Rips complex $\mathcal{R}_r(\mathcal{X})$. As for the data arrangement, only the points on the outer square have inventories. The points at opposite corners A, C hold $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and points B, D hold $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$.



The resulting Rips complex is the flag complex on this 1-skeleton. The coverage complex is more involved. The 3-simplices $[A B W X]$, $[B C X Y]$, $[C D Y Z]$, and $[D A Z W]$ all contain a full basis for V , and hence are in $\mathcal{R}^C(V)$. On the other hand, the 3-simplices $[A W X Z]$ and $[C Y Z X]$ are maximal but only have a single basis vector. Therefore they are in $\mathcal{R}^C(\binom{1}{0})$ but not in $\mathcal{R}^C(\binom{0}{1})$. Likewise, the 3-simplices $[B X Y W]$ and $[D Z W Y]$ are in $\mathcal{R}^C(\binom{0}{1})$ but not $\mathcal{R}^C(\binom{1}{0})$. As $\mathcal{R}^C(V) = \mathcal{R}^C(\langle \binom{1}{0}, \binom{0}{1} \rangle) = \mathcal{R}^C(\binom{1}{0}) \cap \mathcal{R}^C(\binom{0}{1})$, none of the simplices covering the inner square is in $\mathcal{R}^C(V)$.

$\mathcal{R}^C(\binom{0}{0}) = \mathcal{R}$	$\mathcal{R}^C(\binom{1}{0})$	$\mathcal{R}^C(\binom{0}{1})$	$\mathcal{R}^C(V) = \mathcal{R}^C(\binom{1}{1})$

The dilemma here is that the criteria in theorem 3.15 and corollary 3.16 only test $\mathcal{R}^C(V)$, and therefore detect a large hole in the information coverage. However the interior of the small square actually *is* covered; each point in the shadow is in range of points holding a full basis. They just aren't the same points across the entire area. More specifically, each point in the shadow $\mathcal{S} = p(\mathcal{R})$ is covered by vector $\vec{v}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, and is also covered by vector $\vec{v}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ (by cor 3.16). Since $\langle \vec{v}_1, \vec{v}_2 \rangle = V$ a more robust coverage criterion would take this into account.

In the following discussion let \mathcal{X} be a finite planar data arrangement over $V = \mathbb{F}^D$, and let $U \subseteq V$ be a linear subspace. Let $r > 0$ be some finite radius, and let $\mathcal{R} = \mathcal{R}_r(\mathcal{X})$ be the corresponding Rips complex and $\mathcal{S} = p(\mathcal{R})$ the Rips shadow.

Lemma 6.1. *Suppose there are subspaces $W_1, \dots, W_k \subseteq V$ such that $U \subseteq \langle W_1, \dots, W_k \rangle$ and \mathcal{S} is covered by W_i for $i = 1, \dots, k$. Then \mathcal{S} is covered by U .*

Proof. We have assumed that \mathcal{S} is covered by each of the W_i . Therefore for any point $x \in \mathcal{S}$ there are vectors $\vec{v}_1^i, \dots, \vec{v}_{n_i}^i \in \mathcal{M}_r(x)$ such that $W_i \subseteq \langle \vec{v}_1^i, \dots, \vec{v}_{n_i}^i \rangle$ for each W_i . Therefore

$$\begin{aligned} U &\subseteq \langle W_1, \dots, W_k \rangle \\ &\subseteq \langle \vec{v}_1^1, \dots, \vec{v}_{n_1}^1, \dots, \vec{v}_1^k, \dots, \vec{v}_{n_k}^k \rangle \\ &\subseteq \langle \mathcal{M}_r(x) \rangle \end{aligned}$$

□

This offers a refinement of our coverage criterion. Instead of simply testing $\mathcal{R}^C(V)$, now it suffices to find a basis $\{\vec{v}^1, \dots, \vec{v}_D\}$ for V such that the $\mathcal{R}^C(\vec{v}_i)$ all satisfy one of the coverage criteria. But as we can see in the example above, not just any basis will work for this test. For example $\left\{\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}\right\}$ is a basis for V , but does not guarantee coverage, since $\mathcal{R}^C\left(\begin{pmatrix} 1 \\ 1 \end{pmatrix}\right)$ has a defect.

This structure of the coverage complexes of the bubble box can be investigated from the perspective of their Mayer-Vietoris Sequence.

$$\begin{aligned} \dots \rightarrow H_2(\mathcal{R}^C(\vec{v}_1)) \oplus H_2(\mathcal{R}^C(\vec{v}_2)) &\rightarrow H_2(\mathcal{R}^C(\vec{v}_1) \cup \mathcal{R}^C(\vec{v}_2)) \\ &\xrightarrow{\delta_*} H_1(\mathcal{R}^C(\vec{v}_1) \cap \mathcal{R}^C(\vec{v}_2)) \rightarrow H_1(\mathcal{R}^C(\vec{v}_1)) \oplus H_1(\mathcal{R}^C(\vec{v}_2)) \rightarrow \dots \end{aligned}$$

In this case $\mathcal{R}^C(\vec{v}_1)$ and $\mathcal{R}^C(\vec{v}_2)$ intersect on the four 3-simplices around the edge of the square. Each coverage complex also contains two 2-simplices covering the center of the box, but not the same 2-simplices. The Rips complex \mathcal{R} actually contains the central 3-simplex $[W \ X \ Y \ Z]$, but neither of the coverage complexes do. Therefore $\mathcal{R}^C(\vec{v}_1) \cup \mathcal{R}^C(\vec{v}_2)$ contains all the faces of $[W \ X \ Y \ Z]$, but not the 3-simplex itself, giving a generator in $H_2(\mathcal{R}^C(\vec{v}_1) \cup \mathcal{R}^C(\vec{v}_2))$. This generator maps to the generator of the non-trivial 1-cycle in $H_1(\mathcal{R}^C(\vec{v}_1) \cap \mathcal{R}^C(\vec{v}_2))$. This is expected, since both $\mathcal{R}^C(\vec{v}_1)$ and $\mathcal{R}^C(\vec{v}_2)$ are actually contractible, $H_2(\mathcal{R}^C(\vec{v}_1)) = H_2(\mathcal{R}^C(\vec{v}_2)) = 0$ and $H_1(\mathcal{R}^C(\vec{v}_1)) = H_1(\mathcal{R}^C(\vec{v}_2)) = 0$. Therefore, using the fact that

$\mathcal{R}^C(\vec{v}_1) \cap \mathcal{R}^C(\vec{v}_2) = \mathcal{R}^C(V)$, the sequence becomes

$$0 \rightarrow H_2(\mathcal{R}^C(\vec{v}_1) \cup \mathcal{R}^C(\vec{v}_2)) \xrightarrow{\partial_*} H_1(\mathcal{R}^C(V)) \rightarrow 0$$

meaning that $H_2(\mathcal{R}^C(\vec{v}_1) \cup \mathcal{R}^C(\vec{v}_2)) \cong H_1(\mathcal{R}^C(\vec{v}_1) \cap \mathcal{R}^C(\vec{v}_2))$ via the connecting homomorphism.

In this case we see that elements of $H_1(\mathcal{R}^C(V))$ that come from $H_2(\mathcal{R}^C(\vec{v}_1) \cup \mathcal{R}^C(\vec{v}_2))$ via the connecting homomorphism don't actually correspond to holes in coverage. This suggests a coverage criterion along the lines of:

$$H_1(\mathcal{R}^C(V))/\partial_*(H_2(\mathcal{R}^C(\vec{v}_1) \cup \mathcal{R}^C(\vec{v}_2))) = 0 \Rightarrow \text{coverage} \quad (6.2)$$

The following lemma is written for the case where \mathcal{S} has no internal boundaries.

Lemma 6.2. *Let $W_1, W_2 \subseteq U \subseteq V$ be linear subspaces such that $U \subseteq \langle W_1, W_2 \rangle$, and suppose $\mathcal{F} \subseteq \mathcal{R}^C(U)$ is a fence subcomplex for \mathcal{S} with $p(\mathcal{F}) \simeq S^1$.*

If $\delta_ : H_2(\mathcal{R}^C(W_1) \cup \mathcal{R}^C(W_2)) \rightarrow H_1(\mathcal{R}^C(U))$ is surjective, then \mathcal{S} is covered by U .*

Proof. We assumed $\mathcal{F} \subseteq \mathcal{R}^C(U)$, so by the poset reversal property of \mathcal{R}^C , $\mathcal{F} \subseteq \mathcal{R}^C(W_1)$ and $\mathcal{F} \subseteq \mathcal{R}^C(W_2)$. Therefore the 1-cycle γ that gives the generator $[\gamma] \in H_1(\mathcal{F})$ also belongs to a class $[\gamma] \in H_1(\mathcal{R}^C(W_1))$ and $[\gamma] \in H_1(\mathcal{R}^C(W_2))$.

Consider the relevant part of the Mayer-Vietoris sequence.

$$\begin{aligned}
H_2(\mathcal{R}^C(W_1) \cup \mathcal{R}^C(W_2)) &\xrightarrow{\delta_*} H_1(\mathcal{R}^C(U)) \xrightarrow{(i_*, j_*)} H_1(\mathcal{R}^C(W_1)) \oplus H_1(\mathcal{R}^C(W_2)) \\
[\gamma] &\longmapsto i_*[\gamma] \oplus j_*[\gamma]
\end{aligned}$$

If δ_* is surjective, the inclusion (i_*, j_*) into the direct sum must be the zero map. Since the element $([\gamma], [\gamma])$ is in the image of the inclusion it must be zero. Thus both $\mathcal{R}^C(W_1)$ and $\mathcal{R}^C(W_2)$ contain $[\gamma] = 0$. Therefore by corollary 3.18 \mathcal{S} is covered by both W_1 and W_2 and by lemma 6.1 \mathcal{S} is covered by U . \square

This application of the Mayer-Vietoris sequence is interesting for the purpose of investigating the structure of the coverage complex, but it does not seem to help with coverage testing. It seems unlikely one would start with knowledge about the surjectivity of the Mayer-Vietoris connecting homomorphism in order to draw conclusions about whether or not other coverage complexes have a contractible fence subcomplex. It is possible this type of result could be useful for narrowing down the nature of a coverage defect, but we do not have anything presentable on that problem yet.

6.2 Local Bases and Local Coverage

Consider the data arrangement over $V = \mathbf{GF}(2)^2$ shown in figure 6.1. The simplicial complex contains three bubble boxes, each covered by one of the three possible bases for V . Therefore for any $\vec{v} \in V$, the interior of one of the boxes will be missing from $\mathcal{R}^C(\vec{v})$ as shown in figure 6.2.

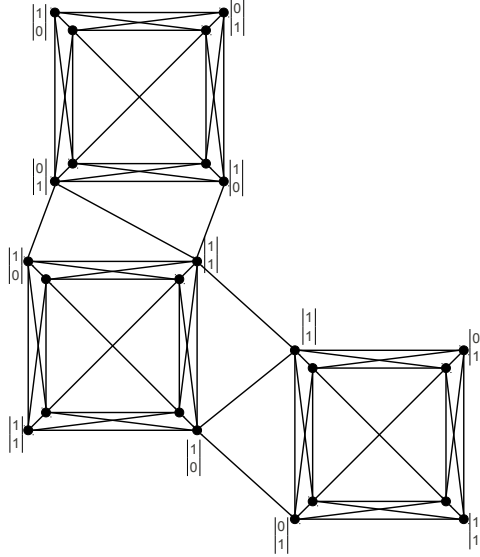


Figure 6.1: A data arrangement containing three bubble boxes, each with a different basis. The Rips shadow is completely covered by $V = \text{GF}(2)^2$, but for any $\vec{v} \in V$, $\mathcal{R}^C(\vec{v})$ will have a hole somewhere.

One way to recognize coverage in situations like this is to look at the coverage complex more locally. In this example we would like to verify coverage for each bubble box individually. Ideally we would break up \mathcal{S} into subsets $\mathcal{S}_1, \dots, \mathcal{S}_n$ such that $\bigcup_{i=1}^n \mathcal{S}_i = \mathcal{S}$ and each \mathcal{S}_i is covered by U . We are challenged by the fact that it is not obvious how to select a portion of a Rips complex that corresponds to a such a sub-shadow $\mathcal{S}_i \subseteq \mathcal{S}$. In fact, even selecting a 1-cycle in the Rips complex such that its shadow is simple is problematic.

Using the coverage criteria in chapter 3 it was important that the outer boundary of the shadow, $\partial^+ \mathcal{S}$, be a simple curve. In chapter 4 we loosened this assumption, but still relied on the fact that if $\partial^+ \mathcal{S}$ contains more than one cycle, that the intersection point must be a Rips vertex. For arbitrary 1-cycles in $\mathcal{R}^C(U)$, the shadow of a cycle may cross transversally in the interior of two edges. If we try to break up \mathcal{S} into sub-domains whose boundaries are the shadows of such cycles, the correspon-

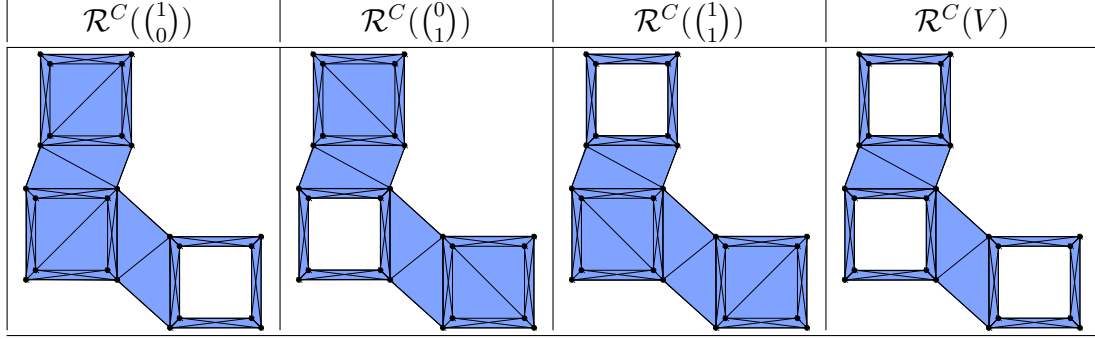


Figure 6.2: Even though it is completely covered by V , for each possible vector in V , the interior of one of the boxes is missing from the coverage complex. $\mathcal{R}^C(V)$, therefore, has three holes.

dence between the cycle in the Rips complex and the boundary of the sub-domain becomes much more complicated. Our solution has to be slightly more indirect.

The following is almost the same as Corollary 4.1 in [21], but different in that we use an arbitrary subcomplex of the Rips complex, and they consider the Rips complex on a subset of the points.

Lemma 6.3. *Let $\Sigma \subseteq \mathcal{R}$ be a subcomplex and $\mathcal{F} = \mathcal{F}^+ \sqcup \mathcal{F}^- \subseteq \Sigma$ a fence subcomplex. For any 2-chain $\alpha \in C_2(\Sigma)$ satisfying theorem 3.19 let Σ_α be the closure of the collection of 2-simplices with non-zero coefficients in α . Then $p(\Sigma_\alpha) = \mathcal{S}$.*

Proof. Suppose there is a point $x \in \mathcal{S} \setminus p(\Sigma_\alpha)$. Then we can factor the projection map in the commutative diagram:

$$\begin{array}{ccc}
 H_2(\Sigma_\alpha, \mathcal{F}) & \xrightarrow{\delta_*} & H_1(\mathcal{F}, \mathcal{F}^-) & (6.3) \\
 \swarrow p_* & \downarrow p_* & \downarrow p_*, \cong & \\
 H_2(\mathbb{R}^2 - x, \partial\mathcal{S}) & & & \\
 \searrow i_* & \downarrow p_* & H_2(\mathbb{R}^2, \partial\mathcal{S}) & \xrightarrow{\delta_*} & H_1(\partial\mathcal{S}, \partial^- \mathcal{S})
 \end{array}$$

Then we can repeat the argument in the proof of theorem 3.19 to obtain a contradiction. \square

Lemma 6.4. *Assume we have a subcomplex $\Sigma \subseteq \mathcal{R}$, a fence subcomplex $\mathcal{F} = \mathcal{F}^+ \sqcup \mathcal{F}^- \subseteq \Sigma$ with $\mathcal{F}^+ \cong S^1$ and $[\gamma] \in H_1(\mathcal{F}^+)$ satisfying corollary 3.20, and a collection of 1-dimensional subcomplexes $\mathcal{F}_1, \dots, \mathcal{F}_n \subseteq \Sigma$ such that:*

- $\mathcal{F}_i \cong S^1$
- $\exists \gamma_i \in C_1(\mathcal{F}_i) \forall i$ such that $\partial \gamma_i = 0$ and $\sum \gamma_i = \gamma$

For any $\alpha \in C_2(\Sigma)$ let Σ_α be the closure of the collection of 2-simplices with non-zero coefficients in α . Define:

$$\mathcal{S}_i = \bigcap_{\substack{\alpha \in C_2(\Sigma) \\ \partial \alpha = \gamma_i}} p(\Sigma_\alpha) \quad (6.4)$$

If $\forall i \mathcal{S}_i \neq \emptyset$ then $\bigcup \mathcal{S}_i = \mathcal{S}$.

Proof. By our assumption that $\forall i \mathcal{S}_i \neq \emptyset$, there is at least one choice of 2-chains $\{\alpha_i\}_{i=1..n}$, $\alpha_i \in C_2(\Sigma)$, such that $\partial \alpha_i = \gamma_i$.

Since there are only finitely many simplices in Σ , and because the 2-chains α_i can be chosen completely independently, if there is a point $x \in \mathcal{S} \setminus \bigcup \mathcal{S}_i$ then there must be some particular choice of 2-chains $\{\alpha_i\}_{i=1..n}$, $\alpha_i \in C_2(\Sigma)$, such that $\partial \alpha_i = \gamma_i$ such that $x \notin \bigcup p(\Sigma_{\alpha_i})$.

But for any such choice of 2-chains, $\partial(\sum \alpha_i) = \sum(\partial \alpha_i) = \gamma$ which is non-zero on \mathcal{F}^+ . Let $\bar{\alpha} = \sum \alpha_i$. Then by lemma 6.3 $\mathcal{S} \subseteq p(\Sigma_{\bar{\alpha}})$.

So there can be no $x \in \mathcal{S} \setminus \bigcup \mathcal{S}_i$, and we conclude $\bigcup \mathcal{S}_i = \mathcal{S}$. \square

This means that we can show that the shadow \mathcal{S} is covered by U by showing that each \mathcal{S}_i is covered by U .

Theorem 6.5. *Let \mathcal{X} be a finite planar data arrangement over V with Rips complex \mathcal{R} and Rips shadow \mathcal{S} , and suppose for $U \subseteq V$ there is a fence subcomplex $\mathcal{F} \subseteq \mathcal{R}^C(U)$ with $\mathcal{F}^+ \simeq S^1$ and generator $[\gamma] \in H_1(\mathcal{F}^+)$.*

Assume we have 1-dimensional subcomplexes $\mathcal{F}_1, \dots, \mathcal{F}_n \subseteq \mathcal{R}^C(U)$ such that:

- $\mathcal{F}_i \cong S^1$
- $\exists \gamma_i \in C_1(\mathcal{F}_i) \forall i$ such that $\partial \gamma_i = 0$ and $\sum \gamma_i = \gamma$
- for each \mathcal{F}_i there exists a basis $\mathcal{B}^i = \{\vec{b}_1^i, \dots, \vec{b}_{m_i}^i\}$ such that $U \subseteq \langle \mathcal{B}^i \rangle$ and $[\gamma_i] = 0$ in $H_1(\mathcal{R}^C(\vec{b}_j^i)) \forall j = 1 \dots m_i$.

Then \mathcal{S} is covered by U .

Proof. For any 2-chain $\alpha \in C_2(\mathcal{R})$ let Σ_α be the closure of the collection of 2-simplices with non-zero coefficients in α . For each γ_i define

$$\mathcal{S}_i = \bigcap_{\substack{\alpha \in C_2(\mathcal{R}) \\ \partial \alpha = \gamma_i}} p(\Sigma_\alpha)$$

We assumed that $[\gamma_i] = 0$ in $H_1(\mathcal{R}^C(\vec{b}_j^i))$ for some $\vec{b}_j^i \in V$, so therefore $[\gamma_i] = 0$ in $H_1(\mathcal{R})$. Therefore the \mathcal{S}_i are non-empty, and by lemma 6.4 $\mathcal{S} = \bigcup \mathcal{S}_i$. That means every point $x \in \mathcal{S}$, is in \mathcal{S}_i for some i .

Suppose $x \in \mathcal{S}_i$. By our assumption there is a basis $\mathcal{B}^i = \{\vec{b}_1^i, \dots, \vec{b}_{m_i}^i\}$ spanning U with $[\gamma_i] = 0$ in $H_1(\mathcal{R}^C(\vec{b}_j^i)) \forall j = 1 \dots m_i$. So for each j there is a

2-chain $\alpha_{i,j} \in C_2(\mathcal{R}^C(\vec{b}_j^i))$ such that $\partial\alpha_{i,j} = \gamma_i$. By construction $\mathcal{S}_i \subseteq p(\Sigma_{\alpha_{i,j}})$, so by lemma 3.11 \mathcal{S}_i is covered by \vec{b}_j^i . This is true for each $\vec{b}_j^i \in \mathcal{B}^i$ (possibly with different $\alpha_{i,j}$), so we conclude that \mathcal{S}_i , and therefore x , is covered by U . \square

This allows us to test for coverage when different regions of the complex require different bases to successfully test for coverage. Note that the way it is stated here, we need $\mathcal{F}_i \subseteq \mathcal{R}^C(U)$ for every \mathcal{F}_i . It *may* be true under weaker assumptions that allow different $\mathcal{F}_{i,j}$ for each $\vec{v}_{i,j}$, but it probably doesn't matter. We can't come up with an example where you would need different dividing cycles for the different vectors in the basis.

6.3 Bubble Baubles and Multi-Fold Intersections

The local coverage criterion of theorem 6.5 is geometrically intuitive, but it doesn't give any help on how to choose the cycles $\alpha_1, \dots, \alpha_n \in C_1(\mathcal{R}^C(U))$ or the bases for U that cover the respective regions. It would be illuminating to have a more general algebraic statement for when a hole is a false hole similar to the relationship we saw for two-fold covers using the Mayer-Vietoris sequence in lemma 6.2.

The Mayer-Vietoris sequence applies when we can break the vector space U into two subspaces $U = \langle W_1, W_2 \rangle$ with $\mathcal{S}^C(W_1) = \mathcal{S}^C(W_2) = \mathcal{S}$. This was true for the bubble boxes which required at most two sheets to cover their central hole, but more interesting constructions are possible. We will call these "Bubble Baubles" because their symmetry and cross-connections make them look like gems.

6.3.1 The Mayer-Vietoris Spectral Sequence

The Mayer-Vietoris Spectral sequence is the spectral sequence $\{E_{p,q}^s\}$ of the double complex of a cover of a topological space in which the columns are the chain complexes of $(p+1)$ -fold intersections of the components of the cover. Let X be a topological space and $\mathcal{U}_1, \dots, \mathcal{U}_k \subseteq X$ a set of subspaces such that $\bigcup_i \mathcal{U}_i = X$. Define the elements of the double complex as

$$E_{p,q}^0 = q\text{-chains in } (p+1)\text{-fold intersections.} \quad (6.5)$$

There is inconsistency in the indexing in (for instance) [67]. Here we'll take 0-fold intersections to be X itself, 1-fold intersections to be the elements $\mathcal{U}_1, \dots, \mathcal{U}_k$ of the cover, the 2-fold intersections to be intersections of two elements of the cover $\mathcal{U}_i \cap \mathcal{U}_j$, and so on.

The vertical differential maps $d^0 : E_{p,q}^0 \longrightarrow E_{p,q-1}^0$ on the columns are the usual boundary maps defined component-wise on the summands of $E_{p,q}^0$. The horizontal differential maps $d^1 : E_{p,q}^0 \longrightarrow E_{p-1,q}^0$ are inclusion maps multiplied by a sign factor derived from the sign of the boundary map on the nerve of the intersection. This is a simple thing but it deserves some exposition. Given a k -element cover as described we can define a $(k-1)$ -simplex on the elements $\mathcal{U} = \{\mathcal{U}_i\}$ of the cover, $\Delta^{\mathcal{U}} = [\mathcal{U}_1 \ \mathcal{U}_2 \ \dots \ \mathcal{U}_k]$. Then we associate the 0-simplices with the elements of the cover itself. The 1-simplices are naturally associated with the 2-fold intersections, so $[\mathcal{U}_i \ \mathcal{U}_j]$ is associated with $\mathcal{U}_i \cap \mathcal{U}_j$, and so on. The full simplex $\Delta^{\mathcal{U}}$ is associated with

$\mathcal{U}_1 \cap \cdots \cap \mathcal{U}_k$. Then we can apply the simplicial boundary map to the faces of $\Delta^{\mathcal{U}}$.

$$\partial[\mathcal{U}_{i_0} \cdots \mathcal{U}_{i_m}] = \sum_{j=0}^m (-1)^j [\mathcal{U}_{i_0} \cdots \widehat{\mathcal{U}_{i_j}} \cdots \mathcal{U}_{i_m}]$$

For example, if $X = A \cup B \cup C$ then $\Delta^{\mathcal{U}} = [A \ B \ C]$ and, for example, $\partial[A \ B] = B - A$

Looking back to the d^1 map we would have

$$E_{1,q}^0 = C_1(B \cap C) \oplus C_1(A \cap C) \oplus C_1(A \cap B)$$

$$E_{0,q}^0 = C_1(A) \oplus C_1(B) \oplus C_1(C)$$

An element of $E_{1,q}^0$ would look like (α, β, γ) . Then

$$d^1 : (\alpha, 0, 0) \mapsto (0, -\alpha, \alpha)$$

$$d^1 : (0, \beta, 0) \mapsto (-\beta, 0, \beta)$$

$$d^1 : (0, 0, \gamma) \mapsto (-\gamma, \gamma, 0)$$

$$d^1 : (\alpha, \beta, \gamma) \mapsto (-\beta - \gamma, -\alpha + \gamma, \alpha + \beta)$$

The feature that makes this a double complex is that the two differentials are anti-commutative, $d^1 d^0 = -d^0 d^1$. This means that the total complex, the direct sums along the diagonals of $E_{\cdot, \cdot}^0$, $\text{Tot}_n = \bigoplus_{p+q=n} E_{p,q}^0$, is a chain complex with differentials $(d^0 + d^1)$.

$$(d^0 + d^1) : \bigoplus_{p+q=n} E_{p,q}^0 \longrightarrow \bigoplus_{p+q=n-1} E_{p,q}^0 \quad (6.6)$$

We quickly check that this is a boundary map $(d^0 + d^1) \circ (d^0 + d^1) = d^0 d^0 + d^0 d^1 + d^1 d^0 + d^1 d^1 = 0 + d^0 d^1 - d^0 d^1 + 0 = 0$.

Then the E^0 page of the spectral sequence looks like

$$\begin{array}{ccccccc}
 & \vdots & & \vdots & & \vdots & \\
 & \downarrow & & \downarrow & & \downarrow & \\
 E_{0,2}^0 & \xleftarrow{d^1} & E_{1,2}^0 & \xleftarrow{d^1} & E_{2,2}^0 & \xleftarrow{\quad} & \dots \\
 \downarrow d^0 & & \downarrow d^0 & & \downarrow d^0 & & \\
 E_{0,1}^0 & \xleftarrow{d^1} & E_{1,1}^0 & \xleftarrow{d^1} & E_{2,1}^0 & \xleftarrow{\quad} & \dots \\
 \downarrow d^0 & & \downarrow d^0 & & \downarrow d^0 & & \\
 E_{0,0}^0 & \xleftarrow{d^1} & E_{1,0}^0 & \xleftarrow{d^1} & E_{2,0}^0 & \xleftarrow{\quad} & \dots
 \end{array} \tag{6.7}$$

The useful thing about this spectral sequence is that the homology of the total complex is given by taking the direct sum along the total degree k diagonals $H_k(\text{Tot}) = \bigoplus_{p+q=k} E_{p,q}^\infty$, and this is isomorphic to the homology of $X = \mathcal{U}_1 \cup \dots \cup \mathcal{U}_k$. If the covering $\mathcal{U} = \{\mathcal{U}_i\}$ is chosen so that all of the higher-degree, and most of the lower degree, intersections are empty, then this spectral sequence can make it possible to compute the homology of complicated objects from simple pieces.

The \mathcal{U}_i in our case are the coverage complexes of subspaces of V or individual vectors, $\mathcal{R}^C(\vec{v})$. For the purpose of recognizing false holes in $\mathcal{R}^C(V)$, however, we aren't really interested in computing the homology of the union of all $\mathcal{R}^C(\vec{v})$. Furthermore the elements of the cover are what they are, and are not chosen to intersect nicely in a way that simplifies the double complex. Finally, for a particular basis $\{\vec{v}_1, \dots, \vec{v}_D\}$ we only have D components to the cover, but in general there are $2^D - 1$ non-zero vectors in V , and to ensure we detect all false holes it could be

necessary to take all of them, giving intractably many non-empty intersections.

So at first it doesn't seem as if the Mayer-Vietoris spectral sequence will be useful computationally, but it still gives us a compact algebraic structure for studying the false holes in $\mathcal{R}^C(V)$.

6.3.2 $(4n)$ -vertex Bubble Baubles

In this discussion fix the Rips radius as $r = 1$. Let $n \in \mathbb{N}$ and consider a set of $4n$ points, X_1, \dots, X_{4n} , placed at equal intervals around a circle of radius $R = 1/\sqrt{2} - \epsilon$, $\epsilon \ll 1$, and another $4n$ points, Y_1, \dots, Y_{4n} placed at the same angles around a smaller concentric circle of radius $R' = \frac{1}{2} - \epsilon$. The vertices on the outer circle are given inventories of one vector each from the vector space $V = \text{GF}(2)^{2n}$. We choose a basis $\vec{v}_1, \dots, \vec{v}_{2n}$ for V and place \vec{v}_i in the inventory of antipodal points, X_i and X_{n+i} , on the larger $4n$ -gon. This is pictured for $4n = 8$ and the standard basis in figure 6.3.

In this arrangement the antipodal vertices in the smaller $4n$ -gon are linked by Rips edges, but the antipodal vertices of the larger $4n$ -gon are not. Also each vertex on the outer circle is within distance 1 of all of the vertices (on both smaller and larger $4n$ -gons) on its same side, including the vertices at 90° to it around the circle. Therefore each vertex X_i on the outer $4n$ -gon forms a Rips $(2n)$ -simplex with all of the smaller-circle vertices on its side of the circle (including the mid-line), but none of those on the other side of the circle. This $(2n)$ -simplex is contained in $\mathcal{R}^C(\vec{v}_i)$, as is its mirror image on the other side of the circle, containing X_{i+n} . But just as with

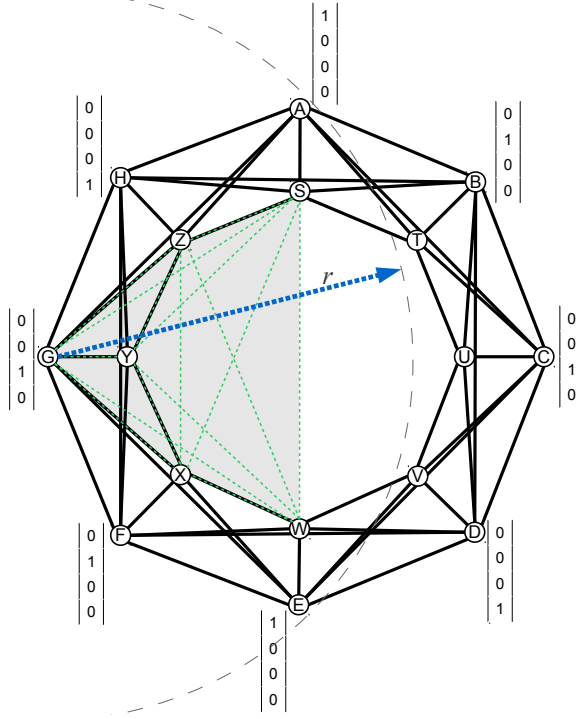


Figure 6.3: A data arrangement for a bubble octagon over $V = \text{GF}(2)^4$. The green dashed edges illustrate a maximal 6-simplex in the Rips complex that is excluded from coverage complexes spanning anything more than $\vec{v} = (0, 0, 1, 0)$.

the bubble boxes in section 6.1 they are covered by different copies of \vec{v}_i . These two simplices cover the interior of the smaller $4n$ -gon. That is, if $\alpha \in C_1(\mathcal{R}^C(\vec{v}_i))$ is the 1-cycle corresponding to the inner $4n$ -gon, then there is a 2-chain $\gamma \in C_2(\mathcal{R}^C(\vec{v}_i))$ with $\partial\gamma = \alpha$. Also X_i is within distance 1 of the two outer-circle vertices at 90° to it, so all of the 3-simplices on adjacent pairs of vertices on the inner and outer polygons are contained in $\mathcal{R}^C(\vec{v}_i)$ as well. So $p(\mathcal{R}^C(\vec{v}_i))$ contains the boundary of the Rips shadow $\partial\mathcal{S}$ and $H_1(\mathcal{R}^C(\vec{v}_i)) = 0$, so \mathcal{S} is covered by \vec{v}_i .

This is true for each \vec{v}_i . In fact the ring of 4-simplices connecting the inner $4n$ -gon to the outer $4n$ -gon is contained in $\mathcal{R}^C(V)$. However the pair of $2n$ -simplices covering the interior of the smaller polygon is different for each \vec{v}_i . For example the Rips edge connecting the two inner-circle vertices at 90° to X_i is included in $\mathcal{R}^C(\vec{v}_i)$,

but not in $\mathcal{R}^C(\vec{v}_j)$ for any $j \neq i$. Just as with the bubble box (the case for $n = 1$), we have $H_1(\mathcal{R}^C(\vec{v}_i) \cap \mathcal{R}^C(\vec{v}_j)) \neq 0$ for any $i \neq j$. Essentially the Rips shadow \mathcal{S} is covered by a different sheet for each basis vector, and those sheets intersect in the collar between the two polygons, but not inside the inner polygon.

We can study this as many instances of the Mayer-Vietoris sequence for a two-fold intersection. But we are actually dealing with a $2n$ -fold intersection in $\mathcal{R}^C(V)$, so it should be enlightening to study the spectral sequence that generalizes the Mayer-Vietoris sequence and how it can distinguish false holes from real holes in the code coverage. Though the construction of the complexes in the next section is slightly more complicated, we will do a spectral sequence computation there, because the bubble 6-gon will have fewer summands.

6.3.3 $(4n + 2)$ -vertex Bubble Baubles

In the case of $(4n + 2)$ -gons the bubble bauble is a little messier because there is no vertex on the $(4n + 2)$ -gon exactly at 90° to any other one. We can achieve the same effect by rotating the outer polygon that carries the information vectors by $2\pi/(2n + 2)$ relative to the smaller polygon. However if we look at this type of $4n + 2$ bubble bauble in isolation the outer boundary of the Rips shadow will always be uncovered in $\mathcal{R}^C(V)$. These problematic edges are the dashed gray ones in figure 6.4. In this example the thin 2-simplex $[F U A]$ is not entirely in range of E or B , and therefore is not included in $\mathcal{R}^C(\vec{v})$ where $\vec{v} = (0, 1, 0)$.

It is still possible to have an data arrangement containing a $(4n + 2)$ bubble

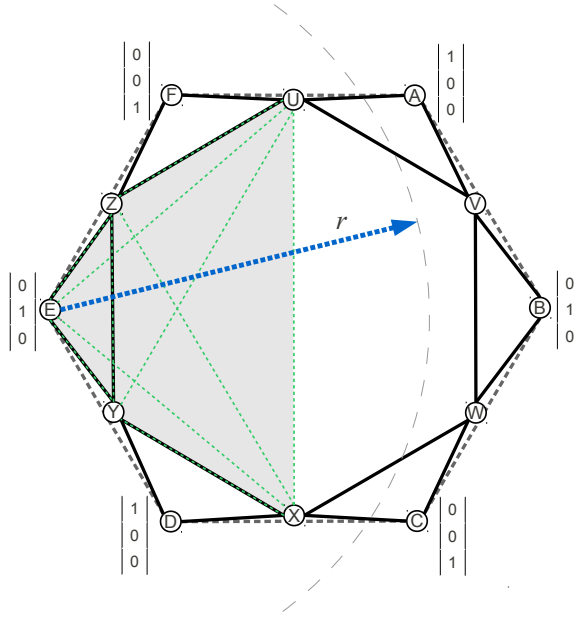


Figure 6.4: A data arrangement for a bubble hexagon over $V = \text{GF}(2)^3$. The solid Rips edges are covered by V . The green dashed edges illustrate a maximal 5-simplex in the Rips complex that is excluded from coverage complexes spanning anything more than $\vec{v} = (0, 1, 0)$.

bauble that satisfies the hypothesis that a fence subcomplex be contained in $\mathcal{R}^C(V)$; it just requires the data arrangement to have more vertices holding data outside the larger polygon. In this discussion we will ignore the issue with the outer boundary being covered because we are interested in the properties of the hole in the center of $\mathcal{R}^C(V)$.

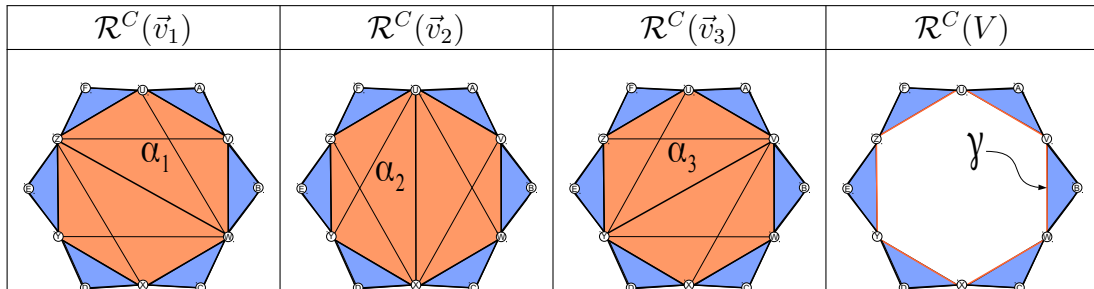


Figure 6.5: The inner hexagon in the bubble hexagon is covered by several partially disjoint 3-simplices. The faces of these 3-simplices give a collection of 2-chains that have the 1-cycle γ as a boundary.

To see how this relates to the spectral sequence let $A = \mathcal{R}^C(\vec{v}_1)$, $B = \mathcal{R}^C(\vec{v}_2)$, and $C = \mathcal{R}^C(\vec{v}_3)$. Note that $A \cup B \cup C \neq \mathcal{R}$, but we aren't really interested in the homology of either of them. In this case each of $A \cap B$, $B \cap C$, and $A \cap C$ has a non-trivial 1-cycle homologous to γ .

The E^0 page of the sequence consists of chain groups.

$$\begin{array}{ccccccc}
 \vdots & & \vdots & & \vdots & & \vdots \\
 \downarrow & & \downarrow & & \downarrow & & \downarrow \\
 C_2(A) \oplus C_2(B) \oplus C_2(C) & \xleftarrow{d^1} & C_2(B \cap C) \oplus C_2(A \cap C) \oplus C_2(A \cap B) & \xleftarrow{d^1} & C_2(A \cap B \cap C) & \xleftarrow{\dots} & \dots \\
 \downarrow d^0 & & \downarrow d^0 & & \downarrow d^0 & & \downarrow d^0 \\
 C_1(A) \oplus C_1(B) \oplus C_1(C) & \xleftarrow{d^1} & C_1(B \cap C) \oplus C_1(A \cap C) \oplus C_1(A \cap B) & \xleftarrow{d^1} & C_1(A \cap B \cap C) & \xleftarrow{\dots} & \dots \\
 \downarrow d^0 & & \downarrow d^0 & & \downarrow d^0 & & \downarrow d^0 \\
 C_0(A) \oplus C_0(B) \oplus C_0(C) & \xleftarrow{d^1} & C_0(B \cap C) \oplus C_0(A \cap C) \oplus C_0(A \cap B) & \xleftarrow{d^1} & C_0(A \cap B \cap C) & \xleftarrow{\dots} & \dots
 \end{array}$$

Let us write this in a simplified way and only include the generators that will be homologically interesting.

$$\begin{array}{ccccccc}
 \vdots & & \vdots & & \vdots & & \vdots \\
 \downarrow & & \downarrow & & \downarrow & & \downarrow \\
 \langle \alpha_1 \rangle \oplus \langle \alpha_2 \rangle \oplus \langle \alpha_3 \rangle & \xleftarrow{d^1} & 0 & \xleftarrow{d^1} & 0 & \xleftarrow{\dots} & \dots \\
 \downarrow d^0 & & \downarrow d^0 & & \downarrow d^0 & & \downarrow d^0 \\
 \langle \gamma \rangle \oplus \langle \gamma \rangle \oplus \langle \gamma \rangle & \xleftarrow{d^1} & \langle \gamma \rangle \oplus \langle \gamma \rangle \oplus \langle \gamma \rangle & \xleftarrow{d^1} & \langle \gamma \rangle & \xleftarrow{\dots} & \dots \\
 \downarrow d^0 & & \downarrow d^0 & & \downarrow d^0 & & \downarrow d^0 \\
 \langle x \rangle \oplus \langle x \rangle \oplus \langle x \rangle & \xleftarrow{d^1} & \langle x \rangle \oplus \langle x \rangle \oplus \langle x \rangle & \xleftarrow{d^1} & \langle x \rangle & \xleftarrow{\dots} & \dots
 \end{array}$$

Computing homology with the vertical maps first amounts to just computing homology component-wise on the direct sums.

$$\begin{array}{ccccccc}
0 & \xleftarrow{d^1} & 0 & \xleftarrow{d^1} & 0 & \xleftarrow{\quad} & \dots \\
0 & \xleftarrow{d^1} & \langle \gamma \rangle \oplus \langle \gamma \rangle \oplus \langle \gamma \rangle & \xleftarrow{d^1} & \langle \gamma \rangle & \xleftarrow{\quad} & \dots \\
\langle x \rangle \oplus \langle x \rangle \oplus \langle x \rangle & \xleftarrow{d^1} & \langle x \rangle \oplus \langle x \rangle \oplus \langle x \rangle & \xleftarrow{d^1} & \langle x \rangle & \xleftarrow{\quad} & \dots
\end{array}$$

Now we focus on the computation of $E_{1,1}^2$. The kernel of $d^1 : E_{1,1}^1 \rightarrow E_{0,1}^1$ is all of $E_{1,1}^1$, since $E_{0,1}^1 = 0$. The image of $d^1 : E_{2,1}^1 \rightarrow E_{1,1}^1$ is a 1-dimensional subspace generated by $(\gamma, -\gamma, \gamma)$.

$$\begin{aligned}
d^1 : E_{2,1}^1 &\longrightarrow E_{1,1}^1 \\
\gamma &\mapsto (\gamma, -\gamma, \gamma)
\end{aligned}$$

So $E_{1,1}^2 = \langle \gamma \rangle \oplus \langle \gamma \rangle \oplus \langle \gamma \rangle / \langle (\gamma, -\gamma, \gamma) \rangle$. It works out that $E_{1,0}^2 = 0$, and the sequence collapses after E^2 . So $E_{1,1}^2$ has rank 2 and is the only non-zero term of total degree 2. This means that $A \cup B \cup C$ has two nontrivial 2-cycles, as expected.

Let us consider the kernel of $d^1 : E_{1,1}^1 \rightarrow E_{0,1}^1$ again for a moment. There are essentially two ways an element of $E_{1,1}^1$ can get into the kernel of this map: either it is already in the kernel of the d^1 map (i.e. it would have been in $\ker(E_{0,1}^0 \xleftarrow{d^1} E_{1,1}^0)$), or it maps to something in $E_{0,1}^0$ that is in the image of $d^0 : E_{0,2}^0 \rightarrow E_{0,1}^0$. The latter correspond to false holes in $\mathcal{R}^C(V)$.

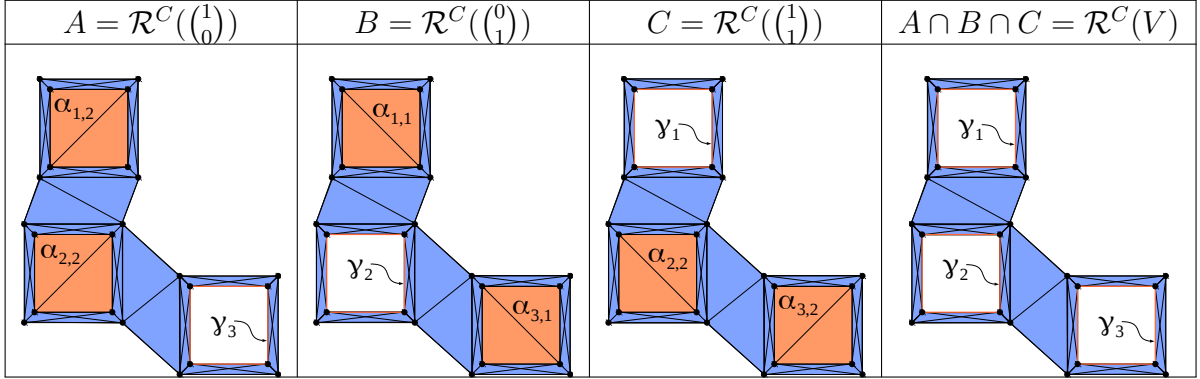


Figure 6.6: The triple bubble box with its key 1-chains and 2-chains highlighted.

6.3.4 The Spectral Sequence for the Triple Bubble Box

Just like the bubble 6-gon, the triple bubble box requires three sheets to verify coverage. But instead of having a single 1-cycle holding together three 2-chains to form two 2-cycles, the triple bubble box has three 1-cycles each holding together two sheets, forming three disjoint 2-cycles.

Let us use the same labels for the subcomplexes as in section 6.3.3. Let $A = \mathcal{R}^C\left(\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right)$, $B = \mathcal{R}^C\left(\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right)$, $C = \mathcal{R}^C\left(\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}\right)$. In this case V is rank 2, so any two-fold intersection of these subcomplexes gives $\mathcal{R}^C(V)$. The E^0 page of the spectral sequence then looks like:

$$\begin{array}{ccccc}
 \vdots & & \vdots & & \vdots \\
 \downarrow & & \downarrow & & \downarrow \\
 \langle \alpha_{1,2}, \alpha_{2,2} \rangle \oplus \langle \alpha_{1,1}, \alpha_{3,1} \rangle \oplus \langle \alpha_{2,2}, \alpha_{3,2} \rangle & \xleftarrow{d^1} & 0 & \xleftarrow{d^1} & 0 \\
 \downarrow d^0 & & \downarrow d^0 & & \downarrow d^0 \\
 \langle \gamma_1, \gamma_2, \gamma_3 \rangle \oplus \langle \gamma_1, \gamma_2, \gamma_3 \rangle \oplus \langle \gamma_1, \gamma_2, \gamma_3 \rangle & \xleftarrow{d^1} & \langle \gamma_1, \gamma_2, \gamma_3 \rangle \oplus \langle \gamma_1, \gamma_2, \gamma_3 \rangle \oplus \langle \gamma_1, \gamma_2, \gamma_3 \rangle & \xleftarrow{d^1} & \langle \gamma_1, \gamma_2, \gamma_3 \rangle \\
 \downarrow d^0 & & \downarrow d^0 & & \downarrow d^0 \\
 \langle x \rangle \oplus \langle x \rangle \oplus \langle x \rangle & \xleftarrow{d^1} & \langle x \rangle \oplus \langle x \rangle \oplus \langle x \rangle & \xleftarrow{d^1} & \langle x \rangle
 \end{array}$$

Computing homology with vertical maps first we obtain the E^1 page:

$$\begin{array}{ccccccc}
 0 & \xleftarrow{d^1} & & 0 & \xleftarrow{d^1} & & 0 \\
 \\
 \langle \gamma_3 \rangle \oplus \langle \gamma_2 \rangle \oplus \langle \gamma_1 \rangle & \xleftarrow{d^1} & \langle \gamma_1, \gamma_2, \gamma_3 \rangle \oplus \langle \gamma_1, \gamma_2, \gamma_3 \rangle \oplus \langle \gamma_1, \gamma_2, \gamma_3 \rangle & \xleftarrow{d^1} & \langle \gamma_1, \gamma_2, \gamma_3 \rangle & & \\
 \\
 \langle x \rangle \oplus \langle x \rangle \oplus \langle x \rangle & \xleftarrow{d^1} & \langle x \rangle \oplus \langle x \rangle \oplus \langle x \rangle & \xleftarrow{d^1} & \langle x \rangle & &
 \end{array}$$

In doing the computations on the E^1 page we have to pay careful attention to which intersections the generators lie in, and keep track of the “orientation” of the intersections. Let us compute the action of the d^1 map below on each of the basis elements.

$$\begin{array}{ccccccc}
 & B \cap C & & A \cap C & & A \cap B & & A \cap B \cap C \\
 E_{1,1}^1 = \langle \gamma_1, \gamma_2, \gamma_3 \rangle \oplus \langle \gamma_1, \gamma_2, \gamma_3 \rangle \oplus \langle \gamma_1, \gamma_2, \gamma_3 \rangle & \xleftarrow{d^1} & E_{2,1}^1 = \langle \gamma_1, \gamma_2, \gamma_3 \rangle & & & & & \\
 (\gamma_1 & , & -\gamma_1 & , & \gamma_1) & \xleftarrow{d^1} & \gamma_1 \\
 (\gamma_2 & , & -\gamma_2 & , & \gamma_2) & \xleftarrow{d^1} & \gamma_2 \\
 (\gamma_3 & , & -\gamma_3 & , & \gamma_3) & \xleftarrow{d^1} & \gamma_3
 \end{array}$$

We see that the map is injective, $\ker = 0$, and so $E_{2,1}^2 = 0$. This also gives us a basis for the image of d^1 is $E_{1,1}^1$. Now we look at the next step of the d^1 map in the $E_{*,1}^1$ row. The first three elements are the in the image $d^1 E_{2,1}^1$, and so necessarily in the kernel. The next three elements are the other three generators for the kernel. Therefore the middle three elements of $E_{1,1}^1$ are a basis for $E_{1,1}^2$. The final three elements map to non-zero values, giving a basis for the image in $E_{0,1}^1$. We see that this map is surjective, so $E_{0,1}^2 = 0$.

$$\begin{array}{ccccccc}
& A & & B & & C & & B \cap C & & A \cap C & & A \cap B \\
E_{0,1}^1 = & \langle \gamma_3 \rangle & \oplus & \langle \gamma_2 \rangle & \oplus & \langle \gamma_1 \rangle & \longleftarrow & E_{1,1}^1 = \langle \gamma_1, \gamma_2, \gamma_3 \rangle & \oplus & \langle \gamma_1, \gamma_2, \gamma_3 \rangle & \oplus & \langle \gamma_1, \gamma_2, \gamma_3 \rangle \\
& (-\gamma_3 + \gamma_3 & , & 0 & , & 0) & \longleftarrow & (0 & , & -\gamma_3 & , & \gamma_3) \\
& (0 & , & \gamma_2 - \gamma_2 & , & 0) & \longleftarrow & (\gamma_2 & , & 0 & , & \gamma_2) \\
& (0 & , & 0 & , & \gamma_1 - \gamma_1) & \longleftarrow & (\gamma_1 & , & -\gamma_1 & , & 0) \\
& (0 & , & 0 & , & 0) & \longleftarrow & (\gamma_3 & , & 0 & , & 0) \\
& (0 & , & 0 & , & 0) & \longleftarrow & (0 & , & \gamma_2 & , & 0) \\
& (0 & , & 0 & , & 0) & \longleftarrow & (0 & , & 0 & , & \gamma_1) \\
& (\gamma_3 & , & 0 & , & 0) & \longleftarrow & (0 & , & 0 & , & \gamma_3) \\
& (0 & , & \gamma_2 & , & 0) & \longleftarrow & (\gamma_2 & , & 0 & , & 0) \\
& (0 & , & 0 & , & \gamma_1) & \longleftarrow & (0 & , & -\gamma_1 & , & 0)
\end{array}$$

The sequence collapses on the E^2 page, leaving just three cycles of total degree 2.

$$\begin{array}{ccc}
0 & & 0 \\
0 & \langle (\gamma_2, 0, 0), (0, -\gamma_1, 0), (0, 0, \gamma_3) \rangle & 0 \\
\langle x \rangle & & 0
\end{array}$$

6.4 The Wormhole Complex

The wormhole complex is an example of a coverage complex which is considered by all of our coverage criteria to have a hole, even though the shadow is completely covered. It can be constructed using a data arrangement over a 1-dimensional vector space, so using local bases or multiple sheets will not fix it. This example exists for both \mathcal{R}^C and \mathcal{R}^{C1} .

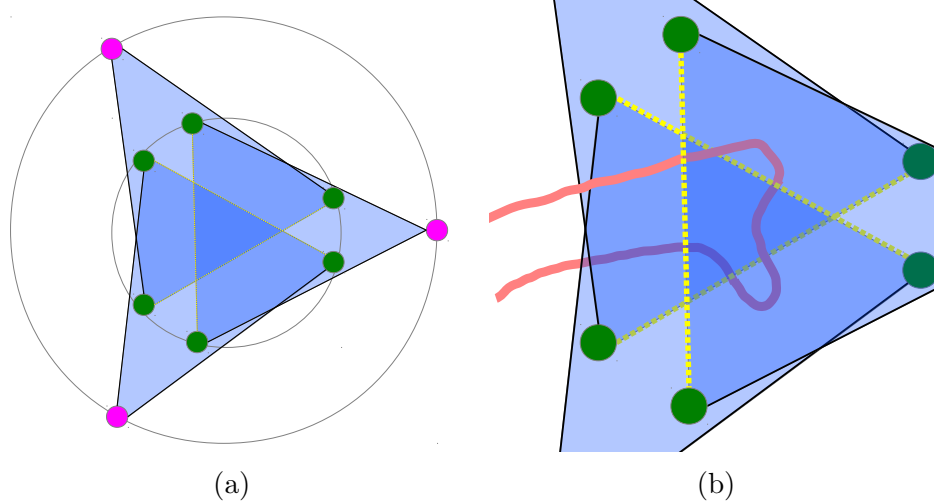


Figure 6.7: The essential structure of the wormhole complex. In the final coverage complex the yellow edges of the blue triangles will be “loose” (not a face of any other 2-simplex), making it possible for the pinkish worm to wiggle under them one at a time.

The essence of the wormhole complex is depicted by the three triangles in figure 6.7a. Vertices and edges have been left out to make it possible to highlight the key features. All three triangles will be Rips simplices, but the yellow dashed edges will be “loose” in $\mathcal{R}^C(V)$ in the sense that they will only be a face of one of the blue 2-simplices and nothing else. That means that, topologically speaking, something could slip under it. Figure 6.7b is an attempt to show how a worm might wiggle under the loose edges of the blue triangles, making its way from the top of the stack to the bottom.

The full construction of a wormhole complex is shown in figure 6.8. The data arrangement is built over a 1-dimensional vector space, $V = \text{GF}(2)^1$. Let r be the Rips radius we will use. We start by placing three pairs of green points at approximately 0° , 120° , and 240° around an inner circle of radius $r_0 = \frac{r}{2}$. Each of these pairs is perturbed by $\pm 10^\circ$ in either direction, as seen in the figures. None of

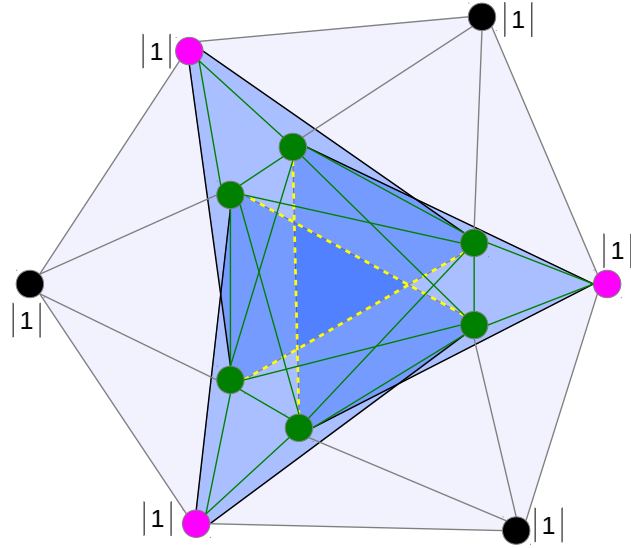


Figure 6.8: The full wormhole complex.

these inner points has any inventory.

Next we place three magenta points at angles of exactly 0° , 120° , and 240° around an outer circle of radius $r_1 = \sqrt{r^2 - \frac{3}{4}r_0^2} - \frac{1}{2}r_0 + \epsilon$. These points will support the blue triangles, and are each given an inventory of (1) (the only non-zero vector we have). The points are positioned to ensure that they are in range of the four closest green points on the inner circle, but out of range of the farthest two inner green points.

Finally we place three black points at angles of exactly 60° , 180° , and 200° around an outermost circle of radius $r_2 = \sqrt{r^2 - \frac{3}{4}r_1^2} + \frac{1}{2}r_1 - \epsilon$. This ensures that the black points are in range of the neighboring magenta points and the closest two inner green points, but just out of range of the two endpoints of the dashed yellow edge. Each of these is also given an inventory of (1).

Building a data arrangement with only the green and magenta points actually produces a wormhole. However the resulting Rips and coverage complexes do not

contain a fence subcomplex. We can see this in figure 6.7; the shadows of boundary edges intersect in their interiors, and so there are shadow vertices that do not correspond to Rips vertices on $\partial\mathcal{S}$. By including the black points we produce a convex outer boundary, and thereby a fence subcomplex. The special achievement here is to produce that outer boundary without tying off any of the loose edges in $\mathcal{R}^C(V)$.

This example appears to ruin all hope that we can have a universal solution to the problem of false holes. At least the local basis and multiple-sheeted approach will not work. There is a fence subcomplex in $\mathcal{R}^C(V)$, but there is no subspace of V which will make the fence contractible. On the other hand this example is incredibly narrow. Even small changes to the position of the points or the data arrangement will destroy the wormhole. This is an odd thing about the data coverage problem; for any reasonable arrangement of points and data, Chambers's isomorphism appears to hold. However carefully-constructed counterexamples such as this show that it cannot be proved in general.

Chapter 7: Examples, Software, and Future Work

This chapter provides a description of software we have written to compute and experiment with coverage complexes, and some examples produced. This includes examples from the previous chapters. We will also summarize some ongoing work that was not pertinent or not complete enough to make it into this thesis.

7.1 The CoverageSim Java Package

`CoverageSim` is a collection of Java classes and JNI wrappers we built to test and experiment with coverage complexes. It includes facilities to build a data arrangement, compute its Rips and coverage complexes, compute the persistent homology groups of the complexes and use the results to discount Rips complex holes, and visualize the combined complexes and their homology generators. It also contains a variety of experimental algorithms for manipulating and optimizing data arrangements.

`CoverageSim` uses the M4RI library for optimized linear algebra computation over finite fields [68]. M4RI is written in C and compiles to a native library, so we have built a Java Native Interface (JNI) wrapper for many of its functions. The wrapper provides a `Mzed` class which represents an m4ri `mzd_t` matrix object. It

exposes the necessary allocator and accessor functions as well as functions to add, multiply, echelonize, and compute ranks, kernels, and basis intersections.

The package also uses the `JavaPlex` package for persistent homology computations [52]. `JavaPlex` is an open source computational topology tool developed mainly at Stanford. It allows one to automate the construction of many types of filtered complexes from metric data, and includes many types of persistence algorithms, including Zigzag Persistence, over any finite field.

We also built on Stanford's `plex-viewer` Java package. `plex-viewer` gives an interface for 3D rendering of certain `JavaPlex` objects. The package uses the JOGL interface to OpenGL. We modified the package and its dependencies to support a more recent version of JOGL, and incorporated GLUT objects. Then we added a `drawCoverageComplex()` method and some support classes to support sensible display of a coverage complex superimposed on a Rips complex with different types of generators highlighted.

`CoverageSim` implements the following classes:

`DPoint` represents a point in a data arrangement. The point has a location and a vector inventory represented as a `Vector<Mzed>`. It also maintains a `HashSet<DPoint>` containing its neighbors and a `HashSet<DSimplex>` containing its Rips cofaces.

`DSimplex` represents a Rips simplex. It contains a `HashSet<DPoint>` containing its member points/vertices, a `HashSet<DPoint>` containing the neighbors common to all points in the simplex, and a third `HashSet<DPoint>`

containing “exclusive neighbors”, the neighbors common to all points of the simplex, excluding the points of the simplex itself. It also maintains two `Mzed` objects: one containing a reduced basis for the space spanned by the simplex, and one containing a reduced basis for the space spanned by the exclusive neighbors.

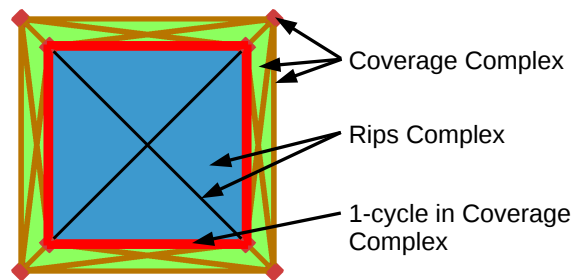
`DComplex` represents a simplicial complex. It is essentially a `HashSet<DSimplex>`. Its main feature is that it implements the `vsCover()` method. This is a sort-of dual to the $\mathcal{R}^C(\cdot)$ operation. The method operates on a simplicial complex Σ (assumed to be a subcomplex of a Rips complex, but it does not really matter), and produces the largest vector space $U \subseteq V$ such that $\Sigma \subseteq \mathcal{R}^C(U)$. The point of this computation is to take a Rips subcomplex that would cover a hole in $\mathcal{R}^C(V)$ and compute what portion of the vector space is already covering the hole.

`CoverageExperiment`

is the base class for all of the examples discussed here. The class maintains several `JavaPlex` streams representing the different simplicial complexes involved. It also includes all the methods to build the Rips and coverage complexes and manages the interface to `plex-viewer`.

`CoverageSim` is used by creating a new subclass of `CoverageExperiment` containing a `main()` method and a constructor that builds the data arrangement. Once the data arrangement is created one can call the inherited methods from `CoverageExperiment` to compute and manipulate the associated complexes.

The following sections contain some examples built with CoverageSim. In the visualizations red points, brown lines, and green fills are coverage complex vertices, edges, and simplices. Black points, black lines, and blue fills are Rips vertices, edges, and simplices that are not in the coverage complex. Red edges are first homology generators in the coverage complex. The display of each of these components can be interactively toggled in the software.



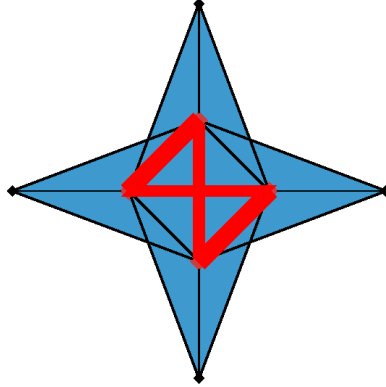


Figure 7.1: The non-lifting example computed by `CoverageSim`.

7.1.1 The Non-Lifting

This coverage complex in figure 7.1 shows the simplest example of a shadow cycle that cannot be lifted to a homotopic cycle in $\mathcal{R}^C(V)$. The black lines and blue triangles show the Rips 1 and 2-simplices, and the red lines highlight a 1-cycle in $H_1(\mathcal{R}^C(V))$. In fact the four 1-simplices highlighted in red are the entire coverage complex.

In this data arrangement there are eight points, and $V = \text{GF}(2)^2$. The four inner points have no inventory, and the outer points are given one vector each. Clockwise starting from the top vertex they are: $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$, $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$, $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$. This means that each outer point is redundant with one of its neighbors, so two edges of the inner square, the ones between redundant outer vertices, are left out of $\mathcal{R}^C(V)$.

Note that this example does not satisfy the fence subcomplex requirement. More points and inventory vectors could be added so that it does, though.

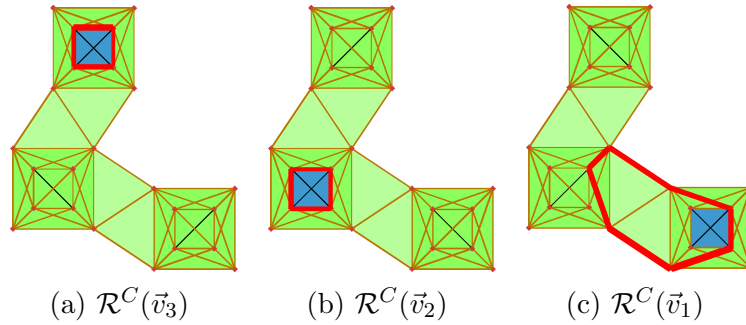


Figure 7.2: The triple bubble box and generators computed by CoverageSim.

7.1.2 The Bubble Box

Figure 7.2 shows $\mathcal{R}^C(\vec{v}_i)$ computed by CoverageSim for each of the non-zero vectors in V . Note that the homology generator found in fig 7.2c is not even close to the shortest cycle around the hole in the coverage complex. Bubble boxes can be used to tile the plane. It turns out that if four bubble boxes are properly arranged, a fifth bubble box will spring up in the space between them, covered by the inventory vectors held by the surrounding points. The interstitial bubble boxes are rotated slightly relative to the main grid. Figure 7.3 shows a 5×5 tiled grid of bubble boxes.

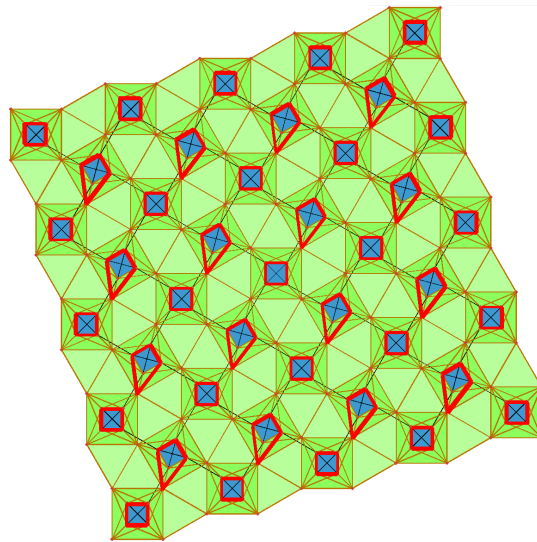


Figure 7.3: A grid of bubble boxes.

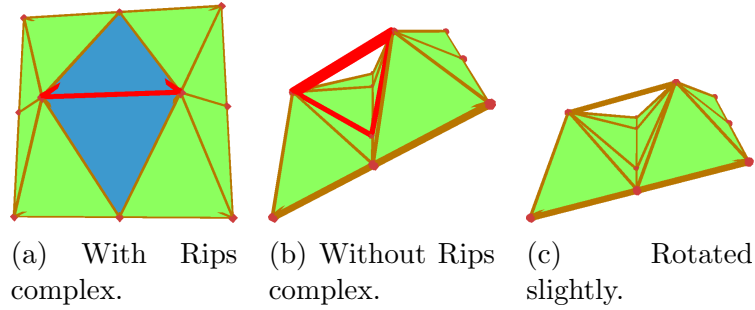


Figure 7.4: A layered loop computed by CoverageSim.

7.1.3 The Layered Loop and Layered Simplex

The layered loop and layered simplex were discussed in section 3.4. Figure 7.4 shows the layered loop example computed by CoverageSim. The points are all planar, but two of the points have been elevated in the visualization so we can see the space under the 1-simplex.

Figure 7.5 shows the layered simplex computed by CoverageSim. The three vertices of the 2-simplex have been similarly elevated for the visualization. The two non-trivial 1-cycles in $\mathcal{R}^C(V)$ are highlighted in red.

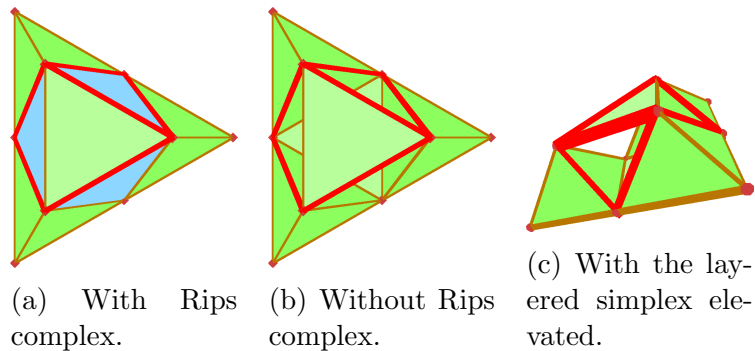


Figure 7.5: A layered simplex computed by CoverageSim.

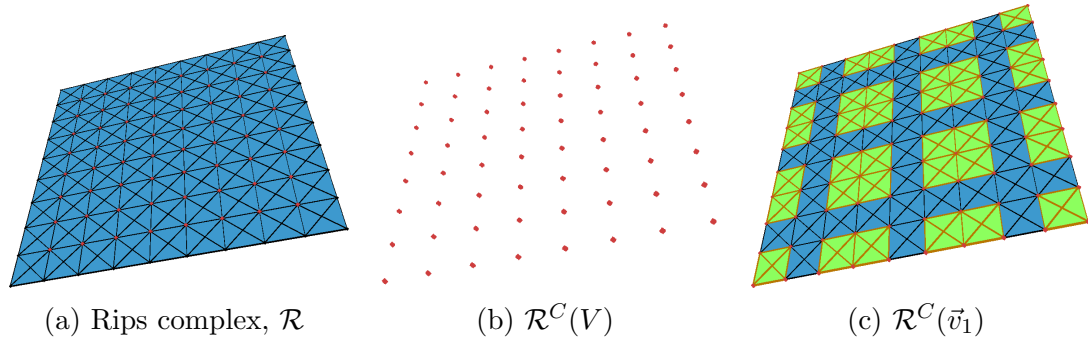


Figure 7.6: An example of resolving all resolvable vertices computed by CoverageSim.

7.1.4 Resolved Hex Points and Resolved Quad Points

This is a verification of our theoretical results about resolvability, and an experimental exercise in optimizing the vector space needed to achieve the resolution. These data arrangements have hexagonally packed and quadratically packed points spaced so that no point falls in the interior of any 2-simplex shadow. Each vertex, except the vertices on the edges of the grid, is then equal to the intersection of several maximal simplices, and therefore should be resolvable. On the other hand the dimension of the vector space needed for the universal data arrangement increases linearly with the number of maximal simplices. These examples establish a pattern that uses a single fixed vector space regardless of the size of the grid. The QuadPoints example uses a vector space of dimension 18, and the HexPoints example uses a vector space of dimension 14. Subfigure 7.6c shows the coverage complex computed for \vec{v}_1 , one of the 18 basis vectors used to resolve the points. The resulting pattern effectively reveals how the vectors are distributed in order to resolve the points.

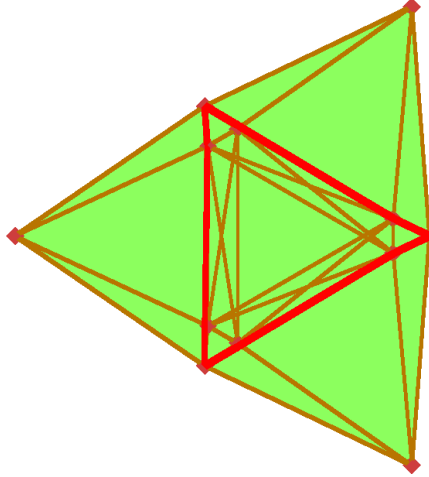


Figure 7.7: The wormhole complex computed by `CoverageSim`.

7.1.5 The Wormhole Complex

This example was thoroughly discussed in section 6.4. We see that even though the Rips shadow appears to be completely covered by the coverage complex, `CoverageSim` discovers the non-trivial 1-cycle in $\mathcal{R}^C(V)$.

7.1.6 The Coverage Grid and Fenced Coverage Square

Here we finally look at data arrangements that aren't contrived examples meant to demonstrate interesting mathematical structure. The **Coverage Grid** is a regular arrangement of points with randomly chosen inventory vectors. The points along the outside of the grid are always given a full basis for V . The Coverage Grid program takes the following arguments:

Grid Width

The number of point to place in each row of the grid

Grid Height

The number of points to place in each column of the grid.

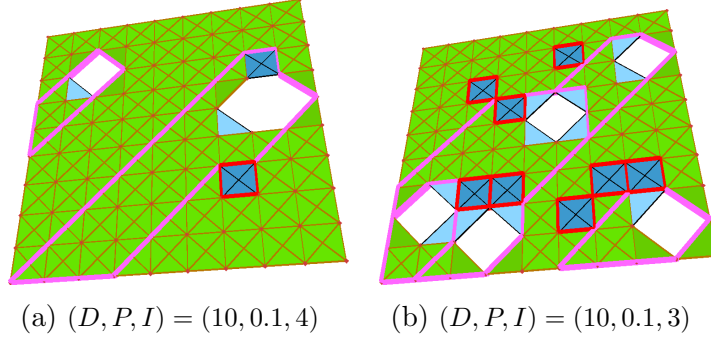


Figure 7.8: Some examples of Coverage Grids computed by CoverageSim.

Grid Spacing

The spacing between rows and columns.

$D = \text{Dimension of } V$

The dimension of the vector space.

$P = \text{Exclusion Probability}$

The probability than any given (non-fence) point is left out of the data arrangement.

$I = \text{Inventory Size}$

The number of random inventory vectors to be given to each (non-fence) point.

Figure 7.8 shows some small coverage grids for small vector spaces. Because in these examples we want to distinguish between 1-cycles that are in the Rips complex and 1-cycles that are only in the coverage complex, we highlight the former in pink. The red cycles still correspond to 1-cycles that are non-zero in $H_1(\mathcal{R}^C(V))$, but become bounding upon inclusion into $H_1(\mathcal{R})$.

We notice that these coverage complexes have a fence subcomplex covering the outer boundary, but that the inner fence subcomplex requirement is not satisfied. Therefore it is possible that if $[\gamma] \in H_1(\mathcal{R}^C(V))$ is a 1-cycle that does not become bounding upon inclusion into $H_1(\mathcal{R})$, the hole in $\mathcal{R}^C(V)$ may be larger than the underlying hole in \mathcal{R} ; without a full fence subcomplex the homological criterion

cannot distinguish the size of the holes. We observe exactly that happening in the examples in the figure.

The **Fenced Coverage Square** is a similar example, except the points inside the outer fence are placed randomly instead of on a grid. The placement is based on a uniform random distribution, but biased so that the points repel each other. The details are not important here. The arguments passed to the Fenced Coverage Square are almost the same as before:

Grid Width

The number of points wide the outer fence will be.

Grid Height

The number of points tall the outer fence will be.

Grid Spacing

The spacing between points on the fence.

$D =$ Dimension of V

The dimension of the vector space.

$N =$ Number of Points

The number of points to place in the interior of the fence.

$I =$ Inventory Size

The number of random inventory vectors to be given to each (non-fence) point.

Figure 7.9 shows some example of Fenced Coverage Squares. Fig 7.9a has parameters similar to those we used for the coverage grids. We see that the program detects two holes that are already present in the Rips complex, and several more holes that are only in the coverage complex. What appears to be just one larger coverage hole on the left side gets divided up into several cycles because it is crossed by several edges that are in $\mathcal{R}^C(V)$.

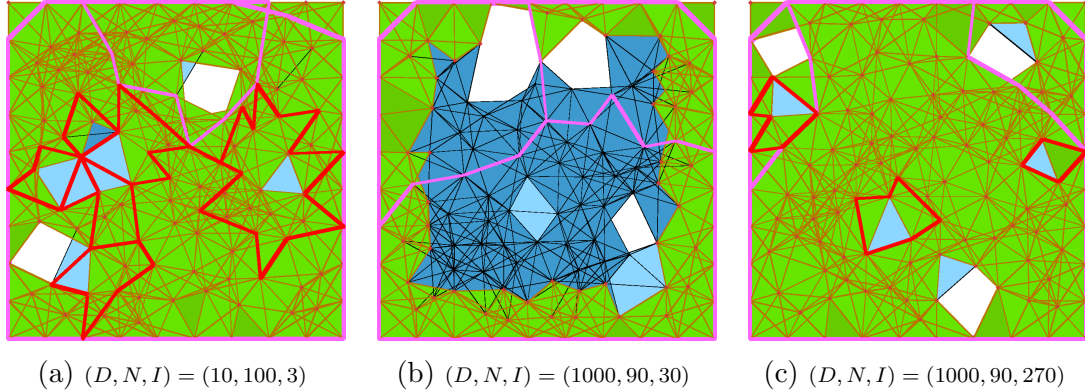


Figure 7.9: Some examples of Fenced Coverage Squares computed by `CoverageSim`. D = dimension of V . N = number of points. I = inventory size of non-fence points.

Figures 7.9b-7.9c show examples with much larger vector spaces. The middle figure has three Rips holes, but registers no coverage defects, even though most of the complex is not covered. Again, this is because we have no fence subcomplex covering the inner shadow boundaries. As a practical issue, we can still perform reasonable coverage testing by checking first that all points in the Rips complex are covered. By lemma 3.31 we know that must be true in the case of full coverage. This would practically bound how much larger a coverage hole could be than an underlying Rips hole.

The final figure is also for a 1000-dimensional vector space. In this case we place 270 vectors in the inventory of each point, and the space is nearly covered. There are still three Rips holes and three coverage holes, and the program successfully distinguishes them.

7.2 Ongoing and Future Work

Here we summarize some extensions to this work that we are investigating or would like to investigate.

7.2.1 Traversing the $\mathcal{R}_r^C(\mathcal{X}, U)$ Lattice With Zig-Zag Persistence

Traditional persistent homology requires a filtration. That is, a strictly increasing sequence of topological spaces. Naturally one would like to track homology classes if parts of the space are removed as well. Zigzag persistence was introduced by Carlsson and de Silva in [69]. It shows that there is an analogue to persistence intervals for linear sequences of modules with maps in either direction.

A **zig-zag module**, \mathbb{V} , is a sequence of vector spaces with maps between them.

$$V_1 \begin{array}{c} \xrightarrow{p_1} \\ \xleftarrow{} \end{array} V_2 \begin{array}{c} \xrightarrow{p_2} \\ \xleftarrow{} \end{array} \dots \begin{array}{c} \xrightarrow{p_{n-1}} \\ \xleftarrow{\phantom{p_{n-1}}} \end{array} V_n$$

Carlsson and de Silva show that \mathbb{V} can be decomposed into a sum of indecomposable “interval modules”. These are analogous to persistence intervals that track the homology classes across changes in the space. Furthermore there are quick algorithms to compute these invariants.

Zigzag persistence has been implemented in `JavaPlex` and `Dionysus` [52, 70]. The algorithms allow for “on-line” computation as a space changes. That is, the state of the computation can be held in memory and updated as simplices are added

and removed from a complex. Obviously if we are experimenting with algorithms that change a data arrangement, and thereby $\mathcal{R}^C(U)$, tracking the effects with these algorithms is much better than separate homology computations after each change. One major complication is the prospect of preserving the 2-step filtration we used to discount holes already in the Rips complex. Aside from that, the lattice of persistence complexes is not an arbitrary collection of simplicial complexes. It would be great to find structure that could be exploited to make homology tracking by zigzags more efficient.

7.2.2 Filtering $\mathcal{R}_r^C(\mathcal{X}, U)$ by Rips Radius

One of our motivations for studying these data arrangements was to put some lower or upper bounds on the work required to gather a full basis for V . What we have now is just several criteria that, for a given fixed Rips radius, allow us to test whether the original data bundle is recoverable from nodes within that radius. Ideally we would like to filter the coverage complex based on the Rips radius, r and look at the persistence intervals. For any fixed $U \subseteq V$ this is a standard persistence computation. But in the bigger picture we have a lattice of persistence modules.

This is a multi-dimensional persistence problem. Multi-dimensional persistence is an analogue of persistence for multi-filtrations [71]. It has been studied by multiple people and even in special cases found to be difficult. However trying to find structure amongst this lattice of persistence modules seems very compelling.

7.2.3 Alternative Coverage Complexes

There are at least two different analogues to the coverage complex that we have investigated. The **Knowledge Complex**, $\mathcal{R}_r^K(\mathcal{X}, U)$, is based on ideas from [13], and captures linear span in a way that is more sensitive to the unintuitive behavior of finite vector spaces. The **Nullspace Complex**, $\mathcal{R}_r^N(\mathcal{X}, U)$, is an attempt to build a simplicial complex capturing the lack of coverage in an area. We believe such a structure could be useful for maintaining code distribution in a CDMA wireless network. Both of these structures have properties similar, but not identical to, the coverage complex.

7.2.4 Optimizing Data Arrangements

For a given data arrangement, suppose we apply our coverage criteria and conclude that $\mathcal{S}^C(V) = \mathcal{S}$. It is entirely likely that vectors could be removed from some of the points in the arrangement while still preserving the coverage. For a given arrangement of points we would like to derive lower and upper bounds on the total number of vectors needed to cover the Rips shadow, and algorithms to heuristically optimize the data arrangement. We have some initial work on this topic, but not enough to include here.

Bibliography

- [1] R. Ahlswede, Ning Cai, S.-Y.R. Li, and R.W. Yeung. Network information flow. *Information Theory, IEEE Transactions on*, 46(4):1204–1216, Jul 2000.
- [2] R. W. Yeung and Z. Zhang. Distributed source coding for satellite communications. *IEEE Transactions on Information Theory*, IT(45):1111–1120, 1999.
- [3] Muriel Medard, Michelle Effros, David Karger, and Tracey Ho. On coding for non-multicast networks. In *Proc. 41st Annual Allerton Conference On Communication, Control AND Computing*, 2003.
- [4] Randall Dougherty, Christopher F. Freiling, and Kenneth Zeger. Networks, matroids, and non-Shannon information inequalities. *IEEE Transactions on Information Theory*, 53(6):1949–1969, 2007.
- [5] Randall Dougherty, Christopher F. Freiling, and Kenneth Zeger. Network coding and matroid theory. *Proceedings of the IEEE*, 99(3):388–405, 2011.
- [6] Robert Ghrist and Yasukai Hiroka. Applications of sheaf cohomology and exact sequences on network codings. In *International Symposium on Nonlinear Theory and its Applications*, 2011.
- [7] Salim Y. El Rouayheb, Alexander Sprintson, and Costas N. Georghiades. On the index coding problem and its relation to network coding and matroid theory. *IEEE Transactions on Information Theory*, 56(7):3187–3195, 2010.
- [8] Yong Wang, Sushant Jain, Margaret Martonosi, and Kevin Fall. Erasure coding based routing in opportunistic networks. In *ACM SIGCOMM Workshop on Delay Tolerant Networking*, 2005.
- [9] Sushant Jain, Michael Demmer, Rabin Patra, and Kevin Fall. Using redundancy to cope with failures in a delay tolerant network. *Proc. SIGCOMM*, 2005.

- [10] Yunfeng Lin, Baochun Li, and Ben Liang. Stochastic analysis of network coding in epidemic routing. *IEEE J.Sel. A. Commun.*, 26(5):794–808, June 2008.
- [11] Christos Gkantsidis and Pablo Rodriguez. Network coding for large scale content distribution. In *IEEE INFOCOM*, page 12, March 2005.
- [12] Agoston Petz, Brenton Walker, Chien-Liang Fok, Calvin Ardi, and Christine Julien. Network coded routing in delay tolerant networks: An experience report. In *Proceedings of the 3rd Extreme Conference on Communication (ExtremeCom)*, 2011.
- [13] Bernhard Haeupler. Analyzing network coding gossip made easy. In *Proceedings of the 43rd annual ACM symposium on Theory of computing*, STOC '11, pages 293–302, New York, NY, USA, 2011. ACM.
- [14] Brenton Walker, Calvin Ardi, Agoston Petz, Jung Ryu, and Christine Julien. Experiments on the spatial distribution of network code diversity in segmented dtns. In *Proceedings of the ACM MobiCom Workshop on Challenged Networks (CHANTS)*, 2011.
- [15] Vin DeSilva, Robert Ghrist, and Abubakr Muhammad. Blind swarms for coverage in 2-d. In Sebastian Thrun, Gaurav S. Sukhatme, and Stefan Schaal, editors, *Robotics: Science and Systems*, pages 335–342. The MIT Press, 2005.
- [16] Vin DeSilva and Robert Ghrist. Coverage in sensor networks via persistent homology. *Alg. & Geom. Topology*, 7:339–358, 2007.
- [17] Vin DeSilva and Robert Ghrist. Homological sensor networks. *Notices of the American Mathematical Society*, 54:2007, 2007.
- [18] Robert Ghrist and Abubakr Muhammad. Coverage and hole-detection in sensor networks via homology. In *Proceedings of the 4th international symposium on Information processing in sensor networks*, IPSN '05, Piscataway, NJ, USA, 2005. IEEE Press.
- [19] E. W. Chambers. *Computing Interesting Topological Features*. PhD thesis, University of Illinois at Urbana-Champaign, 2008.
- [20] Erin W. Chambers, Vin de Silva, Jeff Erickson, and Robert Ghrist. Vietoris-Rips complexes of planar point sets. *Discrete & Computational Geometry*, 44(1):75–90, 2010.
- [21] V. DeSilva and R. Ghrist. Coordinate-free coverage in sensor networks with controlled boundaries via homology. *I. J. Robotic Res.*, 25(12):1205–1222, 2006.
- [22] Yue Wang, Jie Gao, and Joseph S. B. Mitchell. Boundary recognition in sensor networks by topological methods. In Mario Gerla, Chiara Petrioli, and Ramachandran Ramjee, editors, *MOBICOM*, pages 122–133. ACM, 2006.

- [23] Dezun Dong, Yunhao Liu, and Xiangke Liao. Fine-grained boundary recognition in wireless ad hoc and sensor networks by topological methods. In *Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '09, pages 135–144, New York, NY, USA, 2009. ACM.
- [24] E. Sperner. Ein Satz über Untermengen einer endlichen Menge. *Math. Z.*, 27:544–548, 1928.
- [25] J. Moon and L. Moser. On cliques in graphs. *Israel Journal of Mathematics*, 3:23–28, 1965. 10.1007/BF02760024.
- [26] Rajarshi Gupta, Jean Walrand, and Olivier Goldschmidt. Maximal cliques in unit disk graphs: Polynomial approximation. In *PROCEEDINGS INOC 2005*, 2005.
- [27] R. P. Dilworth. A Decomposition Theorem for Partially Ordered Sets. *The Annals of Mathematics*, 51(1):161–166, 1950.
- [28] Micha A. Perles. A proof of Dilworth’s decomposition theorem for partially ordered sets. *Israel Journal of Mathematics*, 1(2):105–107, 1963.
- [29] Erin W. Chambers, Jeff Erickson, and Pratik Worah. Testing contractibility in planar Rips complexes. In *Proceedings of the Twenty-fourth Annual Symposium on Computational Geometry*, SCG '08, pages 251–259, New York, NY, USA, 2008. ACM.
- [30] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, FOCS '00, pages 454–, Washington, DC, USA, 2000. IEEE Computer Society.
- [31] Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete Comput. Geom.*, 33(2):249–274, February 2005.
- [32] A.J. Zomorodian. *Topology for Computing*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2005.
- [33] H. Edelsbrunner and J. Harer. Persistent homology – a survey. In *Surveys on discrete and computational geometry*, volume 453 of *Contemporary Mathematics*, pages 257–282. Amer Mathematical Society, 2008.
- [34] Gunnar Carlsson. Topology and Data. *Bulletin (New Series) Of The American Mathematical Society*, 46(2):255–308, 2009.
- [35] Robert Ghrist. Barcodes: The persistent topology of data. *Bulletin of the American Mathematical Society*, 45:61–75, 2008.
- [36] Shmuel Weinberger. What is . . . persistent homology? *Notices Am. Math. Soc.*, 58(1):36–39, 2011.

- [37] Herbert Edelsbrunner and Michael Kerber. Alexander duality for functions: the persistent behavior of land and water and shore. *CoRR*, arxiv:1109.5052, 2011.
- [38] Moo K. Chung, Peter Bubenik, Peter T. Kim, Kim M. Dalton, and Richard J. Davidson. Persistence diagrams of cortical surface data. In *Information Processing in Medical Imaging, LNCS*, 2009.
- [39] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete Comput. Geom.*, 37(1):103–120, January 2007.
- [40] Frédéric Chazal, David Cohen-Steiner, Marc Glisse, Leonidas J. Guibas, and Steve Y. Oudot. Proximity of persistence modules and their diagrams. In *Proceedings of the Twenty-fifth Annual Symposium on Computational Geometry, SCG '09*, pages 237–246, New York, NY, USA, 2009. ACM.
- [41] David Cohen-Steiner, Herbert Edelsbrunner, and Dmitriy Morozov. Vines and vineyards by updating persistence in linear time. In *Proceedings of the Twenty-second Annual Symposium on Computational Geometry, SCG '06*, pages 119–126, New York, NY, USA, 2006. ACM.
- [42] Dmitriy Morozov. *Homological Illusions of Persistence and Stability*. PhD thesis, Duke University, August 2008.
- [43] Vin de Silva. Plex: Simplicial complexes in MATLAB. Software available at <http://comptop.stanford.edu/programs/>.
- [44] Vin de Silva. A Weak Characterisation of the Delaunay Triangulation. *Geometriae Dedicata*, 135(1):39–64, 2008.
- [45] Vin de Silva and Gunnar Carlsson. Topological estimation using witness complexes. In *Sympos. Point-Based Graphics*, pages 157–166, 2004.
- [46] Gunnar Carlsson, Tigran Ishkhanov, Vin Silva, and Afra Zomorodian. On the local behavior of spaces of natural images. *Int. J. Comput. Vision*, 76(1):1–12, January 2008.
- [47] Brenton Walker. Using persistent homology to recover spatial information from encounter traces. In *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc '08*, pages 371–380, New York, NY, USA, 2008. ACM.
- [48] Xiaojin Zhu. Persistent homology: An introduction and a new text representation for natural language processing. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, IJCAI'13*, pages 1953–1959. AAAI Press, 2013.

- [49] Konstantin Mischaikow, Hiroshi Kokubu, Marian Mrozek, Pawe Pilarczyk, Tomas Gedeon, Jean-Philippe Lessard, and Marcio Gameiro. Chomp computational homology project. Software available at <http://chomp.rutgers.edu/>.
- [50] T. Kaczynski, M. Mrozek, and M. Slusarek. Homology computation by reduction of chain complexes. *Computers & Mathematics with Applications*, 35(4):59–70, February 1998.
- [51] Donald R. Sheehy. Linear-size approximations to the Vietoris-Rips filtration. *Discrete & Computational Geometry*, 49(4):778–796, 2013.
- [52] Andrew Tausz, Mikael Vejdemo-Johansson, and Henry Adams. Javaplex: A research software package for persistent (co)homology. Software available at <http://javaplex.github.io/javaplex/>, 2011.
- [53] Ulrich Bauer, Michael Kerber, and Jan Reininghaus. Phat persistent homology algorithm toolbox. Software available at <http://phat.googlecode.com/>, 2014.
- [54] S. Milgram. The small world problem. *Psychology Today*, 2(1):60–67, 1967.
- [55] Pan Hui, Augustin Chaintreau, James Scott, Richard Gass, Jon Crowcroft, and Christophe Diot. Pocket switched networks and human mobility in conference environments. In *WDTN '05: Proceeding of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 244–251, New York, NY, USA, 2005. ACM Press.
- [56] Magdalena Balazinska and Paul Castro. Characterizing Mobility and Network Usage in a Corporate Wireless Local-Area Network. In *1st International Conference on Mobile Systems, Applications, and Services (MobiSys)*, San Francisco, CA, May 2003.
- [57] Marvin Mcnnett and Geoffrey M. Voelker. Access and mobility of wireless pda users. *SIGMOBILE Mob. Comput. Commun. Rev.*, 9(2):40–55, April 2005.
- [58] Tristan Henderson, David Kotz, and Ilya Abyzov. The changing usage of a mature campus-wide wireless network. In *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 187–201, New York, NY, USA, 2004. ACM.
- [59] Wei-Jen Hsu and Ahmed Helmy. On nodal encounter patterns in wireless lan traces. In *Proceedings of the Second Workshop on Wireless Network Measurements (WinMee 2006)*, 2006.
- [60] Thomas Karagiannis, Jean-Yves Le Boudec, and Milan Vojnovic. Power law and exponential decay of inter contact times between mobile devices. In *MobiCom '07: Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, pages 183–194, New York, NY, USA, 2007. ACM.

- [61] Han Cai and Do Young Eun. Crossing over the bounded domain: from exponential to power-law inter-meeting time in manet. In *MobiCom '07: Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, pages 159–170, New York, NY, USA, 2007. ACM.
- [62] Ling-Jyh Chen, Yung-Chih Chen, Tony Sun, Paruvelli Sreedevi, Kuan-Ta Chen, Chen-Hung Yu, and Hao-Hua Chu. Finding self-similarities in opportunistic people networks. In *INFOCOM*, pages 2286–2290, 2007.
- [63] Nathan Eagle and Alex (Sandy) Pentland. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing*, V10(4):255–268, May 2006.
- [64] Eiko Yoneki, Pan Hui, and Jon Crowcroft. Visualizing community detection in opportunistic networks. In *CHANTS '07: Proceedings of the second workshop on Challenged networks CHANTS*, pages 93–96, New York, NY, USA, 2007. ACM.
- [65] Minkyong Kim, David Kotz, and Songkuk Kim. Extracting a mobility model from real user traces. In *IEEE INFOCOM06, Barcelona, Spain, 2006*.
- [66] Aaron Abrams. *Configuration Spaces and Braid Groups of Graphs*. PhD thesis, UC Berkeley, 2000.
- [67] David Lipsky, Primoz Skraba, and Mikael Vejdemo-Johansson. A spectral sequence for parallelized persistence. *CoRR*, arxiv:1112.1245, 2011.
- [68] Martin Albrecht and Gregory Bard. The M4RI Library – Version 20130416. Software available at <http://m4ri.sagemath.org>, 2013.
- [69] Gunnar E. Carlsson and Vin de Silva. Zigzag persistence. *CoRR*, arxiv:0812.0197, 2008.
- [70] Dmitriy Morozov. Dionysus library for computing persistent homology. Software available at <http://www.mrzv.org/software/dionysus/>, 2012.
- [71] Gunnar Carlsson and Afra Zomorodian. The theory of multidimensional persistence. *Discrete Comput. Geom.*, 42(1):71–93, May 2009.