Lecture notes for Math 840: Multiparameter Persistence

Michael Lesnick
SUNY Albany

Friday 20th September, 2019

Contents

Last updated Friday 20th September, 2019.

About These Notes

I wrote these notes for a one-semester graduate topics course on multi-parameter persistence at SUNY Albany. The class consisted of Ph.D. and Masters students with varied backgrounds, but some prior exposure to homology and basic abstract abstract algebra (vector spaces, groups, rings).

I have tried to make the notes as readable as possible, and I am making them publicly available in the hope that others might be able to benefit from them. Right now, there are (as far as I know) no written introductory materials available on multi-parameter persistent homology—no survey articles, textbooks, etc.—and perhaps these notes can serve as a first step towards filling this gap. That said, I have not yet done any fine polish of the notes, and you may find errors, omissions, or other issues. In particular, there is no bibliography here yet; instead, I have mentioned references inline, sometimes in a somewhat informal way. Also, there are no figures, and these would certainly enhance some parts of the notes. (There are, however, plenty of commutative diagrams.) And as they stand, the notes are not as complete of an introduction to the subject as I would like; we were only able to cover so much in a single semester. Important material that we didn't cover could easily fill a second semester. In any case, your feedback is welcome. In particular, feel free to notify me about any typos.

Bits and pieces of some of the algorithmic material here have been adapted from a paper I coauthored with Matthew Wright called "Computing Minimal Presentations and Betti

Numbers of 2-Parameter Persistent Homology." The material on Quiver representations benefited from discussions with Uli Bauer and Magnus Botnan.

1 Introduction

This is a course about topological data analysis (TDA), and specifically, about an approach to TDA called multi-parameter persistent homology (MPH). 1-parameter persistent homology (1PH) is probably the most widely studied and applied TDA method. MPH is a generalization that arises naturally in a number of places, such as the study of:

- noisy point cloud data,
- tendrils in data,
- time-varying data,
- point clouds equipped with an $\mathbb{R}$-valued function.

MPH yields invariants of data that are richer and more flexible, but also much more complex, than those provided by 1PH. As a result, 1PH theory and methodology tend not to extend naively to the multi-parameter setting. In the face of this, progress on MPH was relatively slow at first, especially on the practical end, while 1PH was progressing very quickly. But this has been changing in the last few years: Progress in computation and software seems to be accelerating; more researchers from the applied topology community are putting MPH at the center of their research programs; and some established researchers in other areas are taking a serious interest in MPH, bringing interesting new perspectives. As a result, MPH is a very vibrant research area. In my judgement, there is likely to be a lot of important progress in the next few years.

The broader goals of this research are to advance MPH as a practical data analysis methodology, and to develop the attendant theoretical foundations. As someone who has worked on MPH for several years, I believe MPH promises to be as useful and usable as 1PH in practical data analysis problems, and I expect that MPH will sooner or later become as widely used as 1PH. But there is a lot of work to do before we get to this point. In this class, we will study MPH with a focus on the problem of realizing the promise of MPH as a practical computational tool.

1.1 Possible Course Topics

Here is a rough, tentative outline of topics I hope to discuss.[1]

- Preliminaries: Posets, and basic category theory,

[1] ADDED LATER: This semester, we've covered many of the above topics but not all of them; see the table of contents and the Epilogue.

- 1-parameter PH, algebraic stability,
- Constructions of MPH from data,
- Representation-theoretic aspects,
- The non-existence of a good barcode for MPH,
- Minimal resolutions, presentations, and Betti numbers: Definition and computation (with emphasis on 2-parameter case),
- Multi-parameter interleavings: Universality, hardness of computation, applications,
- Stability of interlevelset persistent homology,
- The RIVET approach to visualization and analysis of 2-parameter persistence,
- The persistence perspective on clustering,
- Statistical aspects,
- Applications.

2 Preliminaries: Posets, and basic category theory

Commutative diagrams (especially those indexed by posets) play an important role in topological data analysis. Category theory provides a very convenient language for this, and we will often use that language in this course. We will cover the very basics here. If you want to learn more about category theory, Emily Riehl's recently undergraduate book or Steve Awodey's book are good options.

2.1 Posets

A *partially ordered set (or poset for short)* is a set P with a binary relation $<$ (the *partial order*) such that

- $x \not< x$ (irreflexivity),
- $x < y$ and $y \not< x$ implies $x = y$ (anti-symmetry),
- $x < y$ and $y < z$ implies $x < z$ (transitivity).

Example 2.1. Any totally ordered set is a poset, so $\mathbb{N}$, $\mathbb{Z}$, and $\mathbb{R}$ are all posets.

Example 2.2. Given a set S and P any set of subsets S, the inclusion relation on P is a partial order. A special case of this that arises frequently in mathematics (and in TDA theory) is to take P to the set of open sets of a topological space S.

Definition 2.3 (Product Poset)**.** Given posets P and Q, we define a partial order on $P \times Q$ by taking $(p, q) \leq (p', q')$ iff $p \leq p'$ and $q \leq q'$. In particular, this gives a partial order on P^2, and more generally, on P^n for any $n \geq 0$.

Definition 2.4 (Hasse Diagram)**.** A relation $x < y$ in a poset is *minimal* if it admits no factorization $x < z < y$. We can visually represent a finite poset $(P, <)$ as a finite directed graph, with vertices P, and edge (x, y) for each minimal relation $x < y$. This is called the Hasse diagram. More generally, if $(P, <)$ is a poset such that every relation $x < y$ is generated by minimal ones under transitive closure, we define the Hasse diagram in the same way.

Example 2.5. The Hasse diagram of $\{0, 1, 2\}$ is

$$\bullet \to \bullet \to \bullet.$$

The Hasse diagram of $\mathbb{N}$ is

$$\bullet \to \bullet \to \bullet \to \cdots$$

The Hasse diagram of $\{0, 1\}^2$ is

$$\begin{array}{ccc} \bullet & \longrightarrow & \bullet \\ \uparrow & & \uparrow \\ \bullet & \longrightarrow & \bullet \end{array}$$

The Hasse diagram of $\mathbb{N}^2$ is

$$\begin{array}{ccccccc} \vdots & & \vdots & & \vdots & & \\ \uparrow & & \uparrow & & \uparrow & & \\ \bullet & \longrightarrow & \bullet & \longrightarrow & \bullet & \longrightarrow & \cdots \\ \uparrow & & \uparrow & & \uparrow & & \\ \bullet & \longrightarrow & \bullet & \longrightarrow & \bullet & \longrightarrow & \cdots \\ \uparrow & & \uparrow & & \uparrow & & \\ \bullet & \longrightarrow & \bullet & \longrightarrow & \bullet & \longrightarrow & \cdots \end{array}$$

Definition 2.6 (Opposite Poset)**.** Given a poset P with partial order $<$, the *opposite partial order* $>$ is defined by $x > y$ iff $y < x$. We denote the resulting poset by P^{op}.

Example 2.7. The Hasse diagram of $\mathbb{N}^{\mathrm{op}}$ is

$$\bullet \leftarrow \bullet \leftarrow \bullet \leftarrow \cdots$$

2.2 Size Issues in Set Theory

The core definitions of category theory, as given below, necessarily touch on some foundational issues in set theory. We say a few brief words about this. In standard (ZFC) set theory, one cannot define a set of all sets: It follows from a generalization of Cantor's diagonalization argument that for any set S and 2^S its power set, there is no surjection $S \to 2^S$. If there

existed a set of all sets S, then since each element of 2^S is a set, we would have $2^S \subset S$, a contradiction. Similar kinds of arguments show that there is no set of all groups or set of all topological spaces.

Nevertheless, in category theory one does want to have something like a set of all sets. While there are various ways to arrange for this, the key idea is to introduce a distinction between sets and larger objects, called (proper) classes. By viewing the collection of all sets as a proper class rather than a set, we avoid the paradoxical notion of a set of all sets. When we don't want to worry about the formal distinction between a set and a class, we use the word *collection*.

In my work as an applied apologist, I have rarely needed to worry about subtle set-theoretic issues in practice. However, one should at least be aware of the issues. More details on this topic can be found in Mike Shulman's overview "Set Theory for Category Theory."

2.3 Some Basic Category Theory

Definition 2.8. A *category* $\mathcal{C}$ consists of:

- a collection of objects $\mathrm{Ob}\,\mathcal{C}$,
- a set of *morphisms* $\hom(X, Y)$ for every $x, y \in \mathrm{Ob}\,\mathcal{C}$,[2]
- a *composition rule* for morphisms, which for each $f \in \hom(x, y)$ and $g \in \hom(y, z)$, specifies a morphism $g \circ f \in \hom(x, z)$.
- for each $x \in \mathrm{Ob}\,\mathcal{C}$, a distinguished morphism $\mathrm{Id}_X \in \hom(x, x)$, called the *identity*.

These must satisfy the following axioms:

- composition is *associative*, i.e., $(f \circ g) \circ h = f \circ (g \circ h)$,
- $f \circ \mathrm{Id}_X = f$ and $\mathrm{Id}_X \circ g = g$ whenever such composition is defined.

Generalizing the usual notation for functions, if $f \in \hom(X, Y)$ we write $f : X \to Y$. We denote the collection of all morphism in $\mathcal{C}$ as $\hom(C)$.

Example 2.9. The category **Set** has objects all sets and morphisms all functions.

Example 2.10. The category **Top** has objects all topological spaces and morphisms all continuous maps.

Example 2.11. The category **Grp** has objects all groups and morphisms all group homomorphisms.

[2]It is common to avoid specifying that $\hom(X, Y)$ is a set, allowing for the possibility that it is a proper class; one sees both definitions in the literature. However, in practice $\hom(X, Y)$ is typically a set.

Example 2.12. For a fixed field K, the category **Vect** of K-vector spaces has objects all K-vector spaces and the morphisms the linear transformations.

Definition 2.13. A category C is said to be *thin* if for all $x, y \in \operatorname{Ob} C$, $\hom(x, y)$ contains at most one element. Note that in a thin category, the composition operation is trivial, in the sense that it is completely specified by the hom sets.

Example 2.14 (Poset Categories)**.** We can regard a poset P as a thin category:

- The objects are elements of P,
- if $x \leq y$ then $\hom(x, y)$ has one element,
- if $x \not\leq y$ then $\hom(x, y)$ is empty.

Definition 2.15. A category $\mathcal{C}$ is said to be *small* if $\operatorname{Ob} \mathcal{C}$ is a set, rather than a proper class. For example, **Set**, **Top**, and **Grp** are not small, but any poset category is small.

Definition 2.16. A morphism $f : x \to y$ in a category $\mathcal{C}$ is called an *isomorphism* if f has an *inverse*, i.e., a morphism $g : y \to x$ with

$$g \circ f = \mathrm{Id}_x \quad \text{and } f \circ g = \mathrm{Id}_y \,.$$

If there exists an isomorphism $f : x \to y$, we write $x \cong y$.

Definition 2.17 (Functor)**.** Given categories $\mathcal{C}$ and $\mathcal{D}$, a functor $F : \mathcal{C} \to \mathcal{D}$ consists of:

- An object $F(x) \in \operatorname{Ob} D$ for every $x \in \operatorname{Ob} C$,
- A morphism $F(\gamma) \in \hom(F(x), F(y))$ for each $\gamma \in \hom(x, y)$

such that

- F respects the composition operation in $\mathcal{C}$ and $\mathcal{D}$, i.e., $F(f \circ g) = F(f) \circ F(g)$,
- F maps identity morphisms to identity morphisms, i.e., $F(\mathrm{Id}_x) = \mathrm{Id}_{F(x)}$ for all $x \in \operatorname{Ob} C$.

For $X \in \operatorname{Ob} \mathcal{C}$, we often write $F(x)$ as F_x. Similarly, if $\mathcal{C}$ is thin, $x, y \in \operatorname{Ob} \mathcal{C}$, and $\alpha \in \hom(x, y)$, we write $F(\alpha)$ as $F_{x,y}$.

Example 2.18. For any $i \geq 0$, ith singular homology with coefficients in $\mathbb{Z}$ is a functor $H_i : \mathbf{Top} \to \mathbf{Grp}$. Homology with coefficients in a field K is a functor $\mathbf{Top} \to \mathbf{Vect}$.

Example 2.19. Let D be the Hasse diagram of a poset P. Then a commutative diagram of topological spaces indexed by D can be thought of as a functor $F : P \to \mathbf{Top}$, and vice versa. For example, a commutative diagram of spaces

$$\begin{array}{ccc} A & \xrightarrow{f} & B \\ {\scriptstyle g}\uparrow & & \uparrow{\scriptstyle h} \\ C & \xrightarrow{i} & D \end{array}$$

can be thought of as a functor

$$\{1,2\} \times \{1,2\} \to \mathbf{Top}\,.$$

Similarly, a diagram of spaces

$$F_1 \to F_2 \to F_3 \to \cdots$$

can be thought of as a functor $F : \mathbb{N} \to \mathbf{Top}$.

This generalizes immediately to commutative diagrams valued in any category $\mathcal{C}$. In this sense, functors out of posets generalize commutative diagrams.

Example 2.20. A functor $F : \mathbb{R} \to \mathbf{Top}$ is the data of a topological space F_r for each r and continuous maps $F_{r,s}$ for each $r \leq s$ such that

- $F_{r,r} = \mathrm{Id}_{F_r}$ for all r,
- $F_{s,t} \circ F_{r,s} = F_{r,s}$ for all $r \leq s \leq t$.

Definition 2.21 (Category of Small Categories)**. Cat** denotes the category whose objects are small categories and whose morphisms are functors. (Composition of functors and identity functors are defined in the obvious way.)

Definition 2.22 (Natural Transformations)**.** Given two functors $F, G : \mathcal{C} \to \mathcal{D}$, a *natural transformation* $N : F \to G$ is choice of morphism $N_x : F_x \to G_x$ for each $x \in \mathrm{Ob}\,\mathcal{C}$, such that these morphisms commute with those $\mathcal{C}$ and $\mathcal{D}$. That is, for all morphisms $\gamma : x \to y$ in $\hom(\mathcal{C})$, the following diagram commutes:

$$\begin{array}{ccc} F_x & \xrightarrow{F(\gamma)} & F_y \\ {\scriptstyle N_x}\downarrow & & \downarrow{\scriptstyle N_y} \\ G_x & \xrightarrow[G(\gamma)]{} & G_y \end{array}$$

If each of the morphisms N_x is an isomorphism, we call N a *natural isomorphism*

Simple examples such as the following are especially useful for getting an intuition for natural transformations.

Example 2.23. A natural transformation $N : F \to G$ of functors $F, G : \mathbb{N} \to \mathbf{Top}$ is exactly the data of continuous maps

$$N_i : F_i \to G_i\}_{i \in \mathbb{N}}$$

making the following diagram commute:

$$\begin{array}{lccccccc} F: & F_0 & \longrightarrow & F_1 & \longrightarrow & F_2 & \longrightarrow & \cdots \\ & \downarrow{\scriptstyle N_0} & & \downarrow{\scriptstyle N_1} & & \downarrow{\scriptstyle N_2} & & \\ G: & G_0 & \longrightarrow & G_1 & \longrightarrow & G_2 & \longrightarrow & \cdots \end{array}$$

Definition 2.24 (Functor Categories)**.** For categories $\mathcal{C}$ and $\mathcal{D}$ with $\mathcal{C}$ small, the *functor category* $\mathrm{Fun}(\mathcal{C}, \mathcal{D})$ (sometimes also denoted $\mathcal{D}^{\mathcal{C}}$), has as its objects the functors $\mathcal{C} \to \mathcal{D}$ and its morphisms the natural transformations.

Exercise 2.25. *Show that a morphism in $\mathcal{D}^{\mathcal{C}}$ is an isomorphism if and only if it is a natural isomorphism is the sense defined above.*

An *isomorphism* of categories is an invertible functor. One often would like to say that two categories have *essentially the same structure*. For this, the notion of isomorphism is sometimes too rigid. The following weaker notion is more common:

Definition 2.26 (Equivalence of categories)**.** A functor $F : \mathcal{C} \to \mathcal{D}$ is an *equivalence* if there exists a functor $G : \mathcal{D} \to \mathcal{C}$ such that

$$G \circ F \simeq \mathrm{Id}_{\mathcal{C}} \quad \text{and } F \circ G \simeq \mathrm{Id}_{\mathcal{D}}\,.$$

Example 2.27. The category whose objects are integers and whose morphisms $n \to m$ are the $m \times n$ matrices with coefficients in $\mathbb{R}$ is equivalent to, but not isomorphic to, the category of finite dimensional $\mathbb{R}$-vector spaces.

Example 2.28. $\mathbb{R}$ is isomorphic, hence equivalent, to $\mathbb{R}^{\mathrm{op}}$. For example, we may take the isomorphism to send x to $-x$ for all x.

Exercise 2.29.

(a) Describe all functors $\mathbb{Z} \to \mathbb{Z}$,

(b) Describe all equivalences $\mathbb{Z} \to \mathbb{Z}$,

(b) Describe all equivalences $\mathbb{Z} \to \mathbb{Z}^{\mathrm{op}}$.

Proposition 2.30. *A functor $F : \mathcal{C} \to \mathcal{D}$ is an equivalence if and only if*

1. *It is fully faithful, i.e., for all $x, y \in \mathrm{Ob}\,\mathcal{C}$, the map $\hom(x, y) \to \hom(F_x, F_y)$ given by F is a bijection.*
2. *F is* essentially surjective*, i.e. for every $y \in \mathrm{Ob}\,\mathcal{D}$, there is some $c \in \mathrm{Ob}\,\mathcal{C}$ with $F_c \cong d$.*

For a proof, see, e.g., Emily Riehl's book. The "if" direction of the statement requires the axiom of choice.

3 1-Parameter Persistent Homology

The basic persistent homology pipeline is summarized by the following diagram:

$$\boxed{\text{Data}} \to \boxed{\text{Filtration}} \xrightarrow{\text{Homology}} \boxed{\text{Persistence Module}} \xrightarrow{\text{Structure Theorem}} \boxed{\text{Barcode}}$$

3.1 Filtrations and Persistence Modules

To explain this, we start by defining filtrations and persistence modules:

Let $\mathcal{J}$ be one of the poset categories $\mathbb{N}$, $\mathbb{Z}$, or $\mathbb{R}$, or the opposite poset of one of these. Then a ($\mathcal{J}$-indexed) *filtration* is a functor $F : \mathcal{J} \to \mathbf{Top}$ such that $F_r \subset F_s$ whenever $r \leq s$.

Similarly, fixing a field K, a ($\mathcal{J}$-indexed) *persistence module* M is a functor $F : \mathcal{J} \to \mathbf{Vect}$, where as above, **Vect** denotes the category of K-vector spaces.

Thus, in view of Example 2.19, we think of an $\mathbb{N}$-indexed filtration F as a diagram of topological spaces of the form

$$F_0 \hookrightarrow F_1 \hookrightarrow F_2 \hookrightarrow F_3 \to \cdots,$$

and we think of an $\mathbb{N}$-indexed persistence module M as a diagram of K-vector spaces

$$M_1 \to M_2 \to M_3 \to M_3 \to M_4 \to \cdots.$$

An *interval* (in $\mathbb{R}$) is a non-empty subset $I \subset \mathbb{R}$ such that for all $x < y < z \in S$ with $x, y \in I$, we also have $y \in I$. A *barcode* is simply a multiset of intervals with $a < b \in \mathbb{R} \cup \{\infty\}$. (Informally, a multi-set is a set where elements are allowed to have multiple copies.)

3.2 The Persistent Homology Pipeline

As suggested by the diagram, the persistent homology pipeline proceeds in three steps:

- Given some data, we construct a filtration F from the data, in a way that captures some interesting aspect of the shape of the data.
- Post-compose F with the i^{th} homology functor H_i to obtain a persistence module H_iF. Concretely, this means applying homology to each space and each inclusion map in the filtration.
- A structure theorem for persistence modules yields a barcode $\mathcal{B}_M$ as an isomorphism invariant of any persistence module M, under very mild assumptions on M. (In short, the structure theorem tells us that M decomposes in an essentially unique way into simple summands called *interval modules*; see below.) Thus for each $i \geq 0$, we obtain a barcode $\mathcal{B}_{H_iF} := \mathcal{B}_i(F)$.

We interpret each interval in the barcode $\mathcal{B}_i(F)$ as a topological feature of the data, and we interpret the length of the interval as a measure of the robustness of this feature to perturbation of the data. In what follows, we state the structure theorem. We will then discuss several important ways of constructing a filtration from data.

3.3 Structure Theorem for Persistence Modules

Recall that the *direct sum* $V \oplus W$ of K-vector spaces V and W is defined as a set by

$$V \oplus W = \{(v, w) \mid v \in V,\ w \in W\},$$

with addition and scalar multiplication defined coordinate-wise. Moreover, given linear maps $f : V \to V'$ and $g : W \to W'$, we define $f \oplus g : V \oplus W \to V' \oplus W'$ by $f \oplus g(v, w) = (f(v), g(w))$. For any category I and functors $F, G : I \to \mathbf{Vect}$, these definitions induce a definition of direct sum $F \oplus G : I \to \mathbf{Vect}$.

Definition 3.1. An *interval* in $\mathcal{J}$ is subset $I \subset \mathcal{J}$ such that if $a < b < c \in \mathcal{J}$ and $a, c \in I$, then $b \in I$.

For $I \subset \mathcal{J}$ an interval, define the *interval module* K^I to be the persistence module such that

$$K^I_r = \begin{cases} K & \text{if } r \in I, \\ 0 & \text{otherwise.} \end{cases} \qquad K^I_{r,s} = \begin{cases} \mathrm{Id}_K & \text{if } r \leq s \in I, \\ 0 & \text{otherwise.} \end{cases}$$

For example, an interval module over $\mathbb{N}$ looks like this:

$$0 \longrightarrow \cdots \longrightarrow 0 \longrightarrow k \xrightarrow{\mathrm{id}_k} \cdots \xrightarrow{\mathrm{id}_k} k \longrightarrow 0 \longrightarrow \cdots$$

or like this:

$$0 \longrightarrow \cdots \longrightarrow 0 \longrightarrow k \xrightarrow{\mathrm{id}_k} \cdots \xrightarrow{\mathrm{id}_k} k \longrightarrow k \longrightarrow \cdots$$

We say a persistence module M is *pointwise finite dimensional (p.f.d.)* if $\dim(M_r) < \infty$ for all r.

Theorem 3.2 (Structure of Persistence Modules)**.** *If M is an $\mathbb{R}$-indexed or $\mathbb{Z}$-indexed p.f.d. persistence module, then there exists a unique multiset of intervals $\mathcal{B}_M$ such that*

$$M \cong \oplus_{I \in \mathcal{B}_M} K^I.$$

We call $\mathcal{B}_M$ the *barcode of M*.

Remark 3.3 (History)**.** The result was proven for $\mathbb{Z}$-indexed modules by Webb in 1985, and for $\mathbb{R}$-indexed modules by Crawley-Boevey in 2012. The special case of finitely generated $\mathbb{Z}$-indexed (or $\mathbb{R}$-indexed) modules is a slight variant of the (very standard) structure theorem for finitely generated modules over a PID, which can be found in any undergraduate abstract algebra textbook.

Sometime in the next few lectures, we will prove the structure theorem in the special case of finitely generated $\mathbb{Z}$-indexed modules.

3.4 Essentially Discrete Persistence Modules

We will say that a persistence module M is *essentially discrete* if there is an injection $j : \mathbb{Z} \hookrightarrow \mathbb{R}$ with $\lim_{\pm\infty} j(z) = \pm\infty$, such that for all $z \in \mathbb{Z}$ and $r \leq s \in [j(z), j(z+1))$, $M_{r,s}$ is an isomorphism. It's easy to see that the intervals in the barcode of an essentially discrete persistence module are all of the form $[a, b)$, for $a < b \in \mathbb{R} \cup \{\infty\}$.

The persistence modules we encounter in practice are typically essentially discrete.

4 Filtrations in Topological Data Analysis

4.1 Sublevel/Superlevel Filtrations

The flexibility of persistent homology lies in the way we construct a filtration from data. The data one starts with can take any one of several forms, and there are numerous ways of associating a filtration to data. Here we will discuss a few of the most common constructions, starting with sublevel and superlevel filtrations.

Definition 4.1 (Sublevel/Superlevel filtration)**.** For T any topological space and $\gamma : T \to \mathbb{R}$, define the *sublevel-filtration* $\mathcal{S}^{\uparrow}(\gamma)$ to be the $\mathbb{R}$-indexed filtration given by

$$\mathcal{S}^{\uparrow}(\gamma)_r = \{p \in T \mid \gamma(p) \leq r\}.$$

Note that for this construction, it is not necessary that γ be continuous.

Symmetrically, we define the *superlevel filtration* $\mathcal{S}^{\downarrow}(\gamma)$ to be the $\mathbb{R}^{\mathrm{op}}$-indexed filtration given by

$$\mathcal{S}^{\downarrow}(\gamma)_r = \{p \in T \mid \gamma(p) \geq r\}.$$

The persistence barcodes of the sublevel and superlevel filtration do not determine each other. However, we will see later that the sublevel filtration and superlevel filtration have a common 2-parameter refinement called the *interlevel* filtration.[3]

Remark 4.2. In the case where T is a differentiable manifold and $\gamma : T \to \mathbb{R}$ is a Morse function[4], $\mathcal{S}^{\uparrow}(\gamma)$ is a classical object of study in Morse theory. The connection between persistence and Morse theory is very important to how specialists in the field think about persistence.

Example 4.3 (Union-of-balls filtration)**.** Here is one of the most important special cases of the sublevel filtration in TDA: Let $P \subset \mathbb{R}^n$ be a finite set (sometimes called a *point cloud*), and let $d_P : \mathbb{R}^n \to [0, \infty)$ be the distance to P, i.e.,

$$d_P(x) = \min_{y \in P} \|x - y\|.$$

[3] As shown by Carlsson, de Silva, and Morozov in their paper on levelset zigzag persistence, a refinement of sublevel persistent homology called *extended persistence* does determine the superlevel version, and conversely, provided we consider barcodes in all multiple homology degrees..

[4] i.e., differentiable function whose Hessian is everywhere non-degenerate

Then $\mathcal{S}^{\uparrow}(d_P)_r$ is the union of balls of radius r centered at P.

Example 4.4 (density functions)**.** The following example of a supserlevel filtration is of fundamental importance, especially with regards to the statistical foundations of TDA. Let T be a Riemannian manifold (e.g. $\mathbb{R}^n$, or a unit sphere), and $\gamma : T \to \mathbb{R}$ be a probability density function. $\mathcal{S}^{\downarrow}(\gamma)$ topologically encodes information about the modes (i.e. basins of attraction under gradient flow) of the density function, as well as other higher-order topological features.

Example 4.5 (Images)**.** We can think of a 2-D greyscale image as a (discontinuous) function $\gamma : [0,1]^2 \to \mathbb{R}$. (We have discontinuities because the image is pixelated.) We can then consider the filtrations $\mathcal{S}^{\uparrow}(\gamma)$ and $\mathcal{S}^{\downarrow}(\gamma)$. It is common to use the these to study spaces of images, e.g. in a machine learning context. Sometimes the persistent homology is used for image processing, as in recent work of Chung and Day (2018).

4.2 The (Persistent) Nerve Theorem

In the computational setting, it is preferable to avoid working directly with the union-of-balls filtration, and instead work with an equivalent filtration with combinatorial structure. There are several options for this; we will consider the two most commonly ones in TDA, the Čech filtration and the Delaunay filtration (also known as the α-complex filtration). Vietoris-Rips complexes, which are related, will be discussed later.

Definition 4.6. An *abstract simplicial complex* is a collection Δ of non-empty finite subsets (called *simplices*) of a set S with the the property that if $\sigma \in \Delta$ and $\emptyset \neq \tau \subset \sigma$, then $\tau \in \delta$.

We usually use square brackets to denote simplices.

Example 4.7. {[1],[2],[3],[1 2],[1 3],[2 3]} is the abstract simplicial complex whose geometric realization is a triangle.

Definition 4.8 (Nerve)**.** Given a collection of sets $U = \{U^\alpha\}_{\alpha \in S}$ indexed by a set S, the *nerve of* U is the abstract simplicial complex

$$\mathcal{N}(U) = \{\sigma \subset S \mid \cap_{\alpha \in \sigma} U^\alpha \neq \emptyset\}.$$

Thus, $\mathcal{N}(U)$ has

- a 0-simplex for every $U^\alpha \in U$,
- a 1-simplex for every $\alpha, \beta \in S$ with $U^\alpha \cap U^\beta \neq \emptyset$,
- a 2-simplex for every $\alpha, \beta, \gamma \in S$ with $U^\alpha \cap U^\beta \cap U^\gamma \neq \emptyset$,
- and so on for higher simplices.

The *nerve theorem* guarantees that under suitable assumptions, the nerve of a cover of a topological space X is homotopy equivalent to X. The theorem dates back to work of Borsuk, Weil, and Leray in the late 40's and early 50's, and comes in several different flavors. The following version can be found in Hatcher's text, section 4.G. I believe it is originally due originally to Borsuk.

Definition 4.9 (Cover)**.** A *cover* of a set S is a collection of subsets of S whose union is equal to S.

Theorem 4.10 (Nerve Theorem for Open Covers)**.** *If U is an open cover of a paracompact (e.g., metrizable) space X such that all intersections of finitely many elements of U are contractible, then $X \simeq \mathcal{N}(U)$.*

X is said to be *paracompact* if for every open cover of X has a *refinement* that is locally finite. A *refinement* of a cover C of X is a new cover D of X such that every set in D is contained in some set in C.

Outline of proof Theorem 4.10. We outline Hatcher's proof in Section 4.G of his book. For our purposes, the most important aspect of this proof is that one constructs a third space Z and homotopy equivalences

$$X \xleftarrow{\simeq} Z \xrightarrow{\simeq} \mathcal{N}(U).$$

The proof touches on some fundamental constructions in homotopy theory, like nerves of categories and homotopy colimits.

- Consider the poset P whose objects are are finite intersections of elements of U and whose morphisms are inclusions. In fact, the finite intersections specify a functor $F : P \to \mathbf{Top}$.
- We also have a functor $G : P \to \mathbf{Top}$ with $Y(p) = *$ for all $p \in P$ and a natural transformation $f : X \to Y$.
- For any small category I, there is a functor

  $$\mathrm{hcl} : \mathrm{Fun}(I, \mathbf{Top}) \to \mathbf{Top},$$

  the *homotopy colimit functor*, which is very useful in algebraic topology. Roughly, $\mathrm{hcl}(F)$ of a diagram F is a space obtained by gluing together thickened versions of the spaces F_x.
- hcl has the property that given diagrams $A, B : I \to \mathbf{Top}$ and $f : A \to B$ a natural transformation with f_x a homotopy equivalence for all $x \in \mathrm{Ob}\, I$, we have that the induced map $\mathrm{hcl}(A) \to \mathrm{hcl}(B)$ is a weak equivalence. In particular, we obtain a homotopy equivalence $\mathrm{hcl}(F) \to \mathrm{hcl}(G)$.
- Also, $\mathrm{hcl}(G)$ is homeomorphic to $\mathcal{N}(U)$.

- We have a natural map $\mathrm{hcl}(F) \to X$. Paracompact spaces admit partitions of unity. Using this, we show that this map is a homotopy equivalence.

Putting this all together, and taking $Z = \mathrm{hcl}(F)$, we have our desired pair of homotopy equivalences

$$X \xleftarrow{\simeq} Z \xrightarrow{\simeq} \mathcal{N}(U). \qquad \square$$

Another variant of the nerve theorem appears in the computational topology textbook of Edelsbrunner and Harer, who attribute it to Leray. This is the version that we will use below

Theorem 4.11 (Nerve Theorem, version 2)**.** *If U is a finite, closed, convex cover of $X \subset \mathbb{R}^n$, then $X \simeq \mathcal{N}(U)$.*

Remark 4.12. There is also a version of the nerve theorem whose hypothesis and conclusion concern homology, rather than homotopy.

For persistence theory, it is important to have an extension of the nerve theorem to filtrations. A version of this was introduced by Chazal and Oudot in 2008, and another (independently, I believe) by Bendich et al. in 2007. To explain this, we need a bit more language. First, we need to talk about what it means for two diagrams of topological spaces (in particular, two filtrations) to be topologically equivalent. That is, we need an analogue of homotopy equivalence for diagrams of spaces. The following is the standard approach (however, see the remark below).

Definition 4.13 (Weak Equivalence of Diagrams of Spaces)**.** Given a small category $\mathcal{J}$ and $F, G : \mathcal{J} \to \mathbf{Top}$, a natural transformation $N : F \to G$ is called a *objectwise homotopy equivalence* if N_x is a homotopy equivalence for each $x \in \mathrm{Ob}\,\mathcal{J}$. We say F and G are *weakly equivalent* if they are connected by a zig-zag of of objectwise homotopy equivalences. (In fact, this zigzag can always be taken to have only one intermediate diagram.)

Remark 4.14. As a technical aside, we note that in homotopy theory, it is usually considered preferable to consider the variant of Definition 4.13 where we replace homotopy equivalence with *weak homotopy equivalence*; working with weak homotopy equivalence is much more convenient. But that technicality will not be important to us here. In any case, we are primarily interested in spaces having the homotopy type of a CW-complex (a very mild requirement), and here the definitions of homotopy equivalence and weak homotopy equivalence coincide, by Whitehead's theorem.

Remark 4.15. In general, an objectwise homotopy equivalence $f : X \to Y$ does not have a "homotopy inverse," i.e., an objectwise homotopy equivalence $g : Y \to X$ such that $g_r \circ f_r \cong \mathrm{Id}_{X_r}$ and $f_r \circ g_r \cong \mathrm{Id}_{Y_r}$ for all $r \in \mathbb{R}$.

Exercise 4.16. *Prove that if F and G are weakly equivalent, then $H_i F \cong H_i G$ for all $i \geq 0$.*

Definition 4.17 (Nerve Filtration)**.** For any set of $\mathbb{R}$-indexed filtrations U and $U_r := \{F_r \mid F \in U\}$, the indexwise nerves $\mathcal{N}(U_r)$ define a filtration $\mathcal{N}(U)$.

Definition 4.18 (Cover of a Filtration)**.** A set of filtrations U is a *cover* of a filtration F if for all r, U_r is a cover of F_r.

Theorem 4.19 (Persistent Nerve Theorem)**.** *If U is a cover of a filtration F where for each $r \in \mathbb{R}$, U_r and F_r satisfy the hypotheses of either Theorem 4.11 or Theorem 4.10, then F and $\mathcal{N}(U)$ are weakly equivalent.*

Idea of proof. The proof of the nerve theorem outlined above yields homotopy equivalences

$$F_r \xleftarrow{\simeq} Z_r \xrightarrow{\simeq} \mathcal{N}(U_r).$$

These maps are natural with respect r, i.e., they commute with the inclusion maps in the filtration, so they determine objectwise homotopy equivalences.

$$F \xleftarrow{\simeq} Z \xrightarrow{\simeq} \mathcal{N}(U). \qquad \square$$

4.3 Čech and Delaunay Filtrations

For $p \in \mathbb{R}^n$, let $B(p, r)$ be the closed ball of radius r centered at p.

Definition 4.20 (Čech Filtration)**.** For $P \subset \mathbb{R}^n$ finite, define the *Čech Filtration* $\check{\text{C}}\text{ech}(P)_r$ by $\check{\text{C}}\text{ech}(P)_r = \mathcal{N}(U_r)$, where $U_r = \{B(p, r)\}_{p \in P}$.

Proposition 4.21. $\check{\text{C}}\text{ech}(P) \simeq \mathcal{S}^{\uparrow}(d_P)$.

Proof. For each r, U_r is a cover of $\mathcal{S}^{\uparrow}(d_P)_r$ satisfying the hypotheses of the nerve theorem Theorem 4.11, so the result follows from by Theorem 4.19, $\square$

Exercise 4.22. *To illustrate the idea of Remark 4.15, explain why for $P = \{0, 1\} \subset \mathbb{R}$ there is no objectwise homotopy equivalence $\mathcal{S}^{\uparrow}(d_P) \to \check{\text{C}}\text{ech}(P)$.*

As a combinatorial model of the union-of-balls filtration, the Čech filtration is very natural, and it is an important object of study in the TDA literature. However, for computation, the Čech filtration has the key disadvantage that it can be quite large. Relatedly, $\check{\text{C}}\text{ech}(P)$ may have simplices in dimension greater than n.

This motivates the consideration of the *Delaunay filtration* $\text{Del}(P)$, a subfiltration of $\check{\text{C}}\text{ech}(P)$ which is smaller and has simplices in dimension at most n. First, we need the classical definition of the Delaunay Triangulation.

Definition 4.23 (Voronoi diagrams and Delaunay Triangulation)**.** For $P \subset \mathbb{R}^n$ finite and $p \in P$, the *Voronoi cell* of p is the set

$$\text{Vor}(p) := \{x \in \mathbb{R}^n \mid \|p - x\| \leq \|q - x\| \text{ for all } q \in P \setminus \{p\}\}.$$

$\text{Vor}(p)$ is the common solution to a finite set of affine inequalities, so is a convex polytope.

Let

$$\mathrm{Vor}(P) = \{\mathrm{Vor}(p) \mid p \in P\}.$$

$\mathrm{Vor}(P)$ is a closed, convex cover of the plane, whose elements intersect only along their boundaries. The boundaries define a cell decomposition of $\mathbb{R}^n$, which is called the Voronoi diagram.

$\mathcal{N}(\mathrm{Vor}(P))$ is called the *Delaunay triangulation of* P. For generic point sets P, there is a natural embedding of the Delaunay triangulation into $\mathbb{R}^n$, which gives a triangulation of the convex hull of P. (The convex hull of P is the smallest convex set containing P.)

The Delaunay triangulation is known to have at most $\Theta(|P|^{\lceil n/2 \rceil})$ simplices in the worst case. Hence, working with Delaunay triangulations is quite practical in low dimensions, but not in very high dimensions. The topic of computing Delaunay triangulations is a very interesting topic in computational geometry whose history dates back to the early days of the subject. We will not discuss this now.

Definition 4.24 (α-Complex/Delaunay Filtration)**.** Using a construction due to Edelsbrunner, Kirkpatrick, and Seidel (1983), we now endow the Delaunay triangulation with the structure of a filtration. For $p \in P$, let

$$\mathrm{Vor}(p, r) = \mathrm{Vor}(p) \cap B(p, r),$$

and in analogy to the above, let

$$\mathrm{Vor}(P, r) := \{\mathrm{Vor}(p, r) \mid p \in P\}.$$

We define

$$\mathrm{Del}(P)_r := \mathcal{N}(\mathrm{Vor}(P, r)),$$

and call this the *Delaunay complex at radius* r (this is also called an α-complex). Allowing r to vary, we obtain the *Delaunay filtration* $\mathrm{Del}(P)$.

Note that for sufficiently large r, $\mathrm{Del}(P)_r$ is exactly the Delaunay triangulation of P.

Proposition 4.25. *For any finite* $P \subset \mathbb{R}^d$*,*

$$\mathrm{Del}(P) \simeq \mathcal{S}^{\uparrow}(d_P) \simeq \check{\mathrm{C}}\mathrm{ech}(P).$$

In particular, all three filtration have the same barcodes.

Proof. We have that

$$\mathcal{S}^{\uparrow}(d_P)_r = \bigcup_{p \in P} B(p, r) = \bigcup_{p \in P} \mathrm{Vor}(p, r),$$

because every point in $B(p, r) = \bigcup_{p \in P}$ must be closest to some point of P.

Moreover, each $\mathrm{Vor}(p, r)$ is closed and convex, because it is an intersection of a closed ball, which is convex, and a Voronoi region, which is closed an convex. Hence the assumptions of the persistent nerve theorem (Theorem 4.19) hold. This gives $\mathrm{Del}(P) \simeq \mathcal{S}^{\uparrow}(d_P)$. The second equation follows from Proposition 4.21. □

Del(P) is usually much smaller than Čech(P), which makes it computationally very practical in low dimensions.

Remark 4.26 (Computation)**.** The first step in the computation of Del(P) is the computation of the Delaunay triangulation. Then for each simplex σ in Del(P), we compute the radius of the smallest closed ball containing the points indexing σ. This is a standard computation in computational geometry. The computation of Čech(P)(or its k-dimensional skeleton), is similar, except we don't compute the Delaunay triangulation, and we may have to consider many more simplices. Note that although Čech(P) is a much larger object than Del(P), if we are only interested in the filtration up to small r, then computing Čech(P) may be easier than computing Del(P) in high dimensions.

4.4 The Vietoris-Rips Filtration

For data embedded high dimensions or data taking the form of a finite metric space without an embedding into a nice space, it is often preferable to work with the *Vietoris-Rips (VR) filtration*:

Definition 4.27 (Vietoris-Rips Filtration)**.** For $r \in [0, \infty)$, let $N(P)_r$ denote the neighborhood graph of P, i.e., the graph with vertex set P and an edge connecting $p, q \in P$ if and only if $d(p, q) \leq 2r$. We take Rips(P)$_r$ to be the *clique complex* on $N(P)_r$, i.e., the unique largest simplicial complex with 1-skeleton $N(P)_r$.

Exercise 4.28. *For $P \subset \mathbb{R}^n$, clearly* Čech(P)$_r \subset$ Rips(P)$_r$ *for all* $\mathbb{R}$*. Perform the easy check that conversely,* Rips(P)$_r \subset$ Čech(P)$_{2r}$*.*

Remark 4.29. A shown by Ghrist and de Silva, in their paper "Coverage in sensor networks via persistent homology," the result of the exercise can be improved to Rips(P)$_r \subset$ Čech(P)$_{\sqrt{2}r}$. In fact they give a stronger, dimension-dependent bound.

Rips complexes get discussed a lot in TDA, and for many are a favorite example of a filtration constructed from data. The simplicity of the construction is appealing, and there are certainly a lot of data sets where it seems like the natural construction to consider. But given this, successful examples of the use of these filtrations in TDA applications are fewer than one might expect. As I see it, there are several reasons for this:

- There is now very fast and memory-efficient software for computing persistent homology of VR filtrations, namely Ripser and Eirene, but they have appeared only recently.
- Even with the fast software, one can handle only a few thousand points if one wants to build the full VR filtration (as a rough estimate, maybe 5000-20,000 for H_1 on a laptop). But unless the data has very low intrinsic dimensionality one may need more points than that to faithfully capture high-dimensional geometric structure in the data.
- In low dimensions, one can use the Delaunay filtration, which scales better.

- Vietoris-Rips filtrations, and indeed all of the constructions we've discussed so far, are not robust to noise, so even a few outliers might cause problems.

As we will discuss later, the last point is one of the key motivations for multi-parameter persistence.

5 Algebraic Aspects of Multi-Parameter Persistence Modules

To continue with our treatment of 1-parameter persistent homology, the next topics I want to cover are the proof of the structure theorem (in the finitely generated $\mathbb{Z}$-indexed case) and the standard algorithm for computing persistent homology.

But to treat these topics properly, it is best to first develop some algebraic language for working with persistence modules. And since the language we need for the 1-parameter setting is more or less the same as the language we will need for the multi-parameter setting, it will be most efficient to just develop this language in the general setting now. So we will now introduce multi-parameter persistence modules and some of the basic formalism for working with them, then return for some time to the 1-parameter setting.

As in the 1-parameter setting, different indexing categories will be convenient in different settings. However, for now we will restrict attention to persistence modules indexed by products of $\mathbb{N}$.

Definition 5.1 (Persistence modules)**.** For $d \geq 1$, a (d-parameter persistence) module is a functor $\mathbb{N}^d \to \mathbf{Vect}$.

For example, a 2-parameter persistence module (often called a *bipersistence module*) is a diagram of K-vector spaces of the form:

$$\begin{array}{ccccccc}
\vdots & & \vdots & & \vdots & & \\
\uparrow & & \uparrow & & \uparrow & & \\
M_{0,2} & \longrightarrow & M_{1,2} & \longrightarrow & M_{2,2} & \longrightarrow & \cdots \\
\uparrow & & \uparrow & & \uparrow & & \\
M_{0,1} & \longrightarrow & M_{1,1} & \longrightarrow & M_{2,1} & \longrightarrow & \cdots \\
\uparrow & & \uparrow & & \uparrow & & \\
M_{0,0} & \longrightarrow & M_{1,0} & \longrightarrow & M_{2,0} & \longrightarrow & \cdots
\end{array}$$

5.1 d-Parameter Persistent Homology

A (d-parameter) filtration is a functor

$$F : \mathbb{N}^d \to \mathbf{Top}$$

such that $F_x \subset F_y$ whenever $x \leq y$ (in the product partial order on $\mathbb{R}^d$).

In TDA, we obtain d-parameter persistence modules from data via the following partial extension of the persistence pipeline we saw earlier:

$$\boxed{\text{Data}} \to \boxed{d\text{-Parameter Filtration}} \xrightarrow{\text{Homology}} \boxed{d\text{-Parameter Persistence Module}}$$

We'll come back to this version of the pipeline later and talk about how the constructions of filtrations of data we saw earlier extend to give constructions of d-parameter filtrations, $d \geq 2$.

5.2 Very Brief Review of Abstract Algebra

Basic abstract algebra, including familiarity with rings and modules, is a prerequisite for this course. Nevertheless, we briefly review a couple of basic definitions:

- Informally, a *field* is a set F together with multiplication and operations satisfying all the usual axioms of arithmetic. In particular, F has an additive identity 0, a multiplicative identity 1, and each non-zero element $x \in F$ has a multiplicative inverse, i.e., an element $\frac{1}{x}$ such that
$$x \cdot \frac{1}{x} = 1.$$
For example, $\mathbb{Q}$ and $\mathbb{R}$ are fields. So is $\mathbb{Z}/p\mathbb{Z}$ for any prime p.

- A *(commutative) ring* is a set with all the properties of a field, except that non-zero elements needn't have multiplicative inverses. For example, $A^d := K[t_1, \ldots, t_d]$, the polynomial ring in d variables with K coefficients, is a ring, but not a field. $\mathbb{Z}$ is also a ring but not a field. For any integer r, $\mathbb{Z}/r\mathbb{Z}$ is a ring.

- A module over a commutative ring R is an object satisfying exactly the same axioms as an abstract vector space over a field, except that R needn't be a field. In particular, for K a field, a K-vector space and K-module are the same thing.

Note that if M is an R-module and $S \subset R$ is a subring, then M has the structure of an S-module by restricting the action of R on M to S.

5.3 Persistence Modules as d-Graded Modules

As the name *persistence module* suggests, these objects can be seen as modules in the sense of abstract algebra, and sometimes this perspective is very useful.

Since K is a subring of A^d, any A^d-module M has the structure of a K-vector space by restricting the action of A^d on M to K.

Definition 5.2. Let $\mathbf{e}_i$ denotes the i^{th} standard basis vector in $\mathbb{Z}^d$. We say a *d-grading* on a A^d-module M is a vector space decomposition

$$M = \oplus_{z \in \mathbb{N}^d} M_z$$

such that $t_i M_z \subset M_{z+\mathbf{e}_i}$ for all $z \in \mathbb{N}^d$ and $i \in \{1, \ldots, d\}$. An A^d-module M is said to be d-graded if it comes equipped with a d-grading.

A *morphism* $f : M \to N$ of d-graded modules is a module homomorphism (in the usual sense) such that $f(M_z) \subset N_z$ for all $z \in \mathbb{N}^d$. With these morphisms, the d-graded modules form a category **d-mod**.

For M a d-graded module and $m \in M$, we say m is *homogeneous* if $m \in M_z$ for some $z \in \mathbb{N}^d$. By the definition of the direct sum, any $m \in M$ can be written as $m = \sum_{j=1}^{l} m_j$ for some homogeneous elements $m_1, m_2, \ldots, m_l$. A homogeneous submodule of M is one generated by a set of homogeneous elements.

Proposition 5.3 (Carlsson, Zomorodian 2006)**.** *The category* $\text{Fun}(\mathbb{N}^d, \mathbf{Vect})$ *of persistence modules is equivalent (in fact isomorphic) to the category* **d-mod** *of d-graded* A^d*-modules.*

Proof. Define a functor $F : \text{Fun}(\mathbb{N}^d, \mathbf{Vect}) \to \mathbf{d\text{-}mod}$ on objects by $F(M) = \oplus_{z \in \mathbb{N}^d} M_z$, with the action of the polynomial ring specified as follows:

- For $m \in M$ homogenous, $t_i(m) := M_{z,z+\mathbf{e}_i}(m)$ for all $z \in \mathbb{N}^d$ and $i \in \{1, \ldots, d\}$,
- The action of t_i on all of $F(M)$ is then given by linearity. More precisely, if $m = \sum_{j=i}^{l} m_j$ with each m_j homogeneous, then

$$t_i(m) := \sum_{j=1}^{l} t_i(m_j).$$

- Given this, the action of $K[t_1, \ldots, t_d]$ on $F(M)$ is defined via the module axioms in a similar way. For example,

$$(t_1^2 + 2t_2)(m) = t_1(t_1(m)) + 2t_2(m).$$

This indeed gives a well defined d-graded module. Natural transformations $\gamma : M \to N$ induce morphisms $F(\gamma) : F(M) \to F(N)$ in the obvious way.

It remains to check that F is really a functor, and that it in fact is an isomorphism. This is straightforward. □

Exercise 5.4 (Suggested)**.** *Fill in the details of the above proof.*

A^d-modules are the basic objects of study in commutative algebra. They have been studied extensively, and their theory is highly developed. Proposition 5.3 allows us to adapt standard language and constructions for A^d-modules to the study of persistence modules, provided those constructions make sense in the d-graded setting. Fortunately, as a rule of thumb definitions, and arguments in module theory tend to carry over to the d-graded setting (and sometimes become simpler there).

5.4 Free Persistence Modules

For $d \geq 1$ and $a \in \mathbb{N}^d$, let Q^a denote the d-parameter persistence module given by

$$Q^a_z = \begin{cases} K & \text{if } a \leq z, \\ 0 & \text{otherwise,} \end{cases} \qquad Q^a_{y,z} = \begin{cases} \mathrm{Id}_K & \text{if } a \leq y, \\ 0 & \text{otherwise.} \end{cases}$$

Exercise 5.5. *For which $a, b \in \mathbb{N}^d$ is there a non-zero morphism (i.e., natural transformation) $Q^a \to Q^b$?*

For example, $Q^{(1,1)}$ is the following, where all maps between copies of K are the identity:

$$\begin{array}{ccccccc}
\vdots & & \vdots & & \vdots & & \vdots \\
\uparrow & & \uparrow & & \uparrow & & \uparrow \\
0 & \longrightarrow & K & \longrightarrow & K & \longrightarrow & K & \longrightarrow \cdots \\
\uparrow & & \uparrow & & \uparrow & & \uparrow \\
0 & \longrightarrow & K & \longrightarrow & K & \longrightarrow & K & \longrightarrow \cdots \\
\uparrow & & \uparrow & & \uparrow & & \uparrow \\
0 & \longrightarrow & K & \longrightarrow & K & \longrightarrow & K & \longrightarrow \cdots \\
\uparrow & & \uparrow & & \uparrow & & \uparrow \\
0 & \longrightarrow & 0 & \longrightarrow & 0 & \longrightarrow & 0 & \longrightarrow \cdots
\end{array}$$

Definition 5.6. We say a d-parameter persistence module F is *free* if there exists a multiset $\mathcal{A}$ of elements in $\mathbb{N}^d$ such that $F \cong \oplus_{a \in \mathcal{A}} Q^a$.

Example 5.7. The free module $Q^{(2,1)} \oplus Q^{(1,2)}$ is given by the following diagram, where all maps between two copies of the same vector space are the identity:

$$\begin{array}{ccccccc}
\vdots & & \vdots & & \vdots & & \vdots \\
\uparrow & & \uparrow & & \uparrow & & \uparrow \\
0 & \longrightarrow & k & \longrightarrow & k^2 & \longrightarrow & k^2 & \longrightarrow \cdots \\
\uparrow & & \uparrow & & \uparrow & & \uparrow \\
0 & \longrightarrow & k & \xrightarrow{\binom{1}{0}} & k^2 & \longrightarrow & k^2 & \longrightarrow \cdots \\
\uparrow & & \uparrow & & \uparrow{\scriptstyle \binom{0}{1}} & & \uparrow \\
0 & \longrightarrow & 0 & \longrightarrow & k & \longrightarrow & k & \longrightarrow \cdots \\
\uparrow & & \uparrow & & \uparrow & & \uparrow \\
0 & \longrightarrow & 0 & \longrightarrow & 0 & \longrightarrow & 0 & \longrightarrow \cdots
\end{array}$$

5.5 Basic Definitions and Constructions

In what follows, we describe some basic module-theoretic constructions for persistence modules. Most often, I will explain these directly in functorial language, but it should be understood

that the constructions are (up to isomorphism) just d-graded versions of standard module constructions.

Let M be a d-parameter persistence module.

Definition 5.8. A *submodule* N of a persistence module M is a collection of vector spaces $\{N_z \subset M_z\}_{z\in\mathbb{N}^d}$, such that $M_{y,z}(N_y) \subset N_z$. The restrictions of the internal maps $M_{y,z}$ give N the structure of a persistence module.

Exercise 5.9 (Suggested)**.** *Check that under the isomorphism of Proposition 5.3, submodules of persistence modules correspond to homogeneous submodules of d-graded modules.*

Exercise 5.10. *For which $a, b \in \mathbb{N}^d$ do we have $Q^a \subset Q^b$? (This exercise is similar to Exercise 5.5. We covered the $d = 1$ case in class.)*

Remark 5.11. Given a morphism of persistence modules $f : M \to N$, the submodules $\ker f \subset M$ and $\operatorname{im} f \subset N$ are well defined, and are defined indexwise, i.e., $(\ker f)_z = \ker(f_z)$ and $(\operatorname{im} f)_z = \operatorname{im}(f_z)$.

Exercise 5.12. *Check that* $\ker f$ *and* $\operatorname{im} f$ *are in fact well-defined submodules.*

We next define quotients of persistence modules. First we need the following simple fact from linear algebra.

Lemma 5.13 (Induced Maps on Quotients of Vector Spaces)**.** *Given vector spaces V, W, subspaces $V' \subset V$, $W' \subset W$, and a linear map $f : V \to W$ with $f(V') \subset W'$, f descends to a map $\bar{f} : V/V' \to W/W'$.*

Exercise 5.14. *Prove Lemma 5.13.*

Definition 5.15. Let M be a persistence module and $N \subset M$ be a submodule. The *quotient* M/N is given by $(M/N)_z = M_z/N_z$, with the internal maps of M/N the induced maps on quotients. (By Lemma 5.13, these internal maps are well defined.)

Example 5.16.

- Q^0/Q^1 is isomorphic to the following:

$$K \longrightarrow 0 \longrightarrow 0 \longrightarrow 0 \longrightarrow \cdots$$

- In fact, an $\mathbb{N}$-indexed interval module (see Definition 3.1) is isomorphic either to Q^a for some $a \in \mathbb{N}$, or to Q^a/Q^b for some $a < b \in \mathbb{N}$.

- $Q^{0,0}/Q^{1,1}$ is isomorphic to the following:

$$\begin{array}{ccccccc}
\vdots & & \vdots & & \vdots & & \\
\uparrow & & \uparrow & & \uparrow & & \\
K & \longrightarrow & 0 & \longrightarrow & 0 & \longrightarrow & \cdots \\
\uparrow & & \uparrow & & \uparrow & & \\
K & \longrightarrow & 0 & \longrightarrow & 0 & \longrightarrow & \cdots \\
\uparrow & & \uparrow & & \uparrow & & \\
K & \longrightarrow & K & \longrightarrow & K & \longrightarrow & \cdots
\end{array}$$

- $Q^{0,0}/Q^{1,0}$ is isomorphic to the following:

$$\begin{array}{ccccccc}
\vdots & & \vdots & & \vdots & & \\
\uparrow & & \uparrow & & \uparrow & & \\
K & \longrightarrow & 0 & \longrightarrow & 0 & \longrightarrow & \cdots \\
\uparrow & & \uparrow & & \uparrow & & \\
K & \longrightarrow & 0 & \longrightarrow & 0 & \longrightarrow & \cdots \\
\uparrow & & \uparrow & & \uparrow & & \\
K & \longrightarrow & 0 & \longrightarrow & 0 & \longrightarrow & \cdots
\end{array}$$

Given modules W and W' of a module M, let $W + W' \subset M$ be given by

$$(W + W')_z = \{w + w' \mid w \in W_z,\ w \in W'_z\}.$$

This is clearly a submodule of M.

Exercise 5.17 (Already covered in class)**.**

1. *Draw the diagram of vector spaces $Q^{0,1} + Q^{1,0}$.*
2. *Up to isomorphism, what is the module $Q^{0,0}/(Q^{0,1} + Q^{1,0})$?*

If $v \in M_z$ we write $\mathrm{gr}(v) = z$.

Definition 5.18. We say that $S \subset \bigcup_{z \in \mathbb{N}^d} M_z$ is *a set of generators* for M if for any $v \in \bigcup_{z \in \mathbb{N}^d} M_z$,

$$v = \sum_{i=1}^{k} c_i M_{\mathrm{gr}(v_i), gr(v)}(v_i)$$

for some $v_1, v_2, \ldots, v_k \in S$ and scalars $c_1, \ldots, c_k \in K$. We say M is *finitely generated* if there exists a finite set of generators for M.

Exercise 5.19. *Give an example of a 1-parameter persistence module whose vector spaces have dimension at most one, and for which any minimal generating set has two elements.*

Exercise 5.20. *Give an example of a 1-parameter persistence module which is not finitely generated, but whose vector spaces are all of finite dimension.*

5.6 Bases of Free Persistence Modules

Many of the standard ideas of linear algebra adapt in a straightforward way to free persistence modules. For example:

Definition 5.21. We define a *basis* for a free persistence module F to be a minimal homogenous set of generators.

Remark 5.22. Letting 1^a denote the multiplicative identity of $Q^a_a = K$, we have that $\{1^a\}$ is a basis for Q^a. Moreover, if F and G are free modules with basis A and B, then $A \cup B$ is a basis for $A \oplus B$. Hence, for $\mathcal{A}$ a multi-set, we get a basis for a free module $F = \oplus_{a \in \mathcal{A}} Q^a$ by choosing one element for each summand Q^a. We say that the *standard basis* for F is the one obtained by taking each such element to be 1^a.

Though as in linear algebra, bases are usually not unique, the following is true:

Proposition 5.23. *The number of elements at each grade in a basis for a finitely generated free persistence module is independent of the choice of basis.*

Remark 5.24. I suspect that this result extends to a version for arbitrary free persistence modules, without the finite generation hypothesis. In fact, this is immediate in the $\mathbb{N}$-indexed case; the proof is this same. But in the $\mathbb{Z}$-indexed case, I have not checked whether the result holds.

To prove this, we need a multi-graded version of a standard result from commutative algebra called Nakayama's lemma. In order to minimize the amount of notation we need, we will state the result just in the special case we need we need to prove the proposition. (The proof of the more general statement is the same.) For this argument, I am loosely following chapter 1 of Peeva's text "Graded Syzygies," but this material is standard.

For any persistence module M define the submodule M° by

$$M^\circ_z = \langle m \in M_z \mid m = M_{y,z}(m') \text{ for some } y < z \in \mathbb{N}^d \rangle.$$

Thus M° is the submodule consisting of vectors which are "shifts" of vectors at lower indices.

Note that all of the internal maps in the quotient M°/M are trivial.

Example 5.25. For any $a \in \mathbb{N}^d$, $(Q^a)^\circ = Q^{a+\mathbf{e}_1} + Q^{a+\mathbf{e}_2} + \cdots + Q^{a+\mathbf{e}_d}$. For example, $(Q^{(1,1)})^\circ$ is given by the following diagram, where maps between copies of K are the identity:

$$\begin{array}{ccccccc}
\vdots & & \vdots & & \vdots & & \vdots \\
\uparrow & & \uparrow & & \uparrow & & \uparrow \\
0 & \longrightarrow & K & \longrightarrow & K & \longrightarrow & K & \longrightarrow \cdots \\
\uparrow & & \uparrow & & \uparrow & & \uparrow \\
0 & \longrightarrow & K & \longrightarrow & K & \longrightarrow & K & \longrightarrow \cdots \\
\uparrow & & \uparrow & & \uparrow & & \uparrow \\
0 & \longrightarrow & 0 & \longrightarrow & K & \longrightarrow & K & \longrightarrow \cdots \\
\uparrow & & \uparrow & & \uparrow & & \uparrow \\
0 & \longrightarrow & 0 & \longrightarrow & 0 & \longrightarrow & 0 & \longrightarrow \cdots
\end{array}$$

Lemma 5.26 (Multigraded Nakayama's Lemma)**.** *Let M be a finitely generated persistence module.*

(i) If $M^\circ = M$, then $M = 0$.

(ii) If $W \subset M$ is a submodule and $M^\circ + W = M$, then $W = M$.

(iii) If the images of vectors $V = \{v_1, \dots v_k\} \subset M$ in M/M° generate M/M°, then V generates M.

Proof. Let $V \subset \bigcup_{z \in \mathbb{Z}^d} M_z$ be a finite set of generators for M. Assuming $M \neq 0$, choose a minimal element y from the set $\{\mathrm{gr}(v) \mid v \in V\}$. Clearly, $M_y^\circ = 0 \neq M_y$, a contradiction. This proves (i).

To prove (ii), we consider the quotient module M/W. The exercise below asks you to check that $M^\circ + W = M$ then $(M/W)^\circ = M/W$. Now, applying part (i) to (M/W), we find that $M/W = 0$, i.e., $M = W$.

To prove (iii), let $\langle V \rangle \subset M$ be the submodule generated by V. Then since $\langle V \rangle$ descends to a set of generators for M/M°, it follows from the defition of a quotient that $\langle V \rangle + M^\circ = M$. By (ii), we have that $\langle V \rangle = M$, i.e., V generates M. □

Exercise 5.27 (Suggested)**.** *Check the step in the above proof that if $M^\circ + W = M$ then $(M/W)^\circ = M/W$.*

Lemma 5.28. *If M is a finitely generated persistence module, then a minimal set of generators for M descends to a basis for M/M°.*

Proof. Let B be a minimal set of generators for M. Clearly, B descends to a set of generators $\bar{B}$ for (M/M°). If $\bar{B}$ is not a basis, a proper subset $B' \subset B$ descends to a set of generators for M/M°. Part (*iii*) of Nakayama's lemma then gives us that B' generates M, which contradicts the minimality of B. □

Proof of Proposition 5.23. By Lemma 5.28, a basis B for F descends to a basis for F/F°. Equivalently, letting

$$B_z := \{b \in B \mid \mathrm{gr}(b) = z\},$$

we have that for each z, B_z descends to a basis for $(F/F^\circ)_z$. Since the number of elements of a basis of a finite-dimensional vector space is independent of the choice of basis, is $|B_z|$ independent of the choice of B. This gives the result. □

5.7 Matrix Representation of Morphisms Between Free Modules

Let's first recall some basic linear algebra: Let B be an ordered basis of a finitely dimensional vector space V. Denote the i^{th} element of B as B_i. We can represent a vector $v \in V$ with respect to B as a vector $[v]^B \in K^{|B|}$; we take $[v]^B$ to be the unique vector such that

$$v = \sum_i [v]_i^B B_i.$$

Along similar lines, for B' an ordered basis for a vector space W, we represent a linear map $\gamma : V \to W$ via a matrix $[\gamma]^{B',B}$ with coefficients in the field K, by taking the j^{th} column of $[\gamma]^{B',B}$ to be $[\gamma(B_j)]^{B'}$.

Now let's adapt this story to free modules: Let B be an ordered basis of a finitely generated free persistence module F. For $z \in \mathbb{N}^d$, we can represent a vector $v \in F_z$ with respect to B as a vector $[v]^B \in K^{|B|}$; we take $[v]^B$ to be the unique vector such that $[v]^B_i = 0$ if $\text{gr}(B_i) \not\leq z$ and

$$v = \sum_{i:\text{gr}(B_i)\leq z} [v]^B_i F_{\text{gr}(B_i),z} B_i.$$

Thus, $[v]^B$ records the field coefficients in the linear combination of B giving v.

Example 5.29. Let $F = Q^{(1,0)} \oplus Q^{(0,1)}$. Then $F_{0,2} = K$. Let B be the standard basis for F. Then for $v = 1 \subset F_{0,2}$, we have $[v]^B = (0\ 1)^T$.

Along similar lines, for B' a finite ordered basis of a free persistence module F', we represent a morphism $\gamma : F \to F'$ via a matrix $[\gamma]^{B',B}$ with coefficients in the field K, with each row and column labeled by an element of $\mathbb{N}^d$, as follows:

- The j^{th} column of $[\gamma]^{B',B}$ is $[\gamma(B_j)]^{B'}$.
- The label of the j^{th} column is $\text{gr}(B_j)$,
- The label i^{th} row is $\text{gr}(B'_i)$.

Where no confusion is likely, we sometimes write $[\gamma]^{B',B}$ simply as $[\gamma]$.

Example 5.30. Let $F = Q^{(1,0)} \oplus Q^{(0,1)}$, $G = Q^{(0,0)}$ and consider the morphism $\gamma : F \to G$ whose restriction to each summand is the inclusion. Then with respect to the standard bases,

$$[\gamma] = \begin{array}{c} \\ (0,0) \end{array} \begin{array}{c} \begin{array}{cc} (1,0) & (0,1) \end{array} \\ \begin{pmatrix} 1 & 1 \end{pmatrix} \end{array}.$$

To explain where the entries of γ come from, note that by definition, $\gamma_{1,1}$ is the element of K which solves

$$\gamma(1^{(1,0)}) = \gamma_{1,1} G_{(0,0),(1,0)}(1^{(0,0)}).$$

It's easy to see that

$$\gamma(1^{(1,0)}) = G_{(0,0),(1,0)}(1^{(0,0)}) = 1 \in K = G_{(0,1)}.$$

Thus, we must have $\gamma_{1,1} = 1$. Essentially the same argument shows that $\gamma_{1,2} = 1$ as well.

5.8 Column and Row Operations as Change of Basis

We start by recalling a basic fact from linear algebra:

Proposition 5.31. *If $B = \{b_1, \ldots, b_k\}$ is a basis for a vector space V, then for any $i \neq j$ and $c \in K$, the set obtained from B by replacing b_i with $b_i + cb_j$ is also a basis for V.*

We introduce the following notation: For a matrix M, $M_{i,*}$ will denote the i^{th} row of M, and $M_{*,i}$ will denote the i^{th} column of M.

One sees the following proposition in undergraduate linear algebra:

Proposition 5.32. *Let $\gamma : V \to W$ be a linear map of finite-dimensional vector spaces, and let*

$$A = \{a_1, \ldots, a_n\} \quad \text{and} \quad B = \{b_1, \ldots, b_m\}$$

be bases for V and W, respectively. For any $c \in K$,

- *If A' is obtained from A by replacing a_i with $a_i + ca_j$, then $[\gamma]^{B,A'}$ is obtained from $[\gamma]^{B,A}$ by adding $c[\gamma]^{B,A}_{*,j}$ to $[\gamma]^{B,A}_{*,i}$.*
- *Similarly, if B' is obtained from B by replacing b_i with $b_i + cb_j$, then $[\gamma]^{B',A}$ is obtained from $[\gamma]^{B,A}$ by subtracting $c[\gamma]^{B,A}_{i,*}$ from $[\gamma]^{B,A}_{j,*}$.*

Exercise 5.33 (Suggested)**.** *Prove Proposition 5.32.*

Example 5.34. Consider the identity map $\mathrm{Id} : \mathbb{R}^3 \to \mathbb{R}^3$. Let B denote the standard basis for $\mathbb{R}^3$, i.e., $B = \{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$. Then clearly $[\mathrm{Id}]^{B,B}$ is the 3x3 identity matrix. Let $B' = \{\mathbf{e}_1, \mathbf{e}_1 + \mathbf{e}_2, \mathbf{e}_3\}$ Then

$$[\mathrm{Id}]^{B,B'} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$[\mathrm{Id}]^{B',B} = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

These linear algebra results extend readily to finitely generated free modules:

Proposition 5.35. *If $B = \{b_1, \ldots, b_k\}$ is a basis for a free module F, then for any $i \neq j$ with $\mathrm{gr}(b_i) \geq \mathrm{gr}(b_j)$ and $c \in K$, the set B' obtained from B by replacing b_i with $b_i + cF_{\mathrm{gr}(b_j),\mathrm{gr}(b_i)}(b_j)$ is also a basis for F.*

Example 5.36. The standard basis for $F := Q^0 \oplus Q^1$ is $\{1^0, 1^1\}$. According to Proposition 5.35, $\{1^0, 1^1 + F_{0,1}(1^0)\}$ is also a basis for F.

Proof of Proposition 5.35. It's easy to check that $b_i \in \langle B' \rangle$, so B' generates F. By Lemma 5.28, B descends to a basis for F/F°. Given this, it follows from Proposition 5.31 that B' also descends to a basis for F/F°. But if B' is a non-minimal set of generators, then B' descends to a non-minimal set of generators for F/F°, a contradiction. Hence B' is a basis for F. $\square$

Proposition 5.37. *Let $\gamma : F \to G$ be a morphism of finitely generated free modules, and let*

$$A = \{a_1, \ldots, a_n\} \quad \text{and} \quad B = \{b_1, \ldots, b_m\}$$

be bases for F and G, respectively. For any $c \in K$,

- *If $\mathrm{gr}(a_j) \leq \mathrm{gr}(a_i)$ and A' is obtained from A by replacing a_i with*

$$a_i + cF_{\mathrm{gr}(a_j),\mathrm{gr}(a_i)}(a_j),$$

  then $[\gamma]^{B,A'}$ is obtained from $[\gamma]^{B,A}$ by adding $c[\gamma]^{B,A}_{*,j}$ to $[\gamma]^{B,A}_{*,i}$.

- *If $\mathrm{gr}(b_j) \leq \mathrm{gr}(b_i)$ and B' is obtained from B by replacing b_i with*

$$b_i + cG_{\mathrm{gr}(b_j),\mathrm{gr}(b_i)}(b_j),$$

  then $[\gamma]^{B',A}$ is obtained from $[\gamma]^{B,A}$ by subtracting $c[\gamma]^{B,A}_{i,*}$ from $[\gamma]^{B,A}_{j,*}$.

The upshot of Proposition 5.37 is that for a matrix representing a morphism of free modules,

- Adding a column of lower grade to one of equal or higher grade can be interpreted as a change of basis.
- Adding a row of higher grade to one of equal or lower grade can be interpreted as a change of basis.
- Other column and row additions cannot be interpreted as a change of basis.

We will use these ideas to prove the structure theorem for 1-D persistence modules, and to explain why the standard algorithm for computing 1-parameter persistent homology is correct.

5.9 Presentations of persistence modules

A *presentation* of a persistence module M is simply a morphism $\gamma : F \to F'$ of free modules such that $M \cong F'/\operatorname{im}(\gamma)$. If F and F' are both finitely generated, then we can represent γ with respect to a choice of bases for F and F' as a matrix P with each of its rows and columns labeled by an element of $\mathbb{N}^d$. We will call P a *presentation matrix* for M, or by slight abuse of terminology, simply a presentation. Note that P may have 0 columns or 0 rows.

We will denote the labels of the i^{th} row and j^{th} column of P by $\mathrm{gr}(P_{i,*})$ and $\mathrm{gr}(P_{*,j})$, respectively.

Proposition 5.38. *If M is finitely generated, then there exists a presentation $\gamma : F \to F'$ of M with F and F' finitely generated.*

Proposition 5.38 is important because it makes clear that if M is finitely generated, then there exists a presentation matrix for M.

The proof of Proposition 5.38 uses the following result.

Proposition 5.39. *Any submodule of finitely generated persistence module is finitely generated.*

Sketch of proof. As earlier, let $A^d := K[t_1, \ldots, t_n]$. Given the correspondence between d-parameter persistence modules and d-graded A^d-modules, it suffices to prove that any submodule of a finitely generated A^d-module is finitely generated.[5] This is a standard fact in commutative algebra, following from the fact that A^d is a *Noetherian ring*, i.e., every ideal of A^d is finitely generated. See the discussion of Hilbert's basis theorem in Chapter 1 of Eisenbud's text "Commutative Algebra with a View Towards Algebraic Geometry" for short, self-contained proofs of these results.

□

Proof of Proposition 5.38. Choose finite set of generators $\{g_1, \ldots, g_m\}$ for M. Consider the free module

$$F' = \oplus_{i=1}^{m} Q^{\mathrm{gr}(g_i)}.$$

We define a morphism $p : F' \to M$ sending the canonical generator of $1^{\mathrm{gr}(g_i)} \in Q^{\mathrm{gr}(g_i)}_{\mathrm{gr}(g_i)}$ to g_i for each i. (You should check that this in fact yields a well-defined morphism.) Clearly, this morphism is surjective (that is, p_z is surjective for each $z \in \mathbb{N}^d$.). By Proposition 5.39, $\ker p$ is finitely generated. We choose a finite set of generators for $\ker p$, and proceeding exactly as above, we obtain a finitely generated free module F and a surjective map $\gamma : F \to F'$ with $\operatorname{im} \gamma = \ker p$. Then $F'/\operatorname{im} \gamma \cong M$ by the first isomorphism theorem. □

Remark 5.40. Presentations usually not unique. We will discuss the important idea of a *minimal* presentation later in this course. But even this is not unique, though its dimensions are unique and its labels are unique, up to permutation.

Example 5.41. For any d, the 0×0 matrix is a presentation matrix of 0, the trivial d-parameter persistence module. This represents the presentation $0 \to 0$.

For any $a \in \mathbb{N}^d$,

$$\begin{array}{cc} & a \\ a & (1) \end{array}.$$

is also a presentation matrix of 0. This corresponds to the presentation $Q^a \xrightarrow{\mathrm{Id}} Q^a$, whose cokernel is 0.

[5] There is one subtlety here: This sufficiency statement assumes that if a homogeneous submodule N of a d-graded A^d-module M is generated by a finite set S, then it is in fact generated by a finite set of *homogeneous elements*. But this is true: Indeed, N is generated by a set of homogeneous elements T, so every element of $s \in S$ is generated by a finite subset of $T^s \subset T$. Thus, N is generated by the finite set $\cup_{s \in S} T^s$.

Example 5.42 (Presentations of Interval Modules)**.** For this next example, we restrict attention to the case $d = 1$. For $a < b \in \mathbb{N} \cup \{\infty\}$, let

$$[a, b) := \{z \in \mathbb{N} \mid a \leq z < b\}.$$

Then letting $K^{[a,b)}$ the interval module defined in Definition 3.1, we have that $K^{[a,b)} \cong Q^a/Q^b$ if $b < \infty$, and $K^{[a,b)} = Q^a$ if $b = \infty$. Thus, if $b < \infty$, the inclusion $Q^b \hookrightarrow Q^a$ is a presentation for $K^{[a,b)}$. With respect to the standard bases, this is represented by the labeled matrix:

$$\begin{array}{c} \\ a \end{array} \begin{array}{c} b \\ \begin{pmatrix} 1 \end{pmatrix} \end{array}.$$

The 1×0 matrix with row-label a is a presentation for $K^{[a,\infty)} = Q^a$. Conversely, each 1×1 or 1×0 presentation matrix specifies an interval module, or a trivial module (up to isomorphism).

Remark 5.43 (Presentations of direct sums)**.** If labeled matrices P and Q are presentations for persistence modules M and N, then the block diagonal matrix

$$\begin{pmatrix} P & 0 \\ 0 & Q \end{pmatrix},$$

with the row and column labels induced by P and Q in the obvious way, is a presentation for M and N. Conversely, any block-diagonal presentation matrix for a module M specifies a decomposition of M into summands.

Example 5.44. A presentation of $K^{[0,3)} \oplus K^{[1,2)}$ is given by

$$\begin{array}{c} \\ 0 \\ 1 \end{array} \begin{array}{c} \begin{array}{cc} 3 & 2 \end{array} \\ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \end{array}$$

Permuting rows or columns always yields a new presentation matrix; this corresponds to permuting the bases of the free modules. Thus, for example, another presentation for $K^{[0,3)} \oplus K^{[1,2)}$ is given by

$$\begin{array}{c} \\ 0 \\ 1 \end{array} \begin{array}{c} \begin{array}{cc} 2 & 3 \end{array} \\ \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \end{array}$$

6 Proof of the Structure Theorem for Finitely Generated $\mathbb{N}$-Indexed Persistence Modules

We now prove Theorem 3.2, the structure theorem for persistence modules, in the special case of finitely generated $\mathbb{N}$-indexed modules. (The proof for finitely generated $\mathbb{Z}$-indexed modules is exactly the same, in fact.). For convenience we give the statement that we will prove here

Theorem. *If M is an $\mathbb{N}$-indexed or $\mathbb{Z}$-indexed p.f.d. persistence module, then there exists a unique multiset of intervals $\mathcal{B}_M$ in $\mathbb{N}$ such that*

$$M \cong \oplus_{I \in \mathcal{B}_M} K^I.$$

We give an algorithmic proof that will lead naturally into a discussion of the standard algorithm for computing persistent homology. One way to prove the standard structure theorem for finitely generated modules over a principle ideal domain is by putting a presentation matrix into *Smith normal form*, and our proof is a variant of this argument, specialized to the case of persistence modules.

6.1 Existence of Decomposition into Interval Modules

Definition 6.1. We'll say a matrix is in *normal form* if it has at most one non-zero entry in each row and each column.

Lemma 6.2. *A finitely generated 1-parameter persistence module M decomposes as a sum of interval modules if and only if there is a presentation matrix P for M in normal form.*

Proof. This follows from Example 5.42 and Remark 5.43. □

Proof of Existence Part of the Structure Theorem. By Lemma 6.2 it suffices to show that there is a presentation matrix for M in normal form. Let P be any presentation matrix for M, with the row labels and column labels both in increasing order. Such P exists by Proposition 5.38. We will show how to compute a presentation in normal form by applying column and row additions to P. It follows from Proposition 5.37 that adding a scalar multiple of column i to column $j \neq i$ corresponds to a change of basis operation if $i < j$, and that adding a scalar multiple of row i to row $j \neq i$ corresponds to a change of basis operation if $i > j$. Thus, performing "rightward" column additions and "upward" row additions on the presentation matrix yields another presentation matrix for M , but other row and column additions may result in a labeled matrix which is no longer a presentation matrix for M. Thus, it is enough to show that an arbitrary matrix can be transformed into a normal one by rightward column additions and upward row additions. We show this in what follows (Proposition 6.6). □

Remark 6.3. This is not the only way to prove the existence portion of the structure theorem. In particular, it is not necessary to frame the argument in terms of matrices. One nice alternative argument, due to Greg Henselmen-Petrusek, hinges on the very nice observation that given two filtrations F, G of a single finite-dimensional vector space V there is choice of basis B for V such that any subspace of either F or G is the span of a subset of B.

I have chosen to present the argument here because of its close connection with computational material which I in any case want to cover.

6.2 The "Standard Reduction"

We introduce a variant of Gaussian elimination that is also the basis for the standard algorithm for computing persistent homology. We call this the *standard reduction*. As we present it, it involves only column additions. (Symmetrically, one could take it to only involve row additions.)

Let R be an $m \times n$ matrix with coefficients in K. For $j \in \{1, \ldots, n\}$, define the *pivot* of $R_{*,j}$ by

$$\rho_j^R := \begin{cases} \mathsf{null} & \text{if } R_{*,j} = 0, \\ \max\{i \mid R(i,j) \neq 0\} & \text{otherwise.} \end{cases}$$

We say R is *reduced* if $\rho_j^R \neq \rho_k^R$ whenever $j \neq k$ are the indices of non-zero columns in R. Note that if R is reduced, then all columns are linearly independent, so $\operatorname{Rank} R$ is simply the number of non-zero columns of R.

The *standard reduction* takes any matrix D and performs left-to-right column additions to transform D into a reduced matrix R. As we will explain, this algorithm underlies standard computations of persistent homology. It was introduced by Carlsson and Zomorodian in their 2005 paper "Computing Persistent Homology."

Algorithm 1 The Standard Reduction (Outline)

Input: An $m \times n$ matrix D
Output: A reduced $m \times n$ matrix R obtained from D by left-to-right column additions

```
1: R ← D
2: for j = 1 to n do
3:     while ∃ k < j such that null ≠ ρ_j^R = ρ_k^R do
4:         add −(R(ρ_j^R, j) / R(ρ_j^R, k)) R_{*,k} to R_{*,j}
```

We will not worry yet about the details of how this while loop is implemented, or about other low-level details about the algorithm such as how the matrices are stored. The important point for now is that one can always transform a matrix into a reduced one by left-to-right column additions.

Example 6.4. Here and in many examples that follow, I will work with the field $K = \mathbb{Z}/2\mathbb{Z}$. This is the most common choice of field in TDA, and working with this makes the matrix arithmetic simpler. Consider the matrix

$$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

The standard reduction performs the following sequence of column operations:

$$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \xrightarrow{\text{Add col. 1 to col. 2}} \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$

$$\xrightarrow{\text{Add col. 1 to col. 3}} \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix} \xrightarrow{\text{Add col. 2 to col. 3}} \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}.$$

6.3 Putting a Reduced Matrix into Normal Form via Row Operations

Next we explain how to transform a reduced matrix R into normal form by upwards row additions. This is simple:

Algorithm 2 Normalize: Put Reduced Matrix in Normal Form

Input: An $m \times n$ reduced matrix R
Output: A matrix N in normal form, obtained from R by upward row additions.

```
N ← R
for i = m to 1 (in decreasing order) do
    if ∃ a column j of N whose pivot is i then
        for k = i − 1 to 1 (in decreasing order) do
            add −(N_{k,j}/N_{i,j}) N_{i,∗} to N_{k,∗}
```

Example 6.5. We put the reduced matrix that we computed in Example 6.4 in to normal form:

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \xrightarrow{\text{Add row 3 to row 1}} \begin{pmatrix} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \xrightarrow{\text{Add row 3 to row 1}} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

Proposition 6.6. *Applying Algorithm 1 followed by Algorithm 2 puts an arbitrary matrix in normal form, using only rightward column additions and upward row additions.*

Remark 6.7. There is a symmetry here: We could have instead performed the standard reduction on the rows, and then applied Algorithm 2 to the columns. We will return to this point later.

6.4 Uniqueness of the Barcode

The uniqueness of the barcode can be seen as a corollary of a more general result about uniquness of direct sum decompositions, which (I believe) is due to Azumaya. But we will give an elementary proof.

Proof of uniqueness part of the structure theorem. Let M be finitely generated $\mathbb{N}$-indexed persistence module. We have shown that there exists a finite multiset of intervals $\mathcal{B}_M$ in $\mathbb{N}$ such that $M \cong \oplus_{J \in \mathcal{B}_M} K^{[a,b)}$. We need to show that such $\mathcal{B}_M$ is unique. To do so, for each interval $J \in \mathbb{N}$, we give a formula for the number of copies of J in $\mathcal{B}_M$, such that this formula depends only on the isomorphism type of M (and hence not on the choice of decomposition):

- For any $a < b \in \mathbb{N}$, the number of copies of $[a, b)$ in $\mathcal{B}_M$ is

$$\operatorname{Rank} M_{a,b-1} - \operatorname{Rank} M_{a,b} - \operatorname{Rank} M_{a-1,b-1} + \operatorname{Rank} M_{a-1,b}.$$

- For any $a \in \mathbb{N}$, the number of copies of $[a, \infty)$ in $\mathcal{B}_M$ is

$$\lim_{b\to\infty} \operatorname{Rank} M_{a,b} - \lim_{b\to\infty} \operatorname{Rank} M_{a-1,b}.$$

We now explain why these formulae are true: For any interval $J \subset \mathbb{N}$, let $C(J)$ denote the number of intervals in $\mathcal{B}_M$ containing J. To establish the first formula, note that for any $a < b \in \mathbb{N}$, $C([a, b)) = \operatorname{Rank} M_{a,b-1}$. Now clearly, the number of copies of $[a, b)$ in B_M is

$$C([a, b)) - C([a, b + 1)) - C([a - 1, b)) + C([a - 1, b + 1)).$$

The first formula follows. To establish the second formula, note that for any $a \in \mathbb{N}$, the number of intervals in $\mathcal{B}_M$ containing $[a, \infty)$ is

$$\lim_{b\to\infty} \operatorname{Rank} M_{a,b}.$$

The number of copies of $[a, \infty)$ in $\mathcal{B}_M$ is

$$C([a, \infty)) - C([a - 1, \infty)).$$

The second formula follows. □

7 Computing Persistent Homology

It follows from Example 5.42 and Remark 5.43 that a presentation matrix P for M in normal form encodes $\mathcal{B}_M$ as follows:

$$\mathcal{B}_M = \{[\operatorname{gr}(P_{i,*}), \operatorname{gr}(P_{*,j})) \mid P_{i,j} \neq 0 \text{ and } \operatorname{gr}(P_{i,*}) < \operatorname{gr}(P_{*,j})\} \cup \{[\operatorname{gr}(P_{i,*}), \infty) \mid P_{i,*} = 0\}. \quad (1)$$

Example 7.1. Let M be a module having presentation matrix:

$$\begin{array}{c} \\ 0 \\ 1 \\ 2 \end{array} \begin{array}{c} \begin{array}{ccc} 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \end{array}$$

Note that as an unlabeled matrix, this is the same as the one of Example 6.4. Thus, in view of that example and Example 6.5, the following is a presentation matrix for M in normal form:

$$\begin{array}{c} \\ 0 \\ 1 \\ 2 \end{array} \begin{array}{c} \begin{array}{ccc} 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \end{array}$$

Thus, in view of (1), the barcode of M is given by $\mathcal{B}_M = \{[1, 3), [4, 0)\}$.

In fact, one can read the barcode directly of a reduced presentation matrix for M, without computing the normal form, because of the following:

Proposition 7.2.

(i) Upwards row additions do not change the pivots of a matrix.

(ii) In particular, when we apply Algorithm 2 to a reduced matrix R, the pivots entries of R become the non-zero entries of the resulting normal form.

Thus given a presentation matrix for M, to obtain B_M it's enough to run the standard reduction to obtain a reduced presentation matrix P, and use the following formula:

$$\begin{aligned}\mathcal{B}_M = \{[\mathrm{gr}(\rho_j^P), \mathrm{gr}(P_{*,j})) \mid P_{*,j} \neq 0 \text{ and } \mathrm{gr}(\rho_j^P) < \mathrm{gr}(P_{*,j})\} \\ \cup \{[\mathrm{gr}(P_{i,*}), \infty) \mid i \text{ is not the pivot of any column}\}.\end{aligned} \tag{2}$$

However, in practical applications of TDA, one does not have a presentation of M to start; instead one has chain complex $C_\bullet$ such that M is (isomorphic) to a homology module of $C_\bullet$. One usually does not precompute a presentation of the homology module, but rather computes the barcode of M directly from $C_\bullet$. As it's usually carried out, the computation still amounts to an application of the standard reduction. In what follows we will explain all of this in detail.

7.1 Notation for simplicial chain complexes

For S a finite simplicial complex, let

$$C(S) = \cdots \xrightarrow{\partial_{j+1}^S} C_j(S) \xrightarrow{\partial_j^S} C_j(S) \xrightarrow{\partial_{j-1}^S} \cdots \xrightarrow{\partial_2^S} C_1(S) \xrightarrow{\partial_1^S} C_0(S),$$

denote the usual simplicial chain complexes with coefficients in the field K. Thus $C(S)^j$ is the vector space with basis the j-simplices.

With respect to any choice of bases for the vector spaces $C_j(S)$, we can represent $C(S)$ as a sequence of matrices $[\partial_j^S], [\partial_{j-1]}^S], \ldots, [\partial_{j-1]}^S]$.

Example 7.3. Let S be the 2-simplex [1 2 3] regarded as a simplicial complex. Let us work over the field $\mathbb{Z}/2\mathbb{Z}$. Then

$$C(S) = \cdots \to 0 \to 0 \to C(S)^2 \xrightarrow{\partial_2^S} C(S)^1 \xrightarrow{\partial_1^S} C(S)^0,$$

where

- $C_0(S) \cong (\mathbb{Z}/2\mathbb{Z})^3$ is the vector space with basis $\{[1], [2], [3]\}$,
- $C_1(S) \cong (\mathbb{Z}/2\mathbb{Z})^3$ is the vector space with basis $\{[1\ 2], [2\ 3], [1\ 3]\}$,

- $C_2(S) \cong (\mathbb{Z}/2\mathbb{Z})$ is the vector space with basis $\{[1\ 2\ 3]\}$.

With respect to these ordered bases,

$$[\partial_1^S] = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}, [\partial_2^S] = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

7.2 Chain Complexes of Filtrations

Recall that in the usual persistent homology / TDA pipeline, we map a data object P to a filtration F. Let us assume for now that F is $\mathbb{N}$-indexed. In the computational setting, F is assumed to be *finite simplicial* filtration, meaning that:

- F takes values in the category of finite simplicial complexes.
- The filtration stabilizes, i.e., there is some y such that $F_y = F_z$ for all $z \geq y$.

Let us write $S := F_y$. In computations, we store F in memory by storing the single simplicial complex S (there are different ways to do this, but the simplest way is to store each simplex as a list of vertices), together with the *birth index* $b(\sigma)$ of each simplex σ in S, i.e., the $\mathbb{N}$-index where σ first appears in F.

As explained earlier, post-composing F with the homology functor H_i yields a persistence module $H_i(F)$. To compute the barcodes $\mathcal{B}_i(F) := H_i(F)$ from F, one first constructs the chain complex of persistence modules

$$C(F) = \cdots \xrightarrow{\partial^{j+1}} C^j(F) \xrightarrow{\partial^j} C^{j-1}(F) \xrightarrow{\partial^{j-1}} \cdots \xrightarrow{\partial^2} C^1(F) \xrightarrow{\partial^1} C^0(F),$$

where

- $C^j(F)_z$ is the usual simplicial chain vector space $C_j(F_z)j$,
- each internal map $C^j(F)_y \to C^j(F)_z$ of $C^j(F)$ is the inclusion,
- for each z, ∂_z^j is the usual simplicial boundary map $\partial_j^{F_z}$. (Note that the slight abuse of notation here.)

Note that I am flouting convention and putting the simplex dimension as a superscript, not a subscript here. This is because I want to save subscripts for vector space indices. (But for simplicial complexes, I put the dimension as a subscript.)

Exercise 7.4. *Check that*

$$\ker(\partial^j)/\operatorname{im}(\partial^{j+1}) = H_i(F)$$

(where $H_i(F)$ is defined by post-composition with the homology functor.)

It's not hard to show that each $C^j(F)$ is a free 1-parameter persistence module with basis corresponding to the j-simplices of F_y, i.e.,

$$C^j(F) \cong \oplus_{\sigma \text{ a } j\text{-simplex of } S} Q^{b(\sigma)}.$$

Let us fix an order on each set of j-simplices. Then with respect to the resulting ordered bases, the matrix representation of

$$\partial^j : C^j(F) \to C^{j-1}(F)$$

is exactly the matrix representation of

$$\partial_j^S : C_j(S) \to C^{j-1}(S),$$

except that the former has labels for the rows and columns.

It is sometimes convenient to regard the collection of maps $\{\partial^j\}_{j\in\mathbb{N}}$ as a single map

$$\partial : \bigoplus_{j=0}^{\dim S} C^j(F) \to \bigoplus_{j=0}^{\dim S} C^j(F).$$

With respect to the basis for $\bigoplus_{j=0}^{\dim S} C^j(F)$ consisting of all simplices, with simplices in order of increasing dimension, $[\partial]$ is a block matrix whose only only non-zero blocks lie on the superdiagonal, and the non-zero blocks are the matrices $[\partial^j]$.

Example 7.5. To specify a filtration F, we assume that the 2-simplex S of Example 7.3 is filtered so that the birth index of each simplex in S is given by the following table:

simplex	[1]	[2]	[3]	[1 2]	[2 3]	[1 3]	[1 2 3]
birth index	1	2	3	4	5	6	7

With respect to the simplex orderings of the earlier example, the non-zero boundary morphisms

$$\partial^j : C^j(F) \to C^{j-1}(F)$$

are represented by the following labeled matrices:

$$[\partial^1] = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 4 & 5 & 6 \end{array} \\ \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \end{array} \qquad [\partial^2] = \begin{array}{c} \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{c} 7 \end{array} \\ \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \end{array}$$

The labeled matrix $[\partial]$ is thus given by

$$[\partial] = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

7.3 Computing a Barcode Directly from a Chain Complex

We now explain how to compute the barcodes $\mathcal{B}_i(F) = \mathcal{B}_{H_i(F)}$ for each i, directly from the chain complex $C(F)$. We follow the 2005 paper of Carlsson and Zomorodian "computing persistent homology."

Let F be a finite simplicial filtration. Assume that the simplices of F are given a total order such that $\sigma < \tau$ whenever $\dim(\sigma) < \dim(\tau)$ or $(\dim(\sigma) = \dim(\tau)$ and $b(\sigma) < b(\tau))$. We let σ_j refer to the j^{th} simplex in this order. To simplify notation, let us assume that $b(\sigma_j) = j$. Let $[\partial]$ denote the matrix representation of ∂ with respect to this order.

The main result is the following:

Theorem 7.6 (Carlsson, Zomorodian '05)**.** *Let R be the matrix obtained by applying the standard reduction to $[\partial]$. Then*

$$\begin{aligned} \mathcal{B}_i(F) = &\{[j,k) \mid \rho_k^R = j,\ \dim(\sigma_j) = i\} \\ &\cup \{[j,\infty) \mid R_{*,j} = 0,\ \dim(\sigma_j) = i,\ \nexists\ k \text{ such that } \rho_k^R = j\}. \end{aligned}$$

Remark 7.7. A version of this result holds without the assumption that $b(\sigma_j) = j$. The proof is the same, but the statement requires a bit more notation.

This tells us that to compute the persistence barcodes of a filtration, it is enough to reduce each of the boundary matrices.

Example 7.8. Consider the filtration F of Example 7.5. Reducing the boundary matrix $[\partial]$ yields the following matrix:

$$R = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Hence according to Theorem 7.6,

$$\mathcal{B}_0(F) = \{[1,\infty),\ [2,4), [3,5)\}$$
$$\mathcal{B}_1(F) = \{[6,7)\}.$$

The proof of Theorem 7.6 relies on the following fact:

Lemma 7.9. *Let $\gamma : F \to F'$ be a morphism of finitely generated free 1-parameter persistence modules. Then*

(i) $\ker \gamma$ *is a free direct summand of* F*,*

(ii) If B *and* B' *are bases for* F *and* F' *with* $[\gamma]^{B',B}$ *reduced, then*

$$\{B_j \in B \mid [\gamma]^{B,B'}_{*,j} = 0\}$$

is a basis for $\ker \gamma$*.*

Sketch of proof. This follows from the fact that that the non-zero columns of a reduced matrix are linearly independent. We leave the details as an exercise. □

Exercise 7.10. *Fill in the details of the proof of Lemma 7.9.*

Proof of Theorem 7.6. R is computed from $[\partial]$ by a sequence of rightward column additions; each such addition corresponds to a change of basis operation on $C^i(F)$ for some i. Each such change of basis operation also corresponds to an upward row addition on $[\partial]$. If for each column addition we do when computing R, we also do the corresponding upwards row operation, we obtain a matrix Q representing the map ∂ with respect to some choice of bases for the modules $C^i(F)$. By Proposition 7.2, an upward row addition on a reduced matrix again yields a reduced matrix with the same pivots. We can perform all of the row operations after all of the column operations, so it is clear that Q is reduced, and that Q and R have the same pivots.

Consider the submatrix P of Q consisting only of the non-zero columns, and the rows j such that column j is zero. Then by Lemma 7.9, P is a presentation matrix for $\oplus_i H_i(F)$. Moreover, this presentation matrix has block structure inherited from $[\partial]$, and the blocks give presentations for each of the individual modules $H_i(F)$.

The theorem now follows from Eq. (1). □

Exercise 7.11. *For R as in the example of Example 7.8, compute Q and P from R, as in the proof of Theorem 7.6. Explain how we can read the barcodes of F off of P.*

7.4 Practical Computation of Persistent Homology

We now turn briefly, to the problem of fast, scalable, computation of persistent homology. This has been a very active area of research over the last 15-20 years. In this time, several tricks have been discovered that greatly impact the performance of persistent homology computation, leading to several orders of magnitude speedup for some kinds of data. Some the most important progress has happened in just the past few years.

We will not aim to give a thorough account of such progress, but rather just touch on a few key ideas.

Pivot Arrays To complete the specification of the standard reduction (Algorithm 1) it remains to explain how we check the whether the conditional of the while loop is, and how we find k when the conditional does hold. This can be done in constant time, provided we maintain an 1-D array pivs of length m, where $\mathsf{pivs}[i]$ records which column reduced so far, if any, has i as its pivot. We call pivs the *pivot array*. The full algorithm using the pivot array is given below as Algorithm 3.

Algorithm 3 The standard reduction (In Detail)

Input: An $m \times n$ matrix D
Output: A reduced $m \times n$ matrix R obtained from D by rightward column additions

$R \leftarrow D$
Initialize an array pivs of size n, with each entry set to null
for $j = 1$ to n **do**
    while $R(*, j) \neq 0$ **and** $\mathsf{pivs}[\rho_j^R] \neq \mathsf{null}$ **do**
        $k \leftarrow \mathsf{pivs}[\rho_j^R]$
        add $-\frac{R(\rho_j^R, j)}{R(\rho_j^R, k)} R(*, k)$ to $R(*, j)$.
    if $R(*, j) \neq 0$ **then**
        $\mathsf{pivs}[\rho_j^R] \leftarrow j$

Column-Sparse Representation of Matrices In persistent homology computations, the matrices involved are huge, but very sparse. It is therefore necessary to store the matrices in a sparse format.

The standard reduction requires us to work with a sparse format allowing for

- fast access to the non-zero element of largest index in each column,
- fast column addition.

The obvious thing to do, then, is to store each column as some sort of list of pairs (i, c), where i is the index of a non-zero entry in the matrix, and c is the entry. For $\mathbb{Z}/2\mathbb{Z}$ coefficents, c is always 1, so one of course does not need to store c.

Early implementations of the persistent homology implemented this idea using linked lists. Subsequently, the work of Bauer, Kerber, Reininghaus, and Wagner has studied the practical efficiency of a number of sparse data structures for matrix columns, including linked lists, dynamically allocated arrays, *lazy heaps*, and (for $\mathbb{Z}/2\mathbb{Z}$ coefficients) *bit trees*. Both dynamically allocated arrays and lazy heaps seem to be good choices of data structure in practice.

Clearing Since the matrix $[\partial]$ considered above is a block matrix, reducing this matrix is equivalent to reducing the matrix $[\partial^i]$ for each i. In their paper "persistent homology computation with a twist," Chen and Kerber have observed that the reduction of $[\partial^i]$ can be used to expedite the reduction of $[\partial^{i-1}]$.

To explain this suppose that R^i and R^{i-1} are reduced matrices obtained from $[\partial^i]$ and $[\partial^{i-1}]$, respectively, by left-to-right column additions (e.g., via the standard reduction.)

Lemma 7.12 (Clearing lemma)**.** *If the j^{th} column of R^i is non-zero and has pivot k, then the k^{th} column of R^{i-1} is 0.*

Proof. $[\partial^{i-1}]R^i = 0$, so $[\partial^{i-1}]R^i_{*,j} = 0$. This implies that the k^{th} column of $[\partial^{i-1}]$ is a linear combination of columns of smaller index. Since R^{i-1} is obtained from $[\partial^{i-1}]$ by rightward column additions, the k^{th} column of $[\partial^{i-1}]$ is in fact a linear combination of columns of R^{i-1} of smaller index. Since non-zero columns of a reduced matrix are linearly independent, the k^{th} column of R^{i-1} must be 0. □

Hence, for each non-zero column of R^i, we can immediately zero out one column of R^{i-1} before running the standard reduction to compute R^{i-1}. This shortcut is called the *twist optimization*, or alternatively, *clearing*. It has been observed that for typical persistent homology computations, this optimization can yield drastic speedups. This is because empirically, most of the work in a naive application of the persistence algorithm actually goes to zeroing out the columns that will in any case be zeroed out by clearing. Virtually all state-of-the-art persistent homology codes make use of the clearing optimization.

Persistence Computation via Row Operations (The Dual Approach) The usual persistence algorithm, as we have described it, applies the standard reduction to the columns of a matrix. However, one can instead apply the standard reduction to the rows of the matrix, using the reverse order on the rows and hence only performing upward column additions. Let us call this the *upward standard reduction*. This variant of the persistence algorithm is usually described in the literature in terms of cohomology; though one needn't consider cohomology to understand this variant, this perspective can be useful. This idea first appears in work of de Silva, Morozov, and Vejdemo-Johansson.

The key result is this:

Proposition 7.13. *Let R^i be obtained from $[\partial^i]$ by the upward standard reduction. Let ρ^R_j denote the pivot of the j^{th} row pivot of R, i.e., the index of the smallest non-zero entry of the*

j^{th} *row, if the row is non-zero, and* null *if the row is 0. Then we have the following variant of the formula from Theorem 7.6.*

$$\begin{aligned}\mathcal{B}_i(F) =& \{[j,k) \mid \rho_j^R = k,\ \dim(\sigma_j) = i\} \\ & \cup \{[j,\infty) \mid R_{j,*} = 0,\ \dim(\sigma_j) = i,\ \nexists\ k \text{ such that } \rho_j^R = k\}.\end{aligned}$$

This is usually proven by a dualization argument, appealing to an equivalence between persistent cohomology and persistent homology. The key ideas are described in the work of de Silva et al. mentioned above. Correctness probably can also be proven directly, using a proof dual to our proof of the correctness of the column-wise algorithm.

Clearing also works for the row version of the persistence algorithm, provided we store our matrices in a row-sparse format. Here, clearing allows us to zero out rows of $[\partial^{i+1}]$ as we reduce $[\partial^i]$. For some kinds of filtrations, such as Vietoris-Rips filtrations, the authors of PHAT have observed that using clearing together with the row-wise variant of the persistence algorithm is MUCH faster than using clearing in the column-wise variant. This is because we typically only compute persistent homology of Vietoris-Rips filtrations up to some small homology dimension, say up to dimension 2. For such computations, the cost of reducing the matrices $[\partial^i]$ is dominated by the reduction in top dimension, where the matrix is largest. Since clearing with column operations cannot expedite the reduction in top dimension, clearing does relatively little to speed up the column-wise variant of the computation. However, clearing does help with the reduction in top dimension if we use the row-wise approach to persistence computation, and this typically makes a big difference.

8 Stability of Persistent Homology

The first version of the stability theorem for persistent homology was proven by Cohen-Steiner, Edelsbrunner, Harer in 2005. Since then, the stability of persistent homology has been revisited many times in the literature, leading to numerous generalizations and variants of the theorem. Arguably, the single most important advance after the original work of Cohen-Steiner, Edelsbrunner, Harer was the discovery by Chazal and his coauthors in 2009 that the stability of persistent homology admits a very useful purely algebraic generalization.

Here, we will give an overview of some aspects of the stability theory for persistent homology. We will focus on four versions of the stability theorem, each one of which is more general than the last:

- Hausdorff Stability, for point cloud data [Cohen-Steiner et al. '05]
- Stability of sublevel persistent homology [Cohen-Steiner et al. '05]
- Forward/Converse Algebraic stability [Chazal et al. '09] [L. '11]
- Single-morphism algebraic stability (i.e., the induced matching theorem) [Bauer, L. '13, '16]

As we will observe, each of the first three versions of stability follows almost immediately from the one below it on the list.

8.1 Bottleneck Distance

To explain the stability of persistent homology, we will need to define several distances on objects arising in the persistent homology. We begin with the *bottleneck distance* a distance on barcodes. For simplicity, I will give the definition under the assumption that each interval in the barcodes we consider is of the form $[b, d)$, for $b < d \in \mathbb{R}$. However, the definition of the distance extends to arbitrary barcodes without difficulty.

Definition 8.1 (Matching)**.** Given two sets S and T, we say a *matching* $\sigma : S \to T$ is a simply bijection from $S' \to T'$ for some $S' \subset S$ and $T' \subset T$. The definition extends without difficulty to multi-sets; we omit the straightforward details of this. In particular, matchings of barcodes are well-defined.

Definition 8.2 (Bottleneck Distance)**.** For barcodes $\mathcal{C}$, $\mathcal{D}$ and $\delta \geq 0$, we say a matching $\sigma : C \to D$ is a δ-matching if

(i) σ matches each interval in $\mathcal{C} \cup \mathcal{D}$ of length greater than 2δ

(ii) if $\sigma([b, d)) = [b', d'))$, then $|b - b'| < \delta$ and $|d - d'| < \delta$.

We define the *bottleneck distance* d_B by taking

$$d_B(\mathcal{C}, \mathcal{D}) := \inf \{\delta \geq 0 \mid \exists \text{ a } \delta\text{-matching between } \mathcal{C} \text{ and } \mathcal{D}\}.$$

The bottleneck distance is an *extended pseudometric*: This means it is symmetric (i.e., $d_B(\mathcal{C}, \mathcal{D}) = d_B(\mathcal{D}, \mathcal{C})$) and it satisfies the triangle inequality. The bottleneck distance between two barcode can be ∞. (This is the meaning of *extended*.) The bottleneck distance between two different barcodes can be 0. (This is the meaning of *pseudo-*.)

8.2 Hausdorff Stability of Persistent Homology

We will use notation from Section 4.1 of these notes. In particular, recall that for $P \subset \mathbb{R}^n$, $\mathcal{S}^{\uparrow}(d_P)$ is the union-of-balls filtration of P (i.e., the subslevelset filtration of the distance function to P).

Definition 8.3. The *Hausdorff* distance $d_H(P, Q)$ between two subsets P, Q of $\mathbb{R}^n$ is given by

$$d_H(P, Q) = \inf\{\delta > 0 \mid P \subset \mathcal{S}^{\uparrow}(d_Q) \text{ and } Q \subset \mathcal{S}^{\uparrow}(d_P)\}.$$

For $P \subset \mathbb{R}^n$ finite, let us write the sublevel persistence barcode $\mathcal{B}_{H_i(\mathcal{S}^{\uparrow}(d_P)}$ simply as $\mathcal{B}_i(P)$.

Theorem 8.4 (Hausdorff Stability)**.** *For any finite $P, Q \subset \mathbb{R}^n$ and $i \geq 0$, $d_B(\mathcal{B}_i(P), \mathcal{B}_i(P)) \leq d_H(P, Q)$.*

This theorem tells us that in one very reasonable sense, persistent homology is stable with respect to small perturbations of the data.

8.3 Stability of Sublevel Persistence

Definition 8.5. For T be a topological space and $\gamma, \kappa : T \to \mathbb{R}$ any functions, we define the *sup-norm distance* between γ and κ by

$$d_\infty(\gamma, \kappa) := \sup_{x \in T} |\gamma(x) - \kappa(x)|.$$

Theorem 8.6 (Sublevel Stability)**.** *For any topological space T, functions $\gamma, \kappa : T \to \mathbb{R}$ and $i \geq 0$ with $H_i(\mathcal{S}^\uparrow(\gamma))$ and $H_i(\mathcal{S}^\uparrow(\kappa))$ both p.f.d., we have that*

$$d_B(\mathcal{B}_i(\gamma), \mathcal{B}_i(\kappa)) \leq d_\infty(\gamma, \kappa).$$

Exercise 8.7. *Show that for any $P, Q \subset \mathbb{R}^n$, $d_\infty(d_P, d_Q) \geq d_H(P, Q)$. Thus, the sublevel stability of persistent homology implies the Hausdorff stability.*

Remark 8.8. In '09, Chazal, Oudot, Mémoli, Guibas, and Cohen-Steiner showed that sublevel stability can also be applied to get a stability result result for the barcodes of Vietoris-Rips filtrations.

8.4 Algebraic Stability

Perhaps surprisingly, the stability results above admit a simple, purely algebraic generalization. To articulate this, we need to introduce *interleavings* of persistence modules: First, for any $\mathbb{R}$-indexed persistence module M and $\delta \in \mathbb{R}$, we define the *shift of M by δ* to be the persistence module module $M(\delta)$ given by $M(\delta)_r = M_{r+\delta}$, $M(\delta)_{r,s} = M_{r+\delta, s+\delta}$. This definition is functorial, i.e., for any morphism $f : M \to N$ of persistence modules, we have an induced morphism $f(\delta) : M(\delta) \to N(\delta)$.

Note that the internal maps $\{M_{r,r+\delta}\}_{r \in \mathbb{R}}$ assemble into a morphism of persistence modules $\phi^{M,\delta} : M \to M(\delta)$.

We define a δ-interleaving between $\mathbb{R}$-indexed persistence modules M and N to be a pair of morphisms

$$f : M \to N(\delta) \qquad g : N \to M(\delta)$$

such that

$$g(\delta) \circ f = \phi^{M,2\delta} \qquad f(\delta) \circ g = \phi^{N,2\delta}.$$

Said differently, an interleaving is a collection of linear maps

$$\{f_r : M_r \to N_{r+\delta}\}_{r \in \mathbb{R}} \quad \{g_r : N_r \to M_{r+\delta}\}_{r \in \mathbb{R}}$$

such that "everything in sight commutes." That is, the infinite diagram of vector spaces built from M, N, and these linear maps is commutative.

We define the interleaving distance d_I on persistence modules by

$$d_I(M, N) = \inf \{\delta \mid \text{There exists a } \delta\text{-interleaving between } M \text{ and } N\}.$$

d_I is an extended pseudometric on persistence modules.

Remark 8.9. Here, we have considered interleavings between persistence modules, i.e., functors $\mathbb{R} \to \mathbf{Vect}$. However, it is easy to see that interleavings can be defined between two functors $\mathbb{R} \to \mathcal{C}$, for any fixed category $\mathcal{C}$. Often in TDA, we also consider $\mathcal{C} = \mathbf{Top}$ or $\mathcal{C} = \mathbf{Set}$.

Remark 8.10. We can define δ-interleavings and the interleaving distance on $\mathbb{Z}$-indexed modules in the same way.

Recall from Section 3.3 that a persistence module M is said to be p.f.d. if $\dim M_r < \infty$ for all r. By Theorem 3.2, a p.f.d. has a well defined barcode. Here is the algebraic stability result of Chazal et al., origianlly presented in 2009.

Theorem 8.11 (Forward Algebraic Stability)**.** *For any persistence modules M, N, we have*

$$d_B(\mathcal{B}(M), \mathcal{B}(N)) \leq d_I(M, N)).$$

In fact the converse of Theorem 8.11 also holds, and was first proven in my thesis work in 2011 (and published in 2015). The result also was also discovered independently and published later by others.

Theorem 8.12 (Converse Algebraic Stability)**.** *For any persistence modules M, N, we also have*

$$d_B(\mathcal{B}(M), \mathcal{B}(N)) \geq d_I(M, N)).$$

This, together with Theorem 8.11, gives that:

$$d_B(\mathcal{B}(M), \mathcal{B}(N)) = d_I(M, N)). \tag{3}$$

The equation (3) is known as the *isometry theorem*.

Forward algebraic stability is the more difficult direction to prove. The proof of converse algebraic stability is straightforward:

Exercise 8.13 (Proof of converse algebraic stabillity)**.**

(i) Show that converse algebraic stability holds in the special case that M and N are both interval modules or trivial modules.

(ii) Check that if M and N are δ-interleaved, and M' and N' are also δ-interleaved, then $M \oplus M'$ and $N \oplus N'$ are δ-interleaved.

(iii) Show that converse algebraic stability follows from these two facts.

Remark 8.14 (Algebraic Stability Implies Sublevel Stability)**.** It is easy to check that for $\gamma, \kappa : T \to \mathbb{R}$ with $d_\infty(\gamma, \kappa) = \delta$, $\mathcal{S}^\uparrow(\gamma)$ and $\mathcal{S}^\uparrow(\kappa)$ are δ-interleaved. Suppose the interleaving morphisms are f and g. Then applying i^{th} homology to each linear map f_r and g_r yields a δ-interleaving $H_i(f)$, $H_i(g)$ between $H_i \circ \mathcal{S}^\uparrow(\gamma)$ and $H_i \circ \mathcal{S}^\uparrow(\kappa)$. Applying algebraic stability to this interleaving yields the sublevel stability result Theorem 8.6.

Forward algebraic stability is extremely useful in TDA; it and its corollaries are used frequently in the statistical foundations of TDA and also in the computational theory. The full isometry theorem is of interest largely because it suggests an avenue for extending fundamental TDA results from the ordinary 1-parameter persistence setting to the multi-parameter setting, and to other generalized persistence settings. Many key results in TDA are stated using the bottleneck distance, but the bottleneck distance does not admit a naive generalization to settings where we have no barcode. (In particular, it does not generalize to multi-parameter persistence modules.) The interleaving distance, on the other hand, turns out to generalize readily to the multi-parameter setting and to other generalized persistence settings. Using the interleaving distance, we can develop TDA theory in such generalized settings.

8.5 Single-Morphism Algebraic Stability: The Induced Matching Theorem

In 2013, Ulrich Bauer and I wrote a paper showing that algebraic stability, ostensibly a result about pairs of morphisms of persistence modules, in fact follows almost immediately from a general result about single morphisms of persistence modules. This leads to a proof of algebraic stability which is more direct and arguably simpler than the original approach Chazal et al. (A third proof of algebraic stability, due to Bjerkevik, was later found. This is also a nice proof with the advantage that it generalizes readily in certain ways that the the single-morphism proof does not.)

Here, we outline the single-morphism approach to algebraic stability. The approach is based on a construction called the *induced matching*, which associates to any p.f.d. persistence modules $f : M \to N$ a simple, explicit matching of barcodes $\chi_f : \mathcal{B}_M \to \mathcal{B}_N$. To explain how χ_f is defined, let us assume for simplicity that each interval in each barcode is of the form $[b, d)$. (This assumption is usually satisfied in practice, and in any case is not actually needed, but it does allow us to simplify notation.)

It is clear from the way we defined matchings (Definition 8.1) that we can define composition of matchings. Explicitly, given matchings $\sigma : S \to T$ and $\tau : T \to U$, the *composite* matching $\tau \circ \sigma : S \to U$ matches s to u if and only if there exists $t \in T$ such that $\sigma(s) = t$ and $\tau(t) = u$.

We define $\chi_f : \mathcal{B}_M \to \mathcal{B}_N$ as the composite χ_1 of two matchings $\chi_1 : \mathcal{B}_M \to \mathcal{B}_{\operatorname{im} f}$ and $\chi_2 : \mathcal{B}_{\operatorname{im} f} \to \mathcal{B}_{\operatorname{im} f}$. (Recall from Exercise 5.12 and Definition 5.8 that $\operatorname{im} f$ is a well-defined persistence module. If M is p.f.d., then so is $\operatorname{im} f$; hence $\operatorname{im} f$ has a barcode $\mathcal{B}_{\operatorname{im} f}$.)

To define χ_1 we separately match intervals in $\mathcal{B}_M$ and $\mathcal{B}_{\operatorname{im} f}$ of the form $[b,)$ for fixed $b \in \mathbb{R}$. (In terms of the persistence diagram visualization of barcodes, this corresponds to matching points in the along vertical lines.). We match the longest such interval in $\mathcal{B}_M$ to the longest such interval in $\mathcal{B}_{\operatorname{im} f}$, the second-longest such interval in B_M to the second-longest such interval in $\mathcal{B}_{\operatorname{im} f}$, and so on, until we run out of such intervals in one of the barcodes. We then do this for each $b \in \mathbb{R}$. This yields the matching χ_1.

Similarly, to define χ_2 we separately match intervals in $\mathcal{B}_{\operatorname{im} f}$ and $\mathcal{B}_N$ of the form $[,d)$ for

fixed $d \in \mathbb{R}$. (In terms of the persistence diagram visualization of barcodes, this corresponds to matching points along horizontal lines.) We match the longest such interval in $\mathcal{B}_{\text{im}\, f}$ to the longest such interval in $\mathcal{B}_N$, the second-longest such interval in B_M to the second-longest such interval in $\mathcal{B}_{\text{im}\, f}$, and so on, until we run out of such intervals in one of the barcodes. We then do this for each $b \in \mathbb{R}$. This yields the matching χ_2.

(In the presence of multiple copies of the same interval, one needs to be slightly more careful about the definition of these matchings to ensure that the composite is uniquely defined. In short, for each barcode, one fixes an order on the copies of each interval in the barcode, and matches the copies in that order.)

The following is a single-morphism generalization of the algebraic stability theorem.

Theorem 8.15 (Induced Matchings)**.**

(i) If $\chi_f[b, d) = [b', d')$*, then*

$$b' \leq b < d' \leq d. \tag{4}$$

(ii) If each interval in $\mathcal{B}_{\ker f}$ *has length at most* δ*, then*

$$|d - d'| \leq \delta$$

and χ_f *matches each interval in* $\mathcal{B}_M$ *of length greater than* δ*.*

(iii) Dually, if each interval in $\mathcal{B}_{\text{coker}\, f}$ *has length at most* δ*, then*

$$|b - b'| \leq \delta$$

and χ_f *matches each interval in* $\mathcal{B}_N$ *of length greater than* δ*.*

Partial Functoriality of Induced Matchings One can define a category **Barc** whose objects are barcodes and whose morphisms are matchings, such that each matched pair of intervals $([b, d), [b', d'))$ satisfies (4). The matchings χ_f are not functorial (i.e., they do not define a functor $\mathbf{Vect}^{\mathbb{R}} \to \mathbf{Barc}$.) However, they are functorial on the subcategory of $\mathbf{Vect}^{\mathbb{R}}$ consisisting of only monomorphisms (i.e., morphisms of persistence modules that are injective at each index.) The same is true for epimorphisms.

Outline of Proof of the Induced Matching Theorem We briefly outline the proof; see the paper for the full details. We first establish the following structure theorem for submodules of persistence modules. For simplicity we state the result under the assumption that all intervals in the barcodes are of the form $[b, d)$.

Proposition 8.16 (Structure theorem for submodules)**.** *Suppose we have a monomorphism* $j : M \to N$*. Then* $\chi_j : \mathcal{B}_M \to \mathcal{B}_N$ *matches each interval in* $\mathcal{B}_M$*, and for each* $[b, d) \in \mathcal{B}_M$*, we have* $\chi_j[b, d) = [b', d)$ *for some* $b' \leq b$*.*

A dual statement also holds for quotient modules. Together, these yield Theorem 8.15 (i).

We next establish Theorem 8.15 in the case that f is a monomorphism. In this case, $\ker f$ is trivial, so Theorem 8.15 (ii) follows immediately from Proposition 8.16.

For f is a monomorphism, the proof of Theorem 8.15 (iii) proceeds by a sandwiching argument: Let the submodule $N^\delta \subset N$ be defined by

$$N_r^\delta = \{n \in N_r \mid n = N_{r-\delta,r}(n') \text{ for some } n' \in N_{r-\delta}.\}$$

We observe that if each interval in $\mathcal{B}_{\operatorname{coker} f}$ has length at most δ, then $N^\delta \subset \operatorname{im} f$. Noting that $M \cong \operatorname{im} f$, we thus have a factorization by monomorphisms

$$N^\delta \hookrightarrow M \overset{f}{\hookrightarrow} N$$

of the inclusion $j : N^\delta \hookrightarrow N$. We may think of this as a "sandwiching" of M in terms of N. Clearly, $\mathcal{B}_{N^\delta}$ is obtained from $\mathcal{B}_N$ by shortening each interval of $\mathcal{B}_N$ on the left side by δ, and the induced matching $\chi_j : \mathcal{B}_{N^\delta} \hookrightarrow \mathcal{B}_N$ is thus especially simple. By the functoriality of matchings induced by monomorphisms, the sequence of induced matchings

$$\mathcal{B}_{N^\delta} \hookrightarrow \mathcal{B}_M \overset{\chi_f}{\hookrightarrow} \mathcal{B}_N$$

factors χ_j. Applying Proposition 8.16 to this factorization yields Theorem 8.15 (iii).

The dual argument gives Theorem 8.15 (ii) for f an epimorphism. Finally, Theorem 8.15 (ii) and (iii) follow for arbitrary f almost immediately from theses special cases by taking the epi-mono decomposition of f.

Remark 8.17. The categorical structure on **Barc** can be used to give a slick formulation of the induced matching theorem, which transparently expresses the sense in which passage from a persistence module to its barcode preserves categorical structure. This is explained in a followup paper by Bauer and me.

Algebraic Stability from Induced Matchings It remains to explain how the the algebraic stability theorem follows from the induced matching theorem. The key observation is the following, which is an easy exercise:

Exercise 8.18. *Let $f : M \to N(\delta)$ be a δ-interleaving morphism. Then each interval in each of the barcodes $\mathcal{B}_{\ker f}$ and $\mathcal{B}\operatorname{coker} f$ has length at most 2δ.*

Note that for any $\delta \geq 0$, and p.f.d. persistence module N, we have an obvious bijective matching $r^\delta : \mathcal{B}_{N(\delta)} \to \mathcal{B}_N$.

To prove the algebraic stability theorem from the induced matching theorem, we apply the latter to the δ-interleaving morphism $f : M \to N(\delta)$, using Exercise 8.18; this gives us that $r^\delta \circ \chi_f : \mathcal{B}_M \to \mathcal{B}_N$ is a δ-matching. Algebraic stability now follows.

9 Multi-Parameter Filtrations from Data

We now (finally) turn in earnest to multi-parameter persistent homology. So far, we have studied multi-parameter persistence modules a bit, as part of our preparation for the proof of the structure theorem for 1-parameter persistence modues, and for our explanation of the persistence algorithm. (Of course we did not need multi-parameter persistence modules there, but it was natural to just develop some of the machinery in the multi-parameter setting.) But though we have breifly outlined the multi-parameter persistence pipeline in Section 5.1, we have not yet considered how multiparameter filtrations arise from data.

9.1 Motivation

First, we consider four motivations for working in the multi-parameter setting. The first three of these were proposed by Carlsson and Zomorodian in their original paper on multi-parameter persistence.

Motivation 1: Handling Noisy Point Cloud Data We have seen that 1-parameter (e.g. Čech) persistent homology of a point cloud is stable with respect to the Haudorff distance. However, the Hausdorff distance is highly unstable with respect to addition/removal of outliers in the data, and correspondingly, the persistent homology of a point cloud is unstable with respect to such changes. Relatedly, persistent homology, in its ordinary 1-parameter formulation, can be insensitive to interesting structure in high density regions of the data. This is a serious defect: We want to be able to apply persistent homology to real-world scientific data, but such data most often has such outliers/noise. Different solutions have been proposed to address this problem, but they all lead us to 2-parameter persistent homology. The rough idea is that a second persistence parameter will be introduced to control how aggressively we remove outliers from the data.

Motivation 2: Tendrils in Data Carlsson and Zomorodian noted that if a point cloud data set has several "tendrils" or "spikes" emanating from a central core, then one can study the data topologically by removing the "central core" of the data—what then remains is one cluster for each tendril in the data, and one can study this structure using 0^{th} persistence. However, there is typically no clear binary distinction between data points which lie in the central core and which do not. Thus, it is natural to introduce a second persistence parameter to control how aggressively we remove points from the central code.

Motivation 3: Data Endowed with a Function Some point cloud data sets $P \subset \mathbb{R}^n$ comes endowed with a function $\gamma : P \to \mathbb{R}$, or several such functions. For example, in a drug discovery application, P may be the atom-centers of a biomolecule, and γ may be the partial charge function. Time-varying point cloud data (e.g., swarms of birds or insects) can also be formally described this way. To build persistence invariants that are sensistive both to the

structure of this function and to the geometric structure of this data, and to the way that these interact, it is natural to introduce a second persistence parameter which filters the data by the function.

Motivation 4: Sensitivity to width of features in functional data Ordinary sublevel persistence is sensitive to the height of features in the graph of a function but not their width. Indeed, sublevel persistence does not depend at all on a metric on the domain of a function. This means that narrow topological features, like spikes, contribute to barcodes in exactly the same way as features of large width. If we want to build persistence invariants of data which are sensitive to both height and width, it is natural to work in the 2-parameter setting.

9.2 Constructions of Bifiltrations from Data

We now consider several constructions of bifiltrations from data. We err on the side of completeness here, aiming to give a sense of the breadth of interesting constructions of bifiltrations that are available.

Bifiltrations and Bipersistence Modules Earlier, we defined a d-parameter filtration to be a functor $F : \mathbb{N}^d \to \mathbf{Top}$ with $F_s \subset F_t$ whenever $s \leq t$. Here we will focus on the case $d = 2$, in which case we call this a *bifiltration*. We also will regard functors $f : \mathcal{J} \to \mathbf{Top}$ where $\mathcal{J} = \mathbb{R}^2$, $\mathcal{J} = \mathbb{R}^{\mathrm{op}} \times \mathbb{R}$, or $\mathcal{J} = \mathbb{N}^{\mathrm{op}} \times \mathbb{R}$ as bifiltrations, assuming as above that $F_s \subset F_t$ whenever $s \leq t$. We define a *bipersistence module* to be a functor $\mathcal{J} \to \mathbf{Vect}$.

Function-Rips bifiltrations Let P be a finite metric space, and $\gamma : P \to \mathbb{R}$ be any function. We define the *sublevel-Rips* bifiltration $\mathcal{S}^{\uparrow}(\gamma) : \mathbb{R}^2 \to \mathbf{Simp}$ by

$$\mathcal{S}^{\uparrow}(\gamma)_{a,r} = \gamma^{-1}(-\infty, a].$$

Here **Simp** denotes the category of simplicial complexes.

Similarly, we define the the *superlevel-Rips* bifiltration $\mathcal{S}^{\downarrow}(\gamma) : \mathbb{R}^{\mathrm{op}} \times \mathbb{R} \to \mathbf{Simp}$ by

$$\mathcal{S}^{\downarrow}(\gamma)_{a,r} = \gamma^{-1}[a, \infty).$$

Example 9.1 (Density-Rips bifiltration)**.** Fixing a parameter $r > 0$, define $\gamma_r : P \to \mathbb{R}$ by

$$\gamma_r(x) = |\{y \in P \mid y \neq x,\ d(x, y) \leq r\}|.$$

That is, $\gamma_r(x)$ is the number of other points in P within distance r of x. γ_r is an example of a *density* estimate a function whose value is high in dense regions of the data and low near sparse regions of the data.

We call $\mathcal{S}^{\downarrow}(\gamma)$ a density-Rips bifiltration. These turn out to be very useful in the study of noisy point cloud data. (See Section 9.1 above.)

We remark that, there are many other ways of defining a density estimator, e.g., kernel methods (for data in $\mathbb{R}^n$) and k-nearest neighbor methods. These have been widely studied in statistics. All choices depend on some choice of a *bandwidth parameter* (r in our case.)

Example 9.2 (Eccentricty-Rips bifiltration)**.** Define $\gamma : P \to \mathbb{R}$ by

$$\gamma(x) = \frac{1}{|P|} \sum_{y \in P} d(x, y).$$

Thus, $\gamma(x)$ is the average distance of x to all other points in P. We call γ an *eccentricity* function. We call $\mathcal{S}^{\downarrow}(\gamma)$ an eccentricty-Rips bifiltration. When P has the structure of multiple tendrils emanating radially from a central core. Superlevel sets of γ break up into clusters in corresponding to the tendrils of P, and $\mathcal{S}^{\downarrow}(\gamma)$ sees these tendrils as "persistent clusters."

Example 9.3. As indicated in Section 9.1, sometimes an interesting function is $\gamma : P \to \mathbb{R}$ comes to us from an application. For example, in recent work on computational chemistry, Bryn Keller, Ted Wilke and I considered P to be the atom centers of a ligand (drug candidate) and took γ to be the partial charge function. This turned out to be useful for defining a metric on ligands.

Degree-Rips Bilfiltration One disadvantage of the density-Rips filtration described above is that it depends on a choice of bandwidth parameter r. When working with two parameter persistence, we are already mapping data to 2-parameter objects which are already rather complex. It would be nice if these objects did not themselves depend on an additional nuisance parameter, whose value we may not be sure how to choose.

Fortunately, there is a simple way to define a variant of the function-Rips bifiltration which avoids having to choose such a parameter. The key observation is that both the bandwidth parameter r used to define γ_r and the scale parameter b for the Vietoris-Rips complex specify a scale in the metric space. Thus, it is natural to take r to depend on b. For example, we may take $r = 2b$. This yields a bifiltration $\mathrm{DRips} : \mathbb{N}^{op} \times \mathbb{R} \to \mathbb{R}$, the *degree-Rips* bifiltration.

This bifiltration has an appealingly simple alternative description, which we now give: Recall the definition of the neighborhood graph $N(P)_r$ from Section 4.4. For $d \geq 0$, let $N(P)_{d,r}$ be the graph obtained from $N(P)_r$ by removing all vertices of degree less than d. We take $\mathrm{DRips}_{d,r}$ to be the clique complex of $N(P)_{d,r}$.

9.3 Multi-Cover Bifiltration

I learned of the following very natural bifiltration from a 2012 paper by Don Sheehy, "A multi-cover nerve for geometric inference." Given $P \subset \mathbb{R}^n$ finite, let $\mathrm{Cov}(P) : \mathbb{N}^{\mathrm{op}} \times \mathbb{R} \to \mathbf{Top}$ be given by

$$\mathrm{Cov}(P)_{k,r} = \{y \in \mathbb{R}^n \mid \ y \text{ is within distance } r \text{ from at least } k \text{ points of } P\}.$$

This is a natural density-sensitive extension of the usual union-of-balls filtration.

As with the ordinary union-of-balls filtration, it is natural to ask whether there is a simplicial filtration with the same topology. Sheehy gave a very lovely positive answer to this question in the same paper mentioned above. We explain this below.

Subdivision-Rips and Subdivision-Čech Bifiltrations For S any simplicial complex, let $\mathrm{Bary}(S)$ denote the barycentric subdivision of S. Recall that the vertices of $\mathrm{Bary}(S)$ are in bijective correspondence with simplices of S. For $k \geq 0$, let S^k be the subcomplex of $\mathrm{Bary}(S)$ spanned by vertices of $\mathrm{Bary}(S)$ corresponding to simplices of dimension at least k. This defines a filtration

$$S^{\dim S} \subset \cdots \subset S^k \cdots S^{k-1} \cdots \subset S^1 \subset S^0 = \mathrm{Bary}(S).$$

If F is any simplicial filtration, construction this filtration for each simplicial complex of F yields a biltration.

In the case that $F = \mathrm{Cech}(P)$ for some $P \subset \mathbb{R}^n$, this yields a bifiltration $\mathrm{S\check{C}ech}(P) : \mathbb{N}^{\mathrm{op}} \times \mathbb{R} \to \mathbf{Simp}$, which we call the *subdivision-Čech bifiltration*. Similarly, in the case that F $F = \mathrm{Rips}(P)$ for some finite metric space P, this yields a bifiltration $\mathrm{SRips} : \mathbb{N}^{\mathrm{op}} \times \mathbb{R} \to \mathbf{Simp}$, which we call the *subdivision-Rips bifiltration*.

The following is a slight strengthening of the main result of Sheehy's 2012 paper. (The strengthening was and is also known to Sheehy, who explained it to me in person.)

Theorem 9.4 (Sheehy '12)**.** *For any finite $P \subset \mathbb{R}^n$, $\mathrm{S\check{C}ech}(P)$ is weakly equivalent to $\mathrm{Cov}(P)$. In particular, $H_i(\mathrm{S\check{C}ech}(P)) \cong H_i(\mathrm{Cov}(P))$ for all $i \geq 0$.*

The Subdivision-Rips and Subdivision-Čech Bifiltrations are attractive constructions of a density-sensitive bifiltrations on point cloud data: They don't depend on a nuisance parameter, the definitions are elegant, and these constructions turn out to be theoretically well behaved. However, the constructions are prohibitively large for practical use, because the barycentric subdivisions of Čech or Rips complexes simply have too many vertices.

In recent work (2018), Edelsbrunner and Osang have explored the development of an analogue of Delaunay filtrations in the multi-cover setting, with the goal of combinatorially capturing the topology of $\mathrm{Cov}(P)$ with a smaller construction. However, their results do not immediately extend to the full 2-parameter setting. (Instead, they are given for horizontal and vertical lines through the parameter space.)

Interlevel Bifiltration Working in the 2-parameter persistence setting allows us to define, for any function topological space T and function $\gamma : T \to \mathbb{R}$, a very natural bifiltration $\mathcal{S}(\gamma) : \mathbb{R}^{\mathrm{op}} \times \mathbb{R} \to \mathbf{Top}$. We call this the *interlevel bifiltration*. We define

$$\mathcal{S}(\gamma)_{x,y} = \begin{cases} \gamma^{-1}[x,y] & \text{if } x \leq y, \\ 0 & \text{otherwise.} \end{cases}$$

Under mild conditions (e.g., if γ is bounded), $\mathcal{S}(\gamma)_{x,y}$ encodes the superlevel and sublevel filtrations of γ. Interestingly, thanks to a Mayer-Vietoris argument, under mild conditions, the persistence modules $H_i(\mathcal{S}(\gamma)_{x,y})$ turn out to have a very simple algebraic structure—they decompose as a direct sum of simple modules called *block indecomposbales*. Hence, there is a nice notation of barcode for interlevel persistence modules refining the usual sublevel and

superlevel barcodes. This was first explored by Carlsson, de Silva, and Morozov, and then further developed by Bendich et al. That work depends on some strong conditions on the function γ. Recently, those conditions were relaxed considerably by Oudot and Chochoy.

We remark that the interlevel bifiltration is closely related to sheaf-theoretic formulations of persistence as developed e.g. in Justin Curry's thesis. This is an important aspect of the theory of multi-parameter persistence (broadly construed) but we will not have time to discuss in our course.

9.4 1-Critical and Multi-critical Bifiltrations

It is useful, particularly in the computational setting, to distinguish between two different types of bifiltrations (or more generally multi-parameter filtrations).

For simplicity, we state the definitions here for $\mathbb{R}^2$-indexed bifiltrations, though the definitions below also extend immediately to the other indexing categories we've considered for multi-parameter persistence modules.

Definition 9.5. For $F : \mathbb{R}^2 \to \mathbf{Top}$ a bifiltration, we let $\operatorname{colim} F = \bigcup_{r \in \mathbb{R}^2} F_r$, viewed as a topological space with the *final topology*. In the final topology, we take $S \subset \operatorname{colim} F$ to be open if and only if for each $r \in \mathbb{R}^2$, the restriction of S to F_r is open. In most examples we care about, $\operatorname{colim} F = F_{(r,r_2)}$ for all r_1 and r_2 large enough.

Definition 9.6 (1-Critical and Multi-critical Bifiltrations)**.** We say a bifiltration $F : \mathbb{R}^2 \to \mathbf{Top}$ is 1-critical if for each $x \in \operatorname{colim}(F)$, there is a unique minimal $r \in \mathbb{R}^2$ such that $x \in F_r$; we call z the *birth index* of x. If F is not 1-critical, it is said to be *multi-critical*. We say a bifiltration $F : \mathbb{Z}^2 \to \mathbf{Simp}$ is 1-critical (multi-critical) if its geometric realization is.

Exercise 9.7. *Check that the function-Rips, function-Cech, Subdivision-Rips, subdivision-Cech, and interlevel bifiltrations are all 1-critical, and that the degree-Rips and multi-cover filtrations are multi-critical.*

9.5 Multi-Parameter Sublevel Filtrations

Definition 9.8 (Cerri et al. 2011)**.** For T a topological space and $\gamma : T \to \mathbb{R}^n$ any function, define a bifiltration $\mathcal{S}^{\uparrow}(\gamma) : \mathbb{R}^n \to \mathbf{Top}$ by

$$\mathcal{S}^{\uparrow}(\gamma)_r = \{x \in T \mid \gamma(x) \leq r\}.$$

The filtration $\mathcal{S}^{\uparrow}(\gamma)$ is clearly 1-critical. Conversely, we have the following, which tells us that the sublevel multi-filtration generalizes several constructions we've seen so far.

Proposition 9.9 (L., Blumberg 2017)**.** *If F is 1-critical bifiltration for which each inclusion $F_r \hookrightarrow F_s$ is closed, then F is, up to isomorphism, a sublevel filtration. Specifically, let $T = \operatorname{colim} F$, and define $\gamma : T \to \mathbb{R}^n$ by taking $\gamma(x)$ to be the birth index of x. We then have that*

$$\mathcal{S}^{\uparrow}(\gamma) \cong F.$$

The condition that $F_{x,y}$ is a closed inclusion holds for all simplicial filtrations, or more generally for all CW-filtrations.

10 The Difficulty of Defining a Barcode of a 2-Parameter Persistence Modules

We have now described several very natural constructions for mapping data to a bifiltration. As noted earlier, applying homology with coefficients in a field then gives us a persistence module. We now must confront the question of whether we can define a version of the barcode for 2-parameter persistence modules.

10.1 Non-Existence of a "Good Barcode"

Newcomers to 2-parameter persistence often ask whether there is a way to define the barcode of a 2-parameter persistence module as a collection of nice regions in the paper; after all, the usual barcode is a collection of nice regions in $\mathbb{R}$, namely, a collection of intervals. However, there turns out to be no good way to do this, in the following sense:

Definition 10.1. A *good barcode* for an $\mathbb{N}^2$-indexed bipersistence module M is a collection $\mathcal{B}_M$ of subsets of $\mathbb{R}^2$ such that for each $a \leq b \in \mathbb{R}^2$

$$\operatorname{Rank} M_{a,b} = |\{S \in \mathcal{B}_M \mid a, b \in S\}|.$$

One can check that this condition is satisfied by the usual barcode of an $\mathbb{R}$-indexed persistence module. In my view, this notion of goodness is the bare minimum we might reasonably ask of a barcode in the 2-parameter setting (assuming at least that the barcode is defined as a collection of subsets of $\mathbb{R}^2$).

Yet the following example gives a 2-parameter persistence module with no good barcode:

Exercise 10.2. *Consider the $\mathbb{N}^2$-indexed persistence module*

$$\begin{array}{ccccccc}
0 & \longrightarrow & K & \xrightarrow{=} & K & \longrightarrow & 0 \\
\uparrow & & \uparrow\!\uparrow & & \uparrow\!\uparrow & & \uparrow \\
K & \xhookrightarrow{f} & K^2 & \xrightarrow{=} & K^2 & \twoheadrightarrow & K \\
\uparrow & & \uparrow g & & \uparrow = & & \uparrow = \\
0 & \longrightarrow & K & \xhookrightarrow{g} & K^2 & \twoheadrightarrow & K \\
\uparrow & & \uparrow & & \uparrow h & & \uparrow \\
0 & \longrightarrow & 0 & \longrightarrow & K & \longrightarrow & 0
\end{array}$$

where $f = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $g = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, $h = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ and all vector spaces not shown are 0. Check that this has no good barcode.

10.2 A Deeper Look at the Problems with Defining a Barcode

We now investigate more fully algebraic structure of 2-parameter persistence modules, and how the complexities of this structure lead to fundamental difficulties in defining a notion of barcode that is simple enough to work with in practice. This problem was first explored by Carlsson and Zomorodian in their original 2006/2009 paper on multi-parameter persistence. Subsequently, the TDA community has come to understand that the issues are best understood through the lends of standard ideas in quiver representation theory. (That perspective was, arguably, implicit in the paper of Carlsson and Zomorodian, but the language of quiver representations was not explicitly used there.)

We begin with some good news about the algebraic structure of persistence modules. For $\mathcal{J}$ any small category, a functor $M : \mathcal{J} \to \mathbf{Vect}$ is said to be *indecomposable* if whenever $M \cong M_1 \oplus M_2$, we have that either $M_1 = 0$ or $M_2 = 0$.

Theorem 10.3 (Krull-Schmidt-Azumaya Theorem)**.** *For any small category $\mathcal{J}$ and functor $M : \mathcal{J} \to \mathbf{Vect}$, there is a collection of indecomposables $\{M_\alpha\}_{\alpha \in A}$, unique up to isomorphism such that*

$$M \cong \bigoplus_{\alpha \in A} M_\alpha.$$

This in particular tells us that persistence modules decompose in an essentially unique way into indecomposables.

For various special cases, results like this have been known for a long time. Note that if $\mathcal{J}$ has a finite set of objects, the existence portion of this theorem is more or less trivial: if M has a non-trivial direct summand, then M', then $\dim M'_x \leq \dim M_x$ for each object $x \in \mathcal{J}$ and $\dim M'_x < \dim M_x$ for at least one object $x \in \mathcal{J}$. It follows immediately form this fact and the p.f.d. property of M that a direct sum decomposition of M exists. From this, it is also not difficult to deduce the existence result for finitely-generated $\mathbb{Z}^n$-indexed persistence modules or finitely presented $\mathbb{R}^n$-indexed persistence modules. In these cases, the number of indecomposables in the decomposition of M is finite. The uniqueness portion of the theorem is, to the best of my understanding, due to Azumaya. In the generality given here, the full result was sketched in work of Gabriel and Roiter in 1992, proven in detail in work of Crawley-Bowevey from 1994, and then given a short direct proof in recent work of Crawley-Boevey and Botnan in 2018. (See the paper of Crawley-Boevey and Botnan for references.)

Theorem 10.3 tells us that as in the 1-parameter case, an n-parameter persistence module decomoses into essentially unique building blocks. (The structure theorem tells us that for $\mathbb{Z}$ or $\mathbb{R}$-indexed persistence modules, the indecomposables are exactly the interval modules.)

However—and here now is the bad news—for n-parameter persistence modules, $n \geq 2$, it turns out that the collection of isomorphism classes of persistence modules is enormously complicated. Giving an explicit parameterization of this collection is generally considered by mathematicians to be a hopeless problem. Thus while in principle one could define the barcode of a p.f.d. persistence module to be the collection of isomorphism classes of its

indecomposables, we don't have a good way of explicitly describing this barcode. (We can give presentations of the indecomposables, but this description is non-unique, hence unlikely to be directly useful in the way that a bacode is usually used in TDA.)

To flesh out the details of this story, we give a brief overview of quiver representation theory.

10.3 Quiver Representations

Quiver representation theory studies the indecomposables of (not-necessarily) commutative diagrams of vector spaces. It is a decades old subject, and very well developed. Good introductory resources include a Notices article by Derksen and Weyman and the appendix of Steve Oudot's book on persistence. The textbook book of Assem, Simpson, and Skowroński also comes highly recommended by experts in the field. This has three volumes. Some of what is below is discussed in Volume 3. In preparing this exposition, I consulted with Magnus Botnan and Uli Bauer about some technical questions. I thank them for their help.

Definition 10.4. A quiver $Q = (V, E)$ is a finite directed graph. Multiple edges between a pair of vertices are allowed, as are self-edges. We write a directed edge e from vertex v to vertex w as $e : v \to w$.

Definition 10.5. Let us fix a choice of field K. Informally, *representation* M of a quiver $Q = (V, E)$ is a (not necessarily) commutative diagram of vector spaces indexed by Q. Formally, M consists of:

- A choice of K-vectors space M_v for each $v \in V$.
- A choice of linear map $M_{v,w} : M_v \to M_w$ for each edge $e : v \to w \in E$.

Remark 10.6. Any category $\mathcal{J}$ with finite object set and finite hom set determines a quiver $Q = (V, E)$ with $V = \operatorname{ob} \mathcal{J}$ and $E = \hom \mathcal{J}$. A functor $\mathcal{J} \to \mathbf{Vect}$ determines a representation of Q. Thus, quiver representations generalize functors $\mathcal{J} \to \mathbf{Vect}$ for $\mathcal{J}$ is a finite category.

Definition 10.7. For M, N representations of a quiver $Q = (V, E)$, we define a morphism $f : M \to N$ to be a collection of linear maps $\{f_v\}_{v \in V}$ such that for all $e : v \to w \in E$, the following diagram commutes:

$$\begin{array}{ccc} M_v & \xrightarrow{M_e} & N_w \\ {\scriptstyle f_v}\downarrow & & \downarrow{\scriptstyle f_w} \\ N_v & \xrightarrow[N_e]{} & N_w \end{array}$$

Note the similarity to the definition of natural transformation; the definitions are nearly identical. With this definition of morphism, the representations of Q form a category, which we denote as $\operatorname{Rep}(Q)$.

Direct sums of quiver representations are defined pointwise, in essentially the same way as for **Vect**-valued functors. Thus we also obtain a definition of indecomposable representations.

10.4 Classification of Quivers

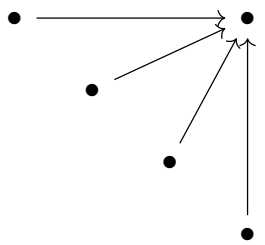
Definition 10.8. A quiver is of *finite type* if it has a finite number of isomorphism classes of indecomposables.

Example 10.9. The structure theorem for persistence modules shows that a quiver of the form $\bullet \to \bullet \to \cdots \to \bullet$ is of finite type.

Definition 10.10. Informally, a quiver is of *tame* type if the collection of all isomorphism classes of indecomposable representations can be parameterized as the disjoint union of countably many 1-parameter families of iso. classes of indecomposables.

Example 10.11. For K an algebraically closed field, the quiver with one vertex and one edge is tame. The indecomposables whose vector space has dimension n correspond bijectively to the $n \times n$ Jordan blocks, and these form a 1-parameter family.

Example 10.12. The following quiver is tame.



Definition 10.13. A quiver Q is of *wild* type if for any other quiver Q', there is a functor $F : \mathrm{Rep}(Q') \to \mathrm{Rep}(Q)$ with the following properties:

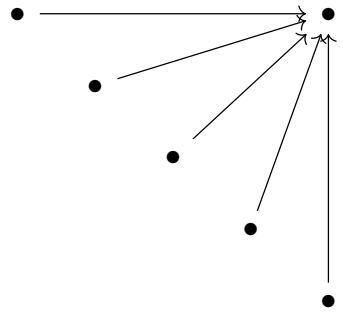
- F is fully faithful
- F is exact (i.e., it preserves exactness of sequences).

Remark 10.14. The literature contains various definitions of wild. The one given here is called *strongly wild type* in Assem et al. volume 3, Chapter XIX, Definition 1.3(b). As explained there, the condition that F is fully faithful implies that F sends indecomposables to indecomposables. The condition that F is fully faithful implies that $F(M) \cong F(N)$ then $M \cong N$, and that if $M = M_1 \oplus M_2$, then $F(M) = F(M_1) \oplus F(M_2)$. In these senses, the functor F "embeds" the representation theory of Q' into the representation theory Q.

Thus, the representation theory of a wild quiver "contains" the representation theory of any other quiver Q.

Example 10.15. The quiver with a single vertex and two edges is of wild type. The construction of F is relatively simple.

Example 10.16. The following quiver is wild.



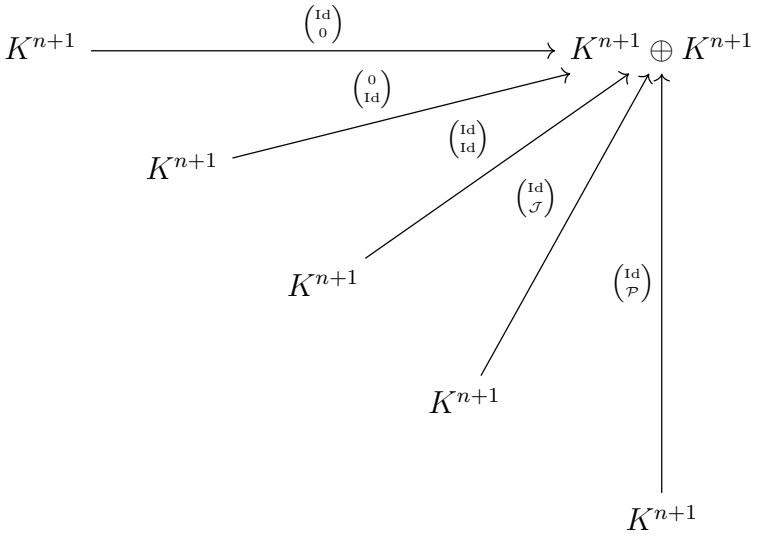
The following "trichotomy" result is remarkable and surprising.

Theorem 10.17 (Drozd)**.** *Over an algebraically closed field K , every quiver is of finite type, tame, or wild.*

The problem of parameterizing the indecomposables of aquiver of wild type is generally considered to be hopelessly difficult. A hint of the complexity of this is given by the following example.

Example 10.18. For the wild quiver of Example 10.16, we exhibit n-parameter families of indecomposable representations, for each positive integer n. The example was presented by Steffan Oppermann's 2017 Lecture at Banff, an introduction to quiver representation theory which is recorded and available online.

For $\vec{\alpha} \in \mathbb{R}^n$, Consider the quiver representation $M(\vec{\alpha})$, given by



where the morphisms are given in block form, $\mathcal{J}$ is the $(n+1) \times (n+1)$ Jordan block with zero diagonal, and $\mathcal{P} = \mathcal{P}(\vec{\alpha})$ is the matrix whose entries on the super-diagonal are $\vec{\alpha}$, and whose remaining entries are 0. $M(\vec{\alpha})$ is indecomposable, and if $\vec{\alpha} \neq \vec{\alpha}'$, then $M(\vec{\alpha}) \ncong M(\vec{\alpha}')$.

Remark 10.19. It follows from Example 10.18 and the definition of wildness that any wild quiver has n-parameter families of indecomposable representations for each positive integer n. Theorem 10.17 then tells us in particular that (for K algebraically closed) if a quiver has 2-parameter families of indecomposable representations, then it has n-parameter families of indecomposable representations for each positive integer n.

Explicit Description of the Finite Type and Tame Quivers It turns out that whether a quiver is finite type does not depend on the orientation of the arrows in the quiver, but only on the underlying directed graph. The same is true for tame type. One can give a complete enumeration of the finite type and tame quivers without much trouble—they fall into just a few families. I gave the enumeration in class, but because it would be burdensome to TeX the diagrams. So for a written reference on this I am just going to refer you to the article "Quiver Representations" by Harm Derksen and Jerzy Weyman, in the Notices of the AMS. The enumeration is given on the third page.

Zigzag Persistence modules Of particular interest to applied topologists is the fact that a quiver of the form

$$\bullet \to \bullet \leftarrow \bullet \to \cdots \leftarrow \bullet \to \bullet \leftarrow \bullet$$

is of finite type. In TDA, we call representations of such a quiver *zigzag persistence modules*. In fact, the indecomposables of a zig-zag persistence module are analogues of the interval modules we have in the case where all the arrows are pointing right, i.e., they look for example like this:

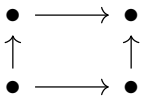
$$0 \to 0 \leftarrow 0 \to K \xleftarrow{\mathrm{Id}_K} K \xrightarrow{\mathrm{Id}_K} K \xleftarrow{\mathrm{Id}_K} K \xrightarrow{\mathrm{Id}_K} K \leftarrow 0 \to 0 \leftarrow 0 \to 0 \leftarrow 0$$

Remark 10.20. Magnus Botnan recently showed that this decomposition result extends to the case where the zig-zag quiver extends out infinitely in both directions. This was a folklore result in representation theory, but there was apparently no proof in the literature. Botnan gives a very nice, short proof, which reduces the problem to the case of finite zigzag quivers.

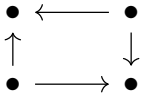
10.5 Commutative Quiver Representations and Bipersistence Modules

The theory outlined above concerns the incomposables of not-necessarily-commutative diagrams of vector spaces. If one restricts attention to commutative diagrams of vector spaces indexed by quivers, the definitions of finite-type, tame, and wild quivers still make sense, and it turns out that there is an analogous trichotomy theorem for the commutative setting. However, the quivers falling into each category are now different, and now do depend on

the orientation of the edges. For example, as Magnus Botnan has explained to me, in the commutative setting, the quiver



is finite type, but the quiver

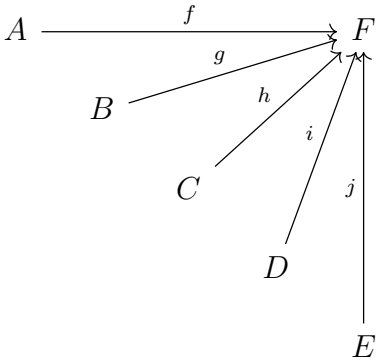


is tame.

For quivers having the shape of a finite rectangular grid with all arrows pointing up or to the right, it is known that the 2×5, 5×2, and 3×3 grids are tame; all grids strictly contained in one of these are finite type; and all other grids are wild.

A commutative representation of a finite rectangular grid G can always be extended to a $\mathbb{Z}^2$-indexed persistence module by taking the vector spaces indexed by vertices not in G to be 0. This extension defines a fully faithful and exact functor. It follows that the representation theory of 2-parameter persistence modules is wild, i.e., as complicated as the representation theory of any quiver representation.

Example 10.21. Let Q be the quiver with 5 sources and one sink from Example 10.16. We show that the wildness of of 2-parameter persistence follows from the wildness of Q. It suffices to define an exact, fully faithful functor $F : \mathrm{Rep}(Q) \to \mathrm{Fun}(\mathbb{Z}^2 \to \mathbf{Vect})$. We define F on objects as follows, If M is given by the following diagram



then $F(M)$ is the bipersistence module whose restriction to a 5×5 grid is

$$\begin{array}{ccccccccc}
A & \xrightarrow{f} & F & \longrightarrow & F & \longrightarrow & F & \longrightarrow & F \\
\uparrow & & {\scriptstyle g}\uparrow & & \uparrow & & \uparrow & & \uparrow \\
0 & \longrightarrow & B & \xrightarrow{g} & F & \longrightarrow & F & \longrightarrow & F \\
\uparrow & & \uparrow & & {\scriptstyle h}\uparrow & & \uparrow & & \uparrow \\
0 & \longrightarrow & 0 & \longrightarrow & C & \xrightarrow{h} & F & \longrightarrow & F \\
\uparrow & & \uparrow & & \uparrow & & {\scriptstyle i}\uparrow & & \uparrow \\
0 & \longrightarrow & 0 & \longrightarrow & 0 & \longrightarrow & D & \xrightarrow{i} & F \\
\uparrow & & \uparrow & & \uparrow & & \uparrow & & {\scriptstyle j}\uparrow \\
0 & \longrightarrow & 0 & \longrightarrow & 0 & \longrightarrow & 0 & \longrightarrow & E
\end{array}$$

where all maps between copies of F are the identity. Outside of the 5×5 grid, we take all vector spaces of $F(M)$ to be 0. The action of F on morphisms is defined in the obvious way. It is easy to check that this functor is fully faithful and exact.

Note that by applying F to the indecomposables of Example 10.18, we get n-parameter families of indecomposable persistence modules for each positive integer n.

11 The Muti-Parameter Interleaving Distance

11.1 Muti-Parameter Interleavings

The definitions of interleavings and the interleaving distance we studied earlier generalize readily to multi-parameter persistence modules. Fix $n \geq 1$ and for $\delta \in [0, \infty)$, let $\vec{\delta} = (\delta, \delta, \ldots, \delta) \in \mathbb{R}^n$.

For any $\mathbb{R}^n$-indexed persistence module M and $\delta \in \mathbb{R}$, we define the *shift of M by δ* to be the $\mathbb{R}^n$-indexed persistence module module $M(\delta)$ given by $M(\delta)_a = M_{a+\vec{\delta}}$, $M(\delta)_{a,b} = M_{a+\vec{\delta},b+\vec{\delta}}$. This definition is functorial, i.e., for any morphism $f : M \to N$ of persistence modules, we have an induced morphism $f(\delta) : M(\delta) \to N(\delta)$. Note that the internal maps $\{M_{a,a+\vec{\delta}}\}_{a \in \mathbb{R}^n}$ assemble into a morphism of persistence modules $\phi^{M,\delta} : M \to M(\delta)$.

Given these definitions, we can define δ-interleavings and the interleaving distance exactly as in the 1-parameter case: A δ-interleaving between $\mathbb{R}^n$-indexed persistence modules M and N is defined to be a pair of morphisms

$$f : M \to N(\delta) \qquad g : N \to M(\delta)$$

such that

$$g(\delta) \circ f = \phi^{M,2\delta} \qquad f(\delta) \circ g = \phi^{N,2\delta}.$$

We define the multi-parameter interleaving distance d_I by

$$d_I(M, N) = \inf \{\delta \mid \text{There exists a } \delta\text{-interleaving between } M \text{ and } N\}.$$

As in the 1-parameter case, d_I is an extended pseudometric on persistence modules. In fact, I showed this in my thesis work that d_I descends to an extended metric on isomorphism classes of finitely presented $\mathbb{R}^n$-indexed modules.

11.2 Universality of the Multi-Parameter Interleaving Distance

The main result from my thesis work shows that when the field K is prime (i.e., $K = \mathbb{Z}/p\mathbb{Z}$ or $K = \mathbb{Q}$), the interleaving distance is the most sensitive distance satisfying a reasonable stability property. Here, we give the precise statement of the result and offer a few brief remarks.

First, we briefly explain the motivation for the result. Prior to this work on universality of the interleaving distance, there had been a number of distances proposed on multi-parameter persistence modules which satisfied a reasonable stability property. In order to develop TDA theory in the multi-parameter setting, we would like to select once and for all a good choice of distance. The universality result for the interleaving distance tells us that in a certain relative sense, the interleaving distance is the optimal choice.

To formulate the result, we first need to extend the definition of the sup-norm distance given in Section 8.3 to $\mathbb{R}^n$-valued functions.

Definition 11.1. For T be a topological space and $\gamma, \kappa : T \to \mathbb{R}^n$ any functions, we define the *sup-norm distance* between γ and κ by

$$d_\infty(\gamma, \kappa) := \sup_{x \in T} \|\gamma(x) - \kappa(x)\|_\infty.$$

Definition 11.2. A distance d (i.e., extended pseudometric) on multi-parameter persistence modules is *stable* if for all topological spaces T and functions $\gamma, \kappa : T \to \mathbb{R}^n$ ad $i \geq 0$, we have

$$d_I(H^i(\mathcal{S}^\uparrow(\gamma)), H^i(\mathcal{S}^\uparrow(\kappa)) \leq d_\infty(\gamma, \kappa).$$

Theorem 11.3 (L. 2015)**.**

(i) The interleaving distance d_I on multi-parameter persistence modules is stable.

(ii) Assume that the field K is prime. Then for any other stable distance d on multi-parameter persistence modules, we have $d(M, N) \leq d_I(M, N)$ for all persistence modules M and N.

A few remarks:

1. The generalization of Theorem 11.3 to arbitrary fields is an open question.
2. In a paper from 2017, Andrew Blumberg and I have established a version of this universality result on filtrations for a homotopy-invariant version of the interleaving distance.

3. A version of Theorem 11.3 was previously known for the special case of 1-parameter persistence and 0^{th} homology. This is work of Frosini and coauthors.

The proof of the Theorem 11.3 boils down to a lifting construction: One shows that if persistence modules M and N are δ-interleaved, then there exists a topological space T and functions $\gamma^M, \gamma^N : T \to \mathbb{R}^n$ such that

1. $d_\infty(\gamma^M, \gamma^N) = \delta$,
2. $H_1(\mathcal{S}^\uparrow(\gamma^M)) \cong M$,
3. $H_1(\mathcal{S}^\uparrow(\gamma^N)) \cong N$.

To show this, the key step is to show that if M and N are δ-interleaved, then there are presentations for M and N that are compatible, in the sense that they differ from one another only by shifting the grades of generators and relations by at most $\vec{\delta}$.

A discussion of applications of the 2-parameter interleaving distance will be added later to these notes.

12 Hardness of Computing the Interleaving Distance

Computation of Bottleneck Distance The bottleneck distance on 1-parameter persistence modules (which as discussed earlier, is equal to the 1-parameter interleaving distance) turns out to be readily computable. The problem can be cast as matching problem of the kind commonly studied in computer science. The state of the art approach is described in a paper 2016 by Kerber and Morozov, Nigmetov and has implemented in the Hera software package. The algorithm runs in time $O(n^{1.5} \log n)$, where n is the total number of intervals in the two barcodes.

The question of if and how the multi-dimensional interleaving distance can be computed has been of interest to researchers working on the theoretical foundations of TDA for several years. The main conjecture here has been that the interleaving distance is *NP-hard to compute*. Recently, this was settled by the combined results of two papers—the first by Bjerkevik and Botnan and the second by Bjerkevik, Botnan, and Kerber. Together, these works show that over a finite field K, the n-parameter interleaving distance is indeed NP-hard to compute for any $n \geq 2$.

In this section, we will give an informal review of NP-hardness and then give a very brief outline of the strategy of Botnan and Bjerkevik for proving NP-hardness. (These papers also have a number of other very interesting results that will not be discussed here.)

12.1 NP-Completeness and NP-Hardness

The following discussion of basic concepts on computer science will be informal. For example, we will not bother to give formal definitions of an algorithm or a computational problem.

Definition 12.1. A computational problem A is said to be *in the complexity class* P if there exists an algorithm which solves A in time polynomial in the size of the input—that is, there is some polynomial f such that the number of operations required by this algorithm for an input of size n is at most $f(n)$.

For example,

- sorting a list of n elements can be solved in $O(n \log n) = O(n^2)$ time, so the problem of sorting a list is in P.
- Solving a linear system in n equations and n variables over a finite field takes $O(n^3)$ time, so is in P.
- Computing the bottleneck distance between two barcodes is in P.

Definition 12.2. A *decision problem* is a computational problem whose answer is "yes" or "no".

Example 12.3. The problem of determining whether a solution exists to a linear system is a decision problem, but actually solving the linear system is not a decision problem.

Definition 12.4. A decision problem A is *in the complexity class NP* if for any instance with the problem for the answer is yes, there is a *certificate of proof* that the answer is yes such that given this certificate, we can verify that the answer is yes in polynomial time.[6]

Example 12.5. Any decision problem in P is in NP: The certificate of proof can be taken to be trivial: To verify the answer is yes, we just solve the problem.

Example 12.6 (3-SAT)**.** Here is a famous example of a problem which is in NP but is not known to lie in P (more on this below): Consider a boolean expression of the form

$$(x_1 \vee x_2 \vee x_3) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_7 \vee x_8 \vee x_9) \wedge \cdots \wedge (x_{3n-2} \vee x_{3n-1} \vee x_{3n}),$$

where each x_i is a variable. The 3-SAT problem is the problem of deciding whether whether exists an assignment of each of the variables such that the expression evaluates to 1.

Here, the certificate of proof for a "yes" instance of the problem is simply an assignment of the variables such that the expression evaluates to 1.

There are many, many problems of importance to applications which are in NP but for which no polynomial time solution is known. One wants establish theoretically that such problems are hard. However, the question of whether every problem in NP is also in P is the most important open problem computer science (the P=NP problem), and is arguably the

[6] (NP stands for "non-deterministic polynomial" time; this is related to the technical concept of non-deterministic Turing machines. This may give the impression that NP some how refers to randomized algorithms, but that is not the case.

second most famous open math problem in the world (the Riemann hypothesis being the first.) The answer to this question is widely thought to be "no."

Since showing directly that a problem is not in P is very hard, we measure the hardness of problems by comparing their hardness to that of other problems. For this, the following definition is key.

Definition 12.7. Given two computational problems A and B, we say that A *reduces to* B *in (polynomial time)* if A can be solved by solving polynomially many instances of B, plus doing a polynomial amount of additional work. We will sometimmes write $A \implies B$ to denote that A reduces to B.

If A reduces to B, we think of B is being at least as hard as A. (In this interpretation, a polynomial amount of work is regarded negligible.)

Definition 12.8. A problem A (not necessarily a decision problem, and not necessarily in NP) is said to be *NP-hard* if any problem in NP reduces to A.

Thus an NP-hard problem is a problem that is at least as hard as any problem in NP, in the above sense; In particular, if you could solve one NP-hard problem in polynomial time, you could solve any problem in NP in polynomial time, which would imply P=NP.

Definition 12.9. A problem which is NP-hard and also in NP is called NP-complete.

Theorem 12.10 (Cook-Levin Theorem)**.** *3-SAT is NP-complete.*

In fact, 3-SAT was the first problem to be shown to be NP-complete.

Proposition 12.11. *If a problem A is NP-hard and A reduces to a problem B, then B is NP-hard.*

Proof. The proof of this amounts to the easy observation that if $C \implies A$ and $A \implies B$, then $C \implies B$. That is, the "reduces to" relation is transitive. □

Proposition 12.11 suggests a general strategy for showing that a problem B is NP-hard: Show that some NP-hard problem A reduces to B. Using this strategy, thousands of problems of interest in applications have been shown to be NP-hard. Of course, this strategy is only viable if one can show directly that at least one problem is NP-hard. That is why the Cook-Levin theorem is important. In fact, showing NP-hardness by reducing to 3-SAT is very common.

12.2 Hardness of Computing the Interleaving Distance

We consider the problem of computing the interleaving distance d_I between finitely generated $\mathbb{Z}^n$-indexed persistence modules. (The hardness result for this case immediately yields a hardness result for the $\mathbb{R}^n$-indexed case.)

We assume the modules are given to us as input via finite presentation.

Definition 12.12. Let 1-IL denote the problem of deciding whether two persistencem odules are 1-interleaved.

Exercise 12.13. *If Modules M and N are δ-interleaved, then they are also δ'-interleaved for all $\delta' > \delta$.*

Lemma 12.14. *1-IL reduces to the problem of computing d_I*

Proof. For $\mathbb{Z}^n$-indexed modules, the interleaving distance can only take values in the non-negative integers. Hence, by the above lemma, if $d_I(M, M) \leq 1$, then M and N are 1-interleaved. On the other hand, if $d_I(M, N) > 1$, then M and N are are not 1-interleaved. □

Thus, to show that computing d_I is NP-hard, it suffices to show that 1-IL is NP-Hard.

Definition 12.15 (The Constrained Invertibility Problem (CI))**.** Consider two $m \times n$ matrices A, B with some of the entries in each matrix set to 0, and remaining entries left unspecified (i.e., given as variables). The constrained invertibility problem is to decide whether there exists an assignment of the each of the unspecified entries of A and B such that the resulting matrices are inverses.

Example 12.16. . This is example is taken from the paper of Bjerkevik, Botnan, and Kerber. For

$$A = \begin{pmatrix} * & * & * \\ * & 0 & * \\ * & * & * \end{pmatrix}, \quad B = \begin{pmatrix} * & * & * \\ * & * & 0 \\ * & 0 & * \end{pmatrix},$$

The contrained invertibility problem has a solution, namely,

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Theorem 12.17 (Bjerkevik, Botnan 2018)**.** *CI reduces to 1-IL.*

The above result represented nice partial progress on the complexity of computing the interleaving distance. The following result

Theorem 12.18 (Bjerkevik, Botnan, Kerber 2018)**.** *3-SAT reduces to CI.*

In view of this theorem and the ones mentioned above, we have a sequence of reductions

$$\text{3-SAT} \implies \text{CI} \implies \text{1-IL} \implies \text{computing } d_I,$$

It follows that d_I is NP-Hard.

Remark 12.19. In fact, Bjerkevik et al. show that computing d_I is NP-Hard even when we restrict attention to persistence modules which are direct sums of indecomposables with vector spaces of dimension at most 1. Moreover, the authors show that the problem is NP-Hard even when we assume that both persistence modules are indecomposable.

13 Invariants of 2-Parameter Persistence Modules

To extend typical applications of persistent homology to the n-parameter setting for $n \geq 2$, one natural thing to do is to work with simple invariants of a persistence module that serve as a surrogate for the barcode. There are many reasonable options for such invariants; which is most appropriate will depend the application one wishes to pursue and what the data looks like.

Here, we introduce a few simple, practical invariants of multi-parameter persistence modules, with an emphasis on the invariants that the software RIVET for 2-parameter persistence computes and visualizes. All the invariants considered here are readily computable, particularly so in the 2-parameter setting that is of most interest to us.

Definition 13.1. An *invariant* of n-parameter persistence modules is a function f from a collection of persistence modules to some set S with the property that $f(M) = f(M')$ whenever $M \cong M'$. An invariant f is said to be *complete* if, conversely, $M \cong M'$ whenever $f(M) = f(M')$.

13.1 The Hilbert Function

Definition 13.2. The *Hilbert function* of a p.f.d. R^n-indexed persistence module M, is the function $\mathrm{hf} : \mathbb{R}^n \to \mathbb{N}$ given by $\mathrm{hf}(a) = \dim M_a$. While this might seem like a lofty name for such a simple object, the name is standard in commutative algebra.

In TDA settings, this invariant has poor stability properties: If I perturb M a small amount with respect to the interleaving distance, the Hilbert function can change quite a lot. this of course is true even for 1-parameter persistence modules, and is part of what motivates persistent homology. Nevertheless, it turns out that the Hilbert function can be very useful in the exploratory analysis and visualization of 2-parameter persistence modules, and also useful for other purposes. For example, in a recent paper, Bryn Keller, Ted Wilke, and I found that the Hilbert function performs well as a molecular signature in the context of virtual ligand screening.

13.2 The Rank Invariant and Fibered Barcode

In the paper which introduced multi-parameter persistent homology, Carlsson and Zomorodian suggested the following invariant as a surrogate for the barcode:

Definition 13.3. For $\mathcal{J} = \mathbb{R}^n$ or $\mathcal{J} = \mathbb{Z}^n$, the *rank invariant* of a p.f.d. $\mathcal{J}$-indexed persistence module M is the function which sends a pair of indices $a \leq b \in \mathbb{R}^n$ to $\mathrm{Rank}\, M_{a,b}$.

Proposition 13.4 (Carlsson, Zomorodian '09)**.** *When $n = 1$ the rank invariant is complete.*

Sketch of proof. Let M be a p.f.d. persistence module. It suffices to observe that for any interval I the number of copies of I in $\mathcal{B}_M$ is determined by the rank invariant. For $\mathbb{N}$-indexed modules, we gave explicit formulae for the barcodes in Section 6.4, which we now recall:

- For any $a < b \in \mathbb{N}$, the number of copies of $[a, b)$ in $\mathcal{B}_M$ is

$$\operatorname{Rank} M_{a,b-1} - \operatorname{Rank} M_{a,b} - \operatorname{Rank} M_{a-1,b-1} + \operatorname{Rank} M_{a-1,b}.$$

- For any $a \in \mathbb{N}$, the number of copies of $[a, \infty)$ in $\mathcal{B}_M$ is

$$\lim_{b\to\infty} \operatorname{Rank} M_{a,b} - \lim_{b\to\infty} \operatorname{Rank} M_{a-1,b}.$$

For finitely generated $\mathbb{Z}$-indexed modules, the same formula completely describes the barcode. For p.f.d. modules, one can also have intervals which extend infinitely to the left, but similar formulae can be given for these as well. Finally, for the case of p.f.d. $\mathbb{R}$-indexed modules, one can also give analogues of these formulae which depend only on the rank; these are implicit in Crawley-Boevey's proof of the structure theorem for $\mathbb{R}$-indexed modules. □

Given the discussion earlier of the wildness of 2-parameter persistence modules, we expect the rank invariant to be incomplete for $n \geq 2$. The following example shows that this is indeed the case:

Example 13.5 (L., Wright, 2015)**.** The two $\mathbb{N}^2$-indexed modules shown below (the first of which is free on two generators) have the same rank invariant, but are not isomorphic:

$$\begin{array}{ccccccc}
\vdots & & \vdots & & \vdots & & \\
\uparrow & & \uparrow & & \uparrow & & \\
k & \longrightarrow & k^2 & \longrightarrow & k^2 & \longrightarrow & \cdots \\
\uparrow & \scriptstyle\binom{1}{0} & \uparrow & & \uparrow & & \\
k & \longrightarrow & k^2 & \longrightarrow & k^2 & \longrightarrow & \cdots \\
\uparrow & & \uparrow\scriptstyle\binom{0}{1} & & \uparrow & & \\
0 & \longrightarrow & k & \longrightarrow & k & \longrightarrow & \cdots
\end{array}
\qquad\qquad
\begin{array}{ccccccc}
\vdots & & \vdots & & \vdots & & \\
\uparrow & & \uparrow & & \uparrow & & \\
k & \longrightarrow & k^2 & \longrightarrow & k^2 & \longrightarrow & \cdots \\
\uparrow & \scriptstyle\binom{1}{0} & \uparrow & & \uparrow & & \\
k & \longrightarrow & k^2 & \longrightarrow & k^2 & \longrightarrow & \cdots \\
\uparrow & & \uparrow\scriptstyle\binom{1}{0} & & \uparrow & & \\
0 & \longrightarrow & k & \longrightarrow & k & \longrightarrow & \cdots
\end{array}$$

While the rank invariant doesn't encode all of the algebraic structure of a 2-parameter persistence module, it evidently does captures some essential information about "what features persist" in the homology module of a bifiltration. As such, Carlsson and Zomorodian's proposal to consider the rank invariant has been influential in the applied topology community, and there has generally been agreement that this invariant is of interest in applications.

Carlsson and Zomorodian did not offer any concrete suggestion for how the rank invariant might be used in practice. However, a simple and nice idea of Cerri, Di Fabio, Ferri, Frosini, and Landi, preseted a few years later, suggested a way forward. Cerri and his coauthors observed that the rank invariant is equivalent to what we will call the *fibered barcode*, which we now define. For simplicity, we will define this just in the $\mathbb{R}^2$-indexed case, though the definition does extend immediately to the $\mathbb{R}^n$-indexed case.

Let $\mathcal{L}$ denote the set of all affine lines in $\mathbb{R}^2$ with non-negative (possibly infinite) slope. For M an $\mathbb{R}^2$-indexed persistence module, the restriction M^L of M along L is a functor

$M^L : L \to \mathbf{Vect}$. L is isomorphic to $\mathbb{R}$ as a poset, so if M is p.f.d., the barcode $\mathcal{B}_{M^L}$, is well defined. This barcodes is a collection of intervals in L.

Definition 13.6. The *fibered barcode* of a p.f.d. bipersistence module M is the function with domain $\mathcal{L}$ given by

$$L \mapsto \mathcal{B}_{M^L}.$$

Proposition 13.7 (Cerri et al. 2011)**.** *The rank invariant and fibered barcode of a bipersistence module determine each other.*

Proof. To show that the fibered barcode barcode determines the rank invariant, consider any $a \leq b \in \mathbb{R}^2$, and let L be the unique line passing through both a and b. $\operatorname{Rank} M_{a,b}$ is exactly the number of intervals in $\mathcal{B}_{M^L}$ containing both a and b. Conversely, by Proposition 13.4, the collection of ranks $\{\operatorname{Rank} M_{a,b}\}_{a \leq b \in L}$ determines $\mathcal{B}_{M^L}$. □

Though the rank invariant and fibered barcode carry the same information, working with the fibered barcode has a number of advantages.

1. The fibered barcode is more amenable to visualization. RIVET introduces an interactive paradigm for this.
2. The fibered barcode has good stability properties that would be difficult to state directly in terms of rank invariants.

Stability of the Fibered Barcode The fibered barcode exhibits two stability properties, which I will call *external* and *internal* stability. Both results are given explicitly in a recent paper by Claudia Landi, "Rank Invariant Stability via Interleavings," though they were implicit (and given explicitly in a different form) in two earlier papers by Cerri and his coauthors.

Proposition 13.8 (External Stability)**.** *For any p.f.d. persistence modules M and N and L a line of finite, positive slope*

$$d_B(\mathcal{B}_{M^L}, \mathcal{B}_{N^L}) \leq c\, d_I(M, N),$$

where c is a constant depending only on the slope of L. ($c = 1$ when L has slope 1, and tends monotonically to ∞ as the slope of L tends either to ∞ or to 0.)

This result suggests that the barcodes of persistence modules along diagonal lines may have a natural role to play in the multi-parameter approach to TDA.

We state the internal result only informally, referring to Landi for the details:

Proposition 13.9 (Internal Stability)**.** *For any p.f.d. persistence modules M the map $L \to \mathcal{B}_{M^L}$ is continuous at all lines of finite, positive slope.*

13.3 Minimal Resolutions and Bigraded Betti Numbers

Another class of very natural invariant of an n-parameter persistence module is the *multi-Betti numbers*. There are several natural ways to define these. Arguably, the most intuitive definition is one given using minimal resolutions, which we now introduce. As we will explain, minimal resolutions (and their variant, minimal presentations) are important in their own right in TDA. For in depth reading on minimal resolutions, Peeva's book "Graded Syzygies" is a good resource.

Some of the text below is adapted from my 2019 paper with Matthew Wright on computing minimal presentations.

It will be convenient here to work with $\mathbb{Z}^n$-indexed modules, though the definitions do also extend to finitely presented $\mathbb{R}^n$-indexed modules.

Definition 13.10. An exact sequence of free persistence modules

$$F := \quad \cdots \xrightarrow{\partial_3} F_2 \xrightarrow{\partial_2} F_1 \xrightarrow{\partial_1} F_0$$

is called a *resolution* of M if $\operatorname{coker}(\partial_1) \cong M$.

For a persistence module N, let $N^\circ \subset N$ denote the submodule generated by the images of all linear maps $N_{a,b}$ with $a < b \in \mathbb{Z}^d$. (This notation was first introduced in Section 5.6.)

Definition 13.11. We say the resolution F is *minimal* if $\operatorname{im} \partial_i \subset F_{i-1}^\circ$ for each i.

The following result justifies the terminology *minimal* in the above definition.

Proposition 13.12 (Peeva Chapter 1, Theorem 7.3)**.** *A resolution F is minimal if and only if for each $i \geq 0$, any basis for F_i descends to a minimal set of generators for* $\operatorname{coker} \partial_{i+1}$.

Proof. This is proof is essentially the same as the one given by Peeva. First suppose that F is minimal, and let B be a basis for F_i. For $b \in B$, let $\bar{b}$ denote its image in the quotient $\operatorname{coker} \partial_{i+1} = F_i / \operatorname{im} \partial_{i+1}$. To arrive at a contradiction, assume that B descends to a non-minimal set of generators for $\operatorname{coker} \partial_{i+1}$. Then there exists $b_1, \dots b_k \in B$ such that $\bar{b}_1 = \sum_{i=2}^k c_i (\operatorname{coker} \partial_{i+1})_{\operatorname{gr} b_i, \operatorname{gr} b_1} \bar{b}_k$ for some $c_i \in K$. We then have that

$$x = b_1 - \sum_{i=2}^{k} c_i F_{\operatorname{gr} b_i, \operatorname{gr} b_1} b_k \in \operatorname{im} \partial_{i+1}.$$

We claim that $x \notin F_i^\circ$: If $x \in F_i^\circ$, then x can be written as a linear combination of basis elements not involving b_1. This together with the above expression for y gives us an expression for b_1 as a linear combination of other basis elements, which contradicts the minimality of B. But since $x \notin F_i^\circ$, we have $\operatorname{im} \partial_{i+1} \notin F_i^\circ$, contradicting the minimality of F. It follows that F_i descends to a minimal set of generators for $\operatorname{coker} \partial_{i+1}$.

To prove the converse, assume that a basis B for F_i descends to a minimal set of generators for $\operatorname{coker} \partial_{i+1}$. To arrive at a contradiction, assume that $\operatorname{im} \partial_{i+1} \not\subset F_i^\circ$. Then $\operatorname{im} \partial_{i+1}$ contains an element of the form

$$b_1 + \sum_{i=2}^{k} c_i F_{\operatorname{gr} b_i, \operatorname{gr} b_1} b_k \in \operatorname{im} \partial_{i+1},$$

where each $b_i \in B$. Thus b_1 descends to linear combination of other basis elements in $\operatorname{coker} \partial_{i+1}$, contradicting that B descends to a minimal set of generators for $\operatorname{coker} \partial_{i+1}$. □

Corollary 13.13. *A minimal resolution of a finitely generated d-parameter persistence module always exists.*

Proof. We construct the resolution inductively in the usual way, taking the chosen generators to be minimal at each step: First, choose a minimal set of generators G_0 for M, and let F_0 be the free module with basis G_0. We then have an obvious map $f_0 : F_0 \to M$ sending $G_0 \subset F_0$ to $G_0 \subset M$. We have seen that because $K[x_1, \ldots, x_n]$ is Noetherian, the kernel of a map of finitely generated persistence modules is finitely generated, so $\ker f_0$ is finitely generated.

Next, choose a minimal set of generators G_1 for $\ker f_0$ and take F_1 to be the free module with basis G_1. We have a canonical map $f_1 : F_1 \to \ker f_0$. Let ∂_1 denote the composition of this map with the inclusion $\ker f \hookrightarrow f_1$.

Continuing in the way, we construct a resolution for M. It follows from the first isomorphism theorem that for each i, a basis for F_i descends to a minimal set of generators for $\operatorname{coker} \partial_{i+1}$. □

Remark 13.14. The finite generation hypothesis is necessary; consider the $\mathbb{Z}$-indexed module with a copy of the field K at each index, and identity maps everywhere. This has no minimal generating set, hence no minimal resolution.

The key result about minimal resolutions is the following:

Theorem 13.15. *Let F be a minimal resolution for a finitely generated d-parameter persistence module M. Any resolution for M is isomorphic to one obtained from F by summing with resolutions of the form*

$$\cdots 0 \to 0 \to G \xrightarrow{\mathrm{id}_G} G \to 0 \to 0 \to \cdots \to 0$$

where G is a free module, and the two copies of G are allowed to appear at any two consecutive indices.

For the proof of Theorem 13.15, see Peeva's book.

Corollary 13.16. *The minimal resolution of a finitely generated bipersistence module is unique up to isomorphism.*

Proof. This follows from Theorem 13.15 and the Krull-Schmidt theorem (Theorem 10.3). □

As a consequence of the uniqueness of minimal resolutions, the following definition is well posed.

Definition 13.17 (Betti Numbers)**.** Let F be a minimal resolution of a finitely generated n-parameter persistence module M. For $i \geq 0$, define the function $\beta_i^M : \mathbb{Z}^d \to \mathbb{N}$ by

$$\beta_i^M := \mathrm{hf}(F_i / F_i^\circ).$$

For $z \in \mathbb{Z}^d$, we call $\beta_i^M(z)$ the i^{th} *(multi-graded) Betti number of M at grade z.*

Remark 13.18. In view of Lemma 5.28, $\beta_i^M(z)$ is the number of elements at grade z in a basis for F_i.

Remark 13.19. An alternative definition of the bigraded Betti numbers is given as follows: Let $\mathbf{K}$ be the d-parameter persistence module with the vector space K at index 0 and the trivial vector space everywhere else. The functor Tor is well defined for persistence modules; we may define

$$\beta_i^M = \mathrm{hf}(\mathrm{Tor}_i(M, \mathbf{K})).$$

It is not too difficult to check that this coincides with the definition of $\beta_i^M : \mathbb{Z}^d \to \mathbb{R}$ given above.

Remark 13.20. These Betti numbers are important in TDA for several reasons, First, in my experience, they are very helpful for visualization and exploratory analysis of 2-parameter persistence modules. Second, they are used by RIVET's computational framework for interactive visualization of fibered barcodes. One can also imagine that the Betti numbers could be useful statistics for machine learning or statistical analysis of data, but I have not seen any serious application in this direction yet.

Theorem 13.21 (Hilbert's syzygy theorem)**.** *Let F be a minimal resolution of a finitely generated n-parameter persistence module M. We then have that $F^i = 0$ for $i > n$.*

Sketch of Proof. To prove this, it is convenient to use the Tor functor definition of β_i^M. $\mathrm{Tor}_i(A, B)$ is the module defined by $H_i(F \otimes B)$, where F is any resolution of A. (The tensor product of d-graded modules is defined by taking the usual module-theoretic tensor product; the d-grading is given by taking the vector space at grade z to be $\bigoplus_{z_1+z_2=z} A_{z_1} \otimes B_{z_2}$.) Tor_i is symmetric, i.e., $\mathrm{Tor}_i(A, B) = \mathrm{Tor}_i(B, A)$ which means we can in fact take a resolution of either A or B. There exists a minimal resolution of $\mathbf{K}$ called the Koszul complex, which has length n. This implies that $\mathrm{Tor}_i(M, \mathbf{K}) = 0$ for $i \geq n$. The result follows. □

Thus, β_i^M is only of interest for $i \leq n$.

The following formula relating the Hilbert function to the bigraded Betti numbers follows from Hilbert's Syzygy theorem by an easy inductive argument; see for example Theorem 16.2 in Peeva's text for a proof of the analogous result in the case of $\mathbb{Z}$-graded $K[t_1, \ldots, t_d]$-modules.

Proposition 13.22. *For M a finitely generated n-parameter persistence module and $z \in \mathbb{Z}$,*

$$\dim M_z = \sum_{i=0}^{n} (-1)^i \sum_{y \leq z} \beta_i^M(y).$$

In my experience, the reason this formula is useful in practice in TDA computations is that if you can compute the Hilbert function of M and β_i^M for $i = 0, \ldots, n-1$, then the formula yields β_n^M quite readily.

14 Computing Minimal Presentations of Bipersistence Modules

In this section, we address the fundamental problem of computing a minimal presentation of a bipersistence module, given as input a bifiltration. This section follows a recent paper by Matthew Wright and me. The d-parameter version of this problem is a classical problem in commutative algebra; it is typically solved using Gröbner basis techniques. However, it turns out that the 2-parameter version of the problem allows for a streamlined approach and exposition. The key algebraic subproblem is the computation of the kernel of a morphism of free bipersistence modules.

The algorithm described here has been implemented in RIVET since 2018, and works well in practice.

14.1 Minimal Presentations

Recall that a *presentation* of a d-parameter persistence module is a morphism $\partial : F^1 \to F^0$ of free persistence modules with $\operatorname{coker}(\partial) \cong M$. Thus, a presentation for M is simply the data of the last morphism in a free resolution for M.

The algorithm we give also computes the Hilbert function and bigraded Betti numbers of M as a side product.

Definition 14.1. A presentation is said to be *minimal* if extends to a minimal resolution.

It follows from the theory of minimal presentations that a minimal presentation of a module is unique up to isomorphism and any minimal presentation can be obtained (up to isomorphism) by summing with maps of the form

$$G \xrightarrow{\mathrm{Id}_G} G \qquad \text{or} \qquad G \to 0,$$

where G is free.

Matrix Representation of Minimal Presentations We have seen that we can represent a morphism of free persistence modules by a matrix with row and column labels. The uniqueness of minimal presentations tells us that the dimensions of this matrix are uniquely determined, as are the row and column labels (up to permutation). However, the matrix itself is not unique, and this means that minimal presentations are unlikely to be useful in TDA in the way that barcodes are in the 1-parameter setting, e.g., as input to machine learning algorithms or statistical tests. Nevertheless, minimal presentations are useful computational intermediates; they encode the full isomorphism structure of a d-parameter persistence module in an efficient way, and can serve as input to algorithms to compute invariants or metrics.

14.2 FI-Reps: Matrix Representations of Short Chain Complexes

The input to our algorithm for computing minimal presentation is (basically) a short (three-term) chain complex of free bipersistence modules, represented in matrix form. We call this input an *FI-Rep*. Such a chain complex has a unique homology module M, and the output of the algorithm is a presentation matrix for M.

Let us explain this in more detail. Here and throughout, we work with $\mathbb{N}^2$-indexed bifiltrations and bipersistence modules, though our algorithm works just as well for $\mathbb{R}^2$-indexed objects.

For M a bipersistence module, let

$$C \xrightarrow{f} D \xrightarrow{g} E.$$

be a chain complex of free bipersistence modules with $\ker g / \operatorname{im} f \cong M$. Choosing ordered bases for C, D, and E, we can represent this chain complex by matrices

$$[f] \quad \text{and} \quad [g],$$

with each row and each column labeled by an element of $\mathbb{N}^2$. In fact, to encode M up to isomorphism, suffices to keep only the column labels: The row labels for $[g]$ turn out to be unnecessary, and the row labels for $[f]$ are same as column labels for $[g]$.

We call the pair of column-labeled matrices $([f], [g])$ an FI-Rep for M; this stands for *free implicit representation*. In practice, we store the matrices $[f]$ and $[g]$ in a column-sparse format, as we did for ordinary persistence computation.

How FI-Reps Arise from Data In Section 7.2, we explained that for any $\mathbb{N}$-indexed filtration F, a straightforward construction gives a chain complex $C(F)$ of free bipersistence modules whose i^{th} homology group is $H_i(F)$. The same is true in the 2-parameter setting, provided our bifiltration F is 1-critical. For a multi-critical bifiltration F, we can still construct the chain complex $C(F)$, but its modules needn't be free. Instead, $C_i(F)$ will be a direct sum $C^j(F) \cong \oplus_{\sigma \text{ a } j\text{-simplex of } \operatorname{colim}(F)} S^\sigma$, where

$$S_z^\sigma = \begin{cases} K & \text{if } \sigma \in F_z, \\ 0 & \text{otherwise.} \end{cases} \qquad S_{y,z}^\sigma = \begin{cases} \mathrm{Id}_K & \text{if } \sigma \in F_y, \\ 0 & \text{otherwise.} \end{cases}$$

For example, if a simplex $\sigma \in \mathrm{colim}(F)$ is born at indices $(3,0), (2,1)$, and $(0,2)$, then S^σ looks like this:

$$\begin{array}{ccccccccc}
\vdots & & \vdots & & \vdots & & \vdots & & \\
\uparrow & & \uparrow & & \uparrow & & \uparrow & & \\
K & \longrightarrow & K & \longrightarrow & K & \longrightarrow & K & \longrightarrow & \cdots \\
\uparrow & & \uparrow & & \uparrow & & \uparrow & & \\
K & \longrightarrow & K & \longrightarrow & K & \longrightarrow & K & \longrightarrow & \cdots \\
\uparrow & & \uparrow & & \uparrow & & \uparrow & & \\
0 & \longrightarrow & 0 & \longrightarrow & K & \longrightarrow & K & \longrightarrow & \cdots \\
\uparrow & & \uparrow & & \uparrow & & \uparrow & & \\
0 & \longrightarrow & 0 & \longrightarrow & 0 & \longrightarrow & K & \longrightarrow & \cdots
\end{array}$$

It is an observation of Chacholski, Scolamiero, and Vaccarino that if the bifiltration F is multicritical, a simple construction converts the short chain chain complex

$$C(F) = C^{j+1} \xrightarrow{\partial^{j+1}} C^j(F) \xrightarrow{\partial^j} C^{j-1}(F)$$

into an FI-Rep for $H_i(F)$. We will not go into the details of this, but they are not difficult.

14.3 Computation of a Semi-Minimal Presentation

Suppose we are given an FI rep $([f], [g])$ for M as above. To compute a miminal presentation for M, we first compute a presentation that has the following partial minimality property:

Definition 14.2. We say a presentation $f : F \to F'$ is *semi-minimal* if each non-minimal summand of the presentation is of the form

$$G \xrightarrow{\mathrm{Id}_G} G.$$

To explain how we compute a minimal presentation from data, we need the following fact:

Proposition 14.3. *If $\gamma : F \to F'$ is a morphism of free finitely presented 2-parameter persistence modules, then γ is free.*

Proof. This follows from Hilbert's Syzygy theorem, together with the structure theorem for minimal presentations. We leave the details as an exercise. □

In outline, our algorithm for computing a semi-minimal presentation proceeds in three steps:

1. Using $[f]$ as input, find a minimal ordered set of generators S for $\operatorname{im} f$.
2. Compute a basis $B_{\ker}$ for $\ker g$.
3. Express each element of S in $B_{\ker}$-coordinates; put resulting column vectors into a matrix P, with column labels the bigrades of S and row labels the bigrades of $B_{\ker}$.

Each of these steps requires requires further explanation.

- Let us note that the columns $[f]$ already specify an ordered set of generators for $\operatorname{im} f$, much in the same way that in ordinary linear algebra, the columns of a matrix representing a linear map T represent vectors spanning $\operatorname{im} T$. However, that set of generators may not be minimal to start.
- Note that in view of Proposition 14.3, $\ker g$ is free, so step 2 makes sense.
- We remark that the algorithms for steps 1 and 2 are very similar: Both steps can be carried out using a bigraded variant of the standard reduction of Section 6.2 that we call the *bigraded reduction*. We give the details below, focusing on the case of kernel computation.
- Step 3 is just ordinary linear algebra: For each element of S, we solve a linear system. It is straightforward to carry this out efficiently in the column-sparse setting, using a version of the standard reduction.
- Even if step 1 is omitted, this approach still yields a presentation, but it may not be semi-minimal.

14.4 Kernel Computation in the 1-Parameter Case

To prepare for a discussion of kernel computation in the 2-parameter case, we consider the same problem in the 1-parameter case.

$f : M \to N$ be a map of free persistence modules, and let B, B' be ordered bases for M and N, with B in order of increasing grade.

The following slight extension of the standard reduction computes $\ker f$:
Input: The column-labeled matrix $R := [f]$ representing f with respect to B and B'
Output: Basis $B_{\ker f}$ for $\ker f$ (represented as column vectors with respect to the basis B):

Algorithm:

1. Run the standard reduction on R with a "slave matrix" V, initially the identity.

2. For each column j zeroed out in R, add the vector in $M_{\text{label}(j)}$ represented by the j^{th} column of V to $B_{\ker f}$.

Example 14.4. Let

$$M = Q^1 \oplus Q^1 \oplus Q^3, \qquad N = Q^0 \oplus Q^0 \oplus Q^0,$$

and let $f : M \to N$ be given with respect to the standard bases by the matrix

$$\begin{array}{c} \\ 0 \\ 0 \\ 0 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 1 & 3 \end{array} \\ \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix} \end{array}$$

The computation of $\ker f$ proceeds as follows.

$$\begin{array}{c} \begin{array}{ccc} 1 & 1 & 3 \end{array} \\ \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix} \\ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{array} \xrightarrow{\text{Add col. 1 to col. 2}} \begin{array}{c} \begin{array}{ccc} 1 & 1 & 3 \end{array} \\ \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{array} \xrightarrow{\text{Add col. 2 to col. 3}} \begin{array}{c} \begin{array}{ccc} 1 & 1 & 3 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix} \\ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \end{array}$$

Thus, we find that

$$B_{\ker f} = \left\{ \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \right\} \in M_3.$$

14.5 Kernel Computation in the 2-Parameter Case

The algorithm for the 1-parameter case turns out to extend without undue pain to the 2-parameter case. The algorithm simultaneously makes essential use of three orders on $\mathbb{N}^2$: The lexicographical order, the colexicographical order, and the usual partial order.To explain the details, let $f : M \to N$ be a map of free bipersistence modules, and let B and B' be ordered bases for M and N, with B in *colexicographical order* on the bigrades. Here is the algorithm:

Input: Column-labeled Matrix $R := [f]$ representing f with respect to the bases B and B'.
Output: Basis for $\ker f$ (represented with respect to the basis B):

Let R^z denote the submatrix of R consisting of columns with label $\leq z$. (Here $\leq$ denotes the *product partial order* on $\mathbb{N}^2$.)
Algorithm:

1. Initialize a slave matrix V to be the identity matrix of dimensions the number of columns of R.

2. For each z in *lexicographical order*:

   - Run persistence algorithm on R^z,
   - Also perform each column operation on V,
   - If column j gets zeroed out, add the vector in M_z represented by the j^{th} column of V to the basis for $\ker f$.

Remark 14.5. In the algorithm above, we never reset or copy R or V, we just perform operations on the single pair of matrices throughout.

Remark 14.6. We can carry out step 1 of the algorithm for computing a semi-minimal presentation using essentially the same algorithm, but without a slave matrix; columns of R with label z that do not get reduced to 0 at index z are added to S.

Remark 14.7. To efficiently implement the above algorithm for computing a kernel, one needs to work with pivot arrays, as in the standard reduction. In fact it is sufficient to maintain a single pivot array for the entire computation. The details of this, which are perhaps not entirely obvious, are given in my paper with Matthew.

Exercise 14.8. *Let*

$$M = Q^{(0,0)} \oplus Q^{(1,0)} \oplus Q^{(0,1)} \oplus Q^{(1,1)}, \qquad N = Q^{(0,0)} \oplus Q^{(0,0)} \oplus Q^{(0,0)},$$

and let $f : M \to N$ be given with respect to the standard bases by the matrix

$$\begin{array}{c} \\ (0,0) \\ (0,0) \\ (0,0) \end{array} \begin{array}{c} \begin{array}{cccc} (0,0) & (1,0) & (0,1) & (1,0) \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \end{array}$$

By running the algorithm for kernel computation described above, show that

$$\mathcal{B}_{\ker f} = \left\{ \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} \in M_{(1,1)} \right\}$$

is a basis for $\ker f$.

In brief, the reason the algorithm works is the following: We are implicitly computing compatible bases for $\ker f_z : M_z \to N_z$ for all z. Just after reducing R^z, the only columns ever added to columns of R^z are also in R^z. Thus computations at earlier indices do not cause problems with the computation at index z. For the full correctness proof, I refer the reader to the paper.

14.6 Minimizing a presentation

Algorithm:
For each column i:

1. Check if the label of the column i is equal to the row-label of the pivot.
2. If so,
   - zero out the row of the pivot by adding column i to columns to the right.
   - Remove both the column i and the row of the pivot.

This is a bigraded variant of the standard procedure in commutative algebra for minimizing a resolution. The algorithm is embarrassingly parallel, and in fact RIVET has a parallel implementation. The proof of correctness is left as an exercise.

14.7 Complexity Bounds

If the $([f], [g])$ is an FI-Rep of a bipersistence module M, where the maximum dimension of either matrix is n, then over a finite field, our algorithm for computing a minimal presentation for M runs in $O(n^3)$ ttme and $O(n^2)$ memory.

15 Epilogoue

In teaching this class, I've come to appreciate more fully that multi-parameter persistence is a *broad* subject. To fully understand the challenge of realizing 2-parameter persistence as a practical tool for applications, one needs to explore the subject from a number of angles, including algebraic, algorithmic, and statistical ones, and one needs to think carefully about what is required of specific applications. In this course, we've covered many of these fundamentals, though there are also some important topics we did not have time for. In particular, we did not:

- cover the statistical aspects of the subject;
- explore practical applications of 2-parameter persistent homology in any detail (applications were discussed only briefly in class, and not at all in the notes);
- discuss sheaf-theoretic viewpoints on multi-parameter persistence, which are interesting and important;
- discuss persistence perspectives on clustering;
- prove the uniqueness statement in the Krull-Schmidt-Azumaya or the structure theorem for minimal resolutions.

Thus, as the semester draws to a close, I feel that the story we've told still remains somewhat incomplete. But perhaps that is fitting for a subject that is still in its adolescence, especially with regard to applications—in spite of encouraging recent progress on a number of fronts, the story of multi-parameter persistence is itself arguably still incomplete, and a part of the TDA community is hard at work to flesh out the narrative. In any case, I hope that this course has given you some appreciation for the subject, and some sense of what the challenges are for the future. I've quite enjoyed teaching the course.

List of Exercises