Algebraic K-Theory of Persistence Modules

Ryan Grady and **Anna Schenfisch***

February 2, 2024

algo

Algorithms TU Eindhoven

Outline

1. Persistence modules
2. Algebraic K-theory of persistence modules

Outline

Persistence Modules

A **persistence module** is a functor

$$I \to A,$$

for a poset category I and abelian category A.

Persistence Modules

A **persistence module** is a functor

$$I \to A,$$

for a poset category I and abelian category A.

- A is often $\mathrm{Vect}_{\mathbb{F}}$ or $\mathrm{grVect}_{\mathbb{F}}$.

Persistence Modules

A **persistence module** is a functor

$$I \to A,$$

for a poset category I and abelian category A.

- A is often $\mathrm{Vect}_{\mathbb{F}}$ or $\mathrm{grVect}_{\mathbb{F}}$.
- Different types of posets I correspond to specific types of persistence modules.

Persistence Modules

A **persistence module** is a functor

$$I \to A,$$

for a poset category I and abelian category A.

- A is often $\mathrm{Vect}_{\mathbb{F}}$ or $\mathrm{grVect}_{\mathbb{F}}$.
- Different types of posets I correspond to specific types of persistence modules.

One-parameter monotone

Persistence Modules

A **persistence module** is a functor

$$I \to A,$$

for a poset category I and abelian category A.

- A is often $\mathrm{Vect}_{\mathbb{F}}$ or $\mathrm{grVect}_{\mathbb{F}}$.
- Different types of posets I correspond to specific types of persistence modules.

One-parameter monotone

One-parameter zig-zag

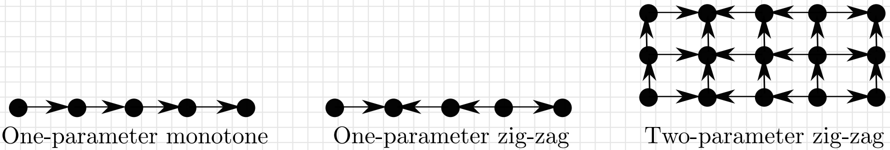
Persistence Modules

A **persistence module** is a functor

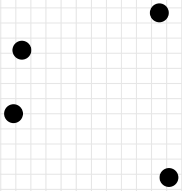
$$I \to A,$$

for a poset category I and abelian category A.

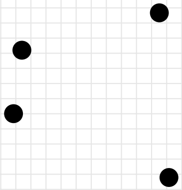
- A is often $\mathrm{Vect}_{\mathbb{F}}$ or $\mathrm{grVect}_{\mathbb{F}}$.
- Different types of posets I correspond to specific types of persistence modules.



Modules From Filtrations

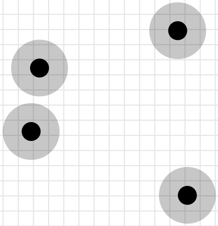


Modules From Filtrations



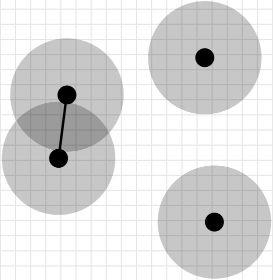
$$\text{radius} = 0$$

Modules From Filtrations



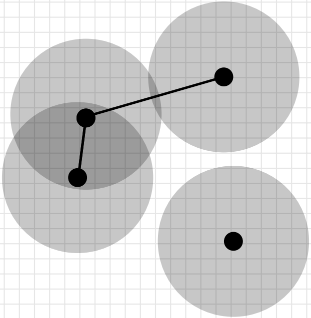
$$\text{radius} = 10$$

Modules From Filtrations



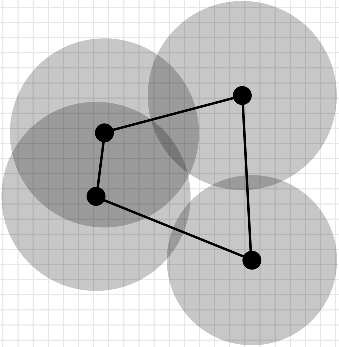
$$\text{radius} = 14$$

Modules From Filtrations



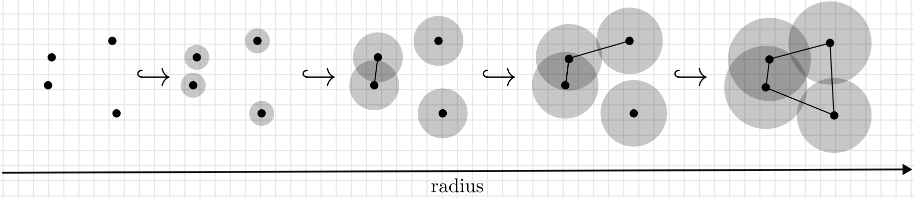
$$\text{radius} = 25$$

Modules From Filtrations

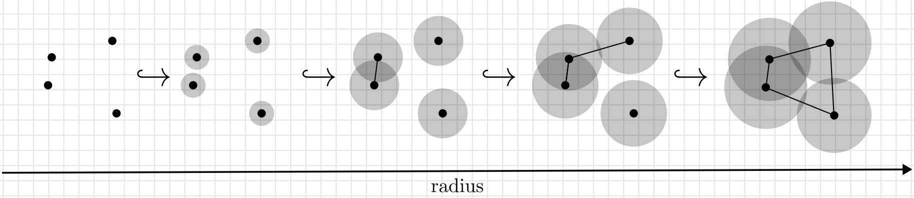


$$\text{radius} = 37$$

Modules From Filtrations

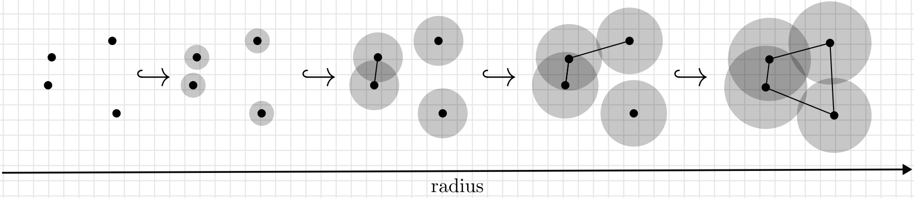


Modules From Filtrations



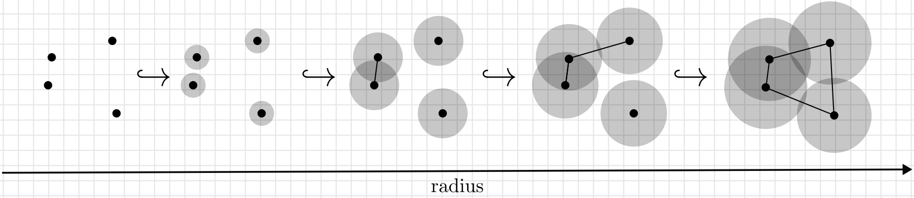
- Suppose we keep track of $\mathbb{H}_0$ at each step in the filtration. This defines a functor $\mathbb{P} : [n] \to \mathsf{Vect}$ (a persistence module!).

Modules From Filtrations



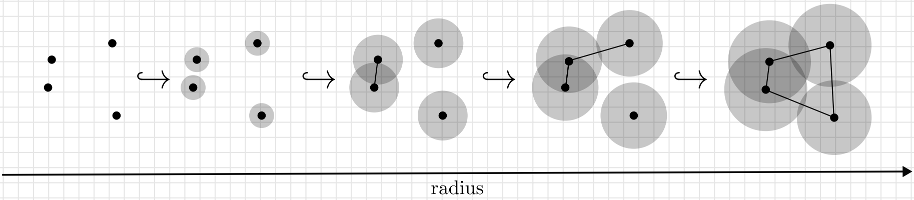
- Suppose we keep track of $\mathbb{H}_0$ at each step in the filtration. This defines a functor $\mathbb{P} : [n] \to \text{Vect}$ (a persistence module!).
- Maps $i \mapsto i+1$ correspond to inclusion maps in the filtration, which induce maps in $\mathbb{H}_0$.

Modules From Filtrations



- Suppose we keep track of $\mathbb{H}_0$ at each step in the filtration. This defines a functor $\mathbb{P} : [n] \to \text{Vect}$ (a persistence module!).
- Maps $i \mapsto i + 1$ correspond to inclusion maps in the filtration, which induce maps in $\mathbb{H}_0$.
- Notice there's a related functor $\mathbb{P} : \mathbb{R} \to \text{Vect}$.

Modules From Filtrations



- Suppose we keep track of $\mathbb{H}_0$ at each step in the filtration. This defines a functor $\mathbb{P} : [n] \to \text{Vect}$ (a persistence module!).
- Maps $i \mapsto i+1$ correspond to inclusion maps in the filtration, which induce maps in $\mathbb{H}_0$.
- Notice there's a related functor $\mathbb{P} : \mathbb{R} \to \text{Vect}$.
- Using additional filter parameters is the usual way to get higher-dimensional persistence modules.

Zig-Zag Grid Modules

- We focus on **d-parameter zig-zag grid persistence modules**, where $I = I_1 \times I_2 \times \ldots \times I_d$, where each I_n is nontrivial and finite. We consider the objects as subsets of $\mathbb{R}$ (but possibly with different ordering).

Zig-Zag Grid Modules

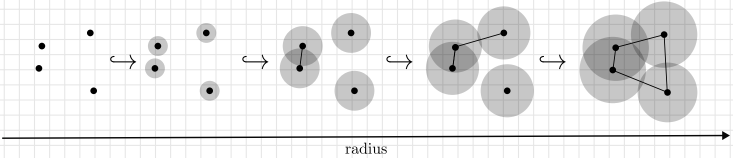
- We focus on **d-parameter zig-zag grid persistence modules**, where $I = I_1 \times I_2 \times \ldots \times I_d$, where each I_n is nontrivial and finite. We consider the objects as subsets of $\mathbb{R}$ (but possibly with different ordering).
- $\mathbb{R}^d$ is our **parameter space**

Zig-Zag Grid Modules

- We focus on **d-parameter zig-zag grid persistence modules**, where $I = I_1 \times I_2 \times \ldots \times I_d$, where each I_n is nontrivial and finite. We consider the objects as subsets of $\mathbb{R}$ (but possibly with different ordering).
- $\mathbb{R}^d$ is our **parameter space**
- Let's consider persistence modules as *stratifying* a parameter space.

Zig-Zag Grid Modules

- We focus on **d-parameter zig-zag grid persistence modules**, where $I = I_1 \times I_2 \times \ldots \times I_d$, where each I_n is nontrivial and finite. We consider the objects as subsets of $\mathbb{R}$ (but possibly with different ordering).
- $\mathbb{R}^d$ is our **parameter space**
- Let's consider persistence modules as *stratifying* a parameter space.
- Example 1:



Zig-Zag Grid Modules

- We focus on **d-parameter zig-zag grid persistence modules**, where $I = I_1 \times I_2 \times \ldots \times I_d$, where each I_n is nontrivial and finite. We consider the objects as subsets of $\mathbb{R}$ (but possibly with different ordering).
- $\mathbb{R}^d$ is our **parameter space**
- Let's consider persistence modules as *stratifying* a parameter space.
- Example 1:

Zig-Zag Grid Modules

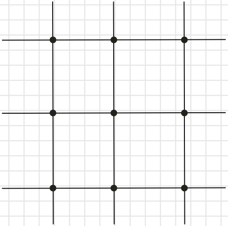
- We focus on **d-parameter zig-zag grid persistence modules**, where $I = I_1 \times I_2 \times \ldots \times I_d$, where each I_n is nontrivial and finite. We consider the objects as subsets of $\mathbb{R}$ (but possibly with different ordering).
- $\mathbb{R}^d$ is our **parameter space**
- Let's consider persistence modules as *stratifying* a parameter space.
- Example 1:

- Example 2:

Zig-Zag Grid Modules

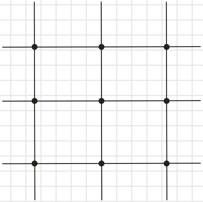
- We focus on **d-parameter zig-zag grid persistence modules**, where $I = I_1 \times I_2 \times \ldots \times I_d$, where each I_n is nontrivial and finite. We consider the objects as subsets of $\mathbb{R}$ (but possibly with different ordering).
- $\mathbb{R}^d$ is our **parameter space**
- Let's consider persistence modules as *stratifying* a parameter space.
- Example 1:

- Example 2:



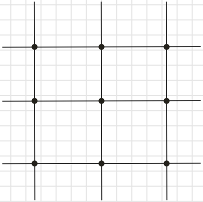
Combinatorial Entrance Path Category

I cubulates $\mathbb{R}^d$, and we stratify $\mathbb{R}^d$ by declaring each i cube of the cubulation to be an i-strata. Call this stratified space $(\mathbb{R}^d; I)$ (a **cubical manifold**).



Combinatorial Entrance Path Category

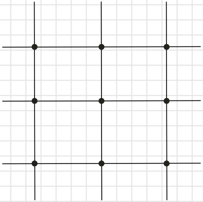
I cubulates $\mathbb{R}^d$, and we stratify $\mathbb{R}^d$ by declaring each i cube of the cubulation to be an i-strata. Call this stratified space $(\mathbb{R}^d; I)$ (a **cubical manifold**).



A category naturally associated to this stratification is $\mathrm{Ent}_{\Delta}(\mathbb{R}^d; I)$

Combinatorial Entrance Path Category

I cubulates $\mathbb{R}^d$, and we stratify $\mathbb{R}^d$ by declaring each i cube of the cubulation to be an i-strata. Call this stratified space $(\mathbb{R}^d; I)$ (a **cubical manifold**).

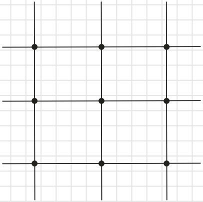


A category naturally associated to this stratification is $\mathrm{Ent}_{\Delta}(\mathbb{R}^d; I)$

- Objects: strata of $(\mathbb{R}^d; I)$

Combinatorial Entrance Path Category

I cubulates $\mathbb{R}^d$, and we stratify $\mathbb{R}^d$ by declaring each i cube of the cubulation to be an i-strata. Call this stratified space $(\mathbb{R}^d; I)$ (a **cubical manifold**).

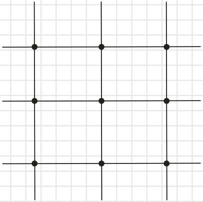


A category naturally associated to this stratification is $\mathrm{Ent}_\Delta(\mathbb{R}^d; I)$

- Objects: strata of $(\mathbb{R}^d; I)$
- Morphisms: $\mathfrak{c} \to \mathfrak{c}'$ iff $\mathfrak{c}$ is a coface of $\mathfrak{c}'$.

Combinatorial Entrance Path Category

I cubulates $\mathbb{R}^d$, and we stratify $\mathbb{R}^d$ by declaring each i cube of the cubulation to be an i-strata. Call this stratified space $(\mathbb{R}^d; I)$ (a **cubical manifold**).



A category naturally associated to this stratification is $\mathrm{Ent}_\Delta(\mathbb{R}^d; I)$

- Objects: strata of $(\mathbb{R}^d; I)$
- Morphisms: $\mathfrak{c} \to \mathfrak{c}'$ iff $\mathfrak{c}$ is a coface of $\mathfrak{c}'$.

Use the notation $\mathrm{pMod}^A(X) := \mathrm{Fun}(\mathrm{Ent}_\Delta(\mathrm{X}; \mathrm{I}), \mathrm{A})$.

$\mathrm{pMod}^A(X) := \mathrm{Fun}(\mathrm{Ent}_\Delta(\mathrm{X};\mathrm{I}), \mathrm{A})$

Things to notice

$\mathrm{pMod}^A(X) := \mathrm{Fun}(\mathrm{Ent}_\Delta(\mathrm{X};\mathrm{I}), \mathrm{A})$

Things to notice

- $\mathrm{Ent}_\Delta(X; I)$ has many more objects and morphisms than our original poset I.

$\mathrm{pMod}^A(X) := \mathrm{Fun}(\mathrm{Ent}_\Delta(\mathrm{X};\mathrm{I}), \mathrm{A})$

Things to notice

- $\mathrm{Ent}_\Delta(X; I)$ has many more objects and morphisms than our original poset I.
  - $\mathrm{Fun}(\mathrm{Ent}_\Delta(\mathrm{X};\mathrm{I}), \mathrm{A})$ generalises $\mathrm{Fun}(\mathrm{I}, \mathrm{A})$.

$\mathrm{pMod}^A(X) := \mathrm{Fun}(\mathrm{Ent}_\Delta(\mathrm{X}; \mathrm{I}), \mathrm{A})$

Things to notice

- $\mathrm{Ent}_\Delta(X; I)$ has many more objects and morphisms than our original poset I.
  - $\mathrm{Fun}(\mathrm{Ent}_\Delta(\mathrm{X}; \mathrm{I}), \mathrm{A})$ generalises $\mathrm{Fun}(\mathrm{I}, \mathrm{A})$.
- Here, we forget parameter values and only keep track of the poset structure of I.

Outline

Algebraic K-Theory

- Algebraic K-theory is a map from (nice) categories to **spectra** (a sequence of based topological spaces, along with particular maps between them).

Algebraic K-Theory

- Algebraic K-theory is a map from (nice) categories to **spectra** (a sequence of based topological spaces, along with particular maps between them).
- Associated to a K spectrum, $\mathbb{K}(C)$, we also have K-groups, one for each $n \in \mathbb{N}_{\geq 0}$. These groups are the homotopy groups of the spectrum, where

$$K_n(C) \cong \mathrm{colim}_i \pi_{i+n}(\mathbb{K}_i(C))$$

.

Algebraic K-Theory

- Algebraic K-theory is a map from (nice) categories to **spectra** (a sequence of based topological spaces, along with particular maps between them).
- Associated to a K spectrum, $\mathbb{K}(C)$, we also have K-groups, one for each $n \in \mathbb{N}_{\geq 0}$. These groups are the homotopy groups of the spectrum, where

$$K_n(C) \cong \text{colim}_i \pi_{i+n}(\mathbb{K}_i(C))$$

.

- K-groups can provide a rich set of invariants of the input category.

Algebraic K-Theory

- Algebraic K-theory is a map from (nice) categories to **spectra** (a sequence of based topological spaces, along with particular maps between them).
- Associated to a K spectrum, $\mathbb{K}(C)$, we also have K-groups, one for each $n \in \mathbb{N}_{\geq 0}$. These groups are the homotopy groups of the spectrum, where

$$K_n(C) \cong \text{colim}_i \pi_{i+n}(\mathbb{K}_i(C))$$

.

- K-groups can provide a rich set of invariants of the input category.
- E.g., in the case of persistence modules, K_0 is the natural home for invariants like Euler characteristic curves, K_1 contains information about transformations between persistence modules, etc.

Main Tool

pMod is an example of a **Waldhausen category**, which has the following nice result...

Main Tool

pMod is an example of a **Waldhausen category**, which has the following nice result...

Waldhausen Additivity (1985)

Suppose that $A \to B \to C$ is a standard split short exact sequence of Waldhausen categories. Then, there is the following equivalence of spectra:

$$\mathbb{K}(B) \simeq \mathbb{K}(A) \vee \mathbb{K}(C).$$

Main Tool

pMod is an example of a **Waldhausen category**, which has the following nice result...

Waldhausen Additivity (1985)

Suppose that $A \to B \to C$ is a standard split short exact sequence of Waldhausen categories. Then, there is the following equivalence of spectra:

$$\mathbb{K}(B) \simeq \mathbb{K}(A) \vee \mathbb{K}(C).$$

To start, can we build a split short exact sequence of categories of one-parameter persistence modules?

A Split Short Exact Sequence

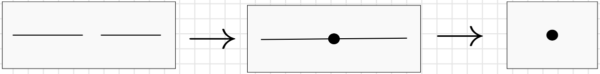
To start, can we build a split short exact sequence of objects in the category of one-parameter persistence modules?

A Split Short Exact Sequence

To start, can we build a split short exact sequence of objects in the category of one-parameter persistence modules?
(let's use sketches of stratified spaces to mean the category of persistence modules over them)

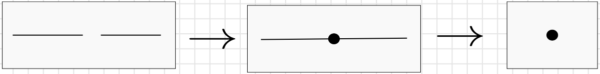
A Split Short Exact Sequence

To start, can we build a split short exact sequence of objects in the category of one-parameter persistence modules?
(let's use sketches of stratified spaces to mean the category of persistence modules over them)



A Split Short Exact Sequence

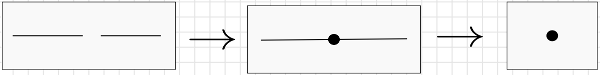
To start, can we build a split short exact sequence of objects in the category of one-parameter persistence modules?
(let's use sketches of stratified spaces to mean the category of persistence modules over them)

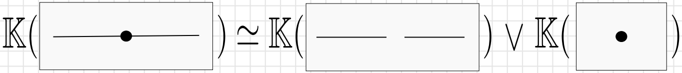


This splits! (and is standard) So ...

A Split Short Exact Sequence

To start, can we build a split short exact sequence of objects in the category of one-parameter persistence modules?
(let's use sketches of stratified spaces to mean the category of persistence modules over them)





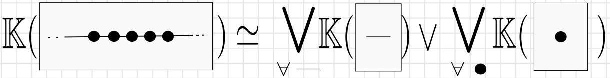
$\mathbb{K}$-theory of Zig-Zag Persistence Modules

To build up the computation for $\mathbb{K}$ of one-parameter persistence modules, we induct on the number of zero-strata.

$\mathbb{K}$-theory of Zig-Zag Persistence Modules

To build up the computation for $\mathbb{K}$ of one-parameter persistence modules, we induct on the number of zero-strata.

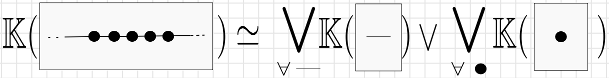
CLAIM:



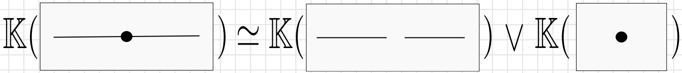
$\mathbb{K}$-theory of Zig-Zag Persistence Modules

To build up the computation for $\mathbb{K}$ of one-parameter persistence modules, we induct on the number of zero-strata.

CLAIM:



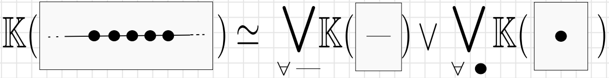
BASE CASE:



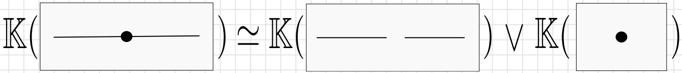
$\mathbb{K}$-theory of Zig-Zag Persistence Modules

To build up the computation for $\mathbb{K}$ of one-parameter persistence modules, we induct on the number of zero-strata.

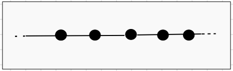
CLAIM:



BASE CASE:



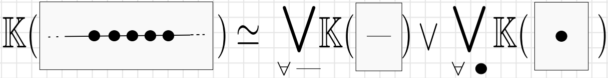
INDUCTION STEP:



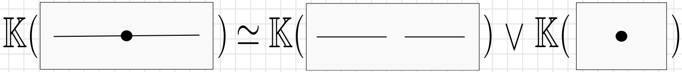
$\mathbb{K}$-theory of Zig-Zag Persistence Modules

To build up the computation for $\mathbb{K}$ of one-parameter persistence modules, we induct on the number of zero-strata.

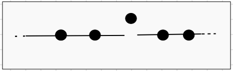
CLAIM:



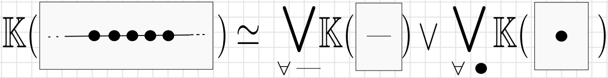
BASE CASE:



INDUCTION STEP:



$\mathbb{K}$-theory of Zig-Zag Persistence Modules



$\mathbb{K}$-theory of Zig-Zag Persistence Modules

K-Theory of One-Parameter Persistence Modules (Grady, S.)

Let X be a cubical one-manifold with finite zero-strata. Then there is an equivalence of spectra

$$\mathbb{K}(\mathrm{pMod}(X)) \simeq \bigvee_{x_1 \in X_1} \mathbb{K}(\mathrm{pMod}(x_1)) \vee \bigvee_{x_0 \in X_0} \mathbb{K}(\mathrm{pMod}(x_0)),$$

where X_i is the set of i-strata of X.

Extension to Multi-Parameter Modules

We use a generalisation of the cutting and pasting from before

Extension to Multi-Parameter Modules

We use a generalisation of the cutting and pasting from before

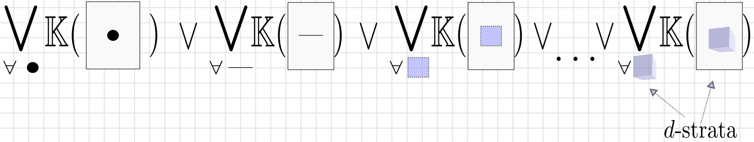
Additivity For Closed Sub-Stratified Spaces (Grady, S.)

Let X be a cubical manifold and let B denote a closed sub-stratified space of X. Then there is an equivalence of spectra

$$\mathbb{K}(\mathrm{pMod}(X)) \simeq \mathbb{K}(\mathrm{pMod}(X \setminus B)) \vee \mathbb{K}(\mathrm{pMod}(B)).$$

Additivity Over Strata

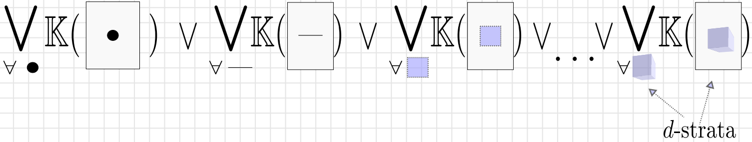
CLAIM: Let X be a cubical manifold that can be tightly embedded as a substratified manifold of some $(\mathbb{R}^d; I)$. The K-theory of persistence modules over X is equivalent with



[1] where d minimal such that X embeds as a stratified space into $(\mathbb{R}^d; I)$

Additivity Over Strata

CLAIM: Let X be a cubical manifold that can be tightly embedded as a substratified manifold of some $(\mathbb{R}^d; I)$. The K-theory of persistence modules over X is equivalent with

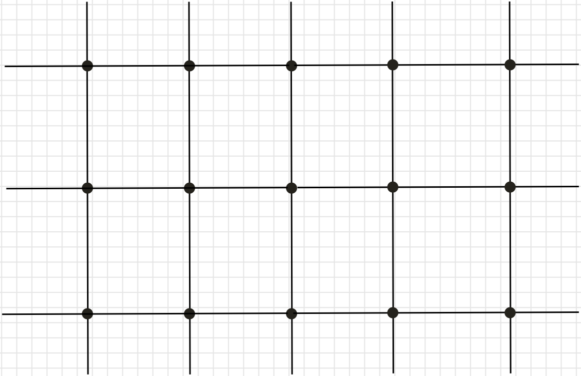


To extend to multi-parameter persistence modules, induct on d, the number of parameters, and on $h = \max_{n=1}^{d}\{|I_n|\}$ the *height* of the module[1].

[1]where d minimal such that X embeds as a stratified space into $(\mathbb{R}^d; I)$

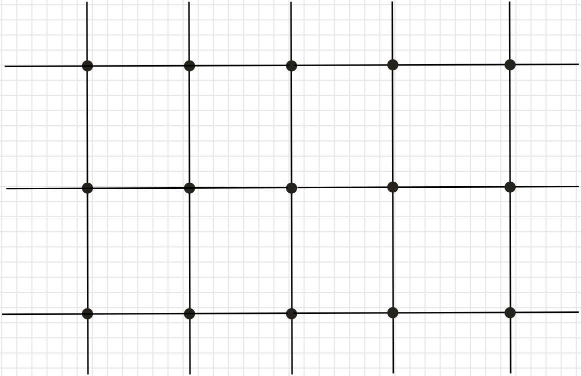
Induction on Height

For the height induction, start with a cubical manifold with height h, then try to break it into pieces with height less than h (and each piece we remove needs to be closed in the space containing it).



Induction on Height

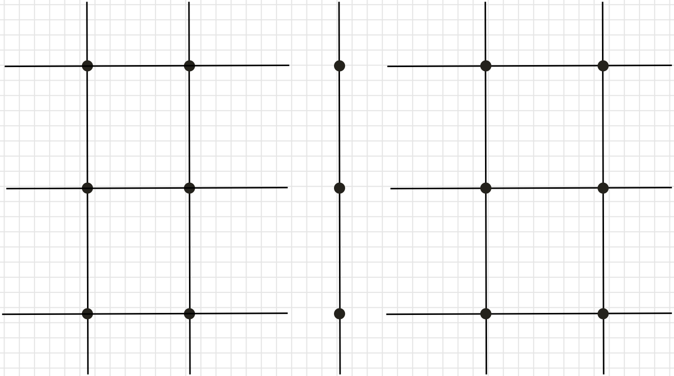
For the height induction, start with a cubical manifold with height h, then try to break it into pieces with height less than h (and each piece we remove needs to be closed in the space containing it).



If a module only has one parameter with maximum height, it's easy...

Induction on Height

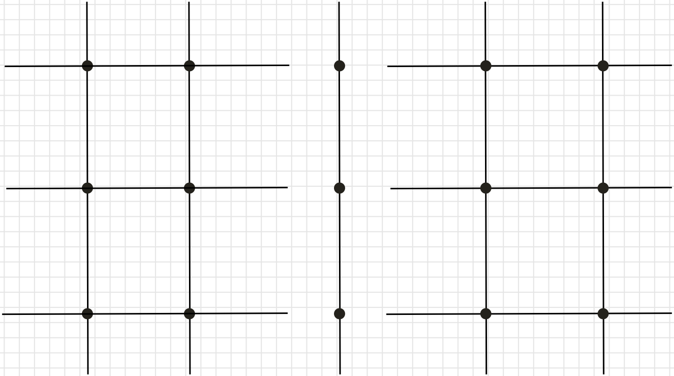
For the height induction, start with a cubical manifold with height h, then try to break it into pieces with height less than h (and each piece we remove needs to be closed in the space containing it).



If a module only has one parameter with maximum height, it's easy...

Induction on Height

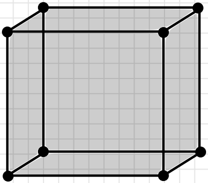
For the height induction, start with a cubical manifold with height h, then try to break it into pieces with height less than h (and each piece we remove needs to be closed in the space containing it).



If a module has multiple parameters with maximum heights, we may need to decompose it into many pieces.

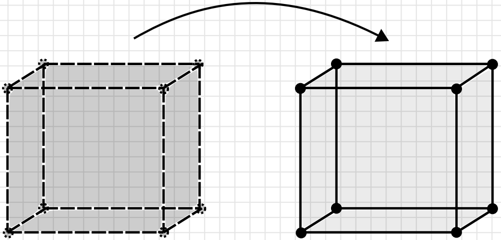
Induction on Number of Parameters

For the dimension induction, we consider a d-parameter module with height two. We want to break it into pieces with dimension less than d (and each piece we remove needs to be closed in the space containing it)



Induction on Number of Parameters

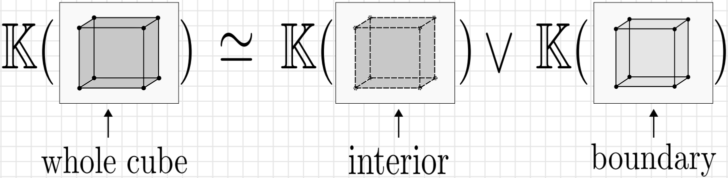
For the induction on parameters, we consider a d-parameter module with height two. We want to break it into pieces that look like lower-dimensional parameter spaces (and each piece we remove needs to be closed in the space containing it)



Step one: separate interior and boundary

Induction on Number of Parameters

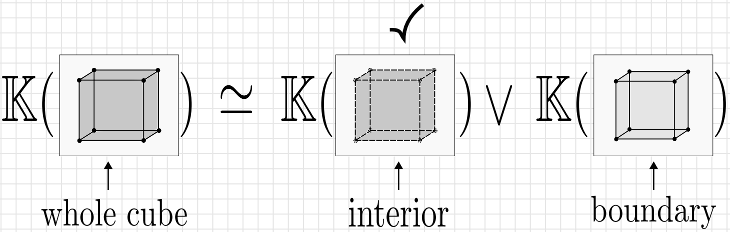
For the induction on parameters, we consider a d-parameter module with height two. We want to break it into pieces that look like lower-dimensional parameter spaces (and each piece we remove needs to be closed in the space containing it)



Step one: separate interior and boundary

Induction on Number of Parameters

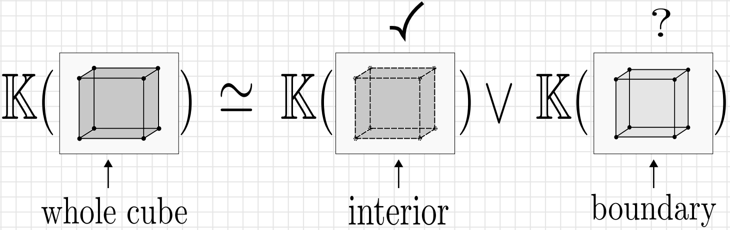
For the induction on parameters, we consider a d-parameter module with height two. We want to break it into pieces that look like lower-dimensional parameter spaces (and each piece we remove needs to be closed in the space containing it)



Step one: separate interior and boundary

Induction on Number of Parameters

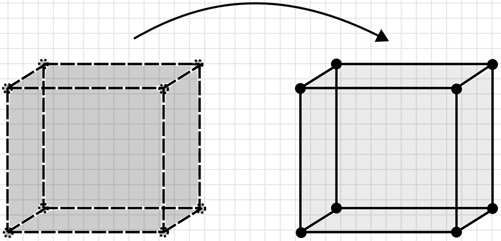
For the induction on parameters, we consider a d-parameter module with height two. We want to break it into pieces that look like lower-dimensional parameter spaces (and each piece we remove needs to be closed in the space containing it)



Step one: separate interior and boundary

Induction on Number of Parameters

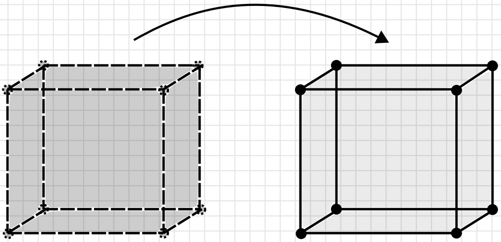
How can we break apart the boundary so that each piece looks like a parameter space with fewer than d parameters?



For $d = 1, 2, 3$, we can convince ourselves this is possible.

Induction on Number of Parameters

How can we break apart the boundary so that each piece looks like a parameter space with fewer than d parameters?



For $d = 1, 2, 3$, we can convince ourselves this is possible.
What about arbitrary finite d? Can we always remove a piece so that the remainder "falls open" into a lower-dimensional space?

Cube Unfolding

Ridge Unfoldings of Cubes (DeSplinter, Devadoss, Readyhough, 2020)

Every ridge unfolding of a finite cube will produce a net.

Cube Unfolding

Ridge Unfoldings of Cubes (DeSplinter, Devadoss, Readyhough, 2020)

Every ridge unfolding of a finite cube will produce a net.

- An **unfolding** of a polytope is the process of cutting along codimension-two faces so that the result can be isometrically immersed in one dimension lower.

Cube Unfolding

Ridge Unfoldings of Cubes (DeSplinter, Devadoss, Readyhough, 2020)

Every ridge unfolding of a finite cube will produce a net.

- An **unfolding** of a polytope is the process of cutting along codimension-two faces so that the result can be isometrically immersed in one dimension lower.
- A **net** means that the unfolded polytope does not self-overlap.

Cube Unfolding

Ridge Unfoldings of Cubes (DeSplinter, Devadoss, Readyhough, 2020)

Every ridge unfolding of a finite cube will produce a net.

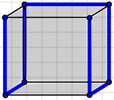
- An **unfolding** of a polytope is the process of cutting along codimension-two faces so that the result can be isometrically immersed in one dimension lower.
- A **net** means that the unfolded polytope does not self-overlap.
- Proof idea: identify a bijection between spanning trees of the dual graph of a d-cube and unfoldings, use the dual graph to show an unfolding must also be a net.

Cube Unfolding

Ridge Unfoldings of Cubes (DeSplinter, Devadoss, Readyhough, 2020)

Every ridge unfolding of a finite cube will produce a net.

This result implies we can always remove a connected collection of codimension-two faces to leave stratified spaces with fewer than d parameters.

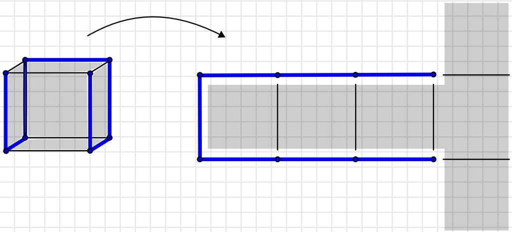


Cube Unfolding

Ridge Unfoldings of Cubes (DeSplinter, Devadoss, Readyhough, 2020)

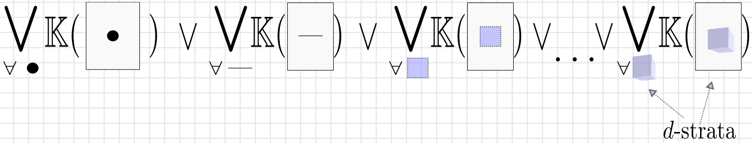
Every ridge unfolding of a finite cube will produce a net.

This result implies we can always remove a connected collection of codimension-two faces to leave a stratified $(d-1)$-parameter space.



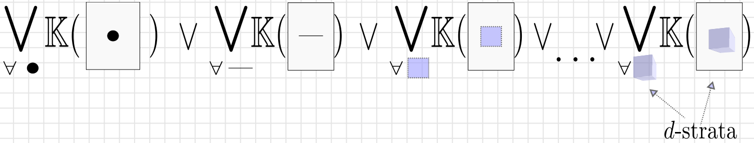
K-Theory of Multi-Parameter Persistence Modules

CLAIM:



K-Theory of Multi-Parameter Persistence Modules

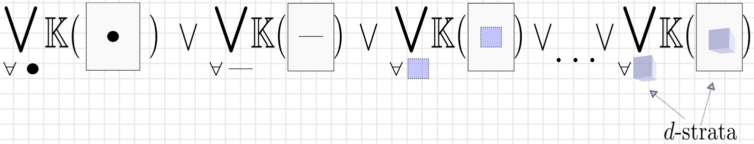
CLAIM:



What is the K-theory of persistence modules over a single strata?

K-Theory of Multi-Parameter Persistence Modules

CLAIM:

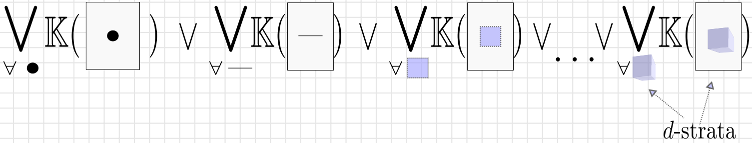


What is the K-theory of persistence modules over a single strata?

- Remember, $\mathrm{pMod}^A(X) := \mathrm{Fun}(\mathrm{Ent}_\Delta(\mathrm{X}), \mathrm{A})$.

K-Theory of Multi-Parameter Persistence Modules

CLAIM:

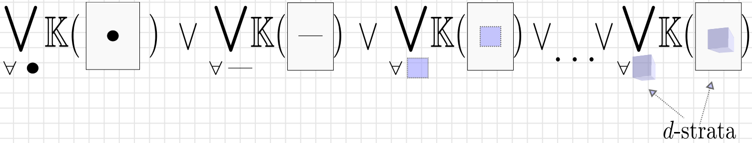


What is the K-theory of persistence modules over a single strata?

- Remember, $\mathrm{pMod}^A(X) := \mathrm{Fun}(\mathrm{Ent}_\Delta(X), A)$.
- Since we have decomposed things until X has a single strata, $\mathrm{Ent}_\Delta(X)$ is the single object, single morphism category.

K-Theory of Multi-Parameter Persistence Modules

CLAIM:

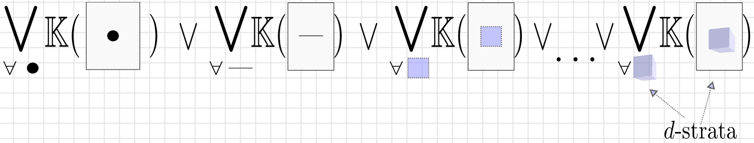


What is the K-theory of persistence modules over a single strata?

- Remember, $\mathrm{pMod}^A(X) := \mathrm{Fun}(\mathrm{Ent}_\Delta(\mathrm{X}), \mathrm{A})$.
- Since we have decomposed things until X has a single strata, $\mathrm{Ent}_\Delta(X)$ is the single object, single morphism category.
- This means $\mathrm{Fun}(\mathrm{Ent}_\Delta(\mathrm{X}), \mathrm{A}) \cong \mathrm{A}$, so that $\mathbb{K}(\mathrm{pMod}^A(X)) \cong \mathbb{K}(A)$.

K-Theory of Multi-Parameter Persistence Modules

CLAIM:



What is the K-theory of persistence modules over a single strata?

- Remember, $\mathrm{pMod}^A(X) := \mathrm{Fun}(\mathrm{Ent}_\Delta(\mathrm{X}), \mathrm{A})$.
- Since we have decomposed things until X has a single strata, $\mathrm{Ent}_\Delta(X)$ is the single object, single morphism category.
- This means $\mathrm{Fun}(\mathrm{Ent}_\Delta(\mathrm{X}), \mathrm{A}) \cong \mathrm{A}$, so that $\mathbb{K}(\mathrm{pMod}^A(X)) \cong \mathbb{K}(A)$.
- The K-theory depends on the target category A.

Vect-Valued Persistence Modules

Let's look at the specific case when A is $\mathrm{Vect}_{\mathbb{F}}$.

Vect-Valued Persistence Modules

Let's look at the specific case when A is $\mathrm{Vect}_{\mathbb{F}}$.

K-Theory of Multi-Parameter Zig-Zag Grid Modules (Grady, S.)

Let X be a cubical manifold embedded as a substratified space of $(\mathbb{R}^d; I)$, with finite I. There is an equivalence of spectra

$$\mathbb{K}(\mathrm{pMod}^{\mathrm{Vect}_{\mathbb{F}}}(X)) \simeq \bigvee_{x_0 \in X_0} \mathbb{K}(\mathbb{F}) \vee \bigvee_{x_1 \in X_1} \mathbb{K}(\mathbb{F}) \vee \ldots \vee \bigvee_{x_d \in X_d} \mathbb{K}(\mathbb{F})$$

where X_i is the set of i-strata of X.

Vect-Valued Persistence Modules

Let's look at the specific case when A is $\mathrm{Vect}_{\mathbb{F}}$.

K-Theory of Multi-Parameter Zig-Zag Grid Modules (Grady, S.)

Let X be a cubical manifold embedded as a substratified space of $(\mathbb{R}^d; I)$, with finite I. There is an equivalence of spectra

$$\mathbb{K}(\mathrm{pMod}^{\mathrm{Vect}_{\mathbb{F}}}(X)) \simeq \bigvee_{x_0 \in X_0} \mathbb{K}(\mathbb{F}) \vee \bigvee_{x_1 \in X_1} \mathbb{K}(\mathbb{F}) \vee \ldots \vee \bigvee_{x_d \in X_d} \mathbb{K}(\mathbb{F})$$

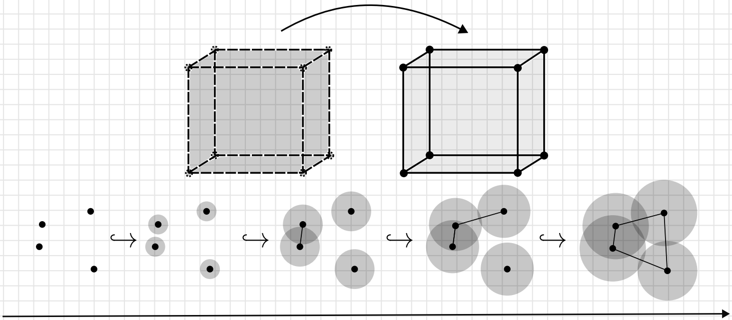
where X_i is the set of i-strata of X.

For instance, $K_0(\mathbb{F}) \cong \mathbb{Z}$, so $K_0(\mathrm{pMod}^{\mathrm{Vect}_{\mathbb{F}}}(X))$ is a direct sum of copies of $\mathbb{Z}$, one for each strata.

Next Directions

- Generalise K-theory results to persistence modules over arbitrary posets.
- Understand K_1 through natural transformations between persistence modules.

Questions?



a.k.schenfisch@tue.nl

One-parameter: https://arxiv.org/abs/2110.04591

Multi-parameter: https://arxiv.org/abs/2306.06540