Computing the Skyscraper Invariant

Jan Jendrysiak
TU Graz

October 11th, 2025

Georgia O'Keeffe, *Radiator Building*

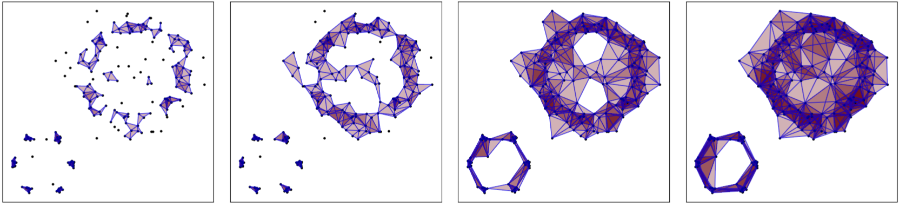
Marc Fersztand
University of Oxford

Persistence under noise

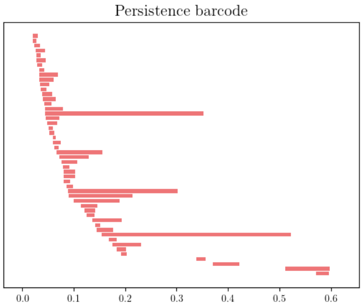
Goal

Estimate the homology groups of noisy samples.

Alpha Complex

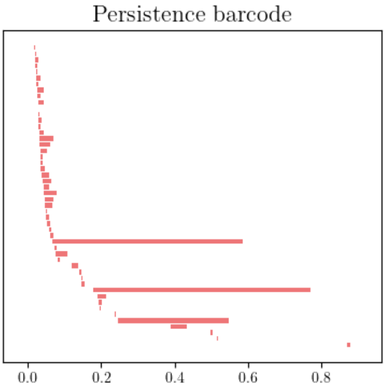


Persistence under noise



Persistence under noise

Persistence under noise



Multiparameter Persistence

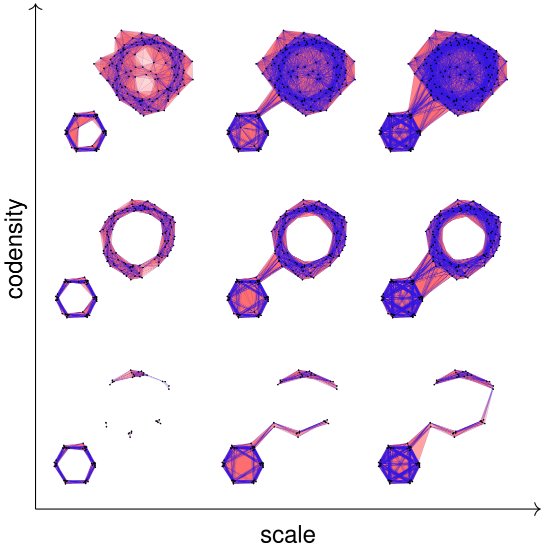
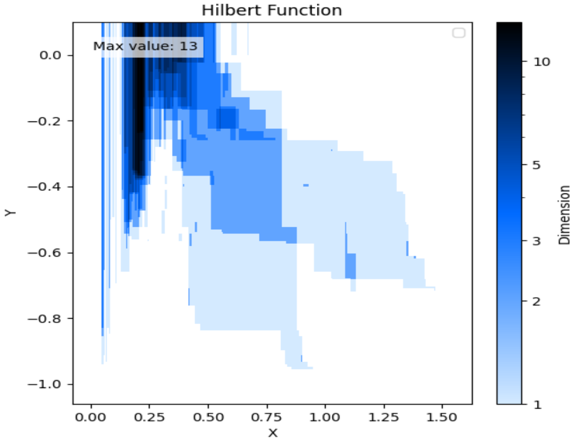


Figure: Density-Rips Bi-filtration

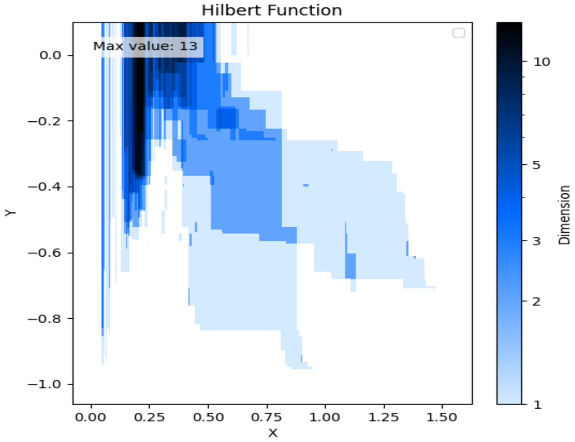
Multiparameter Persistence

$$H_1(-;\mathbb{F}_2) \Longrightarrow \quad \begin{tikzcd}
\mathbb{F}_2 \arrow[r] & 0 \arrow[r] & 0 \\
\mathbb{F}_2^2 \arrow[u, "{\begin{bmatrix} 1 & 0 \end{bmatrix}}"] \arrow[r, "{\begin{bmatrix} 0 & 1 \end{bmatrix}}"] & \mathbb{F}_2 \arrow[u] \arrow[r] & \mathbb{F}_2 \arrow[u] \\
\mathbb{F}_2 \arrow[u, "{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}"] \arrow[r] & 0 \arrow[u] \arrow[r] & 0 \arrow[u]
\end{tikzcd}$$

Multiparameter Persistence



Multiparameter Persistence



Goal

Find an invariant which describes the module in terms of interval-modules.

You could have invented Mumford Stability

for Persistence Modules

You could have invented Mumford Stability
for Persistence Modules

Definition 1

Let $X\colon \mathbb{R}^2 \to \mathbf{Vect}_{\mathbb{F}_2}$ be a persistence module and $\alpha \in \mathbb{R}^2$.
The *lifetime* of $v \in X_\alpha$ is

$$L_X(v) := \{\beta \in \mathbb{R}^2 \mid \beta \geq \alpha \text{ and } X_{\alpha\to\beta}(v) \neq 0\}$$

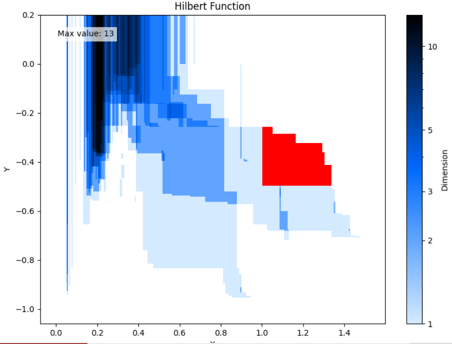
You could have invented Mumford Stability

for Persistence Modules

Definition 1

Let $X : \mathbb{R}^2 \to \mathbf{Vect}_{\mathbb{F}_2}$ be a persistence module and $\alpha \in \mathbb{R}^2$.
The *lifetime* of $v \in X_\alpha$ is

$$L_X(v) := \{\beta \in \mathbb{R}^2 \mid \beta \geq \alpha \text{ and } X_{\alpha\to\beta}(v) \neq 0\}$$



You could have invented Mumford Stability

for Persistence Modules

Definition 2

Let $S \subset X$ be a subset. We denote by $\langle S \rangle \subset X$ its induced submodule.

You could have invented Mumford Stability

for Persistence Modules

Definition 2

Let $S \subset X$ be a subset. We denote by $\langle S \rangle \subset X$ its induced submodule.

$\Rightarrow L_X(v) = \operatorname{supp}\langle\{v\}\rangle$

You could have invented Mumford Stability

for Persistence Modules

Definition 2

Let $S \subset X$ be a subset. We denote by $\langle S \rangle \subset X$ its induced submodule.

$\Rightarrow L_X(v) = \operatorname{supp}\langle\{v\}\rangle$

Observation

If $v \in X_\alpha$, then $\langle\{v\}\rangle$ is an interval module.

You could have invented Mumford Stability

for Persistence Modules

Definition 2

Let $S \subset X$ be a subset. We denote by $\langle S \rangle \subset X$ its induced submodule.

$\Rightarrow L_X(v) = \operatorname{supp}\langle \{v\} \rangle$

Observation

If $v \in X_\alpha$, then $\langle \{v\} \rangle$ is an interval module.

Lifetime Filtration (Miller, Zhang)

Find $v_1 \in X$ with the longest lifetime. $\langle v_1 \rangle \subset X$ is an *interval-module*.
Compute $X_1 := X / \langle v \rangle$ and repeat.

$$\Rightarrow \langle v_1 \rangle \hookrightarrow X^1 \hookrightarrow \ldots \hookrightarrow X \qquad X^k / X^{k-1} = \langle v_k \rangle.$$

You could have invented Mumford Stability

for Persistence Modules

$$
\begin{tikzcd}
\mathbb{F}_2 \arrow[r] & 0 \arrow[r] & 0 \\
\mathbb{F}_2^2 \arrow[u, "\begin{bmatrix} 1 & 0 \end{bmatrix}"] \arrow[r, "\begin{bmatrix} 0 & 1 \end{bmatrix}"] & \mathbb{F}_2 \arrow[u] \arrow[r] & \mathbb{F}_2 \arrow[u] \\
\mathbb{F}_2 \arrow[u, "\begin{bmatrix} 1 \\ 0 \end{bmatrix}"] \arrow[r] & 0 \arrow[u] \arrow[r] & 0 \arrow[u]
\end{tikzcd}
$$

You could have invented Mumford Stability

for Persistence Modules

$$
\begin{tikzcd}
\mathbb{F}_2 \arrow[r] & 0 \arrow[r] & 0 \\
\mathbb{F}_2^2 \arrow[u, "\begin{bmatrix} 1 & 0 \end{bmatrix}"] \arrow[r, "\begin{bmatrix} 0 & 1 \end{bmatrix}"] & \mathbb{F}_2 \arrow[u] \arrow[r] & \mathbb{F}_2 \arrow[u] \\
\mathbb{F}_2 \arrow[u, "\begin{bmatrix} 1 \\ 0 \end{bmatrix}"] \arrow[r] & 0 \arrow[u] \arrow[r] & 0 \arrow[u]
\end{tikzcd}
$$

Question

Which vectors in the above module have the longest lifetime?

You could have invented Mumford Stability

for Persistence Modules

$$v_1 := \begin{pmatrix} 1 \\ 1 \end{pmatrix} \in \begin{tikzcd} \mathbb{F} \arrow[r] & 0 \arrow[r] & 0 \\ \mathbb{F}_2^2 \arrow[u, "{\begin{bmatrix} 1 & 0 \end{bmatrix}}"] \arrow[r, "{\begin{bmatrix} 0 & 1 \end{bmatrix}}"] & \mathbb{F}_2 \arrow[u] \arrow[r] & \mathbb{F}_2 \arrow[u] \\ \mathbb{F}_2 \arrow[u, "{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}"] \arrow[r] & 0 \arrow[u] \arrow[r] & 0 \arrow[u] \end{tikzcd}$$

You could have invented Mumford Stability

for Persistence Modules

$$X/\langle v_1 \rangle \simeq \begin{tikzcd} 0 \arrow[r] & 0 \arrow[r] & 0 \\ \mathbb{F}_2 \arrow[r] \arrow[u] & 0 \arrow[r] \arrow[u] & 0 \arrow[u] \\ \mathbb{F}_2 \arrow[r] \arrow[u] & 0 \arrow[r] \arrow[u] & 0 \arrow[u] \end{tikzcd}$$

You could have invented Mumford Stability

for Persistence Modules

$$
X/\langle v_1 \rangle \simeq \begin{tikzcd}
0 \arrow[r] & 0 \arrow[r] & 0 \\
\mathbb{F}_2 \arrow[r] \arrow[u] & 0 \arrow[r] \arrow[u] & 0 \arrow[u] \\
\mathbb{F}_2 \arrow[r] \arrow[u] & 0 \arrow[r] \arrow[u] & 0 \arrow[u]
\end{tikzcd}
$$

Problem

The element of largest lifetime is not *additive*.

You could have invented Mumford Stability

for Persistence Modules

The module above decomposes as

$$\begin{tikzcd} \mathbb{F}_2 \arrow[r] & 0 \arrow[r] & 0 \\ \mathbb{F}_2 \arrow[r] \arrow[u] & 0 \arrow[r] \arrow[u] & 0 \arrow[u] \\ \mathbb{F}_2 \arrow[r] \arrow[u] & 0 \arrow[r] \arrow[u] & 0 \arrow[u] \end{tikzcd} \bigoplus \begin{tikzcd} 0 \arrow[r] & 0 \arrow[r] & 0 \\ \mathbb{F}_2 \arrow[r] \arrow[u] & \mathbb{F}_2 \arrow[r] \arrow[u] & \mathbb{F}_2 \arrow[u] \\ 0 \arrow[r] \arrow[u] & 0 \arrow[r] \arrow[u] & 0 \arrow[u] \end{tikzcd}$$

You could have invented Mumford Stability

for Persistence Modules

Idea

Instead find the vector of *shortest* lifetime.

You could have invented Mumford Stability
for Persistence Modules

Idea
Instead find the vector of *shortest* lifetime.

Not well defined.

You could have invented Mumford Stability
for Persistence Modules

Idea
Instead find the vector of *shortest* lifetime.

Not well defined. $\Rightarrow$ Fix $\alpha \in \mathbb{R}^2$ and force $v \in X_\alpha$.

You could have invented Mumford Stability
for Persistence Modules

Idea

Instead find the vector of *shortest* lifetime.

Not well defined. $\Rightarrow$ Fix $\alpha \in \mathbb{R}^2$ and force $v \in X_\alpha$.

You could have invented Mumford Stability
for Persistence Modules

Idea

Instead find the vector of *shortest* lifetime.

Not well defined. $\Rightarrow$ Fix $\alpha \in \mathbb{R}^2$ and force $v \in X_\alpha$.

Definition 3

For $v \in X_\alpha$ define its *slope*

$$\mu_\alpha(v) := 1 \bigg/ \int_{\mathbb{R}^2} \mathbb{1}_{L_X(v)} \, dx.$$

You could have invented Mumford Stability

for Persistence Modules

Idea

Instead find the vector of *shortest* lifetime.

Not well defined. $\Rightarrow$ Fix $\alpha \in \mathbb{R}^2$ and force $v \in X_\alpha$.

Definition 3

For $v \in X_\alpha$ define its *slope*

$$\mu_\alpha(v) := 1 \Big/ \int_{\mathbb{R}^2} \mathbb{1}_{L_X(v)} \, dx.$$

The element of highest slope is still not *unique*.

$$\begin{tikzcd}
\mathbb{F}_2 \arrow[r] & 0 \\
\mathbb{F}_2^2 \arrow[u, "{\begin{bmatrix}0 & 1\end{bmatrix}}"] \arrow[r, "{\begin{bmatrix}1 & 0\end{bmatrix}}"] & \mathbb{F}_2 \arrow[u]
\end{tikzcd}$$

You could have invented Mumford Stability
for Persistence Modules

Idea

Consider also *subspaces* of X_α.

You could have invented Mumford Stability

for Persistence Modules

Idea

Consider also *subspaces* of X_α.

Definition 4

Fix $\alpha \in \mathbb{R}^2$. For any vector subspace $V \subset X_\alpha$, define the *slope*

$$\mu_\alpha(V) := \dim_{\mathbb{F}_2} V \bigg/ \int_{\mathbb{R}^2} \dim_{\mathbb{F}_2} \langle V \rangle \ .$$

You could have invented Mumford Stability

for Persistence Modules

Idea

Consider also *subspaces* of X_α.

Definition 4

Fix $\alpha \in \mathbb{R}^2$. For any vector subspace $V \subset X_\alpha$, define the *slope*

$$\mu_\alpha(V) := \dim_{\mathbb{F}_2} V \Big/ \int_{\mathbb{R}^2} \dim_{\mathbb{F}_2} \langle V \rangle \ .$$

$$\begin{tikzcd}
\mathbb{F}_2 \arrow[r] & 0 \arrow[r] & 0 \\
\mathbb{F}_2^2 \arrow[u, "{\begin{bmatrix} 0 & 0 \end{bmatrix}}"] \arrow[r, "\mathrm{Id}"] & \mathbb{F}_2^2 \arrow[u] \arrow[r, "{\begin{bmatrix} 0 & 1 \end{bmatrix}}"] & \mathbb{F}_2 \arrow[u]
\end{tikzcd}$$

You could have invented Mumford Stability

for Persistence Modules

Idea

Consider also *subspaces* of X_α.

Definition 4

Fix $\alpha \in \mathbb{R}^2$. For any vector subspace $V \subset X_\alpha$, define the *slope*

$$\mu_\alpha(V) := \dim_{\mathbb{F}_2} V \Big/ \int_{\mathbb{R}^2} \dim_{\mathbb{F}_2} \langle V \rangle \, .$$

$$\begin{tikzcd}
\mathbb{F}_2 \arrow[r] & 0 \arrow[r] & 0 \\
\mathbb{F}_2^2 \arrow[u, "{\begin{bmatrix} 0 & 0 \end{bmatrix}}"] \arrow[r, "\mathrm{Id}"] & \mathbb{F}_2^2 \arrow[u] \arrow[r, "{\begin{bmatrix} 0 & 1 \end{bmatrix}}"] & \mathbb{F}_2 \arrow[u]
\end{tikzcd}$$

$$\mu(\mathbb{F}_2^2) = 2/5.$$

You could have invented Mumford Stability
for Persistence Modules

Definition 5

- Find a maximal subspace $V \subset X_\alpha$ of highest slope $\mu_\alpha(V)$.
- Set $X_1 := X \big/ \langle V \rangle$ and repeat.

The resulting filtration

$$X^0 \hookrightarrow X^1 \hookrightarrow \ldots \hookrightarrow \langle X_\alpha \rangle \hookrightarrow X$$

is called the *Harder-Narasimhan-Filtration* at α.

Definition 6

A sub-space V is called *stable*, if for all $W \subset V$ we have

$$\mu_\alpha(W) < \mu_\alpha(V).$$

You could have invented Mumford Stability
for Persistence Modules

Definition 5

- Find a maximal subspace $V \subset X_\alpha$ of highest slope $\mu_\alpha(V)$.
- Set $X_1 := X \big/ \langle V \rangle$ and repeat.

The resulting filtration

$$X^0 \hookrightarrow X^1 \hookrightarrow \ldots \hookrightarrow \langle X_\alpha \rangle \hookrightarrow X$$

is called the *Harder-Narasimhan-Filtration* at α.

Definition 6

A sub-space V is called *stable*, if for all $W \subset V$ we have

$$\mu_\alpha(W) < \mu_\alpha(V).$$

Proposition

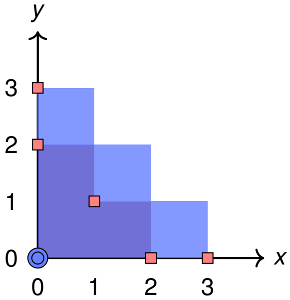
The filtration factors $\langle V \rangle^i := X^i / X^{i-1}$ in the HN-Filtration are stable and sorted by slope.

You could have invented Mumford Stability
for Persistence Modules

Stable modules can be complicated.

You could have invented Mumford Stability
for Persistence Modules

Stable modules can be complicated.

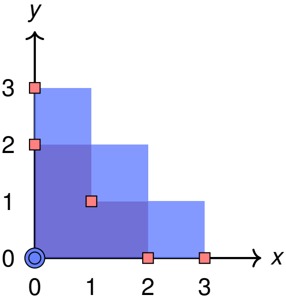


$$
\begin{tikzcd}
\mathbb{F}_2 & \mathbb{F}_2 \\
\mathbb{F}_2^2 \arrow[u, "{\begin{bmatrix}1 & 0\end{bmatrix}}"] \arrow[ur, "{\begin{bmatrix}1 & 1\end{bmatrix}}"] \arrow[r, "{\begin{bmatrix}0 & 1\end{bmatrix}}"'] & \mathbb{F}_2
\end{tikzcd}
$$

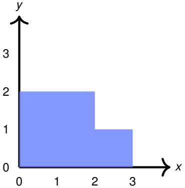
You could have invented Mumford Stability

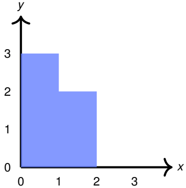
for Persistence Modules

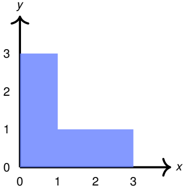
Stable modules can be complicated.



$$\begin{tikzcd} & \mathbb{F}_2 & \mathbb{F}_2 \\ & \mathbb{F}_2^2 \arrow[u, "\begin{bmatrix}1 & 0\end{bmatrix}"] \arrow[ur, "\begin{bmatrix}1 & 1\end{bmatrix}"] \arrow[r, "\begin{bmatrix}0 & 1\end{bmatrix}"'] & \mathbb{F}_2 \end{tikzcd}$$







Mumford Stability for Persistence Modules

Congratulations

You have discovered Mumford-Stability for Persistence Modules!

Mumford Stability for Persistence Modules

Congratulations

You have discovered Mumford-Stability for Persistence Modules!

Stability Conditions

Let ν, ξ be adequate measures on $\mathbb{R}^2$. For any submodule $Y \subset X$ they define a slope

$$\mu_{\nu,\xi}(Y) := \int_{\mathbb{R}^2} \dim Y d\nu \Big/ \int_{\mathbb{R}^2} \dim Y d\xi \ .$$

Mumford Stability for Persistence Modules

Congratulations

You have discovered Mumford-Stability for Persistence Modules!

Stability Conditions

Let ν, ξ be adequate measures on $\mathbb{R}^2$. For any submodule $Y \subset X$ they define a slope

$$\mu_{\nu,\xi}(Y) := \int_{\mathbb{R}^2} \dim Y d\nu \Big/ \int_{\mathbb{R}^2} \dim Y d\xi \ .$$

We are using the special case $\nu = \delta_\alpha, \xi = d\lambda$ - a *Skyscraper*.

The Skyscraper Invariant

Definition 7

For $\alpha \leq \beta \in \mathbb{R}^2$ and $\langle V \rangle^i := X^i / X^{i-1}$ the stable filtration factors of the HN-Filtration. We define the *Skyscraper Invariant* as

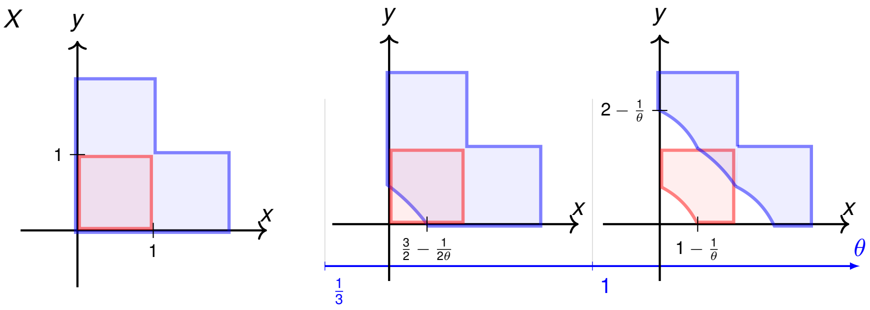
$$s^{\theta}_{\alpha,\beta}(X) := \bigoplus_{i \text{ st. } \mu(V^i) > \theta} \dim_{\mathbb{F}_2} V^i_\beta.$$

The Skyscraper Invariant

Definition 7

For $\alpha \leq \beta \in \mathbb{R}^2$ and $\langle V\rangle^i := X^i/X^{i-1}$ the stable filtration factors of the HN-Filtration. We define the *Skyscraper Invariant* as

$$s^{\theta}_{\alpha,\beta}(X) := \bigoplus_{i \text{ st. } \mu(V^i)>\theta} \dim_{\mathbb{F}_2} V^i_{\beta}.$$



Computing Harder-Narasimhan Filtrations

Definition 8

Let $\varepsilon > 0$. We denote by $E_\varepsilon(X) := (\varepsilon\mathbb{Z})^2 \cap \mathrm{supp}(X)$ the equidistant grid on X.

Computing Harder-Narasimhan Filtrations

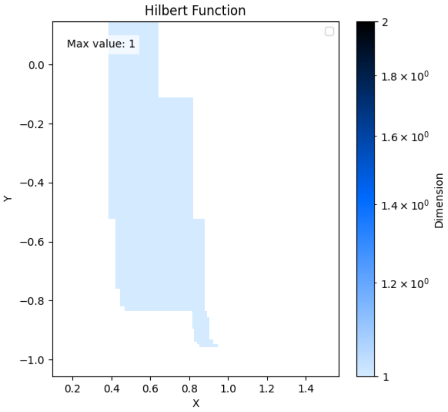
Definition 8

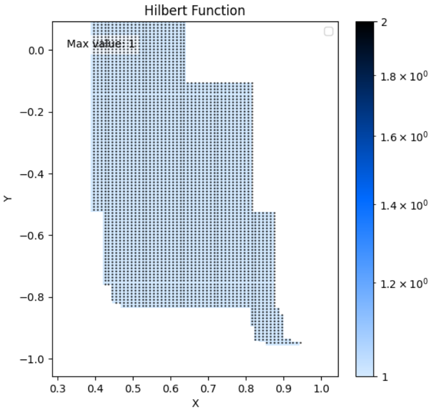
Let $\varepsilon > 0$. We denote by $E_\varepsilon(X) \coloneqq (\varepsilon\mathbb{Z})^2 \cap \operatorname{supp}(X)$ the equidistant grid on X. Define $d \coloneqq \operatorname{diam}(X)/\varepsilon$..

Computing Harder-Narasimhan Filtrations

Definition 8

Let $\varepsilon > 0$. We denote by $E_\varepsilon(X) := (\varepsilon\mathbb{Z})^2 \cap \mathrm{supp}(X)$ the equidistant grid on X. Define $d := \mathrm{diam}(X)/\varepsilon$..





Computing Harder-Narasimhan Filtrations

Task

For each $\alpha \in E_\varepsilon(X)$, compute the HN-filtration of $\langle X_\alpha \rangle \subset X$.

Recall: That means finding vectors of small lifetime.

Computing Harder-Narasimhan Filtrations

Task

For each $\alpha \in E_\varepsilon(X)$, compute the HN-filtration of $\langle X_\alpha \rangle \subset X$.

Recall: That means finding vectors of small lifetime.

Problems

1. $|E_\varepsilon(X)| \in \mathcal{O}(d^2)$

Computing Harder-Narasimhan Filtrations

Task

For each $\alpha \in E_\varepsilon(X)$, compute the HN-filtration of $\langle X_\alpha \rangle \subset X$.

Recall: That means finding vectors of small lifetime.

Problems

1. $|E_\varepsilon(X)| \in \mathcal{O}(d^2)$
2. $\langle X_\alpha \rangle$ can have very high dimension $\Rightarrow$ computing the HN filtration is impossible.

Computing Harder-Narasimhan Filtrations

Task

For each $\alpha \in E_\varepsilon(X)$, compute the HN-filtration of $\langle X_\alpha \rangle \subset X$.

Recall: That means finding vectors of small lifetime.

Problems

1. $|E_\varepsilon(X)| \in \mathcal{O}(d^2)$
2. $\langle X_\alpha \rangle$ can have very high dimension $\Rightarrow$ computing the HN filtration is impossible.

Cheng's algorithm ('24)

Computes the HN filtration of a quiver representation in poly-time.

Computing Harder-Narasimhan Filtrations

Task

For each $\alpha \in E_\varepsilon(X)$, compute the HN-filtration of $\langle X_\alpha \rangle \subset X$.

Recall: That means finding vectors of small lifetime.

Problems

1. $|E_\varepsilon(X)| \in \mathcal{O}(d^2)$
2. $\langle X_\alpha \rangle$ can have very high dimension $\Rightarrow$ computing the HN filtration is impossible.

Cheng's algorithm ('24)

Computes the HN filtration of a quiver representation in poly-time.
Computing the quiver representation is in $\mathcal{O}(d^2 n^3)$.

Computing Harder-Narasimhan Filtrations

Task

For each $\alpha \in E_\varepsilon(X)$, compute the HN-filtration of $\langle X_\alpha \rangle \subset X$.

Recall: That means finding vectors of small lifetime.

Problems

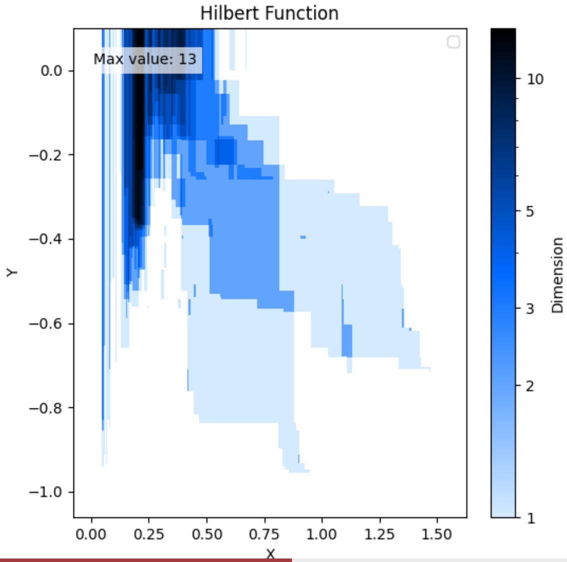
1. $|E_\varepsilon(X)| \in \mathcal{O}(d^2)$
2. $\langle X_\alpha \rangle$ can have very high dimension $\Rightarrow$ computing the HN filtration is impossible.

Cheng's algorithm ('24)

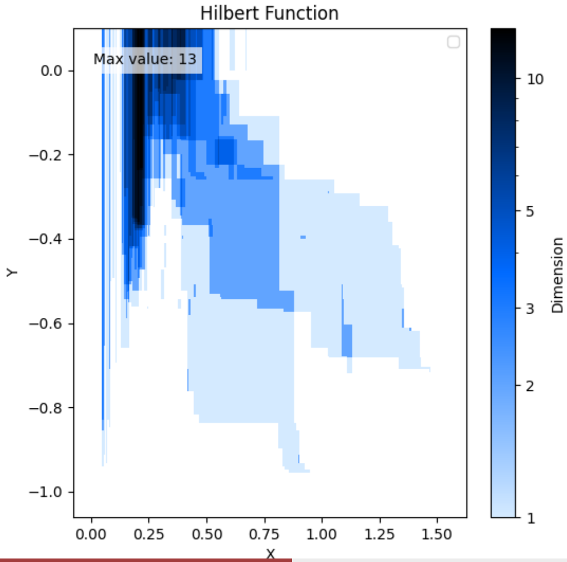
Computes the HN filtration of a quiver representation in poly-time.
Computing the quiver representation is in $\mathcal{O}(d^2 n^3)$.
Cheng's algorithm performs reduction on matrices of size $\mathcal{O}(\theta(X)^2 d^2)$

The Skyscraper Algorithm

The Skyscraper Algorithm

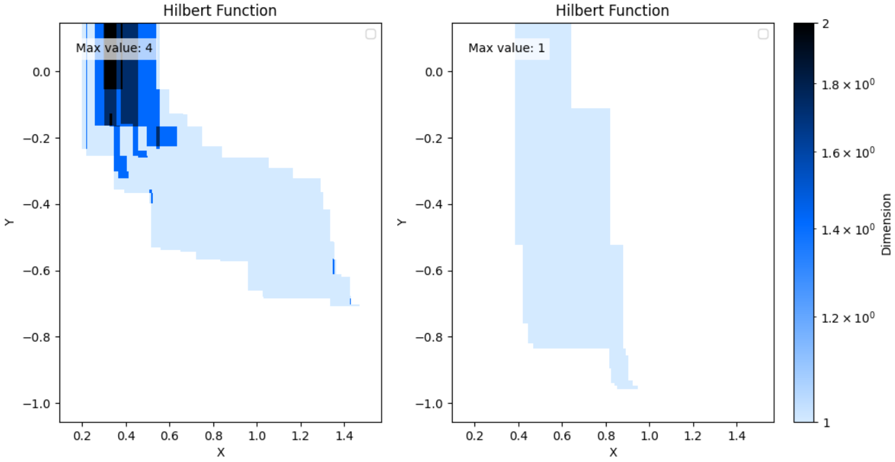


The Skyscraper Algorithm



Decompose with AIDA!

The Skyscraper Algorithm

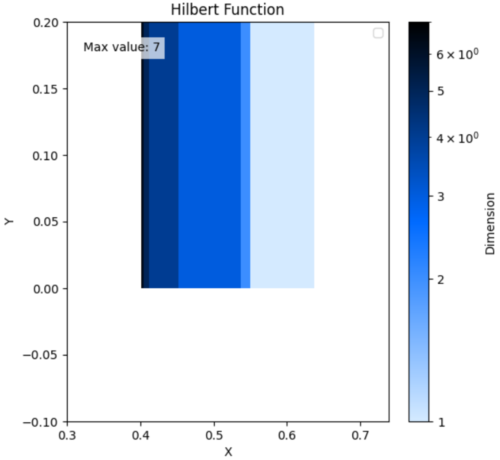


The Skyscraper Algorithm

Compute $\langle X_{(0.4,0.0)} \rangle$:

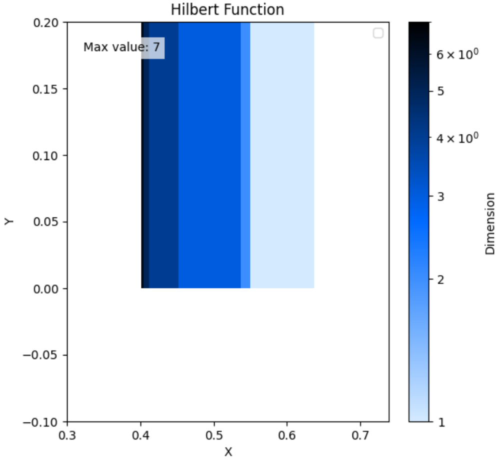
The Skyscraper Algorithm

Compute $\langle X_{(0.4,0.0)} \rangle$:



The Skyscraper Algorithm

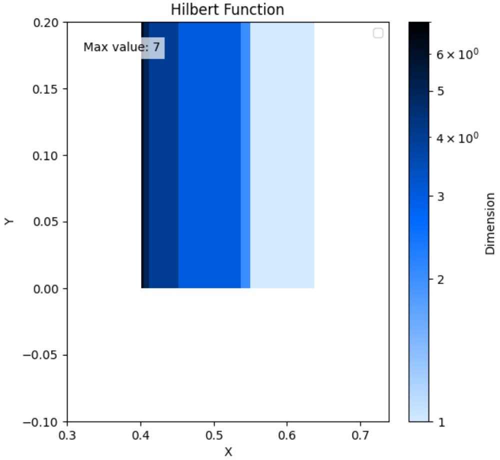
Compute $\langle X_{(0.4,0.0)} \rangle$:



This module can still have large dimension.
$>$ 100 not unusual for larger pointclouds

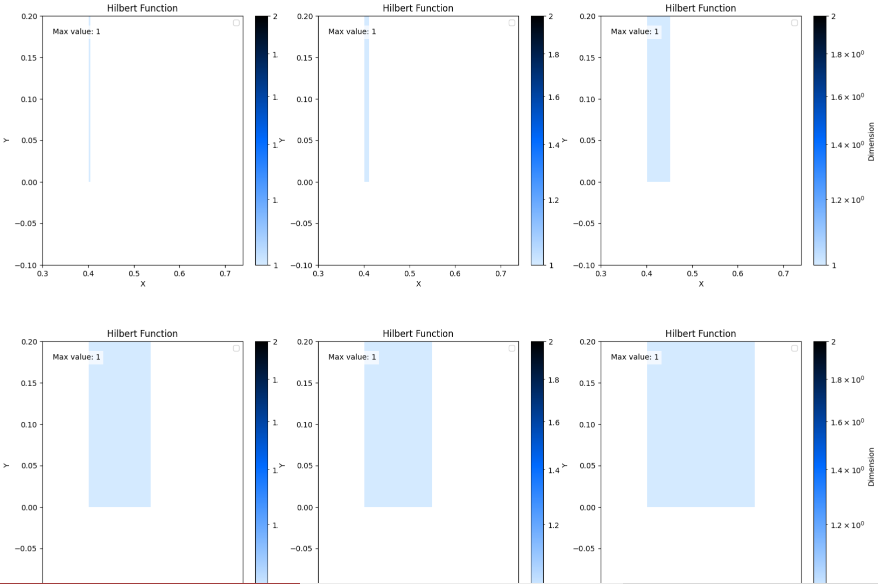
The Skyscraper Algorithm

Compute $\langle X_{(0.4,0.0)} \rangle$:



This module can still have large dimension.
$>$ 100 not unusual for larger pointclouds
$\Rightarrow$ Decompose again!

The Skyscraper Algorithm



Empirical Results

Experimental Observations

The modules $\langle X_\alpha \rangle$ are highly decomposable.

Empirical Results

Experimental Observations

The modules $\langle X_\alpha \rangle$ are highly decomposable.
The pointwise dimension of the indecomposable components of $\langle X_\alpha \rangle$ for typical bifiltrations is always ≤ 3!

Empirical Results

Experimental Observations

The modules $\langle X_\alpha \rangle$ are highly decomposable.
The pointwise dimension of the indecomposable components of $\langle X_\alpha \rangle$ for typical bifiltrations is always ≤ 3!

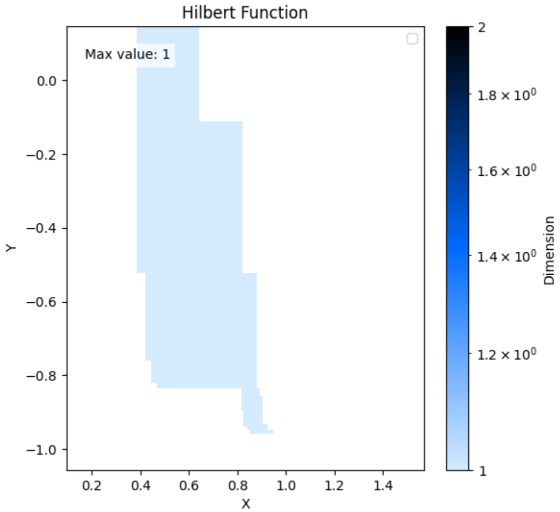
$\Rightarrow$ Can brute-force the computation of the HN filtration.

The Skyscraper Algorithm

Still need to compute this for every $\alpha \in E_\varepsilon(X)$ for every indecomposable component.

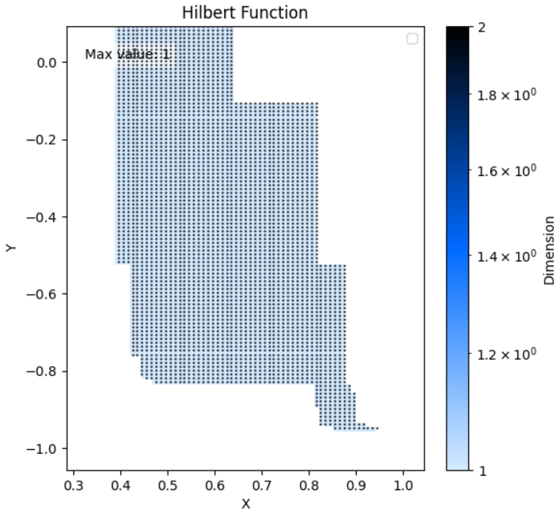
The Skyscraper Algorithm

Still need to compute this for every $\alpha \in E_\varepsilon(X)$ for every indecomposable component.



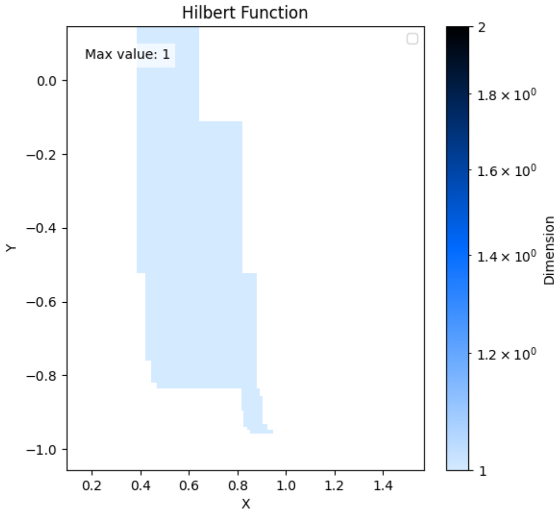
The Skyscraper Algorithm

Still need to compute this for every $\alpha \in E_\varepsilon(X)$ for every indecomposable component.



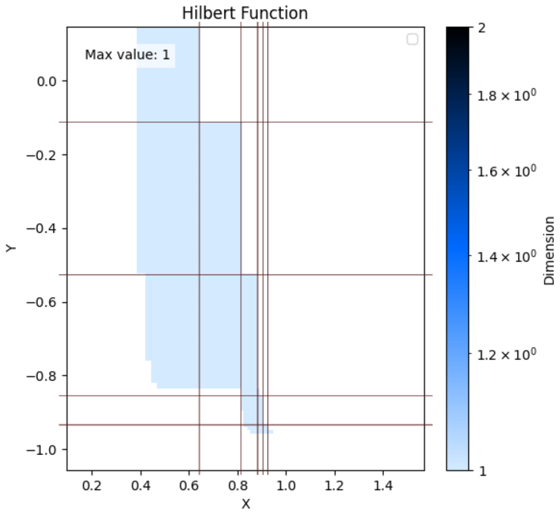
The Skyscraper Algorithm

Still need to compute this for every $\alpha \in E_\varepsilon(X)$ for every indecomposable component.

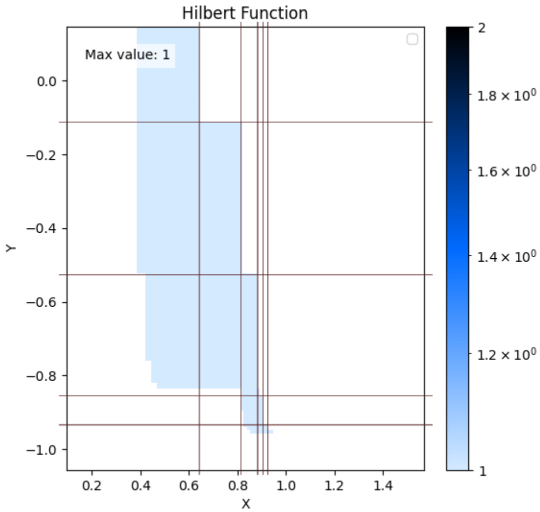


The Induced Grid of a Module

The Induced Grid of a Module



The Induced Grid of a Module

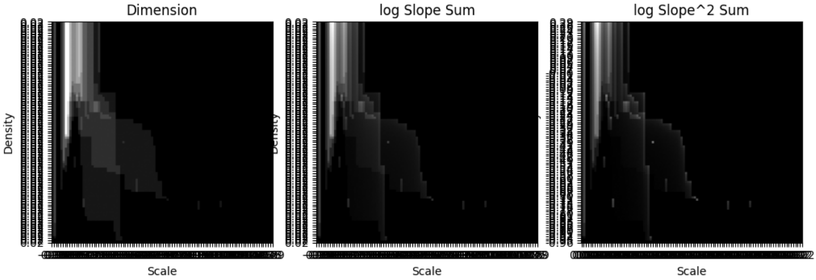


Idea

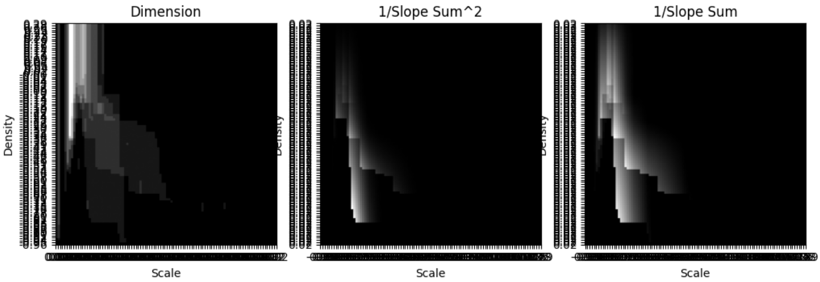
Let $\alpha \leq \beta$. $\langle X_\alpha \rangle$ and $\langle X_\beta \rangle$ are almost the same if there are no generators or relations in $\langle \alpha \rangle \setminus \langle \beta \rangle$.

Results

Results


Dimension
log Slope Sum
log Slope^2 Sum
Density
Scale
Scale
Scale

Results



The size of Stable Submodules

Previous computation, 100×100 grid:
$\sim$ 5.0 s when going over whole grid, $\sim$ 0.5 s using the induced grid.

The size of Stable Submodules

Previous computation, 100×100 grid:
$\sim$ 5.0 s when going over whole grid, $\sim$ 0.5 s using the induced grid.

Main Result

In all examples of density-scale bifiltrations, every stable factor module of the Harder-Narasimhan filtration is an interval.

The size of Stable Submodules

Previous computation, 100×100 grid:
~ 5.0 s when going over whole grid, ~ 0.5 s using the induced grid.

Main Result

In all examples of density-scale bifiltrations, every stable factor module of the Harder-Narasimhan filtration is an interval.

Open Questions

- Is there a geometric reason for this?

The size of Stable Submodules

Previous computation, 100×100 grid:
$\sim$ 5.0 s when going over whole grid, $\sim$ 0.5 s using the induced grid.

Main Result

In all examples of density-scale bifiltrations, every stable factor module of the Harder-Narasimhan filtration is an interval.

Open Questions

- Is there a geometric reason for this?
- How to compute HN filtrations for other central charges

The size of Stable Submodules

Previous computation, 100×100 grid:
$\sim$ 5.0 s when going over whole grid, $\sim$ 0.5 s using the induced grid.

Main Result

In all examples of density-scale bifiltrations, every stable factor module of the Harder-Narasimhan filtration is an interval.

Open Questions

- Is there a geometric reason for this?
- How to compute HN filtrations for other central charges
- Compute the resulting filtered landscapes and signed barcodes fast.

The Skyscraper Algorithm

Software

- github.com/jendjan/Skyscraper-Invariant
- github.com/jendjan/AIDA
- github.com/jendjan/Persistence-Algebra

The Skyscraper Algorithm

Software

- github.com/jendjan/Skyscraper-Invariant
- github.com/jendjan/AIDA
- github.com/jendjan/Persistence-Algebra

Thank you!