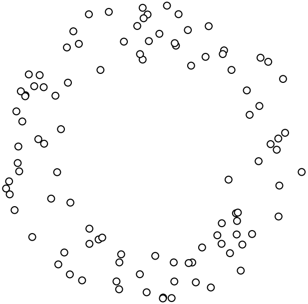
Spectral Sequences for Applied Topology
Thesis Defense

Ryan H. Lewis

Advisor: Gunnar Carlsson

May 20, 2016

Applied Topology

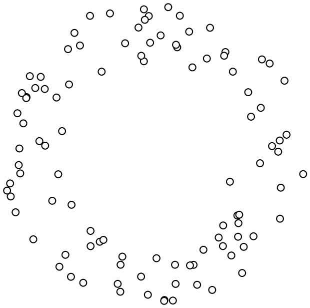


What is this?

What does it look like ?

1. Set of points in $\mathbb{R}^2$

Applied Topology

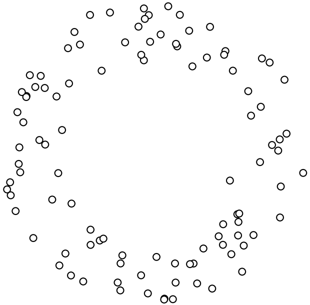


What is this?

What does it look like ?

1. Set of points in $\mathbb{R}^2$
2. Looks like an annulus.

Applied Topology



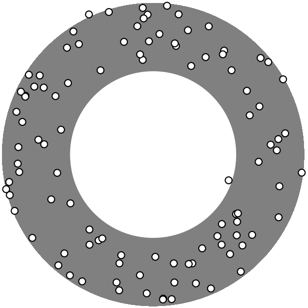
1. Set of points in $\mathbb{R}^2$
2. Looks like an annulus.

What is this?

What does it look like ?

Applied topology about recovering shape from [geometric] data

Applied Topology



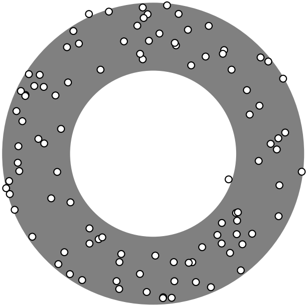
1. Set of points in $\mathbb{R}^2$
2. Looks like an annulus.

What is this?

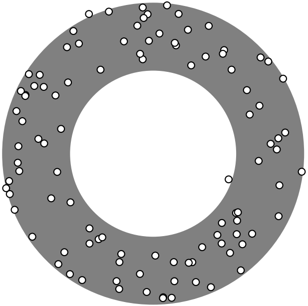
What does it look like ?

Applied topology about recovering shape from [geometric] data

Data Has Shape

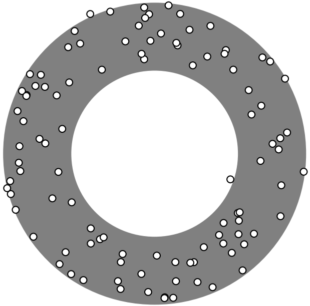


Data Has Shape



2-dimensional

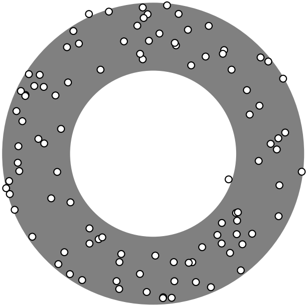
Data Has Shape



2-dimensional

Approximates annulus

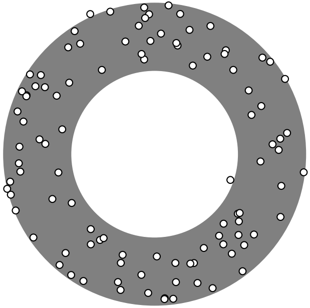
Data Has Shape



2-dimensional

Approximates annulus

Data Has Shape

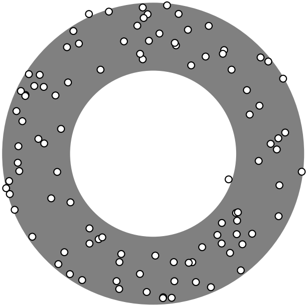


2-dimensional

Approximates annulus

Topological features of annulus:

Data Has Shape



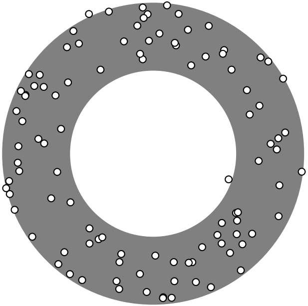
2-dimensional

Approximates annulus

Topological features of annulus:

1 component

Data Has Shape



2-dimensional

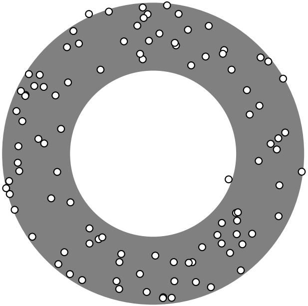
Approximates annulus

Topological features of annulus:

1 component

1 loop

Data Has Shape



2-dimensional

Approximates annulus

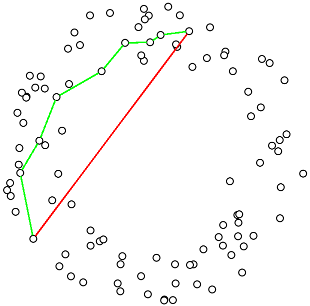
Topological features of annulus:

1 component

1 loop

Goal: Recover *topology* of annulus from point cloud

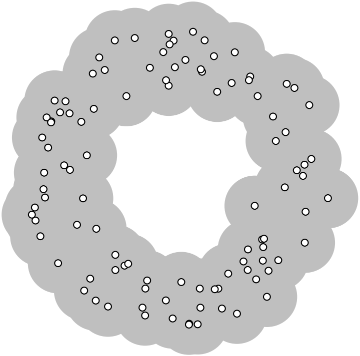
Wherefore topology?



Geometry is too rigid!
Trouble with distance metrics:

Unreliable	Only trust small distances
Ill motivated	The metrics in use may be arbitrarily chosen

Wherefore topology?



Depend only on *nearness*.

Count qualitative features.

Dimension Agnostic.

Applications of Persistent Homology

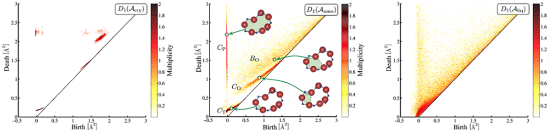


Figure: *Si*02 in different states has different P.H. Credit: Hiraoka

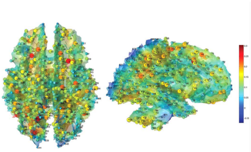


Figure: Neuroscience. Credit: Chung

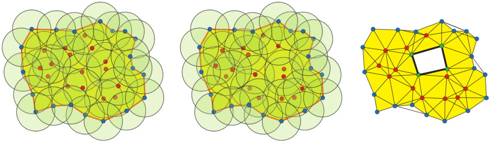


Figure: Sensor Networks

The Agenda

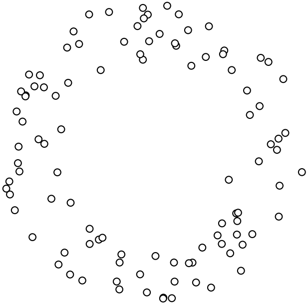
Today we will present distributed algorithms for homology and persistent homology, via spectral sequences.

- Motivation
- **Background**
- Persistent Homology and a Spectral Sequence
- Homology from subcomplexes: Mayer Vietoris
- Mayer Vietoris & Persistence

A model space

For a dataset X we study the topology of the *union of balls*

$$M_\epsilon = \bigcup_{x \in X} B_\epsilon(x)$$

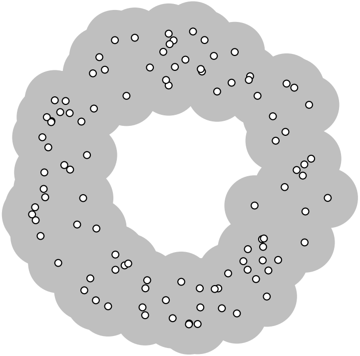


$r = 0$

A model space

For a dataset X we study the topology of the *union of balls*

$$M_\epsilon = \bigcup_{x \in X} B_\epsilon(x)$$

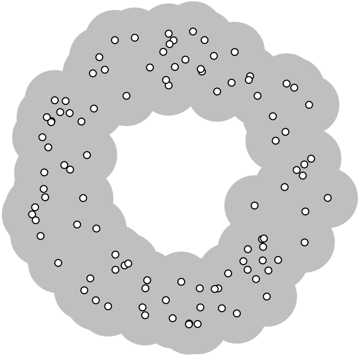


$r = .8$

A model space

For a dataset X we study the topology of the *union of balls*

$$M_\epsilon = \bigcup_{x \in X} B_\epsilon(x)$$



$r = .8$

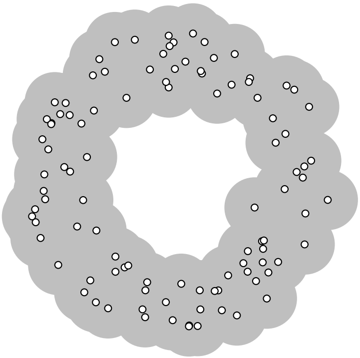
A model space

For a dataset X we study the topology of the *union of balls*

$$M_\epsilon = \bigcup_{x \in X} B_\epsilon(x)$$

Two Issues:

Scale: No natural choice of ϵ!



$r = .8$

A model space

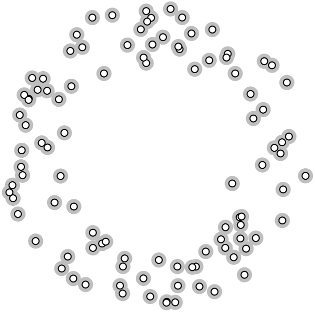
For a dataset X we study the topology of the *union of balls*

$$M_\epsilon = \bigcup_{x \in X} B_\epsilon(x)$$

Two Issues:

Scale: No natural choice of ϵ!

Conception: How to encode M_ϵ on computer?



$r = .2$

A model space

For a dataset X we study the topology of the *union of balls*

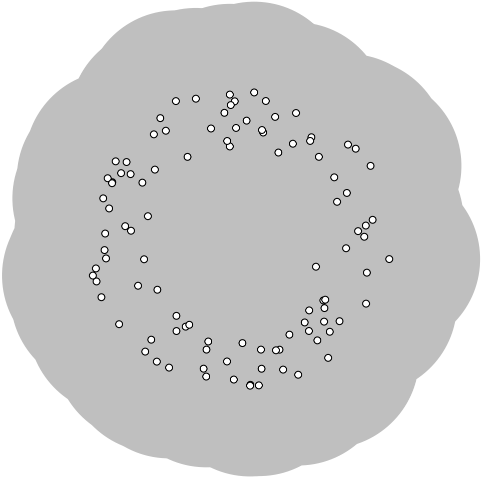
$$M_\epsilon = \bigcup_{x \in X} B_\epsilon(x)$$

Two Issues:

Scale: No natural choice of ϵ!

Conception: How to encode M_ϵ on computer?

Answer: Topology



$r = \infty$

A model space

For a dataset X we study the topology of the *union of balls*

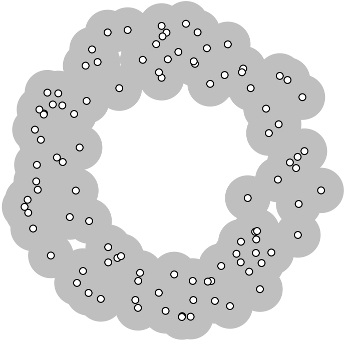
$$M_\epsilon = \bigcup_{x \in X} B_\epsilon(x)$$

Two Issues:

Scale: No natural choice of ϵ!

Conception: How to encode M_ϵ on computer?

Answer: Topology



$r = .6$

Simplicial Complexes

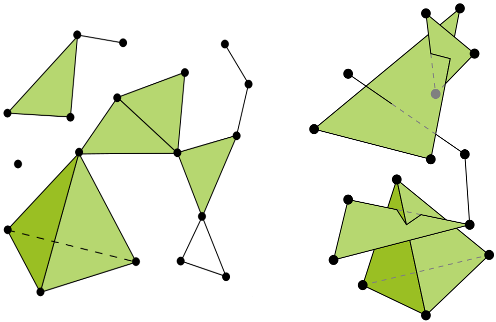
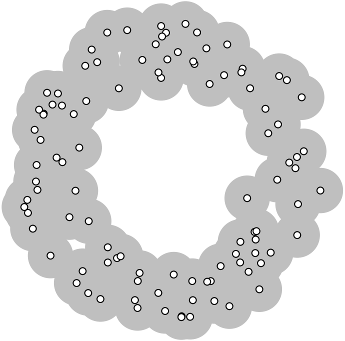


Figure: (left) an example (right) a non example

For a vertex set V a simplicial complex $K \subseteq 2^V$ satisfying:

1. $\{v\} \in K$ for each $v \in V$
2. if $\tau \subseteq \sigma \in K$ then $\tau \in K$.
3. if σ is a k-simplex if $|\sigma| = k + 1$

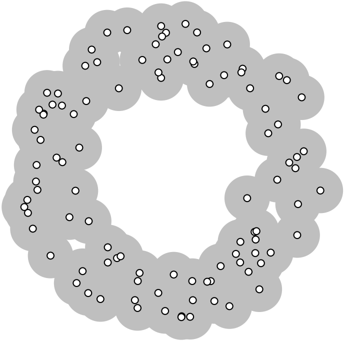
Simplicial Complexes



$\check{C}_{.6}$

The Čech complex $\check{C}_\epsilon$ encodes the intersection pattern of M_ϵ:

Simplicial Complexes

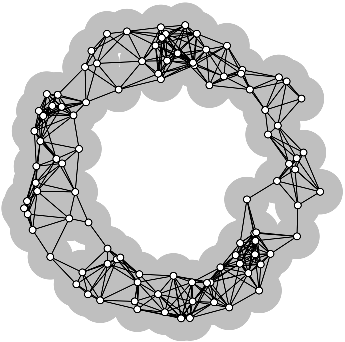


$\check{C}_{.6}$

The Čech complex $\check{C}_\epsilon$ encodes the intersection pattern of M_ϵ: Encode:

- Points as *vertices* (0-cells)

Simplicial Complexes

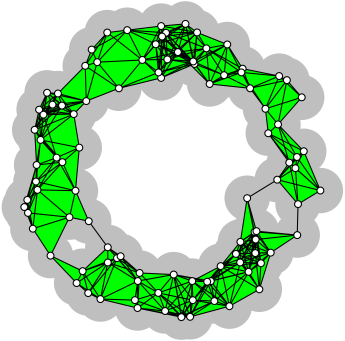


$\check{C}_{.6}$

The Čech complex $\check{C}_\epsilon$ encodes the intersection pattern of M_ϵ: Encode:

- Points as *vertices* (0-cells)
- Pairwise intersections as *edges* (1-cells)

Simplicial Complexes

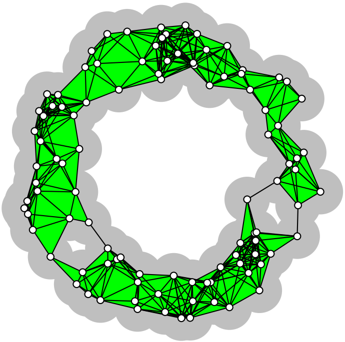


$\check{C}_{.6}$

The Čech complex $\check{C}_\epsilon$ encodes the intersection pattern of M_ϵ: Encode:

- Points as *vertices* (0-cells)
- Pairwise intersections as *edges* (1-cells)
- Threeway intersections as *triangles* (2-cells)

Simplicial Complexes

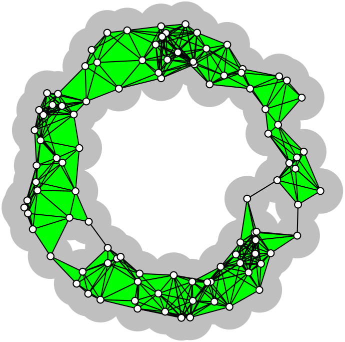


$\check{C}_{.6}$

The Čech complex $\check{C}_\epsilon$ encodes the intersection pattern of M_ϵ: Encode:

- Points as *vertices* (0-cells)
- Pairwise intersections as *edges* (1-cells)
- Threeway intersections as *triangles* (2-cells)
- *k*-way intersections as *(k+1)-cells*

Simplicial Complexes



$\check{C}_{.6}$

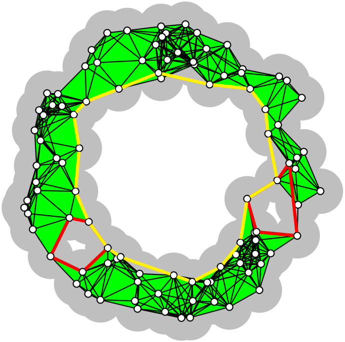
The Čech complex $\check{C}_\epsilon$ encodes the intersection pattern of M_ϵ: Encode:

- Points as *vertices* (0-cells)
- Pairwise intersections as *edges* (1-cells)
- Threeway intersections as *triangles* (2-cells)
- *k*-way intersections as *(k+1)-cells*

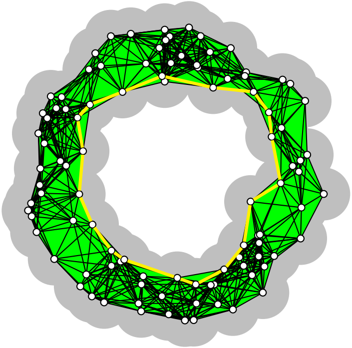
Lemma

Nerve Lemma[Leray '45] $\check{C}_\epsilon$ is topologically equivalent to M_ϵ.

Which ϵ? Don't settle for parameter choices



$\check{C}_{.6}$

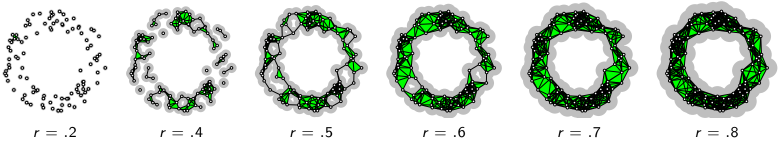


$\check{C}_{.7}$

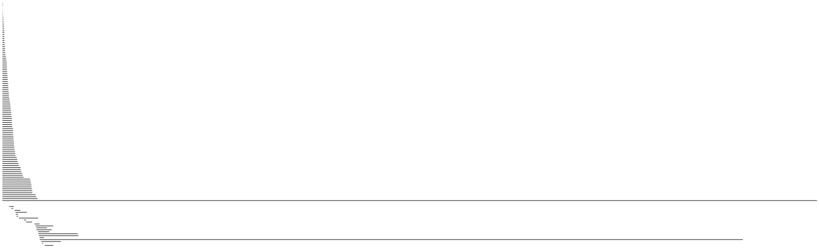
Functorality allows us to systematically track connectivity between scale!

The Persistence Pipeline

Build a sequence of growing spaces from a point sample.

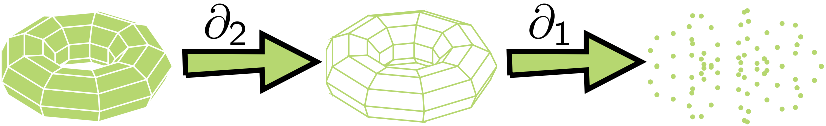


Compute how homology changes as we grow a space.



Results are captured by a *persistence diagram*

Defining Homology

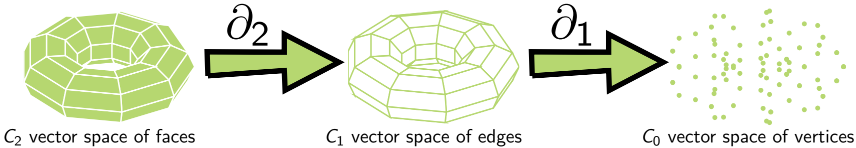


C_2 vector space of faces C_1 vector space of edges C_0 vector space of vertices

Take the linear extension of the boundary operator:

$$\partial_d([v_0, \ldots v_d]) = \sum_{i=0}^{d} (-1)^i [v_0 \ldots, \hat{v}_i, \ldots v_d]$$

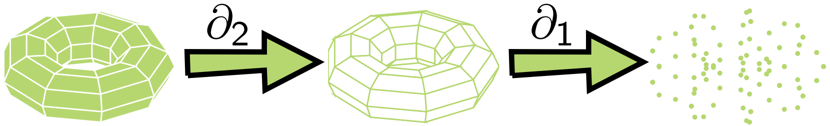
Defining Homology



Take the linear extension of the boundary operator:

$$\partial_d([v_0, \dots v_d]) = \sum_{i=0}^{d} (-1)^i [v_0 \dots, \hat{v}_i, \dots v_d]$$

Defining Homology



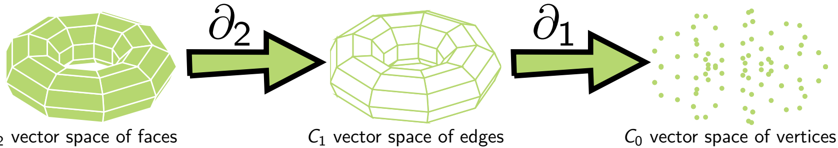
C_2 vector space of faces $\quad$ C_1 vector space of edges $\quad$ C_0 vector space of vertices

Take the linear extension of the boundary operator:

$$\partial_d([v_0, \dots v_d]) = \sum_{i=0}^{d} (-1)^i [v_0 \dots, \hat{v}_i, \dots v_d]$$

Lemma: $\quad \partial_{d-1} \circ \partial_d \equiv 0 \Rightarrow \operatorname{im} \partial_d \subseteq \ker \partial_{d-1}$

Defining Homology



Take the linear extension of the boundary operator:

$$\partial_d([v_0, \dots v_d]) = \sum_{i=0}^{d} (-1)^i [v_0 \dots, \hat{v}_i, \dots v_d]$$

Lemma: $\partial_{d-1} \circ \partial_d \equiv 0 \Rightarrow \operatorname{im} \partial_d \subseteq \ker \partial_{d-1}$

Definition: $H_d(K) = \ker \partial_d / \operatorname{im} \partial_{d+1}$

Homology

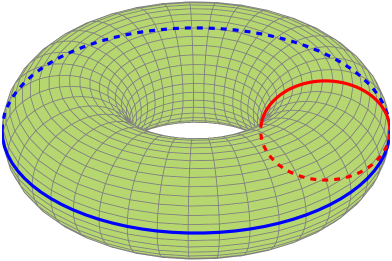


Figure: Homology of the torus T^1

Homology is a module (vector space over a ring)

$\beta_0 = \dim H_0(T^1) = 1$ connected component

Homology

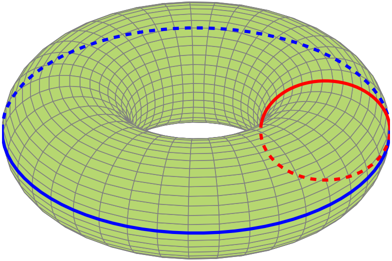


Figure: Homology of the torus T^1

Homology is a module (vector space over a ring)

$\beta_0 = \dim H_0(T^1)$ $= 1$ connected component
$\beta_1 = \dim H_1(T^1)$ $= 2$ loops around the donut hole.

Homology

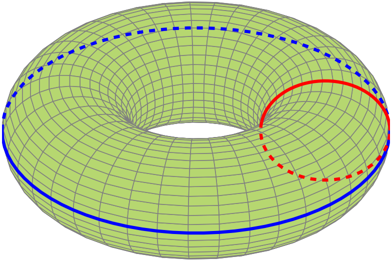
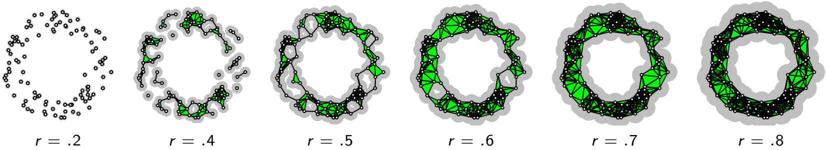


Figure: Homology of the torus T^1

Homology is a module (vector space over a ring)

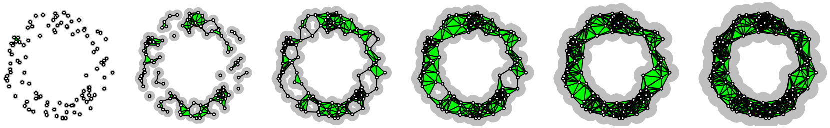
$\beta_0 = \dim H_0(T^1)$ = 1 connected component
$\beta_1 = \dim H_1(T^1)$ = 2 loops around the donut hole.
$\beta_2 = \dim H_2(T^1)$ = 1 void (useful for jelly filling).

Persistent Homology is the Homology of Maps



$$H_d(\check{C}_*) = \bigoplus_{\epsilon} H_d(\check{C}_\epsilon)$$

Persistent Homology is the Homology of Maps

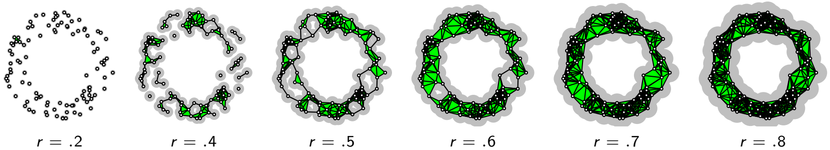


$r = .2$ $r = .4$ $r = .5$ $r = .6$ $r = .7$ $r = .8$

$$H_d(\check{C}_*) = \bigoplus_{\epsilon} H_d(\check{C}_\epsilon)$$

Carlsson and Zomorodian show us that when using field coefficients there is a structure theorem!

Persistent Homology is the Homology of Maps

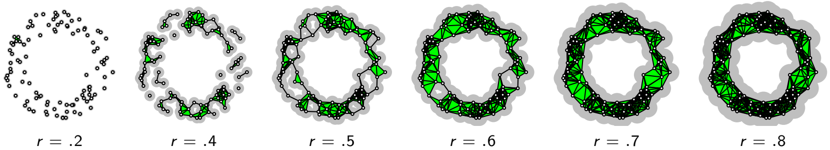


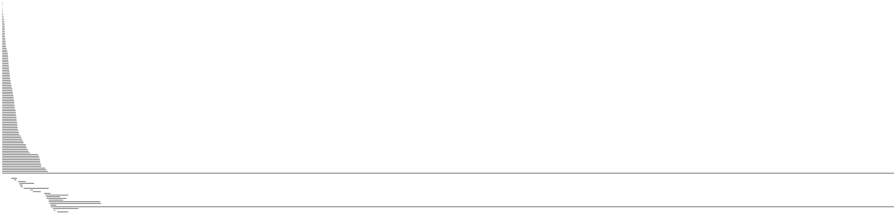
$$H_d(\check{C}_*) = \bigoplus_{\epsilon} H_d(\check{C}_\epsilon)$$

Carlsson and Zomorodian show us that when using field coefficients there is a structure theorem!

$$H_d(\check{C}_*) = \left(\bigoplus_{\alpha} t^{\alpha} k[t]\right) \bigoplus \left(\bigoplus_{[\alpha,\beta)} t^{\alpha} k[t]/t^{\beta}\right)$$

Persistent Homology is the Homology of Maps





$$H_d(\check{C}_*) = \left(\bigoplus_{\alpha} t^{\alpha} k[t]\right) \bigoplus \left(\bigoplus_{[\alpha,\beta)} t^{\alpha} k[t]/t^{\beta}\right)$$

Next: Explain factorization and a parallel persistence algorithm.

The Agenda

- Motivation
- Background
- **Persistent Homology and a Spectral Sequence**
- Homology from subcomplexes: Mayer Vietoris
- Mayer Vietoris & Persistence

Exact Sequences & Linear Algebra

Suppose $A \subset X$ then we have:

$$0 \longrightarrow C_n(A) \xrightarrow{i} C_n(X) \xrightarrow{j} C_n(X/A) \longrightarrow 0$$

This is *exact* meaning that $\ker j = \operatorname{im} i$.

When we compute homology we get a larger exact sequence:

$$\dots \xrightarrow{\delta} H_n(A) \xrightarrow{i} H_n(X) \xrightarrow{j} H_n(X/A) \xrightarrow{\delta} H_{n-1}(A) \xrightarrow{i} \dots$$

where:

$$\delta_n([x]) = [\partial_n(x)]$$

This is the recipe for a basic parallel algorithm.

Exact Sequences & Linear Algebra

We can decompose ∂_X the boundary matrix for X as follows:

$$\begin{bmatrix} \partial_A & \tilde{\delta} \\ 0 & \partial_{X/A} \end{bmatrix}$$

Exact Sequences & Linear Algebra

We can decompose ∂_X the boundary matrix for X as follows:

$$\begin{bmatrix} S_A \cdot P_A & \delta = S_{X/A} \tilde{\delta} D_{X/A} \\ 0 & P_{X/A} \end{bmatrix}$$

Algorithm Analysis

Assume $|A| = n$ and $|X| = m$

Correctness follows from exact sequence.

Algorithm Analysis

Assume $|A| = n$ and $|X| = m$

Correctness follows from exact sequence.

Complexity $O((m-n)^3 + n^3 + (m-n) \cdot m \cdot n)$.

Algorithm Analysis

Assume $|A| = n$ and $|X| = m$

Correctness follows from exact sequence.

Complexity $O((m-n)^3 + n^3 + (m-n) \cdot m \cdot n)$.

Speedup This is minimized when $n = \frac{m}{2}$ and at that point we have a speedup of 2.

Algorithm Analysis

Assume $|A| = n$ and $|X| = m$

Correctness follows from exact sequence.

Complexity $O((m-n)^3 + n^3 + (m-n) \cdot m \cdot n)$.

Speedup This is minimized when $n = \frac{m}{2}$ and at that point we have a speedup of 2.

How to correctly extend this to a sequence of subspaces

$$K_0 \subseteq K_1 \subseteq K_2 \subset \ldots \subset K_n \quad ?$$

Homology from a filtration

How to correctly extend this to a sequence of subspaces

$$K_0 \subseteq K_1 \subseteq K_2 \subset \ldots \subset K_n \quad ?$$

We can decompose ∂_X the boundary matrix X in a similar way:

$$\begin{matrix} \partial_0 & \tilde{d}_0 & \tilde{d}_0 & \cdots & \tilde{d}_{n,n} \\ 0 & \partial_1 & \tilde{d}_0 & \cdots & \tilde{d}_{n-1,n} \\ 0 & 0 & \partial_2 & \cdots & \tilde{d}_{\ldots,n} \\ 0 & 0 & 0 & \cdots & \tilde{d}_{0,n} \\ 0 & 0 & 0 & 0 & \partial_n \end{matrix}$$

Homology from a filtration

How to correctly extend this to a sequence of subspaces

$$K_0 \subseteq K_1 \subseteq K_2 \subset \ldots \subset K_n \quad ?$$

We can decompose ∂_X the boundary matrix X in a similar way:

$$\begin{bmatrix} \partial_0 & \tilde{d}_0 & \tilde{d}_0 & \cdots & \tilde{d}_{n,n} \\ 0 & \partial_1 & \tilde{d}_0 & \cdots & \tilde{d}_{n-1,n} \\ 0 & 0 & \partial_2 & \cdots & \tilde{d}_{\ldots,n} \\ 0 & 0 & 0 & \cdots & \tilde{d}_{0,n} \\ 0 & 0 & 0 & 0 & \partial_n \end{bmatrix}$$

Homology from a filtration

How to correctly extend this to a sequence of subspaces

$$K_0 \subseteq K_1 \subseteq K_2 \subset \ldots \subset K_n \quad ?$$

We can decompose ∂_X the boundary matrix X in a similar way:

$$\begin{matrix} \partial_0 & \tilde{d}_0 & \tilde{d}_0 & \cdots & \tilde{d}_{n,n} \\ 0 & \partial_1 & \tilde{d}_0 & \cdots & \tilde{d}_{n-1,n} \\ 0 & 0 & \partial_2 & \cdots & \tilde{d}_{\ldots,n} \\ 0 & 0 & 0 & \cdots & \tilde{d}_{0,n} \\ 0 & 0 & 0 & 0 & \partial_n \end{matrix}$$

Homology from a filtration

How to correctly extend this to a sequence of subspaces

$$K_0 \subseteq K_1 \subseteq K_2 \subset \ldots \subset K_n \quad ?$$

We can decompose ∂_X the boundary matrix X in a similar way:

$$\begin{array}{ccccc}
\partial_0 & \tilde{d}_0 & \tilde{d}_0 & \cdots & \tilde{d}_{n,n} \\
0 & \partial_1 & \tilde{d}_0 & \cdots & \tilde{d}_{n-1,n} \\
0 & 0 & \partial_2 & \cdots & \tilde{d}_{\ldots,n} \\
0 & 0 & 0 & \cdots & \tilde{d}_{0,n} \\
0 & 0 & 0 & 0 & \partial_n
\end{array}$$

Homology from a filtration

How to correctly extend this to a sequence of subspaces

$$K_0 \subseteq K_1 \subseteq K_2 \subset \ldots \subset K_n \quad ?$$

We can decompose ∂_X the boundary matrix X in a similar way:

$$\begin{bmatrix} \partial_0 & \tilde{d}_0 & \tilde{d}_0 & \cdots & \tilde{d}_{n,n} \\ 0 & \partial_1 & \tilde{d}_0 & \cdots & \tilde{d}_{n-1,n} \\ 0 & 0 & \partial_2 & \cdots & \tilde{d}_{\ldots,n} \\ 0 & 0 & 0 & \cdots & \tilde{d}_{0,n} \\ 0 & 0 & 0 & 0 & \partial_n \end{bmatrix}$$

Homology from a filtration

How to correctly extend this to a sequence of subspaces

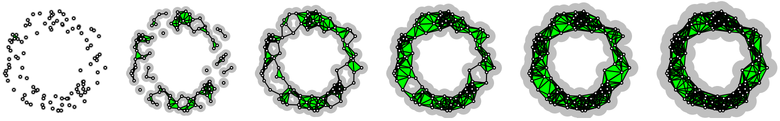
$$K_0 \subseteq K_1 \subseteq K_2 \subset \ldots \subset K_n \quad ?$$

We can decompose ∂_X the boundary matrix X in a similar way:

$$\begin{bmatrix} \partial_0 & \tilde{d}_0 & \tilde{d}_0 & \cdots & \tilde{d}_{n,n} \\ 0 & \partial_1 & \tilde{d}_0 & \cdots & \tilde{d}_{n-1,n} \\ 0 & 0 & \partial_2 & \cdots & \tilde{d}_{\ldots,n} \\ 0 & 0 & 0 & \cdots & \tilde{d}_{0,n} \\ 0 & 0 & 0 & 0 & \partial_n \end{bmatrix}$$

Question: How do you show correctness?

Spectral Sequence of A Filtration

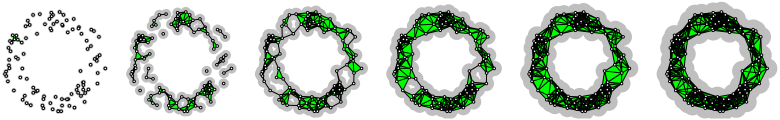


$r = .2$ $r = .4$ $r = .5$ $r = .6$ $r = .7$ $r = .8$

From our sequence of spaces, we have a filtration of chain complexes:

$$0 \xrightarrow{i} C_*(K_0) \xrightarrow{i} C_*(K_1) \xrightarrow{i} \ldots \xrightarrow{i} C_*(K_{n-1}) \xrightarrow{i} C_*(K_n)$$

Spectral Sequence of A Filtration

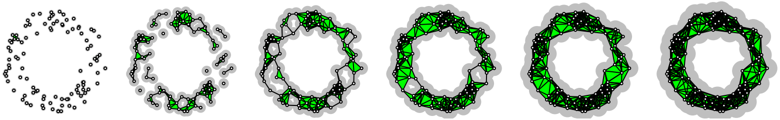


$r = .2$ $r = .4$ $r = .5$ $r = .6$ $r = .7$ $r = .8$

Attach the co-kernel to get short exact sequences

$$\begin{tikzcd}
0 \arrow[r, "i"] & C_*(K_0) \arrow[r, "i"] \arrow[d, "j"] & C_*(K_1) \arrow[r, "i"] \arrow[d, "j"] & \dots \arrow[r, "i"] & C_*(K_{n-1}) \arrow[r, "i"] \arrow[d, "j"] & C_*(K_n) \arrow[d, "j"] \\
& C_*(K_0, 0) & C_*(K_1, K_0) & \dots & C_*(K_{n-1}, K_{n-2}) & C_*(K_n, K_{n-1})
\end{tikzcd}$$

Spectral Sequence of A Filtration



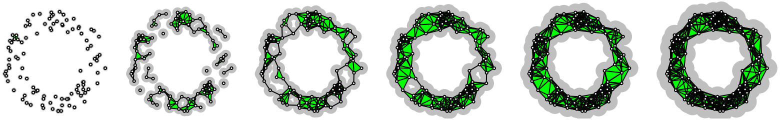
$r = .2$ $r = .4$ $r = .5$ $r = .6$ $r = .7$ $r = .8$

Pass to homology, where triangles are exact.

$$
\begin{tikzcd}
0 \arrow[r, "i"] & H(K_0) \arrow[r, "i"] \arrow[d, "j"] & H(K_1) \arrow[r, "i"] \arrow[d, "j"] & \dots \arrow[r, "i"] & H(K_{n-1}) \arrow[r, "i"] \arrow[d, "j"] & H(K_n) \arrow[d, "j"] \\
& H(K_0) & H(K_1, K_0) \arrow[ul, "\delta"] & \dots \arrow[ul, "\delta"] & H(K_{n-1}, K_{n-2}) \arrow[ul, "\delta"] & H(K_n, K_{n-1}) \arrow[ul, "\delta"]
\end{tikzcd}
$$

$$E^2 = H_*(d_1)$$

Spectral Sequence of A Filtration

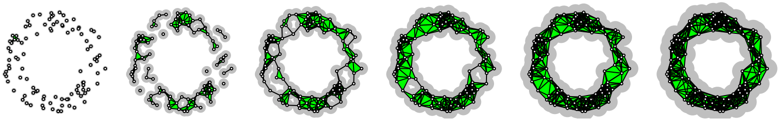


Pass to homology, where triangles are exact.

$$
\begin{tikzcd}
0 \arrow[r, "i"] & H(K_0) \arrow[r, "i"] \arrow[d, "j"] & H(K_1) \arrow[r, "i"] \arrow[d, "j"] & \dots \arrow[r, "i"] & H(K_{n-1}) \arrow[r, "i"] \arrow[d, "j"] & H(K_n) \arrow[d, "j"] \\
& H(K_0) & H(K_1, K_0) \arrow[l, "d_1"] \arrow[ul, "\delta"] & \dots \arrow[l, "d_1"] \arrow[ul, "\delta"] & H(K_{n-1}, K_{n-2}) \arrow[l, "d_1"] \arrow[ul, "\delta"] & H(K_n, K_{n-1}) \arrow[l, "d_1"] \arrow[ul, "\delta"]
\end{tikzcd}
$$

$$E^2 = H_*(d_1)$$

Spectral Sequence of A Filtration



Build a new chain complex by constructing $d_1 = j \circ \delta$

$$\begin{tikzcd}
0 \arrow[r, "i"] & H(K_0) \arrow[r, "i"] \arrow[d, "j"] & H(K_1) \arrow[r, "i"] \arrow[d, "j"] & \dots \arrow[r, "i"] & H(K_{n-1}) \arrow[r, "i"] \arrow[d, "j"] & H(K_n) \arrow[d, "j"] \\
& E_0^2 & E_1^2 \arrow[l, "0"] \arrow[ul, "\delta"'] & \dots \arrow[l, "0"] \arrow[ul, "\delta"'] \arrow[ll, bend left, "d_2"] & E_{n-1}^1 \arrow[l, "0"] \arrow[ul, "\delta"'] \arrow[ll, bend left, "d_2"] & E_n \arrow[l, "0"] \arrow[ul, "\delta"'] \arrow[ll, bend left, "d_2"]
\end{tikzcd}$$

More generally:

$$d_{r,p} : E_{r+p} \to E_p$$

Eventually $d_{n,p} = 0$, when sequence has converged.

Spectral Sequence of A Filtration

Build a new chain complex by constructing $d_1 = j \circ \delta$

$$
\begin{tikzcd}
0 \arrow[r, "i"] & H(K_0) \arrow[r, "i"] \arrow[d, "j"] & H(K_1) \arrow[r, "i"] \arrow[d, "j"] & \dots \arrow[r, "i"] & H(K_{n-1}) \arrow[r, "i"] \arrow[d, "j"] & H(K_n) \arrow[d, "j"] \\
 & E_0^2 & E_1^2 \arrow[l, "0"] \arrow[ul, "\delta"'] & \dots \arrow[l, "0"] \arrow[ul, "\delta"'] \arrow[ll, bend left, "d_2"] & E^1_{n-1} \arrow[l, "0"] \arrow[ul, "\delta"'] \arrow[ll, bend left, "d_2"] & E_n \arrow[l, "0"] \arrow[ul, "\delta"'] \arrow[ll, bend left, "d_2"]
\end{tikzcd}
$$

More generally:

$$d_{r,p} : E_{r+p} \to E_p$$

Eventually $d_{n,p} = 0$, when sequence has converged.

Spectral Sequence of A Filtration

$$\begin{tikzcd}
0 \arrow[r, "i"] & H(K_0) \arrow[r, "i"] \arrow[d, "j"] & H(K_1) \arrow[r, "i"] \arrow[d, "j"] & \dots \arrow[r, "i"] & H(K_{n-1}) \arrow[r, "i"] \arrow[d, "j"] & H(K_n) \arrow[d, "j"] \\
& E_0^\infty & E_1^\infty \arrow[l, "0"] \arrow[ul, "\delta"] & \dots \arrow[l, "0"] \arrow[ul, "\delta"] & E_{n-1}^\infty \arrow[l, "0"] \arrow[ul, "\delta"] & E_n^\infty \arrow[l, "0"] \arrow[ul, "\delta"]
\end{tikzcd}$$

The data on the final page is called the E^∞ page.

Theorem

Free Lunch

$$PH_*(K) \cong \bigoplus_{r,p} t^p \operatorname{Im}(d_{r,p})/t^{p+r} \bigoplus_{p} t^p E_p^\infty$$

Remarks

- Clear connection to persistent homology

Remarks

- Clear connection to persistent homology
- Easy to implement

Remarks

- Clear connection to persistent homology
- Easy to implement
- *Doesn't load balance well* (lose a useful thread at each step)

Remarks

- Clear connection to persistent homology
- Easy to implement
- *Doesn't load balance well* (lose a useful thread at each step)
- Data on each processor eventually coupled to every other processor.

Remarks

- Clear connection to persistent homology
- Easy to implement
- *Doesn't load balance well* (lose a useful thread at each step)
- Data on each processor eventually coupled to every other processor.

Seek an algorithm which works on large parts of the space in parallel, with small intersections

The Agenda

- Motivation
- Background
- Persistent Homology and a Spectral Sequence
- **Homology from subcomplexes: Mayer Vietoris**
- Mayer Vietoris & Persistence

Mayer Vietoris

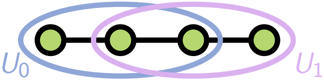


Figure: Space and Cover

When we have a pair of sets there is another short exact sequence:

$$C_*(U_0 \cap U_1) \xrightarrow{x \mapsto (x,x)} C_*(U_0) \bigoplus C_*(U_1) \xrightarrow{(x,y) \mapsto x-y} C_*(U_0 \bigcup U_1) = C_*(X)$$

When you have intersections of 3 or more sets, this becomes the mayer vietoris spectral sequence, with the following initial data:

$$E^0_{p,q} = \langle p\text{-chains in a } q\text{-way intersection} \rangle$$

Mayer Vietoris

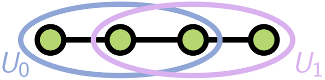


Figure: Space and Cover

When we have a pair of sets there is another short exact sequence:

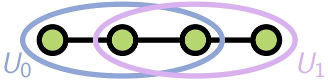
$$C_*(U_0 \cap U_1) \xrightarrow{x \mapsto (x,x)} C_*(U_0) \bigoplus C_*(U_1) \xrightarrow{(x,y) \mapsto x-y} C_*(U_0 \bigcup U_1) = C_*(X)$$

When you have intersections of 3 or more sets, this becomes the mayer vietoris spectral sequence, with the following initial data:

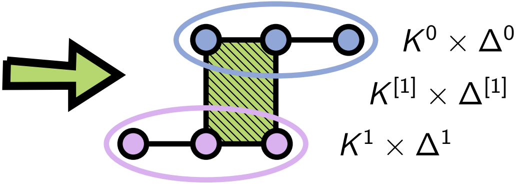
$$E^0_{p,q} = \langle p\text{-chains in a } q\text{-way intersection} \rangle$$

We will come back to this. For now, lets take advantage of this covering.

Compute homology via subcomplexes



Space and Cover



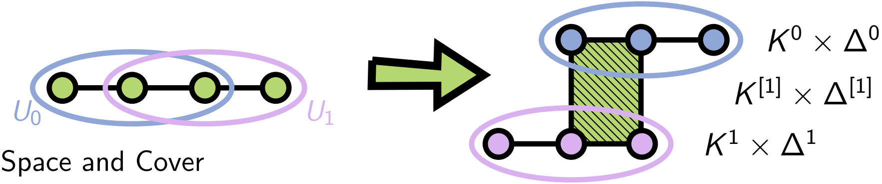
The blowup complex.

Definition:

$$K^U = \bigcup_{J \in N(U)} U_J \times J$$

where $U_J = \bigcap_{j \in J} U_j$ and $N(U)$ is the nerve of U.

Compute homology via subcomplexes



Space and Cover

The blowup complex.

Definition:

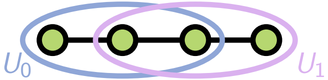
$$K^U = \bigcup_{J \in N(U)} U_J \times J$$

where $U_J = \bigcap_{j \in J} U_j$ and $N(U)$ is the nerve of U.

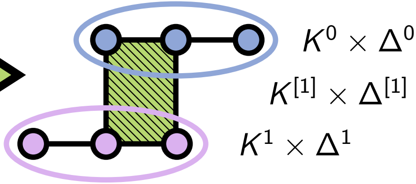
Theorem (Segal)

For any closed cover U, K^U and K are homotopy equivalent.

Compute homology via subcomplexes



Space and Cover



The blowup complex.

Definition:

$$K^U = \bigcup_{J \in N(U)} U_J \times J$$

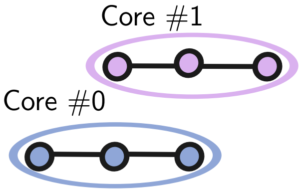
where $U_J = \bigcap_{j \in J} U_j$ and $N(U)$ is the nerve of U.

Theorem (Segal)

For any closed cover U, K^U and K are homotopy equivalent.

Recipe for a parallel *homology* algorithm

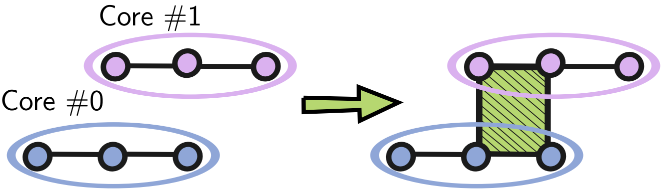
Parallel Algorithm



Algorithm:

Step 1: Compute the homology of each local piece (in parallel)

Parallel Algorithm

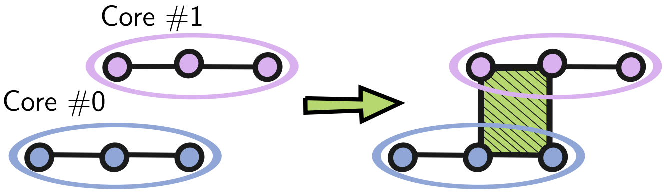


Algorithm:

Step 1: Compute the homology of each local piece (in parallel)

Step 2: Compute the homology with the blowup pieces (in serial)

Parallel Algorithm



Algorithm:

Step 1: Compute the homology of each local piece (in parallel)

Step 2: Compute the homology with the blowup pieces (in serial)

Goal:

- Do most of work Step 1, Do as little as possible in Step 2.
- Equivalent to *quickly* generating "balanced" and "minimal" covers.

Finding Minimum Blowups

Goal: Quickly generate cover, U, with balanced size and "minimal" overlap.

Finding Minimum Blowups

Goal: Quickly generate cover, U, with balanced size and "minimal" overlap.

Formally: Given $\alpha \in (\frac{1}{n}, 1)$ find a covering $U = \{U_1, \ldots, U_n\}$ minimizing:

$$\frac{|K^U|}{|K|}$$

Finding Minimum Blowups

Goal: Quickly generate cover, U, with balanced size and "minimal" overlap.

Formally: Given $\alpha \in (\frac{1}{n}, 1)$ find a covering $U = \{U_1, \ldots, U_n\}$ minimizing:

$$\frac{|K^U|}{|K|}$$

subject to:

$$\max_{i \in [n-1]} |U_i| \leq \alpha |K|$$

Finding Minimum Blowups

Goal: Quickly generate cover, U, with balanced size and "minimal" overlap.

Formally: Given $\alpha \in (\frac{1}{n}, 1)$ find a covering $U = \{U_1, \ldots, U_n\}$ minimizing:

$$\frac{|K^U|}{|K|}$$

subject to:

$$\max_{i \in [n-1]} |U_i| \leq \alpha |K|$$

Finding Minimum Blowups

Bad News:

- $\frac{|K^U|}{|K|}$ is *at worst exponential* in $|U|$

Finding Minimum Blowups

Bad News:

- $\frac{|K^U|}{|K|}$ is *at worst exponential* in $|U|$
- We prove finding α-balanced Minimal Blowups is NP-Hard.

Finding Minimum Blowups

Bad News:

- $\frac{|K^U|}{|K|}$ is *at worst exponential* in $|U|$
- We prove finding α-balanced Minimal Blowups is NP-Hard.

Good News:

- Any cover gives us correct results.

Finding Minimum Blowups

Bad News:

- $\frac{|K^U|}{|K|}$ is *at worst exponential* in $|U|$
- We prove finding α-balanced Minimal Blowups is NP-Hard.

Good News:

- Any cover gives us correct results.
- Approximate minimal blowups should still parallelize well.

Finding Minimum Blowups

Bad News:

- $\frac{|K^U|}{|K|}$ is *at worst exponential* in $|U|$
- We prove finding α-balanced Minimal Blowups is NP-Hard.

Good News:

- Any cover gives us correct results.
- Approximate minimal blowups should still parallelize well.
- We provide a heuristic algorithm with:

Finding Minimum Blowups

Bad News:

- $\frac{|K^U|}{|K|}$ is *at worst exponential* in $|U|$
- We prove finding α-balanced Minimal Blowups is NP-Hard.

Good News:

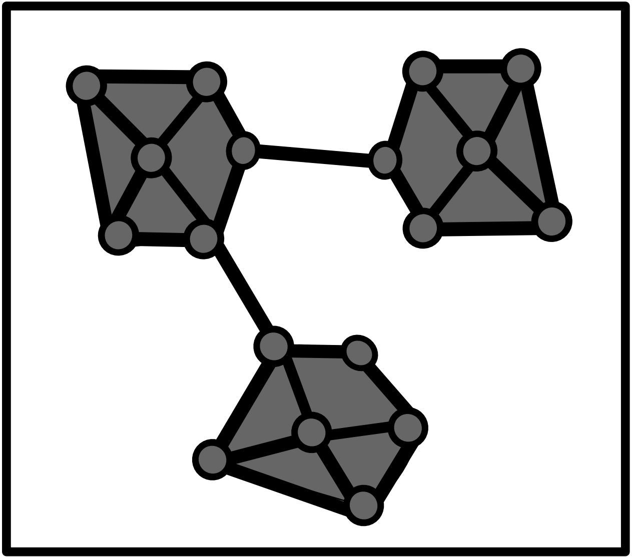
- Any cover gives us correct results.
- Approximate minimal blowups should still parallelize well.
- We provide a heuristic algorithm with:

$$|K^{\mathbf{C}}|/|K| < 3$$

guaranteed!

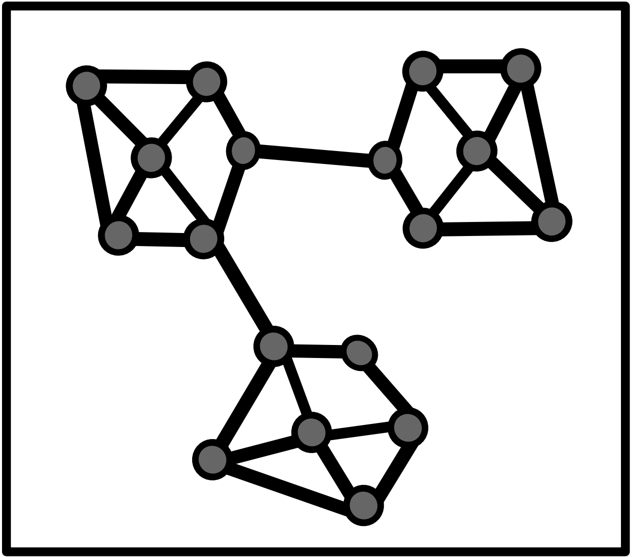
Cover Generation Algorithm

- **Input:** Simplicial Complex K, and number of cover sets n
- **Output:** Covering U into n cover sets.



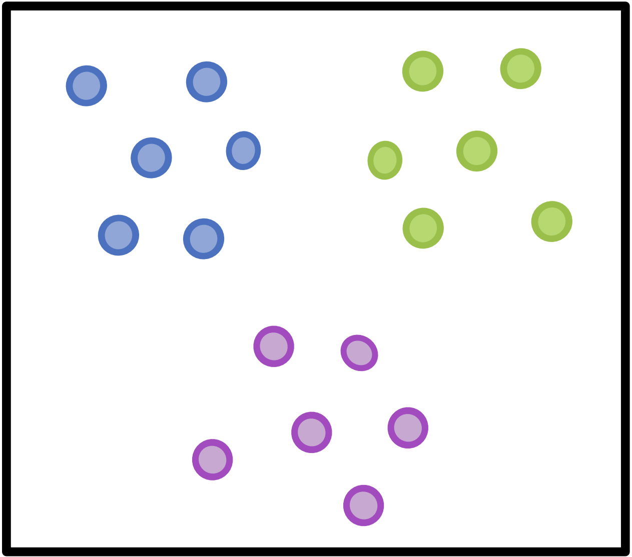
Cover Generation Algorithm

- **Input:** Simplicial Complex K, and number of cover sets n
- **Output:** Covering U into n cover sets.



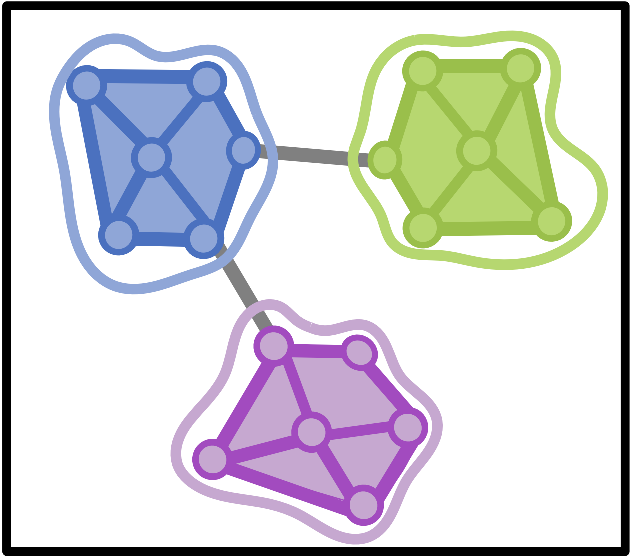
Cover Generation Algorithm

- **Input:** Simplicial Complex K, and number of cover sets n
- **Output:** Covering U into n cover sets.



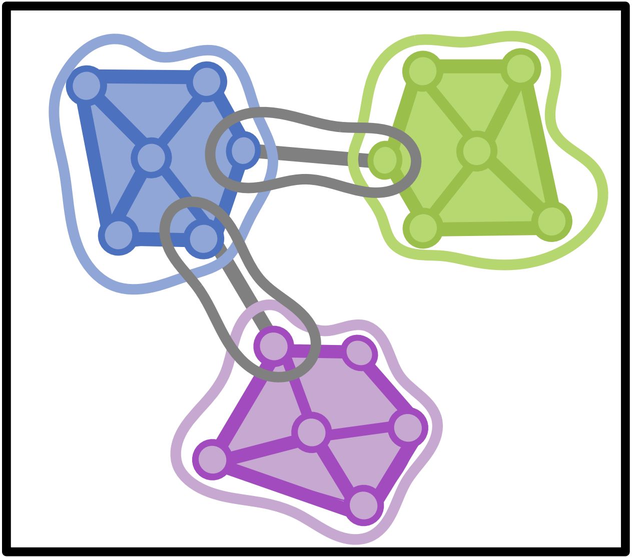
Cover Generation Algorithm

- **Input:** Simplicial Complex K, and number of cover sets n
- **Output:** Covering U into n cover sets.



Cover Generation Algorithm

- **Input:** Simplicial Complex K, and number of cover sets n
- **Output:** Covering U into n cover sets.



Input Datasets

- Connected Cliques (M) 46×10^6 Dim: 10
- Clique (C) 1×10^6 Dim: 19
- Stanford Bunny (B) 9.7×10^6 Dim: 3
- Sphere (S) 19×10^6 Dim: 8
- Erdős-Rényi Random (G) 3.6×10^6 Dim: 4

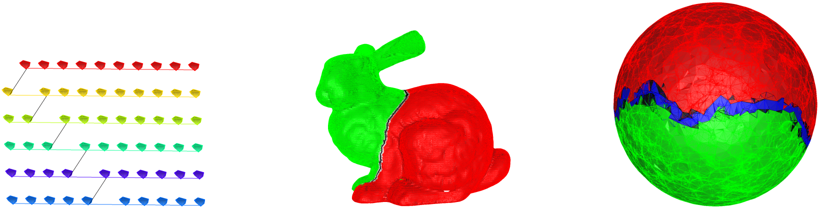


Figure: Visualizations of M, B, S

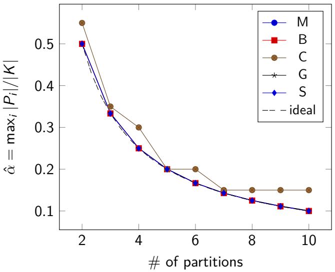


Figure: Partition Balance Ratio $\hat{\alpha}$

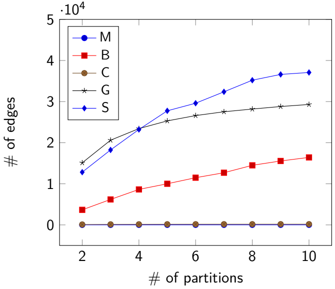


Figure: Edgecut

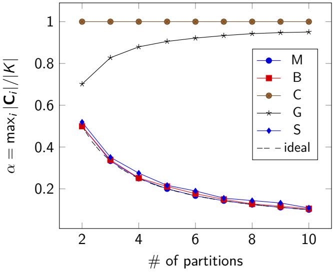


Figure: Balance Ratio for **C**

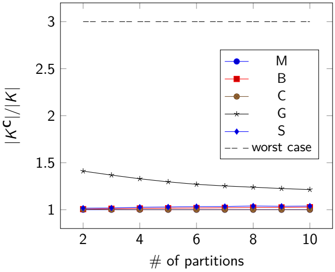


Figure: Blowup Factor

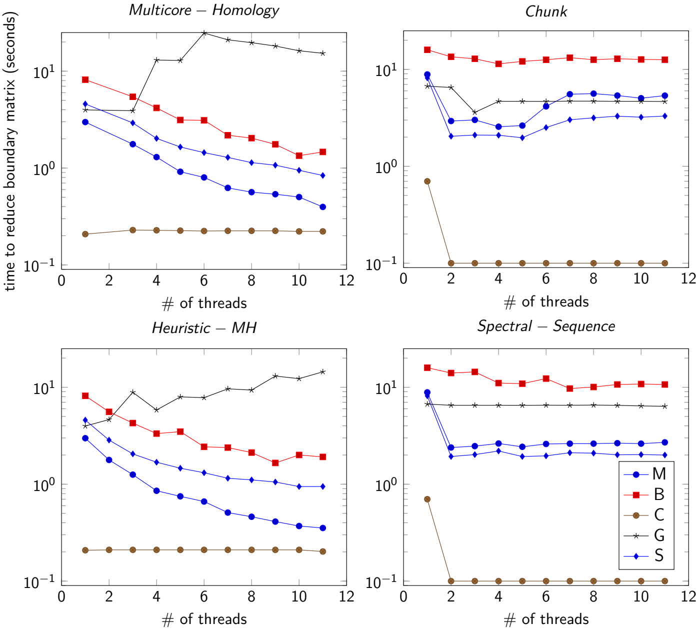


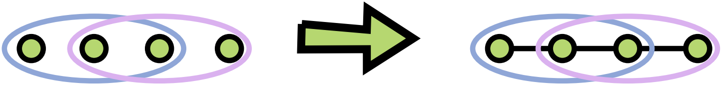
Figure: Time to reduce the boundary matrix for each algorithm.

The Agenda

- Motivation
- Background
- Persistent Homology and a Spectral Sequence
- Homology from subcomplexes: Mayer Vietoris
- **Mayer Vietoris & Persistence**

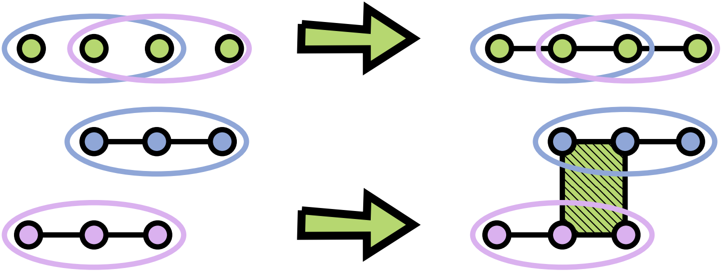
Compute *persistent* homology via subcomplexes

So far computes homology, but, not persistent homology.



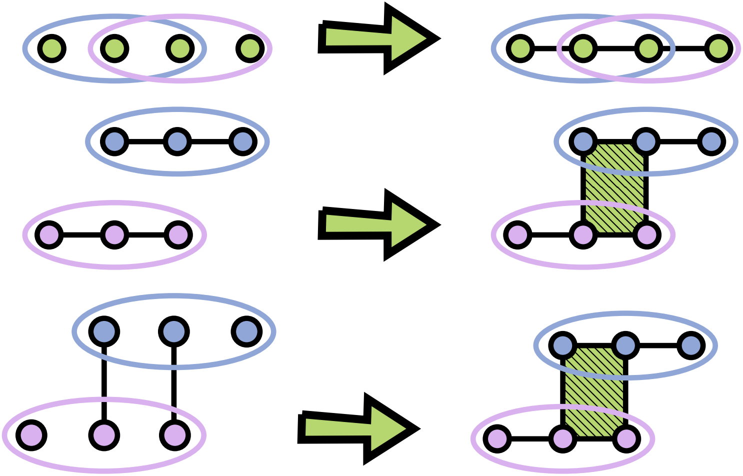
Compute *persistent* homology via subcomplexes

So far computes homology, but, not persistent homology.



Compute *persistent* homology via subcomplexes

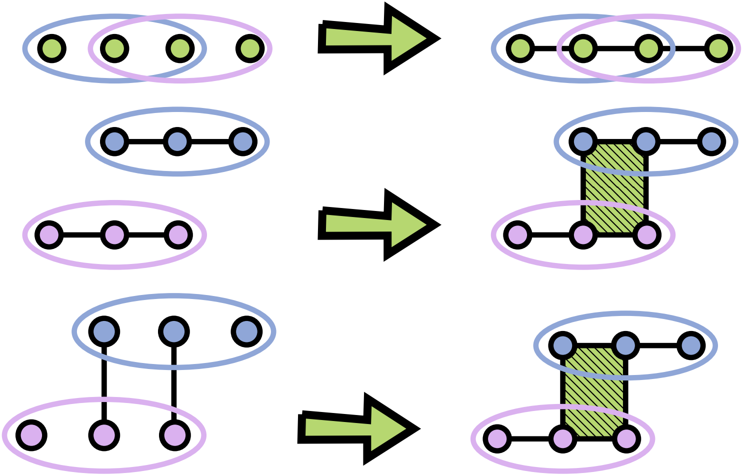
So far computes homology, but, not persistent homology.



Issue: Top filtration unrelated to a possible input filtration.

Compute *persistent* homology via subcomplexes

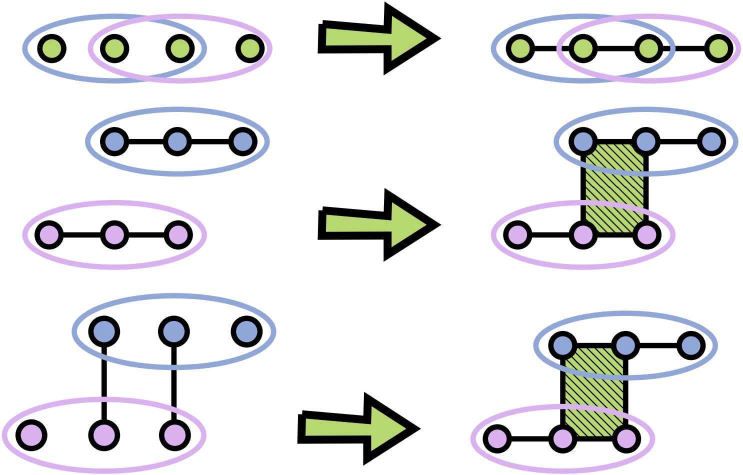
So far computes homology, but, not persistent homology.



Issue: Top filtration unrelated to a possible input filtration.
Issue: Direct use of bottom filtration is not parallel.

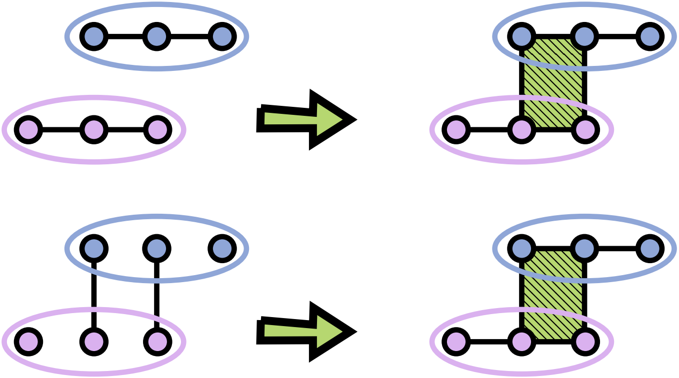
Compute *persistent* homology via subcomplexes

So far computes homology, but, not persistent homology.



Issue: Top filtration unrelated to a possible input filtration.
Issue: Direct use of bottom filtration is not parallel.
Luck: Two filtrations are *related*.

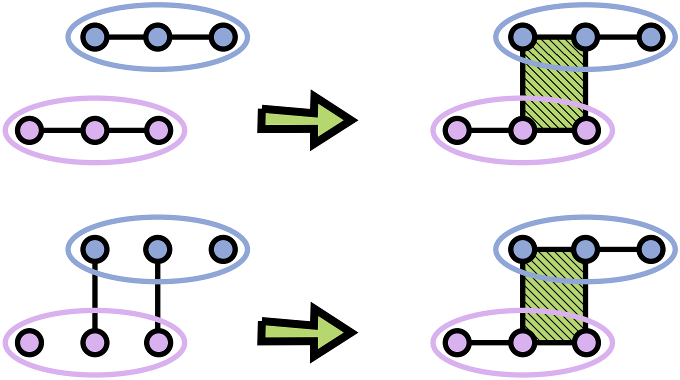
Filtrations



Recall: K^U has cells of the form $\sigma \times \tau$ with $\sigma \in K$ and $\tau \in N$

Difference is about ordering on cells: $\sigma \times \tau < \sigma' \times \tau'$

Filtrations

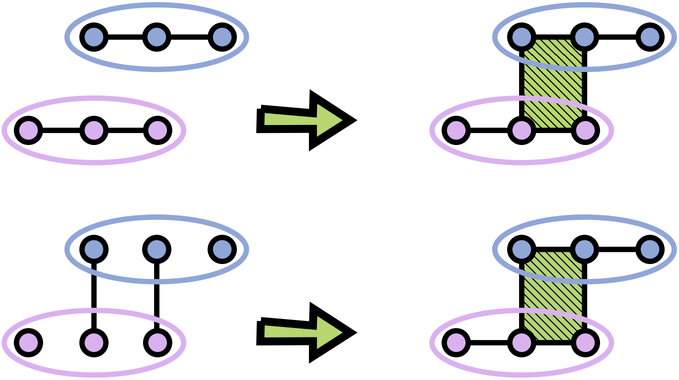


Recall: K^U has cells of the form $\sigma \times \tau$ with $\sigma \in K$ and $\tau \in N$

Difference is about ordering on cells: $\sigma \times \tau < \sigma' \times \tau'$

Picture 1: $\tau < \tau'$ then breaking ties by comparing $\sigma < \sigma'$.

Filtrations



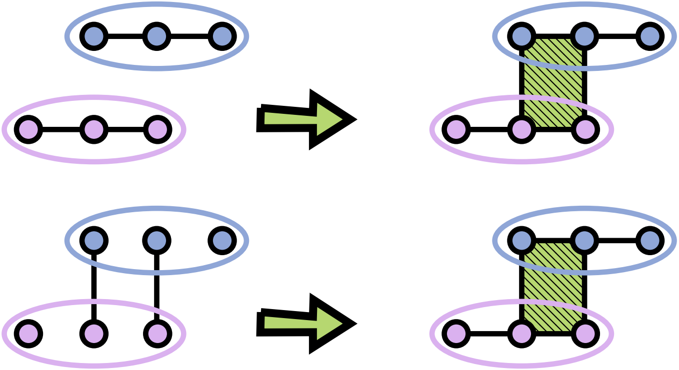
Recall: K^U has cells of the form $\sigma \times \tau$ with $\sigma \in K$ and $\tau \in N$

Difference is about ordering on cells: $\sigma \times \tau < \sigma' \times \tau'$

Picture 1: $\tau < \tau'$ then breaking ties by comparing $\sigma < \sigma'$.

Picture 2: $\sigma < \sigma'$ then breaking ties by comparing $\tau < \tau'$

Filtrations



Recall: K^U has cells of the form $\sigma \times \tau$ with $\sigma \in K$ and $\tau \in N$

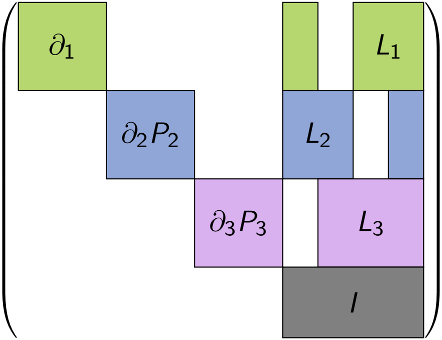
Difference is about ordering on cells: $\sigma \times \tau < \sigma' \times \tau'$

Picture 1: $\tau < \tau'$ then breaking ties by comparing $\sigma < \sigma'$.

Picture 2: $\sigma < \sigma'$ then breaking ties by comparing $\tau < \tau'$

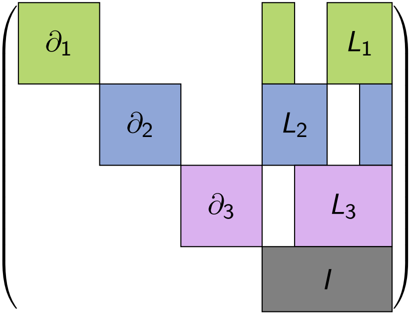
Important Observation: If $\tau = \tau'$ then orders agree!

revisit the blowup



1. ∂_{K^U} in block form, according to incorrect filtration.
2. Reducing the matrix $\Pi' \cdot \partial_{K^U} \cdot \Pi$, where Π permutes between filtrations, results in the correct persistent homology.

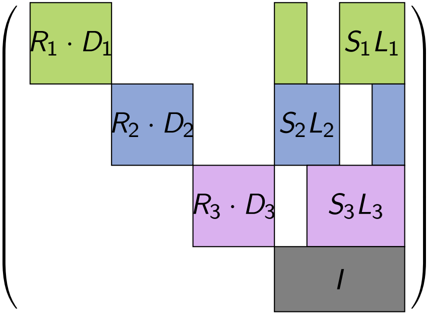
derive persistence with mayer vietoris.



All processors execute these operations with no communication!

Step 1 Reduce all blocks (except I) of a fixed color independently.
$\partial_i = R_i \cdot D_i$

derive persistence with mayer vietoris.

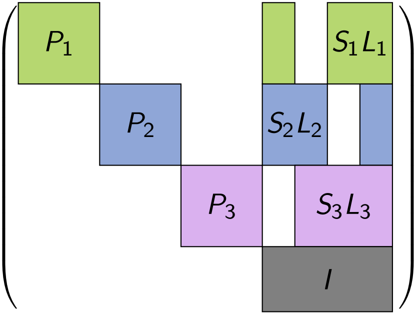


All processors execute these operations with no communication!

Step 1 Reduce all blocks (except I) of a fixed color independently.
$\partial_i = R_i \cdot D_i$

Step 2 Row reduce disjoint union $S^{-1}[R_i \mid L_i] \to [P_i \mid \tilde{L}_i]$

derive persistence with mayer vietoris.



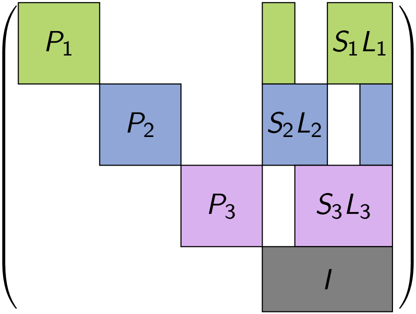
All processors execute these operations with no communication!

Step 1 Reduce all blocks (except I) of a fixed color independently.
$\partial_i = R_i \cdot D_i$

Step 2 Row reduce disjoint union $S^{-1}[R_i \mid L_i] \to [P_i \mid \tilde{L}_i]$

Step 3 Perform all **valid** columns adds from P_i into $\tilde{L}_i$ (perfect parallelism).

derive persistence with mayer vietoris.



All processors execute these operations with no communication!

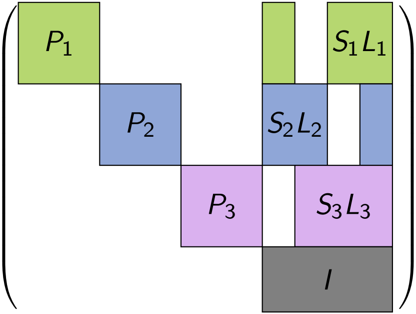
Step 1 Reduce all blocks (except I) of a fixed color independently.
$\partial_i = R_i \cdot D_i$

Step 2 Row reduce disjoint union $S^{-1}[R_i \mid L_i] \to [P_i \mid \tilde{L}_i]$

Step 3 Perform all **valid** columns adds from P_i into $\tilde{L}_i$ (perfect parallelism).

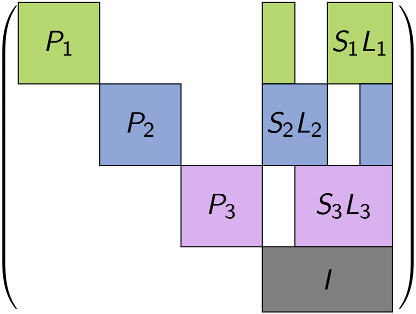
Sofar Not finished yet, but, have not done anything wrong.

derive persistence with mayer vietoris.



Sparsity: Now each P_i has at most 1 nonzero per column.

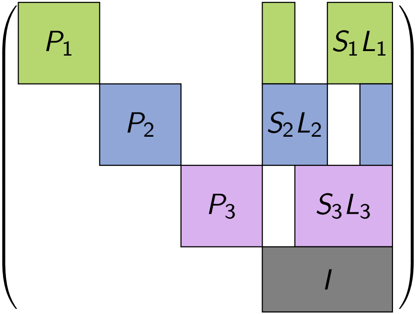
derive persistence with mayer vietoris.



Sparsity: Now each P_i has at most 1 nonzero per column.

Fill in: If $D.U.$ has size m and total size is $m+n$ then fill in is at most $O(mn)$ down from $O((m+n)^2)$

derive persistence with mayer vietoris.

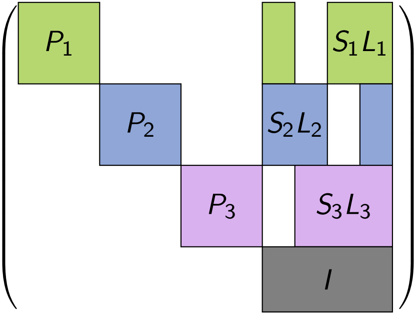


Sparsity: Now each P_i has at most 1 nonzero per column.

Fill in: If $D.U.$ has size m and total size is $m+n$ then fill in is at most $O(mn)$ down from $O((m+n)^2)$

Issue: After permuting, naive reduction, could fill in entire matrix.

derive persistence with mayer vietoris.



Sparsity: Now each P_i has at most 1 nonzero per column.

Fill in: If $D.U.$ has size m and total size is $m+n$ then fill in is at most $O(mn)$ down from $O((m+n)^2)$

Issue: After permuting, naive reduction, could fill in entire matrix.

Solution: Row operations to the rescue!

Cascade

All columns with pivot in row m (after Π):

$$\begin{pmatrix}
\times & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \times & \times & 0 & 0 & 0 & 0 & 0 & 0 \\
\times & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \times & \times & 0 & 0 & 0 & 0 & 0 & 0 \\
\times & 0 & \times & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \times & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\times & 0 & \times & 0 & 0 & 0 & 0 & 0 & 0 \\
\times & 0 & \times & 0 & 0 & 0 & 0 & 0 & 0 \\
\times & \times & \times & \times & \times & \times & \times & \times & \times
\end{pmatrix}$$

Cascade

All columns with pivot in row m (after Π):

$$\begin{pmatrix} \times & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \times & \times & 0 & 0 & 0 & 0 & 0 & 0 \\ \times & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \times & \times & 0 & 0 & 0 & 0 & 0 & 0 \\ \times & 0 & \times & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \times & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \times & 0 & \times & 0 & 0 & 0 & 0 & 0 & 0 \\ \times & 0 & \times & 0 & 0 & 0 & 0 & 0 & 0 \\ \times & \times & \times & \times & \times & \times & \times & \times & \times \end{pmatrix} \rightarrow \begin{pmatrix} \times & \times & \times & \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & 0 & 0 & 0 & 0 & 0 & 0 \\ \times & \times & \times & \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & 0 & 0 & 0 & 0 & 0 & 0 \\ \times & \times & 0 & \times & \times & \times & \times & \times & \times \\ 0 & \times & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \times & \times & 0 & \times & \times & \times & \times & \times & \times \\ \times & \times & 0 & \times & \times & \times & \times & \times & \times \\ \times & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Observation row m is $\alpha \cdot e_m$!

Cascade

All columns with pivot in row m (after Π):

$$
\begin{pmatrix}
\times & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \times & \times & 0 & 0 & 0 & 0 & 0 & 0 \\
\times & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \times & \times & 0 & 0 & 0 & 0 & 0 & 0 \\
\times & 0 & \times & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \times & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\times & 0 & \times & 0 & 0 & 0 & 0 & 0 & 0 \\
\times & 0 & \times & 0 & 0 & 0 & 0 & 0 & 0 \\
\times & \times & \times & \times & \times & \times & \times & \times & \times
\end{pmatrix}
\Rightarrow
\begin{pmatrix}
\times & \times & \times & \times & \times & \times & \times & \times & \times \\
0 & \times & \times & 0 & 0 & 0 & 0 & 0 & 0 \\
\times & \times & \times & \times & \times & \times & \times & \times & \times \\
0 & \times & \times & 0 & 0 & 0 & 0 & 0 & 0 \\
\times & \times & 0 & \times & \times & \times & \times & \times & \times \\
0 & \times & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\times & \times & 0 & \times & \times & \times & \times & \times & \times \\
\times & \times & 0 & \times & \times & \times & \times & \times & \times \\
\times & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
$$

Observation row m is $\alpha \cdot e_m$!

Fix Remove fill by using row m as a pivot row.

Cascade

All columns with pivot in row m (after Π):

$$
\begin{pmatrix}
\times & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \times & \times & 0 & 0 & 0 & 0 & 0 & 0 \\
\times & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \times & \times & 0 & 0 & 0 & 0 & 0 & 0 \\
\times & 0 & \times & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \times & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\times & 0 & \times & 0 & 0 & 0 & 0 & 0 & 0 \\
\times & 0 & \times & 0 & 0 & 0 & 0 & 0 & 0 \\
\times & \times & \times & \times & \times & \times & \times & \times & \times
\end{pmatrix}
\Rightarrow
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \times & \times & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \times & \times & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \times & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \times & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \times & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \times & 0 & 0 & 0 & 0 & 0 & 0 \\
\times & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
$$

Observation row m is $\alpha \cdot e_m$!

Fix Remove fill by using row m as a pivot row.

theorem!

Theorem

In K is a complex with m simplices covered by U and K^U has size $m+n$ then the mayer vietoris algorithm uses $O(mn^2)$ time and $O(mn)$ space.

In practice

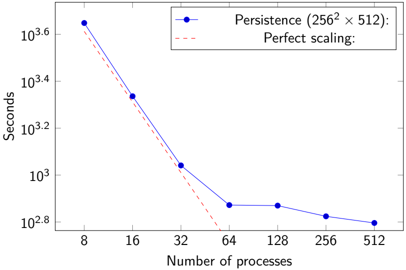


Figure: Times to compute persistence diagram for the $256^2 \times 512$ combustion data set. Credit: Dmitry Morozov.

In practice

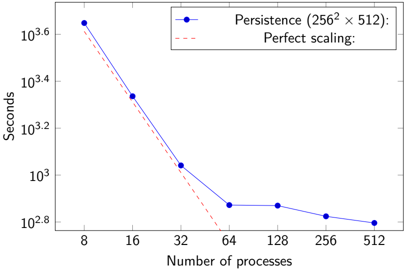


Figure: Times to compute persistence diagram for the $256^2 \times 512$ combustion data set. Credit: Dmitry Morozov.

- An input of size of 1.3×10^6 while quite large, is still considerably smaller than what can be computed today.
- Interesting: memory usage is not closely tracking our space bound.
- Slowdown as number of processes increase matches our intuition, total size of intersection is getting much larger.

Future Directions

1. There is still some room to improve the space complexity of this algorithm, for example, by reducing the factor n in $O(mn)$
2. Algorithm is top heavy, eventually a large matrix is on one machine.
3. We wanted to use M.V. to avoid this! Still some room for more cleverness here.

Mayer Vietoris Spectral Sequence

$$
\begin{tikzcd}
\vdots \arrow[d, "\partial_K"] & \vdots \arrow[d, "\partial_K"] & \vdots \arrow[d, "\partial_K"] & \\
E^0_{2,0} \arrow[d, "\partial_K"] & E^0_{2,1} \arrow[l, "\partial_N"'] \arrow[d, "\partial_K"] & E^0_{2,2} \arrow[l, "\partial_N"'] \arrow[d, "\partial_K"] & \cdots \arrow[l, "\partial_N"'] \\
E^0_{1,0} \arrow[d, "\partial_K"] & E^0_{1,1} \arrow[l, "\partial_N"'] \arrow[d, "\partial_K"] & E^0_{1,2} \arrow[l, "\partial_N"'] \arrow[d, "\partial_K"] & \cdots \arrow[l, "\partial_N"'] \\
E^0_{0,0} & E^0_{0,1} \arrow[l, "\partial_N"'] & E^0_{0,2} \arrow[l, "\partial_N"'] & \cdots \arrow[l, "\partial_N"']
\end{tikzcd}
$$

$$E^0_{p,q} = \langle p\text{-chains in a } q\text{-way intersection}\rangle$$

Mayer Vietoris Spectral Sequence

$$
\begin{tikzcd}
 & \vdots \arrow[d, "\partial_K"] & \vdots \arrow[d, "\partial_K"] & \vdots \arrow[d, "\partial_K"] & \\
C_2\left(K^U\right) & E^0_{2,0} \arrow[d, "\partial_K"] & E^0_{2,1} \arrow[l, "\partial_N"] \arrow[d, "\partial_K"] & E^0_{2,2} \arrow[l, "\partial_N"] \arrow[d, "\partial_K"] & \cdots \arrow[l, "\partial_N"] \\
C_1\left(K^U\right) & E^0_{1,0} \arrow[d, "\partial_K"] & E^0_{1,1} \arrow[l, "\partial_N"] \arrow[d, "\partial_K"] & E^0_{1,2} \arrow[l, "\partial_N"] \arrow[d, "\partial_K"] & \cdots \arrow[l, "\partial_N"] \\
C_0\left(K^U\right) & E^0_{0,0} & E^0_{0,1} \arrow[l, "\partial_N"] & E^0_{0,2} \arrow[l, "\partial_N"] & \cdots \arrow[l, "\partial_N"]
\end{tikzcd}
$$

$$E^0_{p,q} = \langle p\text{-chains in a } q\text{-way intersection}\rangle$$

The first two differentials:

$$d_0 = \partial_K \text{ and } d_1 = \partial_N$$

Mayer Vietoris Spectral Sequence

$$\begin{tikzcd}
\vdots \arrow[d, "\partial_K"] & \vdots \arrow[d, "\partial_K"] & \vdots \arrow[d, "\partial_K"] & \\
E^0_{2,0} \arrow[d, "\partial_K"] & E^0_{2,1} \arrow[l, "\partial_N"] \arrow[d, "\partial_K"] & E^0_{2,2} \arrow[l, "\partial_N"] \arrow[d, "\partial_K"] & \cdots \arrow[l, "\partial_N"] \\
E^0_{1,0} \arrow[d, "\partial_K"] & E^0_{1,1} \arrow[l, "\partial_N"] \arrow[d, "\partial_K"] & E^0_{1,2} \arrow[l, "\partial_N"] \arrow[d, "\partial_K"] & \cdots \arrow[l, "\partial_N"] \\
E^0_{0,0} & E^0_{0,1} \arrow[l, "\partial_N"] & E^0_{0,2} \arrow[l, "\partial_N"] & \cdots \arrow[l, "\partial_N"]
\end{tikzcd}$$

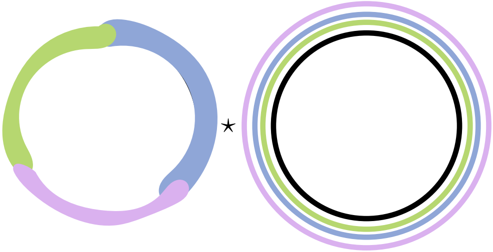
$$E^0_{p,q} = \langle p\text{-chains in a } q\text{-way intersection}\rangle$$

The first two differentials:

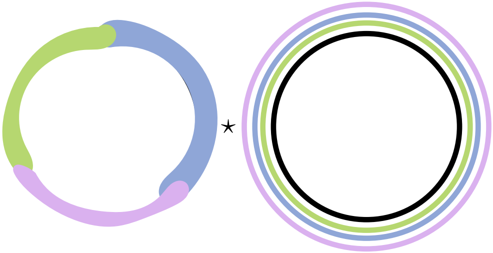
$$d_0 = \partial_K \text{ and } d_1 = \partial_N$$

We can construct the blowup *chain complex* where $C_d = \bigoplus_{p+q=d} E^0_{p,q}$ with $\partial = d_0 + (-1)^q d_1$ Let's try an example!

Example: $S^1 \star S^1$

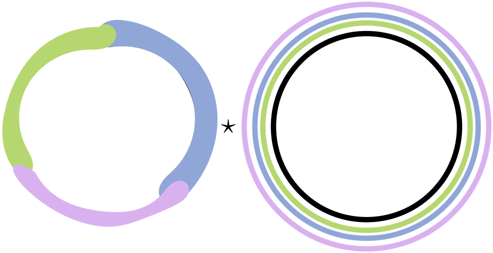


Example: $S^1 \star S^1$



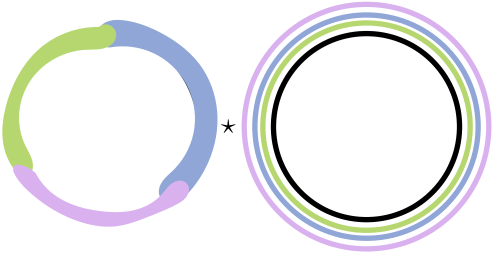
1. Each of the three sets are a copy of $I \star S^1$

Example: $S^1 \star S^1$



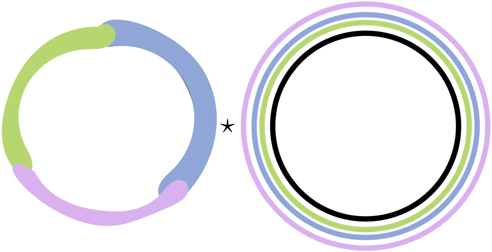
1. Each of the three sets are a copy of $I \star S^1$
2. Each of the three pairwise intersection is $\{\mathrm{pt}\} \star S^1$

Example: $S^1 \star S^1$



1. Each of the three sets are a copy of $I \star S^1$
2. Each of the three pairwise intersection is $\{\mathrm{pt}\} \star S^1$
3. Single triple intersection is a copy of S^1.

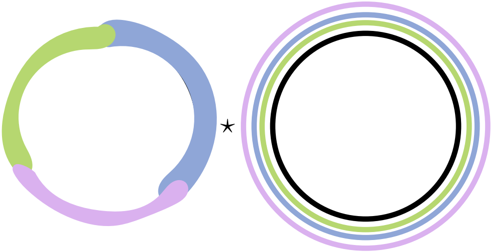
Example: $S^1 \star S^1$



The E_1 page has terms with the following data:

$$
\begin{array}{ccccc}
0 & & 0 & & 1 \\
\\
3 & \xleftarrow[\begin{pmatrix} 1 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 1 \end{pmatrix}]{} & 3 & \xleftarrow[\begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}]{} & 1
\end{array}
$$

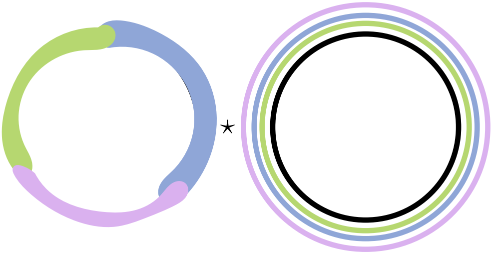
Example: $S^1 \star S^1$



The E_2 page has terms of the following data:

$$\begin{tikzcd}
0 & 0 & 1 \\
1 & 0 & 0 \arrow[ull, "0"']
\end{tikzcd}$$

Example: $S^1 \star S^1$



The E_2 page has terms of the following data:

$$
\begin{tikzcd}
0 & 0 & 1 \\
1 & 0 & 0 \arrow[ull, "0"]
\end{tikzcd}
$$

$$H_d(K^U) = \bigoplus_{p+q=d} E^{\infty}_{p,q}$$

$$H_0(S^1 \star S^1) \cong H_2(S^1 \star S^1) = 1$$

That's all she wrote!

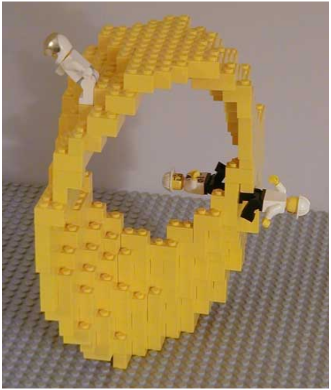


Figure: Lego Möbius Strip:
By Andrew Lipson

Thank you!
Code available at:
`http://github.com/appliedtopology/ctl`

Lego Art from:
`http://andrewlipson.com`