Persistence Algorithm

Lecture 10 - CMSE 890

Prof. Elizabeth Munch

Michigan State University
::
Dept of Computational Mathematics, Science & Engineering

Thurs, Oct 2, 2025

Today

Goals for today:

- Persistence algorithm

Section 1

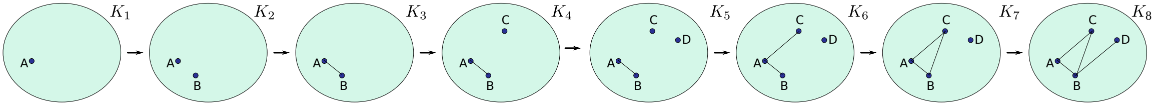
Review of setup

Simplicial Complex Filtration from a Function

Definition

Fix a function $f : K \to \mathbb{R}$ with $(\sigma \leq \tau \Rightarrow f(\sigma) \leq f(\tau))$
Fix values $a_1 \leq a_2 \leq \cdots \leq a_n$
Let $K_i = f^{-1}(-\infty, a_i]$
A filtration $\mathcal{F} = \mathcal{F}(K)$ induced by f is a nested sequence of subcomplexes

$$\mathcal{F} : \emptyset = K_0 \subseteq K_1 \subseteq K_2 \subseteq \cdots \subseteq K_n = K.$$



Persistence module

Given a filtration $\mathcal{F}$, the p-dimensional persistence module is

$$H_p(\mathcal{F}) : 0 = H_p(K_0) \to H_p(K_1) \to H_p(K_2) \cdots \to H_p(K_n) = H_p(K)$$

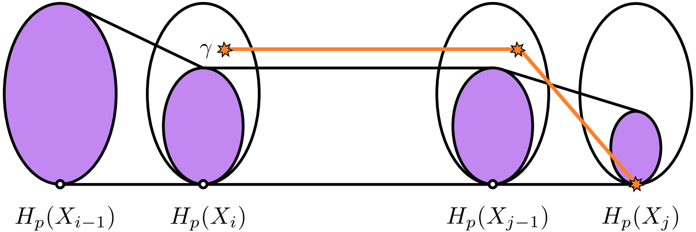
with maps $h_p^{i,j} : H_p(K_i) \to H_p(K_j)$ induced by inclusion.
The pth persistent homology groups are the images induced by inclusion:

$$H_p^{i,j} = \mathrm{Im}(H_p(K_i) \to H_p(K_j)).$$

The pth persistent Betti numbers are the ranks

$$\beta_p^{i,j} = \mathrm{rank} H_p^{i,j}$$

Birth and Death: Book version



Birth

A class $\gamma \in H_p(K_i)$ is born at K_i if it is not in $H_p^{i-1,i}$

Death

A class γ dies entering K_j if $\gamma \in H_p(X_{j-1})$ is not trivial, but $h_p^{j-1,j}(\gamma) = 0$.[a]

[a] Warning: In this version, not all classes die!

Section 2

Persistence Algorithm

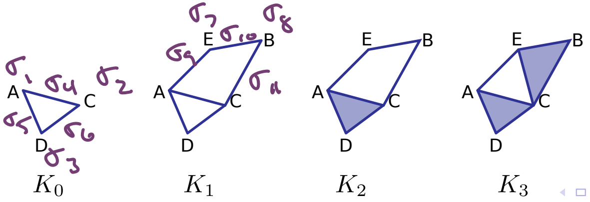
Compatible filtration

$\sigma \mapsto a_i$ with $f(\sigma) \leq a_i$

Let $f : K \to \mathbb{N}$ give the index where any particular simplex appears in the filtration.
A compatible ordering of the simplices of K is a sequence $\sigma_1, \sigma_2, \cdots, \sigma_m$ such that

- $f(\sigma_i) < f(\sigma_j)$ implies $i < j$
- $\sigma_i \leq \sigma_j$ implies $i < j$

(Could have just assumed that the filtration was simplex-wise to begin with)



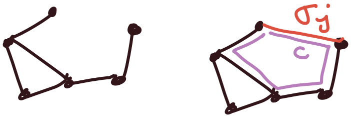
Positive and negative simplices

Simplex-wise filtration:

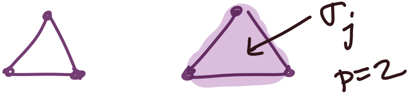
When a p-simplex $\sigma_j = K_j \setminus K_{j-1}$ is added, exactly one of the following two possibilities occurs:

1. A non-boundary p-cycle c along with its classes $[c] + h$ for any class $h \in H_p(K_{j-1})$ are born (created). In this case we call σ_j a positive simplex (also called a creator).
2. An existing $(p-1)$-cycle c along with its class $[c]$ dies (destroyed). In this case we call σ_j a negative simplex (also called a destructor).

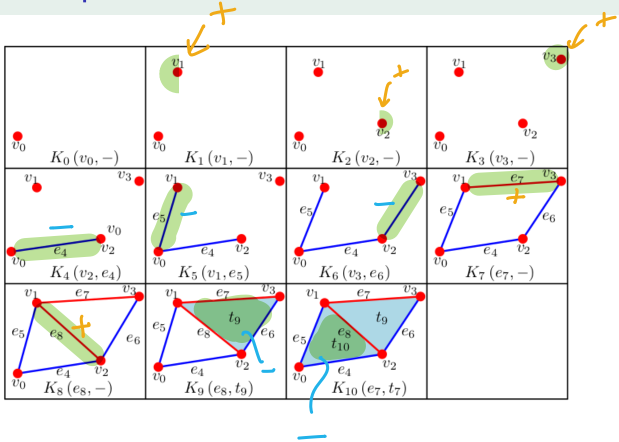
Before After $p=1$



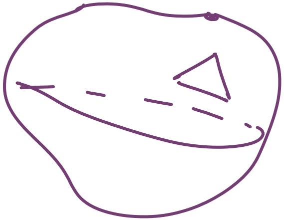
Before After



Example



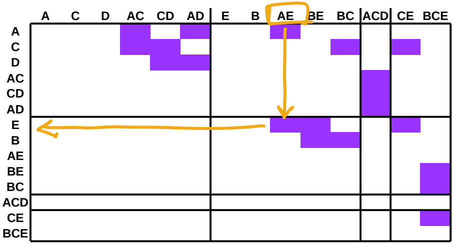
$K_0\,(v_0,-)$
$K_1\,(v_1,-)$
$K_2\,(v_2,-)$
$K_3\,(v_3,-)$
$K_4\,(v_2,e_4)$
$K_5\,(v_1,e_5)$
$K_6\,(v_3,e_6)$
$K_7\,(e_7,-)$
$K_8\,(e_8,-)$
$K_9\,(e_8,t_9)$
$K_{10}\,(e_7,t_7)$



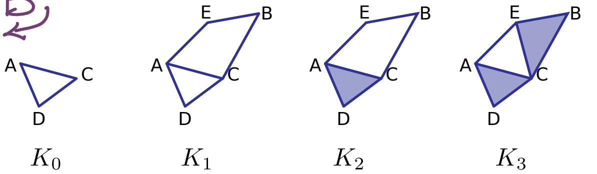
The persistence algorithm

Given the boundary matrix B with rows and columns in this total ordering.
We will convert B to another matrix R using row/col operations. Result: $\underbrace{R = BV}$ for some invertible V.

- Let $low(j)$ be the row index of the lowest one in column j.
- If column j is entirely 0, $low(j)$ is `NaN`.
- R is "reduced" if $low(j) \neq low(j')$ for any $j \neq j'$ which are not entirely zero columns.



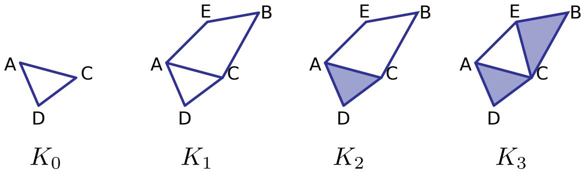
B

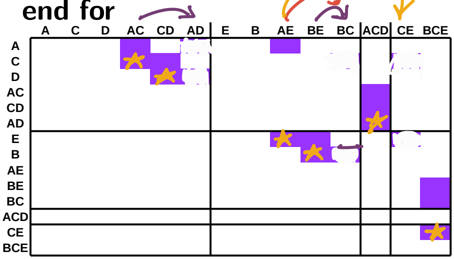


Persistence algorithm

$C_p(K) \xrightarrow{\partial} C_{p-1}(K)$

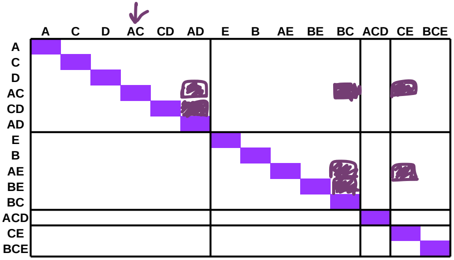
$R = B$
for $j = 1 \cdots m$ **do**
  while $\exists\, j' < j$ with $low(j') = low(j)$ **do**
    add column j' to column j
  end while
end for





(start @ B)

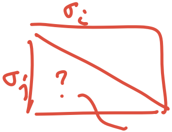
R



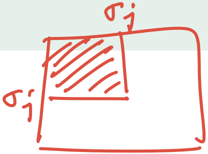
V

$R = B \cdot V$

Idea



σ_j after σ_i

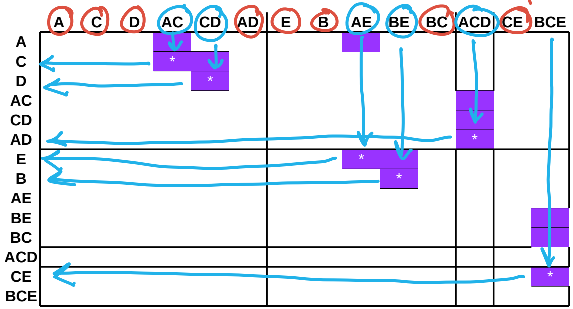


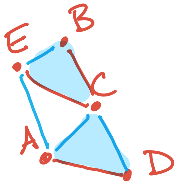
- B is upper triangular, only add from right so stays upper triangular.
- Upperleft corner from (j, j) represents homology for the subcomplex stopping at σ_j.
- If a column is entirely 0, in R the addition of that simplex created a new homology class.
- If a column has a lowest one, then the addition of this simplex killed off a class that was there at the previous step.

Pairing

- Every negative simplex must be paired with a previously entering positive simplex
- Negative simplex has a non-zero column, so it's paired with the simplex from the row of the lowest one

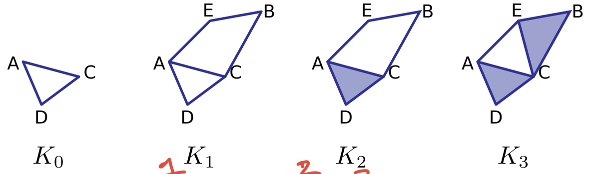
positive
= red
= zero column



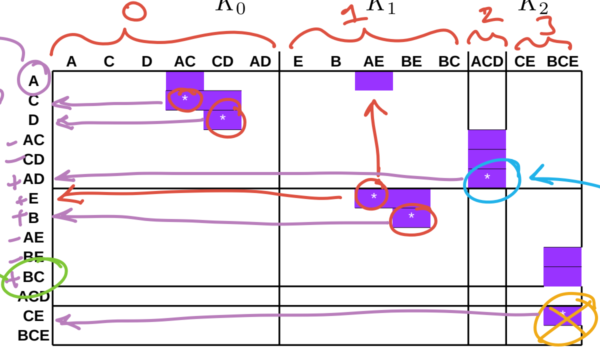


Reading off the persistence diagram

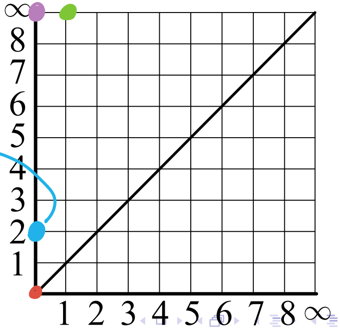
Pos simplex
no pair
⇒ ∞-class



Born @ AD 0
Died @ ACD 2



Pos:
no pair



Section 3

Speedups

History

- Matrix reduction algorithm is from 2000 paper by Edelsbrunner, Letscher, Zomorodian
- Algebraic viewpoint: Carlsson & Zomorodian 2004
- Precursors to the idea:
  - Frosini 1990
  - Robins 1999

Running time of the matrix reduction algorithm

↳ Same as matrix multiplication

↳ $O(n^{\omega})$ — ω: 2.371

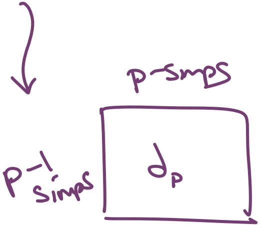
n: number of rows/cols — the number of simplices

This is worst case

Reality is better!

Dimension and persistence

Computing p-dimensional persistence requires up to $p+1$ dimensional simplices



Weird reason:

Cohomology

Do this on the transposed matrix and go faster

The special case of 0-dimensional persistence

Minimum spanning tree, union find

much faster algorithms
only work on 0-dimensional persistence

Other available speedups

Section 4

Code for persistence

Available codebanks

- Ripser
- Dionysus
- Teaspoon
- Gudhi
- Scikit-tda
- Giotto-tda
- Phat
- TDA (R)
- Lots more....

For next time

- Install teaspoon, scikit-tda
  - `pip install teaspoon`
  - `pip install scikit-tda`
- Review Rips/Cech complexes
- Bring computer for next time

oat
gudhi