Today: Finish PH / Reduction
Implementation (Basic) over $\mathbb{F}_2$
Some optimizations

$$0 \xleftarrow{\partial_0} C_0 \xleftarrow{\partial_1} C_1 \xleftarrow{\partial_2} C_2 \xleftarrow{\partial_3} \cdots$$

reduction alg: $A\underline{U} = \underline{R}$
↳ has unique low pivots.

apply reduction alg to each boundary matrix ∂_k in chain complex. Want to read off $H_k = \ker \partial_k / \operatorname{Img} \partial_{k+1}$ from our factorizations

$$\partial_k U_k = R_k \leftarrow$$
$$\partial_{k+1} U_{k+1} = R_{k+1}$$

---

1) if $R_k[j] = 0 \iff U_k[j]$ is a cycle $\in \ker \partial_k$

2) if $R_{k+1}[j]$ has pivot i, then $U_k[i]$ is a cycle $\Rightarrow R_k[i] = 0$, (last time)

Comments: from (1) + upper triangular structure of U, we can conclude a basis for $\ker \partial_k$ is given by $\{U_k[j] \mid R_k[j] = 0 \quad j = 1 \dots n\}$

from (2) we get a basis for $\operatorname{Img} \partial_{k+1}$ by looking at $\{U_k[i] \mid R_{k+1}[j]$ has pivot i for some $j\}$

U upper triangular b/c of reduction ops:

initialization: $U = I_n$ (Input $m \times n$ matrix A)

$R = A$

for $j = 1 \dots n$

  while piv $R[j] > -1$ and $\exists\, j' < j$ s.t.

    $i =$ piv $R[j] =$ piv $R[j']$

    $\alpha = R[i,j] / R[i,j']$

    $R[j] = R[j] - \alpha R[j]$

    $U[j] = U[j] - \alpha U[j']$

} applies U.T. matrix

$$\begin{array}{cc} & \ \ j \\ j' & \begin{bmatrix} 1 & -\alpha & \\ & \ddots & \\ & & 1 \end{bmatrix} \end{array}$$

Prop: the set cycles in U_k whose column index do <u>not</u> appear as a pivot in R_{k+1} form a basis for $H_k = \ker \partial_k / \operatorname{Img} \partial_{k+1}$

i) these cycles are linearly indepen. b/c. they are distinct columns of U.

$$\dim H_k = \dim \ker \partial_k - \dim \underbrace{\operatorname{img} \partial_{k+1}}_{\operatorname{img} \partial_{k+1} = \operatorname{img} R_{k+1}}$$

cycles not pivots $= \dim \ker \partial_k - \dim \operatorname{img} R_{k+1}$
b/c lin indep, and rep non-zero homology, must ~~form~~ a basis for H_k. ∎

Basic procedure to compute H_k:

obtain factorizations $\partial_k U_k = R_k$

$\partial_{k+1} U_{k+1} = R_{k+1}$

Count zero columns of R_k which do not appear as pivot index in R_{k+1}

How to compute reduced maps $\tilde{F}_k : H_k(C_*) \to H_k(D_*)$

$$\tilde{F}_k[x] = \Big[\sum_{i \in I_k^D} \alpha_i u_i\Big] = \sum_{i \in I_k^D} \alpha_i [u_i]$$

How to put a homology rep. in terms of the preferred reps?

$x = U_k^C[i] \quad i \in I_k^C$

$$\underbrace{F_k x}_{\text{in the basis of reduction alg}} \Rightarrow \underbrace{(U_k^D)^{-1} F_k x}_{\text{might not be preferred rep.}}$$

Alg: $F_k : C_k \to D_k$ chain map (maps cycles to cycles) b/c $\partial F = F \partial$

$y \leftarrow (U_k^D)^{-1} F_k x$

$n := \dim D_k$

$$\delta_{k+1}^D \leftarrow (U_k^D)^{-1} R_{k+1} = (U_k^D)^{-1} \partial_{k+1} (U_{k+1}^D)^{-1}$$

for: $j = n, n-1, \dots 1$

if ($y[j] \neq 0$ and j is a pivot of column i of $\hat{\partial}^D_{k+1}$ then):

$\alpha \leftarrow y[j] / \hat{\partial}^D_{k+1}[j, i]$

$y \leftarrow y - \alpha \hat{\partial}^D_{k+1}[i]$

} variant of backward substitution alg.

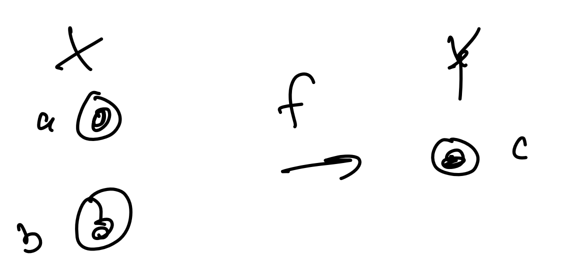
return $y[I^D_k]$

1) hom class of y is invariant. Why? we add elements of img ∂_{k+1} (in the HRB)

2) have preferred rep at the end of the for-loop: why? if we have a coeff not in the index set I^D_k, we eliminate that coeff.

---

to compute reduced map: map each preferred rep in U^C_k, calculate coeffs of img in terms of preferred reps of U^D_k.



Chain maps satisfy

$\partial F = F \partial$

proved for simplicial maps.

$a \to c$ $\quad \partial_0 a = 0$ $\quad \partial_0 c = 0$

$b \to c$ $\quad \partial_0 b = 0$

$C_0(X)$ vect. space. chains are vectors
cycles are chains which are in $\ker \partial_k$
boundaries are chains in $\operatorname{Img} \partial_{k+1}$.

$$0 \xleftarrow{\partial_0} C_0(X) \xleftarrow{\partial_1} C_1(X) \xleftarrow{\partial_2}$$

---

persistant homology:
filtration $X_0 \subseteq X_1 \subseteq X_2 \subseteq \dots$
Hom functor: $H_k(X_0) \xrightarrow{i} H_k(X_1) \xrightarrow{i} \dots$

Claim: you can compute barcode $\{(b_i, d_i)\}$ using reduction alg.

From last time: every k-spx addition either creates the or destroys them

why?

1) order rows/columns of ∂_k in filtration order.

   $\partial_k[1]$ is boundary of 1^{st} k-simplex added
   $\partial_k[2]$ " " 2^{nd} " "

2) run reduction alg.

3) key observation:

reduction for first j columns of ∂_k^j and $\partial_k^{j'}$ for $j < j'$ is going to happen in exactly the same way.

instead of computing everything $H_k(X_j) \to H_k(X_{j+1})$ "re-use computation"

every time we add a simplex, just need to reduce that column.

Recall: $\{(b_i, d_i)\}$ persistence pairs / barcode / persistence diagram.

every pair has an associated homology class which first appears at b_i (not in img of induced map) and enters ker of induced map at d_i.

How to extract: look at R_k, R_{k+1} again (in filtration order)

now, there is a birth of homology class at j if $R_k[j] = 0$; death of class at j' when j appears as a pivot in $R_{k+1}[j']$

(or death at ∞ if this never happens).

Q: is $\{(b_i, d_i)\}$ unique?

what if we use a different alg to compute U_{κ}, R_{κ}?

A: it is unique.

Z & C '05: show that $\{(b_i, d_i)\}$ give classification up to change of basis.

Correspondence with $\underbrace{\mathbb{F}[T]}$ modules

PID if $\mathbb{F}$ is a field.

nice classification.

C & dS '10: give indecomposables of type-A quiver rep.

Optimizations:

1) don't keep track of U_{κ}.

just compute R_{κ}.

2) Clearing: '11: Chen & Kerber "Twist"

'11 dS, M, VJ cohomology.

Observation: if column j of $R_{\kappa+1}$ has pivot i, then column i of R_{κ} will be zero.

if we don't care about recording V_k, we can just set $R_k[i]=0$ without doing any work.

In order to apply this, should process ∂_k in reverse dimension order.

1) $R_{k+1} = \partial_{k+1} V_{k+1}$
2) identify pivots in R_{k+1}
3) "clear" or "kill" columns in ∂_k
4) reduce ∂_k

3: Compression: Bauer, Kerber Reininghaus '14.

remove rows of ∂_{k+1} by analyzing cols in R_k.

Observation: if column i is <u>not</u> zero in R_k, row i will <u>not</u> contain a pivot in R_{k+1}. (b/c pivot in $R_{k+1} \Rightarrow$ zero col of R_k)

these rows are never used in reduction, or barcode extraction, so don't need to even look at them.

Clearing + Compression algorithm.

operates by looking at chunks ~~of~~ matrix independently, use clearing

use compression when combining chunks.

Ripser (Bauer 2017)