MATH 529 – Lecture 23 (04/02/2024)

Reminder: Start thinking about options for the project (explore tutorials on scikit-tda.org)

Today: * details of pairing
* mapper algorithm

Details of Pairing

We show that the lowest 1's are unique, i.e., they do not depend on the final reduced matrix R.

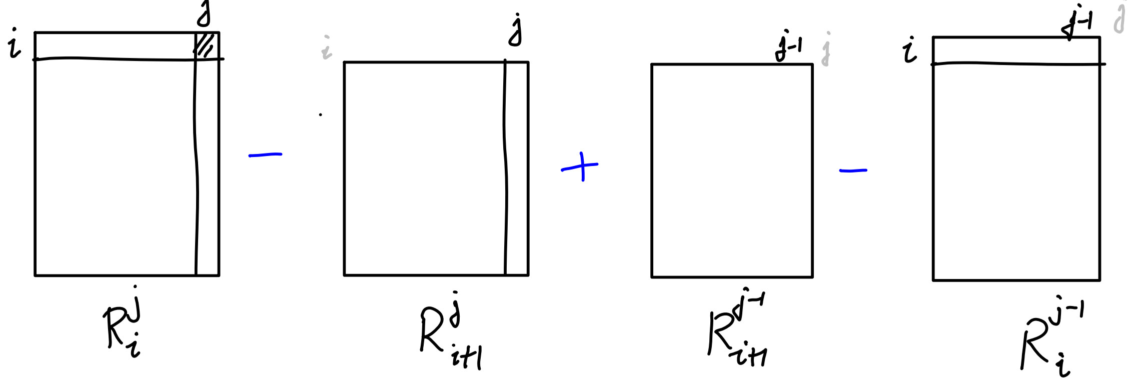
Let's look at R_i^j, the lower left submatrix whose corner element is $R[i,j]$, i.e., remove rows 1 to $i-1$, and columns $j+1$ to m (last $m-j$ colums).

Since we do ECOs left to right, the ranks of R_i^j are presevred.

In particular, $\operatorname{rank} R_i^j = \operatorname{rank} [\partial]_i^j \quad \forall i,j$.

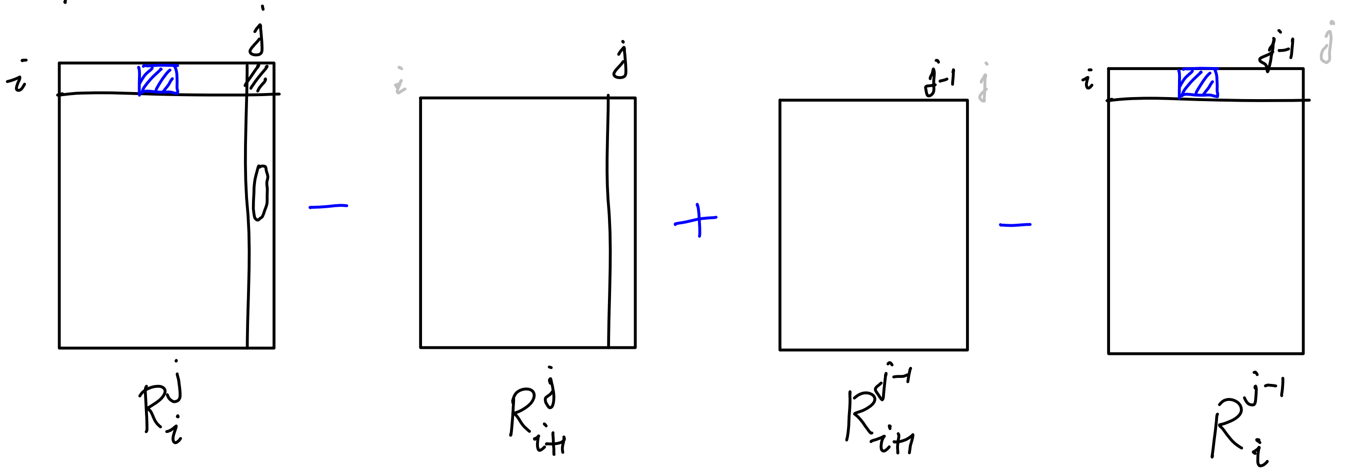
Also $\operatorname{rank} R_i^j = \#$ nonzero columns in R_i^j. You can see this result by observing that any combinations of nonzero columns of R remain non-zero, following how we apply ECOs to obtain R in the first place.

Consider $r_R(i,j) = \operatorname{rank} R_i^j - \operatorname{rank} R_{i+1}^j + \operatorname{rank} R_{i+1}^{j-1} - \operatorname{rank} R_i^{j-1}$.



Note that $r_R(i,j) = r_{[\partial]}(i,j) \quad \forall i,j$; the similar submatrix obtained from $[\partial]$ instead of R.

$$r_R(i,j) = \text{rank } R_i^j - \text{rank } R_{i+1}^j + \text{rank } R_{i+1}^{j-1} - \text{rank } R_i^{j-1}.$$



We consider two possibilities for $R[i,j]$:

<u>$R[i,j]$ is a lowest 1</u> $\Rightarrow$ R_i^j has one extra nonzero row than other 3 matrices

$\Rightarrow \quad r_R(i,j) = 1.$

<u>$R[i,j]$ is not a lowest 1</u> : We consider two subcases here.

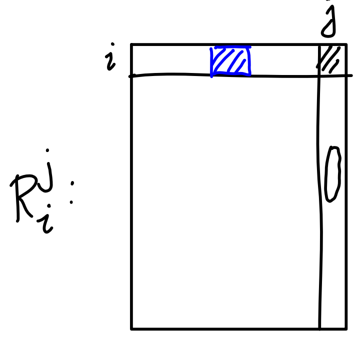
$\hookrightarrow$ There is no lowest 1 in Row i in columns 1 to $j-1$.

$\Rightarrow$ # NZ columns in R_i^j = # nz-columns in R_{i+1}^j

Similarly for R_{i+1}^{j-1} and R_i^{j-1}

$\Rightarrow \quad r_R(i,j) = 0.$

$\hookrightarrow$ There is a lowest 1 in row i in columns 1 to $j-1$:



$r_R(i,j) = 0$ here as well, since R_i^j will have an extra nonzero column than R_{i+1}^j, and similarly, R_i^{j-1} has an extra nonzero column than R_{i+1}^{j-1}.

Hence we get the result that (i,j) are paired independent of the exact form of R, i.e., it depends only on $[\partial]$.

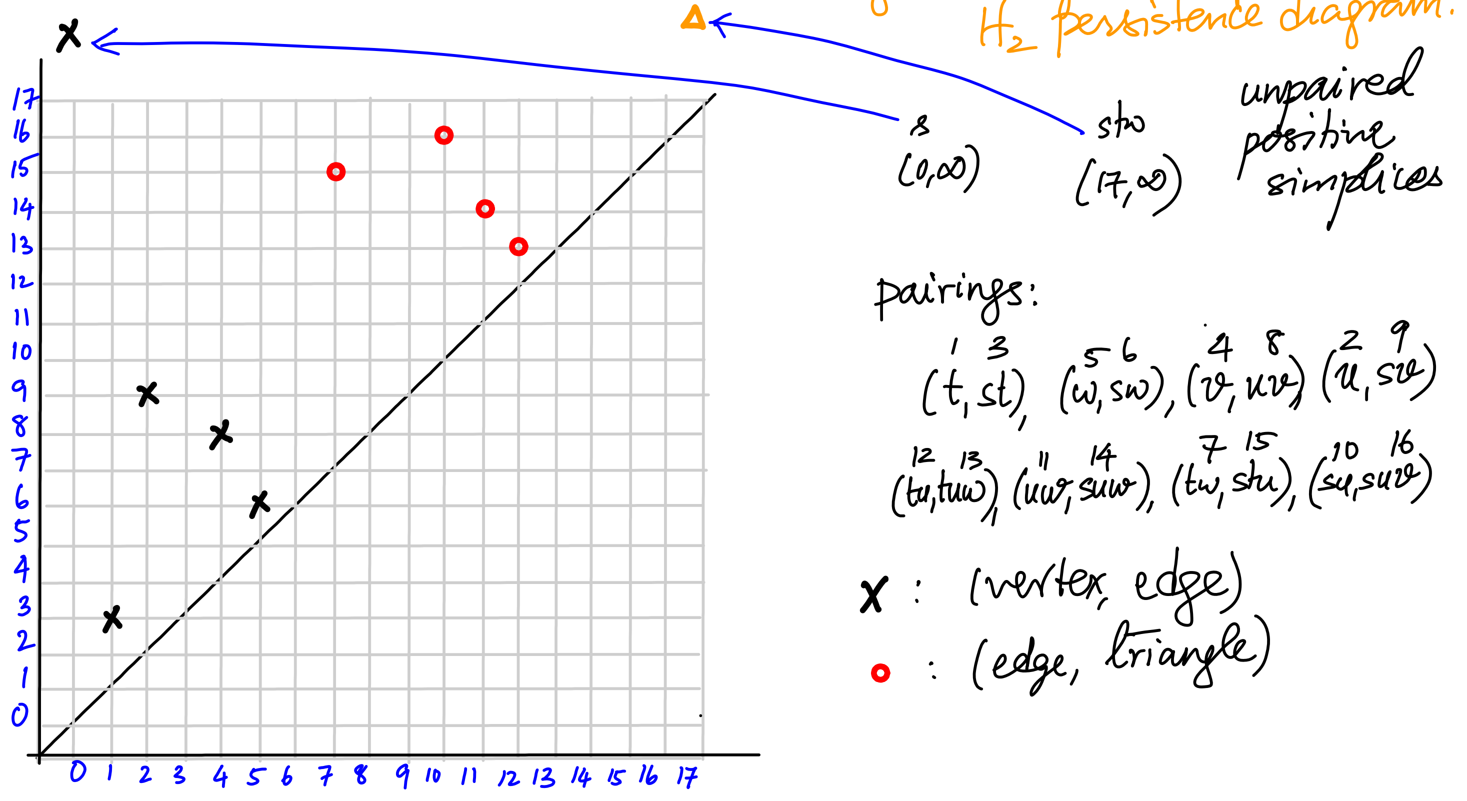
Pairing Lemma $i = \text{low}(j)$ iff $r_\partial(i,j) = 1$. The pairings are independent of R, and depend only on $[\partial]$.

Classification of Simplices

* If column j of R is zero, then σ^j is positive.
* If column j of R is non-zero, σ^j is negative.

  column j stores the boundary of the chain accumulated in the jth column of V, where $R = [\partial]V$.

Here are the 0th and 1st persistence diagrams drawn together — typically, they are drawn separately, along with the single unpaired triangle that gives a point in the H_2 persistence diagram.



pairings:

$(\overset{1}{t}, \overset{3}{st}), (\overset{5}{w}, \overset{6}{sw}), (\overset{4}{v}, \overset{8}{uv}), (\overset{2}{u}, \overset{9}{sv})$

$(\overset{12}{tu}, \overset{13}{tuw}), (\overset{11}{uw}, \overset{14}{suw}), (\overset{7}{tw}, \overset{15}{stu}), (\overset{10}{su}, \overset{16}{suv})$

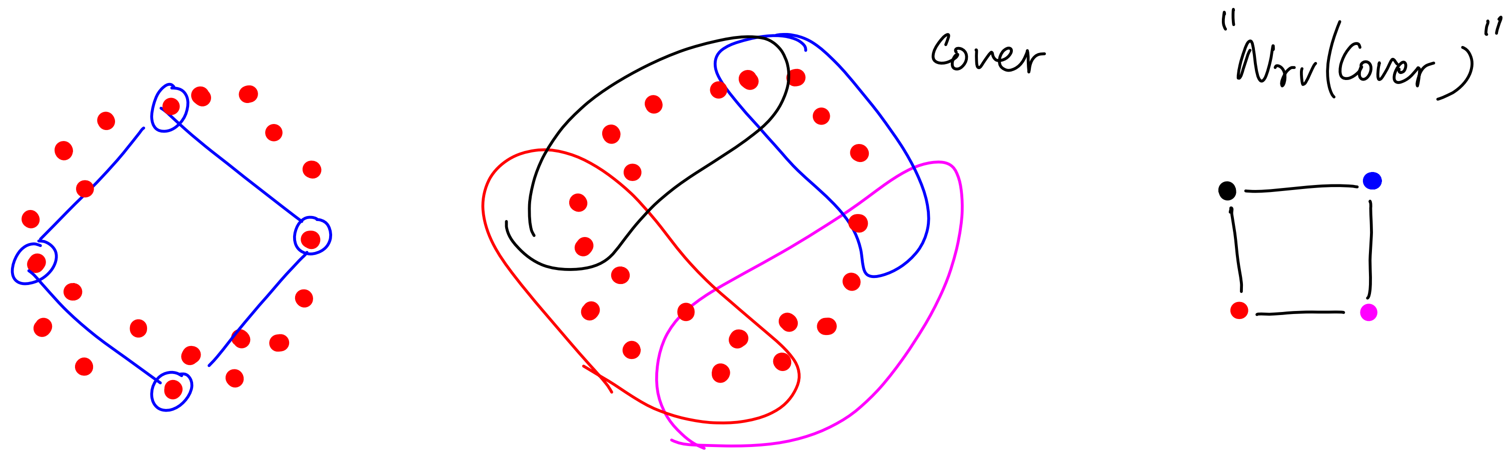
$\times$: (vertex, edge)

○ : (edge, triangle)

Check out Ripser (part of Scikit-tda), GUDHI, Dionysis, etc.

The Mapper Algorithm

The idea is to create "highly compact" summaries of point cloud data. To this end we "cover" the range of values using some open sets, and then represent the points within each such set using a small number of nodes representing clusters. We could then capture intersections between two such sets by edges connecting their nodes. Here is a simple example for a set of points sampled from a circle.



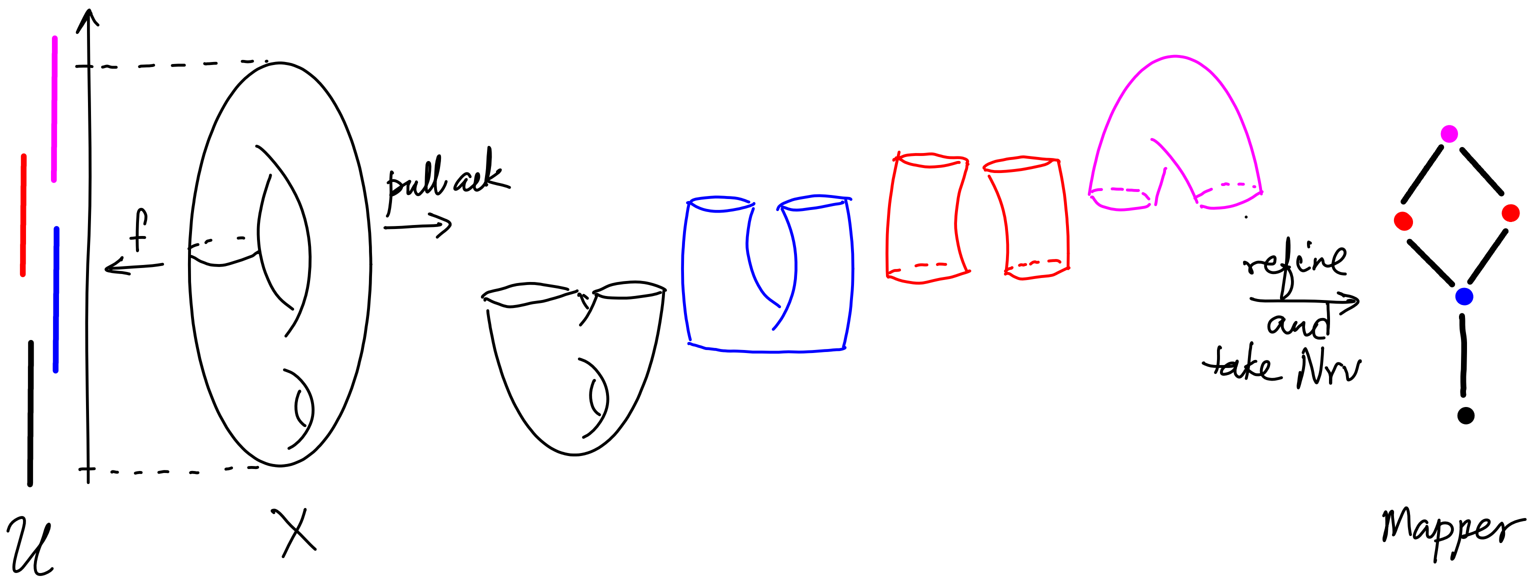
More formally, we take the nerve of the pullback of the cover — in other words, we get a simplicial complex that is much smaller in size than the input point cloud (for appropriate choices of the cover elements). Here, we get a circle with 4 nodes and 4 edges.

A (more) formal definition of mapper

Def Let $f: X \to \mathbb{R}^d$, $d \geq 1$, be a continuous real-valued function, and $\mathcal{U} = (U_i)_{i \in I}$ be an open cover of $f(X)$. The **pullback cover** of X induced by $(f, \mathcal{U})$ is a collection of open sets $(f^{-1}(U_i))_{i \in I}$. The ***refined pullback*** is the collection of connected components of the open sets $f^{-1}(U_i)$, $i \in I$. The ***mapper*** complex $M(f, \mathcal{U})$ is defined as the nerve of the refined pullback.

Here is another pictorial illustration on a connected object. We cover the range of the height of points on the object.

as opposed to a point cloud



Notice that we did "lose" the hole in the bottom portion of the object here!