Poster content
Project IV: Computational Topology,
Epiphany Term Week 1
Ben Napier

1 Theory

We first discuss some intuition about persistent homology, in effort to motivate a following definition, and a (more traditional) alternative definition. Consider a filtration of some simplicial complex. Persistent homology aims to describe how *features* (homology classes) are *born* at a certain part of the filtration, and then *die* at a later date (this terminology will be shortly formalised).

Consider a filtration $\{K_i\}_{i=0}^n$ of some simplicial complex K. For each $i \leq j$, we have an inclusion map $\iota_{i,j} : K_i \hookrightarrow K_j$ from which we form an induced homomorphism on the homology groups $f_p^{i,j} : H_p(K_i) \to H_p(K_j)$, $[c] \mapsto [\iota_{i,j}(c)]$ for each $p \in \mathbb{Z}_{\geq 0}$. Thus the filtration corresponds to a sequence of homology groups as follows.

$$H_p(K_0) \xrightarrow{f_p^{0,1}} H_p(K_1) \xrightarrow{f_p^{1,2}} \dots \xrightarrow{f_p^{n-1,n}} H_p(K_n)$$

As we move from K_i to K_{i+1}, we may gain new homology classes, and we may also lose classes as they become trivial or merge with others.

Definition 1 (Persistent homology)**.** Let $\{K_i\}_{i=0}^n$ be a filtration of some simplicial complex K. For $p \in \mathbb{Z}_{\geq 0}$ and $i, j \in \mathbb{Z}_{\geq 0}$ with $i \leq j$, we define the *pth persistent homology* as

$$H_p^{i,j}(K) = \frac{Z_p(K_i)}{Z_p(K_i) \cap B_p(K_j)}.$$

That is, the cycles of K_i that modulo the boundaries of K_j (sans any boundaries that are not cycles of K_i). This definition is easier to accept, but the following is the more traditional definition and is easier to get around with.

Definition 2 (Persistent homology, alt.)**.** Let $\{K_n\}_{i=0}^n$ be a filtration of some simplicial complex K. Let $f_p^{i,j} : H_p(K_i) \to H_p(K_j)$ be the homomorphism on homology groups induced by the inclusion map $K_i \hookrightarrow K_j$ as previously discussed. We define the *pth persistent homology* as the image of this homomorphism, $H_p^{i,j}(K) = \operatorname{im} f_p^{i,j}$.

The *pth persistent Betti numbers* are defined as one may expect: $\beta_p^{i,j}$ is the rank of the corresponding pth persistent homology group. Now that we have some formal groundwork, we will introduce some terminology. Let $\{K_i\}_{i=0}^n$

be some filtration of a simplicial complex K, let $p \in \mathbb{Z}_{\geq 0}$, and let $i, j \in \mathbb{Z}_{\geq 0}$ with $i \leq j$.

- Let $p \in \mathbb{Z}_{\geq 0}$. A *$p$-hole* is another word for a p-dimensional homology class; that is, $[c] \in H_p(K)$.
- A p-hole $[c]$ is *born* at K_i if $[c] \in H_p(K_i) = H_p^{i,i}(K)$ but $[c] \notin H_p^{i-1,i}(K)$.
- A p-hole $[c]$ that is born at K_i *dies entering* K_j if it merges with an older class from K_{j-1} to K_j; that is, $f_p^{i,j-1}([c]) \notin H_p^{i-1,j-1}(K)$ but $f_p^{i,j}([c]) \in H_p^{i-1,j}(K)$.
- Suppose $[c]$ is a p-hole that is born at K_i and dies entering K_j. The *index persistence* of $[c]$ is $\operatorname{ipers}([c]) = j - i$.
- Denote $\mu_p^{i,j}$ as the number of p-holes that are born at K_i and dies entering K_j; that is, $\mu_p^{i,j} = (\beta_p^{i,j-1} - \beta_p^{i,j}) - (\beta_p^{i-1,j-1} - \beta_p^{i-1,j})$.

We move to visualise persistent Betti numbers in $\mathbb{R}^2$, but to do so we need the following property.

Lemma 1. *Let $\{K_i\}_{i=0}^n$ be a filtration of some simplicial complex K. For each $i, j \in \{0, 1, \ldots, n\}$ with $i \leq j$ and $p \in \mathbb{Z}_{\geq 0}$ we have*

$$\beta_p^{i,j} = \sum_{k \in \{0,\ldots,i\}} \sum_{l \in \{j+1,\ldots,n\}} \mu_p^{k,l}.$$

This fits with our intuition, $\beta_p^{i,j}$ is the number of p-holes that are alive from K_i to K_j; that is, the number of p-holes that are born at K_i or earlier, and die entering K_{j+1} or later.

Up to now, we have only needed a simplicial complex with a filtration defined upon it, but to continue with our visualisation of persistent Betti numbers, we need more. Let $f : K \to \mathbb{R}$ be some map defined on a simplicial complex K such that f is non-decreasing on increasing sequences of faces (that is, if $\sigma \subset \tau \in K \implies f(\sigma) \leq f(\tau)$). Then we let $\{K_i\}_{i=0}^n$ be the induced filtration of K such that $K_0 = \varnothing$, $K_n = K$, and for each $i \in \{1, \ldots, n-1\}$: $K_i = f^{-1}(-\infty, a]$ for some $a \in \mathbb{R}$.

Now, we let $\{a_i\}_{i=0}^n$ be such that $K_i = f^{-1}(-\infty, a_i]$. Suppose $[c]$ is a p-hole that is born at K_i and dies entering K_j. The *persistence* of $[c]$ is $\operatorname{pers}([c]) = a_j - a_i$.

We define the *pth persistence diagram* of f as the multiset

$$\operatorname{dgm}_p(f) = \{(a_i, a_j)^{\mu_p^{i,j}} : \mu_p^{i,j} > 0, 0 \leq i < j \leq n\},$$

where $(a_i, a_j)^{\mu_p^{i,j}}$ denotes the element (a_i, a_j) with multiplicity $\mu_p^{i,j}$. We observe that $\mu_p^{i,i} = 0$.

2 Problem definition

We introduce a simplified problem before tackling the problem of calculating persistent Betti numbers.

Definition 3 (Abstract simplicial complex)**.** An *abstract simplicial complex* A is a finite collection of non-empty sets such that $\alpha \in A$ and $\beta \subset \alpha$ implies $\beta \in A$.

In this section, when referring to a simplicial complex we mean an abstract simplicial complex. The definitions for homology still hold.

For a n-dimensional simplicial complex K, define $\beta_i(K)$ for $i \in \mathbb{Z}_{\geq 0}$ as the ith Betti number of K; that is, $\beta_i(K) = \operatorname{rank} H_i(K)$ where $H_i(K)$ denotes the ith homology group of K. Such a set is well-defined, and so we can define a total function problem. Denote $\mathcal{K}$ the set of all simplicial complexes (up to homeomorphism).

Problem 1 (Betti)**.**
The Betti number problem is the total function problem defined from the language

$$\textsc{Betti} = \{\langle K, \{\beta_i(K)\}_{i=0}^{\dim K}\rangle : K \in \mathcal{K}\}.$$

By default, this problem assumes integral homology, but we define a variation of it with coefficients over a given abelian group. For a simplicial complex K and an abelian group A, we define $\beta_i(K; A) = \operatorname{rank} H_i(K; A)$.

Problem 2 (A-Betti)**.**
Let A be an abelian group. The Betti number with coefficients in A problem is the total function problem defined from the language

$$A\text{-}\textsc{Betti} = \{\langle K, \{\beta_i(K; A)\}_{i=0}^{\dim K}\rangle : K \in \mathcal{K}\}.$$

Remark. The universal coefficients theorem shows us that integral homology determines homology with coefficients over any abelian group; however, it does not necessarily give us an efficient reduction from A-Betti to Betti, for some abelian group A.

We first look at $\mathbb{F}_2$-Betti. Let K be a simplicial complex with simplicial chain complex $(\{C_i\}_{i=0}^n, \{\partial_i\}_{i=0}^n\})$. We pick canonical bases for ∂_i and represent $\partial_i \in M_{|C_{i-1}|\times|C_i|}(\mathbb{F}_2)$. The Betti numbers. We then have $\beta_i(K; \mathbb{F}_2) = \ker \partial_i - \operatorname{rank} \partial_{i+1}$, so we can linearly reduce (in the number of simplices) this problem to the following.

Problem 3 ($\mathbb{F}_2$-RankNullity)**.**
Given $M \in M_{m\times n}(\mathbb{F}_2)$, what is the rank and nullity of M?

Algorithm 1 Efficient algorithm for $\mathbb{F}_2$-RankNullity

```
function CalcRankNull($m \times n$ bit array $M$)
    $N \leftarrow M$
    for $i \leftarrow 1$ to $\min\{m, n\}$ do
        if $N[i, i] = 0$ then
            if there is $a, b \geq i$ s.t. $N[a, b] = 0$ then
                exchange rows $i$ and $a$
                exchange columns $i$ and $b$
            else
                return $N$
            end if
        end if
        for $j \leftarrow i + 1$ to $n$ do                ▷ Clear row $i$
            if $N[i, j] = 1$ then
                add row $x$ to $k$
            end if
        end for
        for $j \leftarrow i + 1$ to $m$ do                ▷ Clear column $i$
            if $N[j, i] = 1$ then
                add row $x$ to $k$
            end if
        end for
    end for
    return $N$
end function
```

Algorithm 1 gives an efficient algorithm which solves $\mathbb{F}_2$-RankNullity. We now analyse the worst-case complexity of this algorithm, which occurs when we input a matrix of ones. At `3`, we loop $\min\{m, n\}$ times. In each loop, we have a full row and column to reduce, which is $m + n - 2i$ at each i.

3 Computation

Definition 4 (Codimension)**.** Let K be a n-simplex and L be an m-simplex such that L is a face of K ($m < n$). The *codimension* of L in K is the difference between the dimensions; that is, $\operatorname{codim}(L) = \dim(K) - \dim(L)$.

Like before, we consider a filtration $\{K_i\}_{i=0}^n$ for a simplicial complex K and impart that $K_0 = \varnothing$ and $K_n = K$. We now place a total ordering on the simplices of K, denoted $\sigma_1, \dots, \sigma_m$, in a way that it agrees with our filtration:

- a face of a simplex precedes the simplex; and
- a simplex in K_i precedes the simplices in K_j which are not in K_i for all $j > i$.

We now construct a square matrix $\partial \in M_m(\mathbb{F}_2)$ such that

$$\partial[i, j] = \begin{cases} 1 & \sigma_i \text{ is a face of } \sigma_j \text{ with codimension } 1, \\ 0 & \text{else.} \end{cases}$$

We now introduce a standard algorithm for the reduction of this boundary matrix to barcodes.

```python
def add_col_to_col(col_in: int, col_out: int) -> int: ...
def low(col: int) -> int: ...
def is_col_reduced(col: int) -> bool: ...

for i in range(len(bmat)):
    while not is_col_reduced(i):
        for j in range(0, i):
            if low(i) == low(j):
                add_col_to_col(j, i)
                break
```

The implementations of the top three functions are omitted, but their actions and return values are described. First, we understand this script expects a matrix `bmat`, a list of lists representing the square matrix ∂. `low` returns the largest row index of an element in a given column with a non-zero entry, and if the column is all zero it returns -1. `is_col_reduced(col)` evaluates true if `col` is the column with smallest index with its value of `low` or if `low(col)` $= -1$; otherwise, it evaluates false. We precisely define these functions below.

- `add_col_to_col` performs the expected column operation on `bmat`.
- $\texttt{low(col)} = \begin{cases} -1 & \text{if } \partial[i+1, \texttt{col}+1] = 0 \text{ for all } i, \\ \min\{i : \partial[i+1, \texttt{col}+1] \neq 0\} & \text{else.} \end{cases}$
- $\texttt{is\_col\_reduced(col)} = \begin{cases} 1 & \text{if } \texttt{low(i)} = \texttt{low(col)} \neq -1 \text{ for all } i < \texttt{col}, \\ 0 & \text{else.} \end{cases}$

We note that each of these functions can be implemented in $O(m)$ time. The exact implementation above runs in $O(m^3)$ time. The reason for the above implementation is to be clear on the exactness of the algorithm.

On to the main loop (line `5` onwards), we first claim that this terminates. Although not immediately obvious, our assumptions upon $\sigma_1, \ldots, \sigma_m$ from which we build ∂ (which `bmat` represents) assert that `is_col_reduced`(i) will evaluate true by repeating `7` to `10`. We have that `is_col_reduced`(j) is true for each $j \in \{0, \ldots i-1\}$. As `is_col_reduced`(i) is false, there is $i_1 < i$ such that `low`(i_1) $=$ `low`(i). So we add the i_1th column to the ith column, making the new value of `low`(i) strictly less than `low`(i_1) (as, by definition, every entry below `low`(i_1) is zero). To appreciate this, we recall that we are in $\mathbb{F}_2$. If `is_col_reduced`(i) is true, we are done; otherwise, there is $i_2 < i$ such that `low`(i_2) $=$ `low`(i) $<$ `low`(i_1). Again, we perform the appropriate column operation. We continue this operation, and as there are only a finite number of rows (that is, a finite number of unique values for `low`), then we must reach a point at which `is_col_reduced`(i) is false.

First, we learn to read the ranks of the homology groups of K. Let $R \in M_m(\mathbb{F}_2)$ be the resulting matrix from running the standard algorithm described above on ∂. Define

$$\begin{aligned} \text{zeros}_p(R) &= \{i : \text{column } i \text{ in } R \text{ is zero and } \dim \sigma_i = p\}, \\ \text{lows}_p(R) &= \{i : \text{low}(j) = i \text{ for some } j \text{ and } \dim \sigma_i = p\}. \end{aligned}$$

If $K_i = \{\sigma_j\}_{j=1}^{i}$, we claim that $\text{zeros}_p(R) - \text{lows}_p(R) = \beta_p$, which can be seen by observing that $\text{zeros}_p(R)$ is the rank of the p-cycles, and $\text{lows}_p(R)$ is the rank of the p-boundaries.

Next, we move to persistent homology. Let $f : K \to \mathbb{R}$ be defined on our simplicial complex as in the last section which induces our filtration, with associated sequence $\{a_i\}_{i=0}^{m}$. If $K_i = \{\sigma_j\}_{j=1}^{i}$ then $(a_i, a_j) \in \text{dgm}_p(f)$ if and only if $i = \text{low}(j)$ and $\dim \sigma_i = p$. Now we assume otherwise, so there is a step in the filtration in which more than one simplex is added. We simply construct the filtration $\{K_i'\}_{i=0}^{m}$ such that $K_i' = \{\sigma_j\}_{j=1}^{i}$ and let

$k' : \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$ be the sequence such that $K_i = K'_{k'(i)}$ for all i. Now define $k : \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$ as $i \mapsto \min\{i' : k(i') \geq i\}$. This is a mapping between our filtration and a filtration which only adds one simplex at a time when we move up. Let (i, j) be the indices of some point in the pth persistence diagram for K'. Then $(a_{k(i)}, a_{k(j)}) \in \mathrm{dgm}_p(f)$

3.1 Sparseness

So we have a cubic algorithm for calculating persistent homology in the number of simplices. Let us consider a triangulation of S^n: $K = \mathcal{P}(\{v_1, \ldots, v_{n+2}\}) \setminus \{\{v_1, \ldots, v_{n+2}\}\}$ and consider the ordering of simplices $\sigma_1, \ldots, \sigma_{2^{n+2}-2}$ as discussed, inducing the filtration $\{K_i\}_{i=0}^{2^{n+2}-2}$. From a computational perspective, we say that there is an exponential count of simplices, which is size of our boundary matrix. So, using the method above, we may calculate the persistent homology of the n-sphere in $O((2^{n+2} - 2)^3) = 2^{O(n)}$ time.

In practice, we may want to form our filtration from spaces with some notion of characterisation between data points. Precisely, let (M, d) be a metric space and we consider the Vietoris-Rips complex of some finite subset $S \subset M$ with diameter l,

$$\mathrm{VR}(S; l) = \{\sigma \subset S : \mathrm{diam}(\sigma) \leq l\}$$

where $\mathrm{diam} : \mathcal{P}(M) \to \mathbb{R}$ such that $S \mapsto \sup_{x,y \in S} d(x, y)$. This induces an ordering on the subcomplexes (consider Vietoris-Rips complex as we let l vary from 0 to ∞), which defines a filtration $\{K_i\}_{i=0}^{n}$ (where $K_0 = \varnothing$ and $K_n = K = \mathrm{VR}(S; \infty)$). We have

$$|K| = |\mathrm{VR}(S; \infty)| = |\mathcal{P}(S)| = 2^{|S|},$$

so we can calculate the persistence diagram of such a filtration in $2^{O(n)}$ time where $n = |S|$.

See below the boundary matrix for $\partial\Delta_2$ (homotopy equivalent to S^1).

$$\begin{pmatrix}
0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

This is a sparse matrix; that is, almost all of the entries are 0. This allows us to make a number of computational speed-ups. Recall the operations used in the implementation earlier, as we keep them in mind when considering a data structure.

We now show the sparse matrix representation from Edelsbrunner and Harer [1], and present an alternative.