Cubical complexes

Meggy-Lie-Anne Chamand, Ana Gelez

Introduction

When talking about homology and persistent homology, it is often assumed that we are studying simplicial complexes and their filtrations. However, in some areas, data is better represented with **cubical complexes**, in which simplices are replaced with cubes. For instance, it is the case when dealing with an image made of square pixels, or a video which can be seen as a three-dimensional image, where the third dimension is time.

More formally, a cube can be defined as a product of unit intervals. An unit interval is either degenerate ($[i, i], i \in \mathbb{N}$) or non-degenerate ($[i, i+1]$). A 2-dimensional cube (a square) spanning from $(1, 3)$ to $(2, 4)$ would then be represented as:

$$[1, 2] \times [3, 4]$$

Similarly to a simplicial complex, we can define a cubical complex as a set of cubes such that if a cube is part of the complex, then its boundary is part of it too.

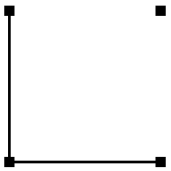


Figure 1: Example of a cubical complex

With this kind of complex too, it is possible to define homology and persistent homology, using similar definitions as for simplicial complexes.

Representing cubical complexes

We can take advantage of the structure of cubical complexes to represent them more efficiently. In a d-dimensional complexe, each cube can be represented as an element of $\mathbb{N}^d$. Each element of this vector corresponds to one dimension. If the coordinate is even, the corresponding interval in the product is degenerate, and if it is odd it is non-degenerate. The interval corresponding to a coordinate x starts at $x \div 2$ (using euclidian division).

The complex in Figure 1 can thus be represented with this set (considering the y axis points down and that $(0, 0)$ is in the upper-left corner):

$$\left\{\binom{0}{0}, \binom{0}{2}, \binom{2}{0}, \binom{2}{2}, \binom{0}{1}, \binom{1}{2}\right\}$$

This representation is much more compact and less ambiguous than listing the vertices that are part of each cube.

Computing homology

It is then possible to compute homology, and more generaly persistent homology. To do so, we implemented the algorithm described by Wagner *et al.* [1].

Using a filtration, that is represented as function that assign to each vertex an index, we build a generalization of this filtration, that assigns an index to all cubes in the complex. Cubes can then be sorted according to their assigned index. The order of the vertices is preserved, and a non-vertex cube is garanteed to be after all its vertices.

We use this ordering to build a boundary matrix: each column corresponds to a cube in the complex (the i-th column corresponds to the cube with filtration index i to be exact). The j-th element in this column is 1 if the cube with filtration index j is part of the boundary of the i-th cube (otherwise, it is 0).

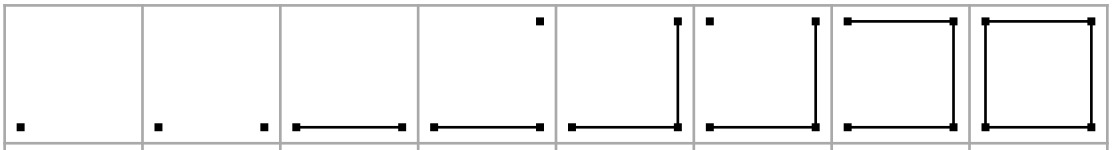
Using the representation described above, it is easy to compute the boundary of any cube: for each coordinate x, if it is non-degenerate, the boundary includes two cubes that both share the same coordinates as the cube excepted for x which is replaced with $x - 1$ and $x + 1$ respectively.

Afterwards, the matrix is reduced, using the algorithm proposed by Chen *et al.* [2]. We can then exploit the equations provided in this same article to build persistence pairs.

Results

A simple example

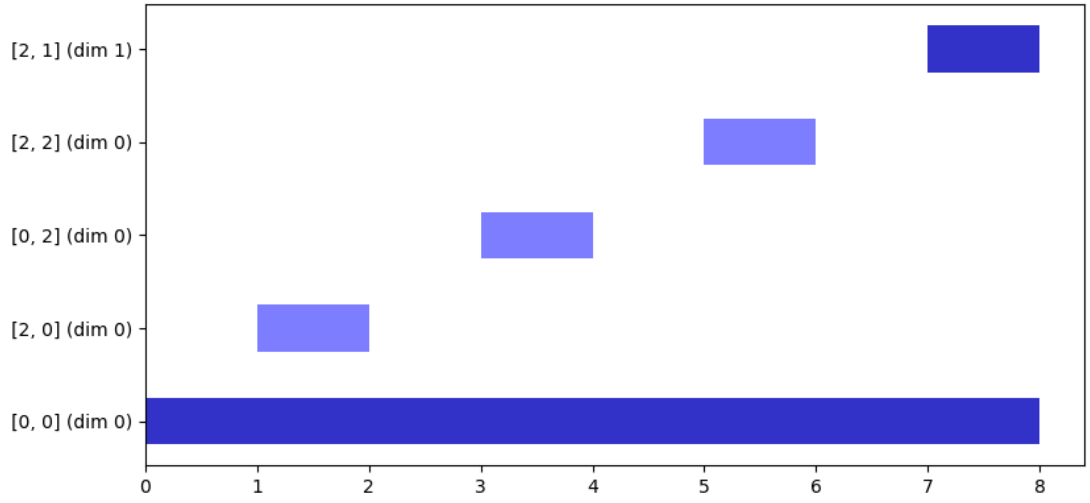
We decided to test our code on a simple example, drawn below (the number below each step is the filtration index of the cube that is added at this step):



Our algorithm gives the following boundary matrix, which is then reduced:

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

If we build the homology pairs and plot them on a bar diagram, this is what we obtain.



The dark blue bars are homology generators: the longest one is for the 0-dimension generator (one connected component) and the short one is for the 1-dimension generator (the loop that appears at the last step). The light blue bars correspond to the addition of a vertex followed by the addition of a line: a new connected component is created and then immediatly killed.

Some classic examples

For these complexes, we used data from SageMath that we converted to our own format. Here are the bar charts that we obtain:

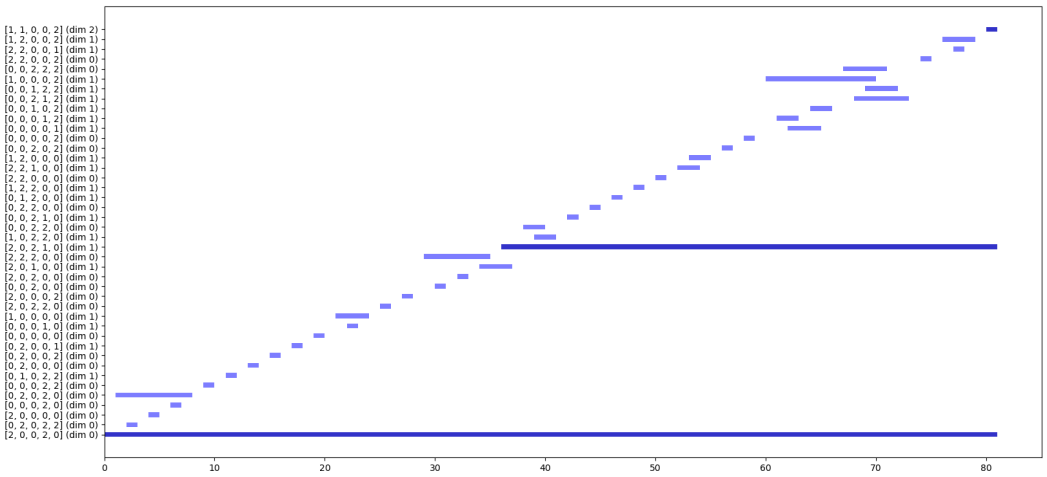


Figure 3: Projective plane bars

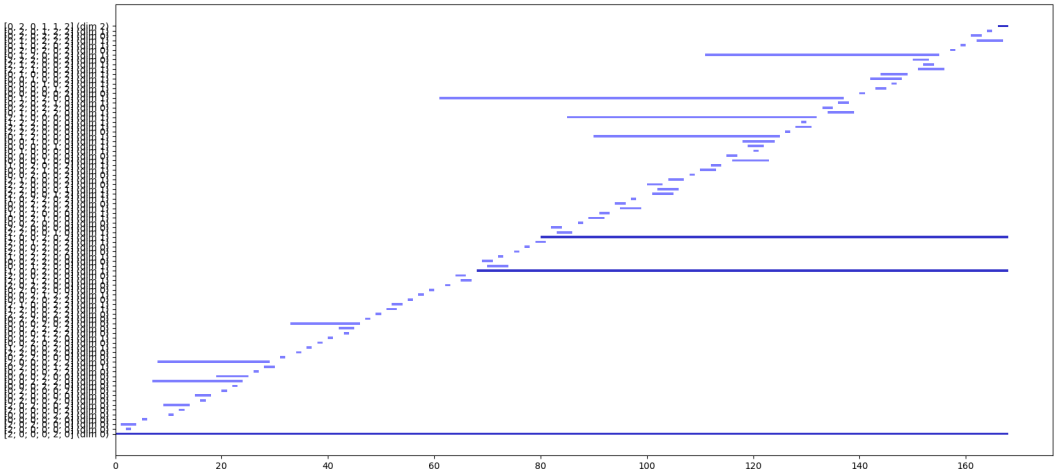


Figure 4: Klein bottle bars

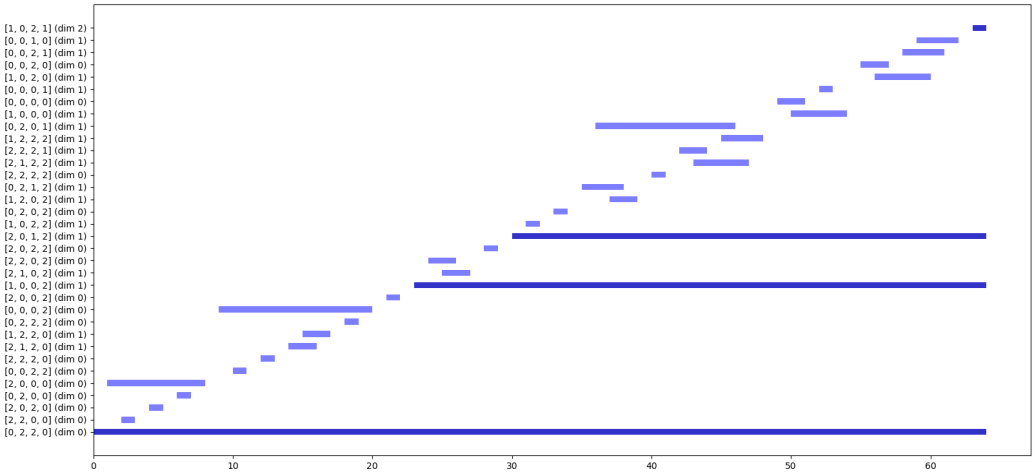


Figure 5: Torus bars

We can see that the number of generator is what we would expect. For instance, the torus has one generator in dimension 0, two in dimension one, and one in dimension two.

Discussion

Out algorithm seems to correctly compute persistent homology for cubical complexes.

Unfortunately, we didn't have enough time to implement the persistent homology algorithm using various coefficients. Our code only supports $\mathbb{Z}_2$.

Using other coefficients would require anoter definition of boundaries that is more general (for instance, when working in $\mathbb{Z}$, cubes in a boundary are oriented and not just "present" or "absent"). The reduction algorithm should also be adapted to work in a different field, when summing columns together.

References

[1] H. Wagner, C. Chen, and E. Vuçini, "Efficient Computation of Persistent Homology for Cubical Data", in *Topological Methods in Data Analysis and Visualization II: Theory, Algorithms, and Applications*, R. Peikert, H. Hauser, H. Carr, and R. Fuchs, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 91–106. doi: 10.1007/978-3-642-23175-9_7.

[2] C. Chen and M. Kerber, "Persistent Homology Computation with a Twist", 2011. [Online]. Available: https://api.semanticscholar.org/CorpusID:18232748