

FINAL PROJECT

Due date: May 12.

The topic of the final project is flexible. You are expected to hand in a .pdf report describing what you did and presenting your results. You can also email supplemental material (e.g. code) in a zip file, if applicable. But the .pdf should be self-contained, in the sense that I should be able to read and grade the project without looking at any supplemental material.

Please clear your project idea with me.

Below are some suggestions for projects, some of which are rather open-ended. For some of the suggestions below, let me warn you that if you are not comfortable with a programming language (e.g. Python), you may need spend to a lot of time getting up to speed with this.

Partner Work. You are welcome to work with one partner, or to work alone. If two people work together, my expectations will be somewhat higher for your project, since you will have twice the manpower.

Below are some ideas for projects.

Project 1: Efficient Implementation of Persistence Computation.

This project has two related parts: First, you will write code which takes the pair of column-labeled matrices $[d^{i+1}]$, $[d^i]$ associated to the chain complex of a finite simplicial filtration F and compute the barcode of the associated homology module $\ker d^i / \text{im } d^{i+1}$. The grades of appearance of simplices in F can be assumed to be natural numbers. You can assume that the matrices are given with respect to an order on simplices such that $\sigma < \tau$ whenever $b(\sigma) < b(\tau)$. (Thus the labels of the columns of each $[d^i]$ are in increasing order.) You should assume that the field is $\mathbb{Z}/2\mathbb{Z}$.

You will implement the column-wise persistence algorithm both with and without clearing. You will then also implement the row-wise persistence algorithm with and without clearing. For these computations, you will use row-sparse and column-sparse representations of the columns, respectively. I suggest you use a dynamically allocated array to represent the columns, though you can also use a linked list. I suggest that you do not construct your own data structure for this, but rather use a standard implementation in your chosen programming language.

Note that you will need to represent the matrix differently depending on whether you are doing row or column operations.

I will provide some test examples in the next few weeks, as a text file. Each matrix will be given in a text file, in some sparse format. For each

test example, you will return the barcodes of these and also report timing results for each of the four computational approaches.

The second part of the project will be analogous, except you will implement minimal presentation computation for bipersistence modules, as explained in my paper with Matthew Wright "Computing Minimal Presentations and Bigraded Betti Numbers of 2-Parameter Persistent Homology," <https://arxiv.org/abs/1902.05708> for 2-parameter persistence modules. For this, you will again be given column-labeled matrices $[d^{i+1}]$, $[d^i]$ associated to the chain complex of a (1-critical) finite simplicial bifiltration F . This means that each column will be labelled by an element of \mathbb{N}^2 . The column grades will be in colexicographical order, i.e., ordered by y-coordinate, with ties broken by x-coordinate. Again you will compute timings, and compare these to the timings in the 1-parameter case. (Each data set I provide in the 1-parameter case will have a 2-parameter counterpart.)

I will accept a submission that completes only the first part of this project for 2/3 credit.

Project Idea 2: Data Analysis. Use 2-parameter persistent homology to perform an analysis of some data. In practice, this likely means using RIVET, available at repo.rivet.online, and perhaps also the Python API for RIVET at <https://github.com/rivetTDA/rivet-python>. However, you are not required to use RIVET.

The analysis of the data may be exploratory; you do not necessarily need to perform any rigorous statistical analysis on the data. You do not necessarily need to make an exciting scientific finding.

However, for you to receive a good grade on this project, I will want to see a good narrative about the data analysis, and evidence of a sincere effort to look for interesting structure. You will likely want to analyze several data sets, not just a single one.

It may also be appropriate to run other analyses on the same data to complement the persistent homology analysis. For example, to get intuition about data, it is sometimes useful to run multidimensional scaling or PCA to linearly embed the data into a 2 or 3 dimensional space. If you are looking for cluster structure in data, you may wish to also run other standard cluster analyses, such as single-linkage, average-linkage, or spectral clustering. Code for all of this is not available in RIVET, but is readily available elsewhere, e.g., for Python.

It may be appropriate for you to address some or all of the following questions: Why was the data chosen? What was the underlying question or hypothesis the analysis sought to address? What methodological issues, if any, were encountered, and how were they addressed? What were the findings of the analysis, and what conclusions, if any, were you able to draw? Does the analysis suggest any directions for future inquiry?

One potential source of data is The Cancer Genome Atlas (TCGA). There is a lot of gene expression data for different cancers. Biologists often cluster

such data and use this clustering as a clinical basis for diagnosing cancer subtypes. It could be interesting to run RIVET with 0th homology (clustering) on some such data. (I have never worked with a raw data set from TCGA, however, and I cannot comment on what data cleaning/preparation you would have to do with this data.)

Project 3: Extend/Improve some part of the RIVET code. This is probably only appropriate for someone with ample C++ experience. There are many options for this. See, e.g., the issue tracker at repo.rivet.online.

Project 4: Exposition. Do a literature review on some topic of interest related to the course. Write up an expository review at a suitable level of detail. This should be a minimum of 10 pages, and should reflect some independent thought about the material, not just regurgitate statements of results. It may be appropriate to identify open problems / directions forward, identify barriers to progress, or compare approaches/contributions of different papers to one another.

Project 5: Theory. Develop some novel bit of theory related to the course.