flagser-laplacian Documentation

Ben Jones

September 2023

1 Introduction

flagser-laplacian computes the spectra of the Persistent Directed Flag Laplacian, defined on direfect flag (clique) complexes. It relies on substantial portions of the software flagser, by Daniel Lütgehetmann [1], to build the directed flag complexes and (co)boundary matrices.

This software takes as input a (filtered) directed graph and outputs the computed spectra. It does not produce a directed graph from point cloud data.

2 Requirements

flagser-laplacian requires a C++17 compiler, CMake, and a MATLAB installation.

This software uses the MATLAB Engine API for C++, may be difficult to use in a Windows environment.

This software also uses the C++ library Eigen for some matrix computations. A full copy of Eigen is included in include/Eigen.

3 Installation and building

You can install flagser-laplacian by first cloning the repository:

```
git clone https://github.com/bdjones13/flagser_laplacian
```

Important: the MATLAB directory in CMakeLists.txt (lines 21 and possibly lines 27 and 33) must agree with your MATLAB installation.

Now you can compile the source code from within the repository:

```
> mkdir build
> cd build
> cmake ..
> make
```

This will produce the executable file `flagser-laplacian`. If you want to use flagser-laplacian from a different directory, you may want to add the directory containing the executable file to your path.

4 Usage

4.1 Input data

The input data file for flagser-laplacian must be in the flagser format:

```
dim 0:
filtration_vertex_0 filtration_vertex_1 ... filtraition_vertex_n
dim 1:
first_vertex_id_of_edge_0 second_vertex_id_of_edge_0 filtration_edge_0
first_vertex_id_of_edge_1 second_vertex_id_of_edge_1 filtration_edge_1
...
first_vertex_id_of_edge_m second_vertex_id_of_edge_m filtration_edge_m
```

Note that the filtration values are not optional. If you do not want to use a filtration, use a filtration value of 0.

4.2 Running flagser-laplacian

```
./flagser-laplacian [options] datafile.flag
```

where datafile.flag is a file in the flagser format described above.

Here is a description of the possible options:

–out-prefix prefix: is an optional parameter that will add `prefix` to the beginning of the output file names. This is useful when making multiple calls to flagser-laplacian.

–max-dim dim: is an optional parameter that will limit the dimension of the spectra to be computed.

Here are some examples of ways to call flagser-laplacian

- `./flagser-laplacian a.flag`
- `./flagser-laplacian test/a.flag`
- `./flagser-laplacian --out-prefix myprefix test/a.flag`
- `./flagser-laplacian --max-dim 2 test/a.flag`
- `./flagser-laplacian --max-dim 2 --out-prefix myprefix2 test/a.flag`

Other options that flagser implemented are not currently available in flagser-laplacian.

4.3 Output

The number of output files will depend on the directed graph you input and if you specified a maximum dimension. There will be an output `prefix_spectra_i.txt` for each dimension i of the directed flag complex, up to the maximum dimension you specify. There will also be an output `prefix_spectra_summary.txt`.

The format for a file `prefix_spectra_i.txt` will be of the form:

```
>
>
> 1
> 0 0 2
> 0 0 0 0.1 1.0
```

Each line represents the spectra of $\Delta_i^{a,b}$, where a is the filtration value corresponding to the row number, and b is the next filtration value. For example, if the filtration is 0, 1.5, 2, 3, the 2nd line of the file will report the spectra for $\Delta_i^{1.5,2}$. **Note:** the final line of each `spectra_i.txt` file is equal to the spectra of Δ_i^b, where b is the final filtration value, e.g. 3 in the above example. This final step is achieved by inserting an artificial final filtration value of $b+1$, and since the directed flag complexes at b and $b+1$ are equal, we will have $\Delta_i^b = \Delta_i^{b,b+1}$. Observe that the first lines may be blank if there are no directed i-cliques at that filtration step. The spectra are printed in non-decreasing order.

The other output file, `prefix_spectra_summary.txt`, is in tab separated value format. The headers are i (filtration index starting from 0), `filtration` (the real number filtration values), `betti_0`, `betti_1` ... `bett_N` (persistent Betti numbers up to the top dimension of the complex or the specified maximum N), `lambda_0`, `lambda_1` ... `lambda_N` (least nonzero persistent eigenvalues). If there are no nonzero eigenvalues, at that filtration level, `lambda_i` is taken to be 0. It might look like this:

```
i   filtration  betti_0 lambda_0
0   0   1   0
1   0.1 2   0.23
```

Note that each tab separator is a tab. Depending on the software you use to view this file, it may not appear as aligned columns.

References

[1] D. Lütgehetmann, D. Govc, J. P. Smith, and R. Levi. Computing Persistent Homology of Directed Flag Complexes. *Algorithms*, 13(1), 2020.