Daniel Müllner Spring 2012

Applied Algebraic Topology
Suggestions for mini-projects

The mini-projects are designed for groups of about three students. Larger groups (up to five students) may be permitted if the topic is extended somewhat in scope or depth. For example, an additional group member could focus on the theoretical background, search for literature and provide the rest of the group with predictions which homology groups should be expected in the experiments.

Final reports are expected to have around 10-15 pages (or slightly more if there are lots of figures—a feature which is most welcome!). The reports are due **June 7**.

1. Persistent homology of letterforms

Investigate whether/how all characters of the Latin alphabet can be distinguished by ordinary and persistent homology. For example, the letters A and B can be distinguished by ordinary homology since they differ in the number of loops, whereas P and D have both the homology of a circle but are distinguishable by persistent homology.

What are suitable persistence functions? E. g. the height function or negative height function? Or is it better for some comparisons to treat the shapes like point clouds, cover them with growing disks and compute persistence of the Čech/Rips/Alpha complex? More ideas?

Can you distinguish all 26 letters from each other? Additionally, take many variants of the letters and see how well your algorithms can classify letters.

i) One group can treat handwritten letters/digits. Start now by sampling alphabets, scan the images and bring the data into a format that you can work with in MATLAB.

ii) A second group can study letterforms from different fonts. Again, make bitmaps and organize them so that you can easily process them once you learn to use the persistent homology software in the course.

iii) There are many more alphabets! Cyrillic, Hebrew, Thai, just to name a few. Especially if there is a native speaker in the group, you are invited to study your favorite alphabet and explain the characteristics and specific difficulties in your report. Chinese and Japanese characters are recommended only if someone can recommend and justify a reasonable subset.

In all cases, it is expected to investigate the letterforms by persistent homology software and report the experimental findings.

2. Simplification of complexes

Homology computations often suffer from memory constraints; on the other hand, simplicial complexes from point cloud data usually contain many more samples than are actually needed to represent the homological features.

This project is to investigate how well reduction techniques can simplify simplicial complexes without changing their topology.

We will highlight in the course that homology groups do not change under homotopy equivalences. A particularly simple homotopy equivalence is an *elementary collapse*: remove a pair of faces (σ, τ), where τ is the single coface of σ. Write an algorithm that finds these pairs and removes them from the complex, until no further reduction is possible.

However, it is known that not every homotopy equivalence can be achieved by elementary collapses, so it is expected that this pruning process stops before a simplicial complex gets into the simplest possible form. Your team should investigate heuristically what reductions in size are possible. How does the order of elementary collapses influence the result? Are there better or worse strategies?

Also, you should spend some thoughts on the choice and generation of the complexes that you are trying to reduce. Which ways are there to generate complexes randomly? Do you want to generate abstract complexes or generate from a geometric model? You may also generate Rips complexes for point cloud data; think about what distribution you would like to sample the point clouds from.

Can you generate interesting complexes which are contractible (i. e. homotopy equivalent to a point) but not trivial cases like trees or full simplices? Even if it might be hard to produce interesting contractible complexes, you could estimate the "complexity" of any simplicial complex by its Betti numbers and measure the success of the reduction techniques in relation to the Betti numbers.

As a possible extension, you could move on from elementary collapses to collapsing simplices: Here you map all vertices of a certain simplex in the data set to a single point. In this way, one simplifies a higher-dimensional simplex to a single point. Again, this step is not always possible without changing the topology, so again it is interesting how far one can go with different strategies.

This project does not explicitly involve persistent homology, only ordinary homology, so you can start work on it immediately.

3. Homology of $SO(n)$ and quotients of $SO(3)$

i) Find a way to sample orthogonal matrices as uniformly as possible in the vector space of all real $(n \times n)$-matrices. When the project is finished, I would encourage you to publish your code on the CompTop homepage, providing other researchers with useful ways of sampling point cloud data from the groups $SO(n)$, as examples of synthetic data sets with interesting topological structure.

Once you can sample from $SO(n)$ (and this is something you can start immediately), treat the samples as a point cloud and try to find out the homology of the special orthogonal groups with persistent homology. It might be difficult to get all homology groups for the higher-dimensional groups from $SO(4)$, but $SO(3)$ should be feasible. Try how high you can get, both in terms of homology degrees and in the dimension of $SO(n)$.

Possible extensions: find out about the homology groups of $SO(n)$ in the literature, how they are calculated, and compare the result to the heuristic findings via persistent homology. Moreover, there are more matrix groups one can sample from: the unitary groups, special unitary groups, symplectic groups,...

ii) Another project might not try to cover the higher $SO(n)$ but quotients of $SO(3)$ by finite subgroups. For example, the rotational symmetries of a regular tetrahedron form a finite subgroup of all rotations in 3-space. Consider two rotation matrices as equivalent if one is the product of the other with a tetrahedral symmetry. Again, sample from the quotient space of $SO(3)$ and compute persistent homology. Also try the symmetries of a cube and a dodecahedron.

4. Ordinary and persistent homology of lens spaces

3-dimensional lens spaces are 3-manifolds which occur as quotients of the sphere S^3 in a simple way. Students working on this project should try first to construct triangulations of a few lens spaces. That means, represent a lens space by a simplicial complex. Hopefully, you find a general recipe to triangulate the lens spaces that works for all of them. From the triangulation, you can compute the homology groups exactly, so do this with homology software. Afterwards, sample points from the lens spaces of your choice and test how well persistent homology can recover the homology groups from samples.

5. Quotients of $SU(2)$ by finite subgroups

The matrix group of special unitary (2×2)-matrices can be identified with the sphere S^3 and the quaternions of unit length. Inside the quaternions is the discrete group of 8 elements $\{\pm 1, \pm i, \pm j, \pm k\}$. Similar to the quotients of $SO(3)$ and the lens spaces, compute the persistent homology of the quotient space from point samples. Feel free to decide in which direction you like to extend this: try to triangulate the quotient space and compute homology exactly, or consider more finite subgroups of $SU(2)$.

6. Statistics on persistence diagrams

At a recent conference, Peter Bubenik (CSU Ohio) presented a new way to draw persistence diagrams. The twist with this new representation is that they can easily be added, and statistical quantities like mean and variance of a collection of persistence diagrams can be computed. This new method can be explored empirically: generate many persistence diagrams from point clouds drawn from a distribution and study their mean etc. Does averaging persistence diagrams help with noisy data? Can one distinguish different shapes by their persistence signatures?