Computing Homology of Hypergraphs

Jackson Earl[1], Emily Heath[2], Cliff Joslyn[3], Brenda Praggastis[3], Emilie Purvine[3]

[1] Illinois State University
[2] University of Illinois at Urbana-Champaign
[3] Pacific Northwest National Laboratory

Background

A ***hypergraph*** is defined to be a collection of subsets *E*, called *edges*, of a set of elements *V*, called *vertices*, and is written as *H = (V, E)*.

Hypergraphs have the ability to capture multi-way relationships that typical graphs used in data science cannot. Graphs can only code pairwise associations between entities, whereas a hypergraph can represent several multi-way relationships between an arbitrary number of entities. It should be noted that hypergraphs generalize graphs in this way [1].

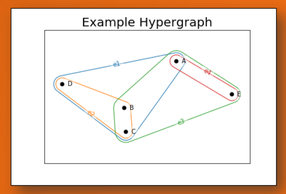

Figure 1. An example of a hypergraph generated using HyperNetX in python.

Motivation

A ***homotopy*** between two continuous functions f and g from a topological space X to another space Y is defined as a continuous function $H: X \times [0,1] \to Y$ such that for all $x \in X$ we have $H(x,0) = f(x)$ and $H(x,1) = g(x)$. We think of H as a continuous deformation of f into g.

Retrieved from https://i.stack.imgur.com/qa0Ja.jpg

Figure 2. A typical schematic demonstrating the homotopy equivalence of a coffee mug and the torus. The torus has Betti numbers $\beta_0 = 1$, $\beta_1 = 2$, and $\beta_2 = 1$.

Algebraic topology is noted as a sub-field of topology that is concerned with algebraic properties of spaces that respect homotopy. Algebraic topology views two spaces as "the same" if they are homotopy equivalent. Well, homology classes and Betti numbers are topological invariants that can tell us whether or not two spaces, such as two hypergraphs, are homotopy equivalent. It is for this reason that these metrics are of key interest.

Methods

To develop a function in python to compute homology required some methodology and gathering insights:

1) Use prior python code written in HyperNetX to articulate updated versions.

2) Test function by comparing results to known homology groups and Betti numbers (figure 3).

3) Explore how Betti numbers change according to other metrics like density (figure 4).

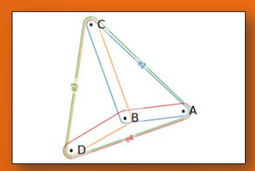

Figure 3. A hollow tetrahedron with four 2-simplexes as its faces. It has known Betti numbers $\beta_0 = 1$, $\beta_1 = 0$, and $\beta_2 = 1$.

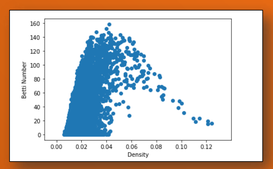

Figure 4. Using NetworkX's random bipartite graph generated, it is shown how Betti numbers are distributed according to the density of hypergraphs.

Abstract

In the modern age of data science, the necessity for efficient and insightful analytical tools that enable us to interpret large data structures inherently presents itself. With the increasing utility of metrics offered by the mathematics of hypergraph theory and algebraic topology, we are able to explore multi-way relational datasets and actively develop such tools. Throughout this research endeavor, one of the primary goals has been to contribute to the development of computational algorithms pertaining to the homology of hypergraphs. More specifically, coding in python to compute the homology groups of a given hypergraph, as well as their Betti numbers have both been top priorities.

Homology Groups and Betti numbers

Computing the homology groups and Betti numbers of a hypergraph is an extensive process, and by no means can it efficiently be done by hand, especially in the case of very large hypergraphs. The general steps with definitions are outlined below:

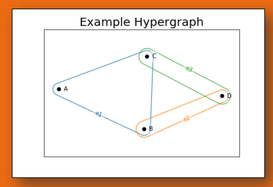

Figure 5. An example of a hypergraph generated using HyperNetX in python.

Step 1. We take the structure of a hypergraph and realize it geometrically as an abstract simplicial complex.

An ***abstract simplicial complex*** is a family K consisting of a finite collection of subsets, or *simplicies*, of a given set X such that if $\sigma \in K$ and $\tau \subseteq \sigma$, then $\tau \in K$.

A k-***simplex*** is an ordered $(k+1)$-tuple of affinely independent points $\sigma = (x_0, \dots, x_n)$. The points in the set $V = \{x_0, \dots, x_n\}$ are the **vertices** of the simplex σ.

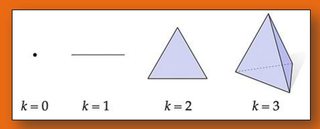

Figure 6. The first four k-simplexes.
Retrieved from http://brickisland.net/DDGFall2017/wp-content/uploads/2017/09/CMU_DDG_Fall2017_02_SimpicialComplex-1.pdf

Step 2. We define the chain groups with respect to the k^{th} Betti number we are interested in.

A k-**chain** is defined to be a formal sum of k-simplicies. The set of k-chains with formal addition over $\mathbb{Z}$ create a $\mathbb{Z}$-module, denoted C_k and called a chain group, with a basis given by the k-simplicies of K.

Step 3. We define a homomorphic boundary map, $\partial_k: C_k \to C_{k-1}$, for each chain group, which can be represented as a matrix.

The ***boundary map*** is given by the operator $\partial_k(\sigma) = \sum_{i=0}^{k}(-1)^k(x_0, \dots \hat{x_i} \dots, x_k)$ where $(x_0, \dots \hat{x_i} \dots, x_k)$ is the $(k-1)$-face of σ.

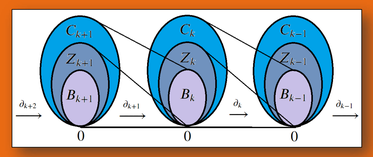

Figure 7. A schematic representing the mapping between chain groups.

The boundary operator presents a sequence of connected chain groups called a ***chain complex*** such that $\dots C_{k+1} \xrightarrow{\partial_{k+1}} C_k \xrightarrow{\partial_k} C_{k-1} \xrightarrow{\partial_{k-1}} \dots$ where $\partial_{k-1} \circ \partial_k = 0$ for all $k > 1$.

Step 4. We compute two important sub-groups called ***cycles*** and ***boundaries*** in terms of their boundary maps.

The k-cycles and k-boundaries are defined as $Z_k(X) = \ker \partial_k$ and $B_k(X) = im\ \partial_{k+1}$ respectively.

Step 5. We can now compute the homology groups and Betti numbers!

The k^{th}homology group is denoted $H_k = Z_k / B_{k+1}$.

The k^{th}Betti number is defined as $\dim H_k = \dim Z_k - \dim B_k$.

Applications to IMDB Dataset

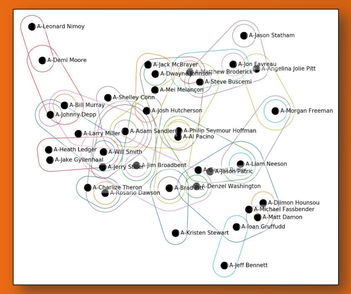

Figure 8. A sample of the IMDB dataset where edges represent directors and the vertices represent actors. The full dataset can be found at https://www.kaggle.com/carolzhangdc/imdb-5000-movie-dataset

After developing a function to compute the Betti numbers of basic hypergraphs, it was tested on actual datasets. Datasets of particular interest are ones that carry multi-way relationships such as IMDB movie data. Relationships between directors and actors can be easily be modeled with a hypergraph, for example. And thus, we have a means of computing homology. A sample of the IMDB dataset is displayed (figure 8).

The Betti numbers computed for this sample dataset are $\beta_0 = 1$, $\beta_1 = 2$, and zero otherwise. To compute the Betti numbers of the whole dataset requires some refinement of the code and a diagnosis of its computational efficiency. For larger datasets, the code often runs into memory errors.

Future goals

1. Understand how Betti numbers correlate to other metrics of hypergraphs such as inclusivity, aspect ratio, degree centrality, etc.

2. Investigate the applications of Smith Normal Form, Singular Value Decomposition, Principal Component Analysis, and other algorithmic approaches to computing homology.

3. Learn exactly how this code handles larger and larger hypergraphs to determine its overall efficiency.

References

[1] Purvine, E., Aksoy, S., Joslyn, C., Nowak, K., Praggastis, B., Robinson, M. (2018). A Topological Approach to Representational Data Models. *Lecture Notes in Computer Science*, volume 10904, 90-109. https://link.springer.com/chapter/10.1007/978-3-319-92043-6_8

Acknowledgements

The 2019 STEM Teacher and Researcher Program and this project have been made possible through support from Chevron (www.chevron.com), the National Science Foundation through the Robert Noyce Program under Grant #1836335 and 1340110, the California State University Office of the Chancellor, and California Polytechnic State University in partnership with Pacific Northwest National Laboratory and the Department of Defense. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funders.