Topological Data Analysis on Libraries.io Data

Brian Friend, Jacob Miller, Jonathan Anderson, Kaixiang Wang

December 18, 2017

Contents

1. Background
   - The Data
   - Topological Data Analysis
2. Methods
   - Data Storage
   - Shell Scripts
   - Software Exploration
   - Cytoscape
   - TDA for R
3. Analysis
   - Traditional TDA

What data?

Package Managers & libraries.io:

- project goal: how are open-source projects connected?
- libraries.io monitors millions of open-source project libraries and dozens of package managers
- keeps track of dependencies, version information, etc.
- libraries.io's main file had millions of lines of this information
- we looked at subset of that data: `CPAN`, `CRAN`, `Brew`, `Dub`

TDA

- Starting out with the high dimensional point cloud
- unknown lower-dimensional structure
- TDA is a collection of methods for "teasing out" topological structure

Definition

Topology is the study of spaces and the features that describe spaces.

Definition

Topological Data Analysis (TDA) is the practice of analyzing the spatial properties of data sets.

Spatial Properties

Properties include the following:

- Connectedness
- How a space can be separated
- The types of holes found in a space

What this means in data terms:

- How many relations are there between data points
- How can data be categorized
- How do the dimensions of data interact

Initial Data Set

Accessing the data:

- Reproducibility using centralized location for data: https://libraries.io/data
- Downsides of Libraries.io's API: formatting, readability, etc.
- Benefits of downloading it ourselves: flexibility, managing data size, filtering data

Extracting Relevant Data

Size & Filtering:

- Size problems:

  compressed data was 5.9 GB,
  and uncompressed was 33 GB

- Focused on "`dependencies.csv`"
- Python wrapper `process.py` separates data into different files
- Scripts for stripping out redundant, missing information

Problems with non-TDA software:

- Limitations of `Python`
- Poor `R` documentation
- Summary of issues:
  - maturity of tools
  - speed
  - memory usage
  - graphical flexibility
  - monitering execution

Explanation & Justification for Cytoscape:

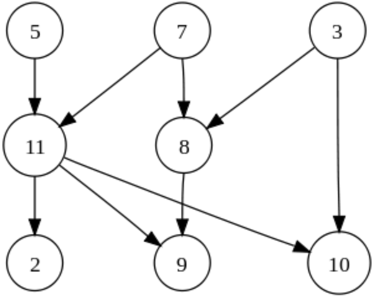
- previous software problems all largely solved by using CytoScape
- maturity: online community, documentation that doesn't require a PhD
- memory usage: larger graphs require more memory but Cytoscape actually lets you use it
- graphics: beautiful and extremely elegant graphing utilities
- micromanagement: change data representation on the fly
- Conclusion: using already developed software allowed us to get to exploring dependency information without reinventing the wheel

For comparison's sake and for quantifying relationships, we're in the rough stages of integrating R's TDA package..

- https://cran.r-project.org/web/packages/TDA/vignettes/article.pdf
- "provides topological information about the underlying space, such as the distance function, the distance to a measure, the kNN density estima- tor, the kernel density estimator, and the kernel distance. The salient topological features of the sublevel sets (or superlevel sets) of these functions can be quantified with persistent homology."

Graph Analysis is essential for TDA

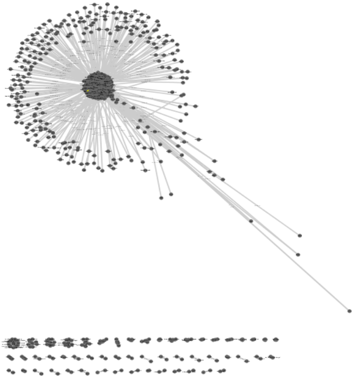
- Graph is a direct virtual way to help people build understanding to a data structure.



There are important parameters to help people analysis the graph.

- Vertices
- Edges
- Orientation

Example graphs from sample Data Dub.csv



Example graphs from sample Data Dub.csv