

# Model Exploration

- final report -

Guy Arad, 303015853

Tamir Denis, 208538702

Yaron Segev, 303067961

Roei Shenberg, 200113041

Alexander Shugaley, 320827694

School of Computer Science, Tel Aviv University, January 2018

## Project High-Level Overview

- A web-app that receives a complex state model and generates a simple, logical, interactive and human readable representation of it.
- Essentially, a user uploaded a .dot graph, selects a Clustering Algorithm (5 implemented), it's parameters, and a labeling scheme (5 implemented). The application then creates a dendrogram and supergraph, sorting the nodes into groups of similar context. Those are presented to the user, and he/she can zoom in and out to find the preferred abstraction.
- Major modules:
  - **Web-app** (user interface)
  - **Engine** (for algorithmic operations)
  - **Tests**

## Major External Libraries & Dependencies

- **Networkx** - graph class; some clustering algorithms (e.g. Spectral Clustering) auxiliary graph manipulation algorithms (e.g. Dijkstra, min-spanning-tree).
- **Scikit** - some clustering algorithms (e.g. k-means).
- **D3.js** - Graph & dendrogram visual presentation in the web-app.
- Additional - Flask, PyYAML, NumPy, Matplotlib, pydot, pandas, pytest, GEM, beautifulsoup4.
- Full requirements and version info available in README.txt

## Engine

- The majority of the algorithmic logic is in this module. It includes numerous sections:
  - **Cluster** (interface) - and the algorithms (all implement and extend the interface): Spectral Clustering, K-means, Branch & Bound (implemented by us, not from an external library), Minimum-Cut, Kernighan-Lin Clustering.
  - **Labeling** (interface) - and the schemes (all implement and extend the interface): PageRanker, Prefix, tf-idf, Path, Random Walk. All implemented by us.
  - **Basic Entities** - Graph and Dendrogram classes.
  - **Linear Programming Solver**
  - **Stopping-Criteria** - a module with a few classes that calculate the stopping condition of the clustering.
  - **Utils** - auxiliary components, includes the FileWriter - creates the json that we send to the web-app.
  - **Main** - includes the high-level flow functions that link all the engine components together.

## Web-app

- **Fronted** - implemented with JQuery, with the D3.js library for representing the graph and the dendrogram.
- **Backend** - Flask web framework is used for backend (python).
- Information from the engine is transferred with in JSON format.
- Web-app is highly modular, as the majority of the info is received from the engine. For example, if you add an additional algorithm to the engine, there's no need to update the web-app code - the information will be received in the JSON and parsed correctly.

## Tests & Debugging

- We have test for most, if not all, of the different modules/classes of the engine. Those tests can be run together as explained in README.txt.
- Pytest tool is used to run tests.

Milestone	Description	Due Date	Progress
Gathering user/ system requirements	<ul style="list-style-type: none"> <li>Meeting Nimrod for project description</li> </ul>	06/11/17	✓
Generating a project mockup	<ul style="list-style-type: none"> <li>Static html</li> <li>Basic classes + interfaces</li> </ul>	13/11/17	✓
Implementing basic engine functionality	<ul style="list-style-type: none"> <li>Two clustering algorithms</li> <li>Naive labeling</li> </ul>	27/11/17	✓
Support end-to-end modularization and exploration	<ul style="list-style-type: none"> <li>Go through entire flow without placeholders</li> </ul>	04/12/17	✓
Web-app model projection	<ul style="list-style-type: none"> <li>Draw Abstract Model</li> <li>Model abstraction</li> </ul>	11/12/17	✓
<ul style="list-style-type: none"> <li>Additional clustering algorithms</li> <li>additional user interface tools</li> </ul>	<ul style="list-style-type: none"> <li>Branch and Bound and GEM using graph embedding.</li> <li>User will be able to select a cut from dendrogram</li> </ul>	25/12/17	✓
<ul style="list-style-type: none"> <li>advanced labeling</li> <li>Final testing</li> <li>Deployment</li> </ul>		8/1/17	✓