Algorithmic Methods of Data Mining

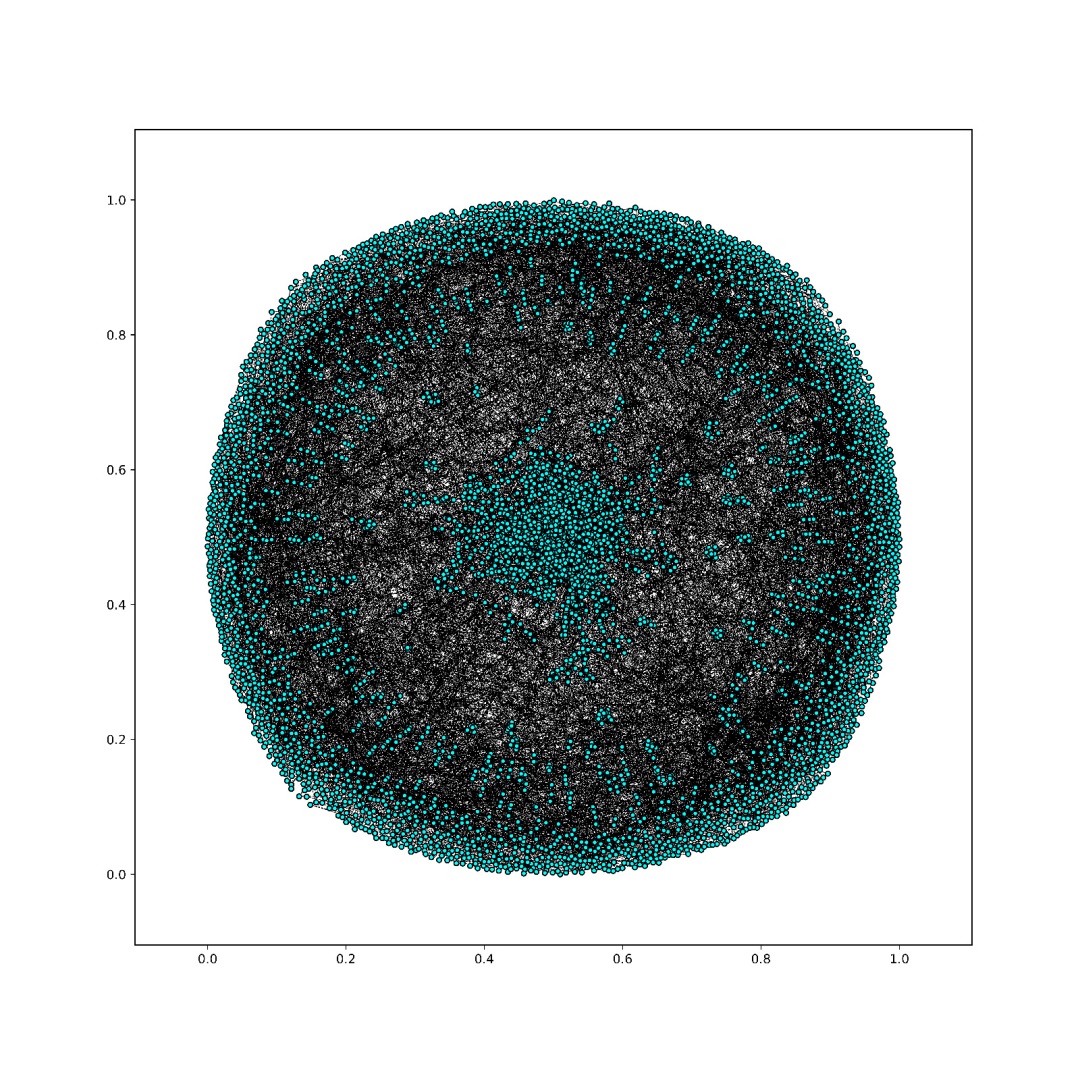
Homework 4

Part 1 – By processing Json file create a graph G

A graph G= (V,E) consist of a set of nodes V, and a set of edges E. In our case we define as nodes the authors and the edges as weighted that is compute by 1- J(p1,p2) or 1- Jaccard similarity between two sets of pubblications thats “are contained” in the authors.

In our case we considered that graph G undirected and we assume that (u,v) and (v,u) are the same edge.

The representation of the graph (reduced file json in input) is shown below:



Part 2 – Compute some statistics and visualizations

Centrality measures how central is the node in the graph. Depending on what we mean by “central”, there are versions of centrality measure.

1. Degree centrality:

is a simple centrality measure that counts how many neighbors a node has.

1. Closeness centrality:

measures the mean distance from a vertex to other vertices. Intuitively a node is

central if its distance from the other nodes is small.

1. Betweeness centrality:

measures the extent to which a vertex lies on paths between other vertices.

Results by the conference “conf/pkdd/2011-1” in input:

author\_id: 20336

degree centrality: 0.0

closeness centrality: 0.0

betweenness centrality: 0.0

author\_id: 252278

degree centrality: 0.0

closeness centrality: 0.0

betweenness centrality: 0.0

author\_id: 3476

degree centrality: 0.025210084033613446

closeness centrality: 0.025210084033613446

betweenness centrality: 0.0

author\_id: 124828

degree centrality: 0.01680672268907563

closeness centrality: 0.01680672268907563

betweenness centrality: 0.0

author\_id: 166813

degree centrality: 0.008403361344537815

closeness centrality: 0.008403361344537815

betweenness centrality: 0.0

author\_id: 255902

degree centrality: 0.01680672268907563

closeness centrality: 0.01680672268907563

betweenness centrality: 0.0001424298532972511

author\_id: 239007

degree centrality: 0.01680672268907563

closeness centrality: 0.01680672268907563

betweenness centrality: 0.0

author\_id: 71582

degree centrality: 0.025210084033613446

closeness centrality: 0.025210084033613446

betweenness centrality: 0.0

author\_id: 255394

degree centrality: 0.01680672268907563

closeness centrality: 0.01680672268907563

betweenness centrality: 0.0

author\_id: 255395

degree centrality: 0.01680672268907563

closeness centrality: 0.01680672268907563

betweenness centrality: 0.0

author\_id: 8612

degree centrality: 0.01680672268907563

closeness centrality: 0.01680672268907563

betweenness centrality: 0.0

author\_id: 20392

degree centrality: 0.008403361344537815

closeness centrality: 0.008403361344537815

betweenness centrality: 0.0

author\_id: 255405

degree centrality: 0.025210084033613446

closeness centrality: 0.025210084033613446

betweenness centrality: 0.0

author\_id: 20405

degree centrality: 0.008403361344537815

closeness centrality: 0.008403361344537815

betweenness centrality: 0.0

author\_id: 255926

degree centrality: 0.01680672268907563

closeness centrality: 0.01680672268907563

betweenness centrality: 0.0

author\_id: 20407

degree centrality: 0.008403361344537815

closeness centrality: 0.008403361344537815

betweenness centrality: 0.0

author\_id: 114625

degree centrality: 0.01680672268907563

closeness centrality: 0.01680672268907563

betweenness centrality: 0.0

author\_id: 114626

degree centrality: 0.01680672268907563

closeness centrality: 0.01680672268907563

betweenness centrality: 0.0

author\_id: 93126

degree centrality: 0.01680672268907563

closeness centrality: 0.01680672268907563

betweenness centrality: 0.0

author\_id: 43462

degree centrality: 0.01680672268907563

closeness centrality: 0.01680672268907563

betweenness centrality: 0.0

author\_id: 210891

degree centrality: 0.01680672268907563

closeness centrality: 0.01680672268907563

betweenness centrality: 0.0

author\_id: 255950

degree centrality: 0.0

closeness centrality: 0.0

betweenness centrality: 0.0

author\_id: 13781

degree centrality: 0.008403361344537815

closeness centrality: 0.008403361344537815

betweenness centrality: 0.0

author\_id: 21462

degree centrality: 0.008403361344537815

closeness centrality: 0.008403361344537815

betweenness centrality: 0.0

author\_id: 256482

degree centrality: 0.0

closeness centrality: 0.0

betweenness centrality: 0.0

author\_id: 256500

degree centrality: 0.008403361344537815

closeness centrality: 0.011204481792717085

betweenness centrality: 0.0

author\_id: 255996

degree centrality: 0.008403361344537815

closeness centrality: 0.008403361344537815

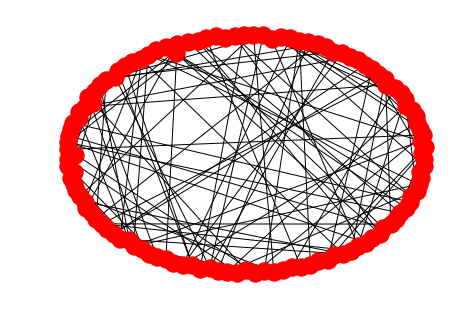
betweenness centrality: 0.0

author\_id: 255487

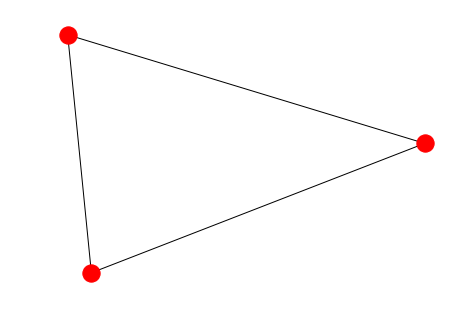
degree centrality: 0.025210084033613446

closeness centrality: 0.025210084033613446

betweenness centrality: 0.0



1. For instance we use author name = michel verleysen and d = 1 and the result is

I’M NOT SURE FOR THIS GRAPH!

Part 3 – Compute some generalized version of the Erdos number.

# To compute “Aris number” we have implemented from scratch the algorithm to compute the shortest path. So we have following the logic of the Dijkstra’s Algorithm.

For instance we compute the weight of the shortest path that connects “george brova” and Aris and the result is:

{256177: 0, 255328: 0.0, 256176: 0.0} that it’s composed by author\_id: weight (in this case weight is zero for everyone because there is only one.