SOCIAL NETWORK ANALYSIS

SACHIN RAMESH
SYRACUSE UNIVERSITY, MARCH 2017

Contents

1. P0 – crawling the website 3
   1.1. Network Selection: 3
   1.2. Crawling: 3
2. P1 – Graph Essentials 4
   2.1. Degree distribution and Power Law Exponent. 4
   2.2. Number of bridges: 4
   2.3. Number of 3 cycles: 5
   2.4. Graph Diameter 5
   2.5. Minimum Spanning Tree: 5
3. P2 – Network Measures 5
   3.1. Average and Local Clustering Coefficient: 5
   3.2. Centrality Measures 5
   3.3. Jaccard Similarity 7
4. P3 – Network Modelling 7
   4.1. Random graph: 7
   4.2. Small World Model 8
   4.3. Preferential Attachment 8
   4.4. Values Comparison: 9

1. P0 – crawling the website

1.1. Network Selection:

I have selected www.gaiaonline.com to crawl and obtain the datasets.

Gaia Online is an English-language, anime-themed social networking and forums-based website. It was founded as go-gaia on February 18, 2003,and the name was changed to GaiaOnline.com in 2004 by its owner, Gaia Interactive. Gaia originally began as an anime linklist and eventually developed a small community but, following a statement by founder Derek Liu (username "Lanzer"), the website moved towards social gaming, and eventually became forum-based. In 2007, over a million posts were made daily, and 7 million unique users visited each month (with over 26 million total registered users). Gaia also won the 2007 Webware 100 award in the Community category and was included in Time Magazine's list of 50 best websites in 2008.In January 2011, the company won the Mashable Best User experience Award for 2010.
The website has around 25,000,000 registered users.

1.2. Crawling:

There are three files main, crawler and operations. Main program initializes the variables, creates the threads for multi-processing and calls the crawler class. Operations is programmed to find and parse the obtained urls and to perform file operations. Crawler crawls the website starting with the given node, it gets all the linked url's and places only user profiles in queue file, iteratively the links in queue are crawled and files are updated accordingly. All the connected url's are saved in the form of anonymized edges that are recorded in "edges.txt" . "name_mapping.txt" maps the node number and the user profiles." url_matching.txt" maps the non-anonymized edge list (user profiles).

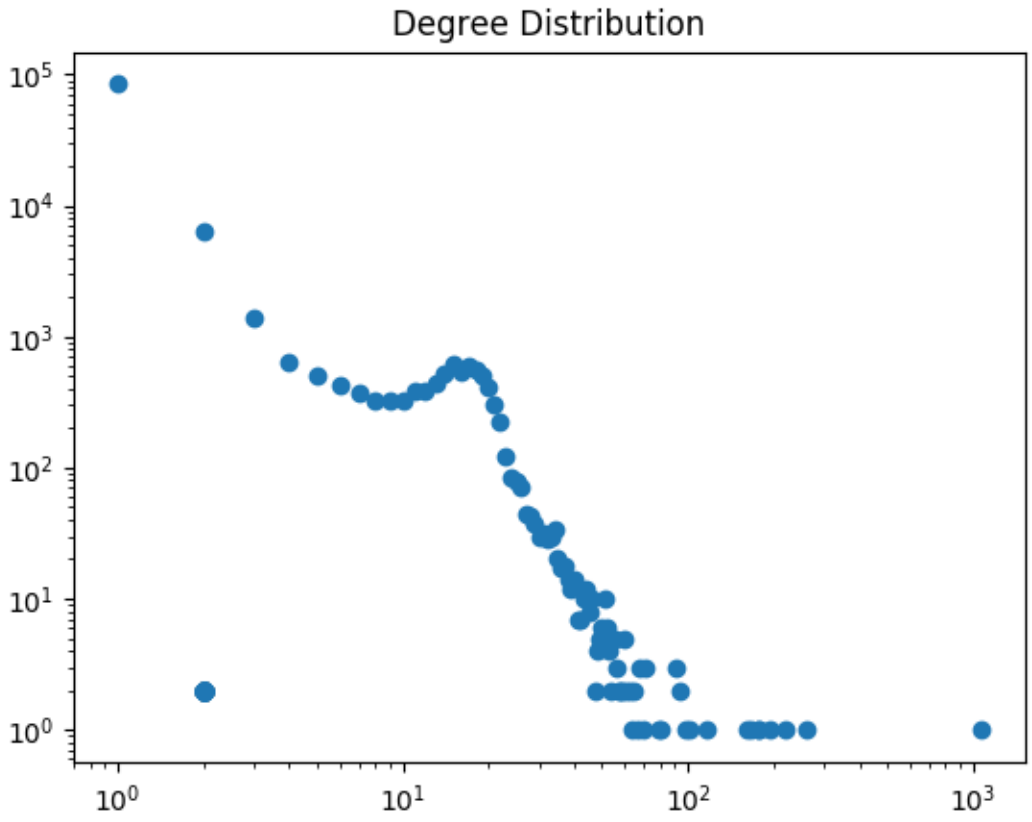
Source code: crawler.py, operations.py, main.py
Location: CIS700 - SMM -Project1\P0\Crawler_Source_code

Deliverables: edges.txt, url_match.txt, name_mapping.txt, crawled.txt
Location: CIS700 - SMM -Project1\P0\DataSetsDeliverables\Gaiaonline

2. P1 – Graph Essentials

2.1. Degree distribution and Power Law Exponent.

Degree distribution plot is shown below:



Degree distribution of the Graph follows the power law distribution.

Average path length = 7.29

Source code : degree_distribution.py
Location: CIS700 - SMM -Project1\P1\degree distribution

2.2. Number of bridges:

The total Number of computed bridges for the graph were
Number of bridges in a graph = 90048
Source code: bridges.py
The text file bridges.txt contains all the bridges(edge list)

Location: CIS700 - SMM -Project1\P1\Bridges

2.3. Number of 3 cycles:

The total number of 3 cycles in the network :
number of triangles 23742
source code: no-of_triangles.py
Location: CIS700 - SMM -Project1\P1\Number of 3 cycles

2.4. Graph Diameter

The value of diameter of the graph is 10.
Source code: diameter.py
Location: CIS700 - SMM -Project1\P1\Diameter

2.5. Minimum Spanning Tree:

With $W_{ij} = |d_i| - |d_j|$
Total Weight of minimum spanning tree is = 1903867
Diameter of the minimum spanning tree is = 8
The computed MST edges and their corresponding weights are reported in "MST_edgelist.txt"
Source code: mst.py
Location: CIS700 - SMM -Project1\P1\MST

3. P2 – Network Measures

3.1. Average and Local Clustering Coefficient:

Average local Clustering co- efficient value = 0.022371169759588902
Global clustering co-efficient = 0.017995133992071893
Local clustering co-efficient was computed using networkx function.
Global cluster co-efficient was calculated by computing the number of triangles and triples
number of triangles 23742
Number of Triples 3958070.0
Source code: Clustering Co-efficient.py
Location: CIS700 - SMM -Project1\P2\Clustering Co-efficient

3.2. Centrality Measures

1.**Eigen vector centrality:**

For node	1	Eigenvector centrality Value is	0.6959798282469405
For node	128	Eigenvector centrality Value is	0.028079406774238234
For node	114	Eigenvector centrality Value is	0.028208094912592352

2.**Degree Centrality:**

Top 3 nodes with maximum values
[0.010570491286486912, 0.00255149789673822, 0.0021574441675122403]

For node	1	Degree centrality Value is	0.010570491286486912
For node	823	Degree centrality Value is	0.00255149789673822
For node	18637	Degree centrality Value is	0.0021574441675122403

3. **Page Rank:**

Top 3 nodes with maximum values

For node	1	Page Rank Value is	0.003443038693734901
For node	823	Page Rank Value is	0.0010248932532207172
For node	18637	Page Rank Value is	0.000858986533193949

Nodes Match:
1----->http://www.gaiaonline.com/profiles/ie-batman/9487660/
823----->http://www.gaiaonline.com/profiles/exiled_royalty/12409880/
18637----->http://www.gaiaonline.com/profiles/sajeku/28773549/
128----->http://www.gaiaonline.com/profiles/morpheus297/25589153/
114----->http://www.gaiaonline.com/profiles/fallingangel54/4547443/

Source code: degree_centality.py, page_rank.py and eigenvector_centrality
Location: CIS700 - SMM -Project1\P2\Centralities

All the 3 centrality value is highest for first node as we have started crawling with that node and

In the website, there is an option to place the (avatars) profile links of closely connected people/friends on the home page.

Among the nodes 823,18637,128,114 above mentioned avatars on the homepage is more for 823 and 18637.

So, degree centrality and pagerank works the most in this case.

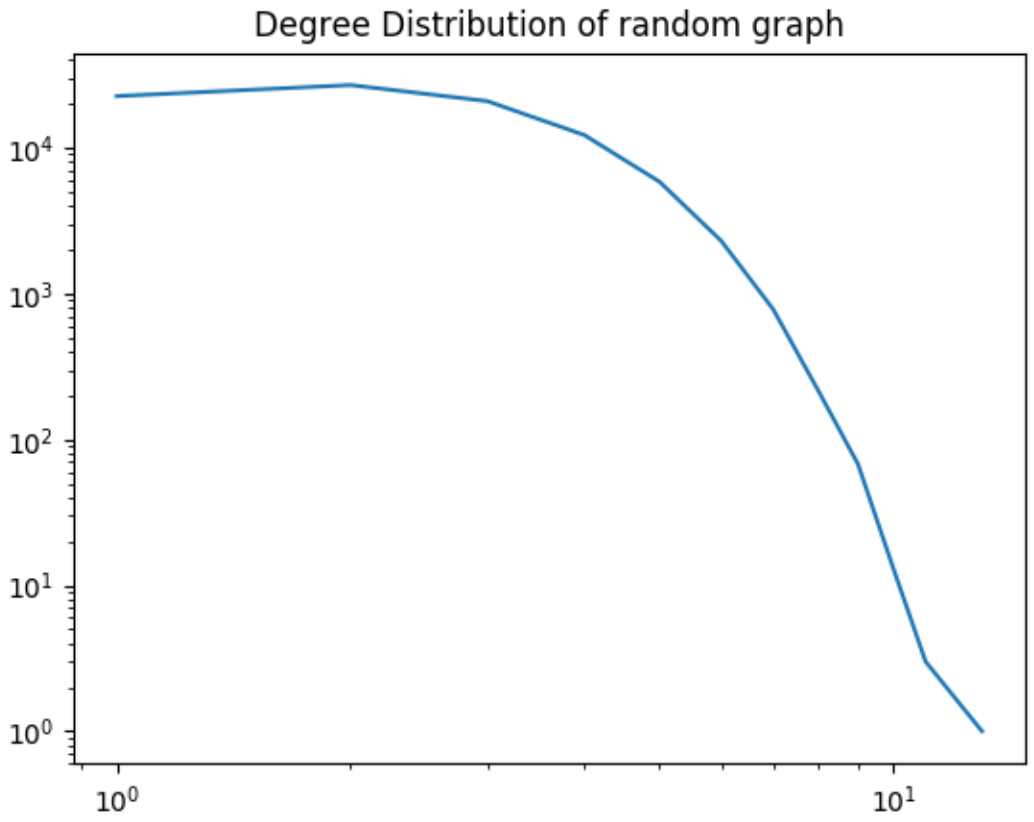
3.3. Jaccard Similarity

Nodes are most similar when the jaccard co-efficient is 1.
The list of most similar pair of nodes is presented in “Pair_of_Most_SimilarNodes.txt”
Source code: jaccard.py
Location: CIS700 - SMM -Project1\P2\Jaccard Similarity

4. P3 – Network Modelling

4.1. Random graph: G(n,p)

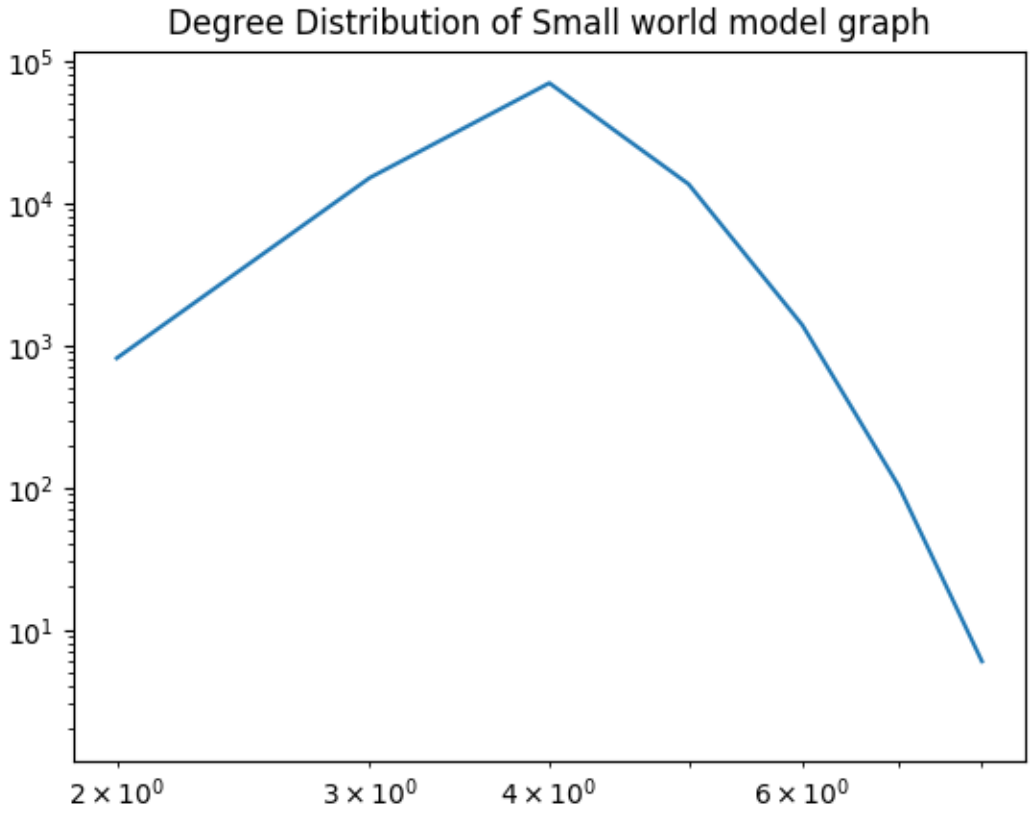
Average path length: 11.9488566990224
Clustering co-efficient : 0.0000228290646606079



Source code: random_graph.py
Location: CIS700 - SMM -Project1\P3\random graph

4.2. Small World Model (watts strogatz model)

Average path length: 16.12175
Clustering co-efficient : 0.5

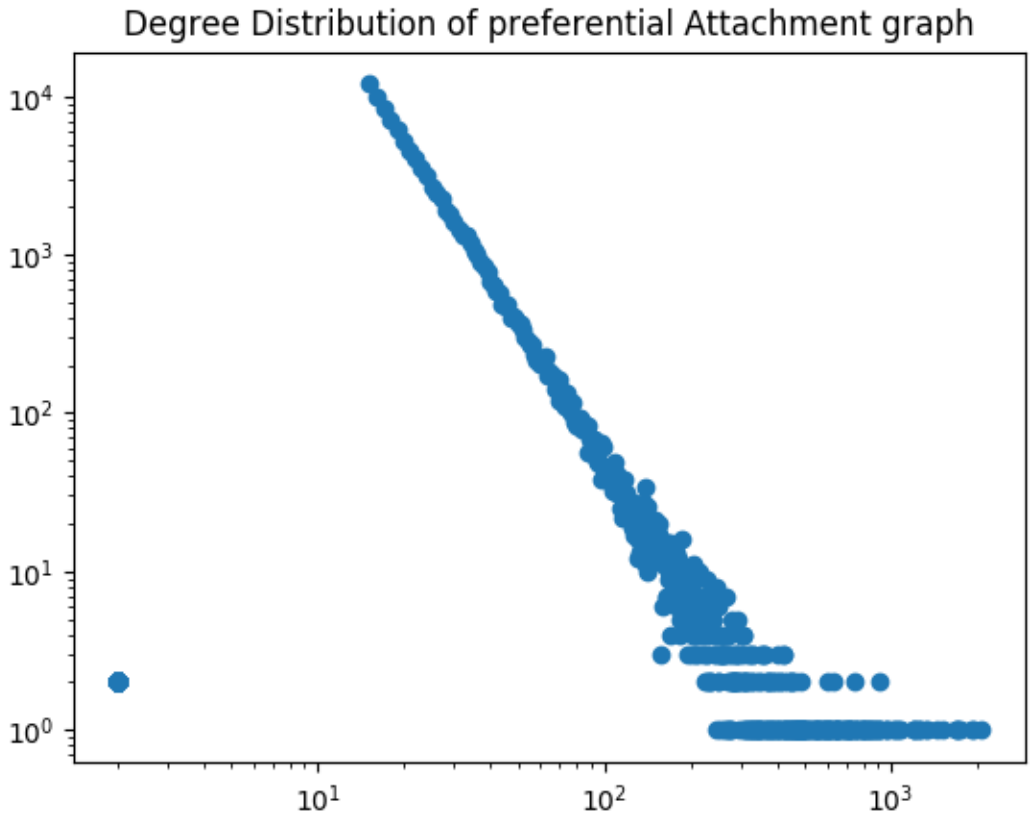


Source code:
Location: CIS700 - SMM -Project1\P3\Small world model

4.3. Preferential Attachment (barabasi albert model)

Average path length: 4.7153337
Clustering co-efficient : 0.0232788764415871

Source code: Preferential attachemnt.py
Location: CIS700 - SMM -Project1\P3\Preferential Attachment



4.4. Values Comparison:

Models	Average path lengths	Clustering Coefficient	Degree distribution
Actual graph	7.296640331	0.02237117	follows power law
random graph	11.9488567	2.28E-05	does not follow power law
small world model	16.12175	0.5	does not follow power law
Preferential attachment	4.7153337	0.023278876	follows power law

Observations:

- Random graphs have high average path length and very low clustering co -efficient compared to actual graph
- Small world model has very high average path length and very high clustering co-efficient compared to actual graph.
- Preferential attachment model has slightly lower average path length and clustering co-efficient is like that of actual graph.