IR Assignment 3
Report

Devika Mittal	2019467
Prashasti Agarwal	2019075
Pritish Wadhwa	2019440

Question 1 - Link Analysis

I. Represent the network in terms of its 'adjacency matrix' as well as 'edge list'.

- Adjacency Matrix
  - First a matrix of size N*N is created (N: Number of Nodes).
  - Then the corresponding edges from Row Node → Column Node is marked as "1"
  - Eg. if we have an edge A → B then **`adjacencyMatrix[A][B] = 1, 0 otherwise`**

- Adjacency List
  - If the source node is already present in the adjacency list
  - Append the destination node to the list of nodes adjacent to the source node
  - Otherwise, add the source node to the adjacency list, with the destination node adjacent to it.
  - Simultaneously add destination node to the adjacency list as well

II. Briefly describe the dataset chosen and report the following

The dataset chosen is the gnutella peer to peer network, August 2008. This represents the snapshot of file sharing from one peer to another in the gnutella protocol. In the Gnutella protocol, the hosts are represented by the nodes and the connections (between hosts) are represented as the edges between the nodes of the graph. This is a directed network.

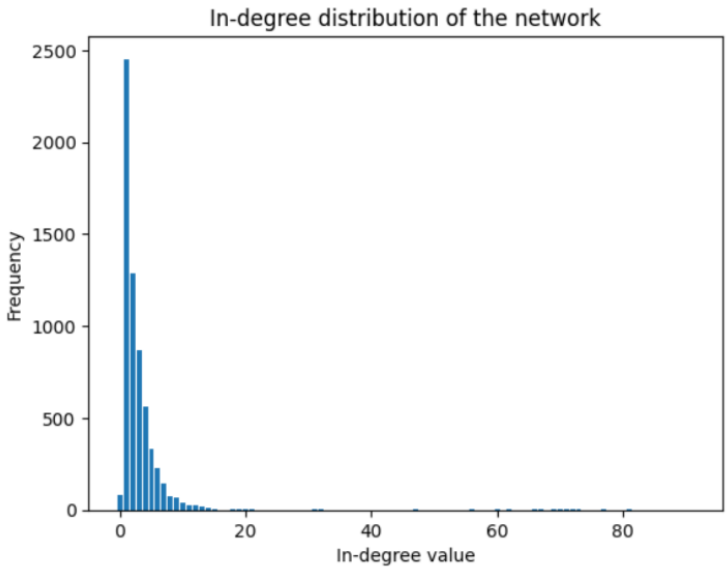
Metric	Result	Formula Used
Number of Nodes	6301	`length(adjacencyList)` Where `adjacencyList` is the Adjacency List created above.

Number of Edges	20777	$\sum_{i}^{all\ Nodes} length(adjacencyList[i])$ The summation of lengths of adjacency lists of each nodes.
Avg In-degree	3.339	mean(inDegree[i]) Mean of inDegrees of all the nodes.
Avg. Out-Degree	8.4288	mean(outDegree[i]) Mean of outDegrees of all the nodes.
Node with Max In-degree	Node 266 with inDegree = 91	Iterate over all the inDegree to find the Node with maximum value.
Node with Max out-degree	Node 5831 with outDegree = 48	Iterate over all the outDegree to find the Node with maximum value.
The density of the network	0.000523	$Density = \frac{Number\ of\ Edges}{Number\ of\ Possible\ Edges}$ Where, Number of Possible Edges = $N * (N - 1)$ for a directed graph. N being the number of nodes.

- Calculation of In-Degreee and Out-Degree
  - We iterate over all the tuples of the graph (a,b) where a → b is an edge.
  - We increment `outDegree[a]` by 1
  - We increment `inDegree[b]` by 1

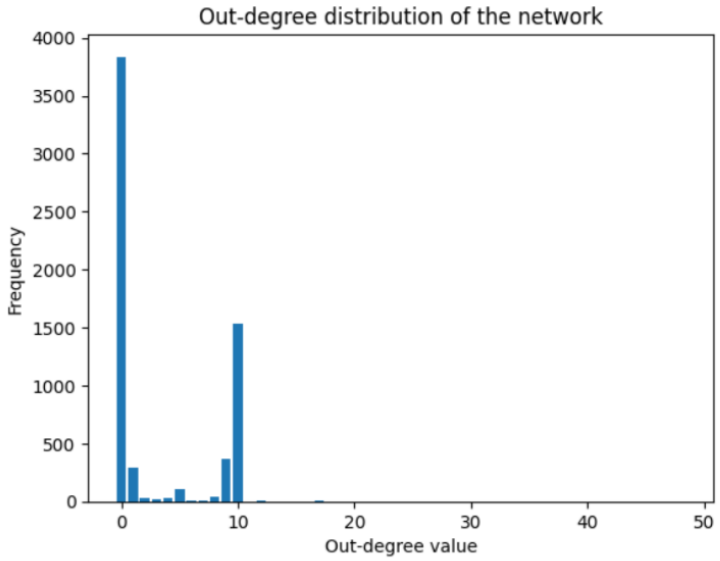
III. Plot degree distribution of the network (in case of a directed graph, plot in-degree and out-degree separately).

- **In-degree distribution**
  - We have the in-degrees of all the nodes from the previous question.
  - We compute the number of nodes having each in-degree value. In other words, the frequency of each in-degree.
  - Then, we plot frequency vs in-degree.



- We can see that the highest frequency is for the small in-degree values of 1-5. That means most nodes only have about these many edges going into it.

- **Out-degree distribution**
  - Same steps as above are followed for the out-degree list.
  - Result -



  - The maximum out-degree frequency is seen for the values 0 and 10.

IV. Calculate the local clustering coefficient of each node and plot the clustering-coefficient distribution (lcc vs frequency of lcc) of the network.

- First, we create an undirected adjacency list where an edge from node a to node b is recorded in the edge lists of both a and b.
- Next, we compute the local clustering coefficient (LCC) for each node
  - All the nodes connected to the current node by an edge are the neighbours of the node (Note - the direction of the edge can be anything - 'to' or 'from' the current node. Hence, the undirected adjacency list is used)

    Given: a graph $G = (V, E)$ where V is the set of vertices and E is set of edges. E_{ij} connects vertex v_i with vertex v_j.

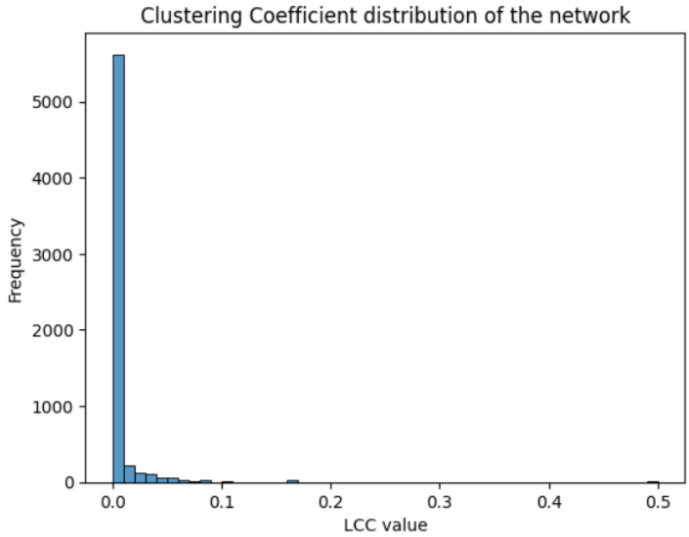
    $N_i = \{v_j : e_{ij} \in E \vee e_{ji} \in E\}$ N_i is the neighbourhood of vertex v_i.

    $k_i = |N_i|$ is the number of vertices in the neighbourhood of vertex v_i
  - The local clustering coefficient for directed graphs -

    $C_i = \frac{|\{e_{jk} : v_j, v_k \in N_i, e_{jk} \in E\}|}{k_i(k_i - 1)}$

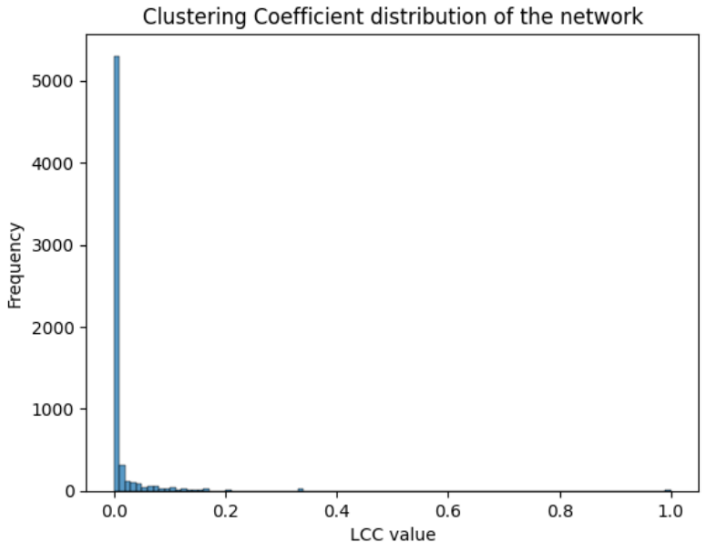
    The LCC of a node is the proportion of number of links between vertices in its neighbourhood divided by total number of links that could possibly exist between them.

    (Reference: https://en.wikipedia.org/wiki/Clustering_coefficient#)
  - If $k_i < 2$, then the LCC is set to 0.
- Given the LCC value for each node, we calculate the frequency of LCC values and plot the distribution -



- Using a slightly different approach that considers the graph to be undirected and then computes the LCC, we get the following distribution of LCC - (Reference - https://neo4j.com/docs/graph-data-science/current/algorithms/local-clustering-coefficient/)

$$C_i = \frac{|\{e_{jk} : v_j, v_k \in N_i, e_{jk} \in E\}|}{(k_i(k_i - 1)) / 2}$$



Question 2 - PageRank, Hubs and Authority

To calculate the Page Rank, Authority and Hub score we used the `networkx DiGraph()`

I. PageRank score for each node

- PageRank was calculated using `nx.pagerank()`
- Some of the nodes with their page rank are (highest to lower):

```
Node  Page Rank
367   0.002378560316719017
249   0.002177444615709691
145   0.0020465276122979673
264   0.00199037802243687
266   0.0019575525218090907
123   0.00186004786420860 8
127   0.0018568582711618506
122   0.001847615807792213
```

II. Authority and Hub score for each node

- Hub Score and Authority is calculated using `nx.hits()`
- Top 10 nodes with their Authority are:

Node	Authority
367	0.021468843025944973
249	0.020290000606545195
123	0.02024512562404484
127	0.019874039870419406
266	0.019718056113905494
264	0.019699500990225955
145	0.019051799048673518
251	0.018910168734512917
427	0.018756678505735917
3	0.01858823919434392

- Top 10 nodes with their Hub Score are:

Node	Hub Score
3459	0.0030323959687676197
366	0.003004853728692918
36	0.002992944003649025
2374	0.0029659158718905528
3693	0.002954220175634604
2020	0.0029082331836166133
1884	0.002903019395056574
1739	0.0028982018008304254
3551	0.002896544426876794
5724	0.0028870934554914938

III. Compare the results obtained from both the algorithms in parts 1 and 2 based on the node scores.

Page Rank

- We initialize rank of all nodes = 1
- Then for each node i,

  $rank(i) = (1 - \beta) + \beta * \sum_{\{\forall j : j \to i \in links\}} rank(j)/\, outDegree(j)$, the sum of page rank of all the nodes that link to it. And β is the damping factor (0.85 in our case).
- Top 10 Nodes with highest page rank are:

Nodes	PageRank
367	0.002378560316719017
249	0.002177444615709691

```
145   0.0020465276122979673
264   0.00199037802243687
266   0.0019575525218090907
123   0.001860047864208608
127   0.0018568582711618506
122   0.001847615807792213
1317  0.0018356434286446008
5     0.0018239540797636209
```

- **Page Rank is highest for the node 367 with inDegree = 86 and outDegree = 8**

Authority

- We initialize authority of all nodes = 1
- Then for each node i, $auth(i) = \sum_{\{\forall\, j\,:\, j \to i \in links\}} hub(j)$, the sum of hub scores of all the nodes that link to it.
- Top 10 Nodes with highest authority scores are:

```
Nodes      Authority
367   0.02146884302594497
249   0.0202900006065452 02
123   0.020245125624044836
127   0.019874039870419406
266   0.019718056113905473
264   0.019699500990225962
145   0.019051799048673514
251   0.018910168734512917
427   0.018756678505735914
3     0.01858823919434392
```

- **Authority Score is highest for the node 367 with inDegree = 86 and outDegree = 8**

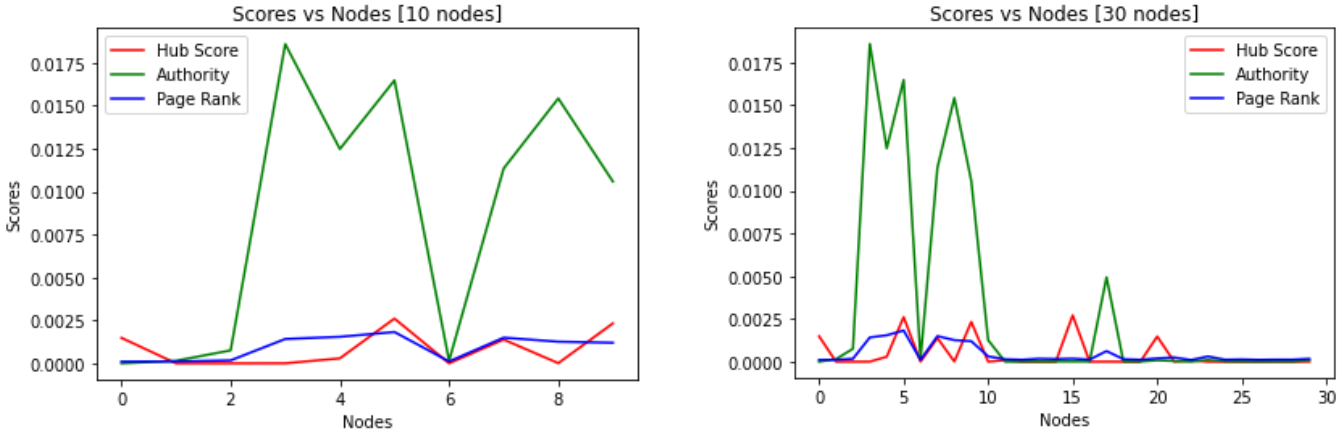
Hub Score

- We initialize hub score of all nodes = 1
- Then for each node i, $hub(i) = \sum_{\{\forall\, j\,:\, i \to j \in links\}} auth(j)$, the sum of authority score of all the nodes it links to.
- Top 10 Nodes with highest Hub Score are:

```
Nodes      Hub Score
3459       0.0030323959687676197
366        0.0030048537286929185
36         0.002992944003649026
2374       0.0029659158718905528
```

```
3693        0.0029542201756346037
2020        0.002908233183616612
1884        0.002903019395056574
1739        0.002898201800830426
3551        0.002896544426876795
5724        0.002887093455491494
```

- **Hub Score is highest for the node 3459 with inDegree = 3 and outDegree = 10**

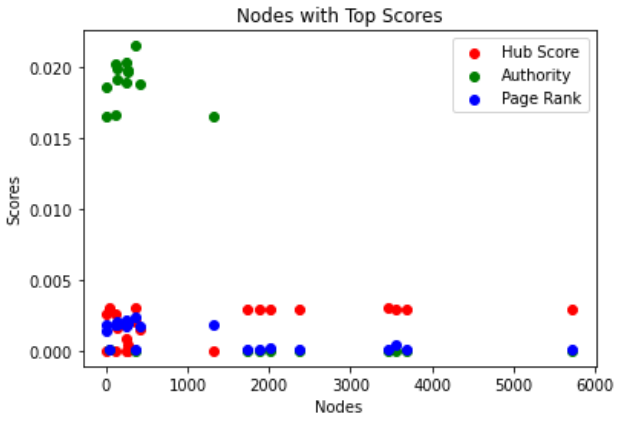


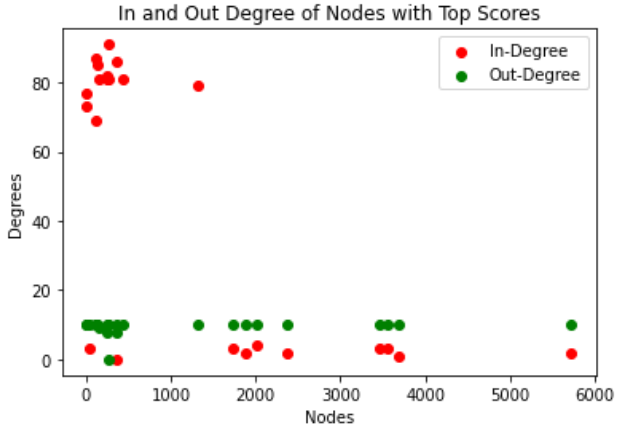
Hub Score, Page Rank and Authority of top 10 and top 30 nodes.

Nodes with one of the highest Hub Scores, Page Ranks and Authority along with their In-degree and Out-Degree

	Node	Hub Score	Authority	Page Rank	In Degree	Out Degree
0	3459	0.003032	2.358939e-05	0.000142	3	10
1	3	0.000003	1.858824e-02	0.001418	77	10
2	5	0.002607	1.647611e-02	0.001824	73	10
3	2374	0.002966	2.682509e-05	0.000131	2	10
4	264	0.000432	1.969950e-02	0.001990	81	10
5	266	0.000000	1.971806e-02	0.001958	91	0
6	1739	0.002898	8.805704e-07	0.000131	3	10
7	145	0.001767	1.905180e-02	0.002047	81	9
8	5724	0.002887	3.232332e-07	0.000123	2	10
9	1884	0.002903	9.497554e-07	0.000119	2	10
10	3551	0.002897	4.239786e-07	0.000385	3	10
11	36	0.002993	4.987197e-05	0.000122	3	10
12	1317	0.000003	1.646102e-02	0.001836	79	10
13	2020	0.002908	1.655121e-06	0.000158	4	10
14	427	0.001468	1.875668e-02	0.001765	81	10
15	3693	0.002954	3.177810e-07	0.000118	1	10

Scatter plot for the Same





Discussion

- We can see that Hub Score follows the same trend as that of out-degrees
- Authority and page rank follows the same trend as that of in-degree.
- Page rank and HITS scores differ but have a lot of nodes common in the top 10 scores,
- Page rank compares the scores based on the incoming links. Higher the incoming links, higher is the Pagerank Score.

- HITS Algorithm computes the authority on the basis of incoming links and Hub Score on the basis of Outgoing Links
- Higher the incoming links, greater is the Authority and higher the outgoing links, greater is the Hub Score.
- According to the HITS algorithm, we can see that for the node with highest authority [367] the incoming edges are from: [366, 2374, 3459, 3551, 3693] , all nodes in the top 10 hub scores
- For the node with the highest Hub Score [3459] , outgoing edges are to: [**3**, 124, **127**, 129, 177, **251**, **266**, **367**, 368, 422] all the nodes with high authority. (**Bold** ones in top 10)
- Part I uses the Page Rank algorithm while Part II uses the HITS algorithm. Although both algorithms offer the same spirit, finding the ranking of a page. HITS operates only on a subgraph.

Sources:

1. https://towardsdatascience.com/hits-algorithm-link-analysis-explanation-and-python-implementation-61f0762fd7cf
2. https://en.wikipedia.org/wiki/HITS_algorithm