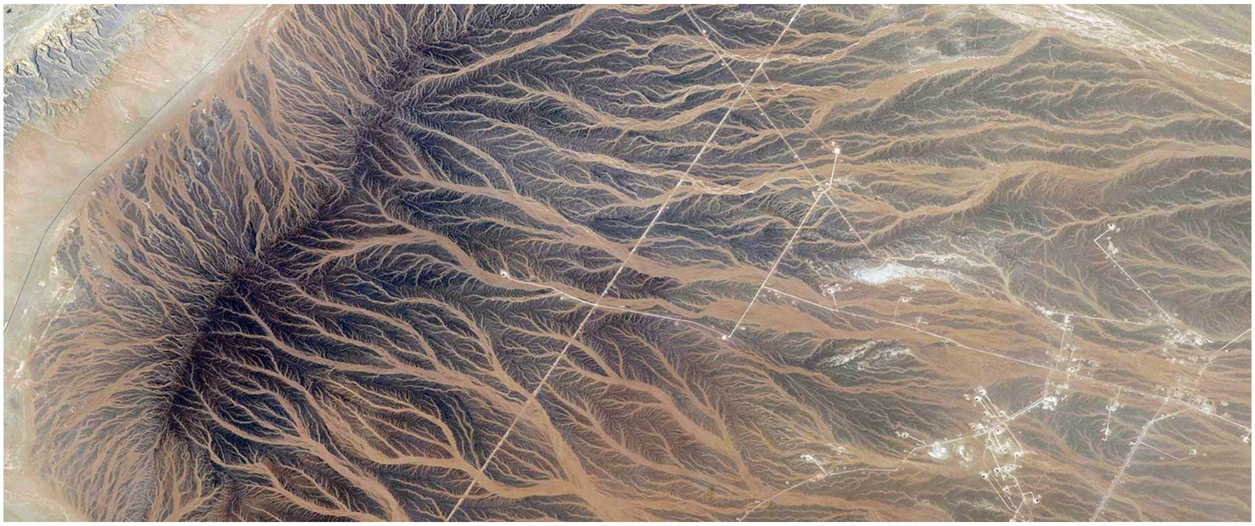


Social Networks Analysis - HW 1

Part 1

1. **Choose a directed, one-mode social network dataset from any source you like (such as the online databases mentioned in class), whose size is at least |V1| = 10, 000. Provide a brief explanation of the network and what it models**

   After some investigations I chose a paper citation network from Stanford University's network data source.

   Arxiv HEP-TH (high energy physics theory) citation graph is from the e-print arXiv and covers all the citations within a dataset of 27,770 papers with 352,807 edges. If a paper *i* cites paper *j*, the graph contains a directed edge from *i* to *j*. If a paper cites, or is cited by, a paper outside the dataset, the graph does not contain any information about this.

   The data covers papers in the period from January 1993 to April 2003 (124 months). It begins within a few months of the inception of the arXiv, and thus represents essentially the complete history of its HEP-TH section.

   The data was originally released as a part of 2003 KDD Cup.

   http://snap.stanford.edu/data/cit-HepTh.html

2. **The original input file can be in any format that allows you to read the data into Python (or some other language you are familiar with). Read the data and**

create a directed graph object G1 = (V1,A1). If the input file type was not .gexf, then write your graph to a .gexf file and re-read from the new file. (You should attach the .gexf file while submitting the assignment)

You can find the relevant piece of code and the file 'cit-HepTh_graph.gexf' attached in the HW folder.

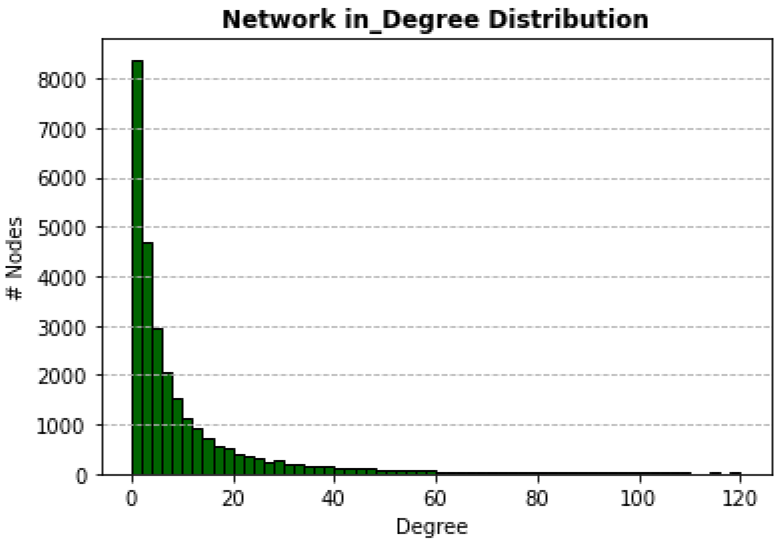
3. **Describe the structure of the network using some basic measures such as the density, average shortest path length, diameter, connectedness. Explain the meaning of their values. Is this a simple directed network? Are there any loops? If not simple, remove multiple arcs and self loops.**

```
Number of nodes: 27770
Number of edges: 352768
is strongly connected graph: False
is weakly connected graph: False
Number of self-loops:  0
Network Density: 0.00046
Average clustering coefficient: 0.157
```

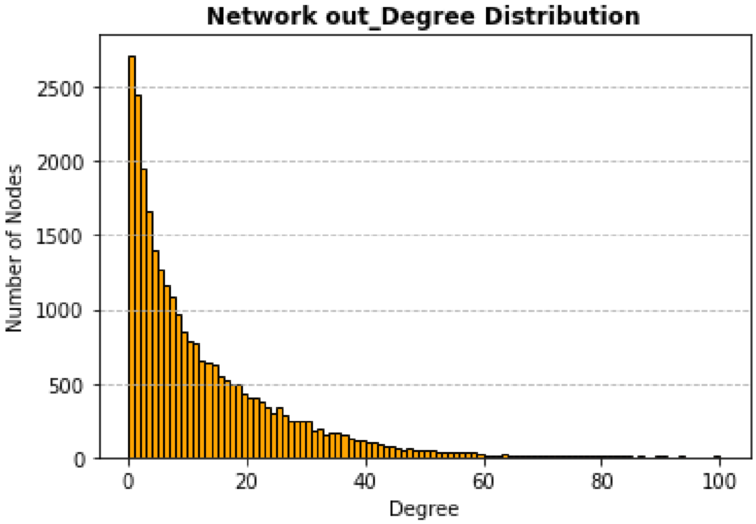
4. **Plot the degree distribution of your network. Does it show signs of a certain network structure, such as a random network or a scale free network? What are the other measures that are important for classifying a network as random, small-world, or scale-free?**

   Degree distribution for in_degree and out_degree and total cases are plotted. As it can easily be seen they follow power law regime. At the end power law exponent has been calculated.

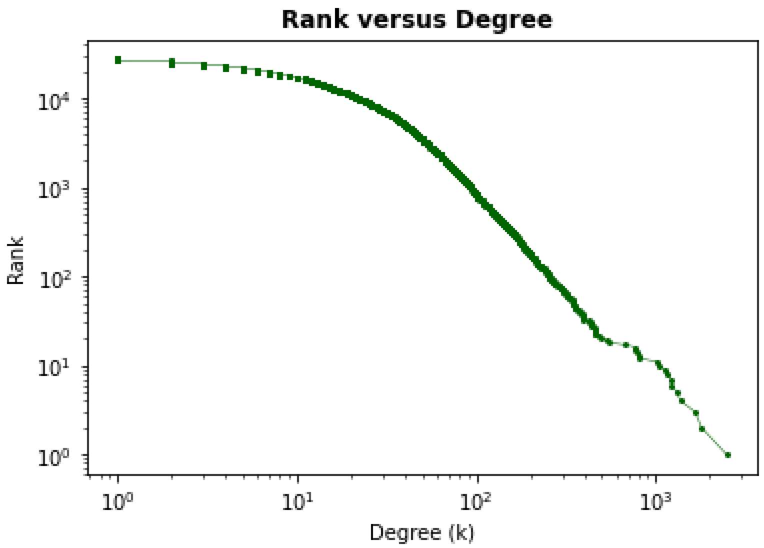
   Power law behavior in degree distribution shows scale free-ness of paper citation network.

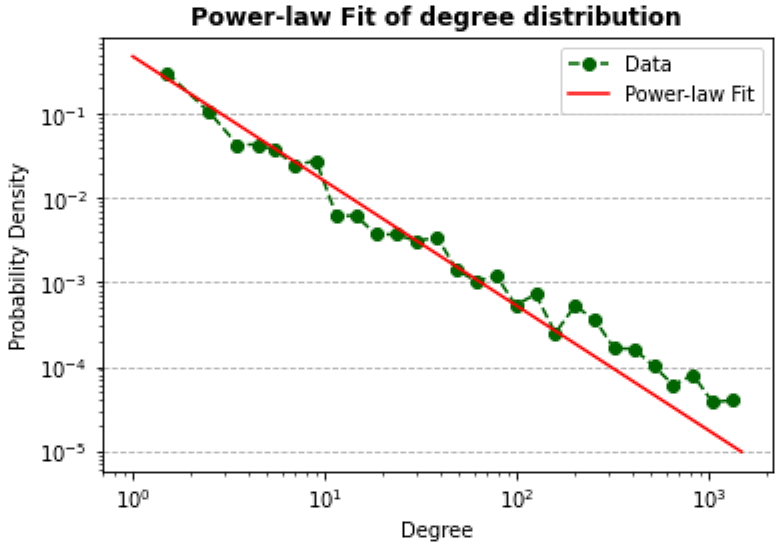


Max in degree: 2414

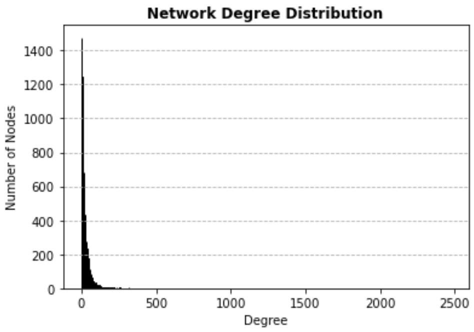


Max out degree : 562





power law exponent is `Alpha =  1.4807117420687688`



Part 2

Part 2

1. **Determine some rule to obtain a subset V2 ⊂ V1 such that 1000 ≤ |V2| ≤ 2000. The rule could be based on some attribute of the nodes. Some examples are: all the nodes whose “label” is “student” or all the nodes with total degree larger than 30. (If it doesn’t work, explain why and randomly select the set V2). Then, obtain an induced subgraph of G1 with respect to V2. Call the new graph G2. What is the new number of nodes and edges in G2, how dense it is?**

   There's no specific threshold that universally defines a high-impact paper, but in many fields, a paper cited more than 100 times can be considered highly impactful. In fast-moving or highly specialized fields, even lower citation counts might signify high impact. Here I choose 50 as this threshold.

   ```
   Number of nodes: 1338
   Number of edges: 21845
   is strongly connected graph False
   is weakly connected graph False
   Number of self-loops:  0
   Network Density: 0.012211373878784017
   Average clustering coefficient: 0.18445224374154545
   ```

2. **Pick a random node v ∈ V2. Find three nodes which are in the same strongly connected component as v on G2, by using a graph search method such as BFS or DFS**

   You can see the methid in the jupter notebook.

   ```
   Randomly selected node: 9907026
   Three nodes from the same strongly connected component:
                                ['9809145', '111005', '6215']
   ```

3. **Find weakly connected components of G2 by implementing an algorithm (not using a built-in function).**

   You can see my algorithm:

I used the result of the next question here, where I used the union approach to construct a undirected network and transform it to an adjacency matrix. Now my function get adjm as input instead of a nx graph.

```
def neighbours(adjm, i): #finds excited neighbors of root i
    #i = root
    neighbours=[i]
    for j in range(len(adjm)):
        if adjm[i,j]==1:
            neighbours.append(j)
    return neighbours
#finds clusters around a root using recursive algorithm
def BFS(adjm, root, cluster):

    if cluster is None : #makes a list
        cluster = []

    for neighbour in neighbours(adjm, root):
        if (neighbour not in cluster)==1:

            cluster.append(neighbour)
            BFS(adjm, neighbour, cluster)

    return (len(cluster),cluster)
def components(adjm):
    lis = [] #list for nodes that are already member of a clu
    cluster_size = [] # list for cluster sizes
    for i in range(len(adjm)):
        if i not in lis: #avoids repeating in cluster finding
            p = BFS(adjm, i, cluster = None)#finds cluster ar
            lis = lis+p[1] #adds new cluster nodes to lis
            cluster_size.append(p[0]) #adds new cluster's len

    cluster_size.sort(reverse = True)
    #return(np.array(lis))
    return(cluster_size[0],cluster_size[1],len(lis))
```

4. **Transform your network into an undirected one using one of the approaches discussed in the class (union or intersection), taking what your network models into account. Explain the reasons of your choice. Call the new network G3 = (V3, E3). Make sure V3 = V2.**

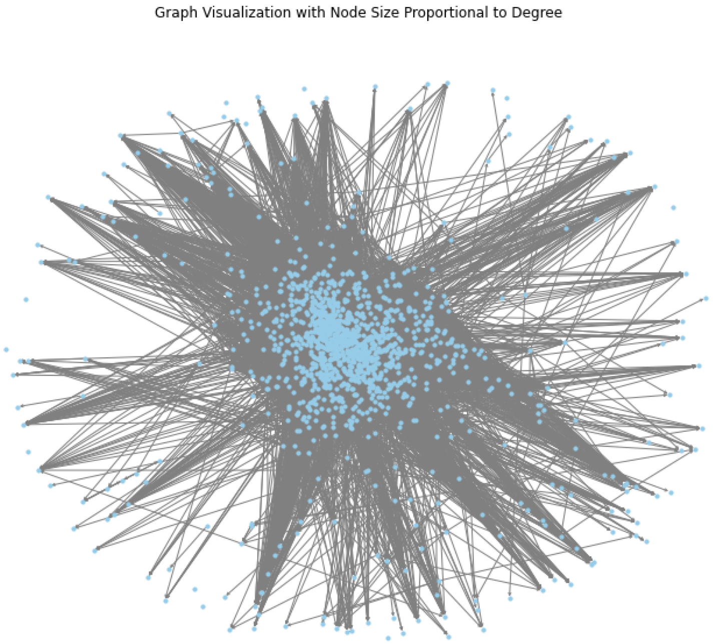
   We have two ways:

   1. **Union Approach**: In this method, an edge between two nodes in the undirected graph is created if there is at least one directed edge (in either direction) between these nodes in the original directed graph. This approach retains all connections but loses the directionality information.
   2. **Intersection Approach**: This method creates an edge between two nodes in the undirected graph only if there is a bidirectional connection (i.e., edges in both directions) between these nodes in the original directed graph. This approach is more restrictive and results in a graph that only retains reciprocated relationships.

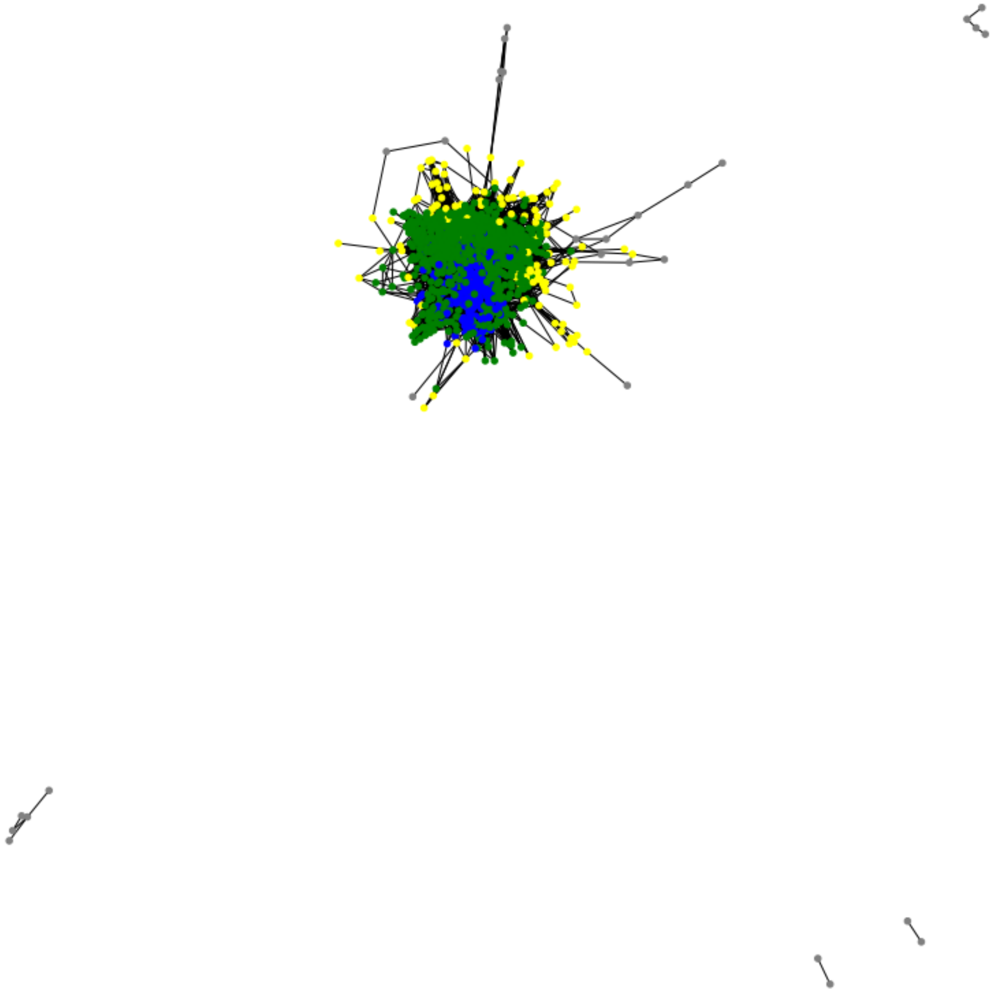
   For our network union approach is more meaningful since in the context of doing research double sided citing doesn't give us a specific information and union approach most of the necessary information.

5. **Visualize G3 such that the node size is a function of degree.**

   I wrote the functio, it didn't appear as I wanted :



1. **Determine the node with the largest betweenness centrality score on G3 and determine the 1-step, 2-step, and 3-step connections of that node. Visualize G3 such that each of these node subsets has a different color (also different from the rest of V3).**



Graph Visualization with Different Colors for Node Subsets