Analysis of Bipartite Networks

Aditi Das (202001259),[*] Mohammad Tejabwala (202001406),[†] and Bhavajna Kallakuri (202003046)[‡]
Dhirubhai Ambani Institute of Information & Communication Technology,
Gandhinagar, Gujarat 382007, India
SC435, Introduction to Complex Networks

In this paper, we analyse the bipartite networks consisting nodes (such as book, coding question) having several attributes (such as genres, topic tags). We analyse several properties of the nodes such as degree distribution, degree centrality, projection, community detection, and modularity.

I. Introduction

We have worked with bipartite graphs in this paper. A bipartite graph is a graph whose vertices can be divided into two disjoint and independent sets U and V, that is, every edge connects a vertex in U to a vertex in V and there are no edges among nodes in the same set. A bipartite graph is shown here:,

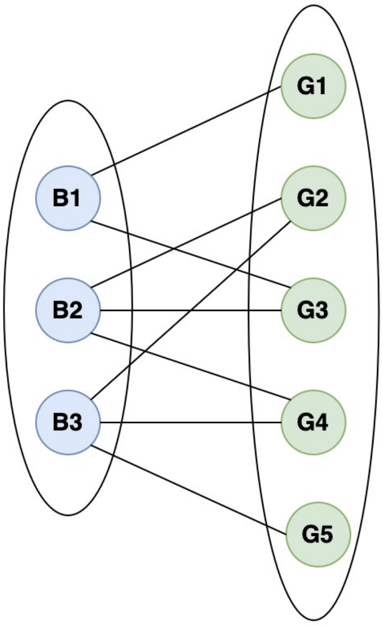

FIG. 1: A Bipartite graph consisting of 2 disjoint datasets, B and G

II. About the Datasets

We used the following datasets for our analysis:

1. **Books and Genres**: This dataset comprises information on books and genres, where a book may be associated with multiple genres. Books and Genres represent the nodes in the graph. Here, an edge exists between a book node and a genre node if a book belongs to the genre.This implies that there cannot be any edge between 2 books or 2 genres, resulting into 2 disjoint sets and a bipartite graph. The dataset encompasses a total of 90,708 nodes and 668,867 edges, with 1,178 nodes corresponding to genres and 89,530 nodes corresponding to books. Another thing to note here is that a genre can be seen in at most 17 books, and on an average, it can be oserved in 7.47 books. The dataset can be found here

2. **Codeforces problems and their topics**: This dataset comprises of contest problems along with their associated topics. A bipartite graph is constructed by representing problems and topics as nodes, with edges connecting a problem node to the corresponding topic nodes. In this configuration, the graph consists of 8,344 nodes and 28,359 edges, with 8,279 nodes corresponding to problems and the remaining 65 nodes representing topics. On average, each problem is associated with 3.42 topics, while individual problems can be linked to a maximum of 12 topics. The dataset can be found here.

3. **Leetcode Problems and related topics** This dataset consists of Leetcode problems and its related topics. Similar to the above dataset, a graph can be drawn by considering problems and related topics as nodes and including an edge between a problem node and a topic node only if the problem is related to that topic thus forming a bipartite graph. There are 2307 nodes in this graph, with 2236 of them being problems and remaining 71 being topics. Also there are 6453 edges in this graph. On an average, each problem is associated to 2.88 topics and at most associated to 10 topics. The dataset can be found here.

All the nodes across all these datasets have been provided with a set of associated attributes. Notably, the maximum number of attributes linked to any given node is considerably small when compared with the overall number of nodes within the datasets.

[*] Electronic address: 202001259@daiict.ac.in
[†] Electronic address: 202001406@daiict.ac.in
[‡] Electronic address: 202003046@daiict.ac.in

III. Degree distribution

In the graph we analyse the degree distribution by plotting the degree on the x-scale and the frequency of the degree occurring in the graph in the y-scale. We then plot the degree vs CCDF plot. The CCDF for a value k is given by Probability(degree $\geq$ k).

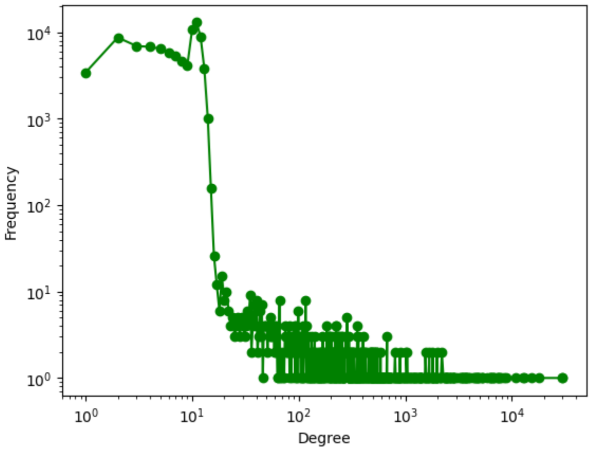

FIG. 2: Degree distribution of the whole graph

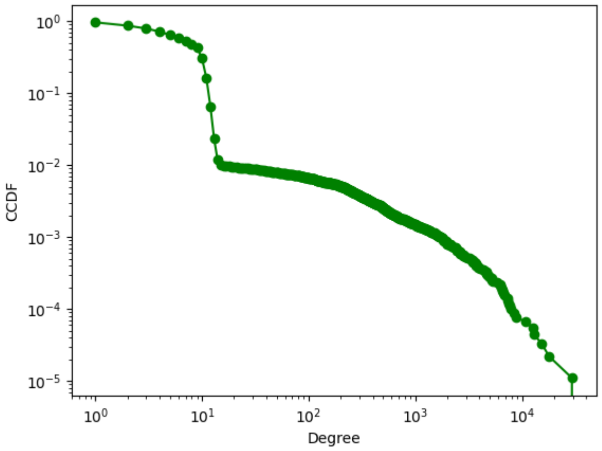

FIG. 3: Degree vs CCDF for the whole graph

In our case the degree vs CCDF plot is not a straight line so that implies that it is not a power law. Now it compels us to think that why a naturally occurring phenomenon would not occur. If we comprehend the data then we know that 89530 book nodes which are about 98.7% nodes have degrees between 1 and 17. And the rest 1178 genre nodes have degrees between 1 and 29743. So a higher proportion of nodes are having degrees in lower range so the CCDF value seem higher in the range 1 to 17. Now from 17 onwards only the genre nodes contribute to the degree which constitute about 1.3% of nodes so there is sudden fall in the CCDF curve.

If we plot the degrees of book nodes and genre nodes in separate plots then we can make some other observations

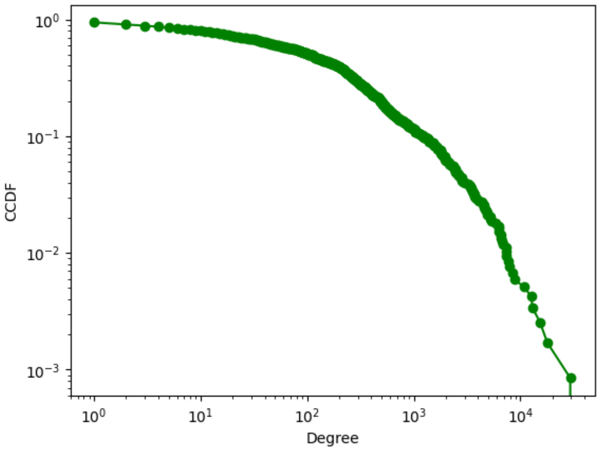

FIG. 4: Genre degree vs CCDF

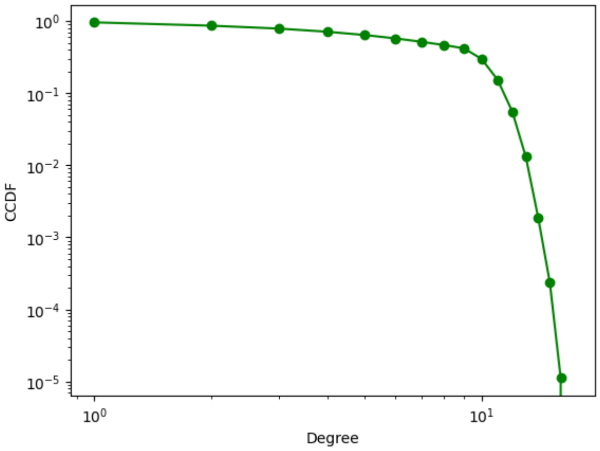

FIG. 5: Book degree vs CCDF

In the CCDF plot of genres, we can observe that it retains the same shape as the CCDF plot of the whole graph from degree 17 onwards. In the CCDF plot of books there is constant line from 1 to about 10 which indicates that the degree frequency remains almost constant and then suddenly there is a drop in the degree frequency which can be trend of the dataset as there are about 29% books with greater than 10 genres.

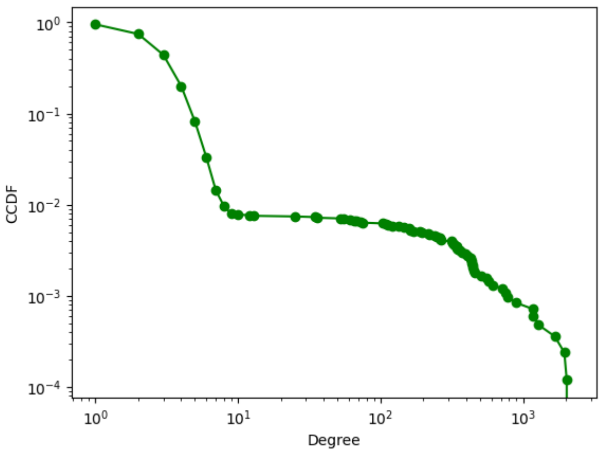

FIG. 6: Degree distribution for the whole graph

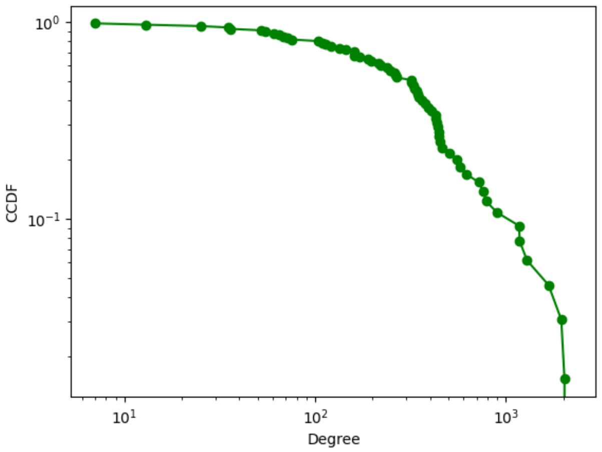

FIG. 7: Codeforces topics degree vs CCDF

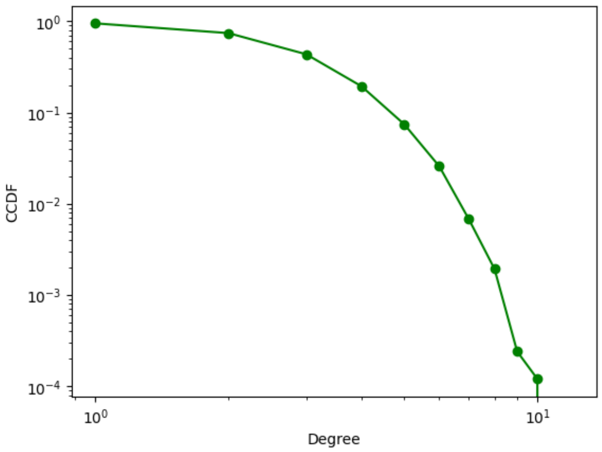

FIG. 8: Codeforces problems degree vs CCDF

In the degree distribution of the whole graph, we see that the degree in the range from 1 to 12 has a higher frequency than the rest of the graph. The same reasoning follows in the case also. As the number of codeforces problem nodes are 8279 which is about 99.2% of the total nodes have degrees in the range from 1 to 12. The rest nodes have degrees in the range from 7 to 2221. As the proportion of the rest nodes is lower it follows that the frequency is low in that range.

In the CCDF plot the topics resemble the portion of the degree distribution graph from degree 7 to 2221 and the CCDF plot of the problems resemble the degree distribution graph from degree 1 to 10.

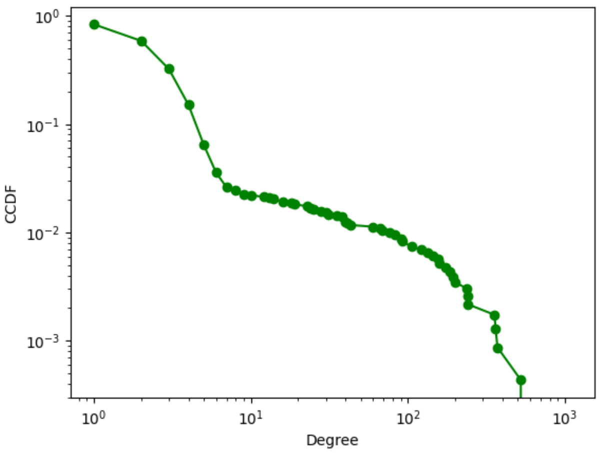

FIG. 9: Degree distribution for the whole graph

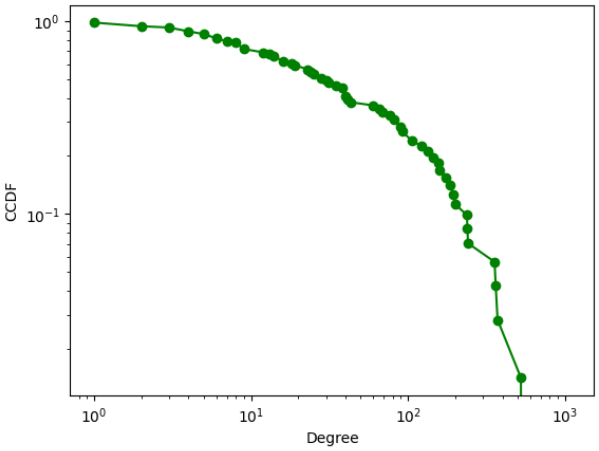

FIG. 10: Leetcode topics degree vs CCDF

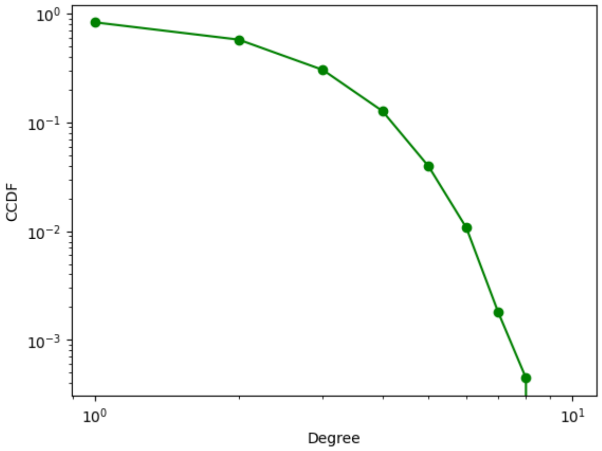

FIG. 11: Leetcode problems degree vs CCDF

In the degree distribution for the whole graph we see that there is a sudden drop in the CCDF from about degree 10. The reason for this is that about 2236 which is about 96.92% nodes have degrees between 1 and 10. So the degree frequency tends to be high in that region.

In the topic degrees CCDF plot, it resembles the portion of the graph of the degree distribution from degree 10 onwards. In the problem degrees CCDF plot, it resembles the portion of the degree distribution graph from degree 1 to 10.

IV. Community detection

Louvain Community Detection Algorithm is a method to extract the community structure of a network. This is a heuristic method based on modularity optimization.

The algorithm works in 2 steps. On the first step it assigns every node to be in its own community and then for each node it tries to find the maximum positive modularity gain by moving each node to all of its neighbor communities. If no positive gain is achieved the node remains in its original community.

The modularity gain obtained by moving an isolated node i into a community C can easily be calculated by the following formula: $\Delta Q = \frac{k_{i,in}}{2m} - \gamma \frac{\Sigma_{tot} \cdot k_i}{2m^2}$

where, m is the size of the graph, $k_{i,in}$ is the sum of the weights of the links from i to nodes in C, k_i is the sum of the weights of the links incident to node i, Σ_{tot} is the sum of the weights of the links incident to nodes in C and γ is the resolution parameter.

The first phase of the algorithm continues until no individual move can improve the modularity.

The second phase consists in building a new network whose nodes are now the communities found in the first phase. To do so, the weights of the links between the new nodes are given by the sum of the weight of the links between nodes in the corresponding two communities. Once this phase is complete it is possible to reapply the first phase creating bigger communities with increased modularity [1].

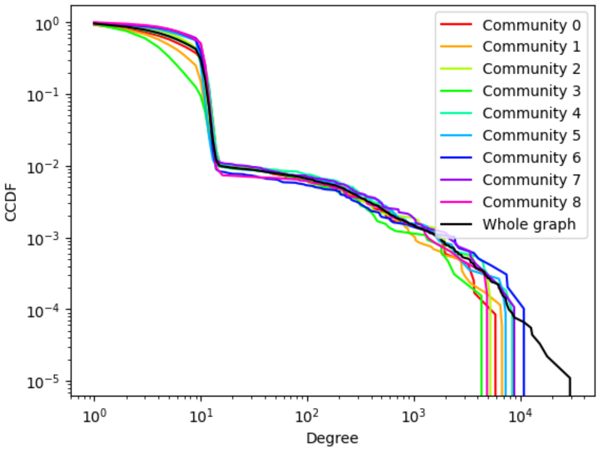

FIG. 12: Degree vs CCDF for the communities and the whole graph

In the figure, we can observe that 9 communities are detected by the algorithm and the CCDF of the communities is similar to that of the original graph and they retain the original properties of the dataset. We observed that the average ratio of genre nodes in a community and the ratio of genre nodes in the whole graph to be 0.012 so the community retains the proportion. Here, we can observe that the frequency of nodes having degrees between 1 and 10 is very high when compared to the other degrees, hence resulting in a steep decrease thereafter, which is also observed in the degree distribution of whole graph. So, we can say that similar to the whole graph even in the communities, frequencies of nodes with lower degree is high.

We calculated the probability that any two genres which belong to a community also co-exist in any book. For each genre in a book, we took the maximal subset of the intersection of the community and the book genre, and then we are divided it by the number of genres in the book. As we are to calculate probability, we divided the obtained value by number of books in the dataset and for the books dataset, the value is 0.48, meaning that with a probability of 0.48, we could say that prediction of two genres are related (belong to same coummunity) as they co-exist in a book too.

We anaylsed regarding why genres are co-existing in the communities. We know that the communities are formed such that the connections within the community are dense and with other communities sparse. The only connections that exist in our graph are between book and genre nodes. So we computed the average common neighbors and normalised common neighbors for each community and compared them with the average and normalised common neighbors for the whole graph.

Below formulae are given for a graph G, genre node list GL,

Common Neighbors of $i, j = C_{i,j} = C_{j,i} = \sum_k A_{ki}Akj$, In the above formula, k represents a node in the graph G, A represents the adjacency matrix of G and the value of A_{ki} is equal to 1 if there exists a book k belonging to a genre i.

Average Common Neighbors of graph G,

$$ACN = \frac{\sum_{i,j \in GL} C_{i,j}}{\binom{n}{2}}$$

where, n is the number of genres in the Graph G.

Normalised Common Neighbors of graph G,

$$NCN = \frac{\sum_{i,j \in GL} C_{i,j}}{(\binom{n}{2} * m)}$$

where, n, m are equal to the number of genres and books in the Graph G respectively.

Community Length	**Average Common Neighbors**	**Normalised Common Neighbors**
6954	22.68	0.0033
7660	18.96	0.0025
11093	11.51	0.0011
9688	34.31	0.0036
8714	22.56	0.0026
5988	75.11	0.0126
18623	6.26	0.0003
11215	8.35	0.0008
10773	17.08	0.0016

TABLE I: Average Common Neighbours Genres in different Communities

When considering the whole dataset, total number of nodes are 90708, value of average common neighbors is Average Common Neighbours is 2.02 and the value of Normalised Common Neighbours is 0.000022.

The results indeed show that the average common neighbors in the community are about 3.44 to 33.2 times larger than the whole graph. And then normalised common neighbors are about 177.77 to 466.67 times larger than the whole graph.

V. Modularity

Modularity is a measure of the structure of networks or graphs which measures the strength of division of a network into modules (also called groups, clusters or communities). Networks with high modularity have dense connections between the nodes within modules but sparse connections between nodes in different modules [2]. The order in which the nodes are considered can affect the final output. Monte Carlo methods are algorithms which use random numbers to sample and obtain numerical answers to many physical, social problems. In the Louvain algorithm the ordering happens using a random shuffle [1]. So we can apply Monte Carlo method to get the distribution of the community length and modularity values. We applied 100 monte carlo simulations, below are the histograms of modularity and community length for Books and Genres dataset.

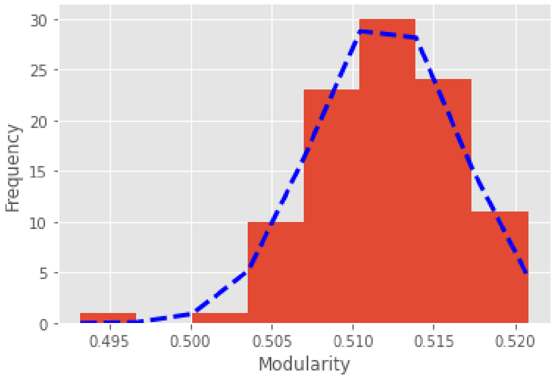

FIG. 13: Histogram of Modularity

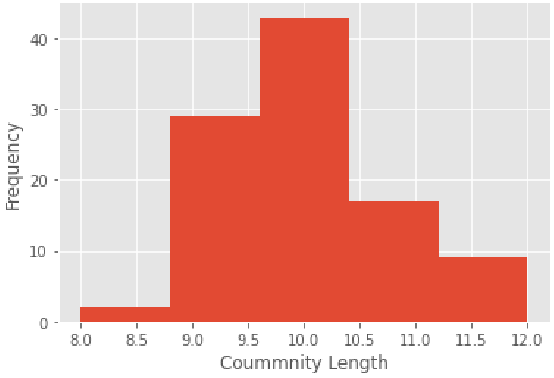

FIG. 14: Histogram of Coummnity length

We can see that the modularity value saturates in the range from 0.495 to 0.520 and also that the histogram of modularity graph and community length follows a normal distribution.

Below are the histograms for the codeforces dataset, for 1000 iterations.

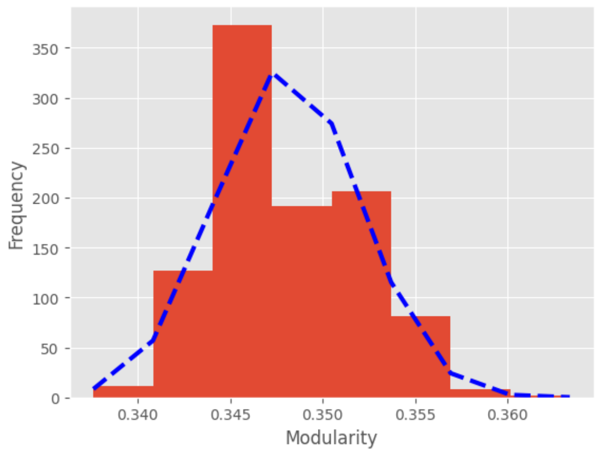

FIG. 15: Histogram of Modularity

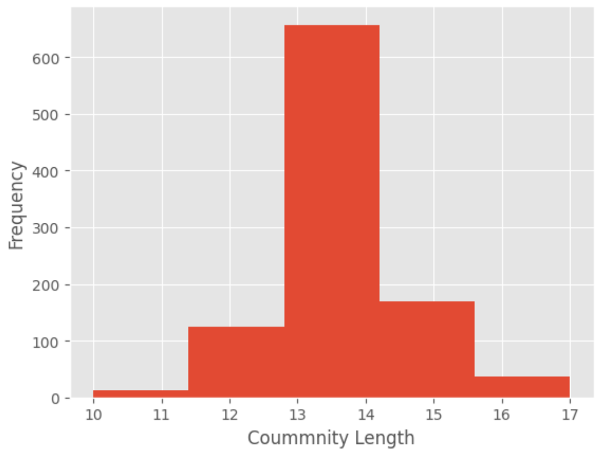

FIG. 16: Histogram of Coummnity length

VI. Degree centrality

The degree centrality of a node is simply its degree—the number of edges it has. The higher the degree, the more central the node is. This can be an effective measure, since many nodes with high degrees also have high centrality by other measures [3]. The degree of a node is given by $CD(i) = k_i = \sum_j A_{ij}$. Normalized degree centrality $CD_i^* = \frac{1}{n-1}CD(i)$. The purpose of our experiment was to examine how crucial degree centrality takes a part if a new node is added to current graph and edges are being drawn.

For the experiment we randomly shuffled and split the data of the books and genres in $80 : 20$ ratio such that 80% of the data would be used for training and 20% for testing the experiment. The 80% of the data was used for creating the graph and computing the degree centrality for all of the genre nodes. Then we computed the rank of each genre based on its degree centrality value. (Sorted the genres in descending order of its degree centrality value). Then we computed the frequency of each rank in the testing data; How often the rank occurs in the testing portion of the data. For genres that did not exist in training data we assumed them taking the largest rank. Using the rank obtained, we calculated $nDCG$ [4] score whose value is equal to 0.9958, meaning that our predictions are true with a probability 0.99.

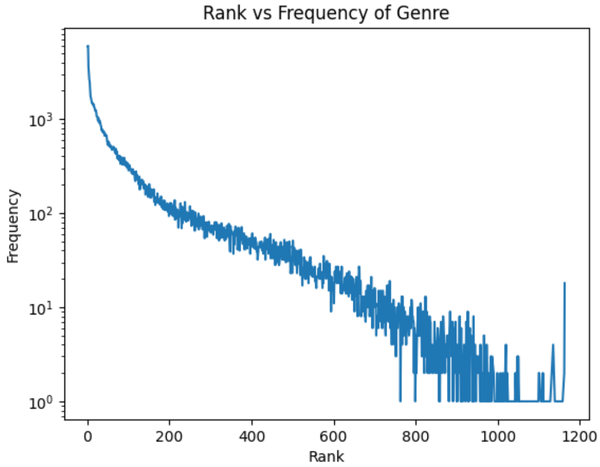

FIG. 17: Genre rank vs Frequency in the testing data

The result of the experiment clearly portraits the convention of ”rich getting richer” i.e. the genres with smaller ranks are amongst those with the highest frequencies and the trend continues for higher ranks with lower frequencies. There is sudden spike at the end which indicates the unknown genres in the construction of the graph which are present in the testing data.

VII. Link to the Code:

- https://colab.research.google.com/drive/1mGW4MH_IZMzGhTLj6xsx8mSmSUIwcbTL?usp=sharing
- https://colab.research.google.com/drive/1m1iqfavt0bToDAQ3823vae4Ee1oy69R-?usp=sharing

VIII. Conclusion

In conclusion, it is difficult to deduce from the network of nodes having certain attributes. But we can conclude a few things:

- The pattern is followed in all the datasets in the CCDF of the whole as well as individual bipartite sets. They all retain the same shape.
- The communities formed mimic the graph in the CCDF distribution as though it were a subgraph of the original graph.
- The average and normalised number of neighbors in the genres is quite larger than in the communities than the entire graph
- Monte Carlo simulations of the Modularity and Com-

munity length show a normal distribution and saturation in the range.

[1] Networkx contributors. Louvain communities - networkx documentation. https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.community.louvain.louvain_communities.html.

[2] Wikipedia contributors. Modularity (networks) — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Modularity_(networks)&oldid=1178597324, 2023. [Online; accessed 18-November-2023].

[3] Sciencedirect contributors. Introduction to social media investigation. https://www.sciencedirect.com/book/9780128016565/introduction-to-social-media-investigation.

[4] Wikipedia contributors. Discounted cumulative gain — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Discounted_cumulative_gain&oldid=1184326174, 2023. [Online; accessed 30-November-2023].