Online Social Network Analysis: A Co-authorship Network

Indrajit Ghosh
ighosh2@hawk.iit.edu
Computer Science
Illinois Institute of Technology

Yaswanth Chowdary Kosaraju
ykosaraju@hawk.iit.edu
Computer Science
Illinois Institute of Technology

February 2024

Abstract

This project uses co-authorship data to create a network to investigate how academics collaborate. Our primary focus was maintaining privacy while gathering information and building a network graph of between 400 and 500 distinct researchers. We mapped this network visually using graph analysis software like NetworkX and Gephi, and we computed key metrics including PageRank, Clustering Coefficient, and Degree Distribution. These metrics enabled us to pinpoint important researchers and determine the primary collaboration patterns. This paper outlines the actions we performed, the issues we encountered and resolved, and the lessons we discovered in scholarly collaboration.

1 Introduction

This project delves into the complex realm of academic collaborations by developing a co-authorship network, underscoring the interconnectedness of researchers across various fields to foster knowledge progression. In today's digital academic landscape, analyzing these networks is crucial for unveiling the collaborative frameworks essential for innovation. Utilizing co-authorship data, this study meticulously constructs a network graph to analyze academic collaboration patterns with sophisticated graph analysis tools. This introduction outlines the project's methodology, analytical approaches, and the significance of our discoveries for social network analysis. Focusing on Machine Learning within Computer Science, we examine co-authorship relations in scholarly publications. The project is structured into four key segments: Data Collection, Data Visualization, Network Measure Analysis, and Results Discussion, utilizing the open-access "**arXiv**" repository and free distribution for data collection, using NetworkX and Gephi for visualization. Metrics such as Degree Distribution, Clustering Coefficient, and PageRank are used to highlight important nodes and reveal the network's inherent structures.

2 Data Collection

Our project targeted the arXiv platform, a leading repository for electronic preprints in fields like physics, mathematics, computer science, and more, specifically focusing on the Computer Science - Machine Learning category (cs.LG). To collect data, we developed a Python script that utilized the requests library for HTTP requests to the arXiv API and xml.etree.ElementTree for parsing XML responses and the CSV package to convert output. We strategically selected the "cs.LG" category to map the co-authorship network within this dynamic research area. Our queries aimed to retrieve 250 articles per request, the maximum allowed, to construct a comprehensive dataset.

- **Dataset:** We have successfully compiled a dataset that includes a sizable number of authors, ranging between 400 to 500 nodes, all of whom have contributed to the field of Machine Learning through their publications. To organize this information, we've utilized a Python script that retrieves two separate CSV files. The first, titled "**nodes.csv**," catalogs each author, ensuring that every individual is distinctly represented as a single node within our dataset. The second file, "**edges.csv**," records the connections between these authors, illustrating the collaborative networks that exist as they have co-authored papers or worked on projects together within the Machine Learning community. This dataset is essential for understanding the collaboration patterns and will serve as the basis for our network analysis.

- **Challenges Encountered:** In our attempt to construct a co-authorship network for the machine learning community, we faced challenges such as adhering to arXiv API's limits, ensuring author and co-authorship uniqueness, and managing XML parsing complexities. Our strategies included utilizing the requests library for API interaction, leveraging Python's set to maintain data uniqueness, and employing xml.etree.ElementTree for efficient XML parsing. Meticulous logic was implemented to accurately record unique author pairs, ensuring a detailed and reliable dataset. These deliberate coding and data management approaches were essential for overcoming the encountered obstacles, resulting in a comprehensive and precise co-authorship network.

- **Privacy and Data Usage Policies:**The Data Usage Policies and Privacy Policy of arXiv are essential for anyone using its platform and API. The Privacy Policy, which is available at info.arxiv.org/help/policies/privacy_policy.html ,describes how arXiv gathers, utilizes, and distributes personal data and places a high priority on safeguarding user information during a variety of interactions, including submissions and platform use. Usually, it would include data rights, consent, and how to access and update personal information.
The data usage policy, available at arxiv.org/help/api, offers crucial rules on appropriate use for developers and researchers using the arXiv API. These standards enable academic research and non-commercial applications. This policy requires the appropriate identification and respectful usage of data to prevent undue strain on arXiv's servers and to guarantee the long-term viability of the API.

3 Data Visualization

On successfully gathering the CSV documents for the edge list and node list, we performed some data-cleaning activity, since arXiv provides author identification methods in their repository, further author disambiguation is currently out of scope and our sample dataset has only distinct authors.

- **Data Cleaning:** The data cleaning process involved removing duplicate rows from both the nodes and edges CSV files to ensure each entry is unique. Additionally, all names were standardized to Title Case (e.g., "john doe" to "John Doe") and any leading or trailing whitespace was eliminated. This not only improved the consistency and readability of the dataset but also prepared it for more accurate analysis by ensuring that each author and connection is represented distinctly. The cleaned data was then saved to new CSV files, 'edges_cleaned.csv' and 'nodes_cleaned.csv', for further use.

- **Visualization:** Our developed Python script outlines a process for visualizing a co-authorship network using data from **nodes_cleaned.csv**" and "**edges_cleaned.csv**" as input files. This involves loading the data into pandas DataFrames, initializing a graph with NetworkX,adding nodes and edges based on the DataFrame contents, and then drawing the network graph. Nodes represent authors, colored red, and edges represent co-author relationships, colored green. The visualization includes custom settings for layout and appearance, and a legend is added to explain the colors used for nodes and edges. This approach highlights the interconnectedness of authors within the machine-learning publishing community working together on some research papers. The nodes without any interconnections depict authors who have worked on research papers without any co-authors. For the below graph visualization, we have used the "spring layout" algorithm of NetworkX.

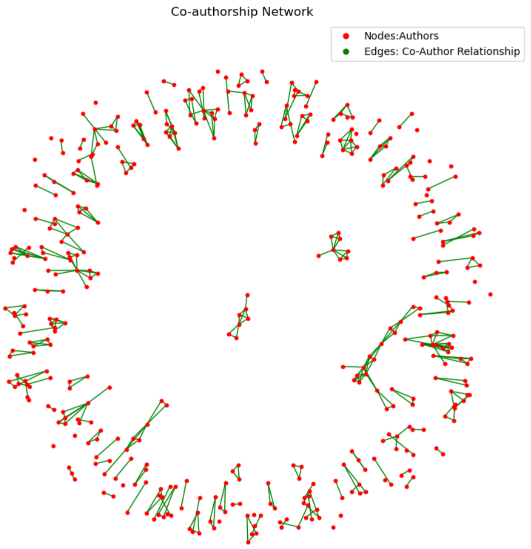

Figure 1: A Co-Authorship Network generated using NetworkX

For the sake of comparison, we have also generated the graph using another visualization tool called "Gephi"

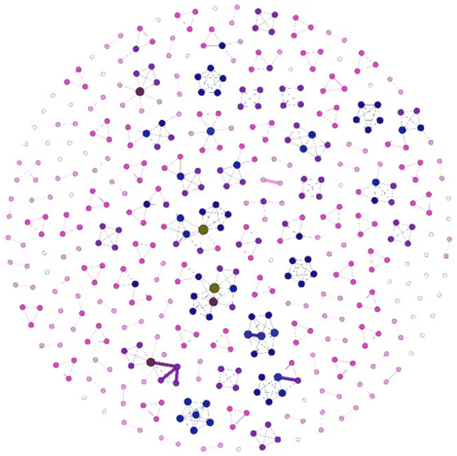

Figure 2: A Co-Authorship Network generated using Gephi

- **Final tool selection:** Even though Gephi was visually impressive, we opted for NetworkX over Gephi because it offered benefits like enhanced programming flexibility for network manipulation, efficient handling of larger datasets, and the ability to perform customized network analyses. NetworkX's integration into Python's ecosystem allows for seamless inclusion in broader data science and machine learning workflows, offering a more versatile approach to network analysis and research.

4 Network metric analysis

Network metric analysis involves examining various parameters to understand the structure, behavior, and importance of nodes within a network. Here's a brief overview of the three network metric parameters you've mentioned:

1. **Degree Distribution:** Degree distribution analyzes the distribution of node degrees in a network. The degree of a node is the number of edges incident to it. Degree distribution provides insights into how nodes are connected and the overall connectivity pattern of the network. It helps identify nodes with high connectivity (hubs) and nodes with low connectivity (peripheral nodes).

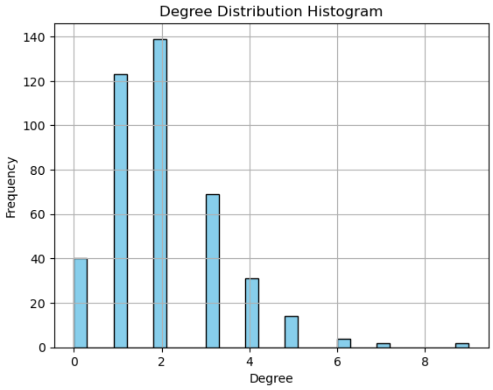

Figure 3: A histogram illustrating the Degree distribution network metric.

- **Methdology:** We used Pandas to read a CSV file containing graph data and constructed a graph representation using

NetworkX. Then, we calculated the degree of each node to analyze the degree distribution.

- **Results:** The histogram generated using Matplotlib provided insights into the distribution of node degrees within the network, highlighting hubs and peripheral nodes.

2. **Clustering Coefficient:** The clustering coefficient measures the degree to which nodes in a network tend to cluster together. It quantifies the extent to which nodes' neighbors are also connected. A high clustering coefficient indicates a highly interconnected network with clusters or communities, while a low clustering coefficient suggests a more random or sparse network structure.

   - **Methdology:** We utilized NetworkX to compute the average clustering coefficient of the network, indicating the degree of clustering or interconnected among nodes.
   - **Results:** The calculated clustering coefficient provided insights into the network's structure and the tendency of nodes to form clusters or communities. Conclusion: Analyzing the clustering coefficient helps understand the network's level of cohesion and the presence of tightly-knit communities.

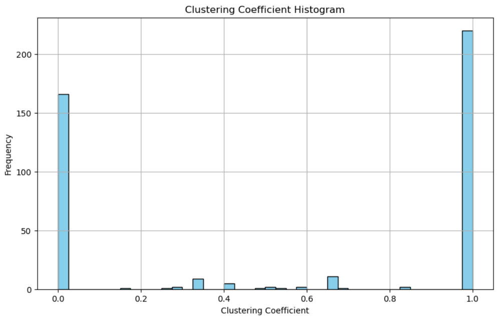

Figure 4: A histogram illustrating the clustering coefficient network metric.

3. **PageRank:** PageRank is a link analysis algorithm used to evaluate the importance of nodes in a network, originally developed by Google's founders Larry Page and Sergey Brin. PageRank assigns each node a score based on the number and quality of links pointing to it. Nodes with higher PageRank scores are considered more important or influential within the network.

   - **Methdology:** We applied the PageRank algorithm provided by NetworkX to assign PageRank scores to each node in the network.
   - **Results:** The PageRank scores, stored in a Pandas DataFrame, allowed us to visualize the distribution of node importance within the network using a histogram created with Matplotlib.

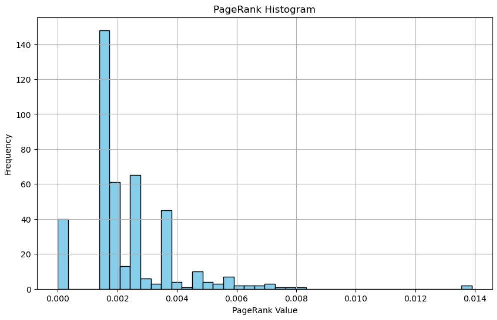

Figure 5: A histogram illustrating the PageRank network metric.

5 Discussion of Results

5.1 Visualization Analysis

- **Community Clusters:** The visualization from Figure 1 shows distinct clusters of nodes, which indicates groups of authors who collaborated. This suggests a community structure within the network, where certain researchers might work closely together, potentially within the same subfields of machine learning.
- **Isolated Pairs and Nodes:** We observe several pairs or small groups of nodes that are only connected to each other and not to the larger network. This reflects occasional collaborations or authors who have published a

limited number of papers. We can also observe some isolated nodes have no edges connected suggesting that a percentage of authors work independently on their papers.

- **Network Density:** The overall spread of the network with many disconnected components suggests a low density. This might imply that while there are collaborations, the machine-learning community has many independent clusters of researchers who do not interconnect widely with other clusters.

5.2 Network metric Analysis

1. **Degree distribution:**
   The degree distribution histogram from Fig 3 offers insights into the collaboration patterns among authors who publish research papers. It reveals that collaboration habits vary widely across the network. Some authors prefer to work independently, as evidenced by a degree value of '0', indicating they publish without co-authors. On the other hand, the bulk of the authors tend to collaborate but keep their circle relatively small, typically working with just one or two other researchers. This is reflected in the most common degree values of '1' and '2', highlighting a preference for limited partnerships in research endeavors.

   Moreover, while collaboration is common, extensive collaboration with a wide range of co-authors is rare. Only a small fraction of authors show a degree of '4', '6', or '8', indicating they have engaged with a larger number of different co-authors. This pattern suggests a few key figures or hubs within the network who play a pivotal role in bridging different research groups or areas. These individuals, with higher degrees of collaboration, contribute significantly to the diversity and interconnectivity of the research community. However, such extensive collaboration is not the norm, with most authors preferring more focused collaborative relationships. This distribution of collaboration degrees highlights the varied nature of academic partnerships, ranging from solitary work to broad, multi-author projects.

2. **Clustering Coefficient:**
   The histogram of clustering coefficients from Fig 4 displays two main peaks, meaning it's bimodal. This tells us there are two main types of collaboration patterns among the authors. A lot of authors either have a very high or a very low clustering coefficient.

   When authors have a high clustering coefficient, it means they are part of a close-knit group. In such groups, members often work together on research papers, indicating strong collaboration ties within these clusters. These are like tight circles of friends who tend to do a lot together.

   On the other hand, a low clustering coefficient points to authors who have connections that are more spread out. These authors might collaborate with others, but their co-authors don't necessarily collaborate. It's similar to someone who has friends from different groups, but these friends don't know each other well.

3. **PageRank:**
   The PageRank histogram from Fig 5 shows that the majority of authors have low PageRank scores, with a smaller number of authors having higher scores. This pattern means that most authors in the network don't have a wide-reaching influence, as their PageRank scores are low. However, there's a noticeable group of authors with higher PageRank values, indicating they have more influence over the network.

   These higher PageRank scores suggest that these authors play a key role in the collaboration network. They might be involved in many projects or be connected to many other authors, making them central figures. Their central position could make them crucial for spreading ideas, research findings, and influencing the direction of research within the network.

   This distribution is interesting because it highlights the varied roles authors play in the research community. While many contribute within smaller circles or specific areas, a select few have a broader impact, connecting different groups and fields. Understanding who these influential authors are and how they contribute to the network could provide insights into how research evolves and spreads within the community.

5.3 Further Questions

The Co-authorship network provides light on a number of important aspects of the dynamics of collaboration within the machine learning research community. The information suggests that a few key individuals might play a significant role in each of their clusters and have a lot of influence throughout the community. The patterns of cooperation, whether they are primarily cross-disciplinary or within the same fields, may provide insight into the nature of research collaborations. Furthermore, figuring out how these partnerships have evolved over time would provide a window into how machine learning research is developing, pointing out trends and changes in the emphasis on collaboration.

5.4 Future Next Steps Investigations

To further understand the Co-authorship network, we aim to conduct a detailed node-level analysis to pinpoint key authors and assess the structure of collaborations. We'll also deploy community detection algorithms to uncover how different areas within machine learning are connected. Additionally, a temporal analysis could shed light on how collaborative patterns have evolved, providing insights into the field's development over time.

6 Conclusion

In summary, this project gave us practical experience in collecting, showing, and studying data from the arXiv platform to understand how networks and co-author relationships work. We did this by writing code to get the data, using tools like Networkx and Gephi to visualize the data, and calculating metrics like degree distribution, clustering coefficient, and PageRank to see how the network is structured.

The report we made shows step-by-step how we collected, displayed, and analyzed the co-authorship data. We also made sure to follow good practices, like checking the platform's privacy rules before collecting any data. By using graphs and numbers, we learned a lot about how authors are connected and how information flows between them.

There's still more we can do to make our analysis better, like using bigger datasets, adding more ways to measure the network, and getting more information about the data. But for now, we've learned a lot about how to collect, show, and measure co-authorship networks, which sets us up well for doing more advanced work in the future.

7 GitHub Repository

The code, generated node, and edge list along with project details can be found in the below GitHub repository-
https://github.com/ighosh2/CoAuthorshipNetwork

8 References

1. https://networkx.org/documentation/stable/reference/index.html

2. https://arxiv.org/

3. https://gephi.org/users/

4. https://www.tutorialspoint.com/graph_theory/index.html

5. https://medium.com/nerd-for-tech/co-author-network-analysis-using-deepwalk-ac7fd63c82aa

6. Pandas. (n.d.). pandas - Python Data Analysis Library. https://pandas.pydata.org/

7. Gephi - The Open Graph Viz Platform. (n.d.). Gephi- The Open Graph Viz Platform. https://gephi.org

8. NumPy package-https://numpy.org/