Visualization of Heterogenous Network of Publication Data

A Project

presented to the

Faculty of

University of North Carolina,

Greensboro

For the Degree

Master of Science

in

Computer Science

Advisors : Dr. Lixin Fu, Dr. Chunjiang Zhu

by

(Vaishnavi Peddireddy)

(Dec 2023)

© (2023 and Vaishnavi Peddireddy)

A logo of a university of north carolina

Description automatically generated

**Table of Contents**

1. Introduction ……………………………………………………3
2. Environment Setup ……………………………………………………3
3. Data Acquisition ……………………………………………………4
4. Data Preparation …………………………………………………4
5. Graph Construction …………………………………………………5
6. Visualization …………………………………………………… 7
7. Querying the Graph ……………………………………………………13
8. Advanced Topics ……………………………………………………14
9. Troubleshooting ……………………………………………………14
10. Acknowledgements ……………………………………………………16
11. Conclusion ……………………………………………………17
12. References …………………………………………………..18

**1. Introduction**

The field of bioinformatics is rich with complex and interrelated data that necessitates robust analytical techniques for meaningful interpretation. This document outlines the methodology for constructing and analyzing a heterogeneous graph derived from the PubMed dataset, a comprehensive resource of medical literature. Our objectives are to:

* Create a graph-based representation of the dataset.
* Apply community detection algorithms to uncover inherent structures.
* Develop an interactive visualization for exploratory data analysis.

This documentation will serve as a guide for replicating the analysis, providing insights into the tools and techniques used in the process.

One of the key objectives of this project is to make complex network analysis accessible and insightful to users, ranging from academic researchers to students. By providing an intuitive interface for exploring the ACM network, the project bridges the gap between sophisticated data analysis and user-friendly visualization.

The findings from this project are expected to have significant implications for understanding collaboration patterns in academic research. By uncovering the hidden structures within the ACM network, the project contributes to the broader field of network analysis and offers a model for similar studies in other academic or organizational contexts.

**2. Environment Setup**

Before delving into the analysis, it is critical to prepare the computing environment. The following software and libraries are required:

* Python 3.8 or higher: The core programming language used.
* NetworkX: A Python library for creating and analyzing complex networks.
* Pyvis: A Python library for visualizing networks interactively.
* MongoDB: A NoSQL database used to store and manage graph data.

**3. Data Acquisition**



**4. Data Preparation**

Overview

The data preparation phase was informed by our prior experience with the **link.dat** dataset from PubMed. This dataset, structured as a tab-delimited edge list, provided insights into how to effectively handle and represent the relationships within the ACM data graph. By reversing the techniques applied to the **link.dat** file, we tailored our approach to suit the ACM dataset's unique structure.

Informed by the Link Data

The **link.dat** dataset comprised of detailed linkage information with edge attributes like 'type' and 'weight'. Using NetworkX, we read and processed this data into a directed graph, which then served as a blueprint for handling the ACM data:

This approach revealed the importance of edge attributes in understanding the depth and strength of academic relationships, guiding us to apply similar considerations to the ACM data.

Application to ACM Data

Leveraging the insights gained, we processed the ACM dataset by:

1. **Adapting the Edge List Reading Technique**: We applied the **read\_edgelist** method used for the **link.dat** to the ACM dataset, ensuring that each paper, author, and venue was accurately represented as nodes within the graph.
2. **Integrating Edge Attributes**: Attributes analogous to 'type' and 'weight' were incorporated into the ACM graph to reflect the nature of authorship and publication strength, mirroring the approach taken with the PubMed data.
3. **Normalization and Cleaning**: Inspired by the preprocessing of **link.dat**, we normalized and cleaned the ACM data to facilitate a direct comparison between the two datasets and to maintain consistency across our analyses.

Reverse Engineering for Enhanced Analysis

The process of reverse engineering the **link.dat** data handling methods for the ACM dataset allowed us to:

* Develop a robust and consistent framework for data preparation across different academic datasets.
* Enhance the ACM graph with additional dimensions of analysis, such as the weighted importance of papers and the role of venues in academic collaborations.
* Ensure that the prepared data was primed for the application of community detection algorithms, with a reliable representation of the complex network of academic relationships.

Outcome

The preparation of the ACM dataset, guided by our experience with **link.dat**, resulted in a well-structured and enriched network graph. This graph was not only more detailed but also more aligned with proven data handling techniques, setting a strong foundation for the subsequent phases of network analysis and community detection.

**5.Graph Construction Techniques**

The graph construction utilized NetworkX to erect a directed graph, effectively capturing the directional flow of information within the academic landscape. The nodes represented entities such as authors, papers, and venues, while the edges depicted the relationships among these entities. Attributes were meticulously assigned to both nodes and edges, encapsulating characteristics such as publication count and collaboration frequency. The directed graph was later converted to an undirected format to facilitate specific clustering algorithms.

PubMed data with 4 types of nodes

* Genes
* Diseases
* Species
* Chemicals

10 different edge types

* GENE-causing-DISEASE
* DISEASE-and-DISEASE
* CHEMICAL-in-GENE
* CHEMICAL-in-DISEASE
* CHEMICAL-and-CHEMICAL
* CHEMICAL-in-SPECIES
* SPECIES-with-GENE
* SPECIES-with-DISEASE
* SPECIES-and-SPECIES

**Getting node pairs with top12 and bottom 12 aggregated similarity scores**

**Top 12**

[((18040, 8394), 0.6667), ((62334, 56628), 0.6269), ((22455, 2498), 0.625), ((49214, 56628), 0.5906), ((826, 55484), 0.589), ((8008, 5357), 0.5757), ((27757, 5357), 0.5731), ((55365, 22455), 0.5714), ((50093, 58380), 0.5654), ((14439, 58380), 0.558), ((31714, 50093), 0.5545), ((19446, 826), 0.5528)].

**Bottom 12**

[((55365, 30122), 0.1407), ((34455, 58380), 0.1399), ((37323, 30122), 0.1313), ((5879, 30122), 0.1306), ((3562, 53458), 0.1291), ((47671, 41252), 0.128), ((28816, 58380), 0.119), ((55365, 3562), 0.1154), ((3562, 983), 0.101), ((2498, 3562), 0.0842), ((37323, 3562), 0.0842), ((32267, 7360), 0.0589)]

**6. Visualization**

**Graphs Using PubMed Dataset :**

## Graph plotted for top 12 node pairs and their neighbors:

**A network of colorful dots and lines

Description automatically generated**

## Graph plotted for bottom 12 node pairs and their neighbors:

A network of lines and dots

Description automatically generated

**Graphs For ACM Dataset:**

**A screen shot of a computer code

Description automatically generated**

A network of blue dots and lines

Description automatically generated

The nodes (points) in the graph represent authors, venues, and papers. The edges (lines) between the nodes indicate the relationships between these entities:

An edge connecting an author to a paper suggests that the author has written that paper.

An edge connecting a paper to a venue suggests that the paper was presented at that venue.

There are edges connecting authors to venues, implying that the authors have presented at those venues.

The structure of the graph shows how authors are interconnected through their papers and the venues where they present their work. Large clusters in the graph could indicate prolific authors, popular venues, or highly cited papers.

**Graphs Generated for the User Interface :**

**A screen shot of a computer program

Description automatically generated**

**Input :**  Search with the paper ID and it will give us the output with the authors related to it and at which Venue it is presented.

**A screenshot of a search box

Description automatically generated**

**Output :**  In the output the circles represent authors, square represent paper and triangle as venue

**A diagram of a company

Description automatically generated**

**Input :** Search with the author Id or author name and it will give us the output with the authors related to each other through a common paper.

**A screenshot of a computer

Description automatically generated**

**Output : :**  In the output the circles represent authors, square represent papers.

**A diagram of a diagram

Description automatically generated**

**Input :**  Now all the papers and its related authors which were presented at a venue will be displayed.

**A search box with text

Description automatically generated**

**Output :** The output shows all the papers and its related authors.

**A close-up of a diagram

Description automatically generated**

**Input :**  Now the input will be the single author ID or author name using that we could get the list of work done by the author.

A close up of a screen

Description automatically generated

**Output :**  It gives us the visualization of author’s work

A diagram of a diagram

Description automatically generated

**7. Querying the Graph**

**7.1 Purpose of Graph Queries**

Graph queries are essential for extracting actionable insights from complex networks. They allow us to pinpoint specific relationships, measure properties of nodes and edges, and understand the overall structure of the graph. In the context of the ACM heterogeneous graph, queries can help us identify influential authors, collaborative clusters, and the impact of particular papers or venues.

**7.2 Querying with NetworkX**

NetworkX provides a rich set of functions for querying graphs. These functions range from simple lookups, like retrieving all neighbors of a node, to more complex searches, such as finding the shortest path between nodes or extracting subgraphs that meet certain criteria.

**7.4 Insights from Queries**

The ability to query the graph effectively opens up a range of analytical possibilities. For instance, by querying the graph, we can identify the most prolific authors, the most cited papers, or emerging trends in research topics. These insights are not only valuable for academic researchers but also for institutions, publishers, and funding bodies looking to understand the dynamics of academic collaboration and knowledge creation.

**7.5 Visualizing Query Results**

To make the data more accessible and understandable, visualization is key. Visualizing the results of a graph query can reveal patterns and insights that may not be immediately obvious from raw data alone.

**8: Advanced Topics**

Scaling New Heights: Performance and Efficiency

**Tackling the Giants:** Data can be overwhelming, especially when it grows exponentially. It's like our project suddenly decided to go on a data-feeding frenzy! We need smarter ways to handle this. Think of it as preparing a feast for a city using just your home kitchen - challenging, right? Here’s where distributed computing and efficient data structures come into play. We’ll explore how technologies like Spark GraphX or Dask can help us cook up solutions for our data-hungry graph.

**Fine-Tuning for Speed:** Speed is of the essence, particularly with a massive dataset like ACM's. It's like fine-tuning a race car; every millisecond counts. We delve into the art of making our queries zippy, using indexing and clever algorithms that know the shortcuts within our graph.

**9: Troubleshooting**

**9.1 Data Preparation Hurdles**

**Incomplete or Missing Data**

* Real-world datasets are rarely perfect. Missing data can lead to skewed analysis and inaccurate results. We'll discuss strategies like imputation techniques or using statistical methods to handle missing values.

**Inconsistent Data Formats**

* Working with data coming from different sources often means dealing with various formats. We'll cover how to standardize these formats for seamless integration into your analysis pipeline.

**9.2 Graph Construction Challenges**

**Duplicate Entries in the Graph**

* Duplicates can distort graph metrics and lead to false insights. We'll explore methods to identify and remove duplicate nodes or edges, ensuring the integrity of our graph.

**Incorrect Edge Connections**

* Misconnections in a graph can change the entire narrative of the data story. We'll delve into validation techniques to ensure that the relationships represented in the graph accurately reflect the underlying data.

**9.3 Database Integration Snags**

**Database Connection Issues**

* Connectivity problems with MongoDB can halt progress. We'll look at common causes of connection failures and how to troubleshoot them, including configuration checks and network diagnostics.

**Inefficient Queries Slowing Down Performance**

* Optimizing database queries is crucial for performance, especially with large datasets. We'll discuss how to write efficient MongoDB queries and use indexing to speed up response times.

**9.4 Visualization Pitfalls**

**Overwhelming and Cluttered Graph Visualizations**

* Overly complex visualizations can be hard to interpret. We'll cover techniques to simplify and clarify visualizations, such as using filters, adjusting layouts, or highlighting key information.

**Non-Responsive Interactive Elements**

* Interactive features that don't work as expected can be frustrating. We'll troubleshoot common issues with interactive visualizations, like event handling or rendering problems.

**9.5 Debugging Strategies**

**Interpreting Error Messages**

* Error messages can be cryptic. We'll provide tips for decoding and acting on error messages, turning them from frustrations into useful guides.

**Unexpected Code Behavior**

* When code doesn't behave as expected, systematic debugging is key. We'll discuss approaches to debug your code, including using logging, breakpoints, and unit tests to isolate and resolve issues.

**10.Acknowledgments**

As we reach the conclusion of this journey, it's important to pause and express gratitude to those who have contributed to the success of this project. The journey through data analysis, graph construction, and the exploration of the ACM dataset has been both challenging and enlightening, and it would not have been possible without the support and guidance of many.

**Mentors and Advisors:** First and foremost, a heartfelt thank you to our academic and professional mentors(Dr.Fu,Dr.Zhu). Their expertise and insights have been invaluable in guiding the direction of this project. Their patience and willingness to share knowledge have not only helped in overcoming technical hurdles but also in shaping our analytical thinking.

**Team Members:** To my fellow team members, your collaboration has been the backbone of this project. The countless brainstorming sessions, code reviews, and collaborative problem-solving efforts have been integral to our success. Your diverse perspectives and skill sets have enriched this project in countless ways.

**Open-Source Community:** A special mention to the open source community, whose tools and libraries have been instrumental in our work. The contributions of countless developers around the world have made it possible to tackle complex problems more efficiently. The open-source spirit of collaboration and knowledge sharing is truly inspiring.

**11.Conclusion**

As we conclude this comprehensive exploration into constructing and analyzing a heterogeneous graph using the ACM dataset, it's essential to reflect on the journey we've undertaken and the insights we've gained. This endeavor was more than just an academic exercise; it was a deep dive into the intricate world of data science, graph theory, and database management.

**Key Accomplishments:**

* Successfully constructed a complex heterogeneous graph representing the multifaceted relationships within the ACM dataset.
* Implemented and demonstrated various techniques for effective data preprocessing, graph analysis, and visualization.
* Integrated advanced machine learning techniques to extract deeper insights from the graph data.
* Addressed and overcome numerous technical challenges, offering practical solutions and strategies for troubleshooting.

**Learnings and Implications**

This journey has provided valuable insights into the nature of academic collaboration networks. The visualization and analysis of the graph have not only illuminated the interconnected nature of research and publication but have also showcased the potential of graph theory in uncovering hidden patterns and relationships in large datasets. The skills and methodologies developed through this project have broad applications, extending beyond academia to industries that rely on network analysis and data-driven decision-making.

**Future Directions**

The field of graph analysis is dynamic and rapidly evolving. Future work can expand upon this project in several ways:

* Incorporating more diverse datasets to create more comprehensive graphs.
* Applying more sophisticated machine learning models, like Graph Neural Networks, to predict trends and patterns.
* Exploring real-time data analysis for dynamic graph updates and insights.
* Enhancing the scalability and efficiency of the graph analysis process for larger datasets.

**Final Reflections**

This report stands as a testament to the power of combining rigorous data analysis with innovative graph theory techniques. It highlights the importance of continual learning and adaptation in the ever-changing landscape of data science. We hope this document serves as a valuable resource for others embarking on similar journeys and contributes to the growing body of knowledge in this exciting field

**12.References**

Perrone, Giancarlo, Jose Unpingco, and Haw-minn Lu. "Network visualizations with Pyvis and VisJS." *arXiv preprint arXiv:2006.04951* (2020).

Perrone, G., Unpingco, J., & Lu, H. M. (2020). NETWORK VISUALIZATIONS WITH PYVIS AND VISJS. arXiv preprint arXiv:2006.04951.

Perrone, Giancarlo, Jose Unpingco, and Haw-minn Lu. "NETWORK VISUALIZATIONS WITH PYVIS AND VISJS." arXiv preprint arXiv:2006.04951 (2020).

Perrone, G., Unpingco, J. and Lu, H.M., 2020. NETWORK VISUALIZATIONS WITH PYVIS AND VISJS. arXiv preprint arXiv:2006.04951.

Perrone G, Unpingco J, Lu HM. NETWORK VISUALIZATIONS WITH PYVIS AND VISJS. arXiv preprint arXiv:2006.04951. 2020 Jun.