Amrita Vishwa Vidyapeetham

Amrita School of Computing

Technical Report

Leader-Based Community Detection Algorithm in Attributed Networks

Team : DB4

AM.EN.U4CSE20321	:	Devesh Kumar V V
AM.EN.U4CSE20339	:	Krishnpriya Dinesan
AM.EN.U4CSE20366	:	Sreesankar S
AM.EN.U4CSE20152	:	Pranav B Nair

Project Guide : Deepthi L.R Project Coordinator : Siji Rani S

Signature : Signature :

December 20, 2023

Contents

Abstract

This project, titled ”Leader-Based Community Detection Algorithm in Attributed Networks,” addresses the challenge of identifying central leaders and organizing communities within complex networks enriched with attributes. The relevance of this problem lies in its potential to revolutionize decision-making, optimize resource allocation, and enhance collaboration efficiency across diverse domains. Motivated by the need for nuanced insights into leadership dynamics, our project proposes a novel algorithm integrating data analysis techniques and advanced community detection methods. Persistent challenges include the intricate interplay between attributes and network structures. Our work aims to overcome these challenges, contributing to a deeper understanding of attributed networks for informed decision-making and resource optimization.

1 Introduction

1.1 Motivation

In the modern landscape of network analysis and community detection, the identification of central leaders and the organization of communities within attributed datasets has become imperative. This project, titled ”Leader-Based Community Detection Algorithm in Attributed Networks,” aims to address this need through the application of advanced data analysis techniques and community detection algorithms.

1.2 Problem Description

The focus of our project is on identifying central leaders and organizing communities within attributed datasets.The complex interplay between attributes and network structures poses a challenging problem that traditional community detection algorithms struggle to address adequately. This project seeks to unravel leadership dynamics and community structures within these complex networks.

1.3 Example Scenario

Consider a corporate environment where individuals possess diverse skills and expertise, contributing to the organization's success. Identifying key leaders and understanding how different communities collaborate can significantly enhance organizational efficiency. For instance, in a project requiring cross-functional collaboration, our algorithm aims to unveil leaders orchestrating collaborations and optimize community structures for seamless information flow.

1.4 Persisting Challenges

The persistent challenges lie in the intricate interplay between attributes and network structures. Existing solutions often overlook the nuanced relationships within attributed networks, resulting in a lack of precision in identifying central leaders and organizing communities.

1.5 Existing Solutions

While traditional community detection algorithms, such as modularity or Louvain, have proven effective in simpler network structures, their application to attributed networks remains a challenge. Our approach integrates data analysis techniques with advanced community detection algorithms to provide a more accurate understanding of relationships within attributed networks.

1.6 Solution Approach

Our approach involves the application of state-of-the-art data analysis techniques and community detection algorithms tailored for attributed networks. By integrating these methods, we aim to provide insightful visualizations that unravel leadership dynamics and community structures within the dataset.

1.7 Specific Research Objectives

- Develop a leader-based community detection algorithm considering both network structure and attribute information.
- Implement and evaluate the algorithm on diverse attributed datasets to assess its effectiveness in different scenarios.
- Compare the performance of our algorithm with existing community detection methods in attributed networks.
- Provide a user-friendly visualization tool to interpret and explore the identified leadership dynamics and community structures.

1.8 Contributions of this Work

The contributions of our work include:

1. Proposing a novel leader-based community detection algorithm tailored for attributed networks.
2. Offering insights into leadership dynamics and community structures that enhance decision-making processes.
3. Providing a tool for visualizing and interpreting the results, fostering a deeper understanding of attributed networks.

Through these contributions, we aim to advance the field of community detection in attributed networks and empower decision-makers with valuable insights for efficient collaboration and resource allocation.

2 Literature Survey

2.1 Leader Aware Community Detection in Complex Networks

”Leader-aware Community Detection in Complex Networks” (Springer, 2019) proposes a pioneering algorithm for community detection in complex networks. It automatically identifies structures and leaders but is constrained by a static network assumption, impacting performance on large networks. Future research should explore dynamic adaptability, diverse leadership score definitions, and scalability enhancements, advancing community detection methodologies.

2.2 Leader Similarity-Based Community Detection Approach for Social Networks

The 2020 IEEE paper introduces the Leader Similarity Based Community Detection (LSBCD) algorithm, addressing challenges in simultaneous community detection and leader selection in social networks. While focusing on non-overlapping communities, its limitations in handling extremely large and dynamic networks prompt ongoing research for enhancing real-world applicability, with evaluations conducted on dynamic social media platforms, exploring LSBCD's scalability for big data scenarios.

2.3 Leader-Based Community Detection Algorithm in Attributed Networks

The 2021 IEEE paper introduces the TALB method, a Leader-Based Community Detection Algorithm for Attributed Networks. TALB integrates topological and attribute information using a dependency tree, tackling efficiency challenges in large networks. The authors propose parallelization and distributed computing for scalability, emphasizing the algorithm's robustness through real-world evaluations under perturbations.

2.4 Community Detection in Attributed Networks Using Graph Wavelets

”Community Detection in Attributed Networks Using Graph Wavelets,” introduces a groundbreaking approach to community detection. Employing spectral graph wavelets, it tackles challenges like computational complexity and sensitivity to graph structure. The method filters attributes, creating a new network capturing nuances at various scales. The authors discuss extending the framework and applying other kernel functions for detecting multi-scale community structures, offering valuable insights into analyzing complex attributed networks.

Summary of the background study is presented in Table 1

Table 1: *Summary of the Related works*

Paper	Title/ Year	Problem addressed	Contributions	Limitations	Open Problems
Leader-aware Community Detection in Complex Networks [Springer]	2019	Study the problem of community detection in complex networks and propose a novel method, which can detect community structures automatically as well as identify community leaders.	Leader-aware community detection algorithm that incorporates the concept of leadership to improve the accuracy of community detection	Assumes a static network, where the connections between nodes remain constant throughout the analysis. The algorithm may not perform optimally on very large networks	Investigating the algorithm's performance on dynamic networks, exploring the impact of different leadership score definitions, and enhancing the algorithm's scalability for large-scale networks.

Table 1: *Summary of the Related works*

Paper	Title/ Year	Problem addressed	Contributions	Limitations	Open Problems
Leader Similarity Based Community Detection Approach for Social Networks [IEEE]	2020	Addresses the challenge of simultaneous community detection and leader selection in social networks without prior knowledge of community sizes and numbers.	Proposes the ”Leader Similarity Based Community Detection (LSBCD)” algorithm	Algorithm's focus is on non-overlapping communities Another limitation is on the applicability to extremely large and dynamic networks(real-time social media platforms)	Extending LSBCD for overlapping communities in social networks to enhance real-world applicability. Testing on dynamic social media platforms to assess its performance.Scalability for big data scenarios.
Leader-Based Community Detection Algorithm in Attributed Networks [IEEE]	2021	Need for an improved community detection method that incorporates both topological and attribute information to maintain the integrity of information in complex networks.	Introduces TALB, a leader-based method that integrates topological and attribute information. A dependency tree is formed by combining attribute similarity matrices with network topology.	Faces efficiency challenges in large, dense networks. Incorporating topological and attribute information may increase computational overhead.Addressing this requires efficient algorithms and extensive evaluations for real-world applicability	Parallelization, distributed computing, or sampling approaches to improve scalability. Also investigate the robustness of the TALB method under different perturbations, such as node removal, edge addition, or changes in attribute data.

Table 1: *Summary of the Related works*

Paper	Title/ Year	Problem addressed	Contributions	Limitations	Open Problems
Community Detection in Attributed Networks Using Graph Wavelets. [IEEE]	2022	Graph signal processing-based approach to community detection in attributed networks.	Spectral graph wavelets to filter the attributes and constructs a new network from the graph filtered attributes across different scales.	Computational complexity, sensitivity to graph structure, and challenges with continuous attributes and attribute dependencies	Other kernel functions and the extension of this framework to detect multi-scale community structure

3 Proposed Methodology

Two algorithms Implemented:

- **Leader Selection Algorithm:** This algorithm will focus on identifying influential leaders within the network, considering both topological characteristics and attribute information. It should effectively select nodes that can guide the community detection process.

- **Community Detection Algorithm:** This algorithm will form the core of the project, aiming to detect communities by combining attribute similarity matrices with network topology. It should efficiently identify cohesive and meaningful communities within the attributed network.

3.1 Leader Selection Algorithm

Here we will be using the Eigen vector centrality and the attribute information to compute a Composite/Leadership score which will be later considered to sort and rank the total list to get the top leader's list.

- **Eigen Vector Centrality:** It is a measure of the influence or importance of a node in a network, based on the concept that connections to high-scoring nodes contribute more to a node's centrality.

3.2 Community Computation

Community computation involves using algorithms to identify clusters or subgroups within a network, revealing patterns and structures. It's crucial in network science, aiding the understanding of relationships and organization in various systems, from social networks to biological networks.

- **Louvain Method:** This method optimizes a modularity score to find communities in a network. It can be extended to attributed graphs by considering node attributes in the modularity calculation.

- **Greedy Modularity:** The key idea behind this algorithm is to iteratively add nodes to communities in a way that maximizes the modularity of the resulting partition.

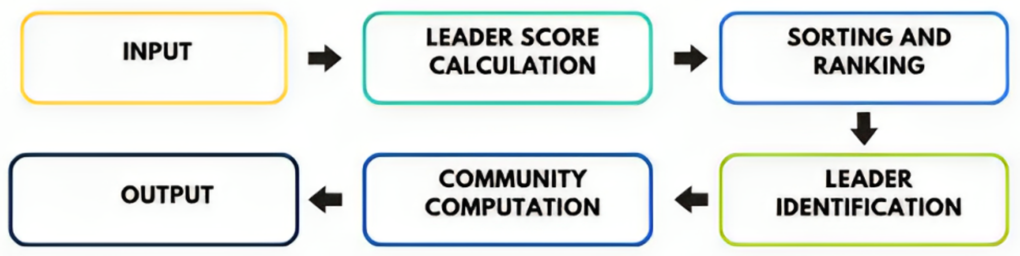


Figure 1: *Proposed Modules*

3.3 Algorithms

3.3.1 Louvain Method/ Greedy Modularity Algorithm

The Louvain Method, also known as Greedy Modularity, is a popular algorithm for community detection in networks. The algorithm aims to optimize a quality function called modularity, which measures the strength of division of a network into communities.
Here are the key steps:

- **Initialization:** Start with each node in its own community.
- **Iterative Optimization:** Iteratively consider moving each node to its neighbor's community, assessing the change in modularity.
  Greedily move the node to the community that maximizes the modularity gain.
- **Merge Communities:**Construct a new network where communities from the previous step are treated as nodes.
  Repeat the optimization process on this new network.
- **Repeat:** Continue the process iteratively until no more modularity improvement is possible.
- **Output:**The final communities represent the optimal division of the network based on modularity.

The Louvain Method efficiently identifies community structures by iteratively refining the assignment of nodes to communities, yielding high modularity values.
Algorithm for Louvain Method/ Greedy Modularity is shown in Algorithm 1

Algorithm 1 Louvain Method (Greedy Modularity)

```
procedure LouvainMethod(Graph $G$)
    // Initialization
    Initialize each node as a single-node community
    Initialize modularity $Q = 0$
    repeat
        // Phase 1: Node Movement
        for each node $v$ in $G$ do
            for each neighbor $w$ of $v$ do
                // Calculate modularity gain
                $calculateModularityGain(v, w)$
            end for
        end for
        // Move nodes to maximize modularity
        $moveNodesToMaximizeModularity()$
        // Phase 2: Community Aggregation
        $createNewGraphWithCommunities()$
        // Update modularity
        $Q' = calculateModularityOfNewGraph()$
        // Check for modularity improvement
```

3.3.2 Advantages of Louvain Method/Greedy Modularity

- **Modularity Optimization:** Maximizes modularity score for strong community division.

- **Attribute Consideration:** Extends to attributed networks, incorporating node attributes in modularity optimization.

- **Community Quality:** Focuses on high-quality communities with similar nodes.

- **Scalability:** Efficient for large networks.

- **Adaptability:** Suited for dynamic networks, capturing evolving community structures.

- **Widespread Adoption:** Popular choice across various fields for effective community detection.

- **Flexibility and Adaptability:** Flexible and can be applied to a variety of network types, including social networks, biological networks, citation networks, and more.

Algorithm 1 Louvain Method (Greedy Modularity)

```
if $Q' > Q$ then
    $Q = Q'$
else
    break // No further improvement possible
end if
no more improvement is possible
// Output the final communities
return finalCommunities

procedure calculateModularityGain(Node $v$, Community $C$)
    // Calculate the change in modularity
    $\Delta Q = (\Delta Q removing v from its current community) + (\Delta Q adding v to community C)$
end procedure
procedure moveNodesToMaximizeModularity
    // Greedily move nodes to maximize modularity gain
    for each node $v$ in $G$ do
        move $v$ to the community that maximizes modularity gain
    end for
end procedure
procedure createNewGraphWithCommunities
    // Construct a new graph with communities as nodes
    for each community $C$ in current partition do
        create a supernode for $C$
    end for
    for each edge $(u, v)$ in $G$ do
        if communities of $u$ and $v$ are different then
            add an edge between the corresponding supernodes
        end if
    end for
end procedure
procedure calculateModularityOfNewGraph
    // Calculate modularity of the new graph
    $Q' = \sum$ over all communities $C$: $(fraction of edges within C - (degree of C/(2 \times total number of edges))^2)$
    return $Q'$
end procedure=0
```

4 Experimental Results

4.1 Experimental Setup

4.1.1 Software

Programming Language:

- **Python (version 3.x):** A versatile language with a rich ecosystem of libraries, including NetworkX for graph processing.

Python Libraries:

- **NetworkX:** A Python library for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks.

Visualisation:

- **Matplotlib and Seaborn:** Python libraries for creating static, interactive, and aesthetically pleasing visualisations.

- **Gephi (Optional):** A powerful graph visualization tool that can be used for exploring and visualizing the network.

Integrated Development Environment (IDE):

- **JupyterLab:** An interactive development environment for Jupyter Notebooks, allowing for easy code development and visualisation.

4.1.2 Hardware

Computational Resources

For Smaller Networks:

- Standard Laptop or Desktop:

- Processor (CPU): Quad-core or higher for better performance.

- Memory (RAM): 8 GB or higher for efficient processing.

- Storage: Sufficient local storage for the dataset and analysis.

For Larger Networks:

Cloud Platforms (Optional):

- AWS, Google Cloud, or Azure.

- Utilize virtual machines (VMs) with appropriate specifications based on the size of the network.
- Consider instances with multiple vCPUs and sufficient RAM.

Memory (RAM):

- Ensure sufficient RAM for efficient processing.
- Smaller Networks: 8 GB or higher.
- Larger Networks: 16 GB or more depending on the network size.

Processor (CPU):

- Choose a multi-core processor for parallel processing:
- Smaller Networks: Quad-core or higher.
- Larger Networks: Consider instances with multiple vCPUs.

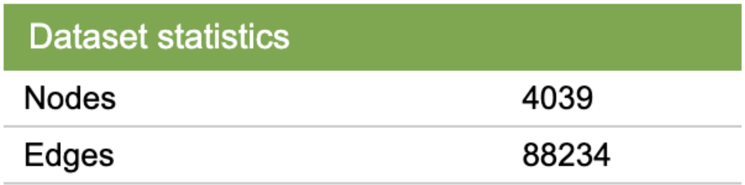
Storage:

- Adequate storage space for storing the dataset and any intermediate results:
- Smaller Networks: Standard HDD or SSD with sufficient capacity.
- Larger Networks: SSDs are recommended for improved read and write speeds.

4.2 Experiment 1

4.2.1 Dataset-1(Facebook Dataset)

This dataset consists of friends lists from Facebook. Facebook data was collected from survey participants using this Facebook app. The dataset includes nodes as profiles and edges as their connection.



Dataset statistics	
Nodes	4039
Edges	88234

Figure 2: *Dataset 1*

Each node in the dataset represents a Facebook profile or user account. Nodes could include individuals who participated in the survey or whose data was collected through a Facebook app. Edges between nodes represent connections or friendships between individuals on Facebook. If there is an edge between nodes A and B, it indicates that the user associated with node A is friends with the user associated with node B.

Visualisation of the Dataset

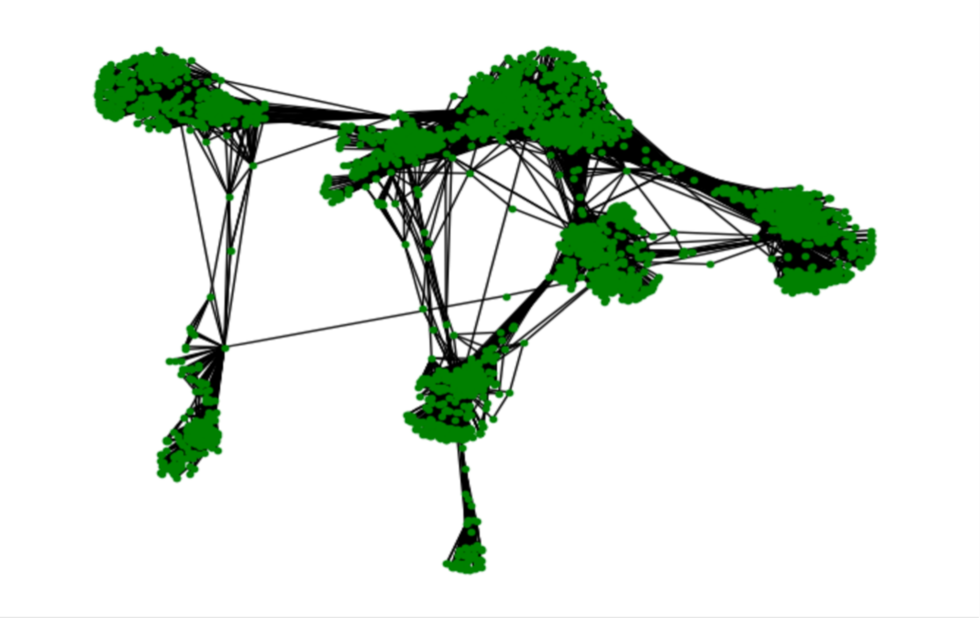


Figure 3: *Visualisation of Dataset 1*

Community

The communities formed using the Greedy Modularity Algorithm represent groups of nodes in the social network that exhibit a higher density of connections within the group compared to connections between groups. These communities capture the underlying structures of social interactions, identifying clusters of individuals with stronger ties and shared connections. The algorithm optimizes the modularity of the network partition, revealing meaningful patterns of community organization. Further analysis and exploration of these communities may provide insights into social dynamics, relationship patterns, and potential influential nodes within the network.

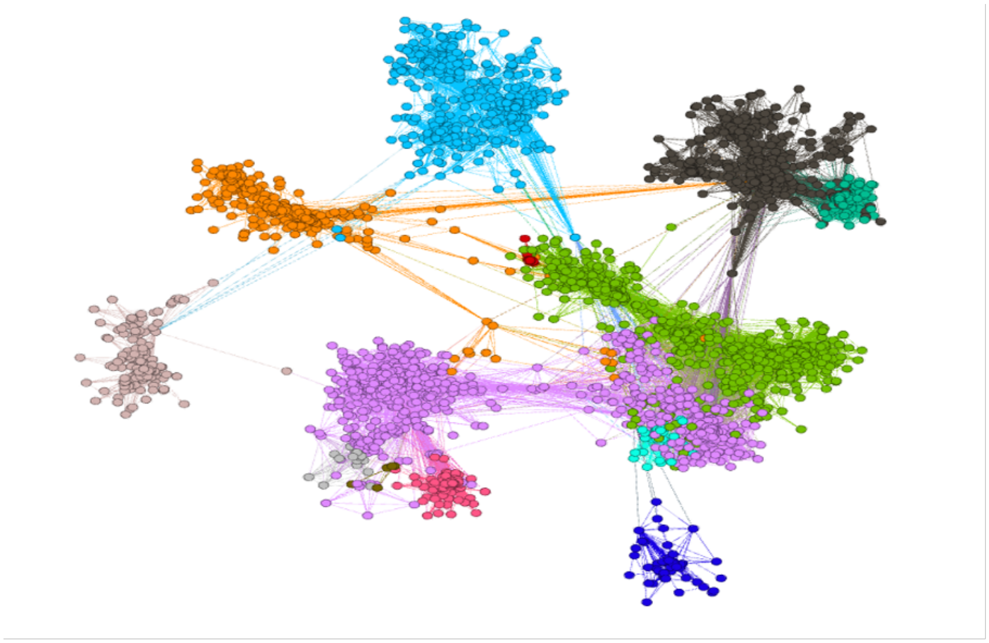
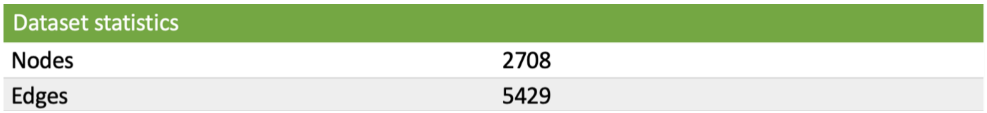


Figure 4: *Visualisation of communities - Dataset 1*

4.3 Experiment 2

4.3.1 Dataset-2 (Cora Dataset)

The Cora dataset is a citation network dataset representing scientific publications, where nodes correspond to documents and edges represent citations between documents.



Dataset statistics	
Nodes	2708
Edges	5429

Figure 5: *Dataset 2*

The Cora dataset comprises 2708 machine learning papers classified into seven categories. Each paper is represented in the .content file with a unique ID, binary values indicating the presence of words from a 1433-word vocabulary, and a class label. The .cites file details the citation graph, specifying links between papers with the direction from citing paper to cited paper. The papers were selected to ensure mutual citations, resulting in a vocabulary after preprocessing, including stemming and stopword removal, with 1433 words, and words with a document frequency below 10 were removed.

Visualisation of the Dataset

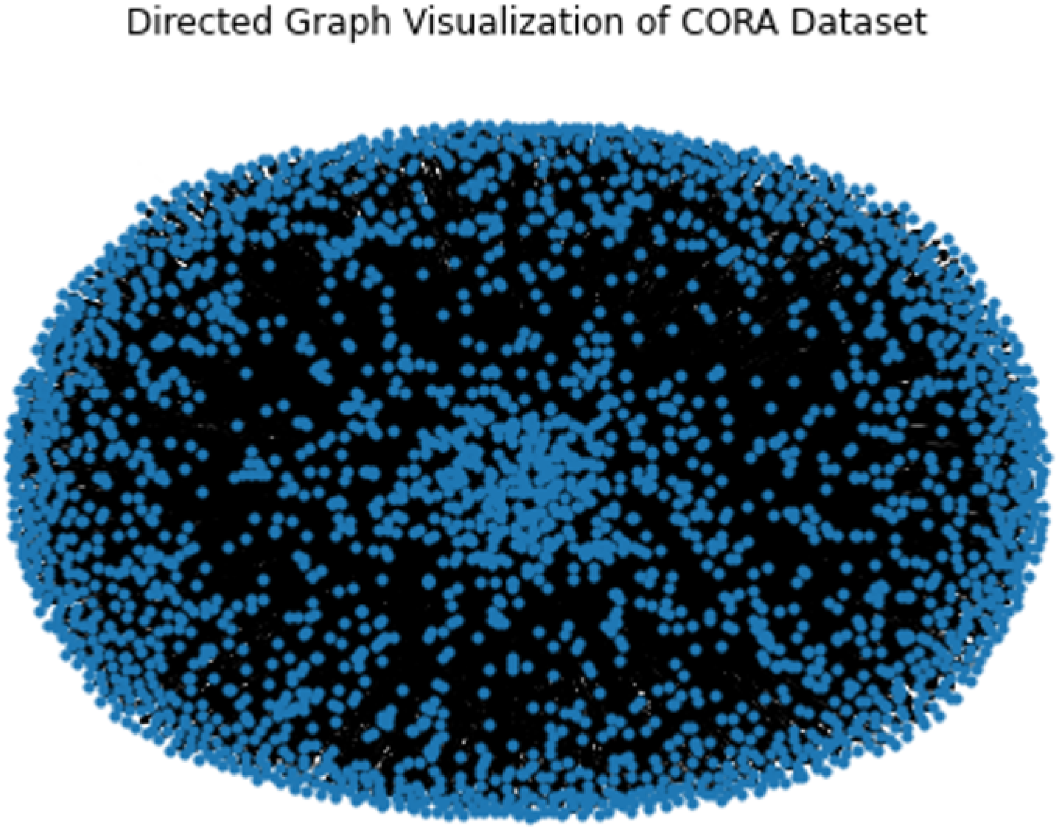


Figure 6: *Visualisation of Dataset 2*

yEd is a popular desktop application used for creating diagrams. It is a powerful diagramming tool that allows users to create a wide variety of diagrams, including flowcharts, network diagrams, organizational charts, mind maps, and more. yEd is developed by yWorks, a company specializing in software tools and services for diagramming and graph visualization.

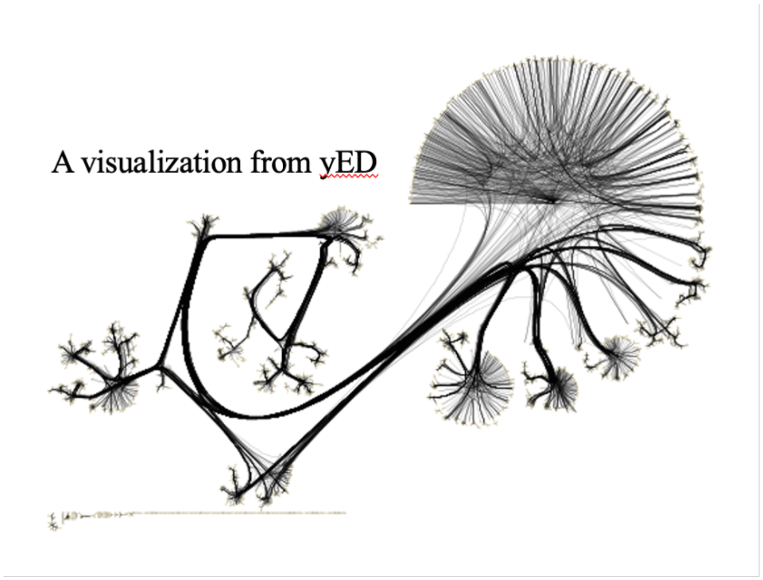


Figure 7: *Visualisation of Dataset 2 From yEd*

Community

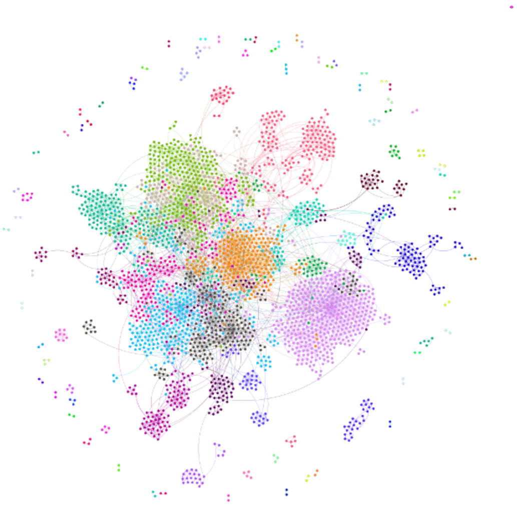


Figure 8: *Visualisation of communities - Dataset 2*

The algorithm is designed to optimize modularity, a measure that quantifies the quality of the division of a network into communities. Modularity reflects the extent to which the network can be divided into densely connected communities with sparser connections between them. Integration of attribute information enhanced the algorithm's ability to identify cohesive and meaningful communities.

5 Conclusions

This paper presented a comprehensive exploration into the realm of community detection in attributed networks, focusing on the development and application of a novel "Leader-Based Community Detection Algorithm." The motivation behind this work stemmed from the contemporary necessity to identify central leaders and understand community structures within complex networks enriched with attributes. Our approach combined advanced data analysis techniques and community detection algorithms tailored for attributed networks. The project commenced with an insightful literature survey, establishing the background and discussing existing methods, including Leader-Aware Community Detection, Leader Similarity-Based Community Detection, and other notable approaches. The proposed methodology comprised two key algorithms: the Leader Selection Algorithm, utilizing Eigenvector centrality and attribute information, and the Community Detection Algorithm, employing the Louvain Method/Greedy Modularity.

5.1 Findings and Contributions

Experimental results on two distinct datasets, namely the Facebook and Cora datasets, demonstrated the effectiveness of the proposed algorithm in uncovering meaningful communities within attributed networks. The visualizations showcased clear community structures, offering valuable insights into social dynamics and scientific collaboration patterns. The Louvain Method/Greedy Modularity algorithm, known for its modularity optimization, attribute consideration, scalability, and adaptability, proved to be a robust choice for community detection in attributed networks.

5.2 Future Scope

While the current work marks a significant stride in community detection, avenues for future research abound. The algorithm's efficiency can be further enhanced through parallelization, distributed computing, or sampling approaches, particularly for larger networks. The robustness of the proposed algorithm under different perturbations, such as node removal or changes in attribute data, warrants exploration. Additionally, extending the algorithm to handle overlapping communities and evaluating its performance on dynamic networks would contribute to its real-world applicability.

In conclusion, the Leader-Based Community Detection Algorithm presented in this paper provides a promising framework for understanding complex networks with attributes. The contributions and findings of this work open new vistas for research in the dynamic and evolving field of community detection, paving the way for informed decision-making and resource optimization.

6 References

1. A. Kesarwani, A. Singh, K. Gaurav and A. K. Shankhwar, "Leader Similarity Based Community Detection Approach for Social Networks," 2020 IEEE International Conference for Innovation in Technology (INOCON), Bangluru, India, 2020, pp. 1-6, doi: 10.1109/INOCO

2. D. -D. Lu, "Leader-Based Community Detection Algorithm in Attributed Networks," in IEEE Access, vol. 9, pp. 119666-119674, 2021, doi: 10.1109/ACCESS.2021.3109124.

3. Sun, Heli Du, Hongxia Huang, Jianbin Li, Yang Sun, Zhongbin He, Liang Jia, Xiaolin Zhao, Zhongmeng. (2020). Leader-aware community detection in complex networks. Knowledge and Information Systems. 62. 10.1007/s10115-019-01362-1.

4. Ahajjam, Sara Mohamed, El Haddad Hassan, Badir. (2018). A new scalable leader-community detection approach for community detection in social networks. Social Networks. 54. 41-49. 10.1016/j.socnet.2017.11.004.

5. Yakoubi, Zied Kanawati, Rushed. (2014). LICOD: A Leader-driven algorithm for community detection in complex networks. Vietnam Journal of Computer Science. 1. 241-256. 10.1007/s40595-014-0025-6.

6. Shah, Devavrat Zaman, Tauhid. (2010). Community Detection in Networks: The Leader-Follower Algorithm.