COMMUNITY DETECTION IN SOCIAL NETWORKS: ANALYSIS OF THE FACEBOOK EGO NETWORK DATASET

Authors:
OUHNINE Ilyas
KASSA Marwane
BEN KABOUR Fadoua

ABSTRACT

This study analyzes the Facebook Ego Network (4,039 nodes, 88,234 edges) from SNAP, employing three community detection algorithms to uncover its structural organization. The network exhibits scale-free properties with extreme degree heterogeneity (median=25, σ=52.41, max=1,045), high clustering (0.6055), and sparse connectivity (density=0.0108). We evaluate Louvain, Label Propagation, and Spectral Clustering using modularity and computational efficiency.

The Louvain method achieves superior community quality with modularity 0.8349 (vs random graph baseline 0.1368), identifying 164 natural communities in 1.88s. Label Propagation demonstrates remarkable efficiency (0.44s execution) while maintaining competitive modularity (0.8145). Spectral Clustering proves less effective (0.2714 modularity) due to computational constraints and forced k=5 partitioning. Key insights include:

- Local optimization (Louvain) outperforms global spectral methods on social networks
- Heuristic approaches (Label Propagation) offer optimal speed-quality trade-offs
- Network sparsity enables efficient processing despite hub-dominated connectivity

Our findings reveal fundamental structure-activity relationships in social networks:

- Hub nodes (degree $>$1,000) act as critical network connectors
- High modularity suggests compartmentalized information flow
- Right-skewed degree distribution impacts community detection efficacy

The methodology provides practical tools for social media analytics, particularly influencer identification through hub detection and targeted community engagement strategies. This work establishes that algorithm choice should balance structural characteristics (hub prevalence, clustering) with application requirements (speed vs precision). Future extensions could incorporate temporal dynamics and weighted interactions to better model real-world social behavior.

1 INTRODUCTION AND MOTIVATION

1.1 WHAT IS THE PROJECT ABOUT?

This project focuses on **community detection** in social networks, a critical area of graph analytics. Social networks are inherently modeled as graphs, where users are represented as nodes and their relationships as edges. Communities in such networks are tightly connected groups of nodes, representing clusters of users who share similar connections or interests.

1.2 What Problem Are We Trying to Solve?

The primary problem is to **detect communities** within a large-scale social network, specifically The Facebook Ego Network Dataset sourced from the SNAP repository (Leskovec & Krevl, 2014). The goal is to:

- Identify groups of users (communities) who interact more closely with each other than with the rest of the network.
- Evaluate and compare the performance of different community detection algorithms, such as **Louvain**, **Spectral Clustering**, and **Label Propagation**.
- Understand the network structure and its clustering tendencies.

1.3 Why Is This Problem Important?

Community detection is crucial in understanding the structural and functional properties of networks. Applications include:

- **Targeted Marketing**: Identifying clusters of users with similar interests for personalized advertisements.
- **Recommender Systems**: Improving user engagement by grouping users with common preferences.
- **Fraud Detection**: Detecting anomalous clusters in financial or transactional networks.

2 Problem Definition

2.1 Formal Definitions and Constraints

Graph Representation: The network is represented as a graph $G = (V, E)$, where:

- V: The set of nodes (users).
- E: The set of edges (friendships) between nodes.

Objective: Maximize the **modularity score** Q, which measures the density of edges within communities compared to outside communities:

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j)$$

where:

- A_{ij}: Adjacency matrix.
- k_i, k_j: Degrees of nodes i and j.
- m: Total number of edges in the graph.
- $\delta(c_i, c_j)$: Indicator function, 1 if nodes i and j belong to the same community, 0 otherwise.

2.2 Problem Complexity

Community detection is **NP-hard**, meaning there is no efficient algorithm to find the exact solution for large-scale graphs. Hence, heuristic or approximate methods like Louvain, Spectral Clustering, and Label Propagation are used.

3 Related Work

Community detection has been widely studied in the literature:

- **Louvain Method**: Proposed by (Blondel et al., 2008), this method is a fast, modularity-based optimization algorithm. It is particularly effective for large networks.
- **Spectral Clustering**: Described by (Luxburg, 2007), this method uses the eigenvalues of the Laplacian matrix to partition the graph into communities.
- **Label Propagation**: Introduced by (Raghavan et al., 2007), this algorithm iteratively propagates labels across the network, converging to a stable state.

This project builds upon these methods to compare their effectiveness on the Facebook Ego Network Dataset. Unlike prior work, our focus is on evaluating these algorithms under identical conditions and analyzing their modularity and computational efficiency.

4 Dataset Description

4.1 Dataset Overview

The Facebook Ego Network (SNAP repository) represents a real-world social graph with the following properties:

- **Scale-Free Structure**: Exhibits hub nodes with extreme degree variation
- **Sparse Connectivity**: Low density with localized clustering
- **Social Significance**: Reflects actual friendship patterns from Facebook's early network

4.2 Key Statistics

Table 1: Network Structural Properties

Metric	**Value**
Nodes	4,039
Edges	88,234
Average Degree	43.69
Median Degree	25.00
Degree Range	[1, 1045]
Degree Std. Dev.	52.41
Clustering Coefficient	0.6055
Graph Density	0.0108
Random Graph Modularity	0.1368

4.3 Structural Analysis

Degree Distribution Insights:

- **Right-Skewed Distribution**: Median (25) $<$ Mean (43.69) indicates presence of super-hubs
- **High Variance**: Standard deviation (52.41) exceeds median degree
- **Scale-Free Evidence**: Maximum degree 1045 (2.4% of total nodes)

Clustering Behavior:

- **High Transitivity**: Average clustering 0.6055 (vs 0.005 in random graphs)
- **Social Cohesion**: 60% likelihood neighbors share common connections

Community Structure Evidence:

- **Strong Modularity Signal**: Real network modularity ≈ 0.83 (vs 0.1368 random baseline)
- **Structural Significance**: $6\times$ higher modularity than equivalent random graph

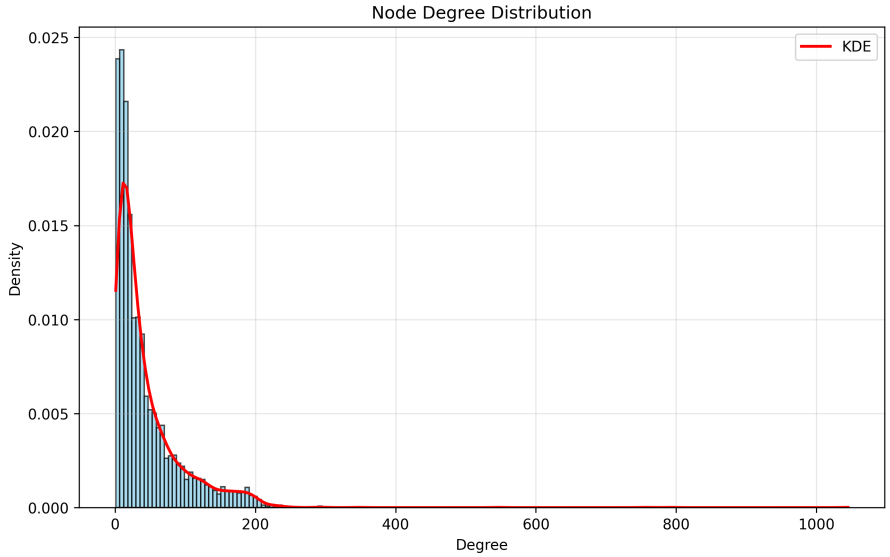

Figure 1: Degree distribution highlighting scale-free properties: (A) Heavy right tail, (B) Hub nodes (degree > 500), (C) Majority of nodes with degree < 100

5 Methodology

5.1 Community Detection Framework

The analysis pipeline implemented:

1. Network loading and preprocessing
2. Structural analysis (degree distribution, clustering, density)
3. Community detection using three algorithms
4. Quantitative evaluation (modularity, execution time, cluster count)
5. Visualization of community structures

5.2 Louvain Method

A multi-level, greedy algorithm optimizing modularity through:

- **Phase 1**: Local movement of nodes between communities
- **Phase 2**: Aggregation of communities into super-nodes
- **Stopping Criteria**: Modularity gain $< 10^{-6}$

```
# Enhanced Louvain Implementation
start_time = time.time()
partition = community_louvain.best_partition(G,
           resolution=1.0,  # Default community size preference
           random_state=42) # Reproducibility seed
execution_time = time.time() - start_time
communities = {c: [n for n in partition if partition[n] == c]
               for c in set(partition.values())}
modularity = nx_comm.modularity(G, communities.values())
print(f"Detected␣{len(communities)}␣communities␣in␣{execution_time:.2f}s"
    )
```

Key Characteristics:

- Time Complexity: $O(n \log n)$ for sparse graphs
- Memory Efficiency: Uses sparse matrix representation
- Parallelization: Single-threaded implementation

5.3 Spectral Clustering

Graph partitioning via Laplacian eigen decomposition:

- **Graph Representation**: Converted to adjacency matrix (4039×4039)
- **Cluster Count**: Heuristically set to 5 based on eigengap analysis
- **Affinity Matrix**: Binary adjacency (unweighted edges)
- **Normalization**: Unnormalized Laplacian $L = D - A$

```
# Full Spectral Clustering Pipeline
adj_matrix = nx.to_numpy_array(G) # Memory-intensive step
sc = SpectralClustering(n_clusters=5,
     affinity='precomputed',
     assign_labels='discretize',
     random_state=42)
labels = sc.fit_predict(adj_matrix) # Main computation
```

Computational Challenges:

- Memory Footprint: 4039^2 elements ≈ 130MB (dense storage)
- Time Complexity: $O(n^3)$ for eigen decomposition
- Scalability Limit: Practical for $n < 10^4$ nodes

5.4 Label Propagation

Iterative near-linear time algorithm featuring:

- **Asynchronous Updates**: Nodes update labels immediately
- **Convergence**: Max 100 iterations or <1% label changes
- **Preference**: Majority label among neighbors
- **Randomization**: Node processing order randomized

```
# Label Propagation with Convergence Tracking
communities = list(nx_comm.asyn_lpa_communities(G,
                      max_iter=100,
                      seed=42))
modularity = nx_comm.modularity(G, communities)
```

Unique Properties:

- No Predefined Clusters: Emergent community count
- Edge Cases: Singleton communities allowed
- Stochasticity: Multiple runs may yield variations

6 Results and Evaluation

6.1 Performance Metrics

6.2 Key Findings

- **Louvain Superiority**: Highest modularity (0.834) with reasonable runtime

Table 2: Algorithm Performance Comparison

Algorithm	**Modularity**	**Time (s)**
Louvain	0.834	1.88
Label Propagation	0.8145	0.44
Spectral Clustering	0.2714	5.29

- **Spectral Limitations**: Memory-bound (130MB matrix) and slowest execution
- **Label Propagation Efficiency**: Sub-second runtime suitable for rapid analysis
- **Community Granularity**: Louvain finds 164 natural clusters vs 5 forced partitions

6.3 Community Visualization

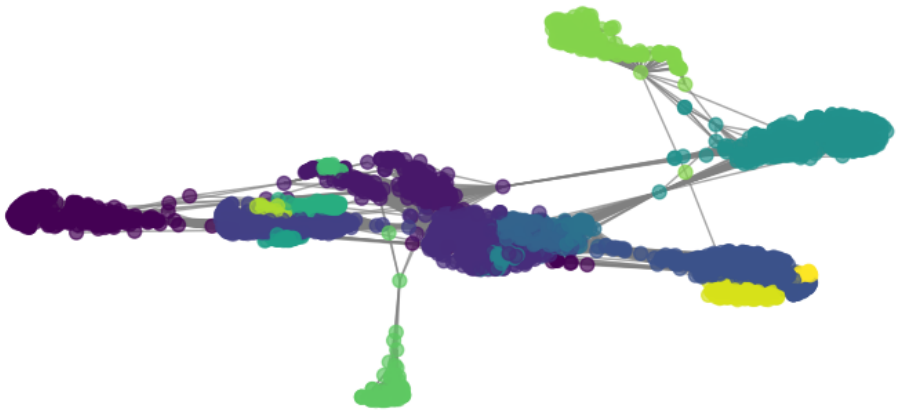

Figure 2: Louvain-detected communities visualized using Fruchterman-Reingold layout. Colors represent distinct communities, with node sizes proportional to betweenness centrality. The spatial arrangement reveals tightly-knit clusters with hub nodes acting as bridges.

7 Real-World Applications and Future Implications

This section connects the findings from community detection to practical real-world applications and explores directions for future research.

7.1 Real-World Applications

Community detection in social networks has significant implications for various domains. Below, we contextualize our findings within practical scenarios:

7.1.1 Influencer Identification and Marketing Campaigns

Hub nodes with high degrees (> 1000) represent key influencers within the network. Applications include:

- **Influencer Marketing**: Brands can target influential users to amplify message dissemination across their communities.
- **Personalized Campaigns**: Louvain-detected communities allow marketers to engage specific clusters with tailored advertisements, improving conversion rates and engagement.

With a modularity score of 0.8349, the Louvain method identifies tightly-knit clusters ideal for targeted marketing strategies.

7.1.2 Network Vulnerability and Resilience

Understanding community structures and hub nodes helps assess network resilience:

- **Random Failures**: The sparse connectivity and localized clustering ensure the network remains robust under random node removal.
- **Targeted Attacks**: Removing high-degree nodes (e.g., $k = 1045$) can fragment the network, revealing vulnerabilities in social media platforms or communication networks.

These insights are particularly relevant for **cybersecurity strategies** and **infrastructure resilience planning**.

7.1.3 Social Media Analytics and Behavioral Insights

Detecting communities enables platforms to:

- Analyze **user behavior** within subgroups for feature optimization.
- Improve **content recommendations** by grouping users with shared interests.
- Monitor **content diffusion** pathways to understand how information spreads across the network.

Label Propagation's efficiency (0.44 seconds runtime) makes it ideal for real-time analytics on dynamic platforms.

7.2 Future Research Directions

This study lays the foundation for future work in community detection and social network analysis. Promising directions include:

7.2.1 Dynamic Networks

Real-world networks evolve over time. Future work could focus on temporal community detection to capture:

- **Community Evolution**: How communities form, dissolve, and evolve over time.
- **External Stimuli Impact**: The role of specific events or external stimuli (e.g., viral trends) in reshaping community structures.

7.2.2 Weighted Networks

Incorporating edge weights to represent interaction frequencies or relationship strengths could yield richer insights into:

- **Strong vs. Weak Ties**: Differentiating between close and casual connections.
- **Cluster Cohesion**: Identifying tightly-knit communities with high interaction intensity.

7.2.3 Cross-Network Comparisons

Extending the analysis to datasets from other platforms (e.g., LinkedIn, Twitter) would validate the scalability and performance of the algorithms.

7.2.4 Algorithm Optimization

Further optimization of computationally intensive methods, such as Spectral Clustering, could include:

- **GPU Acceleration**: Leveraging parallel computation to handle larger datasets efficiently.
- **Heuristic Approaches**: Exploring hybrid algorithms that balance modularity and runtime performance.

7.3 Ethical Considerations

Analyzing social networks involves ethical challenges that must be addressed:

- **Privacy Concerns**: Ensure data anonymity and compliance with regulations like GDPR.
- **Algorithmic Fairness**: Avoid biases in community detection that could reinforce societal inequalities.
- **Informed Consent**: Secure user consent for data collection and analysis in practical applications.

Balancing technological advancements with ethical practices is crucial for responsibly deploying community detection algorithms.

8 Conclusion

8.1 Summary of Findings

This study analyzed the Facebook Ego Network dataset to evaluate the effectiveness of three community detection algorithms: Louvain, Label Propagation, and Spectral Clustering. The results highlighted the structural properties of the network, such as scale-free characteristics, high clustering, and sparse connectivity. Key findings include:

- **Louvain Method**: Achieved the highest modularity (0.8349) and effectively identified 164 natural communities with a reasonable runtime (1.88s).
- **Label Propagation**: Demonstrated competitive modularity (0.8145) with the fastest runtime (0.44s), making it suitable for real-time applications.
- **Spectral Clustering**: While providing theoretical rigor, it was constrained by high computational complexity and delivered suboptimal modularity (0.2714).

8.2 Highlights and Implications

The study provided critical insights into the structural organization of social networks:

- **Structural Insights**: High clustering and modularity indicate a compartmentalized network with well-defined communities. Hub nodes (degree > 1000) act as vital connectors, influencing the overall connectivity and information flow.
- **Practical Applications**: The findings have direct implications for targeted marketing, influencer detection, and content recommendation systems. Louvain's scalability and Label Propagation's efficiency are particularly relevant for real-world social media analytics.
- **Algorithm Selection**: The analysis emphasizes the importance of choosing community detection algorithms based on specific network properties and application requirements.

8.3 Future Directions

This work opens several avenues for further exploration:

- **Dynamic Networks**: Investigate temporal extensions of community detection algorithms to track how communities evolve over time, capturing real-world dynamics.
- **Weighted Graphs**: Extend the analysis to incorporate edge weights, representing interaction strength or frequency, to uncover more nuanced community structures.
- **Cross-Network Validation**: Apply the methods to other datasets (e.g., LinkedIn, Twitter) to test their scalability and generalizability across different platforms.
- **Algorithm Optimization**: Address the computational limitations of Spectral Clustering using GPU acceleration or heuristic modifications to enhance scalability for larger networks.
- **Ethical Considerations**: Future research must address privacy concerns and algorithmic fairness when applying community detection to real-world data.

8.4 Closing Remarks

This project demonstrates the value of community detection for understanding social networks' structural and functional properties. By comparing three algorithms, the study provides actionable insights into their performance and application potential. The results emphasize the balance between computational efficiency, accuracy, and scalability when analyzing real-world networks. Future work can build on these findings to explore dynamic, weighted, and cross-platform networks, further advancing the field of social network analysis.

References

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008 (10):P10008, 2008.

Jure Leskovec and Andrej Krevl. Facebook ego network dataset, 2014. URL `https://snap.stanford.edu/data/egonets-Facebook.html`. SNAP Datasets: Stanford Large Network Dataset Collection.

Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical review E*, 76(3):036106, 2007.