Social Network and Friend Recommendation Algorithms Analysis

Project Proposal

Team Name: aad.js

---

1. Project Overview

1.1 Title

Social Network and Friend Recommendation Algorithms Analysis

1.2 Abstract

This project aims to perform comprehensive analysis on a social network dataset to understand network structure, identify communities, and analyse friend recommendation algorithms. The project will implement and compare multiple graph algorithms for network analysis and link prediction, evaluating their performance, scalability, and real-world applicability.

1.3 Dataset

Source: Stanford Network Analysis Project (SNAP) - Facebook Social Circles Dataset

- Undirected social graph
- Nodes represent users
- Edges represent friendships
- Have data on user personalities

---

2. Project Objectives

2.1 Primary Objectives

1. Analyze the structural properties of the social network
2. Identify and characterize communities within the network
3. Develop and evaluate friend recommendation algorithms
4. Compare traditional heuristic methods with modern embedding-based approaches

2.2 Secondary Objectives

1. Understand the distribution of user influence and importance
2. Evaluate scalability of different algorithms on large-scale data
3. Provide interpretable insights for recommendation decisions

---

3. Methodology

3.1 Network Structure Analysis

3.1.1 Connected Components Analysis

Identify and analyze connectivity patterns using:

- **Breadth-First Search (BFS):** Explore network layer by layer
- **Depth-First Search (DFS):** Explore network paths deeply
- **Union-Find Algorithm:** Efficiently track connected components
- Additional graph traversal algorithms

Deliverables:

- Number and size distribution of connected components
- Giant component analysis

- Network connectivity metrics

3.1.2 Centrality Analysis

Measure node importance using multiple centrality metrics:

- **Degree Centrality:** Number of direct connections
- **Betweenness Centrality:** Frequency of node appearing on shortest paths
- **Closeness Centrality:** Average distance to all other nodes
- **Eigenvector Centrality:** Influence based on connections to influential nodes

Deliverables:

- Top influential users identification
- Centrality distribution analysis
- Correlation analysis between different centrality measures

3.2 Community Detection

Implement and compare well-known community detection algorithms:

3.2.1 Algorithms to Implement

1. **Girvan-Newman Algorithm**
    - Edge betweenness-based divisive method
    - Iteratively removes edges with highest betweenness
2. **Louvain Modularity Algorithm**
    - Fast greedy optimization method
    - Maximizes network modularity
3. **Leiden Algorithm**
    - Improved version of Louvain
    - Guarantees well-connected communities

Deliverables:

- Community structure visualization
- Modularity scores comparison
- Community size distribution
- Runtime performance analysis

3.3 User Personality Modeling

3.3.1 Concept

Enhance recommendation realism by assigning personality tags to users.

3.3.2 Implementation

- Assign random tags to each user (e.g., "gamer", "foodie", "extrovert", "introvert", "traveler")
- Use tags to simulate user preferences and behavior
- Incorporate personality compatibility in recommendation evaluation

Deliverables:

- Tag distribution analysis
- Personality-based network clustering

3.4 Friend Recommendation System

3.4.1 Experimental Setup

- **Training Set:** Subset of the original graph
- **Test Set:** Hidden edges from the original graph
- **Simulation:** Accept/reject recommendations based on ground truth
- **Evaluation:** Calculate precision, recall, and other metrics

3.4.2 Algorithms to Implement and Compare

Heuristic-Based Methods:

1. **Common Neighbors:** Count of mutual friends
2. **Jaccard Coefficient:** Normalized common neighbors
3. **Adamic-Adar Index:** Weighted common neighbors by rarity
4. **Preferential Attachment:** Product of node degrees
5. **Resource Allocation:** Resource flow through common neighbors

Embedding-Based Methods: 6. **Node2Vec:** Random walk-based graph embeddings
7. **DeepWalk:** Deep learning approach for node representations

3.4.3 Evaluation Metrics

Performance Metrics:

- **Precision:** Proportion of correct recommendations
- **Recall:** Proportion of actual friends identified
- **F1 Score:** Harmonic mean of precision and recall
- **ROC-AUC:** Area under ROC curve for binary link prediction
- **Mean Average Precision (MAP):** Ranking quality metric

Operational Metrics:

- **Runtime Analysis:** Execution time per algorithm
- **Scalability:** Performance on varying dataset sizes
- **Memory Usage:** Resource requirements

Qualitative Metrics:

- **Interpretability:** Ease of understanding recommendations
- **Explainability:** Ability to justify recommendations

---

4. Expected Outcomes

4.1 Analytical Insights

- Comprehensive understanding of social network structure
- Identification of key influencers and communities
- Community characteristics and interaction patterns

4.2 Comparative Study

- Performance comparison of recommendation algorithms
- Trade-offs between heuristic and embedding-based methods
- Scalability analysis for production deployment

4.3 Practical Recommendations

- Best algorithm recommendations for different scenarios
- Guidelines for implementing friend recommendation systems
- Insights into factors affecting recommendation quality

7. Deliverables

1. **Python code or Jupyter Notebooks:** Documented code for all analyses
2. **Final Report:** Comprehensive analysis with visualizations
3. **Presentation:** Key findings and recommendations
4. **Recommendation System:** Functional prototype with evaluation results

---

8. References

- SNAP Dataset: http://snap.stanford.edu/data/
- Research papers on community detection and link prediction algorithms

---

9. Notes

- Deliverables and algorithms to be analysed may change a bit over the course of project
- Used Generative AI to *refine* this proposal