Connecting Dots

An Algorithmic Approach to Friend Recommendation Systems

Course Details: Data Structures and algorithms (CSL 2020)

Course Instructor – Dr.Suchetana Chakraborty

IIT JODHPUR 2024

Project agenda

- Meet the Team
- Project Ideation
- Problem Statement
- Conclusion
- Current Status
- Acknowledgements

meet the team

Mentors

Garvit Chugh
(D20CS051)

Mentors

Pranav Goswami
(B20CS016)

Aditya Apurva
(B22ME003)

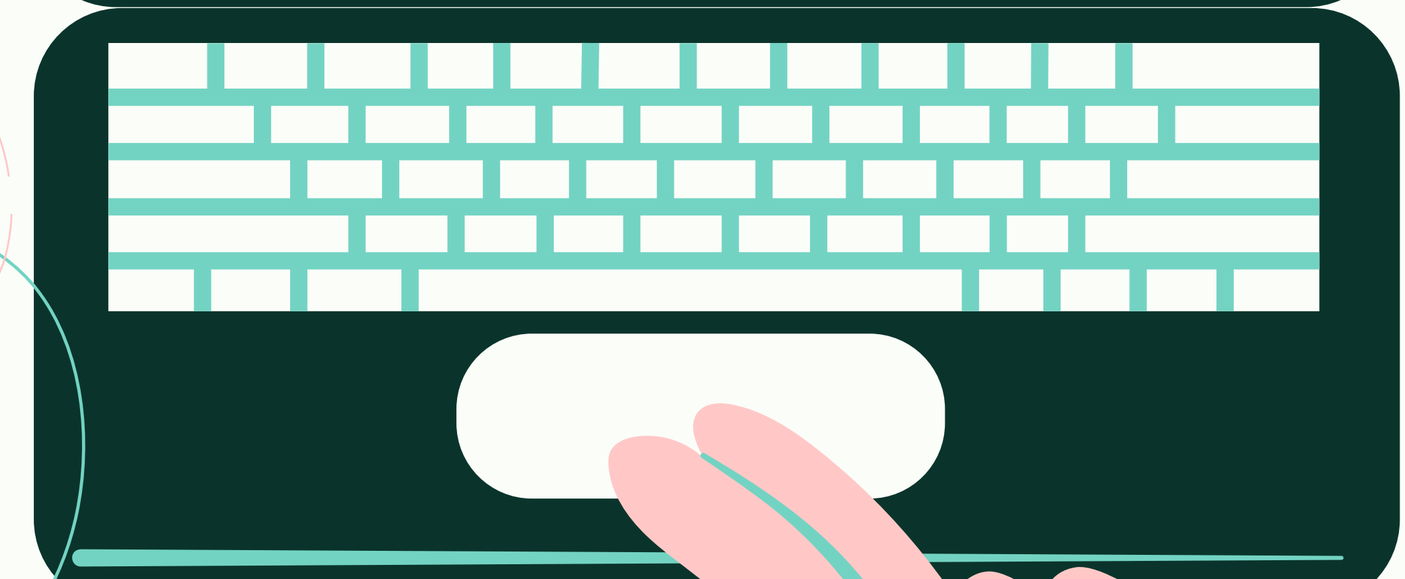
Om Singh
(B22ME044)

Avichal Sinha
(B22CH004)

Deekshant Singh
(B22CI010)

PROBLEM STATEMENT

Navigating social networks can be overwhelming due to the vast number of users and their complex interconnections. Our project develops a Friend Recommendation System using data structures and algorithms to simplify this process. It analyzes a user's network to suggest potential friends, enhancing the discovery of new, relevant connections.



Let’s Break Down The Problem





Domain Mapping

The domain of this project is social networking.It uses a graph data structure where each node represents a user and an edge between two nodes indicates a friendship. The system recommends friends to a user based on their existing connections.

Problem Relevance

It helps users discover new connections, thereby enhancing their networking experience. By suggesting potential friends, the system can help users build a more robust and meaningful network.

Challenge

Friend recommendation is challenging because it requires analyzing complex and large-scale social network data.Moreover, the recommendations need to be personalized and relevant to each user.

Data-Driven Promise

By analyzing a user’s existing connections and their connections’ connections, the system can identify potential friends that the user is likely to know or have shared interests with. This approach is more effective and scalable than manual or random recommendations.

Improvements Suggested

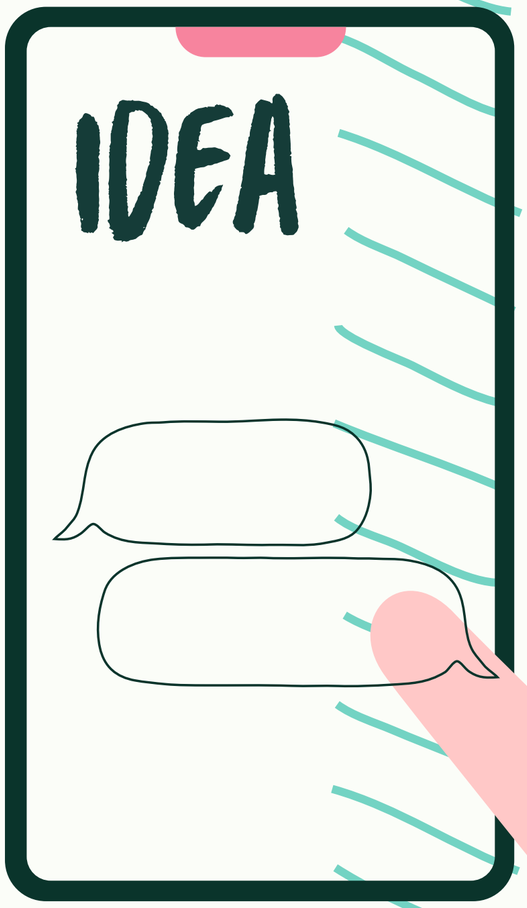
The adjacency matrix is a powerful data structure used to represent the graph, making it easy to identify direct friends (first-degree connections) and friends of friends (second-degree connections). The system uses efficient algorithms to traverse this graph and identify potential friends based on these connections.

GitHub Repository Link

The repository includes all the source code files, along with a README file that provides a detailed explanation of the project and instructions on how to run the code.

Highlighting Results

The friend recommendation system has shown promising results in preliminary tests. It successfully identifies potential friends based on existing connections. This leads to more meaningful and relevant friend suggestions, enhancing the user's experience on the social networking platform.



Cost Analysis

The primary cost associated with this system is computational. The system needs to analyze the adjacency matrix, which can be large for a social network with many users. However, the use of efficient data structures and algorithms helps to manage this cost.

Cost-Benefit Analysis

By making it easier for users to discover new connections, the system enhances user engagement on the social networking platform. This can lead to increased user activity and retention, which are valuable for the success and growth of the platform. Therefore, the benefits of implementing this system outweigh the computational costs.

CURRENT Status

Data Structures:

- Investigated the use of various data structures such as matrices, arrays, and graphs to represent social connections.
- Observed the adoption of advanced data structures like adjacency lists and hash tables for optimizing recommendation algorithms.

Machine Learning Techniques:

- Identified a trend towards integrating machine learning techniques for personalized friend recommendations.
- Noted the use of collaborative filtering, matrix factorization, and deep learning models to enhance recommendation accuracy.

Algorithmic Advances:

- Discovered novel algorithms for analyzing social networks and generating friend recommendations.
- Found research papers proposing efficient graph-based algorithms to identify potential connections.

FRIEND-RECOMMENDATION SYSTEM

Algorithmic Scalability:

- Some algorithms exhibit poor scalability when applied to massive social networks with millions of users and connections.
- Lack of optimization techniques to address scalability issues and ensure real-time recommendation generation.

Data Structure Complexity:

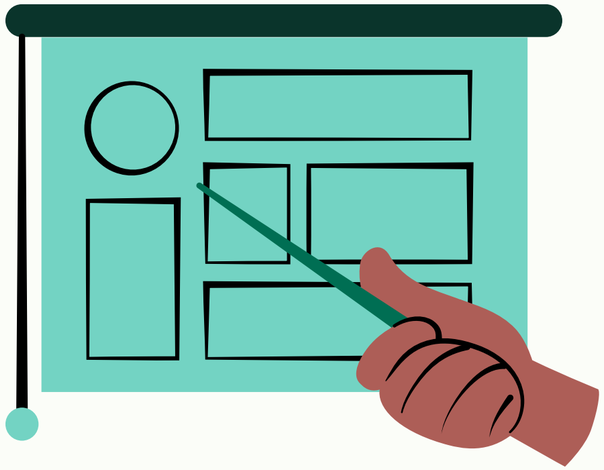
- Many existing approaches rely heavily on complex data structures, leading to increased memory and processing requirements.
- Limited scalability due to the inefficiency of data structures in handling large-scale social networks.

Limitations of Existing Approaches

Project Conclusions

Key Findings

1. Project provided valuable insights into the practical application of data structures and algorithms.
2. Using an adjacency matrix to represent a social network graph proved to be effective to map and analyze user connections.
3. System successfully identifies potential friends based on existing connections, demonstrating the power of algorithmic analysis in making personalized recommendations.



Future Scope

1. Potential to enhance recommendations by considering additional factors like shared interests, activities, or locations.
2. Scope to visualize the social network graph using tools like Python for intuitive understanding and further analysis.

Innovation Summary

1. Application of fundamental data structures and algorithms to a real-world problem.
2. Enhancement of user experience through personalized recommendations.
3. Designed to handle large-scale social network data, demonstrating scalability.

Acknowledgements

1. Personality-Based Friends Recommendation System for Social Network
2. Friend Recommendations using GraphSAGE
3. Challenges & Limitation in Recommender Systems
4. Friendship Recommendation System Using Topological Structure of Social Networks

Contribution

Task and Responsibilities

OM SINGH

- **Responsible for writing the main C code.**
- **Handles user input, adjacency matrix creation, friend recommendation logic, and stack implementation.**
- **Writes the main function (main()), adjacency_count, adjacency_result, push, isPresent, Display_stack functions, and the write_to_csv function.**
- **Ensures the C code runs smoothly and handles user interactions effectively.**

ADITYA

- Responsible for writing the Python code.
- Creates the Python script to visualize the graph using NetworkX and Matplotlib.
- Reads the adjacency matrix from the CSV file, constructs a directed graph, and visualizes it.
- Writes the Python script to load the adjacency matrix, create a directed graph, and draw it.
- Assists in writing the C code.

AVICHAL

- **Assists in writing the C code.**
- **Reviews the C code to ensure correctness and efficiency.**
- **Conducts thorough testing of the C code**
- **Ensures that the C code follows best practices and is well-documented**
- **Responsible for creating the PowerPoint presentation.**
- **Assists with documenting the codebase, including comments, documentation strings, and README files, to facilitate understanding and usage by other team members or external contributors.**

DEEKSHANT

- **Creates the Python script to visualize the graph using NetworkX and Matplotlib.**
- **Responsible for creating the PowerPoint presentation.**
- **Incorporates visuals, diagrams, and code snippets into the presentation slides.**
- **Designs and implements the user interface (UI) for the project.**
- **Ensures that the UI enhances the user experience and facilitates interaction with the application.**

THANK YOU

for this project!