

Social Network Analysis System

Ever wondered who the most influential people in a social network are? Or how information spreads through friend groups? This C++ project helps you answer these questions and more by analyzing social networks from multiple angles.

What Does This Do?

Think of this as a detective toolkit for understanding social networks. We've built four main tools that work together:

- **Finding Key Influencers** - Who are the "bridge" people connecting different groups?
- **Recommending Friends** - Who should connect based on mutual friends and shared interests?
- **Tracking Information Spread** - How does news or content spread through the network?
- **Combining Insights** - What happens when we look at all these factors together?

The Cool Stuff We Built

Finding the Network Bridges

Using an algorithm called Brandes' method, we can quickly find people who act as bridges between different social groups. These are the folks who, if they left, would make it harder for information to flow between communities. Think of them as the social glue holding different groups together.

Predicting How Things Go Viral

We simulate how information spreads using something called the Independent Cascade Model. Imagine dropping a pebble in water and watching the ripples spread - that's similar to how we model information spreading through friend networks. We can even figure out which people you'd want to "seed" with information to reach the most people.

Smart Friend Suggestions

We use two different approaches:

Jaccard Coefficient - This is like saying "you and this person have 5 mutual friends out of 10 total friends between you, so maybe you should connect!"

Adamic-Adar Index - This one's more sophisticated. It says that having a mutual friend who doesn't know many people is more meaningful than sharing a friend who knows everyone. Quality over quantity!

The Big Picture View

Sometimes one metric isn't enough. Our hybrid analysis combines everything to answer questions like "which new friendship would boost someone's influence the most?" or "how would connecting these two people change

information flow?"

Built to Last

We've tested everything thoroughly using Google Test, and there's a friendly menu system so you don't need to be a command-line wizard to use it.

Who Built This?

This was a team effort! Here's who worked on what:

- **Vijna Maradithaya** tackled the challenging Betweenness Centrality implementation
- **Anhad Singh** built the Jaccard-based friend recommendations and wrote all our tests
- **Raghav** implemented the Influence Maximization algorithms
- **Randip** created the Adamic-Adar recommendation system

Getting It Running

What You'll Need

Before you start, make sure you have:

- Git (to download the code)
- CMake version 3.14 or newer (this helps build the project)
- A modern C++ compiler that supports C++17 (like g++ or clang++)

Let's Get Started

Step 1: Grab the code

```
bash
git clone https://github.com/RaghavMaheshwari2024/Social-Network-Analysis.git
cd Social-Network-Analysis
```

Step 2: Set up your build folder

```
bash
mkdir build
cd build
```

Step 3: Build everything

```
bash  
  
cmake ..  
make
```

Step 4: Run the program

```
bash  
  
cd src  
./sna
```

The program will look for a file called `0.edges` in the main project folder - that's your network data.

Step 5: Make sure everything works

```
bash  
  
cd build  
./runTests
```

This runs all our tests to verify everything's working correctly.

The Science Behind It

Our betweenness centrality feature is based on a clever algorithm by Ulrik Brandes from 2001. He figured out how to calculate this metric way faster than previous methods - going from $O(N^3)$ complexity down to $O(NM)$. That's a huge deal when you're analyzing large networks!


If you're curious about the math, check out the original paper: [A Faster Algorithm for Betweenness Centrality](#).

How Everything's Organized

```
Social-Network-Analysis/  
├── src/          # All the code and the main program  
├── build/        # Where everything gets compiled (you create this)  
├── 0.edges       # Your network data goes here  
├── CMakeLists.txt # Build configuration  
└── README.md     # You're reading it!
```

Want to Help?

We'd love contributions! Found a bug? Have an idea for a new feature? Feel free to open an issue or submit a pull request on GitHub.

Built with  and lots of debugging by the team at [Social Network Analysis Project](#)