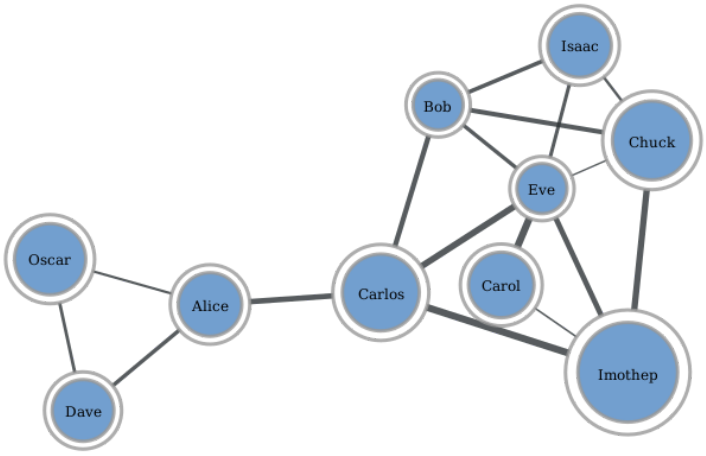
Project Proposal: Algorithms for Social Cluster Identification and Friend Recommendation

Jeremy Donhowe, APPM 3310-001
Carl Cortright, APPM 3310-002
Emily Volk, APPM 3310-001
Weibin Huang, APPM 3310-002



Visualization of a simple social graph. Photo Credit: graph-tool.skewed.de

Background

Since the inception of social networks, there has been significant interest in understanding how social clusters form and how we can use these clusters to better recommend friends to users. Companies like Facebook, Twitter, and Google have a significant stake in understanding friend networks in order to keep their users engaged; when users see relevant friend suggestions, they form tighter social clusters online, helping social networks form richer interactions between friends, encouraging their users to come back for more content (and ads use these platforms to make money). However, as a side note, social networks have recently experienced some criticism originating in these recommendation algorithms, namely that establishing social clusters causes homogeneity in a user's online relation. Better understanding clustering and recommendation algorithms is a useful, and timely, exploration.

Objective

To analyze social social media data using graphical layout algorithms to identify friend clusters and help friends find each other using intelligent friend recommendation.

Prior Work

There exists a significant body of work on the topic of friend recommendation for online platforms. Dave Troy has helped lay the foundation with his work on people maps (1). Researchers from Google have investigated how people naturally form implicit groups based on relationships like family, friends, and co-workers (2). Graph theory is also a common topic in linear algebra and discrete mathematics classes (3).

Techniques and Tools

In this project we plan on using a variety of graph algorithms and matrix-based analyses to identify clusters within social networks. We plan on using eigenvectors to calculate the most popular nodes in the network. By looking at our largest eigenvector we will be able to discern which dimension that the majority of its magnitude is, thus pointing to the most popular node. We plan to do this not only for people's immediate friends, but friends' friends as well. To calculate clusters within the network, we will calculate a local clustering coefficient for each vertex in the network (4), using these coefficients to identify the specific vertices that form tight clusters. To visualize this clustering we will use force directed algorithms to generate clustered visualizations contained in popular graph visualization tools like the Python graph-tool library (5).

After we identify the implicit clusters formed within the network, our plan is to use these clusters to predict relevant friend recommendations for each user. Within each cluster we will identify the vertices that have many common connections but are not yet connected. We will further enhance this recommendation by using a weighted inner product of each user's metadata, including their birthdate, college attended, and hobbies.

Dataset

For this project we will use an open source dataset from the Stanford Large Network Dataset Collection (6). Provided is a crowd-sourced dataset of over 4000 vertices and 88,000 edges from anonymous Facebook users. The dataset includes metadata for each user needed to predicted similarity in age, interests, and education.

References

(1) The Math Behind People Maps, Dave Troy, http://davetroy.com/posts/the-math-behind-peoplemaps

(2) Suggesting Friends Using the Implicit Social Graph, https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/36371.pdf

(3) Applied Linear Algebra by Peter J. Olver and Chehrzad Shakiban, Prentice Hall, 2006, ISBN 0-13-147382-4.

(4) Collective Dynamics of 'Small World' Networks, watts-collective, 1998 http://www.nature.com/nature/journal/v393/n6684/full/393440a0.html

(5) graph-tool, https://graph-tool.skewed.de/

(6) Stanford Large Network Dataset Collection, https://snap.stanford.edu/data/index.html