Project Goals

Source Location: Stanford Large Network Dataset Collection: Social Circles from Twitter

Format:

- nodeId.edges: The edges in the ego network for the node 'nodeId'. Edges are undirected for facebook, and directed (a follows b) for twitter and gplus. The 'ego' node does not appear, but it is assumed that they follow every node id that appears in this file.
- nodeId.circles: The set of circles for the ego node. Each line contains one circle, consisting of a series of node ids. The first entry in each line is the name of the circle.
- nodeId.feat: The features for each of the nodes that appears in the edge file.
- nodeId.egofeat: The features for the ego user.
- nodeId.featnames: The names of each of the feature dimensions. Features are '1' if the user has this property in their profile, and '0' otherwise. This file has been anonymized for facebook users, since the names of the features would reveal private data.

Project Idea

The foundational idea behind our project is to use the Twitter data from the Stanford Large Network Dataset Collection's social circles datasets in order to go through Twitter accounts and analyze which accounts in the set are most likely to follow one another based on a respective account's features. First, we would obtain the data regarding the users in the form of R Documentation (featnames), where each user would be represented by a node on the graph that is created, and each of the edges are represented by who a user follows. The graph in itself is directed, as one user following another is independent of whether the followed user follows back. Using this information, we would create a graph (again, users are nodes and edges are who is followed). The features of each user are stored in the nodes and the features will be represented as dummy variables (represented as 1 for having the feature and 0 for not having). Each of the edges will then be weighted in accordance to the set of features the node (user) has. In order to traverse the graph, We will use a **Breadth First Search (BFS Traversal) algorithm**, as this method allows us to go through each of the users, or nodes, and collect the needed feature data about them. However, we would then use **Dijkstra's algorithm** to find the nearest neighbor of the current node, despite its worse time space complexities due to its ability to be applied to weighted edges. Afterwards, we will implement a variant of the **PageRank algorithm**. Utilizing this algorithm, the last step will be to find the most likely followers based on common features and the shared edges.