P2:Social AnalyticsUsingGraphDatabases

RobertDates

September 25, 2016

1 Introduction

This assignment introduced me to the basic formal approaches to Social Network Analysis. We first had to analize data and make conclusion without the aide any data visualization software. We then had to learned how graph database software is very useful in organizing and analyzing datasets. Specifically I focused on the neo4j graph database, and was able to find various social network statistics such as node and edge count, clustering coefficients, and betweenness centrality.

2 Objective

We needed to find the statistics as well as prove or disprove two hypotheses that were made on the social network datasets provided.

3 Plan of Action

I used python as the scripting language as well as a neo4j server power through cypher database querying language. Neo4j is a highly scalable, native graph database purpose-built to leverage not only data but also its relationships. The dataset we were provided with is a Facebook data set that represents random sample data taken from the application and is hosted through Snap.py. The data package included multiple datasets instances that represent a random individuals social circle and friend connections. Finding number of nodes required me to recognize unique numbers within the edge data. Finding the number of edges directly correlated to the edge file as well since Facebook is not directional. For determining betweenness of centrality, I needed to essentially find the middle member in friend of friend relationships and find the node that acquired the highest magnitude of

this stat. Lastly for the clustering coefficient I had to utilize the following formula. N_v represents the number of links between neighbors of a particular node. K_v represents the degree of connections that are directly linked to a node with magnitude of 1.

$$CC = \frac{2 * N_v}{K_v(K_v - 1)}$$

4 Hypothesis Responses

ssingh3-101 Shorter the path between two nodes, more likely they tend to be in the same circle.

By referencing test.py you can see a variety of neo4j queries. I would like for you to focus on the query starting at line 61 that is named connecting path query. This query allows one to find the shortest path between two nodes. I focused on the 3980.circles dataset for this hypothesis. I wanted to go ahead and collect all the shortest paths from this dataset 3980.circles for each circle. After I collected shortest paths I took the max shortest path from each circle by using the 3980.edges data through use of the shortest path query. I was surprised to find that most of the nodes in the circle were not connected. If they were connected they typically had a magnitude of three or greater. This made sense to me based on the clustering coefficient being relatively small which means there is more of a star like pattern in this social dataset. So after exploring this I feel this disproves the statement. Since this data shows that most of the individuals in the social circles are not even connected meaning there is not path or short path to find there is no conclusion that can be made about their likeliness to be in the same circle. Quantitatively this makes sense since the clustering coefficient is very small meaning they are not very close.

czhao13-01 People who have both same hometown and same university are more likely to have connection to each other.

For this hypothesis it was a little bit harder to make the connection. The hypothesis depends on people having the same hometown and university. Following finding this out we could use those that have such qualifications to utilize the friend neo4j query I compiled. Looking at test.py on line 66 this hows a query that establishes the friends of certain indiviudals a simple compare could show if these users are connected. Also utilizing the shortest path query this same objective could be accomplished. This result would then allow me to make a conclusion to prove or disprove the relation of hometown and education to be linked to likeliness of connection.