

Q1: Experiment Overview

There were 133 students in total. They were asked to randomly take interviews of each other. If anyone got impressed by someone's interview, then they have to checkbox his/her name in the google form provided. They were given 1 hr for the same. The impression network collected is sent to us in the form of a google sheet. We have to run the experiment on this dataset and choose the top leader by running a random walk on the graph with teleportation. The graph formed will be a directed graph. Person 1 may got impressed by Person 2's answer but Person 2 may not get impressed by Person 1. So these 2 nodes will be undirected. It can be bidirectional as well. First we will constructed the graph. The database is loaded from a csv file containing information about connections between individuals. The first step is loading the data. It involves loading the data from a csv file using the pandas library. The dataset is stored in a pandas data form, where each node represents an individual and their connections with other individuals. Then we constructed the graph using network library. Each node in the graph represents an individual and edges represent connections between Individuals. The graph is constructed based on the data loaded from the csv file. On running the code multiple times, i observed that there were few nodes which were highlighted separately every time in the graph. On analyzing that, these were the nodes which had maximum impression on the people (top few nodes). Each row in the data frame corresponds to a node and connections between individuals are represented as directed edges in the graph. The constructed network is plotted using matplotlib. The nodes are labelled with their corresponding e-mail addresses (they were actually their entry numbers but in the .csv file, they were under the column e-mail addresses) and the graph is displayed with sky blue nodes and black text. A random walk algorithm is applied to the network to identify the "superleader" node. The superleader is the node / person by which majority of the people got impressed. The random walk simulates a process where an individual starts from a random node and transfers the network by randomly choosing neighbors at each step. The number of times each node is visited during the random wall is recorded. It somewhere follows the gold coin algorithm which is used by google and other search engines. The node with the highest count is the superleader. As the search engines do, they assign a gold coin / point to a webpage / hyperlink which is visited by the user and increment by 1 whenever it is visited again. At the end, the hyperlink with the maximum gold coins is displayed as the top results. It also follows the algorithm rich gets richer. But our activity does not follow this as we did not know the rich one here initially.

Q2: Matrix Factorization for Missing Links

We also had to recommend missing links using the matrix method. Matrix factorization is performed on the adjacency matrix of the network to identify missing links. An adjacency matrix in CS is a square matrix used to represent

a finite graph. The adjacency matrix represents the connections between individuals in the network. Each entry in the adjacency matrix indicates whether there is a connection edge between 2 nodes in the graph, the adjacency matrix is asymmetric. Matrix factorization is used to decompose on adjacency matrix and matrix Q . The error between the observed and predicted values is computed using a suitable loss function, such as mean squared error. The error is calculated for each observed entry in the adjacency matrix. After completing the optimization process, the factor matrices P and Q are multiplied to reconstruct the adjacency matrix. By comparing the reconstructed matrix with the original adjacency matrix, missing links in the network can be identified. If the reconstructed value exceeds a certain threshold(0.75), it suggests the presence of a missing link between two nodes in the network.

Q3: Shortest Path Calculation

We have to calculate the shortest path between any two nodes. So I randomly selected two nodes (source node and target node) and first calculated its shortest path and then shortest path length. It determines the shortest path directed from one node to another.