

December 3, 2019
David D'Souza
300065238

CSI2110 Assignment IV Report

This report explains my solution to assignment 4. It explains the classes, methods, and my solution to the assignment.

Overview of classes

A4

The main method to be run is found in the class A4. It calls the readGraph method which reads the csv file (in this case "email-dnc.edges") and returns a Graph object. Then it prints the number of nodes in the graph and the 10 most influential nodes. It also writes to a text file ("test.txt") all the nodes and their pr value along with what was printing to the terminal.

GraphNode

I created a new class called GraphNode to represent the nodes in the graph. GraphNode encapsulates the number of the node, its pr value, and a list of the nodes that are directed toward the node. The methods in this class are all accessory methods to get and set the encapsulated values. The readGraph method in A4 was changed to accommodate a list of GraphNodes.

Graph

The Graph class has a constant for the damping factor (0.85 in this case) and three other private variables – two lists of graph nodes, one of type Integer that just has the number of each node and one of type GraphNode, and one map of the edges (the node is the key and the value is the list of nodes that the node directs to). The way that the arguments are passed in from the readGraph method in the A4 class, the index for the same node in the two lists are the same. When the constructor is called, calculateNodesIn is called.

Explanation of non-trivial methods and solution:

calculateNodesIn

This private method in the Graph class iterates through the list of graph nodes and populates the list of nodes that are directed toward each node in the graph. It loops through all the nodes in the graph, get the value associates to it in the edges map and adds the current node to the nodesIn list.

calculatePr

This private method in the Graph class iterates through the list of graph nodes and assigns each node's pr value. This is done easily, because after calculateNodesIn is run, each node has a list of

all the nodes that is directed toward it. The formula used to calculate the pr of each node A is $PR(A) = (1-d) + d (PR(T1)/C(T1) + \dots + PR(Tn)/C(Tn))$.

mostInfluence

This public method in the Graph class returns an array of the 10 nodes in the graph with the highest pr value. calculatePR is first run 20 times for the pr values of the nodes to be accurate. Then the graph node iterates over and the ten nodes with the greatest pr values are saved into an array and returned.

Analysis

calculateNodesIn iterates through each node in the graph of size n. In each iteration, the list of nodes that the node directs to are processed, the worst case being that the node is connected to every other node in the graph (n-1 nodes). Within the processing, the index of the current node is searched for, the worst case being that it's at the end of the list. So, calculateNodesIn runs in $O(n*(n-1) * n)$ or $O(n^3)$.

calculatePr iterates and processes are similar to calculateNodesIn. So, its running time is also $O(n^3)$.

mostInfluence calls calculatePr 20 times and then iterates through the graph nodes once. So, it runs in $O(20n^3 + n)$ or $O(n^3)$.