CS 495 - Introduction to Web Science

Fall 2014

Assignment 4

by

Eric Littley
UIN: 00821698

October 9, 2014

Instructor

Dr. Michael Nelson

Department of Computer Science
Old Dominion University

Honor Pledge

I pledge to support the Honor System of Old Dominion University. I will refrain from any form of academic dishonesty or deception, such as cheating or plagiarism. I am aware that as a member of the academic community it is my responsibility to turn in all suspected violations of the Honor Code. I will report to a hearing if summoned.

Signed: Eric Littley

Contents

1 Introduction

Assignment 4 requires the use of the list of 1000 URIs downloaded in assignment 2. 100 URIs from the list were selected and the HTML representation was downloaded for each. The HTML was then parsed for outbound links. The links were stored in files that were in turn parsed into a dot format. This format is used by Gephi and Graphviz tools to generate graph visualizations of the data. Gephi is also used to perform calculations for: HITS and PageRank, average degree of separation, the network diameter, and the strongly connected components.

2 Design

This assignment was broken down into four parts: selecting the 100 URIs to use, downloading and parsing HTML for the URIs, putting the parsed files in dot format, and graphing and computing on the dot file. The first attempt at this assignment, used the first 100 URIs that had been saved. There was no other parameter used for selection. Also, the initial script that downloaded HTML and extracted links (parsehtmllinks.py) extracted every single link from each HTML file. The script responsible for creating the dot file used hashes for the URIs instead of the URIs themselves, this was an attempt to make everything look more uniform the first time around. However, when these dot files were graphed they had over 7500 nodes and the nodes were completely unreadable (partially because overlapping was not disabled). The Git Hub directory for assignment 4 contains a directory called galaxies, this contains the PNG pictures of some of these early attempts, they are not included in the report, because they were too big and complex to accurately display (but they look pretty cool). Although it it was nearly impossible to see the contents of each node, the graphs made it clear that the links used were highly independent with little linkage.

Three major things had to change: less nodes needed to be used, the nodes selected needed to be more strongly connected, and some form of human understandable representation of the URI needed to be used (not hashes). The solution was to select 100 URIs based on the content in the URI. The first 100 that contained the word "twitter" were selected. To minimize the number of nodes, the parsing script capped the number of links from each URI at 25. The representation of the URI used for the nodes consisted of the domain name of the website followed by the first 6 letters of the hash value of that URI (for example: www.cnn.com/4i5er5), this form allowed for a fairly compact, unique, and understandable string for each node. Also, the 100 nodes that contained the original 100 URIs were shaded red and all the other nodes were shaded blue. This allows the observer of the graphs to quickly see which of the original links are connected. Figure 1 below shows the Graphviz graph generated from these URIs. Twopi was used for Figure 1. The graph generated by Gephi is shown in figure 2. The labels are difficult to see in figure 1, but in the github the image is uploaded as twitter100final.png. The Gephi image did not hold the labels generated for some reason so the Gephi graph does not show the labels.

3 Graphs and Results

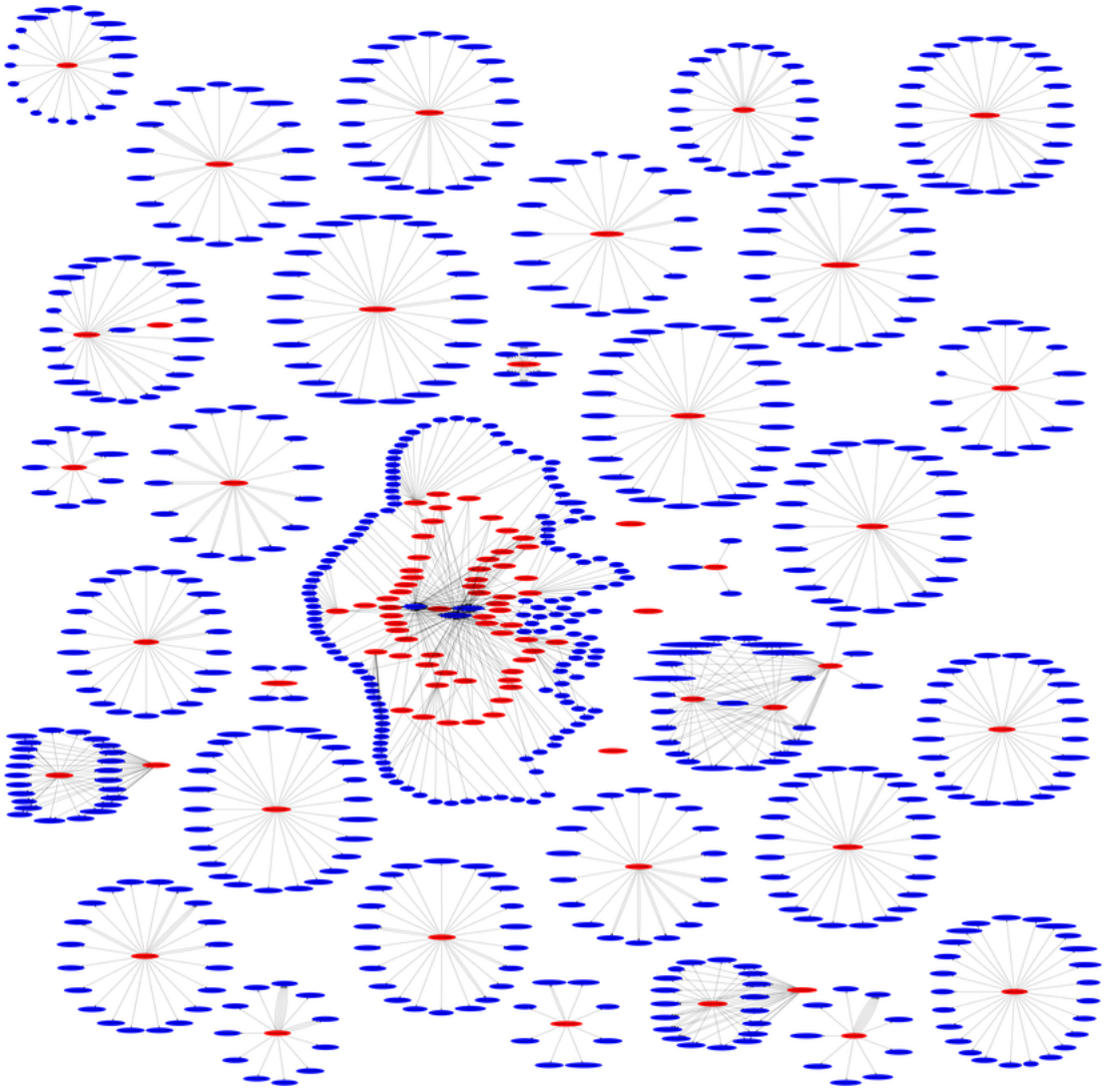


Figure 1: Twopi Graph

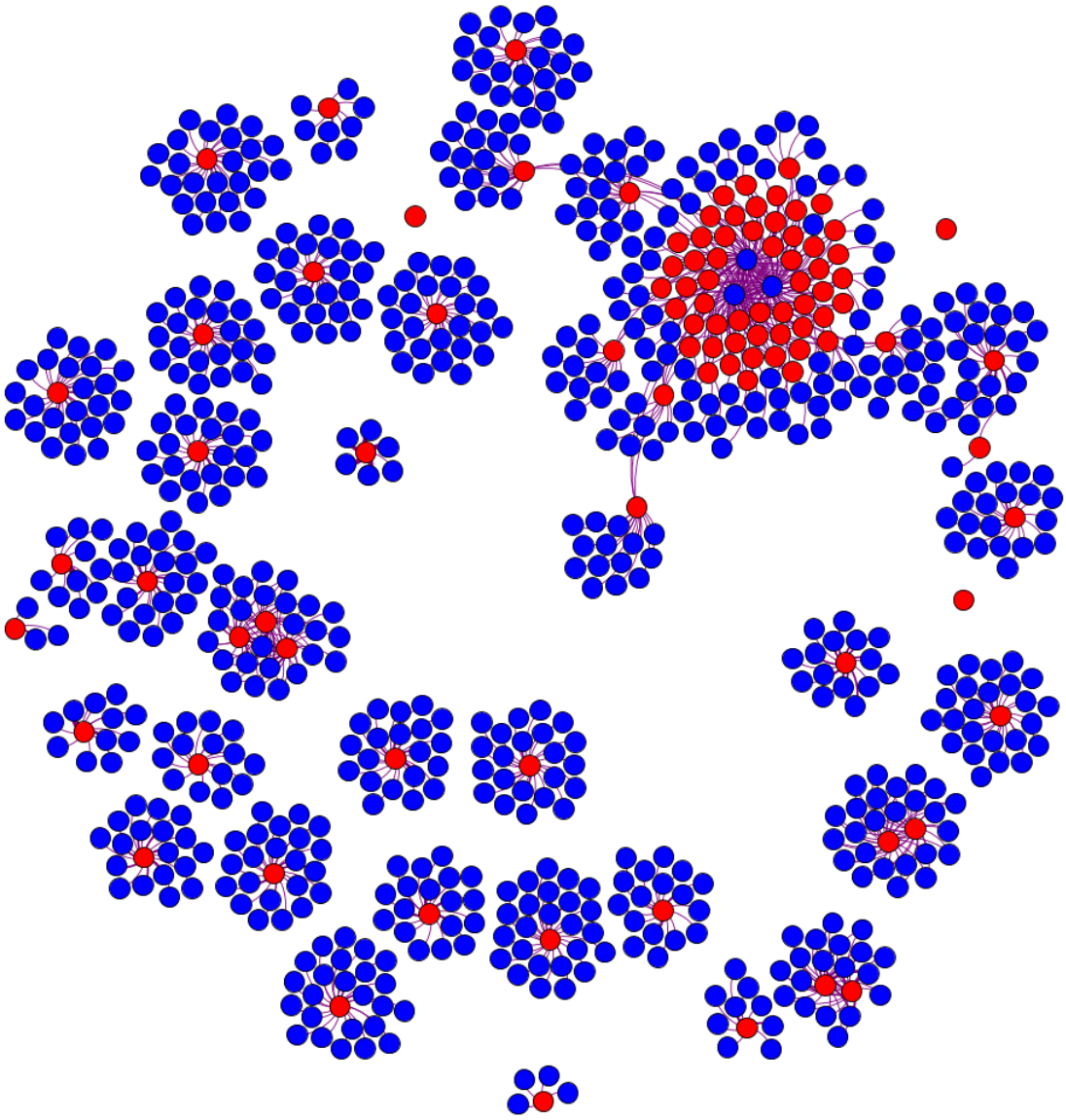
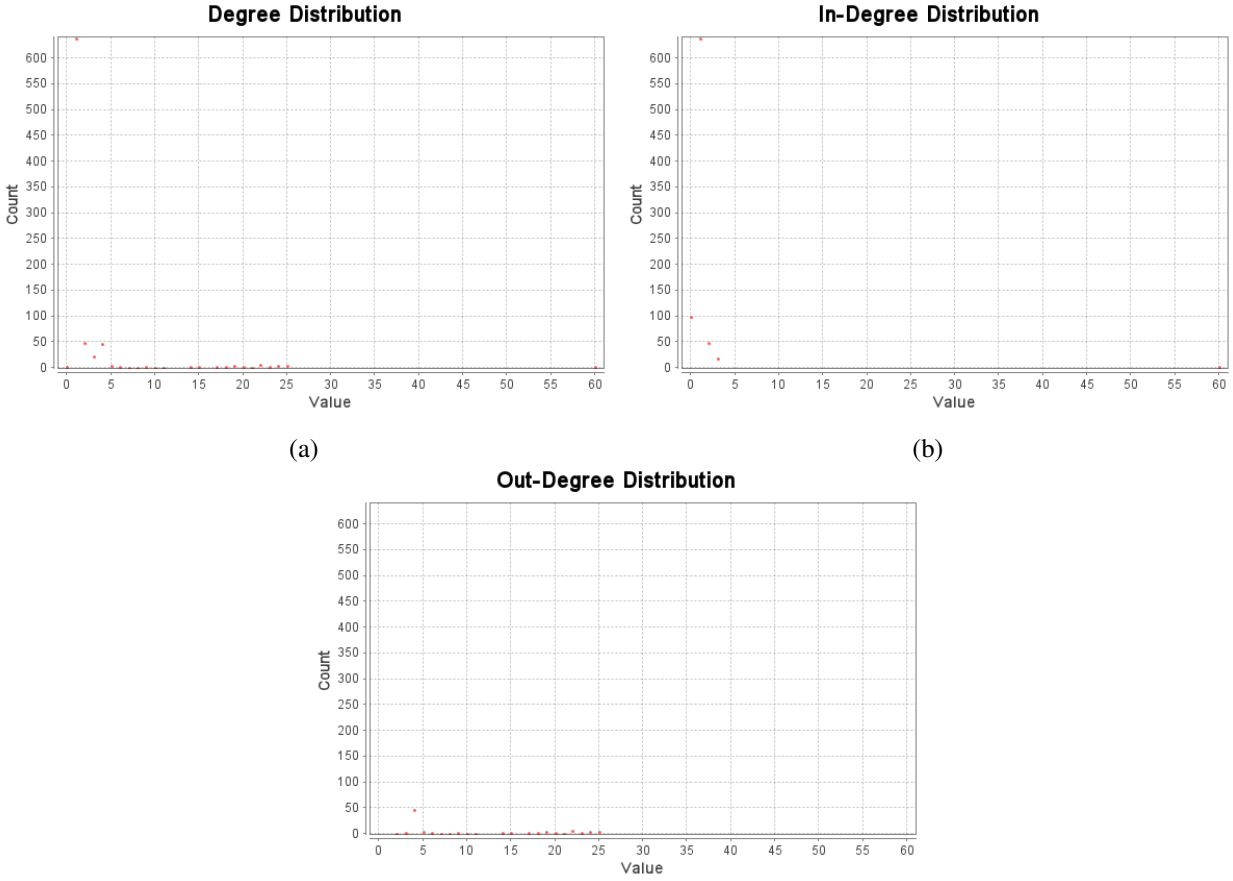


Figure 2: Gephi Graph

3.1 Results

The following graphs were optained from Gephi. The default values were used for all of them.

Average Weighted Degree: 1.341



(a) (b)

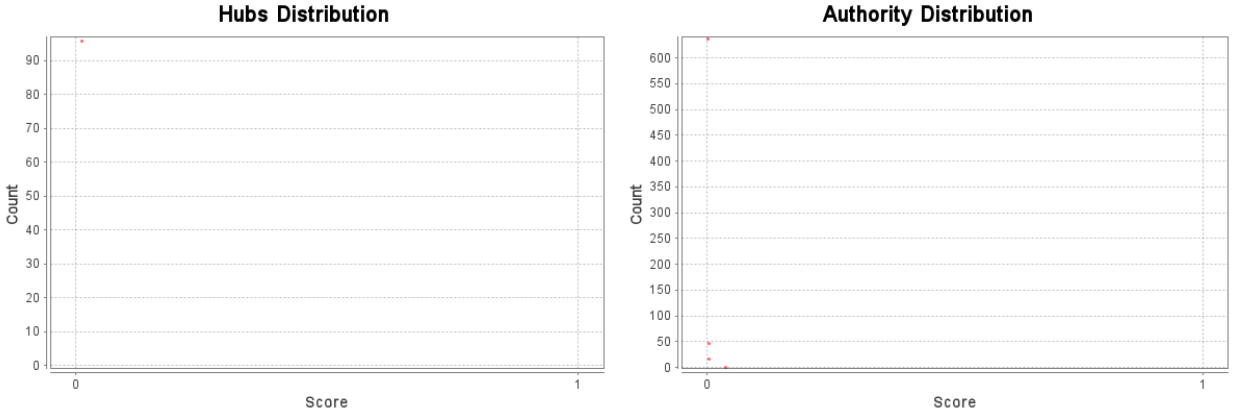
(c)

Figure 3

E= 1.0E–4

Epsilon = 0.001

Probability=0.85



(a)

(b)

(c)

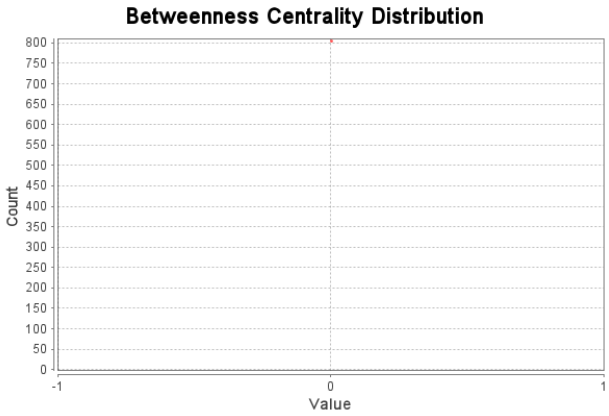
Results:

Diameter: 1

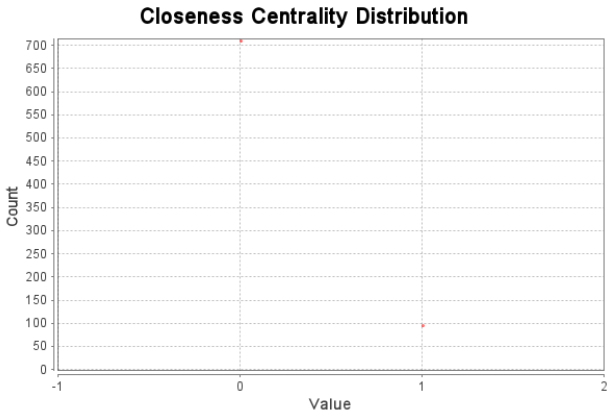
Radius: 0

Average Path length: 1.0

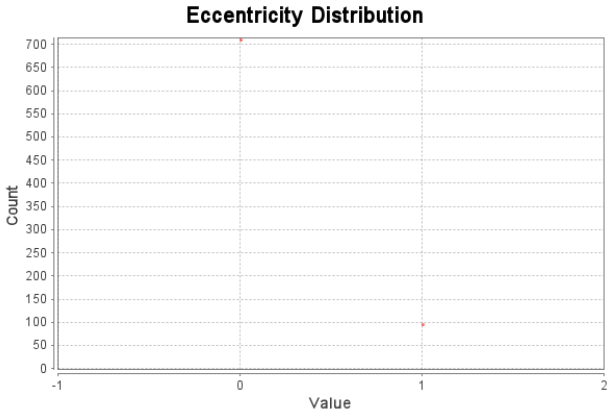
Number of shortest paths: 971



(a)



(b)



(c)

Results:

Number of Weakly Connected Components: 35

Number of Stronlgy Connected Components: 807

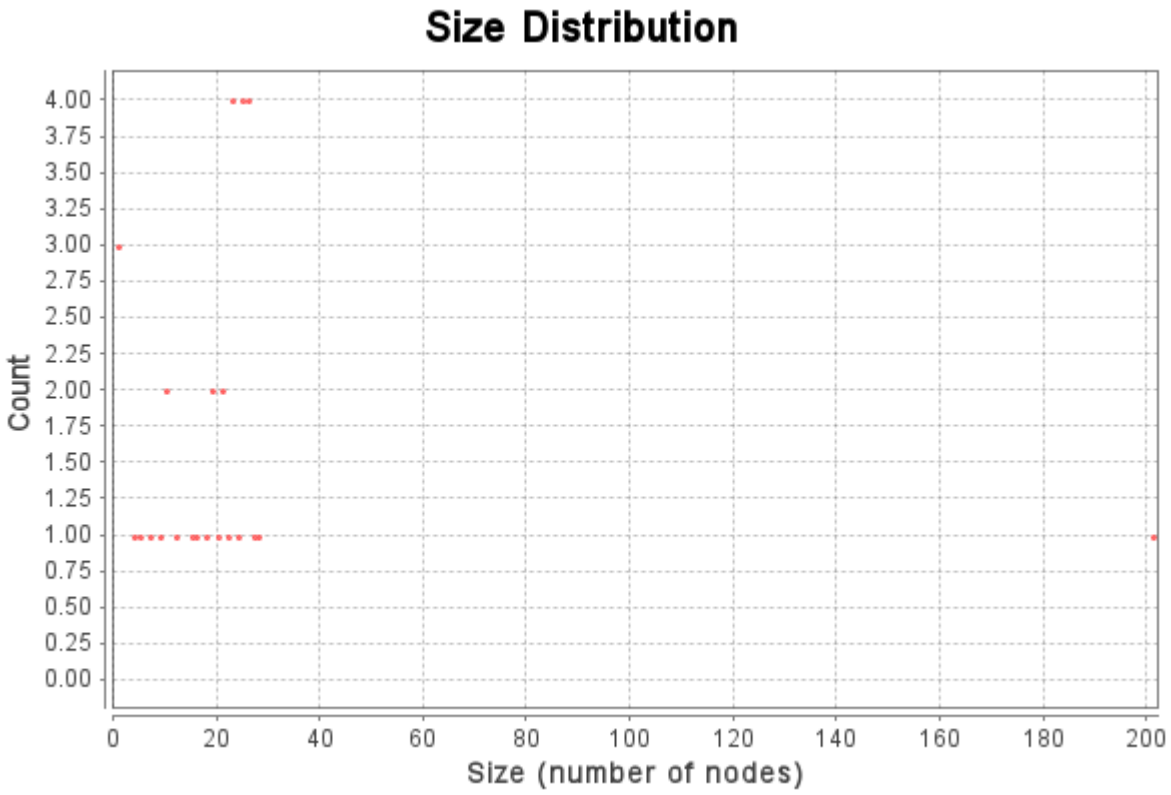


Figure 6