Complex Networks [CS60078] 2023-24

Assignment 1: Calculating Centrality Measures

Deadline for submission: 10 February 2024, 23:59 IST

General Instructions

1. In the assignment we will use the CORA network (description provided in the next page).

2. Your codes should print out exactly what is asked, and in the specified format.

3. How and what to submit: Submit one .zip or .tar.gz file containing a compressed folder that should contain all source codes, all files to be submitted (as per the task descriptions given below) and an instructions file (see next point). Name the compressed file the same as your roll number. Example: name the compressed file “19CS60R00.zip” or “19CS60R00.tar.gz” if your roll number is 19CS60R00.

4. Along with the source codes and files asked in the tasks, also submit an additional text file called “instructions.txt” where you should state how to run your codes as well as any additional information you want to convey, such as the version of Python. The instructions.txt file should also contain your name and roll number.

5. We should be able to run your submitted code in a computer with a reasonable configuration (for instance 2GB or more RAM) by following your submitted instructions. If any part of your code takes a long time to run (e.g., more than 10 minutes) report that in the instruction file with an estimate of time required.

6. The assignment should be done individually by each student. You should not copy any code from one another, or from any web source. Plagiarized codes will be awarded zero for the whole assignment. **You should not share your codes with anyone even after the submission deadline has passed.**

7. **Submit your response here: https://forms.gle/Aj1h1iT3eSVHYxP87**

Dataset Description

This directory contains a selection of the Cora dataset (https://linqs.org/datasets/#cora).

The Cora dataset consists of Machine Learning papers. These papers are classified into one of the following seven classes:

Case_Based
Genetic_Algorithms
Neural_Networks
Probabilistic_Methods
Reinforcement_Learning
Rule_Learning
Theory

The papers were selected in a way such that in the final corpus every paper cites or is cited by atleast one other paper. There are 2708 papers in the whole corpus.
After stemming and removing stopwords we are left with a vocabulary of size 1433 unique words. All words with document frequency less than 10 were removed.

THE DIRECTORY CONTAINS TWO FILES:

The .content file contains descriptions of the papers in the following format:

```
<paper_id> <word_attributes>+ <class_label>
```

The first entry in each line contains the unique string ID of the paper followed by binary values indicating whether each word in the vocabulary is present (indicated by 1) or absent (indicated by 0) in the paper. Finally, the last entry in the line contains the class label of the paper.

The .cites file contains the citation graph of the corpus. Each line describes a link in the following format:

```
<ID of cited paper> <ID of citing paper>
```

Each line contains two paper IDs. The first entry is the ID of the paper being cited and the second ID stands for the paper which contains the citation. **The direction of the link is from right to left. If a line is represented by "paper1 paper2" then the link is "paper2->paper1". Note that your results will be wrong if you do not assume the correct convention.**

Task Description:

Write a code to compute the following centrality metrics for a graph:

1. Closeness centrality for node i, given by $C_i = \frac{n-1}{\sum_j d_{ij}}$, where d_{ij} is the length of the shortest path from i to j, and n is the number of nodes in the graph. Note that this is a slight variation from what was covered in class such that high closeness centrality indicates a more important node.

2. Betweenness centrality for node i, given by $B_i = \frac{1}{(n-1)(n-2)} \sum_{st} \frac{n^i_{st}}{g_{st}}$, where n^i_{st} is the number of shortest paths between nodes s and t which pass through i, and g_{st} is the total number of shortest paths between nodes s and t. Note that the scaling factor would have been $\frac{2}{(n-1)(n-2)}$ in the case of an undirected graph.

3. PageRank for a node is calculated using the standard PageRank power-iteration method. Use a damping factor, α = 0.8.

Please note the following carefully:

1. Your code file should be named gen_centrality.py, and should NOT take any input arguments. Include the dataset in your submission.

2. It should output a text file each for the centrality measures, which contains a line for each of the nodes. Name the output files “closeness.txt”, “betweenness.txt” and “pagerank.txt”. Generate the files inside a folder called “centralities”.

3. Each line in the output files has the format: nodeID <white space> centrality value. The centrality values can be rounded up to 6 decimal places. **The nodes in each file should be sorted by the centrality value.**