TSKS33 Hands-On Session 3

Fall 2023

Version of this document: October 17, 2023

Preparation Work (to be Done Before Coming to the Lab)

- Study the course material on centrality metrics, Chapter 5 in the course notes by E. G. Larsson, and Chapter 2 in the course textbook.

Important: A lot of material on centrality, of varying quality, and different variants of the metrics and algorithms, can be found on the Internet. For the lab, use the versions of the metrics and algorithms described in the course notes by E. G. Larsson.

Summary of the Task

The task is to implement a few centrality metrics and experiment with them on sub-networks of the English Wikipedia project. Nodes correspond to Wikipedia pages, and edges correspond to links between the pages.

There are twenty sub-networks available, numbered by $N = 1, 2, 3, ..., 20$. The files `titles/`N`.txt` and `links/`N`.txt` contain the Nth sub-network:

- The file `titles/`N`.txt` contains the titles of the page/article that the nodes represent. The title of node i is found on row i in the file.
- The file `links/`N`.txt` contains the links between the nodes. Each row in the file represents a link from the node in the left column to the node in the right column. Note that the network is directed.

All files are located in the working directory, `TSKS33-sandbox/session3/`.

Examination will be on network 1. To help debugging, we provide correct answers for network 2. Also work on networks number 3, 11 and 17 to get a feeling for how the different metrics work.

Note: built-in functions for centrality in SNAP, NetworkX, or similar libraries may **not** be used. However, standard linear algebra functions such as eigenvalue decomposition can of course be used.

Task 1

Calculate the in- and out-degrees of all articles. Show the results in two lists: one list that shows the in- and out-degrees of the five nodes with the highest in-degree, and one list shows the in- and out-degrees of the five nodes with highest out-degree.

After computing the centrality scores, normalize them so that they sum to unity.

Task 2

Calculate the hub and authority centrality of all articles. Display the result in the same way as in the previous task.

After computing the centrality scores, normalize them so that they sum to unity.

Task 3

Calculate the eigenvector centrality of all articles. Use the eigenvector corresponding to the *largest* eigenvalue. Display the result in the same way as in the previous task.

After computing the centrality scores, normalize them so that they sum to unity.

Task 4

Calculate the Katz Centrality of each article, using

$$\alpha = 0.85 \cdot \frac{1}{|\lambda_{\max}|},$$

where $\lambda_{\max}$ is the largest eigenvalue of the adjacency matrix. List the top five articles and their Katz score.

After computing the centrality scores, normalize them so that they sum to unity.

Task 5

Calculate the Google PageRank score of each article in your network. Use the version of PageRank explained in the course material, and compute the PageRank scores by using the closed-form solution. Use the parameter value $\alpha = 0.85$. List the top five articles and their PageRank scores. Also try some other values of α and comment on the result.

After computing the centrality scores, normalize them so that they sum to unity.

Task 6

Implement the iterative version of PageRank. For the top-three articles found in Task 5, plot how the PageRank score evolves for 1, 2, 3, ..., 100 iterations. In the same plot, show also the “exact” closed-form solution from Task 5.

After computing the centrality scores, normalize them so that they sum to unity.

Task 7

Comment on the results of Tasks 1–5. Considering the top 5 articles, and in particular the “winner”, is the result plausible?

Task 8

Reflect on similarities/differences between the results of the different centrality metrics, and the known pros and cons of them.

Which of the metrics would you use in practice?

Structure of the data files

- The files with the links, `links/`*N*`.txt`, list all edges, one per row. In this example, node 1 points to nodes 2 and 3, node 2 points to node 5, and so forth:

```
FromNode  ToNode
1           2
1           3
2           5
.           .
.           .
```

Note that the adjacency matrix, $\boldsymbol{A}$, is not symmetric: $\boldsymbol{A} \neq \boldsymbol{A}^T$.

- The title files, `titles/`N`.txt`, contain the Wikipedia article titles. Row n in this file is the title of node n in the "links" file. There is exactly one title per row.

  Note: Be careful when loading this file, since some of the titles include commas (,). Make sure, if you read this file, that you have as many titles as you have nodes.

Hints

- To obtain a list with the titles of the articles, the following code can be used: `titles = open('titles/1.txt','r').read().strip().splitlines()`
- It can happen that the largest and next largest eigenvalues are extremely close. Due to numerical rounding errors, they may switch order.

Answers to Network #2 (for Debugging)

The file `answers_file2.pdf` contains a table with the correct answers for the networks in `links/2.txt` and `titles/2.txt`. A plot of the norm differences discussed in Task 6 is also included. This file is provided to help you validate your code.

Examination

- Individual oral examination takes place in class (computer lab). Students are also expected to be able to answer questions relating to the course material. All students must study the pertinent course material before coming to the lab.
- Before asking for oral examination, make sure to have all your plots, figures and code readily available. It is suggested to save all plots and code prepared in the lab by collating them into a document, for example, in PowerPoint or LibreOffice.
- Collaboration on this homework in small groups is encouraged, but each student should perform programming work individually, and individually demonstrate understanding of all tasks.
- Library functions from the Python standard library, numpy, scipy, matplotlib, and SNAP may be used freely. Copying of code from the Internet or from other students is prohibited.

- The program code you have written should be uploaded to Lisam (go to “submissions” and then select the lab session number) for an anti-plagiarism check.

  Plagiarism (copying of code from other students, from the Internet, or other sources) is a serious offense at LiU and normally leads to the filing of a disciplinary case.

Note: in the examination, show the results for **network 1**.