Social Media Analysis (SS2016) Dr. Krishna P. Gummadi

Exercise Sheet 4 – Eigenvector centrality

Max Planck Institute for Software Systems / Saarland University

Instructions

1. You have to submit this assignment (either digitally or on paper) by 10:00 am on 9th June 2016.
2. Irrespective of your submission preferences (digital or paper), you have to email the code of the coding assignments to the instructors.
3. To submit your assignments digitally, email your solutions to sma-ss16-instructors@mpi-sws.org, with the subject as "[SMA] HW 4 : <last name 1> <matriculation number 1>, <last name 2> <matriculation number 2> solutions".
4. Add your full name(s) and the matriculation number(s) in the beginning of all types of assignment solution submissions (in digital copy, on paper, inside the code as comment etc).
5. **You can submit solutions to both section A and section B. Your grade for this exercise will be maximum of the grade for section A and grade for section B)**. For example if you score 15 in section A and 0 in section B, your final grade for this exercise will be 15.
6. Instructions for coding assignment - Section A:
   (a) You are free to use any programming language you are comfortable with.
   (b) You should be able to run your code in a computer with a reasonable configuration (for instance 2GB or more RAM).
   (c) Please include the results outputted from your code into your solution sheet as described in the respective tasks. Additionally, you should also submit your code and instructions on how to run your code in a text file via email to the instructors. The name of the main code file for Problem 1 - Task 1 should be: **compute.eigenvector.py** (or python or some other programming language). Similarly the name of the running instruction file should be: **compute.eigenvector.howtorun.txt**. If there are any additional resources (eg. library) that are required to run your code, please mention those in the how to run file.
   (d) We should be able to run the submitted code on our machine. If it fails to run on our machine, you will not get any marks for the coding part of the assignment.
   (e) 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.

Section A

Problem 1 (15 marks)

Motivation and dataset description

The goal of this exercise is to demonstrate that eigen vector centrality can be computed using power method (`https://en.wikipedia.org/wiki/Power_iteration#The_method`), and that this method essentially gives the same result as more computationally intensive methods of finding all eigen vectors. We further investigate which are the top nodes in the graph using eigenvector centrality. We use two datasets from the previous assignment for this demonstration. The description of the two networks to be used are given below in Table 1:

Graph Name	**edge file**	**id to name map file**	**Description**
USairport_2010	USairport_2010.elist	USairport_2010.names	Network of airports. Two airports are connected by an edge if there is a flight between those two airports.
bible	bible.elist	bible.names	Network of nouns (names of people and places) in the Bible. Two nouns are connected by an edge if they appear in the same verse.

Table 1: Description of two real world datasets – bible and USairport.

Please note the following points carefully about the datasets mentioned in Table 1:

1. The datasets are downloadable from `http://courses.mpi-sws.org/sma-ss16/assignmentData/centrality/centralityDataSets.html`
2. All the elist files contain edgelists of respective networks, where each line is an edge of the form "id1< *whitespace* >id2".
3. Some elist files might contain a third column (to denote weight of an edge). Ignore this third column for the purpose of this assignment.
4. Consider all the graphs as undirected and unweighted.
5. The ".names" files contain the mapping from node id in the graph to actual names of those nodes (for example, airport codes and biblical nouns.). These will be required for semantically interpreting the results.
6. The airport codes can be mapped to actual airport names using the file "lookupAirportCode.csv" downloadable from same page (`http://courses.mpi-sws.org/sma-ss16/assignmentData/centrality/centralityDataSets.html`).

Task 1 (10 marks)

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

- **Eigenvector centrality** The centrality score of a node is its corresponding value in the leading eigenvector (with highest eigen value) of the adjacency matrix of network.

For this centrality write a code called **compute.eigenvector.py** (or cpp or some other programming language), which perform the following:

1. Construct the adjacency matrix of the network from elist. [2]
2. Find the eigen vectors of the constructed matrix using some standard library function. Consequently identify the highest eigen value and corresponding eigne vector (i.e., leading eigen vector). [2]
3. Now use the power method, take a column matrix as your initial vector where there is one row for each node and all values of the matrix is 1. After matrix multiplication in each iteration, normalize the resulting column vector with the norm of the vector before next iteration. [2]
4. Stop the power iteration when your process converges at iteration i, i.e. when the vector obtained from iteration i and iteration (i + 1) are aligned in the same direction (angle between them is 0 or the absolute value of cosine of the angle between those two vectors is 1.). [2]
5. Find the eigenvalue after your power method converged at iteration i. Hint: When a vector has converged, if you multiply the converged vector with adjacency matrix the norm of the vector is simply multiplied by the highest eigen value. [2]
6. The eigenvector you obtain after the power method converges, gives you the eigenvector centrality values of each corresponding node. Using this find the top 10 nodes (with names from the .names files) according to eigenvector centrality. [2]

Your code should take the elist file and name file as inputs and at the end should output the following lines:

1. === <networkName> : BEGIN RESULT ===
2. <networkName> : converged on iteraction <iteration number>
3. <networkName> : Highest eigen value using library <eigen value>
4. <networkName> : Highest eigen value using power method <eigen value>
5. <networkName> : Top 10 nodes using eigen vector centrality. (Then print out top 10 node ids (most central to least central) and their names. in 10 lines.)
6. === <networkName> : END RESULT ===

Append your code output for two files in **compare.eigenvector.results.**< *graphname* >**.pdf/doc/docx/...**, or submit on paper. If your output is wrong we might not check the code.

[2+2+2+2+2+2=12]

Task 2 (3 marks)

Take top 10 nodes found using *degree centrality, closeness centrality and harmonic closeness centrality (from previous assignment) and eigen vector centrality (using power method in this assignment)* for each of these 2 networks : "USairport_2010", "bible" and perform the following tasks:

1. For each of the two graphs, create a table with four columns. Each column must correspond to one centrality value and name the column headings accordingly. In each column, put the top 10 entities (biblical nouns or airport names) according to the corresponding centrality value. Please note, that you will need to convert the node ids to the actual names using the ".names" files. Add the resulting tables in

the corresponding submission files **compare.centrality.results.**< *graphname* >**.pdf/doc/docx/...**, or submit on paper.

Based on the above tables, comment on how eigenvector centrality measure compares with other measures (in your opinion), for finding the most important nodes for each of the two datasets and why. Summarize your most interesting observations in a few sentences. Again, add your observations and reasoning in the files **analyze.centrality.results.**< *graphname* >**.pdf/doc/docx/...** or submit on paper.

[1.5 × 2 = 3]

Section B

Problem 2 (Research, 15 marks)

Identify a novel and significant problem based on the topics discussed in the class so far. The problem may be about devising some new model, proving a new theorem, identifying a new application or any other idea that you might have. You must first clearly identify the problem you are interested in solving, then motivate why you think this problem is important to answer and non-trivial to solve. Finally, you must propose / sketch a solution that you have in mind to address this problem.

Note that this part of the Exercise is aimed for the students who enjoy research and are motivated to identify and solve problems that remain unaddressed in today's world.

The grading for this section will be highly subjective and will be judged based on the novelty of the identified problem, motivation of the problem and the solution sketch that you provide. If you identify a problem that also interests us, we would be happy to follow it up with you. Additionally, we would invite some students who propose an exceptional problem, to present their work briefly in one of the tutorials.