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

Exercise Sheet 3 – Centrality Measures

Max Planck Institute for Software Systems / Saarland University

Instructions

1. You have to submit this assignment (either digitally or on paper) by 14:00 on 2nd 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 3 : <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. A reference for an efficient way to implement graph classes and methods is the github code base for snap at `https://github.com/snap-stanford/snap`.

   (b) Make sure that your code *only* contains the functions, classes and data structures that are used for computing the required metrics for the graphs.

   (c) You should be able to run your code in a computer with a reasonable configuration (for instance 2GB or more RAM).

   (d) 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: **analyze.centrality.cpp** (or python or some other programming language). Similarly the name of the running instruction file should be: **analyze.centrality.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.

   (e) 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.

   (f) 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 (35 marks)

Motivation and dataset description

As also explained in the class, the relationships between the entities of many real world datasets can be expressed as graphs. Their representation as graphs, in turn makes it possible to reason about the relative importance of the different entities by computing the centrality measures over these graphs. This exercise is aimed to demonstrate this process, by requiring you to compute the different centrality measures (degree, closeness, betweenness and harmonic centralities) over multiple real world datasets represented as graphs. Since it is not always clear which centrality metric would work the best for each graph, you also need to reason about how well the different metrics capture the most important or central nodes in the different graphs.

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.
imdb_actor	imdb_actor.elist	imdb_actor.names	Network of actors collected from IMDB. Two actors are connected by an edge if they appeared in the same movie.
recipe network	American_recipes.elist	American_recipes.names	Network of ingredients in American recipes, collected from allrecipes.com. There is an edge between two ingredients if they are used in the same dish.
	South-America_recipes.elist	South-America_recipes.names	Network of ingredients in South american recipes, collected from allrecipes.com. There is an edge between two ingredients if they are used in the same dish.
	India_recipes.elist	India_recipes.names	Network of ingredients in Indian recipes, collected from allrecipes.com. There is an edge between two ingredients if they are used in the same dish.
	Germany_recipes.elist	Germany_recipes.names	Network of ingredients in German recipes, collected from allrecipes.com. There is an edge between two ingredients if they are used in the same dish.

Table 1: Location and description of some real world datasets.

The real world networks that you would be using for this exercise, can be downloaded from the following page : `http://courses.mpi-sws.org/sma-ss16/assignmentData/centrality/centralityDataSets.html`.

This page also contains how we extracted the graph structure for the recipe network. This is meant to demonstrate to you that although it is quite possible to collect and convert real world data into graphs, the task of creating the network is non-trivial, both conceptually and methodologically. In this assignment, however, we are already giving you all the cleaned graphs. Table 1 gives a brief description of each of the real world dataset network.

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

1. All of these 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.
5. The ".names" files contain the mapping from node id in the graph to actual names of those nodes (for example, airport codes, biblical nouns, names of actors etc.). 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`).
7. For the closeness centrality computation, consider only the largest connected component of the graphs.

Task 1 (10 marks)

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

1. **Degree centrality**
2. **Closeness centrality** for node i, given by $C_i = \frac{n}{\sum\limits_j d_{ij}}$, where d_{ij} is the length of the shortest path from i to j, and n is the number of nodes in the graph.
3. **Harmonic closeness centrality** for node i, given by $C_i = \frac{1}{n-1} \sum\limits_{j(\neq i)} \frac{1}{d_{ij}}$, where d_{ij} is the length of the shortest path from i to j, and n is the number of nodes in the graph.
4. **Betweenness centrality** for node i, given by $C_i = \sum\limits_{st} \frac{n^i_{st}}{g_{st}}$, 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.

Your code should take the elist file as an input and should output a text file each for the four centrality measures, which contains a line for each node and has the format : nodeID <white space> centrality value. Each file should be sorted by the centrality value.

Please note the following carefully:

1. Your code file should be named **analyze.centrality.cpp** (or python or some other programming language).

2. You should also submit an instructions file named **analyze.centrality.howtorun.txt** to tell us how to compile and run your code.

3. You can build your code on SNAP (using graph classes etc.). The source for cpp version of SNAP is at `https://github.com/snap-stanford/snap`.

4. There are already functions in SNAP (cpp version) called "GetBetweennessCentr" and "GetClosenessCentr". Please **don't copy** these functions, you need to implement these centralities yourself, as extensions to SNAP's graph framework.

5. In fact you can implement more efficient ways of computing centralities in your code. There are multiple algorithms, like variation of Floyd Warshall algorithm, Johnson's algorithm and Brandes' algorithm. You can read more about them here: `https://en.wikipedia.org/wiki/Betweenness_centrality#Algorithms` .

6. Please also mention in your how to run file which algorithm you used for computing each centrality (otherwise we would deduct marks).

7. Before each function in your code, write (in comments) any functions or data structures (that are written by you) that this function uses. This is to ensure that you **only include functions or classes in your file that your code actually uses**.

8. This task is crucial for solving the next tasks. Therefore, if it fails to run on our machine you will not get any points for this assignment.

[5]

Task 2 (15 marks)

Run your centrality code for *degree centrality, closeness centrality and harmonic closeness centrality* on the elist files of each of these 3 networks : "USairport_2010", "bible" and "imdb_actor", and perform the following tasks:

1. Submit the three resulting output text files (containing the nodeID and centrality values for all nodes in the graph, sorted according to the centrality values), named as < *graphname* >**.degree.txt**, < *graphname* >**.closeness.txt**, and < *graphname* >**.harmonic. txt**.

2. Plot the centrality values in the four files, where the X-axis of each plot are the nodes (sorted by their centrality values) and Y axis is the corresponding centrality value. Include the plots in your assignment submission, and the names of the plots should be < *graphname* >**.degree.png**, < *graphname* >**.closeness.png**, and < *graphname* >**.harmonic .png**.

3. In few sentences each, summarize your most interesting observations about the plots generated in the previous steps. Submit your answers in a file called **analyze.centrality.results.**< *graphname* > **.pdf/doc/docx/...**. Alternatively you can also submit the answers on paper.

4. For each of the three graphs, create a table with three 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, airport names or actors 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 **analyze.centrality.results.**< *graphname* >**.pdf/doc/docx/...**, or submit on paper.

5. Based on the above tables, comment on which centrality measure works best (in your opinion) for each of the datasets for finding most important nodes 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.

$[(1 + 1 + 1 + 1 + 1) \times 3 = 15]$

Task 3 (10 marks)

Run your centrality code for *degree centrality, closeness centrality, harmonic closeness centrality and betweenness centrality* on the elist files of each of these 4 recipe networks : "American_recipes", "SouthAmerica_recipes", "India_recipes" and "Germany_recipes"', and perform the following tasks:

1. Submit the four resulting output text files (containing the nodeID and centrality values for all nodes in the graph, sorted according to the centrality values), named as < *graphname* >**.degree.txt**, < *graphname* >**.closeness.txt**, < *graphname* >**.betweenness.txt**, and < *graphname* >**.harmonic.txt**.

2. For each of the four centrality measures, create a table with four columns - one corresponding to each type of cuisine (American, South-american Indian and German). In each column, put the top 10 entities (ingredient names) sorted 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. Mention the centrality measure which is reflected in each table in its caption. Add the resulting tables in the file **compare.country.ingredient.results.pdf/doc/docx/...**, or submit on paper.

3. Based on the above tables, for each centrality measure comment on how the central ingredients differ across cuisines and whether or not it matches with your intuition. Summarize your most interesting observations in a few sentences for each of the centrality measures. Again, add your observations and reasoning in the file **compare.country.ingredient.results.pdf/doc/docx/...** or submit on paper.

$[(0.5 + 1 + 1) \times 4 = 10]$

Section B

Problem 2 (Research, 35 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.