PROJECT: NETWORK SCIENCE-BASED ANALYSIS OF COLLABORATION NETWORK OF DATA SCIENTISTS

NETWORK SCIENCE

TOTAL MARKS: 30

PROJECT DESCRIPTION

Data scientists are increasingly playing critical role in all data-driven endeavors in today's world. In this open-ended project, the goal is to explore the following question: ***Can network science help us to understanding research collaboration among data scientists over time and select a subset of them for various tasks?*** Here we measure research collaboration as co-authorship among individuals in scientific papers/articles.

In order to understand research collaboration of data scientists, we need to have access to the list of publications of each scientist. To this end, you should use the *DBLP computer science bibliography* (https://dblp.uni-trier.de/) to seek answer to the grand question. It is an on-line reference for bibliographic information on major computer science publications. It has evolved from an early small experimental web server to a popular open-data service for the whole computer science community. *DBLP* indexes over 5.9 million publications, published by more than 2.9 million authors. Specifically, it contains the temporal history of publications of each author (e.g., institutions, year of publication, co-authorship, publication venue) including data scientists. Each individual is uniquely identified by their DBLP address. In this project, your goal is to analyze this data source (you can download individual data scientist's data in XML format from DBLP), to answer a set of questions.

For ease of reference, a list of data scientists, DBLP address, and some attributes are provided to you as input (*DataScientists.xls* file). Note that the names in the file and their corresponding names in DBLP entry may not be identical. You should use the latter for the project. The *expertise* field in the file has a value between 1 to 10, which you can randomly assign to each individual. Also, like any real-world data, the file is not clean and has duplicate records. Note that the input file is configurable and hence you should not be hard coding anything.

1. What are the network properties of the **collaboration network**? Note that the network should only contain individuals in the input file as nodes. No other individual should be part of this network.
2. How has the collaboration network and its properties evolved over time? That is, the program should be able to analyze the network properties over time (at yearly granularity).
3. Assume that we create a **random network** from the set of individuals in the input file. How does the properties of this network differ from the real collaboration network in (1)?
4. The real collaboration network is expected to have high-degree nodes (hubs) residing with many low-degree nodes. High-degree nodes sometimes may have undesirable influence on their co-authors for some team-based task (e.g., reviewing someone's paper or grant application) especially when the high-degree node is an influential person. Hence, the goal is to transform the collaboration network to a network with the following characteristics:
    a. The transformed network has **smaller** giant component and **larger** number of isolates than the original collaboration network.
    b. The maximum degree of any node **does not exceed** beyond a user-specified k_{max}, referred to as *collaboration cutoff*, which is typically smaller than the degrees of hubs.
    c. Ensure rich diversity of individuals (country, expertise, institutions) in the transformed network.

   A straightforward solution is to simply remove some of the nodes associated with bridges as well as those connected to high-degree nodes. However, a more judicious strategy would be to ensure that the transformed network maintains the *diversity* of individuals w.r.t country, expertise, and institutions as close to the original network as possible. Design and implement an algorithm to generate the transformed network from the network created in (1).

Note that the above question set is **not overly prescriptive** in order to facilitate unleashing of your creativity. You are free to pose additional questions that you think are relevant to this project.

DEVELOPMENT ENVIRONMENT

You <u>must</u> use **Python 3.0 (or higher)** in **Windows** environment for your project. You are free to use any publicly available Python-compatible libraries (i.e., can be installed using pip) for your development.

SUBMISSION REQUIREMENTS

Your submission should include the followings:

- In order to facilitate grading, you should submit **a single** program file: *project.py* with a *main()* method. **Note that we shall be running the project.py file** (using the Pychamp IDE) to execute the software. Make sure your code follows good coding practice: sufficient comments, proper variable/function naming, etc. Your software folder should have two subfolders, **Input** and **Results**. The **Input** folder contains the input file and a configuration file, if any (e.g., you can specify k_{max} in it). The **Results** folder contains all results of your program (save all plots, figures, using *png* format). Add a read-me file if necessary to specify steps for successfully running your code.
- **Softcopy report** containing details of the features supported by your software, algorithms, analysis and insights, and results of the above questions. Lastly, you should also discuss limitations of the software (if any).
- Project submission will be done via the "Project" menu at NTULearn. Note that only submit one zipped file. Please remember to add a read me file to mention how to run your program successfully.