INFS7450 Social Media Analytics

Project 2 – Link Prediction

Semester 1, 2023

Marks:	15 marks (15%)
Submission Due:	26 May 2023 16:00 (Brisbane Time)
Deliverables:	See deliverables part
How to submit:	Electronic submission via Blackboard

Goal: This project aims to design and implement an effective algorithm to predict social links. The implemented algorithm will be evaluated on a real-life co-author dataset, and its performance will be reported. Project details will be included in this assignment handout. Students are required to finish this project individually.

Dataset: In this project, you will be working with a co-author network. The dataset contains the following files:

training.txt: This file contains training dataset for you to develop your prediction methods. Each line of this file is a link in the network during the training time period.
val_positive.txt and ***val_negative.txt***: This is the validation set. These two files contain validation links for you to tune and validate your developed methods.
test.txt: This is the test set which contains the unlabelled edges to be ranked.
example.txt: This is an example result file. You must follow the format of this file to submit your results.

The dataset is available from UQ blackboard. See /Assessment/INFS7450 Project Two.

Tasks:

1. Predict the missing links formed in the future. (**15 marks**)
   Overview: The provided co-author network has 5,242 nodes, 11,696 edges. The edges of the whole co-author network are then split into three parts, which are E_{train} (11,496 edges), $E_{validation}$ (including two parts: 100 positive edges in ***val_positive.txt*** which were randomly selected from the complete dataset and 10,000 negative edges in ***val_negative.txt*** which were built at random and not overlapped with E_{train} and 100 positive edges in $E_{validation}$), and E_{test} (100 positive edges and 10,000 negative edges which were constructed in the same way but not overlapped with $E_{validation}$ and unlabeled). The missing 100 positive edges are formed among the core nodes (whose degrees are larger than 3). Based on the given training and validation sets of the co-author network, you are required to write a program to rank the unlabeled edges in the test set. For each pair of nodes in the test set, your program should compute a proximity score. Rank the 10,100 pairs of nodes according to your computed proximity score in descending order and the Top-100 edges (or pairs of nodes) will be compared with the ground truth to compute accuracy.
   Input: The provided network datasets.
   Output: The predicted Top-100 edges.

Requirements:

1. There is no restriction on algorithms, packages.
2. You are not allowed to use any generative models such as **ChatGPT.**
3. You can refer to the codes provided in the tutorial, but are not allowed to directly reuse or copy them.
4. You are not allowed to look at the code of any other student. **All submitted codes and reports will be subject to electronic plagiarism.**

Programming Languages:

1. Python and NetworkX are recommended. However, you have your own choices of preferred programming languages including, but not limited to, Python, MATLAB, Java, C, C++, etc.

Deliverables (!!VERY IMPORTANT):

Your submission must include the following:

1. A source code file.
2. A report (.pdf). See the given appendix for an example template.
3. A text file of the predicted Top-100 node pairs (edges). The format of this file must follow the format of the provided example file - ***example.txt.***
4. Name all the submitted files with your student ID. For example, 41234567.zip for the source code, 41234567.txt for your submitted results and 41234567.pdf for your report.
5. Submit one archive file with your student number as the file name (e.g. 41234567.zip). Make sure that all the files mentioned above are in the archive file.

Please make sure your submission totally follows the above requirements!

Marking criteria (Total marks: 15):

- 15 marks = 4 marks (code) + 7 marks (results) + 4 marks (report)
- Your results should be reproducible and your codes should be readable. If your codes cannot be executed or generate the results as reported, the corresponding marks for the code and results will be deducted.
- Results Mark will be calculated as follows:
    A. If your accuracy >=0.9: 7 marks (full marks)
    B. If 0.8<= your accuracy <0.9: 6 marks
    C. If 0.7<= your accuracy <0.8: 5 marks
    D. If 0.6<= your accuracy <0.7: 4 marks
    E. If 0.5<= your accuracy <0.6: 3 marks
    F. If 0.3<= your accuracy <0.5: 2 marks
    G. If 0.1<= your accuracy <0.3: 1 marks
    H. If your accuracy <0.1: 0 marks

  where "Accuracy = The number of correctly predicted edges/100". Accuracy is calculated based on your submitted results compared against the ground truth.
- **Time Limitation:** The prediction running time of your code must be less than an hour. **Otherwise, the score for your code part will be 0**.

Kaggle:

We will be deploying our projects on the Kaggle platform. Students are encouraged to upload their results to Kaggle and compete with other classmates.

You can join the Kaggle competition via the below links:
https://www.kaggle.com/t/b606de45b1f34201932dce6b509924a0

Please note that the scores on this platform will not be considered in the final grading. The final score will still be determined by the results you upload to Blackboard.