Assignment 2: Social recommendations

Due: NOVEMBER 25, 2017.

In this assignment, you will write a program that makes friend recommendations.

Recommendation systems

Facebook suggests people you may be friends with. Netflix suggests movies you might like. Amazon suggests products to buy. How do they do that? In this assignment, you will implement one simple way to make such suggestions, called collaborative filtering.

Collaborative filtering says that, if your past behavior/preferences were similar to some other users, then your future behavior may be as well. As a concrete example, suppose that you like John, Paul, and George, and other people like John, Paul, George, and Ringo. Then, it stands to reason that you will like Ringo as well, even if you had never previously heard of him. The recommender system does not have to understand anything about what "John", "Paul", "George", and "Ringo" are — they could be everything, and the algorithm would work identically.

In this assignment, you will implement a collaborative filtering recommendation system for suggesting friends.

Representing a social network as a graph

A graph or network represents relationships among things. Things are represented as nodes, and relationships are represented as edges. One common use for a graph is to represent travel possibilities, such as on an airline map. The nodes of the graph are cities, and the edges show which cities are directly connected. Then, you can use the graph to plan travel. Another common use for a graph is to represent friendship among people in a social network. An edge between person *A* and person *B* means that *A* considers *B* a friend, and also *B* considers *A* a friend.

Recommending friends

In this assignment, you will implement one mechanism for recommending a new friend in a social network. A simple way to state this question is, "For user X, who is the best person to recommend as a friend?"

You will answer a more comprehensive question: "For user X, list some *non-friends* in order, starting with the best friend recommendation and ending with the worst." A non-friend is a user who is not X and is not a friend of X. Note that the recommendations might not be symmetric: the best friend recommendation for X might be Y, but the best friend recommendation for Y might be W.

Your task will be to write code that, given a user U in the social network, produces friend recommendations for U, in order from best to worst. You will do this by assigning each potential friend a number called a score, where higher scores indicate a better match. Then you can sort your list according to the score. Given user X, if two people Y and Z would be equally good as new friends for X (they have the same score), then they should be listed in numerical order (for numerical user IDs).

Problem 1: Recommend by number of common friends

If non-friend Y is your friend's friend, then maybe Y should be your friend too. If person Y is the friend of many of your friends, then Y is an even better recommendation. The best friend recommendation is the person with whom you have the largest number of mutual friends. You will implement this heuristic.

As a concrete example, consider the user "A".

A has two friends in common with B (C and D).
A has two friends in common with F (C and G).
A has one friend in common with K (G).
A has one friend in common with W (G).
A has one friend in common with P (G).
A has no friends in common with the N.
A has no friends in common with M.

Therefore, B and F are the best friend recommendations for A, and the N and M are the worst friend recommendations. (In fact, the N and M are such poor friend recommendations that your program will not even suggest them.)

In overall, write a program that produce suggestions for friends. As a hint, consider the following main steps:

- Find the friends of friends for a user in question
- Find the common friends between the user in question and his/her friends of friends
- Compute the number of common friends for each friend of friend
- Sort the friends of friends based on their corresponding number of common friends
- Recommend the top-4 friend of friends.

Any programming language for your assignment is acceptable.

Please explain any assumptions you made.

Send your codes at kostas.stefanidis@uta.fi before NOVEMBER 25, 2017. Some instructions on how to run your codes are necessary.

Also send the top-4 recommendations for 4 particular users, namely, the users with id: 20341, 33722, 35571 and 25017.