Assignment 2: Similar Items, Data Streams, Page Rank

Formative, Weight(10%), Learning objectives (1,2,3),
Abstraction (4), Design (4), Communication (4), Data (5), Programming (5)

Due date: 11:59pm, 18 September 2016, Weight: 10.0 % of the course

1 Overview

Assignments should be done in groups consisting of two students. If you have problems finding a group partner use the forum to search for group partners or contact the lecturer.

2 Assignment

Exercise 1 *Filtering Streams (similar to Exercises of 4.3 in Rajaraman and Ullman) (8 + 7 points)*

1. For the situation of our running example in Section 4.3.1 with changed conditions (10 billion bits, 2 billion members of the set S), calculate the false-positive rate when using three hash functions. Do the same for four hash functions.
2. As a function of n, the number of bits and m the number of members in the set S, what number of hash functions minimizes the false-positive rate?

Exercise 2 *PageRank (25+10 points)*

1. Implement the PageRank Algorithm as discussed in Section 5.1 and 5.2 in Rajaraman and Ullman. Your implementation should make use of the improvements regarding efficiency and the methods of dealing with dead-ends and spider traps.
2. Run your algorithm on the Google Web Graph 2002 available at

   `http://snap.stanford.edu/data/web-Google.html`

   and provide a file listing the PageRank for each node. Report separately, the ordered list of the ten nodes having the largest PageRank.

Your approach should be as efficient as possible in terms of runtime and memory requirements.

Exercise 3 *Frequent Itemsets (15+15+10+10 points)*

For this exercise, we follow consider the A-Priori Algorithm in Section 6.2 in Rajaraman and Ullman.

1. Implement the A-Priori Algorithm for generating all frequent pairs.
2. Generalise your approach to all frequent itemsets as outlined in Section 6.2.6
3. Run your approaches on the datasets T10I4D100K, T40I10D100K, chess, connect, mushroom, pumsb, pumsb_star provided at

   `http://fimi.ua.ac.be/data/`

   and report the outcomes.
4. Experiment with different support thresholds s for the different datasets and report on the number of frequent sets for $k = 1, \ldots, 5$ for these thresholds. What is an appropriate support threshold s for each of the datasets (give a justification).

Your approach should be efficient as possible in terms of runtime and memory requirements. Report on challenges that you might have observed in the implementation and by running the experiments.

3 Procedure for handing in the assignment

Work should be handed in using the course forum. The submission should include:

- pdf file of your solutions for theoretical assignments. The solutions should contain a detailed description of how to obtain the result.
- all source files
- descriptions of your implementations to understand your code
- files containing the results of your algorithms on the benchmark sets
- computation times of the algorithms on the benchmark sets
- a README.txt file containing instructions on how to run the code, and also the names, student numbers, and email addresses of the group members

In addition, there will be a discussion session where you will have to explain your solutions:

1. Individuals (not groups) will be chosen randomly (by a program) to explain their solutions to parts of exercises in the assignment.
2. Each member of a group shall be able to explain their solution. An answer like "my group partner did this hence I can't explain this solution" will be counted as "no explanation" ($\rightarrow$ 0 marks for the particular part).
3. Absence without a formal excuse[1] for the session will be counted as "no explanation" if the person is chosen to explain his/her solution.

[1] Markus Wagner and Junhua Wu need to be informed via email *before* the discussion session.