Assignment 1

This is the second part of Assignment 1. The deadline for this part is at the beginning of the class on November 26th. You should turn in the code for question 3, and either turn-in the remaining questions, or submit them on paper. For late submissions the late policy on the page of the course will be applied. The details for the turn-in are on the page of the course.

Question 1

Exercise 6.3.1 from Chapter 6 from the book Mining Massive Datasets by Anand Rajaraman, Jeff Ullman, and Jure Leskovec.

Question 2

Assume that you are given as input the Twitter graph, with the information of who follows whom. You want to find *(s,t)-bicliques*, that is, sets of users, where *s* users all follow the same *t* users. The *s* users are users with common interests, while the *t* users are users of the same "type". Describe an efficient algorithm for solving this problem that makes use of the APriori idea.

Question 3

The goal of this question is to use frequent itemset mining in practice.

As the input dataset you will use a collection of approximately 57K of FourSquare "tips" on restaurants in New York. The details about the data are on the web page of the course. In this dataset a "basket" is a tip, and the "items" are words. The data needs to be preprocessed so that noisy data is removed, the words are normalized (no punctuation, lower case), and the English stop-words are removed. A list of stop words can also be found at the web page of the course.

You will use an existing software package for mining frequent itemsets. You can use either an implementation from FIMI repository, or the WEKA software. Details on how to download the software are on the page of the course. You may need to transform your data to the format required by each package.

Run the code for obtaining the frequent itemsets using a high enough support threshold that does not produce a huge amount of frequent itemsets. If the software you chose supports it, output closed and

maximal itemsets. Transform the output to a readable form (e.g., transform items from ids to words). Then inspect the results and comment on itemsets that you found.

You should turn in the following:

- All the code that you have written yourselves.
- The output frequent itemsets.
- A short report where you describe how you did the preprocessing, the choice of support threshold, and the commentary on the output.