University of Ioannina,
Department of Computer Science & Engineering

Data Mining – CSE 012/CS 059
Spring 2017

Assignment 1

This is the first assignment. The deadline for the assignment is March 31, 11:59 pm. You should turn in the code for question 1, and submit the remaining questions either electronically, or on paper. For late submissions the late policy on the page of the course will be applied. Details for the turn-in, and how to write reports are on the Assignments web page of the course. The examination of the Assignment will be in the week before Easter.

Question 1 (Reservoir Sampling)

In this question you are required to modify the Reservoir Sampling algorithm to sample $\boldsymbol{K}$ items from a stream of N items.

1. Describe the algorithm for sampling K items uniformly at random from a stream of N items. The algorithm should work in a single pass over the data, reading the items one by one, without prior knowledge of the size of the stream N, and using $O(K)$ of memory (assume the size of an item is fixed). **Do not** write code or pseudocode for this part; just explain the steps of algorithm in English.
2. Prove that your algorithm produces a uniform sample, that is, for every $i, 1 \leq i \leq N$, the i-th element has probability K/N to appear in the sample.
3. Write a program **in Python** that implements the sampling algorithm. Your program should sample K lines from a text document. It should be possible to use the program from command line, it should take as command line argument the value of K, read lines from the standard input, and output the sample in the standard output. For example the following command should print a random sample of 10 lines from the file input.txt:
   "`sample.py 10 < input.txt`".

Question 2

On the Assignments page of the course there is a file "data.csv". The file contains three comma-separated columns of 1000 values, with column names A, B, and C. The values in B and C are a function of those in A. Your goal is to find the relationship between columns B, C, and column A. Hand in an Iron Python Notebook, which should contain the code for processing the data, the plots and computations that you did, and a report with your conclusions.

Question 3

In this question you will implement and apply the Apriori algorithm for finding frequent itemsets. For the implementation you can consider the basic version of the algorithm we described in class.

The application of the algorithm will be on a set of tweets that are given in the file "tweets_dataset.txt" in the Assignments page of the course. The file contains tweets from followers of Donald Trump and Hillary Clinton which were collected in the time interval October 25-30, 2016, during the election campaign. Our goal is to find interesting co-occurrences of tags and handles in the tweets (these would be the items for our study).

First, you need to clean up the data. The file contains tab-separated entries with 14 columns that correspond to the following fields:

Name, ScreenName, UserID, FollowersCount, FriendsCount, Location, Description, CreatedAt, StatusID, Language, Place, RetweetCount, FavoriteCount, Text

Each line of the file is a tweet. Throw away all tweets that are retweets (the text starts with RT), and from the text keep only the hashtags (words that start with #) and the handles (words that start with @). In some tweets there are punctuation marks at the end of the hashtag or handle that you need to clean up, or the hashtags are concatenated. Clean up also obviously incorrect cases (you can experiment with a sample to find such cases). Create a "basket" for each tweet that contains at least one hashtag or handle. Save the result in a new file.

Using the dataset you created print some basic statistics, so as to give an idea of what your data look like. (e.g., number of baskets, number of distinct items, mean number of items per basket, histogram of basket sizes, number of hashtags vs handles, etc). Comment on your results.

Apply the Apriori algorithm for finding frequent itemsets on your data using support threshold between 0.02% and 0.05% of the dataset size. Find the maximal frequent itemsets (frequent itemsets that are not contained in a larger frequent itemset). Given that the frequent itemsets are not that many you can use a simple implementation to find the maximal frequent itemsets. Create a plot that shows how the number of itemsets changes for different support thresholds.

Go through the results (especially for the most frequent of the itemsets) and commend if you see any interesting co-occurrences, and what they may mean. Comment in your report about the co-occurrences that are interesting.

Hand in your code (it is recommended to do the assignment in Python, but you may use some any other language), the files with the maximal frequent itemsets, and your report with the plots and your observations (you can put code and report together in an IPython notebook). The report is very important in the evaluation of the assignment, since you will comment on what is interesting in the data.

Bonus: A different way to create "baskets" is by putting together all hashtags and handles tweeted by a single user (no repetitions). Find the maximal frequent itemsets in this case, and compare with the case where we have a basket per tweet.