TI2736-B: Assignment 1
Big Data Processing

Due date: 27.11.2016 (11.59pm)

Please submit your report and code via Blackboard: the report should be in PDF format, the source code files should be submitted as separate files with proper code documentation. Please submit a properly formatted report: it should contain your name, student number, understandable answers (sentences, not notes or keyword lists) with a numbering that makes it clear which answer refers to which assignment question.

1. Big data & data streams

   (a) Read *The unreasonable effectiveness of data* by Halevy et al. (you find it on Blackboard). It discusses the use of Web data (i.e. big data) for various machine learning tasks. State what the authors indicate as the main advantage of Web data and the main disadvantage of Web data. Describe four settings (domains, applications) where *small data is better than big data*.

   (b) Imagine a company that wants to predict the spread of chickenpox in the Netherlands: when a local area is predicted to have a strong likelihood of a chickenpox outbreak in the near future, kindergartens and schools in the area are notified to be more alert and isolate children suspected of having chickenpox-like symptoms (this in turn will decrease the overall number of infected children, reducing health care costs). The company decides to base its daily predictions on Twitter data, in particular those tweets that mention "chickenpox", "waterpokken" or mention specific symptomps of chickenpox. The company builds a sophisticated machine learning algorithm that takes the gender, age, location and profession of the user into account to determine the veracity of the user's tweets. It promises to predict chickenbox outbreaks with 95% accuracy.
   What do you think about this idea, is it going to work or not? Argue for or against this idea (based on the knowledge you gained in the first lecture) in 150-200 words.

   (c) In lecture 2 the basic stream processing architecture was shown. Each element of a stream is processed by the stream processor and summaries of the stream are created. Take a look at the Meetup RSVP stream visualized here:
   `http://bit.ly/2fYr0Oy`
   Each element of the stream is a meetup (=informal gathering around a topic) user who just positively responded to a meetup invitation. Shown is the user name, the user avatar, the meetup name, the response time and the location of

the meetup. Given this data stream, think about four useful standing queries that can be applied to it. For each query, write down the kind of data stream summaries that need to be maintained to answer the query.
To help you, here is a concrete example: one standing query may be the list of cities that host more than 1% of all meetups. To answer this query a number of counter/value pairs need to be maintained by the SPACESAVING algorithm.

(d) Imagine you want to build a system that can determine the drug-interaction side-effects to any number of medications. A specific use case would be to help doctors prescribe the correct medication in cases where patients already take a number of pharmaceuticals for a range of issues. This system should work world-wided and include the latest research findings in drug interactions. Which (big) data sources would you need for such an app? What are the challenges you are likely to encounter with respect to the data?

2. Frequency counter algorithms

(a) In this exercise you are asked to implement the two frequency counter algorithms introduced in lecture 2: FREQUENT and SPACESAVING.
The skeleton Matlab code for this exercise is available on Blackboard. It contains a simulated stream R of randomly generated integers and has a number of TODOs for you to implement. If you prefer another programming language, feel free to use it.

- For both algorithms, how large is the error you observe with varying k, m and standard deviation (σ)? What influence do these parameters have on the accuracy of the approximation?
- Slide 41 of lecture 2 discusses the boundary of FREQUENT's approximation error. For the setting of $m = 50,000$, $k = 5$ and $\sigma = 1$ compute the lower and upper bound for each of the $k - 1$ frequent items and compare it to the estimate your implementation achieves.
- Please include your code when handing in this assignment.

3. Bloom filters

(a) Given are two hash functions for a $n = 20$ bit Bloom filter:

$$h_1 : x \rightarrow ((3x + 1) mod\ 29) mod\ n \tag{1}$$

$$h_2 : x \rightarrow ((5x + 7) mod\ 23) mod\ n. \tag{2}$$

Work out the bit vectors for two separate Bloom filters, each with the following input (values of x):

- BF_1: 20 17 100 7 33
- BF_2: 14 8 22 9 6

(b) In the lecture, we briefly discussed counting Bloom filters to support element deletions. A colleague of mine has the following idea as an alternative to counting Bloom filters: in order to support element deletions we are going to use two standard bit-vector based Bloom filters (B_{add} and B_{del}) in tandem to implement a filter that allows deletions. B_{add} is used as usual: every element added to our filter sets its corresponding bits in B_{add}. Elements that are to be deleted from our filter are added to B_{del}. When we test whether an element occurs in our filter, we now have to check both B_{add} and B_{del}:

- To test whether an element e is in our filter, we first check B_{add}; if it returns `FALSE` e is definitely not in our filter. If `TRUE` is returned, we also check whether e appears B_{del}. If it does not (implying that e has not been deleted), our filter returns `TRUE`, otherwise it returns `FALSE` (implying that e has been deleted).

Does this idea adhere to the promise of Bloom filters that false positives may occur, but false negatives are impossible? If you think yes, argue why. If you think no, provide a concrete example of a violation of this promise.

(c) Assume a Bloom filter with k hash functions and a bit vector of size n (all bits are set to 0 initially). Further assume that every position in the bit vector can be selected by each hash function with equal probability. Compute the probability that a particular bit b_i in the bit vector is set after the insertion of x elements.

(d) How would you estimate the number of times an element e has been added to a counting Bloom filter based on just the information provided in the filter (i.e. you are not allowed to make changes to the counting Bloom filter)? What can you say about the accuracy of your estimation?

(e) Imagine you want to build a spell checking application for Dutch which contains a dictionary of 500,000 words. Given a text, each word is tested by the spell checker - when a word it is not found in the spell checker's dictionary, it is assumed to be erroneous.

i. Derive an *estimate* of how much memory you would need when implementing such a dictionary via a set data structure (e.g. java.util.HashSet).

ii. Assume that you are given a method which converts each term into a unique integer to use as input for the hash functions. If you use a Bloom filter and want to aim for a 2% error rate (with $k = 5$ and $k = 10$), what would be the memory footprint of the respective Bloom filters?