CMPSCI590D: Algorithms for Data Science

Homework 2

Instructor: Barna Saha — Posted: Oct 11th, Due: Nov 1st

Do not look up materials on the Web. You can consult the reference books mentioned on the course website, and also the class slides for solving the homework problems. You may work in a group of size at most 3*. Every person is allowed to talk to at most 2 others. All communications are bidirected: A talks with B, automatically implies B talks with A. Therefore, those who are working in a group of* 3 *are not allowed to talk with anyone else outside the group. Mention any collaboration clearly in the submission. Submit one homework solution per group. No late homework will be accepted.*

For programming assignments, submit your code with a detailed readme file that contains instruction for running it. Also include any test dataset that you have used and results obtained to show correctness of your implementation.

*Total Point:*150*, Bonus Point:*30

1. Consider a biased coin with probability $p = \frac{1}{3}$ of landing heads and probability $\frac{2}{3}$ of landing tails. Suppose the coin is flipped some number n of times, and let X_i be a random variable denoting the ith flip, where $X_i = 1$ means heads, and $X_i = 0$ means tails. Use the Chernoff bound to determine a value for n so that the probability that more than half of the coin flips come out heads is less that 0.001. [30]

2. Suppose we have n bits of memory available, and our set S has m members. Instead of using k hash functions for implementing a Bloom filter, we could divide the n bits into k arrays, and hash once to each array. As a function of n, m, and k, what is the probability of a false positive? How does it compare with using k hash functions into a single array? [30]

   For this exercise we will use the following dataset: `https://barnasahadotcom.files.wordpress.com/2016/01/words.xlsx` which contains a list of valid words.

   Implement a Bloom filter based spell-checker that uses 400 Kbytes of space. Now generate 100 random five letter words, and feed them into the bloom filter. For each word accepted by the bloom filter check the dictionary to see if that word indeed exists to determine the false positive rate. How does the false positive rate changes with the number of hash functions?

   Provide full explanation of your implementation. [30]

3. For this exercise we will use the following twitter data set. `https://www.cs.duke.edu/courses/fall15/compsci590.4/assignment2/tweetstream.zip` (2.1G). The meaning of the fields of a tweet can be found at `https://dev.twitter.com/overview/api/tweets`.

   Consider the following algorithm for finding frequent item.

   Maintain a list of items being counted. Initially the list is empty. For each item, if it is the same as some item on the list, increment its counter by one. If it differs from all the items on the list, then if there are less than k items on the list, add the item to the list with its counter set to one. If there are already k items on the list decrement each of the current counters by one. Delete an element from the list if its count becomes zero.

(a) Show that if the total stream size is m, then any item that has frequency $> \frac{m}{k+1}$ times occur in the list.

Implement the above algorithm for $k = 500$, and return all the hashtags that occur at least 0.002th fraction of times in the dataset. It is ok to return the first 15 characters of the hashtag.

[40]

(b) Now implement the Count-Min data structure along with min-heap such that any hashtag that occurs at least 0.002th fraction of times are returned, and any hashtag that is returned has frequency at least 0.001th fraction of the whole dataset size. You should have sufficient confidence on your answer.

Compare the results and space requirements of the two algorithms from (a) and (b).

[50]