CMPSCI514: Algorithms for Data Science

Homework 1

Instructor: Barna Saha — Posted: Feb 7th, Due: Feb 19th

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 2*. No further communication is allowed. Submit one homework solution per group through Moodle. 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/plots obtained to show correctness of your implementation.

*Total Point:*200*, Bonus Point:*20

Exercise 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* i*th 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*. [20]*

Exercise 2. *Suppose you play a simple game with your friend where you flip a coin. If the coin is heads, your friend pays you a dollar. If it's tails, you pay your friend a dollar.*

(a) *Suppose you play the game* 100 *times, what is your expected pay off? [5]*

(b) *Suppose your friend decides to trick you, and swaps the fair coin for a biased coin that comes up tails with probability* 0.7*. What is your expected pay off if you play* 100 *times? [5]*

(c) *Use Markov Inequality to give an upper bound on the probability that your friend gets more than* 50 *after* 100 *rounds. [10]*

Exercise 3.

(a) *Suppose that we roll a standard fair die* 100 *times, Let* X *be the random variable denoting the sum of numbers that appear over the* 100 *rolls. Use Chebyshev's inequality to bound* $\Pr[|X - 350| \geq 50]$*. [10]*

(b) *Chebyshevs inequality uses the variance of a random variable to bound its deviation from its expectation. We can also use higher moments. Suppose that we have a random variable* X *and an even integer* k *for which* $E[(X - E[X])^k]$ *is finite. Show that*

$$Prob\left[|X - E[X]| > t\left(E[(X - E[X])^k]\right)^{1/k}\right] \leq \frac{1}{t^k}. \quad [20]$$

Exercise 4. *Suppose you throw m balls into n bins, each ball equally likely to go into any of the n bins; imagine $m \geq n$. Let random variable B_i denote the number of balls in bin i. What is $E[B_i]$?*

- *Suppose $m = 100n \ln n$. Use the Chernoff bound to show that the number of balls in bin i does not differ from the expectation by more than (say) $25 \ln n$ with probability at least $1 - \frac{1}{n^2}$. Hence, show that the load of the heaviest and lightest bins differ by at most a constant factor with probability at least $1 - \frac{1}{n}$. [20]*
- *For general $m = \Omega(n \ln n)$, show that the number of balls in all the bins lie in the range $\frac{m}{n} \pm O(\sqrt{\frac{m}{n} \ln n})$ with probability at least $1 - \frac{1}{n}$. [20]*
- *(Extra Credit) Now suppose $m = n$. Show that the height of the heaviest bin is $O(\frac{\ln n}{\ln \ln n})$ with probability $1 - o(1)$. [20]*

Exercise 5. *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]