Asmt 4: Streaming

Turn in through Canvas by 1:05pm:
Wednesday, October 22
100 points

Overview

In this assignment you will explore finding frequent items in data sets, with emphasis on streaming techniques designed to work at enormous scale. For simplicity you will work on more manageably sized data sets, and simulate the stream by just processing with a for loop.

You will use two data sets for this assignment:

- `http://www.cs.utah.edu/~jeffp/teaching/DM/A4/S1.txt`
- `http://www.cs.utah.edu/~jeffp/teaching/DM/A4/S2.txt`

The first data set S1 has a set of $m = 3{,}000{,}000$ characters, and the second one S2 has $m = 4{,}000{,}000$ characters. The order of the file represents the order of the stream.

As usual, it is recommended that you use LaTeX for this assignment. If you do not, you may lose points if your assignment is difficult to read or hard to follow. Find a sample form in this directory: `http://www.cs.utah.edu/~jeffp/teaching/latex/`

You could also utilize an LaTeX template specifically created for this assignment. Click here.

1 Streaming Algorithms

A (30 points): [Sampling.] Implement a stream sampling algorithm. Either Reservoir sampling based, or bottom-k (where each item is assigned a random value $u_i \sim \mathsf{Unif}[0, 1)$, and you return the items with smallest u_i values); it can be with-replacement or without-replacement. Run two versions, with $k = 10$ and with $k = 100$. To get an estimate of counter f_j, we can compute the number of items matching j in the sample times (m/k). Although, for this question we will estimate the fraction of the stream that each j represents. For this, take the number of items matching j and multiple by $(1/k)$.

- Which version of stream sampling did you implement?
- For $k = 10$, report the estimate of the fraction of the stream for each item which appears in the sample, for stream S1 and S2. *(e.g., if item $j =$* `a` *appears* 3 *times out of* $k = 10$ *in the sample, then report for* `a` *the fraction* $3/10$ *or* 0.3*).*
- For $k = 100$, report the estimate of the fraction of the stream for each item which appears in the sample, for streams S1 and S2.

B (30 points): [Misra-Gries.] Run the Misra-Gries Algorithm (see Algorithm **11.2.2** in M4D book) with $(k - 1) = 9$ counters on streams S1 and S2.

- Report the output of the counters at the end of the stream. In addition to each counter report the estimated ratio for each label using the estimated count/m.
- In each stream, use just the counters to report which characters *might* occur at least 20% of the time.
- In each stream, use just the counters to report which characters *must* occur at least 20% of the time.

C (30 points): [CountMin Sketch.] Build a CountMin Sketch (see Algorithm **11.2.3** in M4D) with $k = 10$ counters using $t = 5$ hash functions. Run it on streams S1 and S2.

- For both streams, report the estimated counts for characters `a`, `b`, and `c`. In addition to each counter report the estimated ratio of each of these labels using the estimated count/m.
- Just from the output of the sketch, with probably $1 - \delta = 31/32$ (that is assuming the randomness in the algorithm does not do something bad), which objects *might* occur at least 20% of the time.
- Just from the output of the sketch, with probably $1 - \delta = 31/32$ report which objects *must* occur at least 20% of the time.

D (10 points): How would your implementation of these algorithms need to change (to answer the same questions) if each object of the stream was a "word" seen on Twitter, and the stream contained all tweets concatenated together?

2 BONUS

A: (2 points) We described a streaming algorithm for sampling k items without replacement in a stream proportional to their weights w_i. For each item we generated a random value $u_i \sim \mathsf{Unif}[0, 1)$ and then a priority $\rho_i = -\frac{1}{w_i}\ln(u_i)$. The sample is the set of k items with smallest priorities.

An alternative method instead creates a value $q_i = u_i^{1/w_i}$ and retains the items with the largest k values in q_i. Show that these approaches are equivalent.