Homework 3

Algorithms for Big Data
CS498ABD Spring 2019
Due: 10am, Wednesday, March 27th

Instructions:

- Each home work can be done in a group of size at most two. Only one home work needs to be submitted per group. However, we recommend that each of you think about the problems on your own first.

- Homework needs to be submitted in pdf format on Gradescope. See `https://courses.engr.illinois.edu/cs374/fa2018/hw-policies.html` for more detailed instructions on Gradescope submissions.

- Follow academic integrity policies as laid out in student code. You can consult sources but cite all of them including discussions with other class mates. Write in your own words. See the site mentioned in the preceding item for more detailed policies.

Exercise 1: Frequent items and Misra-Greis Algorithm We saw the deterministic Misra-Greis algorithm that uses k counters and outputs an estimate $\hat{f}_i$ for each f_i such that $f_i - \frac{m}{k+1} \le \hat{f}_i \le f_i$. Here m is the total number of elements in the stream.

- Let m' be the sum of the counters at the end of the algorithm. Show that the actual estimate provided by the algorithm is slightly stonger, namely, for each i,

$$f_i - \frac{m - m'}{k+1} \le \hat{f}_i \le f_i.$$

- Suppose we have run the (one-pass) Misra-Gries algorithm on two streams σ_1 and σ_2 thereby obtaining a summary for each stream consisting of k counters. Consider the following algorithm for merging these two summaries to produce a single k-counter summary.

  1. Combine the two sets of counters, adding up counts for any common items.
  2. If more than k counters remain:
     (a) $c \leftarrow$ value of $(k+1)$th counter, based on decreasing order of value.
     (b) Reduce each counter by c and delete all keys with non-positive values.

  Prove that the resulting summary is good for the combined stream $\sigma_1 \cdot \sigma_2$ (concatenation of the two streams) in the sense that frequency estimates obtained from it satisfy the bounds given in the previous part.

Exercise 2: Count Sketch   In the Count-Sketch analysis we showed that if we choose $w = 3/\epsilon^2$ and $d = \Omega(\log(n))$ that for each i we obtain an estimate $\tilde{x}_i$ such that with high probability $|\tilde{x}_i - x_i| \leq \epsilon\|x\|_2$. This can be pessimistic in situations where the data is highly skewed with most of the $\|x\|_2$ is concentrated in a few coordinates. To make this precise, for some fixed parameter $\ell \in \mathbb{N}$, let $y_i \in \mathbb{R}^n$ be the vector defined by the ℓ largest coordinates (by absolute value) of x, as well as the ith coordinate of x, to 0. (All other coordinates are the same as x) Prove that for $\ell = 1/\epsilon^2$, if w is chosen to be $6/\epsilon^2$ and $d = O(\log n)$, then for all $i \in [n]$, with high probability, we have

$$|\tilde{x}_i - x_i| \leq \epsilon\|y_i\|_2.$$

Exercise 3. JL preserves angles   Recall that the distributional JL lemma implies that a projection matrix Π chosen from an appropriate distribution preserves length of any fixed vector x to within a $(1 \pm \epsilon)$-factor with probability $1 - \delta$ if the number of dimensions in the projection is $O\big(\log(1/\delta)/\epsilon^2\big)$.

1. Suppose we have two unit vectors u, v. Prove that Π preserves the dot product between u and v to within a ϵ-additive factor with probability $1 - \delta$ with a slight increase in dimensions.

2. Show that with a slight increase in dimension, Π preserves the angle between any two vectors u, v up to a $\pm\epsilon$-additive factor.

Hint: Taylor expansion...

Exercise 4. Distinct elements in strict turnstile streams.   In class, we say how to estimate F_0 in the cash register setting where each item increases the frequency of a coordinate. Here we develop a different algorithm that allows us to estimate F_0 in the strict turnstile model where each item in the stream may increase or decrease a coordinate value, but every coordinate is always nonnegative.

More explicitly, the stream might look like

$$(i_1, \Delta_1), (i_2, \Delta_2), (i_3, \Delta_3), \ldots, (i_m, \Delta_m)$$

where for each j, $i_j \in [n]$ and $\Delta_j \in \mathbb{R}$ (positive or negative). The resulting frequency vector is

$$f = \sum_{j=1}^{m} \Delta_{i_j} e_{i_j}.$$

(Here $e_i \in \{0, 1\}^n$ is the ith indicator vector.) In the strict turnstile model, we are promised that $f_i \geq 0$ for all i. The number of distinct elements, d, becomes the number of coordinates i with $f_i > 0$. Note that this is at most, and *not necessarily equal to*, the number of distinct elements to every appear in the stream.

Our goal is to develop an algorithm for this harder streaming model. You may assume that $d \geq c/\epsilon^2$ for some conveniently large constant c (or alternatively, the algorithm is allowed to output "$d \leq c/\epsilon^2$" when d is too small.)

1. Consider an ideal hash function $h : [n] \to [k]$, where we assume that k is sufficiently large relative to $\dfrac{1}{\epsilon}$ (say $k \geq c_0/\epsilon^2$ for any convenient constant c_0 and sufficiently small $\epsilon > 0$). Let p be the probability that at none of the d distinct elements have hash value 1. We first observe that p is increasing in k and decreasing in d.

   (a) Show that for $k \leq d/(1 + \epsilon)$, we have

   $$p \leq (1 - c\epsilon)e^{-1}$$

   for some constant $c > 0$.

   (b) Show that for $k \geq (1 + \epsilon)d$, we have

   $$p \geq (1 + c\epsilon)e^{-1}$$

   for some constant $c > 0$.

2. As a warmup, consider the unit incremental case ($\Delta_j = 1$ for all j). For each integer $i \in \mathbb{Z}$, let $k_i = \left\lceil (1 + \epsilon)^i \right\rceil$, and let p_i be the probability that an ideal hash function $h_i : [n] \to [k_i]$ hashes none of the d distinct elements to 1. Suppose that for each i, we had a value $\widetilde{p}_i$ such that

   $$(1 - c\epsilon)p_i \leq \widetilde{p}_i \leq (1 + c\epsilon)p_i$$

   for some conveniently small constant $c > 0$. Show how to use these values $\{\widetilde{p}_i, i \in \mathbb{Z}\}$ to identify a value i such that $(1 + \epsilon)^{i-2} \leq d \leq (1 + \epsilon)^{i+2}$.

3. Show how to extend these ideas to obtain an $(1 \pm \epsilon)$-multiplicative approximation for F_0 with high probability in $O\!\left(\log^2 n/\epsilon^3\right)$ space in the incremental setting (not including space for the ideal hash functions).[1]

4. Let us now return to the strict turnstile settings, whether the Δ_j's might be negative, and the goal is to estimate the number of coordinates with strictly positive value at the end of the stream. In the previous two parts, we analyze an algorithm that simply counted the number of elements from the stream to hash to the first bucket. For the turnstile setting this won't work, because an element that is added and then deleted can still be hashed to the first bucket and effect our estimate. Modify the algorithm to work for turnstile streams, using the same amount of space.

---

[1]In a full implementation, the ideal hash functions can be replaced with one with k-wise independence for $k = O\!\left(\log(n)/\epsilon^2\right)$.