Algorithms for Big Data Fall 2020

Homework 3

Due: Friday 10/23/20, 5pm PT

- **Solving 2 of the following 3 problems** will lead to full credit. You may attempt all four problems, but the grading will be based on the 3 problems with the highest scores.
- You may work in groups of size 1-3. If you do, please hand-in a single assignment with everyone's names on it. It is strongly encouraged to type up the solutions in Latex.
- If the question asks to prove something, you must write out a formal mathematical proof.
- If the question involves analyzing an algorithm, you must formally explain the time and/or space usage, along with the approximation guarantees (when applicable).
- When you are asked to prove a bound, it suffices to prove it up to multiplicative constants, i.e., using $O(\cdot)$, $\Theta(\cdot)$, or $\Omega(\cdot)$ notation. No need to optimize (multiplicative) constants!
- You may use other resources, but you must cite them. If you use any external sources, you still must provide a complete and self-contained proof/result for the homework solution.

Problem 1: Tales of different norms

(a) Prove that the following two relationships hold for any vector $x \in \mathbb{R}^n$:

$$\|x\|_\infty \leq \|x\|_2 \leq \sqrt{n} \cdot \|x\|_\infty \qquad \text{and} \qquad \|x\|_2 \leq \|x\|_1 \leq \sqrt{n} \cdot \|x\|_2.$$

(b) Provide example vectors that satisfy each of the above four inequalities with an equality.

Problem 2: Streaming Sampling Revisited

Let $a_1, a_2, \ldots, a_n$ be a stream of n integers (not necessarily distinct) in the range $\{1, 2, \ldots, n\}$. The algorithm knows n up front. Each a_i will arrive one-by-one. The algorithm may compute something and update the storage, but then the value may not be accessed again (unless it is explicitly stored). The space is the maximum amount of memory used throughout. For each of these, you must prove that the algorithm works correctly, and provide a bound on the space.

(a) Assume that you know $A = \|\vec{a}\|_2^2 = \sum_{i=1}^n a_i^2$, the sum-of-squares of the values in the stream. Provide an algorithm using $O(\log n)$ space to sample an element a_i from the stream with probability exactly $p_i = \frac{a_i^2}{A}$.

(b) Now, assume that you **do not know** A ahead of time. Provide an algorithm using $O\left(\log^2 n\right)$ space that samples a_i from the stream with probability approximately $p_i = \frac{a_i^2}{A}$. More precisely, you should sample a_i with probability $\widetilde{p}_i$ satisfying $\frac{p_i}{4} \leq \widetilde{p}_i \leq 4p_i$ for all $i \in [n]$.

Hint: Use many different samples like the ones from (a) depending on the true value of A, and in parallel, compute A exactly so that you know which sample to use for the output.

(c) Improve your algorithm from (b). Now, given ε in the range $0 < \varepsilon < 1$, your sampling probabilities should satisfy $(1-\varepsilon)p_i \leq \widetilde{p}_i \leq (1+\varepsilon)p_i$.

Problem 3: Implementing a Sketching Algorithm

Implement and test one of the algorithms from the class, that is, choose one of the following options: (i) Morris for approximate counting, (ii) FM for distinct elements, or (iii) AMS for ℓ_2 estimation. Implement the algorithm and the + and ++ variants for the one you choose.

(a) Demonstrate/compare the performance of the three variants of the algorithm (normal, +, ++). Set the input size(s) to be large enough to see some difference in their performance.

(b) Provide results (in a table or plot, clearly labeled) for at least 2 different parameter settings (and list the parameters). Briefly discuss the results and any interesting observations.

(c) Provide the results of 10 repetitions for each of the two parameter settings (in a table or plot, clearly labeled), to demonstrate the probability of failure (and list the parameters). Briefly discuss the results and any interesting observations.

(d) Discuss how theory relates to practice, with quantitative results to back up your claims. For example, if the theory is pessimistic, then show that the results in practice are better with the same/improved parameters.