Problem Set 2

Sublinear Algorithms

Due Thursday, October 9

1. You are given $n \times n$ matrices A, B, C whose elements are from $\mathbb{Z}_2$ (i.e., the integers mod 2). Show a randomized algorithm running in $O(n^2)$ time which checks whether $AB = C$. The algorithm should output YES if $AB = C$ and output NO with at least $3/4$ probability if $AB \neq C$.

2. We saw a couple different norms for sparse recovery in our study of Count-Min and Count-Sketch, and we will see more in the future.

   We say that (k, C)-approximate ℓ_p/ℓ_q recovery of a vector x finds an $\overline{x}$ such that
   $$\|x - \overline{x}\|_p \leq C \min_{k\text{-sparse } x'} \|x - x'\|_q$$

   In this problem we study implications among the various guarantees. We say that (k, C) ℓ_p/ℓ_q recovery "implies" (k', C') $\ell_{p'}/\ell_{q'}$ recovery if, given any vector $\overline{x}$ satisfying the former, we can construct a vector $\overline{x}'$ satisfying the latter.

   Some of the parts below describe the transformation required to get the implication, while for others you need to identify the transformation. Suppose that $C > 1$ and $0 < \epsilon < 1$.

   (a) $(k, \epsilon/k)$ ℓ_∞/ℓ_1 recovery implies $(k, 1 + O(\epsilon))$ ℓ_1/ℓ_1 recovery by restricting to the largest k coordinates.

   (b) $(k, \sqrt{\epsilon/k})$ ℓ_∞/ℓ_2 recovery implies $(k, 1 + O(\epsilon))$ ℓ_2/ℓ_2 recovery by restricting to the largest $2k$ coordinates.

   (c) $(2k, C)$ ℓ_2/ℓ_2 recovery implies $(k, C/\sqrt{k})$ ℓ_2/ℓ_1 recovery.

   (d) $(k, C/\sqrt{k})$ ℓ_2/ℓ_1 recovery implies $(k, O(C))$ ℓ_1/ℓ_1 recovery.

3. Give a *deterministic* algorithm for the heavy-hitters problem.

   (a) Suppose a stream has m elements in $[n]$, and let x_i denote the number of elements equal to i. Construct a deterministic streaming algorithm using $O(\log(mn))$ bits of space that outputs a single integer j such that, if there exists an i such that $x_i > m/2$, then the algorithm outputs $j = i$.

   (b) Extend your result to use $O(k \log(mn))$ bits to output k numbers such that, for any i with $x_i > m/(k+1)$, then i is in the list being output.

   (c) [Optional] Give a deterministic algorithm that supports streaming insertions *and deletions* and gets a result similar to parts (a) or (b). In particular, use $O(k \log^c(mn))$ space to output $O(k)$ numbers such that for any i with $x_i > \|x\|_1/k$, i is in the output list.

4. The power dissipated by a resistor with resistance r going between two vertices of voltage v_1 and v_2 is $(v_1 - v_2)^2/r$. We can think about a resistor network as a multigraph, where each edge is associated with a resistance r_e. If we assign a set of voltages v_i to the vertices, then the total power dissipated is simply the sum over all resistors of the power dissipated by that resistor.

   Consider maintaining a resistor network under a stream with two kinds of updates:

   - $\textsc{Insert}((i,j), \text{“}tag\text{”}, r)$ which inserts a new resistor labeled “tag” of resistance r between i and j.
   - $\textsc{Delete}((i,j), \text{“}tag\text{”}, r)$ which deletes the resistor labeled “tag” of resistance r between i and j.

   (a) Give a streaming algorithm to maintain a sketch such that, for any set S of vertices, you can estimate the energy used by the circuit if the nodes of S are set to 1 volt and the rest are set to 0 volts. You should use $O(n\frac{1}{\epsilon^2}\log(1/\delta))$ words to get an $1 \pm \epsilon$ approximation with probability $1 - \delta$ for each S.

   (b) Extend this to estimate the energy used by the circuit for any assignment $v_1, \ldots, v_n$ of voltages to vertices, to error $1 \pm \epsilon$ with probability $1 - \delta$.

(c) Suppose now that we only allow insertions of resistors. Show how to use $O(\frac{n}{\epsilon^2} \log^c n)$ bits to have a sketch that with high probability can estimate the energy of *every* assignment of voltages to vertices up to $1 \pm \epsilon$ error.

Hint: You may use the fact that spectral sparsifiers exist. In particular, for any weighted graph G on n vertices, there is an efficient offline algorithm to construct a graph H on those vertices with only $O(\frac{n}{\epsilon^2} \log^c n)$ edges that matches the energy of *every* assignment of voltages to vertices up to $1 \pm \epsilon$ error.