Problem Set 2

Sublinear Algorithms

Due Tuesday, October 4

1. [**Deferred from problem set 1**] Recall the AMS sketch from class for $\|\cdot\|_2$ estimation: a random $m \times n$ matrix A with entries $A_{ij} \in \{\pm 1/\sqrt{m}\}$ is drawn for $m = O(1/\epsilon^2)$, and $\|x\|_2^2$ is estimated as $\|Ax\|_2^2$. With at least 3/4 probability, we had

$$(1-\epsilon)\|x\|_2^2 \le \|Ax\|_2^2 \le (1+\epsilon)\|x\|_2^2. \tag{1}$$

   (a) Consider the following matrix instead: for each $i \in [n]$, let the ith column of A have a single ± 1 in a random row, and 0s elsewhere. Because this matrix is sparse, it can be maintained under turnstile updates in *constant* time. Show that this A still satisfies (1) with 3/4 probability for $m = O(1/\epsilon^2)$.

   (b) Using constant-wise independent hash functions, show how to generate A using only $O(\log n)$ bits of randomness. Think about: how much independence do you need?

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

$$\|x - \overline{x}\|_p \le C \min_{k\text{-sparse } x'} \|x - x'\|_q$$

   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. 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.