A Strong Separation for Adversarially Robust ℓ_0 Estimation for Linear Sketches

Elena Gribelyuk[1] Honghao Lin[2] David P. Woodruff[2] Huacheng Yu[1] Samson Zhou[3]

Princeton University[1] Carnegie Mellon University[2] Texas A & M University[3]

Adversarially Robust Streaming

- Input: Elements of an underlying data set S, which arrives sequentially and *adversarially*
- Output: Evaluation (or approximation) of a given function
- Goal: Use space *sublinear* in the size m of the input S

- Adversarially Robust: "Future queries may depend on previous queries"
- Motivation: Database queries, adversarial ML

 1 4 2 3 

 1 4 2 1 4 

Distinct Elements

- Given a set S of m elements from $[n]$, let f_i be the frequency of element i. (How often it appears)
- Let F_0 be the frequency moment of the vector:

$$F_0 = |\{i : f_i \neq 0\}|$$

- Goal: Given a set S of m elements from $[n]$ and an accuracy parameter ε, output a $(1+\varepsilon)$-approximation to F_0
- Motivation: Traffic monitoring

Linear Sketch

- Algorithm maintains Ax for a matrix A throughout the stream
  - In the streaming model, the entries of A should be *poly(n)* bounded integers

- All insertion-deletion streaming algorithms on a sufficiently long stream might as well be linear sketches [LNW14, AHLW16]

Our Contribution

- There is a constant $\varepsilon = \Omega(1)$ such that any linear sketch that produces $(1+\varepsilon)$-approximation to ℓ_0 on an adversarial insertion-deletion stream using $r < n^c$ rows, for a constant $c > 0$, can be broken in $\tilde{O}(r^8)$ queries.

Attack Outline

- Adversary wants to gradually learn the sketching matrix

- Strategy:
  1. Iteratively identify the significant coordinates and set them to zero in all future queries
  2. After we have learned all such coordinates, the query algorithm must rely on the other coordinates, for which the sketch Ax only has "small" information

Pre-processing the Sketch Matrix

- The algorithm has access to linear sketch Ax
- Pre-process the matrix A into a larger matrix A' that separates the significant coordinates: add row e_i for each significant i
- WLOG, we can assume the algorithm actually use A' instead of A. Only gives the algorithm "more" information

$$\underset{A}{\begin{bmatrix} 0 & & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 999 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}} \longrightarrow \underset{A'}{\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}$$

- Iterative process: whenever there is a significant i, zero column i and add a row e_i .

- Resulting matrix A' is a combination of a sparse part S and a dense part D

$$A' = \begin{bmatrix} S \\ D \end{bmatrix}$$

- The row of S is one-sparse
- The D has no significant columns
- The columns of S and D are disjoint.

We Show only $O(rs\log n)$ rows added to A!

- How to quantify significant coordinates?
- i is significant if there exists:
  - $y \in \mathbb{R}^r$ such that $(\mathrm{FRAC}(y^\top A)_i)^2 \geq \frac{1}{s}\sum_i(\mathrm{FRAC}(y^\top A)_i)^2$

Interactive Fingerprinting Code Problem

- Make a number of adaptive queires to learn S.
- Query $q^t \in \{0,1\}^n$

- A set $S \subset [n]$ with $|S| = \ell$
- Observe q_S^t, output a^t
  - If $q_S^t = 1^\ell$, $a^t = 1$
  - If $q_S^t = 0^\ell$, $a^t = 0$

There exists an interactive fingerprinting code with queries $\tilde{O}(|S|^2)$ [SU15]

Overall Attack

1. Pre-process the matrix A into a matrix A' that is a combination of a sparse part S and a dense part D
2. Attack sparse part S using fingerprinting code
3. Argue dense part D doesn't help

Attacking the Dense Part

- Design a family of distributions $\mathcal{D}$ over $[-R, \dots, -1, 0, 1, \dots, R]$ with $R = \mathrm{poly}(r\, s \log n)$ such that:
  - For $D_p \in \mathcal{D}$ with $p \in [a,b]$, we have $\Pr_{X\sim D_p}[X=0] = 1-p$
  - For any $p, p' \in [a,b]$, the total variation distance between Dx_p and $Dx_{p'}$ is small, i.e., $\frac{1}{\mathrm{poly}(n)}$

$$Ax = \begin{bmatrix} S \\ D \end{bmatrix} x = \begin{bmatrix} Sx_S \\ Dx_D \end{bmatrix}$$

- Dx_D doesn't help to get the value of p, the only help part is Sx_S.