Sublinear Time and Space Algorithms 2022B – Lecture 5
Adversarially Robust Streaming, Flip-Number and Sparse-Dense Tradeoff*

Shay Sapir

1 Adversarially Robust Streaming Algorithms

Let us consider tracking algorithms, which track a function throughout the stream updates.

Tracking algorithms: Let $x^{(t)}$ be the frequency vector after t updates. An algorithm is said to be (ε, δ)-strong ℓ_2-tracking if w.p. $1-\delta$ it outputs an estimate $R_t \in (1 \pm \varepsilon)\|x^{(t)}\|_2$ for all $t \in [m]$.

Exer: Design a (ε, δ)-strong ℓ_2-tracking algorithm using $O(\epsilon^2 \log(n/\delta))$ words.

Hint: amplify AMS to success probability $1 - \delta/m$.

In the setting that we discussed until this point, the stream was fixed in advance. Let us now consider streams that may change according to the output of the algorithm (which corresponds to interactions with the environment).

Adaptive streams: For $t = 1, ..., m$,

1. Adversary chooses the next stream update σ_t.

2. Streaming algorithm process σ_t and outputs an estimate R_t.

3. Adversary observes R_t.

Algorithms in this model that satisfy the tracking property are called *Adversarially Robust*.

[Hardt-Woodruff, 2013]: Adversarially robust linear sketches for ℓ_2-norm have sketching dimension $\Omega(n)$.

Intuition: the adversary can use the observations to learn the sketching matrix A, and then query a non-zero vector x such that $Ax = 0$. The algorithm then must report the same output for x and $2x$.

*These notes summarize the material covered in class, usually skipping proofs, details, examples and so forth, and possibly adding some remarks, or pointers. The exercises are for self-practice and need not be handed in. In the interest of brevity, most references and credits were omitted.

2 Flip Number and Sketch Switching

Theorem 1 [Ben-Eliezer,Jayaram,Woodruff,Yogev,2020]: In insertion-only streams, there is an adversarially robust algorithm for ℓ_2-norm using $\tilde{O}(\varepsilon^{-3})$ bits of space.

Flip number [BJWY20]: For a function f and stream σ, the flip number $\lambda_\varepsilon(f,\sigma)$ is the size k of the largest subsequence $1 \le t_1 \le ... \le t_k \le m$ such that $f(x^{(t_i)}) \notin (1 \pm \varepsilon) f(x^{(t_{i+1})})$ for all $i \in [k-1]$. The flip number of f is $\lambda_\varepsilon(f) = \max_\sigma\{\lambda_\varepsilon(f,\sigma)\}$.

Lemma 2: In insertion-only streams, $\lambda_\varepsilon(\|\cdot\|_2) = O(\varepsilon^{-1}\log n)$.

Proof: Was seen in class.

Algorithm Sketch Switching:

1. Init: $\rho = 1, g = 0$.
2. run $\lambda = \lambda_{\varepsilon/8}(\|\cdot\|_2)$ ind. copies of an $(\varepsilon/8, \delta/\lambda)$-strong ℓ_2-tracking algorithm, denoted $A_1, ..., A_\lambda$.
3. update y as the current output of A_ρ.
4. if $g \notin (1 \pm \varepsilon/2)y$, then set $g = y$ and increment ρ.
5. output g.

Steps 2-5 happen for every stream update.

Space: $\tilde{O}(\epsilon^{-2}\lambda)$.

Analysis: We can assume the adversary is deterministic (Yao's principle). Why? Consider randomized adversary s.t. the algorithm fails w.p. p (over the randomness of alg + adversary). Then by an averaging argument, there is at least one choice for the randomness of the adversary for which the streaming algorithm fails w.p. p. Fix that randomness.

Let t_1 be the time-step when the "if condition" was fulfilled for ρ, hence from this time on, the algorithm "switches" to $A_{\rho+1}$. Let $y_1 = A_\rho(t_1)$. Consider the stream that would be if the output is always y_1. For this output, the adversary responds with a stream that is independent of $A_{\rho+1}$. Hence $A_{\rho+1}$ is correct w.p. $1 - \delta/\lambda$ until the next time that the "if condition" is fulfilled.

Exer: Assuming that the "active" A_ρ always output a $(1+\varepsilon/8)$-approximation of the ℓ_2-norm, prove that the "if condition" is fulfilled at most $\lambda_{\varepsilon/8}(\|\cdot\|_2)$ times.

3 Streams with deletions

λ can be as large as m if there are many insertions + deletions to the same coordinate, which corresponds to a sparse vector.

Theorem 3: There is an adversarially robust streaming algorithm for ℓ_2-norm using $\tilde{O}_\varepsilon(m^{2/3})$ bits of space.[1]

This is based on [Ben-Eliezer,Eden,Onak, 2022], who achieved a better bound of $\tilde{O}_\varepsilon(m^{2/5})$ bits of space. They use differential privacy for this improvement, which was not discussed in class. Finding the tight bounds is an open question.

Sparse Recovery:

Input: x is a k-sparse vector.

Goal: recover x exactly using a linear sketch of dimension $\tilde{O}(k)$.

Exer: Show that CountMin/CountSketch with $\tilde{O}(k)$ buckets solve Sparse Recovery.

High level algorithm of Theorem 3:

- If x is T-sparse, then maintain it explicitly.
- If x is T-dense (not sparse), then use the Sketch Switching algorithm.
- Use Sparse Recovery algorithm to recover x when it becomes sparse (after being dense).

Lemma: The flip number for the ℓ_2-norm of T-dense vectors is bounded by $O_\varepsilon(m/\sqrt{T})$.

Proof: was seen in class.

Space: the Sparse Recovery algorithm uses $\tilde{O}(T)$ bits, and the Sketch Switching uses $\tilde{O}_\varepsilon(m/\sqrt{T})$ bits of space. Pick $T = m^{2/3}$, resulting in the desired space bound.

One may need to track the number of non-zeros in x in order to decide if the regime is sparse or dense. This was not discussed in class, and can be done using a Distinct Elements algorithm for streams with deletions, and by a slight change of the algorithm: make it such that if x is $2T$-dense then we use Sketch Switching, and if $\|x\|_0 \in [T, 2T]$, then either maintain it explicitly or use Sketch Switching.

[1] The notation $\tilde{O}_\varepsilon(\cdot)$ hides multiplicative factor $\text{poly}(\varepsilon, \log n)$.