Lecture 3 :- Concentration and saving memory (II)

So far :- $X_i = I[G_i \text{ is "bad"}]$

$M :=$ Median of $(G_1 \, . \, . \, . \, . \, G_k)$, G_i independent.

$E[X_i] = p \leq 1/8$, and

$$\Pr[|M - F_2| \geq \varepsilon F_2] \leq \Pr\left[\sum_{i=1}^{k} X_i \geq \frac{k}{2}\right]$$

$$\leq \Pr\left[\sum_{i=1}^{k} (X_i - p) \geq k\left(\tfrac{1}{2} - p\right)\right]$$

$$\leq \exp\left(-2k \cdot \left(\tfrac{1}{2} - p\right)^2\right)$$

$$\leq \exp\left(-\frac{9k}{32}\right) \quad \text{—} (*)$$

(From the Chernoff-Hoeffding bound discussed above.)

We wanted $\Pr[|M - F_2| > \varepsilon F_2] \leq \delta$.

From $(*)$ this requires $k \geq \left\lceil \frac{32}{9} \log\left(\frac{1}{\delta}\right) \right\rceil$.

What we gained :- Only $O(\log(1/\delta))$ samples of the G_i are needed to drive the error probability to below δ!

Summing up :- Fix $\varepsilon > 0$ (accuracy parameter)
$\delta > 0$ (error probability tolerance).

1. If our stream is over D dimensions:

$$Y_{ij} = \left(\sum_{\ell=1}^{D} f_\ell \, u_{\ell,(i,j)} \right)^2 \qquad \left[f_\ell = \text{Frequency of the } \ell\text{th element in the stream.} \right]$$

Requirement:- For a fixed (i,j) pair.

the collection $(u_{\ell,(i,j)})_{\ell=1}^{D}$ of ℓ $\{+1,-1\}$ r.v.s with expectation 0 is 4-wise independent.

2. $$G_i := \frac{1}{s} \sum_{j=1}^{s} Y_{i,j} \qquad s \geq \left\lceil \frac{16}{\varepsilon^2} \right\rceil .$$

We needed $(Y_{i,j})_{j=1}^{s}$ (fixed i) to be at least pairwise independent. — 1

3. $M = \text{Median}(G_1, \ldots G_k)$, $k \geq \left\lceil \frac{32}{9} \log\left(\frac{1}{\delta}\right) \right\rceil$

We need $G_1 \ldots G_k$ to be independent. — 2

Both 1 and 2 are satisfied if we assume the the vectors $(u_{\ell,(i,j)})_{\ell=1}^{D}$, as (i,j) vary, are <u>independent</u> $\left[O\left(\frac{1}{\varepsilon^2} \log\left(\frac{1}{\delta}\right)\right) \text{ vectors} \right]$

<u>Thus</u>:-

We need storage for $sk = O\left(\frac{1}{\varepsilon^2} \log\left(\frac{1}{\delta}\right)\right)$ counters. Y_{ij}.

But we make <u>all</u> the $u_{\ell,(i,j)}$ <u>independent</u> we will have to store $\Omega\left(D \cdot \frac{1}{\varepsilon^2} \log\left(\frac{1}{\delta}\right)\right)$ random bits.

But the whole point of this exercise was to <u>not</u> need storage that grows linearly in terms of the <u>alphabet size</u> of the stream.

- With $\Omega(D)$ storage, we might just use an associative array (Hash Map / Balanced binary tree map).

So, how do we get rid of this linear dependence on D?

We need to use that each of the sk arrays only needs to have 4-wise independent (and not necessarily fully independent!!) signs.

<u>Limited independence</u>:- We only want to store a <u>small</u> number of truly random bits. (called the 'seed') such that a <u>deterministic algorithm</u> working with this seed can produce a much larger sequence of bits, which look (say) pairwise independent to an "outside observer" who has not seen the seed.

<u>A construction</u>:- Store as seed $s = s_0 s_1 \ldots s_n$ sampled uniformly at random from $\{0,1\}^{n+1}$.

Algorithm :- When the observer asks for bit indexed i where $0 \leq i \leq 2^n - 1$, the alogrithm outputs

$$A(s, i) := s_0 + \bigoplus_{j=1}^{n} \left(s_j \cdot I[j\text{th bit of } i \text{ is } 1]\right)$$

Claim :- $\forall\ i, j,\ i \neq j,\ 0 \leq i, j \leq 2^n - 1,\ \forall\ \alpha, \beta \in \{0,1\}$,

$$\Pr_{s \underset{u.a.r.}{\sim} \{0,1\}^n} [A(s, i) = \alpha \text{ and } A(s, j) = \beta]$$

$$= \Pr_{s \underset{uar}{\sim} \{0,1\}^n} [A(s, i) = \alpha] \cdot \Pr_{s \underset{uar}{\sim} \{0,1\}^n} [A(s, j) = \beta]$$

$$= \frac{1}{4}.$$ [Chor-Goldreich]

Second construction: k-wise [BCH, Reed-Solomon]
independence random variables.

Fix a finite field $\mathbb{F}$:- A field is just a set with a 0 element, a 1 element (different from the 0 element) commutative addition and multiplication, with multiplication distributing over addition, in which every element (except 0) has a unique multiplicative inverse and every element has an additive inverse.

Examples :- $\mathbb{C}$, $\mathbb{R}$, $\mathbb{Q}$.

Non-examples :- $\mathbb{Z}$,

A field with finitely many elements is a finite field. [Any finite field has size

p^d for some p prime (including 2) and d a positive integer. Further, there is exactly one finite field of each such size]

One can define polynomials over a finite field $\mathbb{F}$

$$p(x) = \sum_{i=0}^{d} \alpha_i x^i, \quad \alpha_i \in \mathbb{F}.$$

Fact:-

- If p, q are polynomials of degree d over $\mathbb{F}$, and $p(\beta_i) = q(\beta_i)$ for at least $d+1$ distinct $\beta_i \in \mathbb{F}$ then $p = q$.

- If $\gamma_0 \dots \gamma_d \in \mathbb{F}$ are distinct values and $\beta_0, \dots \beta_d \in \mathbb{F}$ are arbitrary values then there always exist a polynomial p s.t.
$$p(\gamma_i) = \beta_i, \quad 0 \leq i \leq d.$$

This polynomial is unique by the previous fact.

Construction

k-wise independent r.v. taking values in $\mathbb{F}$. $k \leq \log |\mathbb{F}|$

- Pick $\alpha_0, \dots \alpha_{k-1}$ u.a.r. from $\mathbb{F}$ ["Seed"].

- Outside observers inputs $\gamma \in \mathbb{F}$. The algorithm outputs:
$$A(\vec{\alpha}, \gamma) = \sum_{i=0}^{k-1} \alpha_i \gamma^i$$

<u>Claim</u>:- Let $\gamma_1, \gamma_2, \ldots \gamma_k \in \mathbb{F}$ be <u>distinct</u>, and $\beta_1, \beta_2 \ldots \beta_k \in \mathbb{F}$ arbitrary. Then.

$$\Pr_{\vec{\alpha} \in_{u.a.r} \mathbb{F}}\left[\bigwedge_{j=1}^{k}\left[A(\vec{\alpha}, \gamma_j) = \beta_j\right]\right] = \prod_{j=1}^{k} \Pr\left[A(\vec{\alpha}, \gamma_j) = \beta_j\right]$$

$$= \frac{1}{|\mathbb{F}|^k}.$$

- Just the previous fact in disguise.!!

(There is a <u>unique</u> $\vec{\alpha} \in \mathbb{F}^k$ for which $A(\vec{\alpha}, \gamma_j) = \beta_j$ for k distinct γ_j.)

So when $\vec{\alpha} \in \mathbb{F}^k$ is chosen u.a.r. the random variables $(A(\vec{\alpha}, \gamma))_{\gamma \in \mathbb{F}}$ are $\mathbb{F}$-valued k-wise independent random variable.

<u>Requirement</u>:- D random-variables
4- wise independent
$\{+1, -1\}$- valued, with uniform distribution

An option: choose $\mathbb{F} = \mathbb{F}_p$ where p is the smallest prime satisfying $p > D$.

<u>Next option</u>:- choose d smallest s.t. $2^d > D$ and take $\mathbb{F} = \mathbb{F}_{2^d}$.