maximal load: $O(\log n / \log\log n)$ } aim
Space required: $O(\log^2 n / \log\log n)$

Construction

each function described in $O(\log n \log\log n)$

evaluated in $O(\log n \log\log n)$

IDEA: concatenate output of $O(\log\log n)$ functions which are gradually more independent. Each function f is described using d functions $h_1, \dots, h_d$

$$f(x) = \underbrace{h_1(x)}_{\text{binary string}} \circ h_2(x) \circ \cdots \underbrace{\circ}_{\text{concatenate}} h_d(x)$$

Gradually more independent

k-wise
k increase

h_1: $O(1)$-wise indep.

h_2: $O(h_1)$-wise indep.

...

h_d: $O(\log n / \log\log n)$-wise indep.

Note: ① output length decreases at the same time

$h_1 : \Omega(\log n) \Rightarrow h_d : O(\log\log n)$

② each of $h_1, \dots, h_d$ could be described/computed in $O(\log n)$ bits/time

Definitions

$[n] \Rightarrow \{1, \dots, n\}$

$U_n \Rightarrow$ Uniform distribution over set $\{0,1\}^n$

$x \leftarrow X \Rightarrow$ sample x from X for a r.v. X.

$x \in S \Rightarrow$ sample x uniformly from finite set S.

$SD(X, Y) \Rightarrow$ statistical distance between two r.v. over finite domain Ω

$$= \frac{1}{2} \sum_{\omega \in \Omega} |\Pr[X=\omega] - \Pr[Y=\omega]|$$

$x \circ y \Rightarrow$ concatenate x, y bit string

unit cost RAM model

$\Rightarrow$ elements are taken from a universe of size u and each element can be stored using $\omega = O(\log u)$ bits.

<u>k-wise δ-dependent</u>

For a family of $f: [n] \to [v]$, it is k-__ δ-___ iff

$$SD(X, Y) \le \delta$$

where X = distribution $(f(x_1), f(x_2), \dots, f(x_k))$ for any distinct $x_1, \dots x_k \in [u]$

Y = uniform distribution over $[v]^k$

Describe Space: $O(k \max\{\log u, \log v\})$ bits

Evaluation Time: $O(k)$

<u>ε-biased Distribution</u> [NN93]

r.v.s $X_1, X_2, \dots, X_n$ over $\{0,1\}$ is ε-biased if for any $S \ne \emptyset \subseteq [n]$,

$$|\Pr(\oplus_{i \in S} X_i = 1) - \Pr(\oplus_{i \in S} X_i = 0)| \le \varepsilon$$

↑ XOR operation

[AGH⁺92, Sec. 5] constructs an ε-biased distribution over $\{0,1\}^n$ where each point could be specified using $O(\log(n/\varepsilon))$ bits where each bit could be calculated using $O(\log(n/\varepsilon))$ times.

In RAM, t bits could be calculated in time $O(\log(n/\varepsilon)\, t)$

$\boxed{\text{may need proof}}$

min-entropy

for a r.v. X, its min-entropy is

$$H_\infty(X) = -\log(\max_x \Pr[X=x])$$

negative log of most probability of $X=x$.

k-source

k-source is a r.v. X with its min_entropy $H_\infty(X) > k$.

(T, k)-block source

r.v. $X = (X_1, \dots, X_T)$, for any $i \in [T]$

$$H_\infty(X_i | X_1 = x_1, \dots, X_{i-1} = x_{i-1}) \geq k$$

(T, k, ε)-block source

r.v. $X = (X_1, \dots, X_T)$, for any $i \in [T]$

$$\Pr_{(x_1, \dots, x_{i-1}) \leftarrow (X_1, \dots, X_{i-1})} [H_\infty(X_i | X_1 = x_1, \dots, X_{i-1} = x_{i-1}) \geq k] \geq 1 - \varepsilon$$

Pr of X is (T, k)-block source $\geq 1 - \varepsilon$

Theorem

Corollary 2.1

Fact

for any k an ε-biased distribution is also k-wise δ-dependent for $\delta = \varepsilon 2^{k/2}$

For any u, v, $v = 2^i$, there exists a family of k-wise δ-dependent function $f: [u] \to [v]$ described in $O(\log u + k\log v + \log(1/\delta))$ bits and calculated in $O(\log u + k\log v + \log(1/\delta))$ in RAM.

check AGH92 for proof.

Corollary 2.2

Let $X_1, \dots, X_n \in \{0,1\}$ be $2k$-wise δ-dependent r.v. for $k \in \mathbb{N}$, $0 \le \delta < 1$.

Let $X = \sum X_i$ and $\mu = E[X]$. Then, for any $t > 0$,

$$\Pr[|X - \mu| > t] \le 2\left(\frac{2nk}{t^2}\right)^k + \delta\left(\frac{n}{t}\right)^{2k}.$$

How X diverge from μ. proceed in paper using Markov Ineq. Lemma 2.2 from [BR94]

may not be used by our construction.

Lemma 2.4 [GW97] ← may need a proof (not sure from 4.1/4.2 main result)

r.v. $X_1 \in [0,1]^{n_1}$, $X_2 \in [0,1]^{n_2}$, $H_\infty(X_1 X_2) \geqslant n_1 + n_2 - \Delta$

① $H_\infty(X_1) \geqslant n_1 - \Delta$

② for any $\varepsilon > 0$,

$$\Pr_{x_1 \leftarrow X_1}[H_\infty(X_2 | X_1 = x_1) < n_2 - \Delta - \log(1/\varepsilon)] < \varepsilon$$

Corollary 2.5.

Any $X = (X_1, \dots, X_T)$ over $(\{0,1\}^n)^T$, $H_\infty X \geqslant Tn - \Delta$

is a $(T, n-\Delta-\log(1/\varepsilon), \varepsilon)$-block source for any $\varepsilon > 0$.

def of (T, k, ε)-block source
and lemma 2.4.

Theorem 3.1

any constant $c > 0$, integer n, $u = \text{poly}(n)$, there exists a family F of $f: [u] \to [n]$ that:

① describe using $O(\log n \log\log n)$ bits

② $f(x)$ can be computed in $O(\log n \log\log n)$ time

③ $\exists \gamma > 0$ such that for any $S \subseteq [n]$

$$\Pr_{f \in F}\left[\max_{i \in [n]} |f^{-1}(i) \cap S| \le \frac{\gamma \log n}{\log\log n}\right] > 1 - \frac{1}{n^c}$$

maximum load $= O\left(\frac{\gamma \log n}{\log\log n}\right)$ with high probability.

Construction

assume $n = 2^k$

let $d = O(\log\log n)$, and for every $i \in [d]$, let H_i be a family of k_i-wise δ-dependent functions $[u] \to \{0,1\}^{l_i}$, where:

① $n_0 = n$ $\quad n_i = \frac{n_{i-1}}{2^{l_i}}$ for every $i \in [d]$

$n_1 = \frac{n}{2^{l_1}}$ $\quad n_2 = \frac{n_1}{2^{l_2}} = \frac{n}{2^{l_1+l_2}}$ $\dots$ $n_d = \frac{n}{2^{\sum l_i}}$

② $l_i = \left\lfloor \frac{\log n_{i-1}}{4} \right\rfloor$ for every $i \in [d-1]$, $l_d = \log n - \sum_{i=1}^{d-1} l_i$

$l_i = \left\lfloor \frac{\log n_{i-1}}{4} \right\rfloor = \left\lfloor \frac{\log n - \sum l_i}{4} \right\rfloor$ for $i \in [d-1]$

$l_d = \log n - \sum_{i=1}^{d-1} l_i$

③ $k_i l_i = \Theta(\log n)$ for every $i \in [d-1]$, $k_d = \Theta(\log n / \log\log n)$

$k_i l_i = k_i \left\lfloor \frac{\log n - \sum l_i}{4} \right\rfloor$

④ $f = \text{poly}(\frac{1}{n})$

By corollary 2.1, we have a family H_i:

need detail.

h_i is represented using $\underline{O(\log u + k_i l_i + \log(1/\delta)) = \log n}$ space/time

We define our function family F as

$$f(x) = h_1(x) \circ h_2(x) \circ \cdots \circ h_d(x)$$

for $h_i \in H_i$

We visualize the construction as a reversed tree with d layers.

Layer 0 has 1 bin with all balls.

hashed using h_1 → Layer 1 has 2^{l_1} intermediate bins.

hashed using h_2 → Layer 2 has 2^{l_2} intermediate bins.

. . . . . .

hashed using h_d → Layer d has n final bins

Lemma 3.2

For any $i = \{0, \ldots, d-2\}$, $\alpha = \Omega(1/\log\log n)$, $0 < \alpha_i < 1$ and set $S_i \subseteq [u]$ of size at most $(1+\alpha_i)n_i$,

$$\Pr_{h_{i+1} \in H_{i+1}}\left[\max_{y \in [0,1]^{l_{i+1}}} |h_{i+1}^{-1}(y) \cap S_i| \leq (1+\alpha)(1+\alpha_i)n_{i+1}\right] \geq 1 - \frac{1}{n^{c+1}}$$

number of balls in any bin of layer $i+1 \leq (1+\Omega(\frac{1}{\log\log n}))(1+\alpha_i)n_{i+1}$

with high prob.

$\boxed{\text{Application}}$

① storing elements using linear probing

② augmenting k-wise independence function using our construction without affect the space/time requirement

need some evidence or proof.

Proof For Lemma 3.2

- Fix $y \in \{0,1\}^{L_{i+1}}$, let $X = |h_{i+1}^{-1}(y) \cap S_i|$
- assume WLOG, $|S_i| \geq \lfloor (1+\alpha_i) n_i \rfloor$ or we could add dummy elements to enlarge a smaller S_i. ? How.
- Then, X = sum of $|S_i|$ indicator random variables that are k_{i+1}-wise δ-dependent.

new
$$\begin{cases} X_j = 1 \text{ if element } j \in S_i \text{ is hashed into } h_{i+1} \\ X = \sum_j^{|S_i|} X_j \qquad E[X] = \sum_j^{|S_i|} E[X_j] = \sum_j^{|S_i|} \frac{1}{2^{L_{i+1}}} = \frac{|S_i|}{2^{L_{i+1}}} \end{cases}$$

if uniform, total $2^{L_{i+1}}$ bins

- Since X = Sum of $|S_i|$ k_{i+1}-wise δ-dependent r.v., we could apply Lemma 2.2:

$$k = \frac{k_{i+1}}{2}, \quad \mu = E[X] = |S_i|/2^{L_{i+1}}$$

$$\Rightarrow \Pr[X > (1+\alpha)\mu] \leq 2\left(\frac{|S_i| k_{i+1}}{(\alpha\mu)^2}\right)^{k_{i+1}/2} + \delta\left(\frac{|S_i|}{\alpha\mu}\right)^{k_{i+1}/2}$$

replace μ by $|S_i|/2^{L_{i+1}}$

$$= 2\left(\frac{|S_i| k_{i+1}}{|S_i|^2/2^{2L_{i+1}}\alpha^2}\right)^{k_{i+1}/2} + \delta\left(\frac{|S_i|}{|S_i|/2^{L_{i+1}}\alpha}\right)^{k_{i+1}/2}$$

$$= 2\left(\frac{2^{2L_{i+1}} k_{i+1}}{|S_i|\alpha^2}\right)^{k_{i+1}/2} + \delta\left(\frac{2^{L_{i+1}}}{\alpha}\right)^{k_{i+1}/2}$$

- Now, we will upper bound each section

1) Notice that, in our construction, ① $L_{i+1} = \left\lfloor \frac{\log n_i}{4} \right\rfloor \leq \frac{\log(n_i)}{4}$

and ② $|S_i| \geq (1+\alpha_i) n_i - 1 \geq n_i$ ③ $\alpha = \Omega(1/\log\log n)$

$$2\left(\frac{2^{2L_{i+1}} k_{i+1}}{\alpha^2 |S_i|}\right)^{k_{i+1}/2} \leq 2\left(\frac{2^{\frac{\log(n_i)}{2}} k_{i+1}}{\alpha^2 |S_i|}\right)^{k_{i+1}/2} \quad \text{since ①}$$

$$= 2\left(\frac{\sqrt{n_i}\, k_{i+1}}{\alpha^2 |S_i|}\right)^{k_{i+1}/2}$$

$$\leq 2\left(\frac{\sqrt{n_i}\, k_{i+1}}{\alpha^2 n_i}\right)^{k_{i+1}/2} \quad \text{Since ②} \Rightarrow \frac{1}{|S_i|} < \frac{1}{n_i}.$$

$$2\left(\frac{k_{i+1}}{\alpha^2}\right)^{k_{i+1}/2}$$

$$= 2\left(\frac{k_{i+1}}{\alpha^2 \sqrt{n_i}}\right)^{k_{i+1}/2}$$

$$\leq 2\left(\frac{k_{i+1}}{\alpha^2 2^{2L_{i+1}}}\right)^{k_{i+1}/2} \quad \text{Since } 2^{2L_{i+1}} \leq 2^{\frac{\log(n_i)}{2}} = \sqrt{n_i} \Rightarrow \frac{1}{\sqrt{n_i}} \leq \frac{1}{2^{2L_{i+1}}}$$

$$= 2\left(\frac{k_{i+1}^{k_{i+1}/2}}{\alpha^{k_{i+1}} 2^{L_{i+1}k_{i+1}}}\right)$$

$$= 2\left(\frac{k_{i+1}}{\alpha^{k_{i+1}} 2^{\sqrt{2}\log n}}\right)^{k_{i+1}/2}$$ by construction.

$$= 2\left(\frac{k_{i+1}}{\alpha^{k_{i+1}} 2^{\log n^c}}\right)^{k_{i+1}/2}$$ set $L_i k_i = \log n^c$

$k_{i+1} = \frac{k_i L_i}{L_i} = \frac{4\log n^c}{\log n_i} = \frac{4c\log n}{\log n_i}$

$\alpha = \sqrt{2}\, n^{\frac{1}{\sqrt{k}}} = \Omega\left(\frac{1}{\log\log n}\right)$

$$= 2\left(\frac{k_{i+1}}{\alpha^2}\right)^{k_{i+1}/2} \cdot \frac{1}{n^c}$$

$$= 2\left(\frac{k_{i+1}}{\sqrt{2}\, n^{\frac{1}{k_{i+1}}}}\right)^{k_{i+1}/2} \cdot \frac{1}{n^c}$$

$$= 2\left(\frac{1}{n^2}\right)^{\frac{2c\log n}{\log n_i}} \cdot \frac{1}{n^c}$$

$$\leq 2\,\frac{1}{n^2} \cdot \frac{1}{n^c}$$ since $\frac{\log n}{\log n_i} > 1$

$$2\left(\frac{2^{2L_{i+1}} k_{i+1}}{\alpha^2 |S_i|}\right)^{k_{i+1}/2} \leq 2\,\frac{1}{n^{c+2}}$$ $\Leftarrow$ result.

2) Notice that $f = \mathrm{poly}(\frac{1}{n})$, then

$$f\left(\frac{2^{L_{i+1}}}{\alpha}\right)^{k_{i+1}} = f\,\frac{2^{L_{i+1}k_{i+1}}}{\alpha^{k_{i+1}}}$$

$$= f\,\frac{2^{\log n^c}}{\alpha^{k_{i+1}}}$$

$$\leq f\, n^c$$

$$\leq \frac{1}{2n^{2c+2}} \cdot n^c$$ $f = \frac{1}{2n^{2c+2}}$

$$\leq \frac{1}{2n^{c+2}}$$

Thus, we get

$$\Pr[X > (1+\alpha)(1+\alpha_i)n_{i+1}] = \Pr\left[X > (1+\alpha)(1+\alpha_i)\frac{n_i}{2^{L_{i+1}}}\right]$$ since $n_{i+1} = \frac{n_i}{2^{L_{i+1}}}$

$$\leq \Pr\left[X > (1+\alpha)\frac{|S_i|}{2^{L_{i+1}}}\right]$$ since $|S_i| \leq (1+\alpha_i)n_i$

$$= \Pr[X > (1+\alpha)\mu]$$ since $\mu = \frac{|S_i|}{2^{L_{i+1}}}$

$$\leq \frac{1}{2n^{c+2}} + \frac{1}{2n^{c+2}}$$ by upper bound.

$$= \frac{1}{n^{c+2}}$$

We have at most n different y since $y = \{0,1\}^{L_{i+1}} \Rightarrow 2^{L_{i+1}} \leq 2^{\log n_i - 1} \leq 2^{\log n} = n$.

$\Rightarrow$ union bound over y $\frac{1}{n^{c+1}}$ and subtract from 1

$1 - \frac{1}{n^{c+1}}$

$\square$

Proof for Theorem 3.1

Description Length / Evaluation time

ith layer $\Rightarrow h_i$ is k_i-wise δ-dependent function: $[u] \Rightarrow \{0,1\}^{l_i}$

$v = 2^{l_i}$, $u = poly(n)$, $\delta = poly(\frac{1}{n})$

By Corollary 2.1, we know there exists a family of k_i-wise δ-dependent functions for:

$$space = O(\log u + k \log v + \log(1/\delta))$$
$$= O(\log u + k_i \log 2^{l_i} \log(\frac{1}{poly(\frac{1}{n})}))$$
$$= O(\log u + k_i l_i \log(\frac{1}{\delta}))$$
$$= O(\log n)$$

$time = O(\log u + k \log v + \log(1/\delta)) = O(\log n)$ also.

Maximum Load

- Fix a set $S \subseteq [u]$ of size n. We inductively argue that:

for every level $i \in \{0, \dots, d-1\}$, with probability $\geq 1 - \frac{i}{n^{c+1}}$, the maximum load in level i is at most $(1+\alpha)^i n_i$ per bin.

for $\alpha = \Omega(1/\log\log n)$

$\boxed{\text{Base Case}}$ $i=0$

single bin with $n_0 = n = (1+\alpha)^0 n = (1+\alpha)^0 n_0$ ✓

$\boxed{\text{Inductive Hypothesis}}$

claim holds for level i.

$\boxed{\text{Inductive Case}}$

Lemma 3.2 with $(1+\alpha_i) = (1+\alpha)^i$: number of balls in any bin of layer $i+1$ $\leq (1+\alpha)^{i+1} n_{i+1}$ with prob $\geq 1 - \frac{1}{n^{c+1}}$. Union bound over all $n_i \leq n$ bins, we could know that with prob $\geq 1 - \frac{i}{n^{c+1}} \geq 1 - \frac{i+1}{n^{c+1}}$, the maximum load in level $i+1$ is $(1+\alpha)^{i+1} n_{i+1}$.

Another way:

at most n bins.

$\text{Prob}(\text{number of balls in one bin of layer } i+1 > (1+\alpha)^{i+1} n_{i+1}) \le \frac{1}{n^{c+1}}$

Union bound

$\Rightarrow \text{Prob}(\text{there exists a bin of layer } i+1 > (1+\alpha)^{i+1} n_{i+1}) \le \frac{n}{n^{c+1}}$

$\Rightarrow \text{Prob}(\text{there doesn't exist a bin of layer } i+1 \le (1+\alpha)^{i+1} n_{i+1}) \ge 1 - \frac{n}{n^{c+1}}$

$\ge 1 - \frac{i+1}{n^{c+1}}$

$\Rightarrow \text{Prob}(\text{maximum load of layer } i+1 = (1+\alpha)^{i+1} n_{i+1}) \ge 1 - \frac{i+1}{n^{c+1}}$

Thus, claim holds in inductive case.

- Now, we want to upper bound n_{d-1}, the expected load in layer $d-1$

By the induction claim, we know, with probability at least $1-(d-1)/n^{c+1}$, the maximum load on level $d-1$ is $(1+\alpha)^{d-1} n_{d-1} \le 2 n_{d-1}$ for $d = O(\log\log n)$.

For every $i \in [d-1]$,

$$l_i \ge c \log n_{i-1})/4 \quad -1$$

$$\Rightarrow n_i = n_{i-1}/2^{l_i} \le 2 n_{i-1}^{3/4} \quad \text{since } \frac{1}{2^{l_i}} \le \frac{1}{2^{\log n_{i-1}/4}} = \frac{1}{n_{i-1}^{3/4}}$$

$$\Rightarrow n_{i-1} \le 2 n_{i-2}^{3/4} \Rightarrow n_i \le 2(2 n_{i-2})^{3/4} = 2^{1+3/4} n_{i-2}^{(3/4)^2}$$

$$\Rightarrow n_{i-1} \le 2^{\sum_{j=0}^{i-1} (3/4)^j} n^{(3/4)^i} \le 2^4 n^{(3/4)^i} = 16 n^{(3/4)^i}$$

Thus, for an appropriate choice of $d = O(\log\log n)$ it holds

$$n_{d-1} \le \log n$$

For example, $d = \log_{3/4}\left(\frac{\log \frac{\log n}{16}}{\log n}\right) \in O(\log\log n)$

$$n_{d-1} \le 16 n^{(3/4)^d} = 16 n^{(3/4)^{\log_{3/4}\left(\frac{\log \frac{\log n}{16}}{\log n}\right)}}$$

$$= 16 n^{\frac{\log \frac{\log n}{16}}{\log n}}$$

$$= 16 n^{\log_n\left(\frac{\log n}{16}\right)}$$

$$= 16 \frac{\log n}{16}$$

$$= \log n.$$

Also, by definition of n_i, we know $n_i = \frac{n}{2^{\sum l_i}}$, $l_d = \log n - \sum_i^{d-1} l_i = \log n_{d-1}$

Thus, we know that

since $L_d = \log n_{d-1} \Rightarrow 2^{L_d} = 2^{\log n_{d-1}} = n_{d-1}$ bins.

the maximum load on level $d-1$ is $(1+\alpha)^{d-1} n_{d-1} \le 2 n_{d-1} \le 2\log n$.

These elements are hashed into n_{d-1} bins using the function h_d which is k_d-wise δ-dependent, where $k_d = \Omega(\log n / \log\log n)$. Therefore, the probability that any $t = \gamma \log n / \log\log n < k_d$ elements from level $d-1$ are hashed into any specific bin in level d is at most.

$$\underbrace{\binom{2n_{d-1}}{t}}_{\text{all comb of } t \text{ within } 2n_{d-1}}\Big(\big(\underbrace{\tfrac{1}{n_{d-1}}}_{\text{one bin}}\big)^t + \delta\Big) \le \left(\frac{2en_{d-1}}{t}\right)^t \left(\left(\frac{1}{n_{d-1}}\right)^t + \delta\right)$$

$$= \left(\frac{2e}{t}\right)^t + \delta\left(\frac{2en_{d-1}}{t}\right)^t$$

$$= \left(\frac{2e\log\log n}{\gamma \log n}\right)^{\frac{\gamma\log n}{\log\log n}} + \delta\left(\frac{2en_{d-1}\log\log n}{\gamma\log n}\right)^{\frac{\gamma\log n}{\log\log n}}$$

since $n_{d-1} < \log n \longrightarrow$
$$\le \left(\frac{2e\log\log n}{\gamma\log n}\right)^{\frac{\gamma\log n}{\log\log n}} + \delta\left(\frac{2e\log\log n}{\gamma}\right)^{\frac{\gamma\log n}{\log\log n}}$$

detail(?) $\longrightarrow$
$$\le \frac{1}{2n^{c+3}} + \frac{1}{2n^{c+3}}$$

$$= \frac{1}{n^{c+3}}$$

for $t = \frac{\gamma\log n}{\log\log n}$ and $\delta = \mathrm{poly}(\frac{1}{n})$

This holds for any pair of bins in level d and $d-1$, union over them implies that:

$$\Pr(\text{a bin with more than } t \text{ elements}) \le \frac{1}{n^{c+1}}$$ since $2^{L_{d-1}} < 2^{L_d} = n_{d-1} \le n$

This implies that

Union bound $\Rightarrow$ $\Pr(\text{there exists a bin with more than } t \text{ elements}) \le \frac{n}{n^{c+1}}$

$\Pr(\text{there doesn't exist a bin with more than } t \text{ elements}) \ge 1 - \frac{n}{n^{c+1}}$

$\ge 1 - \frac{d}{n^{c+1}}$

$\Rightarrow \Pr(\text{maximum load is } t) \ge 1 - \frac{d}{n^{c+1}} > 1 - \frac{1}{n^c}$

$\boxed{}$