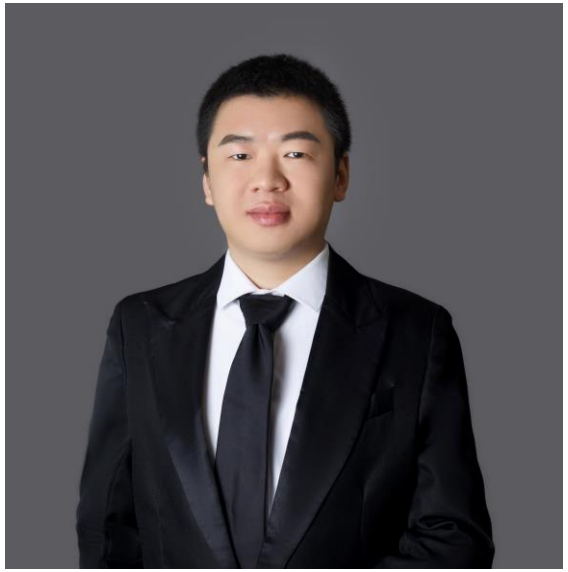
Explicit Min-wise Hash Families with Optimal Size

Xue Chen*



Shengtang Huang*

Xin Li†

* University of Science and Technology of China

† Johns Hopkins University

Min-wise Hash Family

Definition (Min-wise Hash Family [Broder, Charikar, Frieze, Mitzenmacher '00])

- We say $\mathcal{H} = \{h : [N] \to [M]\}$ is a min-wise hash family with (multiplicative) error δ, if for any $X \subseteq [N]$ and $y \in X$,

$$\Pr_{h\sim\mathcal{H}}[h(y) < \min h(X\backslash y)] := \Pr_{h\sim\mathcal{H}}\left[h(y) < \min_{x\in X\backslash y} h(x)\right] = \frac{1\pm\delta}{|X|}.$$

Definition (k-min-wise Hash Family [Feigenblat, Porat, Shiftan '11])

- We say $\mathcal{H} = \{h : [N] \to [M]\}$ is a k-min-wise hash family with (multiplicative) error δ, if for any $X \subseteq [N]$ and $Y \in \binom{X}{\leq k}$,

$$\Pr_{h\sim\mathcal{H}}[\max h(Y) < \min h(X\backslash Y)] := \Pr_{h\sim\mathcal{H}}\left[\max_{y\in Y} h(y) < \min_{x\in X\backslash Y} h(x)\right] = \frac{1\pm\delta}{\binom{|X|}{|Y|}}.$$

- Play crucial roles in the design of graph algorithms and streaming algorithms

Problem

- **Goal:** Construct an explicit (k-)min-wise hash family with small size and error.
- **Short seed length** $= \log_2 |\mathcal{H}|$ (number of random bits used to generate a hash function).
- **Explicitness:** This family $\mathcal{H}$ should be efficiently computable in polynomial time.

Prior Works

Reference	Min-wise hash	k-min-wise hash
[Indyk '01] [Feigenblat, Porat, Shiftan '11] ***t*-wise independence**	$O(\log(1/\delta)\log N)$	$O\big((\log(1/\delta)+k\log\log(1/\delta))\log N\big)$

Prior Works

Reference	Min-wise hash	k-min-wise hash
[Indyk '01] [Feigenblat, Porat, Shiftan '11] **t-wise independence**	$O(\log(1/\delta)\log N)$	$O\big((\log(1/\delta)+k\log\log(1/\delta))\log N\big)$
[Saks, Srinivasan, Zhou, Zuckerman '00] [Gopalan, Yehudayoff '20] **PRG-based**	$O(\log(N/\delta)\log\log(N/\delta))$	$O\big((k\log N+\log(1/\delta))\cdot\log(k\log N+\log(1/\delta))\big)$

Prior Works

Reference	Min-wise hash	k-min-wise hash
[Indyk '01] [Feigenblat, Porat, Shiftan '11] **t-wise independence**	$O(\log(1/\delta)\log N)$	$O\big((\log(1/\delta)+k\log\log(1/\delta))\log N\big)$
[Saks, Srinivasan, Zhou, Zuckerman '00] [Gopalan, Yehudayoff '20] **PRG-based**	$O(\log(N/\delta)\log\log(N/\delta))$	$O\big((k\log N+\log(1/\delta))\cdot\log(k\log N+\log(1/\delta))\big)$
Non-explicit Constructions	$\boldsymbol{O(\log(N/\delta))}$	$\boldsymbol{O(k\log N+\log(1/\delta))}$

Our Results

Reference	Min-wise hash	k-min-wise hash
[Indyk '01] [Feigenblat, Porat, Shiftan '11] **t-wise independence**	$O(\log(1/\delta)\log N)$	$O\big((\log(1/\delta)+k\log\log(1/\delta))\log N\big)$
[Saks, Srinivasan, Zhou, Zuckerman '00] [Gopalan, Yehudayoff '20] **PRG-based**	$O(\log(N/\delta)\log\log(N/\delta))$	$O\big((k\log N+\log(1/\delta))\cdot\log(k\log N+\log(1/\delta))\big)$
Non-explicit Constructions	$\boldsymbol{O(\log(N/\delta))}$	$\boldsymbol{O(k\log N+\log(1/\delta))}$
Our Results	$\boldsymbol{O(\log N)}$ $\boldsymbol{\delta=2^{-o\left(\frac{\log N}{\log\log N}\right)}}$	$\boldsymbol{O(k\log N)}$ $\boldsymbol{\delta=2^{-o\left(\frac{\log N}{\log\log N}\right)}, k=\log^{O(1)} N}$

Outline

1st Step – Balls into Bins

2nd Step – Recycle Randomness

3rd Step – Domain Reduction

$$\Pr_{h\sim\mathcal{H}}[h(y) < \min h(X\backslash y)] = \frac{1 \pm \delta}{|X|}$$

$$= \sum_{\theta=1}^{M} \Pr_{h\sim\mathcal{H}}[h(y) = \theta \wedge \min h(X\backslash y) > \theta]$$

Two Level Hash Structure

$$h(x) = \sigma_{\varphi(x)}(x)$$

- **Balls into Bins:** small max-load, concentration.

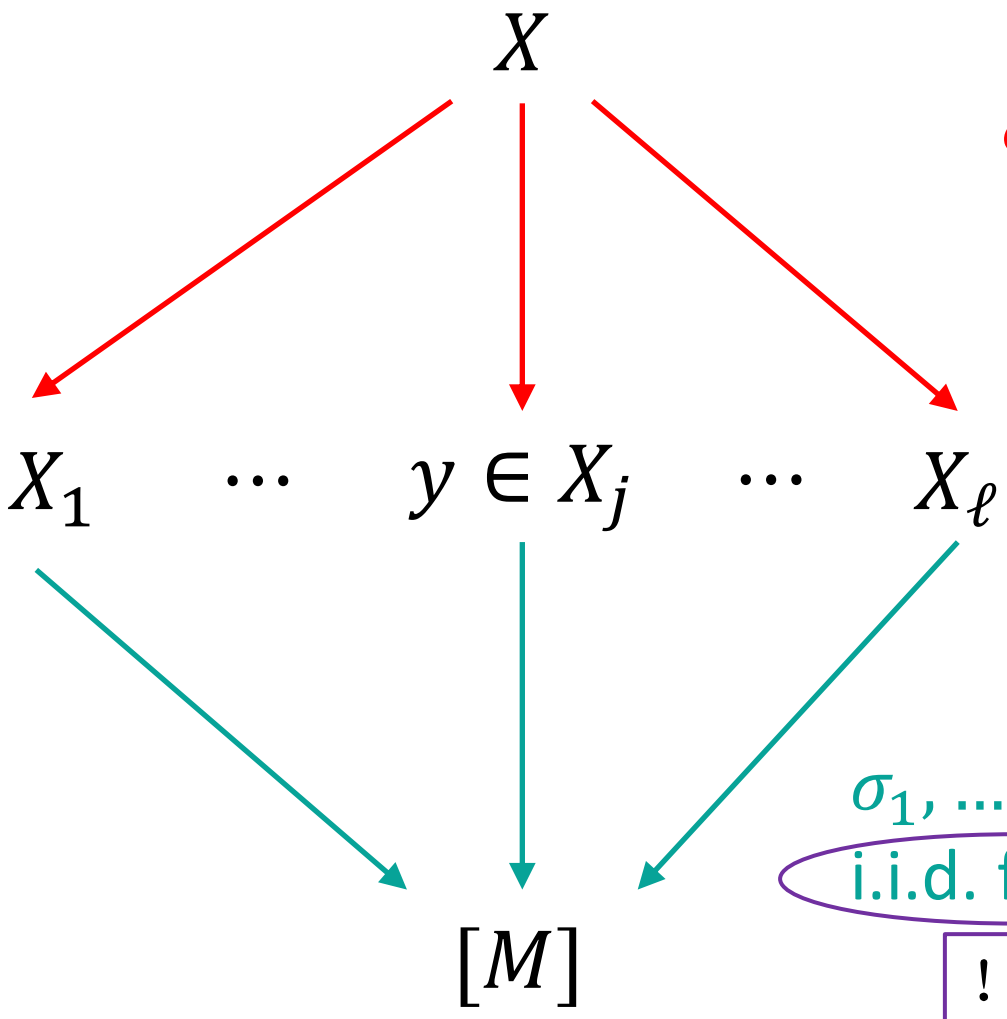
$$|X_i| \approx |X|/\ell, \forall i \in [\ell]$$

$$\max |X_i| \text{ is small}$$

- **Each Block:** Since $|X_i|$ is small, there exists $\mathcal{G} = \{\sigma: [N] \to [M]\}$ with $|\mathcal{G}| = \text{poly}(N)$, s.t.

$$\Pr_{\sigma_i \sim \mathcal{G}}[\min \sigma_i(X_i) > \theta]$$

$$\approx \Pr_{h \sim U}[\min h(X_i) > \theta]$$



$\varphi: [N] \to [\ell]$

Constant-wise independence

$\sigma_1, \ldots, \sigma_\ell: [N] \to [M]$

i.i.d. from $\mathcal{G}$

! Require many random bits.

Recycle Randomness

$$\underbrace{(\min \sigma_1(X_1) > \theta)}_{\text{Event } A_1} \wedge \cdots \wedge \underbrace{\left(\sigma_j(y) = \theta \wedge \min \sigma_j\left(X_j \backslash y\right) > \theta\right)}_{\text{Event } A_j} \wedge \cdots \wedge \underbrace{(\min \sigma_\ell(X_\ell) > \theta)}_{\text{Event } A_\ell}$$

$\sigma_1, \ldots, \sigma_\ell$ i.i.d. from $\mathcal{G}$

$\approx$

$$(\min \sigma_1'(X_1) > \theta) \wedge \cdots \wedge \left(\sigma_j'(y) = \theta \wedge \min \sigma_j'\left(X_j \backslash y\right) > \theta\right) \wedge \cdots \wedge (\min \sigma_\ell'(X_\ell) > \theta)$$

$\sigma_1', \ldots, \sigma_\ell'$ are correlated

Nisan-Zuckerman Pseudorandom Generator

Definition (Extractor)

- $\mathrm{Ext}: \{0,1\}^p \times \{0,1\}^d \to \{0,1\}^q$ is a (k,ε)-extractor, if for any random source X over $\{0,1\}^p$ with min-entropy $H_\infty(X) \geq k$, it holds that $\mathrm{Ext}(X, U_d)$ is ε-close to U_q.

- Nisan-Zuckerman PRG:

$$\mathrm{NZPRG}(w, s_1, \dots, s_\ell) = \big(\mathrm{Ext}(w, s_1), \dots, \mathrm{Ext}(w, s_\ell)\big) \in (\{0,1\}^q)^\ell,$$
$$w \sim U_p, s_1, \dots, s_\ell \text{ i.i.d. from } U_d.$$

$$\underbrace{(\min \sigma_1'(X_1) > \theta)}_{\text{Event } A_1} \wedge \cdots \wedge \underbrace{\left(\sigma_j'(y) = \theta \wedge \min \sigma_j'\left(X_j \backslash y\right) > \theta\right)}_{\text{Event } A_j} \wedge \cdots \wedge \underbrace{(\min \sigma_\ell'(X_\ell) > \theta)}_{\text{Event } A_\ell}$$

$$\mathrm{Ext}: \{0,1\}^p \times \{0,1\}^d \to \mathcal{G}$$
$$\sigma_1' = \mathrm{Ext}(w, s_1), \dots, \sigma_\ell' = \mathrm{Ext}(w, s_\ell)$$
$$w \sim U_p, s_1, \dots, s_\ell \text{ i.i.d. from } U_d$$

One bin is Sensitive to Multiplicative Error

$$h(x) = \sigma'_{\varphi(x)}(x)$$

$$\Pr_{h\sim\mathcal{H}}[h(y) = \theta \wedge \min h(X\backslash y) > \theta]$$

$$\| \qquad\qquad \varphi\colon [N] \to [\ell]$$

$$P_1[\mathrm{NZ}] \cdots P_j[\mathrm{NZ}] \cdots P_\ell[\mathrm{NZ}]$$

$$\approx \qquad\qquad \sigma'_1, \ldots, \sigma'_\ell\colon [N] \to [M] \text{ from NZPRG, ExtErr} = N^{-o(1)}$$

$P_i[\mathcal{G}] \approx P_i[U]$

$$P_i[\mathrm{NZ}] = \Pr_{\sigma'_i:\mathrm{NZPRG}}[A_i]$$

$$P_i[\mathcal{G}] \coloneqq \Pr_{\sigma_i\sim\mathcal{G}}[A_i]$$

$$P_i[U] \coloneqq \Pr_{h\sim U}[A_i]$$

? We hope that this special bin does not need to pay for the error from extractor.

$$(P_1[U] \pm \mathrm{ExtErr}) \cdots (P_j[U] \pm \mathrm{ExtErr}) \cdots (P_\ell[U] \pm \mathrm{ExtErr})$$

$$P_j[U] = \Pr_{h\sim U}[h(y) = \theta \wedge \min h(X_j\backslash y) > \theta] \le 1/M \implies P_j[U] \pm \mathrm{ExtErr} = P_j[U] \cdot (1 \pm M \cdot \mathrm{ExtErr})$$

! $P_j[U]$ is very small, and it makes this bin very sensitive to multiplicative error.

Change the Order of Inputs

$$\underbrace{(\min \sigma_1'(X_1) > \theta)}_{\text{Event } A_1} \wedge \cdots \wedge \underbrace{\left(\sigma_j'(y) = \theta \wedge \min \sigma_j'\left(X_j \backslash y\right) > \theta\right)}_{\text{Event } A_j} \wedge \cdots \wedge \underbrace{(\min \sigma_\ell'(X_\ell) > \theta)}_{\text{Event } A_\ell}$$

$$\text{Ext}: \{0,1\}^p \times \{0,1\}^d \to \mathcal{G}$$

$$\sigma_1' = \text{Ext}(w, s_1), \ldots, \sigma_\ell' = \text{Ext}(w, s_\ell)$$

$$w \sim U_p, s_1, \ldots, s_\ell \text{ i.i.d. from } U_d$$

★ $s_1, \ldots s_\ell$ (or correspondingly $\sigma_1', \ldots, \sigma_\ell'$) are symmetric.

Change the Order of Inputs

$$\underbrace{\left(\sigma_j'(y) = \theta \wedge \min \sigma_j'\left(X_j \backslash y\right) > \theta\right)}_{\text{Event } A_j} \wedge \underbrace{\left(\min \sigma_1'(X_1) > \theta\right)}_{\text{Event } A_1} \wedge \cdots \wedge \underbrace{\left(\min \sigma_\ell'(X_\ell) > \theta\right)}_{\text{Event } A_\ell}$$

$$\text{Ext}: \{0,1\}^p \times \{0,1\}^d \to \mathcal{G}$$

$$\sigma_j' = \text{Ext}\left(w, s_j\right), \sigma_1' = \text{Ext}(w, s_1), \dots, \sigma_\ell' = \text{Ext}(w, s_\ell)$$

$$w \sim U_p, s_j, s_1, \dots, s_\ell \text{ i.i.d. from } U_d$$

$$P_j[\mathrm{NZ}] \cdot P_1[\mathrm{NZ}] \cdots P_\ell[\mathrm{NZ}]$$

$$\approx$$

$$(P_j[U] \pm \mathrm{ExtErr}) \cdot (P_1[U] \pm \mathrm{ExtErr}) \cdots (P_\ell[U] \pm \mathrm{ExtErr})$$

$H_\infty(w = U_p) = p \geq k$

$\Rightarrow \mathrm{Ext}(w, s_j) \approx_{\mathrm{ExtErr}} \mathcal{G}$

★ w has no entropy loss at the beginning.

? We expect strong properties for $\sigma_j' = \mathrm{Ext}(w, s_j)$ than other $\sigma_i' = \mathrm{Ext}(w, s_i)$.

Special Extractor

$$P_j[\mathrm{NZ}] \cdot P_1[\mathrm{NZ}] \cdots P_\ell[\mathrm{NZ}]$$

$$\approx$$

$$(P_j[U] \pm \mathrm{ExtErr}) \cdot (P_1[U] \pm \mathrm{ExtErr}) \cdots (P_\ell[U] \pm \mathrm{ExtErr})$$

$H_\infty(w = U_p) = p \geq k$

$\Rightarrow \mathrm{Ext}(w, s_j) \approx_{\mathrm{ExtErr}} \mathcal{G}$

$\sigma'_j = \mathrm{Ext}(w, s_j) \sim \mathcal{G}$

★ w has no entropy loss at the beginning.

? We expect strong properties for $\sigma'_j = \mathrm{Ext}(w, s_j)$ than other $\sigma'_i = \mathrm{Ext}(w, s_i)$.

Lemma

- Given any p and $k < p$, for any error ε, there exists an explicit (k, ε)-extractor $\mathrm{Ext}: \{0,1\}^p \times \{0,1\}^d \to \{0,1\}^q$ with $q = k/2$ and $d = O(\log(p/\varepsilon))$. And Ext satisfies an extra property: $\mathrm{Ext}(U_p, s) = U_q$ for any fixed seed s.

Open Problems

Smaller error with optimal seed length?

Extending the result for larger k (like $\sqrt{N}$) on k-min-wise hash?

Faster evaluation time with optimal seed length?

Thank you for listening!