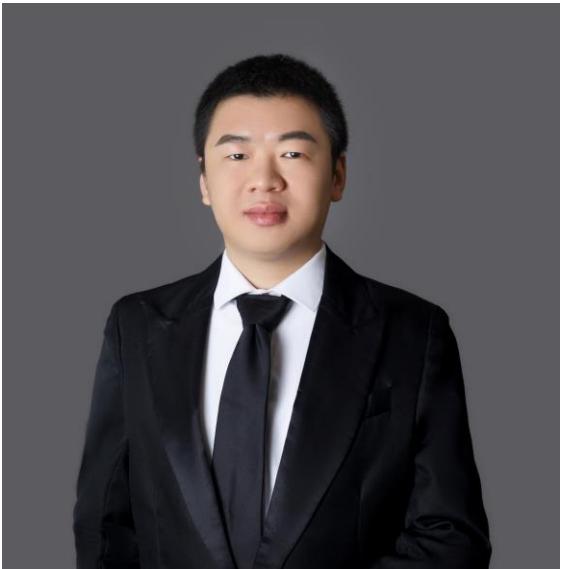


Explicit Min-wise Hash Families with Optimal Size

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

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

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

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

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

- 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, Shiffan '11] | $O(\log(1/\delta) \log N)$ | $O((\log(1/\delta) + k \log \log(1/\delta)) \log N)$ |

Prior Works

| Reference | Min-wise hash | k -min-wise hash |
|--|---|--|
| [Indyk '01] [Feigenblat, Porat, Shiftan '11] | $O(\log(1/\delta) \log N)$ | $O((\log(1/\delta) + k \log \log(1/\delta)) \log N)$ |
| [Saks, Srinivasan, Zhou, Zuckerman '00] [Gopalan, Yehudayoff '20] | $O(\log(N/\delta) \log \log(N/\delta))$ | $O((k \log N + \log(1/\delta)) \cdot \log(k \log N + \log(1/\delta)))$ |

Prior Works

| Reference | Min-wise hash | k -min-wise hash |
|--|---|--|
| [Indyk '01] [Feigenblat, Porat, Shiftan '11] | $O(\log(1/\delta) \log N)$ | $O((\log(1/\delta) + k \log \log(1/\delta)) \log N)$ |
| [Saks, Srinivasan, Zhou, Zuckerman '00] [Gopalan, Yehudayoff '20] | $O(\log(N/\delta) \log \log(N/\delta))$ | $O((k \log N + \log(1/\delta)) \cdot \log(k \log N + \log(1/\delta)))$ |
| Non-explicit Constructions | $O(\log(N/\delta))$ | $O(k \log N + \log(1/\delta))$ |

Our Results

| Reference | Min-wise hash | k -min-wise hash |
|--|---|--|
| [Indyk '01] [Feigenblat, Porat, Shiftan '11] | $O(\log(1/\delta) \log N)$ | $O((\log(1/\delta) + k \log \log(1/\delta)) \log N)$ |
| [Saks, Srinivasan, Zhou, Zuckerman '00] [Gopalan, Yehudayoff '20] | $O(\log(N/\delta) \log \log(N/\delta))$ | $O((k \log N + \log(1/\delta)) \cdot \log(k \log N + \log(1/\delta)))$ |
| Non-explicit Constructions | $O(\log(N/\delta))$ | $O(k \log N + \log(1/\delta))$ |
| Our Results | $O(\log N)$ $\delta = 2^{-O\left(\frac{\log N}{\log \log N}\right)}$ | $O(k \log N)$ $\delta = 2^{-O\left(\frac{\log N}{\log \log N}\right)}, k = \log^{o(1)} N$ |

Outline

1st STEP – BALLS INTO BINS

2nd STEP – RECYCLE RANDOMNESS

3rd STEP – DOMAIN REDUCTION

$$\Pr_{h \sim \mathcal{H}} [h(y) < \min h(X \setminus y)] = \frac{1 \pm \delta}{|X|}$$
$$= \sum_{\theta=1}^M \Pr_{h \sim \mathcal{H}} [h(y) = \theta \wedge \min h(X \setminus y) > \theta]$$

Two Level Hash Structure

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

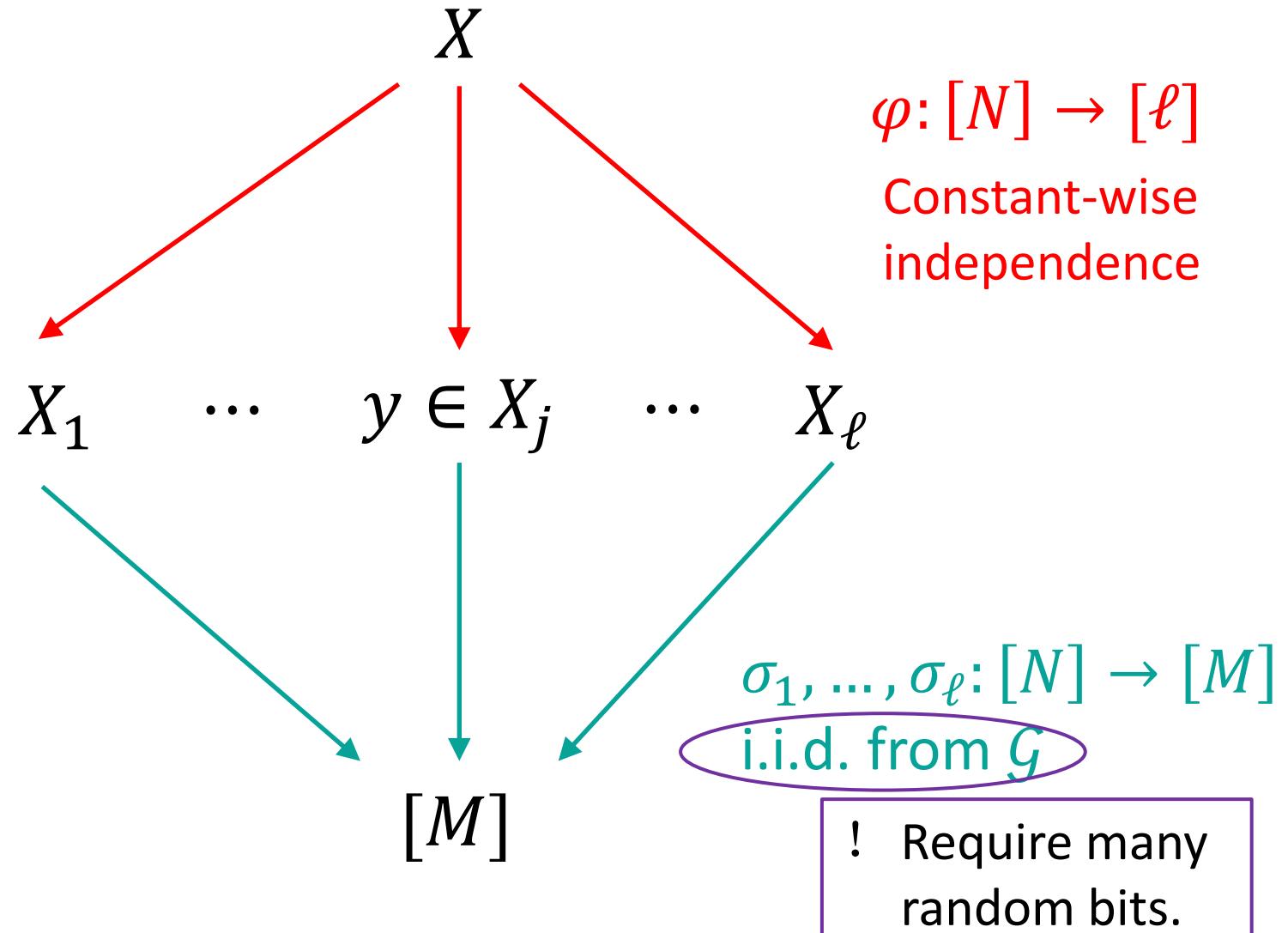
$$|X_i| \approx |X|/\ell, \forall i \in [\ell]$$

max $|X_i|$ is small

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

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

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



Recycle Randomness

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

Event A_1

Event A_j

Event A_ℓ

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

|||

$$(\min \sigma'_1(X_1) > \theta) \wedge \cdots \wedge (\sigma'_j(y) = \theta \wedge \min \sigma'_j(X_j \setminus y) > \theta) \wedge \cdots \wedge (\min \sigma'_\ell(X_\ell) > \theta)$$

$\sigma'_1, \dots, \sigma'_\ell$ are correlated

Nisan-Zuckerman Pseudorandom Generator

Definition (Extractor)

- $\text{Ext} : \{0, 1\}^p \times \{0, 1\}^d \rightarrow \{0, 1\}^q$ 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 $\text{Ext}(X, U_d)$ is ε -close to U_q .
- Nisan-Zuckerman PRG:

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

$$(\underbrace{\min \sigma'_1(X_1) > \theta} \wedge \dots \wedge \underbrace{(\sigma'_j(y) = \theta \wedge \min \sigma'_j(X_j \setminus y) > \theta)} \wedge \dots \wedge \underbrace{\min \sigma'_\ell(X_\ell) > \theta})$$

Event A_1

Event A_j

Event A_ℓ

Ext: $\{0, 1\}^p \times \{0, 1\}^d \rightarrow \mathcal{G}$
 $\sigma'_1 = \text{Ext}(w, s_1), \dots, \sigma'_\ell = \text{Ext}(w, s_\ell)$
 $w \sim U_p, s_1, \dots, s_\ell$ i.i.d. from U_d

One bin is Sensitive to Multiplicative Error

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

$$\Pr_{h \sim \mathcal{H}} [h(y) = \theta \wedge \min_{y \in X} h(X \setminus y) > \theta]$$

||

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

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

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

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

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

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

||

$$\sigma'_1, \dots, \sigma'_\ell: [N] \rightarrow [M] \text{ from NZPRG, } \text{ExtErr} = N^{-o(1)}$$

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

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

$$P_j[U] = \Pr_{h \sim U} [h(y) = \theta \wedge \min_{y \in X_j \setminus \{j\}} h(X_j \setminus y) > \theta] \leq 1/M \Rightarrow P_j[U] \pm \text{ExtErr} = P_j[U] \cdot (1 \pm M \cdot \text{ExtErr})$$

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

Change the Order of Inputs

$$(\underbrace{\min \sigma'_1(X_1) > \theta} \wedge \cdots \wedge \underbrace{(\sigma'_j(y) = \theta \wedge \min \sigma'_j(X_j \setminus y) > \theta)}_{\text{Event } A_j} \wedge \cdots \wedge \underbrace{\min \sigma'_{\ell}(X_{\ell}) > \theta}_{\text{Event } A_{\ell}})$$

Event A_1

Event A_j

Event A_{ℓ}

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

$$\sigma'_1 = \text{Ext}(w, s_1), \dots, \sigma'_{\ell} = \text{Ext}(w, s_{\ell})$$

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

★ s_1, \dots, s_{ℓ} (or correspondingly $\sigma'_1, \dots, \sigma'_{\ell}$) are symmetric.

Change the Order of Inputs

$$\left(\underbrace{\sigma'_j(y) = \theta \wedge \min \sigma'_j(X_j \setminus y) > \theta}_{\text{Event } A_j} \right) \wedge \left(\underbrace{\min \sigma'_1(X_1) > \theta}_{\text{Event } A_1} \right) \wedge \cdots \wedge \left(\underbrace{\min \sigma'_{\ell}(X_{\ell}) > \theta}_{\text{Event } A_{\ell}} \right)$$

Event A_j

Event A_1

Event A_{ℓ}

$$\begin{aligned} \text{Ext: } & \{0,1\}^p \times \{0,1\}^d \rightarrow \mathcal{G} \\ \sigma'_j &= \text{Ext}(w, s_j), \sigma'_1 = \text{Ext}(w, s_1), \dots, \sigma'_{\ell} = \text{Ext}(w, s_{\ell}) \\ w &\sim U_p, s_j, s_1, \dots, s_{\ell} \text{ i.i.d. from } U_d \end{aligned}$$

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

?

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

$$\begin{aligned} H_\infty(w = U_p) &= p \geq k \\ \Rightarrow \text{Ext}(w, s_j) &\approx_{\text{ExtErr}} \mathcal{G} \end{aligned}$$

★ w has no entropy loss at the beginning.
? We expect strong properties for $\sigma'_j = \text{Ext}(w, s_j)$ than other $\sigma'_i = \text{Ext}(w, s_i)$.

Special Extractor

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

?



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

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

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

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

★ w has no entropy loss at the beginning.

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

Lemma

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