A Fast Hadamard Transform for Signals with Sub-linear Sparsity

Robin Scheibler Saeid Haghighatshoar Martin Vetterli

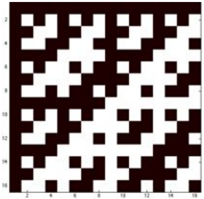
School of Computer and Communication Sciences
École Polytechnique Fédérale de Lausanne, Switzerland

October 3, 2013

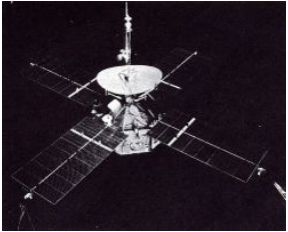
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Why the Hadamard transform ?

- **Historically**, low computation approximation to DFT.
- **Coding**, 1969 *Mariner* Mars probe.
- **Communication**, orthogonal codes in WCDMA.
- **Compressed sensing**, maximally incoherent with Dirac basis.
- **Spectroscopy**, design of instruments with lower noise.



16 $\times$ 16 Hadamard matrix



Mariner probe

Related work

Ghazi et al., 2013. (Next talk!)

- Spectrum bucketing through downsampling.
- Two-dimensional sparse DFT.
- Constant probability of failure.

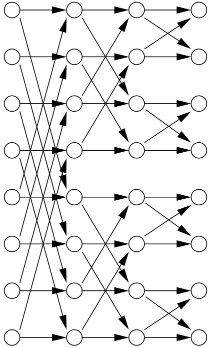
Pawar & Ramchandran, 2013.

- Spectrum bucketing through downsampling.
- One-dimensional sparse DFT.
- Length is power of small co-prime numbers.
- Probability of failure asymptotically vanishing.

Fast Hadamard transform

- Butterfly structure similar to FFT.
- Time complexity $O(N \log_2 N)$.
- Sample complexity N.
- \+ Universal, i.e. works for all signals.
- – Does not exploit signal structure (e.g. sparsity).

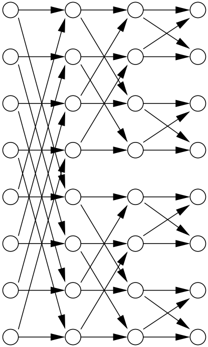
Can we do better ?



Fast Hadamard transform

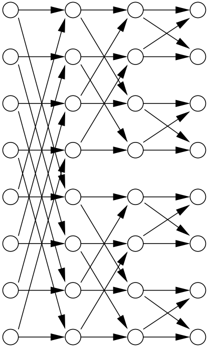
- Butterfly structure similar to FFT.
- Time complexity $O(N \log_2 N)$.
- Sample complexity N.
- + Universal, i.e. works for all signals.
- − Does not exploit signal structure (e.g. sparsity).

Can we do better ?



Fast Hadamard transform

- Butterfly structure similar to FFT.
- Time complexity $O(N \log_2 N)$.
- Sample complexity N.
- \+ Universal, i.e. works for all signals.
- – Does not exploit signal structure (e.g. sparsity).

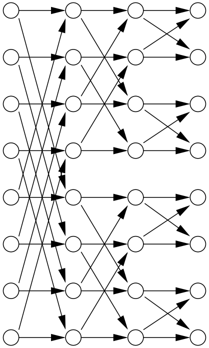


Can we do better ?

Fast Hadamard transform

- ► Butterfly structure similar to FFT.
- ► Time complexity $O(N \log_2 N)$.
- ► Sample complexity N.
- + Universal, i.e. works for all signals.
- – Does not exploit signal structure (e.g. sparsity).

Can we do better ?



Contribution: Sparse fast Hadamard transform

Assumptions

- The signal is exaclty K-sparse in the transform domain.
- Sub-linear sparsity regime $K = O(N^{\alpha})$, $0 < \alpha < 1$.
- Support of the signal is uniformly random.

Contribution

An algorithm computing the K non-zero coefficients with:

- Time complexity $O(K \log_2 K \log_2 \frac{N}{K})$.
- Sample complexity $O(K \log_2 \frac{N}{K})$.
- Probability of failure asymptotically vanishes.

Contribution: Sparse fast Hadamard transform

Assumptions

- The signal is exaclty K-sparse in the transform domain.
- Sub-linear sparsity regime $K = O(N^\alpha)$, $0 < \alpha < 1$.
- Support of the signal is uniformly random.

Contribution

An algorithm computing the K non-zero coefficients with:

- Time complexity $O(K \log_2 K \log_2 \frac{N}{K})$.
- Sample complexity $O(K \log_2 \frac{N}{K})$.
- Probability of failure asymptotically vanishes.

Outline

1. Sparse FHT algorithm
2. Analysis of probability of failure
3. Empirical results

Another look at the Hadamard transform

- Consider indices of $x \in \mathbb{R}^N$, $N = 2^n$.
- Take the binary expansion of indices.
- Represent signal on hypercube.
- Take DFT in every direction.

$$\mathcal{I} = \{0, \ldots, 2^3 - 1\}$$

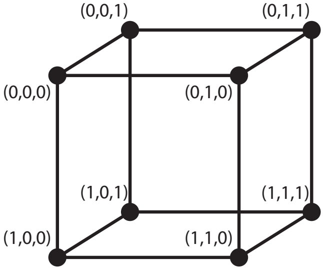
Another look at the Hadamard transform

- Consider indices of $x \in \mathbb{R}^N$, $N = 2^n$.
- Take the binary expansion of indices.
- Represent signal on hypercube.
- Take DFT in every direction.

$$\mathcal{I} = \{(0,0,0), \ldots, (1,1,1)\}$$

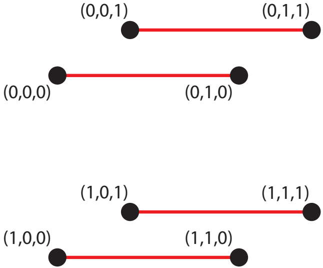
Another look at the Hadamard transform

- Consider indices of $x \in \mathbb{R}^N$, $N = 2^n$.
- Take the binary expansion of indices.
- Represent signal on hypercube.
- Take DFT in every direction.



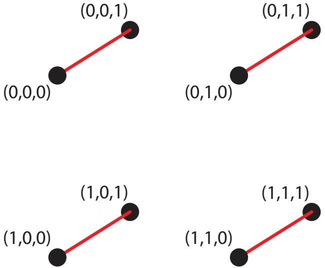
Another look at the Hadamard transform

- Consider indices of $x \in \mathbb{R}^N$, $N = 2^n$.
- Take the binary expansion of indices.
- Represent signal on hypercube.
- Take DFT in every direction.



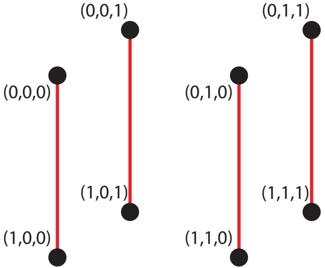
Another look at the Hadamard transform

- Consider indices of $x \in \mathbb{R}^N$, $N = 2^n$.
- Take the binary expansion of indices.
- Represent signal on hypercube.
- Take DFT in every direction.



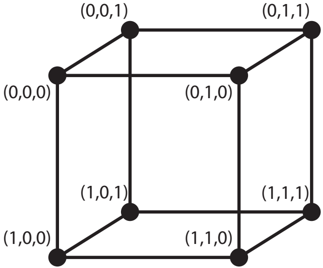
Another look at the Hadamard transform

- Consider indices of $x \in \mathbb{R}^N$, $N = 2^n$.
- Take the binary expansion of indices.
- Represent signal on hypercube.
- Take DFT in every direction.



Another look at the Hadamard transform

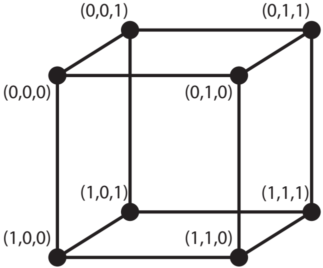
- Consider indices of $x \in \mathbb{R}^N$, $N = 2^n$.
- Take the binary expansion of indices.
- Represent signal on hypercube.
- Take DFT in every direction.



$$X_{k_0,\ldots,k_{n-1}} = \sum_{m_0=0}^{1} \cdots \sum_{m_{n-1}=0}^{1} (-1)^{k_0 m_0 + \cdots + k_{n-1} m_{n-1}} x_{m_0,\ldots,m_{n-1}},$$

Another look at the Hadamard transform

- Consider indices of $x \in \mathbb{R}^N$, $N = 2^n$.
- Take the binary expansion of indices.
- Represent signal on hypercube.
- Take DFT in every direction.

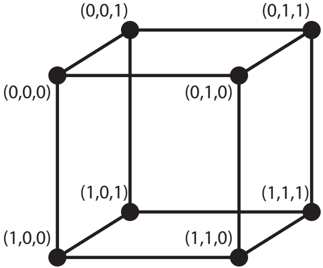


$$X_k = \sum_{m \in \mathbb{F}_2^n} (-1)^{\langle k\,,\,m\rangle} x_m, \qquad k, m \in \mathbb{F}_2^n, \quad \langle k\,,\,m\rangle = \sum_{i=0}^{n-1} k_i\, m_i.$$

Treat indices as binary vectors.

Another look at the Hadamard transform

- Consider indices of $x \in \mathbb{R}^N$, $N = 2^n$.
- Take the binary expansion of indices.
- Represent signal on hypercube.
- Take DFT in every direction.



$$X_k = \sum_{m \in \mathbb{F}_2^n} (-1)^{\langle k\,,\,m \rangle} x_m, \qquad k, m \in \mathbb{F}_2^n, \quad \langle k\,,\,m \rangle = \sum_{i=0}^{n-1} k_i\, m_i.$$

Treat indices as binary vectors.

Hadamard property I: downsampling/aliasing

Given $B = 2^b$, a divider of $N = 2^n$, and $\mathcal{H} \in \mathbb{F}_2^{b\times n}$, where rows of $\mathcal{H}$ are a subset of rows of identity matrix,

$$x_{\mathcal{H}^T m} \overset{WHT}{\longleftrightarrow} \sum_{i \in \mathcal{N}(\mathcal{H})} X_{\mathcal{H}^T k + i}, \quad m, k \in \mathbb{F}_2^b.$$

e.g. $\mathcal{H} = \left[0_{b\times(n-b)} \; I_b \right]$ selects the b high order bits.

Hadamard property I: downsampling/aliasing

Given $B = 2^b$, a divider of $N = 2^n$, and $\mathcal{H} \in \mathbb{F}_2^{b \times n}$, where rows of $\mathcal{H}$ are a subset of rows of identity matrix,

$$x_{\mathcal{H}^T m} \overset{WHT}{\longleftrightarrow} \sum_{i \in \mathcal{N}(\mathcal{H})} X_{\mathcal{H}^T k + i}, \quad m, k \in \mathbb{F}_2^b.$$

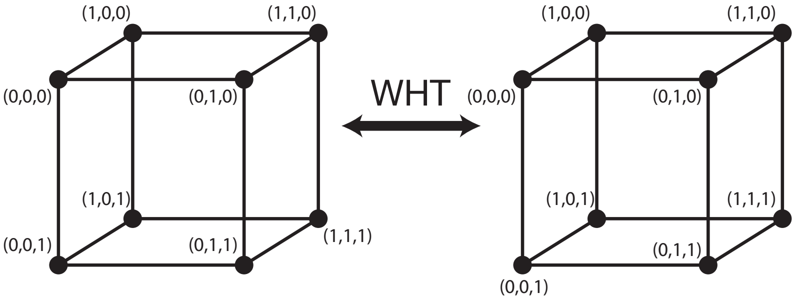
e.g. $\mathcal{H} = \begin{bmatrix} 0_{b \times (n-b)} & I_b \end{bmatrix}$ selects the b high order bits.

Hadamard property I: downsampling/aliasing

Given $B = 2^b$, a divider of $N = 2^n$, and $\mathcal{H} \in \mathbb{F}_2^{b \times n}$,
where rows of $\mathcal{H}$ are a subset of rows of identity matrix,

$$x_{\mathcal{H}^T m} \overset{WHT}{\longleftrightarrow} \sum_{i \in \mathcal{N}(\mathcal{H})} X_{\mathcal{H}^T k + i}, \quad m, k \in \mathbb{F}_2^b.$$

e.g. $\mathcal{H} = \left[0_{b \times (n-b)} \; I_b \right]$ selects the b high order bits.

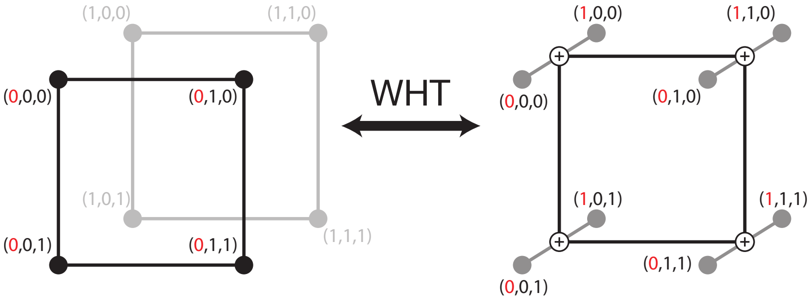


Hadamard property I: downsampling/aliasing

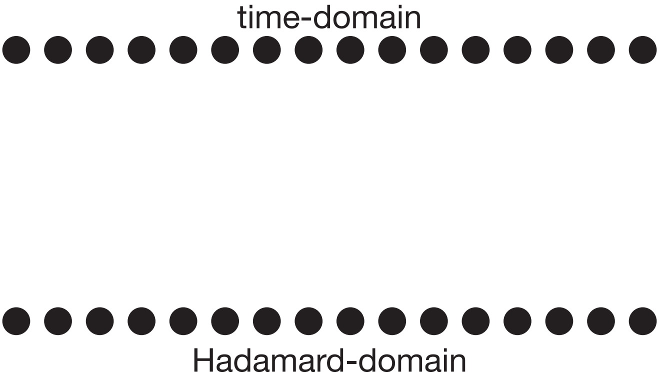
Given $B = 2^b$, a divider of $N = 2^n$, and $\mathcal{H} \in \mathbb{F}_2^{b \times n}$, where rows of $\mathcal{H}$ are a subset of rows of identity matrix,

$$x_{\mathcal{H}^T m} \overset{WHT}{\longleftrightarrow} \sum_{i \in \mathcal{N}(\mathcal{H})} X_{\mathcal{H}^T k + i}, \quad m, k \in \mathbb{F}_2^b.$$

e.g. $\mathcal{H} = \left[0_{b \times (n-b)} \; I_b \right]$ selects the b high order bits.

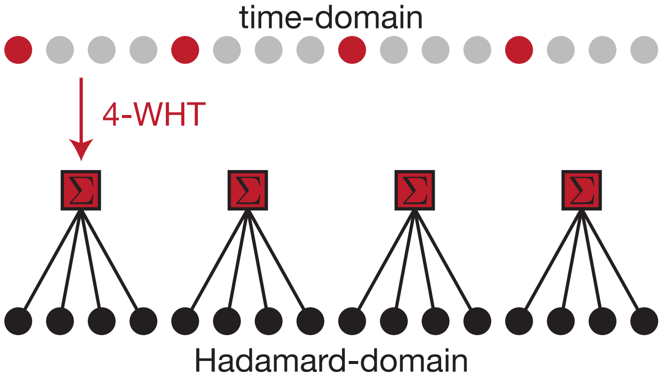


Aliasing induced bipartite graph



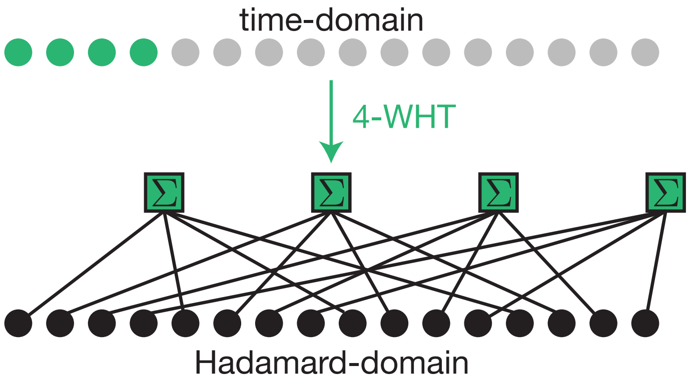
- Downsampling induces an aliasing pattern.
- Different downsamplings produce different patterns.

Aliasing induced bipartite graph



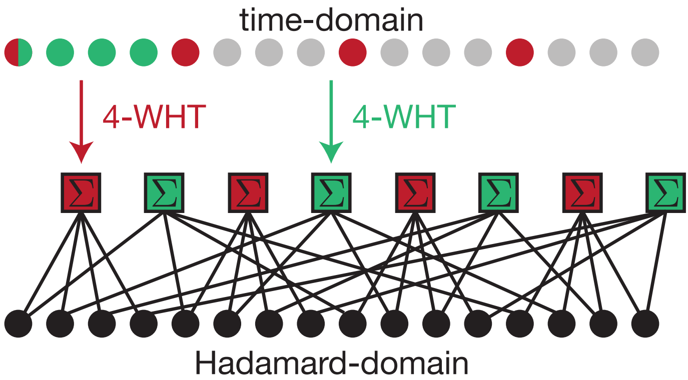
- Downsampling induces an aliasing pattern.
- Different downsamplings produce different patterns.

Aliasing induced bipartite graph



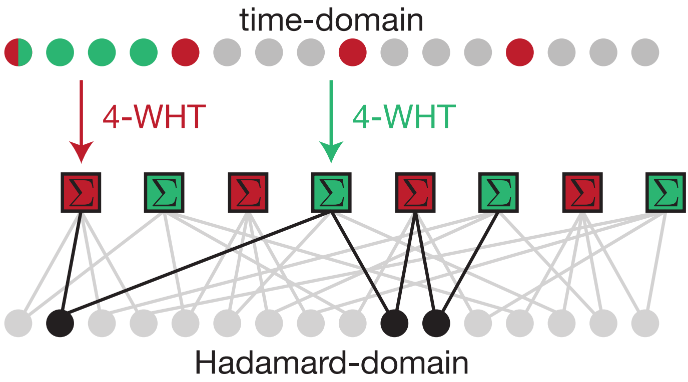
- Downsampling induces an aliasing pattern.
- Different downsamplings produce different patterns.

Aliasing induced bipartite graph



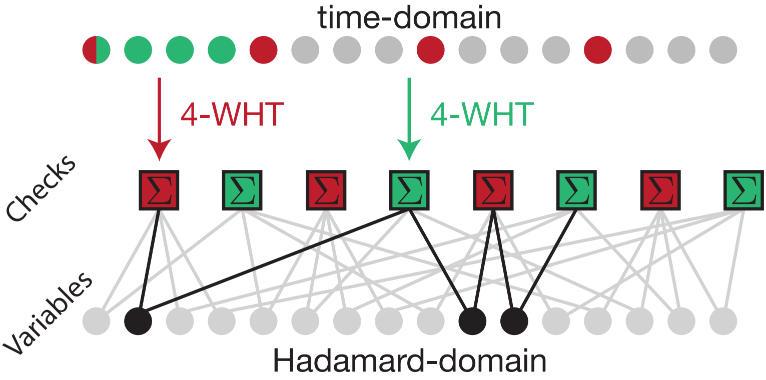
- Downsampling induces an aliasing pattern.
- Different downsamplings produce different patterns.

Aliasing induced bipartite graph



- Downsampling induces an aliasing pattern.
- Different downsamplings produce different patterns.

Aliasing induced bipartite graph

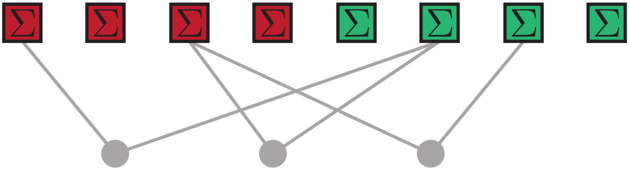


- Downsampling induces an aliasing pattern.
- Different downsamplings produce different patterns.

Genie-aided peeling decoder

A genie indicates us

- if a check is connected to only one variable (singleton),
- in that case, the genie also gives the index of that variable.



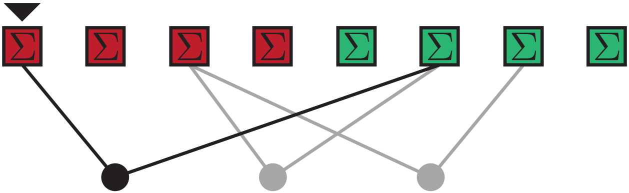
Peeling decoder algorithm:

1. Find a singleton check: $\{X_1, X_8, X_{11}\}$
2. Peel it off.
3. Repeat until nothing left.

Genie-aided peeling decoder

A genie indicates us

- if a check is connected to only one variable (singleton),
- in that case, the genie also gives the index of that variable.



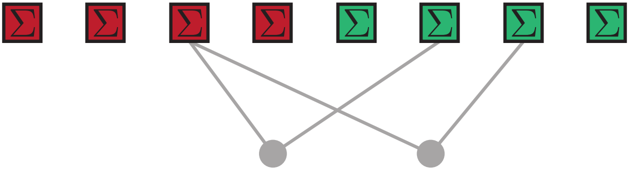
Peeling decoder algorithm:

1. Find a singleton check: $\{X_1, X_8, X_{11}\}$
2. Peel it off.
3. Repeat until nothing left.

Genie-aided peeling decoder

A genie indicates us

- if a check is connected to only one variable (singleton),
- in that case, the genie also gives the index of that variable.



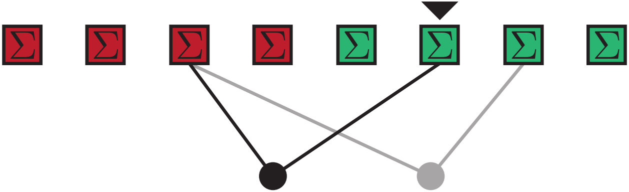
Peeling decoder algorithm:

1. Find a singleton check: $\{X_1, X_8, X_{11}\}$
2. Peel it off.
3. Repeat until nothing left.

Genie-aided peeling decoder

A genie indicates us

- if a check is connected to only one variable (singleton),
- in that case, the genie also gives the index of that variable.



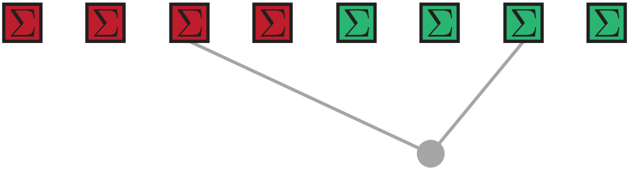
Peeling decoder algorithm:

1. Find a singleton check: $\{X_1, X_8, X_{11}\}$
2. Peel it off.
3. Repeat until nothing left.

Genie-aided peeling decoder

A genie indicates us

- if a check is connected to only one variable (singleton),
- in that case, the genie also gives the index of that variable.



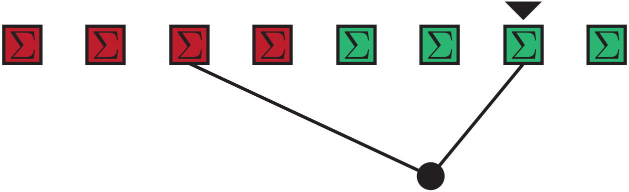
Peeling decoder algorithm:

1. Find a singleton check: $\{X_1, X_8, X_{11}\}$
2. Peel it off.
3. Repeat until nothing left.

Genie-aided peeling decoder

A genie indicates us

- if a check is connected to only one variable (singleton),
- in that case, the genie also gives the index of that variable.



Peeling decoder algorithm:

1. Find a singleton check: $\{X_1, X_8, X_{11}\}$
2. Peel it off.
3. Repeat until nothing left.

Genie-aided peeling decoder

A genie indicates us

- if a check is connected to only one variable (singleton),
- in that case, the genie also gives the index of that variable.

Success

Peeling decoder algorithm:

1. Find a singleton check: $\{X_1, X_8, X_{11}\}$
2. Peel it off.
3. Repeat until nothing left.

Hadamard property II: shift/modulation

Theorem (shift/modulation)

Given $p \in \mathbb{F}_2^n$,

$$x_{m+p} \overset{WHT}{\longleftrightarrow} X_k \, (-1)^{\langle p \, , \, k \rangle}.$$

Consequence

The signal can be modulated in frequency by manipulating the time-domain samples.

Hadamard property II: shift/modulation

Theorem (shift/modulation)

Given $p \in \mathbb{F}_2^n$*,*

$$x_{m+p} \overset{WHT}{\longleftrightarrow} X_k \, (-1)^{\langle p \, , \, k \rangle}.$$

Consequence

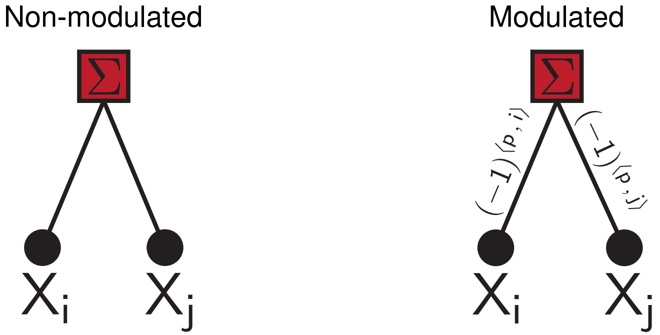
The signal can be modulated in frequency by manipulating the time-domain samples.

How to construct the Genie

Non-modulated | Modulated

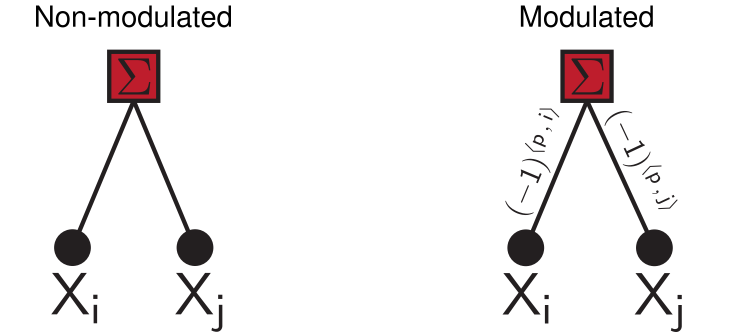
- Collision: if ≥ 2 variables connected to same check.
- $\frac{X_i(-1)^{\langle p\,,\,i\rangle}+X_j(-1)^{\langle p\,,\,j\rangle}}{X_i+X_j} \neq \pm 1$, (mild assumption on distribution of X).

How to construct the Genie



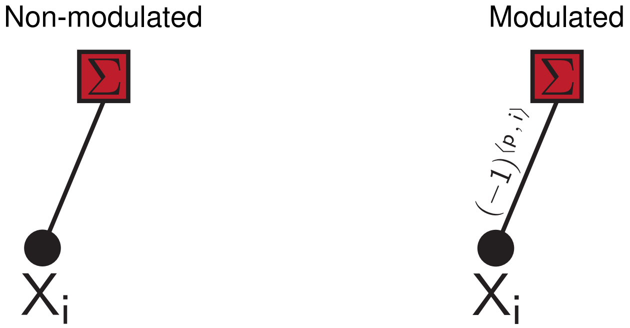
- Collision: if $\geq$ 2 variables connected to same check.
- $\frac{X_i(-1)^{\langle p,i\rangle}+X_j(-1)^{\langle p,j\rangle}}{X_i+X_j} \neq \pm 1$, (mild assumption on distribution of X).

How to construct the Genie



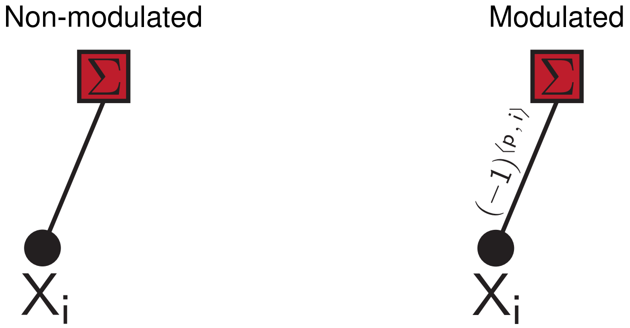
- Collision: if $\geq$ 2 variables connected to same check.
- $\frac{X_i(-1)^{\langle p,i \rangle} + X_j(-1)^{\langle p,j \rangle}}{X_i + X_j} \neq \pm 1$, (mild assumption on distribution of X).

How to construct the Genie



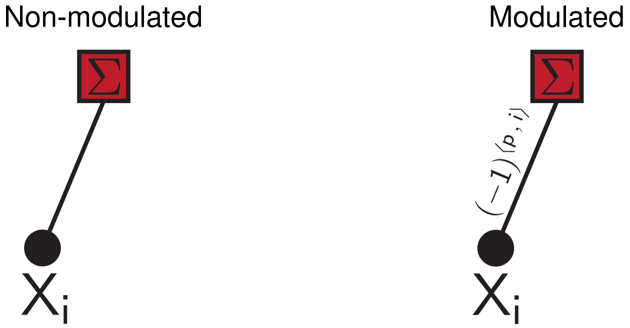
- Singleton: only one variable connected to check.
- $\frac{X_i(-1)^{\langle p,i \rangle}}{X_i} = (-1)^{\langle p,i \rangle} = \pm 1$. We can know $\langle p, i \rangle$!
- $O(\log_2 \frac{N}{K})$ measurements sufficient to recover index i, (dimension of null-space of downsampling matrix $\mathcal{H}$).

How to construct the Genie



- Singleton: only one variable connected to check.
- $\frac{X_i(-1)^{\langle p\,,\,i\rangle}}{X_i} = (-1)^{\langle p\,,\,i\rangle} = \pm 1$. We can know $\langle p\,,\,i\rangle$!
- $O(\log_2 \frac{N}{K})$ measurements sufficient to recover index i, (dimension of null-space of downsampling matrix $\mathcal{H}$).

How to construct the Genie



- Singleton: only one variable connected to check.
- $\frac{X_i(-1)^{\langle p , i \rangle}}{X_i} = (-1)^{\langle p , i \rangle} = \pm 1$. We can know $\langle p , i \rangle$!
- $O(\log_2 \frac{N}{K})$ measurements sufficient to recover index i, (dimension of null-space of downsampling matrix $\mathcal{H}$).

How to construct the Genie

- Singleton: only one variable connected to check.
- $\frac{X_i(-1)^{\langle p\,,\,i\rangle}}{X_i} = (-1)^{\langle p\,,\,i\rangle} = \pm 1$. We can know $\langle p\,,\,i\rangle$!
- $O(\log_2 \frac{N}{K})$ measurements sufficient to recover index i, (dimension of null-space of downsampling matrix $\mathcal{H}$).

Sparse fast Hadamard transform

Algorithm

1. Set number of checks per downsampling $B = O(K)$.
2. Choose C downsampling matrices $\mathcal{H}_1, \dots, \mathcal{H}_C$.
3. Compute $C(\log_2 N/K + 1)$ size-K fast Hadamard transform, each takes $O(K \log_2 K)$.
4. Decode non-zero coefficients using peeling decoder.

Performance

- Time complexity – $O(K \log_2 K \log_2 N/K)$.
- Sample complexity – $O(K \log_2 \frac{N}{K})$.
- How to construct $\mathcal{H}_1, \dots, \mathcal{H}_C$? Probability of success ?

Sparse fast Hadamard transform

Algorithm

1. Set number of checks per downsampling $B = O(K)$.
2. Choose C downsampling matrices $\mathcal{H}_1, \ldots, \mathcal{H}_C$.
3. Compute $C(\log_2 N/K + 1)$ size-K fast Hadamard transform, each takes $O(K \log_2 K)$.
4. Decode non-zero coefficients using peeling decoder.

Performance

- Time complexity – $O(K \log_2 K \log_2 N/K)$.
- Sample complexity – $O(K \log_2 \frac{N}{K})$.
- How to construct $\mathcal{H}_1, \ldots, \mathcal{H}_C$? Probability of success ?

Sparse fast Hadamard transform

Algorithm

1. Set number of checks per downsampling $B = O(K)$.
2. Choose C downsampling matrices $\mathcal{H}_1, \ldots, \mathcal{H}_C$.
3. Compute $C(\log_2 N/K + 1)$ size-K fast Hadamard transform, each takes $O(K \log_2 K)$.
4. Decode non-zero coefficients using peeling decoder.

Performance

- Time complexity – $O(K \log_2 K \log_2 N/K)$.
- Sample complexity – $O(K \log_2 \frac{N}{K})$.
- How to construct $\mathcal{H}_1, \ldots, \mathcal{H}_C$? Probability of success ?

Very sparse regime

Setting

- $K = O(N^\alpha)$, $0 < \alpha < 1/3$.
- Uniformly random support.
- Study asymptotic probability of failure as $n \to \infty$.

Downsampling matrices construction

- Achieves values $\alpha = \frac{1}{C}$, i.e. $b = \frac{n}{C}$.
- Deterministic downsampling matrices $\mathcal{H}_1, \ldots, \mathcal{H}_C$,

Very sparse regime

Setting

- $K = O(N^\alpha)$, $0 < \alpha < 1/3$.
- Uniformly random support.
- Study asymptotic probability of failure as $n \to \infty$.

Downsampling matrices construction

- Achieves values $\alpha = \frac{1}{C}$, i.e. $b = \frac{n}{C}$.
- Deterministic downsampling matrices $\mathcal{H}_1, \ldots, \mathcal{H}_C$,

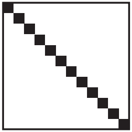
Very sparse regime

Setting

- $K = O(N^\alpha)$, $0 < \alpha < 1/3$.
- Uniformly random support.
- Study asymptotic probability of failure as $n \to \infty$.

Downsampling matrices construction

- Achieves values $\alpha = \frac{1}{C}$, i.e. $b = \frac{n}{C}$.
- Deterministic downsampling matrices $\mathcal{H}_1, \ldots, \mathcal{H}_C$,



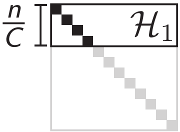
Very sparse regime

Setting

- $K = O(N^{\alpha})$, $0 < \alpha < 1/3$.
- Uniformly random support.
- Study asymptotic probability of failure as $n \to \infty$.

Downsampling matrices construction

- Achieves values $\alpha = \frac{1}{C}$, i.e. $b = \frac{n}{C}$.
- Deterministic downsampling matrices $\mathcal{H}_1, \ldots, \mathcal{H}_C$,



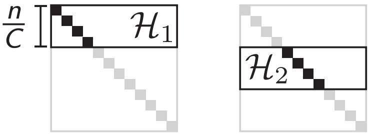
Very sparse regime

Setting

- $K = O(N^{\alpha})$, $0 < \alpha < 1/3$.
- Uniformly random support.
- Study asymptotic probability of failure as $n \to \infty$.

Downsampling matrices construction

- Achieves values $\alpha = \frac{1}{C}$, i.e. $b = \frac{n}{C}$.
- Deterministic downsampling matrices $\mathcal{H}_1, \ldots, \mathcal{H}_C$,



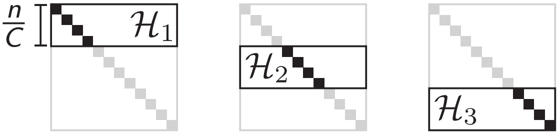
Very sparse regime

Setting

- $K = O(N^{\alpha})$, $0 < \alpha < 1/3$.
- Uniformly random support.
- Study asymptotic probability of failure as $n \to \infty$.

Downsampling matrices construction

- Achieves values $\alpha = \frac{1}{C}$, i.e. $b = \frac{n}{C}$.
- Deterministic downsampling matrices $\mathcal{H}_1, \ldots, \mathcal{H}_C$,



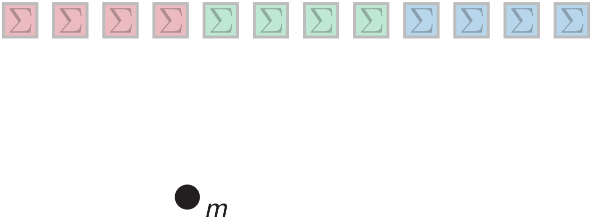
Balls-and-bins model

Uniformly random support model

- Theorem: Both constructions are equivalent.
  Proof: By construction, all rows of $\mathcal{H}_i$ are linearly independent. □
- Reduces to LDPC decoding analysis.
  - Error correcting code design (Luby et al. 2001).
  - FFAST (Sparse FFT algorithm) (Pawar & Ramchandran 2013).

Balls-and-bins model

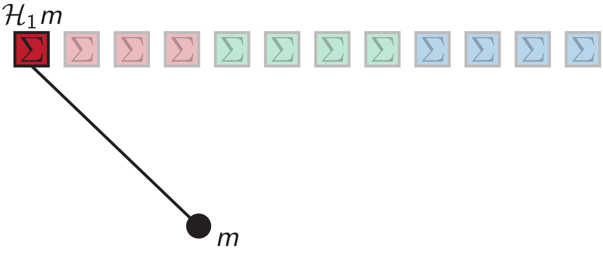
Uniformly random support model



- **Theorem:** Both constructions are equivalent.
  Proof: By construction, all rows of $\mathcal{H}_i$ are linearly independent. □
- Reduces to LDPC decoding analysis.
  - Error correcting code design (Luby et al. 2001).
  - FFAST (Sparse FFT algorithm) (Pawar & Ramchandran 2013).

Balls-and-bins model

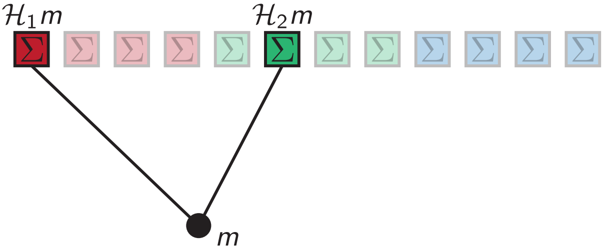
Uniformly random support model



- **Theorem:** Both constructions are equivalent.
  Proof: By construction, all rows of $\mathcal{H}_i$ are linearly independent. □
- Reduces to LDPC decoding analysis.
  - Error correcting code design (Luby et al. 2001).
  - FFAST (Sparse FFT algorithm) (Pawar & Ramchandran 2013).

Balls-and-bins model

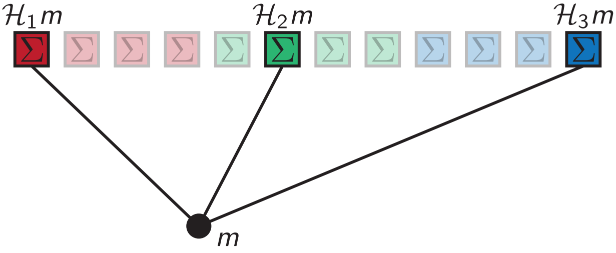
Uniformly random support model



- **Theorem:** Both constructions are equivalent.
  Proof: By construction, all rows of $\mathcal{H}_i$ are linearly independent. □
- Reduces to LDPC decoding analysis.
  - Error correcting code design (Luby et al. 2001).
  - FFAST (Sparse FFT algorithm) (Pawar & Ramchandran 2013).

Balls-and-bins model

Uniformly random support model



- Theorem: Both constructions are equivalent.
  Proof: By construction, all rows of $\mathcal{H}_i$ are linearly independent. □
- Reduces to LDPC decoding analysis.
  - Error correcting code design (Luby et al. 2001).
  - FFAST (Sparse FFT algorithm) (Pawar & Ramchandran 2013).

Balls-and-bins model

Balls-and-bins model

- **Theorem:** Both constructions are equivalent.
  Proof: By construction, all rows of $\mathcal{H}_i$ are linearly independent. □
- Reduces to LDPC decoding analysis.
  - Error correcting code design (Luby et al. 2001).
  - FFAST (Sparse FFT algorithm) (Pawar & Ramchandran 2013).

Balls-and-bins model

Balls-and-bins model

- **Theorem:** Both constructions are equivalent.
  Proof: By construction, all rows of $\mathcal{H}_i$ are linearly independent. □
- Reduces to LDPC decoding analysis.
  - Error correcting code design (Luby et al. 2001).
  - FFAST (Sparse FFT algorithm) (Pawar & Ramchandran 2013).

Balls-and-bins model

Balls-and-bins model

- **Theorem:** Both constructions are equivalent.
  Proof: By construction, all rows of $\mathcal{H}_i$ are linearly independent. □
- Reduces to LDPC decoding analysis.
  - Error correcting code design (Luby et al. 2001).
  - FFAST (Sparse FFT algorithm) (Pawar & Ramchandran 2013).

Balls-and-bins model

Balls-and-bins model

- **Theorem:** Both constructions are equivalent.
  Proof: By construction, all rows of $\mathcal{H}_i$ are linearly independent. □
- Reduces to LDPC decoding analysis.
  - Error correcting code design (Luby et al. 2001).
  - FFAST (Sparse FFT algorithm) (Pawar & Ramchandran 2013).

Balls-and-bins model

Balls-and-bins model

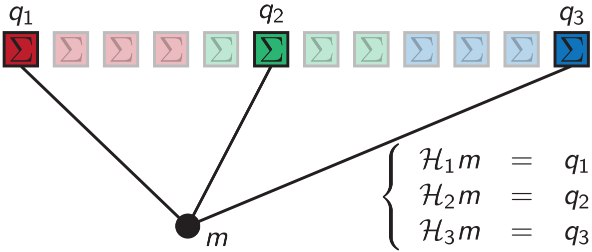
$$\begin{cases} \mathcal{H}_1 m = q_1 \\ \mathcal{H}_2 m = q_2 \\ \mathcal{H}_3 m = q_3 \end{cases}$$

- Theorem: Both constructions are equivalent.

  Proof: By construction, all rows of $\mathcal{H}_i$ are linearly independent. □
- Reduces to LDPC decoding analysis.
  - Error correcting code design (Luby et al. 2001).
  - FFAST (Sparse FFT algorithm) (Pawar & Ramchandran 2013).

Balls-and-bins model

Balls-and-bins model



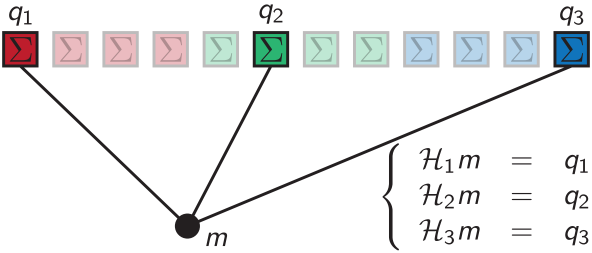
$$\begin{cases} \mathcal{H}_1 m = q_1 \\ \mathcal{H}_2 m = q_2 \\ \mathcal{H}_3 m = q_3 \end{cases}$$

- Theorem: Both constructions are equivalent.

  Proof: By construction, all rows of $\mathcal{H}_i$ are linearly independent. □
- Reduces to LDPC decoding analysis.
  - Error correcting code design (Luby et al. 2001).
  - FFAST (Sparse FFT algorithm) (Pawar & Ramchandran 2013).

Balls-and-bins model

Balls-and-bins model

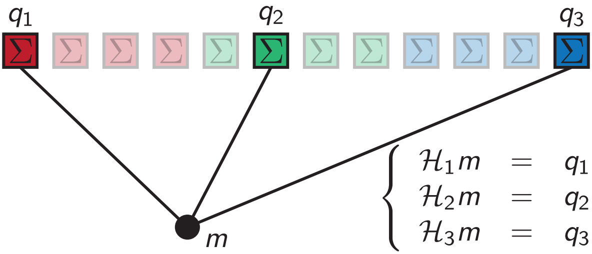


$$\begin{cases} \mathcal{H}_1 m = q_1 \\ \mathcal{H}_2 m = q_2 \\ \mathcal{H}_3 m = q_3 \end{cases}$$

- **Theorem:** Both constructions are equivalent.
  Proof: By construction, all rows of $\mathcal{H}_i$ are linearly independent. □
- Reduces to LDPC decoding analysis.
  - Error correcting code design (Luby et al. 2001).
  - FFAST (Sparse FFT algorithm) (Pawar & Ramchandran 2013).

Balls-and-bins model

Balls-and-bins model

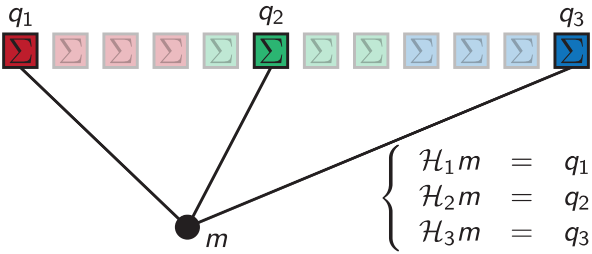


$$\begin{cases} \mathcal{H}_1 m = q_1 \\ \mathcal{H}_2 m = q_2 \\ \mathcal{H}_3 m = q_3 \end{cases}$$

- **Theorem:** Both constructions are equivalent.
  Proof: By construction, all rows of $\mathcal{H}_i$ are linearly independent. □
- Reduces to LDPC decoding analysis.
  - Error correcting code design (Luby et al. 2001).
  - FFAST (Sparse FFT algorithm) (Pawar & Ramchandran 2013).

Balls-and-bins model

Balls-and-bins model



$$\begin{cases} \mathcal{H}_1 m = q_1 \\ \mathcal{H}_2 m = q_2 \\ \mathcal{H}_3 m = q_3 \end{cases}$$

- Theorem: Both constructions are equivalent.
  Proof: By construction, all rows of $\mathcal{H}_i$ are linearly independent. □
- Reduces to LDPC decoding analysis.
  - Error correcting code design (Luby et al. 2001).
  - FFAST (Sparse FFT algorithm) (Pawar & Ramchandran 2013).

Extension to less-sparse regime

- $K = O(N^\alpha)$, $2/3 \le \alpha < 1$.
- *Balls-and-bins* model not equivalent anymore.
- Let $\alpha = 1 - \frac{1}{C}$. Construct $\mathcal{H}_1, \ldots, \mathcal{H}_C$,
- By construction: $\mathcal{N}(\mathcal{H}_i) \bigcap \mathcal{N}(\mathcal{H}_j) = 0$.

Extension to less-sparse regime

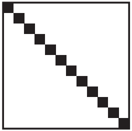
- $K = O(N^{\alpha})$, $2/3 \leq \alpha < 1$.
- *Balls-and-bins* model not equivalent anymore.
- Let $\alpha = 1 - \frac{1}{C}$. Construct $\mathcal{H}_1, \ldots, \mathcal{H}_C$,
- By construction: $\mathcal{N}(\mathcal{H}_i) \bigcap \mathcal{N}(\mathcal{H}_j) = 0$.

Extension to less-sparse regime

- $K = O(N^{\alpha})$, $2/3 \le \alpha < 1$.
- *Balls-and-bins* model not equivalent anymore.
- Let $\alpha = 1 - \frac{1}{C}$. Construct $\mathcal{H}_1, \ldots, \mathcal{H}_C$,
- By construction: $\mathcal{N}(\mathcal{H}_i) \bigcap \mathcal{N}(\mathcal{H}_j) = 0$.

Extension to less-sparse regime

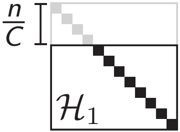
- $K = O(N^{\alpha})$, $2/3 \leq \alpha < 1$.
- *Balls-and-bins* model not equivalent anymore.
- Let $\alpha = 1 - \frac{1}{C}$. Construct $\mathcal{H}_1, \ldots, \mathcal{H}_C$,



- By construction: $\mathcal{N}(\mathcal{H}_i) \bigcap \mathcal{N}(\mathcal{H}_j) = 0$.

Extension to less-sparse regime

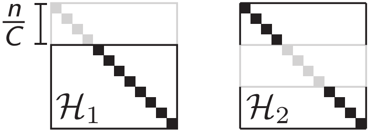
- $K = O(N^{\alpha})$, $2/3 \leq \alpha < 1$.
- *Balls-and-bins* model not equivalent anymore.
- Let $\alpha = 1 - \frac{1}{C}$. Construct $\mathcal{H}_1, \ldots, \mathcal{H}_C$,



- By construction: $\mathcal{N}(\mathcal{H}_i) \bigcap \mathcal{N}(\mathcal{H}_j) = 0$.

Extension to less-sparse regime

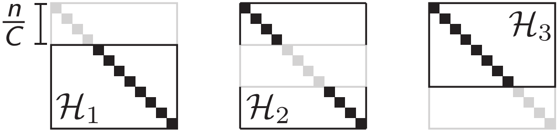
- $K = O(N^{\alpha})$, $2/3 \leq \alpha < 1$.
- *Balls-and-bins* model not equivalent anymore.
- Let $\alpha = 1 - \frac{1}{C}$. Construct $\mathcal{H}_1, \ldots, \mathcal{H}_C$,



- By construction: $\mathcal{N}(\mathcal{H}_i) \bigcap \mathcal{N}(\mathcal{H}_j) = 0$.

Extension to less-sparse regime

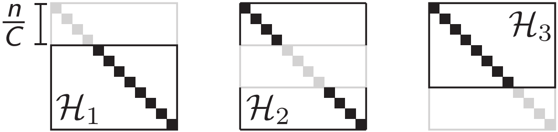
- $K = O(N^{\alpha})$, $2/3 \leq \alpha < 1$.
- *Balls-and-bins* model not equivalent anymore.
- Let $\alpha = 1 - \frac{1}{C}$. Construct $\mathcal{H}_1, \ldots, \mathcal{H}_C$,



- By construction: $\mathcal{N}(\mathcal{H}_i) \bigcap \mathcal{N}(\mathcal{H}_j) = 0$.

Extension to less-sparse regime

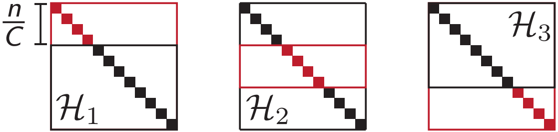
- $K = O(N^\alpha)$, $2/3 \leq \alpha < 1$.
- *Balls-and-bins* model not equivalent anymore.
- Let $\alpha = 1 - \frac{1}{C}$. Construct $\mathcal{H}_1, \ldots, \mathcal{H}_C$,



- By construction: $\mathcal{N}(\mathcal{H}_i) \bigcap \mathcal{N}(\mathcal{H}_j) = 0$.

Extension to less-sparse regime

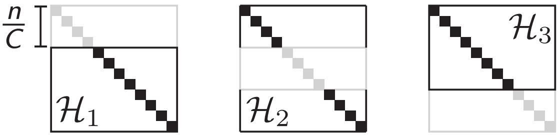
- $K = O(N^{\alpha})$, $2/3 \leq \alpha < 1$.
- *Balls-and-bins* model not equivalent anymore.
- Let $\alpha = 1 - \frac{1}{C}$. Construct $\mathcal{H}_1, \ldots, \mathcal{H}_C$,



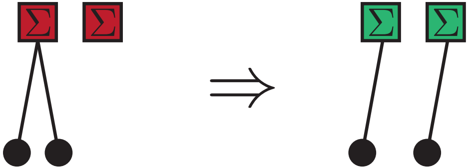
- By construction: $\mathcal{N}(\mathcal{H}_i) \bigcap \mathcal{N}(\mathcal{H}_j) = 0$.

Extension to less-sparse regime

- $K = O(N^\alpha)$, $2/3 \leq \alpha < 1$.
- *Balls-and-bins* model not equivalent anymore.
- Let $\alpha = 1 - \frac{1}{C}$. Construct $\mathcal{H}_1, \ldots, \mathcal{H}_C$,

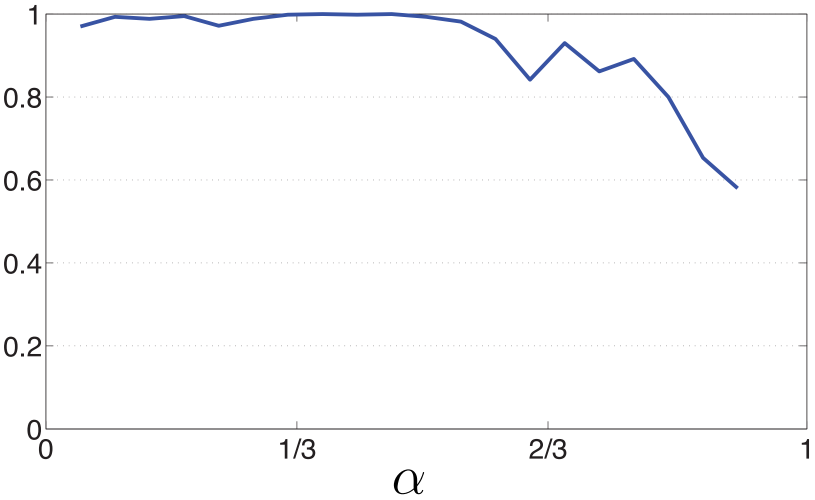


- By construction: $\mathcal{N}(\mathcal{H}_i) \bigcap \mathcal{N}(\mathcal{H}_j) = 0$.

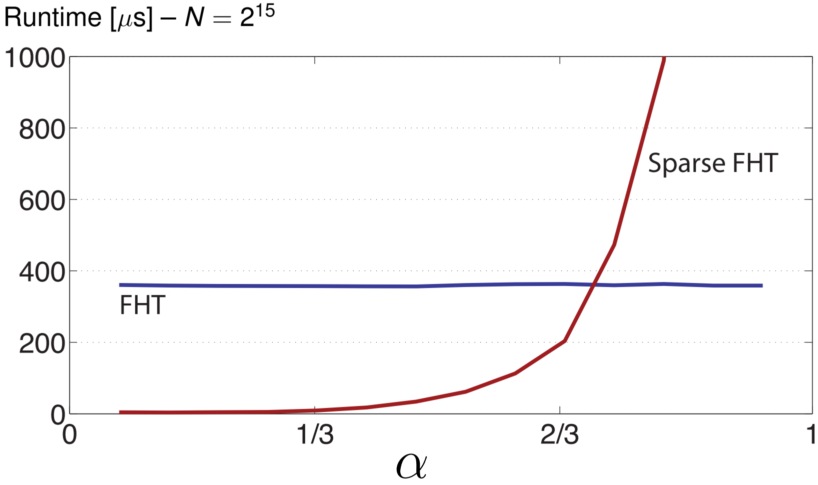


SparseFHT – Probability of success

Probability of success - $N = 2^{22}$



SparseFHT vs. FHT



Conclusion

Contribution

- Sparse fast Hadamard algorithm.
- Time complexity $O(K \log_2 K \log_2 \frac{N}{K})$.
- Sample complexity $O(K \log_2 \frac{N}{K})$.
- Probability of success asymptotically equal to 1.

What's next ?

- Investigate noisy case.

Conclusion

Contribution

- Sparse fast Hadamard algorithm.
- Time complexity $O(K \log_2 K \log_2 \frac{N}{K})$.
- Sample complexity $O(K \log_2 \frac{N}{K})$.
- Probability of success asymptotically equal to 1.

What's next ?

- Investigate noisy case.

Thanks for your attention!

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Code and figures available at

http://lcav.epfl.ch/page-99903.html