Locality preserving hash functions, a partial order and tiles in binary space

Victor S. Miller
Joint with Don Coppersmith, Dan Gordon and Peter Ostapenko

IDA Center for Communications Research, Princeton, NJ

29 April, 2021

Finding two needles in a haystack

The problem

- Given a list $L = (x^{(1)}, \dots, x^{(N)})$ of n-bit bit strings.

Finding two needles in a haystack

The problem

- Given a list $L = (x^{(1)}, \ldots, x^{(N)})$ of n-bit bit strings.
- Plant a pair $(x^{(i)}, x^{(j)})$ where, Hamming distance (number of bits that disagree) $d_H(x^{(i)}, x^{(j)}) = k \ll n$, and everything else is random.

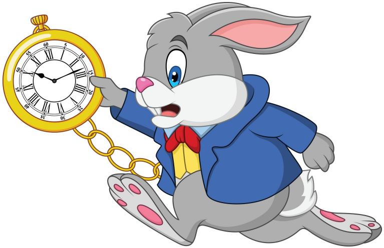
Finding two needles in a haystack

The problem

- Given a list $L = (x^{(1)}, \ldots, x^{(N)})$ of n-bit bit strings.
- Plant a pair $(x^{(i)}, x^{(j)})$ where, Hamming distance (number of bits that disagree) $d_H(x^{(i)}, x^{(j)}) = k \ll n$, and everything else is random.
- Find $x^{(i)}, x^{(j)}$.

Have we gone down a “Rabbit Hole”?

- The isoperimetric inequality for the Hamming Cube.
- Syndrome Decoding.
- An interesting partial order.
- Discrete tiles in a binary space.
- Fast Hadamard Transform.
- Linear Programming.
- Bin Packing.



A first attempt

Comments

- $x^{(i)}$ are random: $d_H(x^{(i)}, x^{(j)}) \approx n/2$. Any "hit" is not spurious.

A first attempt

Comments

- $x^{(i)}$ are random: $d_H(x^{(i)}, x^{(j)}) \approx n/2$. Any "hit" is not spurious.
- *Exhaustion*: Try all pairs.
- Work is $N(N-1)/2$.
- For $N = 10^9$ that's a lot of work.

A first attempt

Comments

- $x^{(i)}$ are random: $d_H(x^{(i)}, x^{(j)}) \approx n/2$. Any "hit" is not spurious.
- *Exhaustion*: Try all pairs.
- Work is $N(N-1)/2$.
- For $N = 10^9$ that's a lot of work.
- Can we do better?

A better (?) idea

- Use a "hash" function $f : \mathbb{B}^n \to \mathbb{B}^r$.
- Put string x into the "bucket" labeled with $f(x)$.

A better (?) idea

- Use a "hash" function $f : \mathbb{B}^n \to \mathbb{B}^r$.
- Put string x into the "bucket" labeled with $f(x)$.
- Only compare bitstrings in the same bucket.

A better (?) idea

- Use a “hash” function $f : \mathbb{B}^n \to \mathbb{B}^r$.
- Put string x into the “bucket” labeled with $f(x)$.
- Only compare bitstrings in the same bucket.
- Cuts down number of comparisons by a factor of 2^{n-r}.

A better (?) idea

- Use a "hash" function $f : \mathbb{B}^n \to \mathbb{B}^r$.
- Put string x into the "bucket" labeled with $f(x)$.
- Only compare bitstrings in the same bucket.
- Cuts down number of comparisons by a factor of 2^{n-r}.
- Call sought for pair w and $\tilde{w}$.
- Will work well if probability of of w and $\tilde{w}$ being in the same bucket is large enough.

A better (?) idea

- Use a "hash" function $f : \mathbb{B}^n \to \mathbb{B}^r$.
- Put string x into the "bucket" labeled with $f(x)$.
- Only compare bitstrings in the same bucket.
- Cuts down number of comparisons by a factor of 2^{n-r}.
- Call sought for pair w and $\tilde{w}$.
- Will work well if probability of of w and $\tilde{w}$ being in the same bucket is large enough.
- Loses certainty.

A better (?) idea

- Use a "hash" function $f : \mathbb{B}^n \to \mathbb{B}^r$.
- Put string x into the "bucket" labeled with $f(x)$.
- Only compare bitstrings in the same bucket.
- Cuts down number of comparisons by a factor of 2^{n-r}.
- Call sought for pair w and $\tilde{w}$.
- Will work well if probability of of w and $\tilde{w}$ being in the same bucket is large enough.
- Loses certainty.
- If we fail try another hash function.

A better (?) idea

- Use a "hash" function $f : \mathbb{B}^n \to \mathbb{B}^r$.
- Put string x into the "bucket" labeled with $f(x)$.
- Only compare bitstrings in the same bucket.
- Cuts down number of comparisons by a factor of 2^{n-r}.
- Call sought for pair w and $\tilde{w}$.
- Will work well if probability of of w and $\tilde{w}$ being in the same bucket is large enough.
- Loses certainty.
- If we fail try another hash function.
- *Question:* what are the best f to use?

Channels and Codes

- The *Binary Symmetric Channel* BSC(p) takes each bit and flips it independently with probability p.

Channels and Codes

- The *Binary Symmetric Channel* $\mathrm{BSC}(p)$ takes each bit and flips it independently with probability p.
- Pass a bitstring x through $\mathrm{BSC}(p)$ to get $\tilde{x}$.

Channels and Codes

- The *Binary Symmetric Channel* BSC(p) takes each bit and flips it independently with probability p.
- Pass a bitstring x through BSC(p) to get $\tilde{x}$.
- A *code* $S \subset \mathbb{B}^n$ is a subset of bit strings.

Channels and Codes

- The *Binary Symmetric Channel* $\mathrm{BSC}(p)$ takes each bit and flips it independently with probability p.
- Pass a bitstring x through $\mathrm{BSC}(p)$ to get $\tilde{x}$.
- A *code* $S \subset \mathbb{B}^n$ is a subset of bit strings.
- *Error detecting*: if $x \in S$ and $\tilde{x} \notin S$ we've detected an error.

Channels and Codes

- The *Binary Symmetric Channel* BSC(p) takes each bit and flips it independently with probability p.
- Pass a bitstring x through BSC(p) to get $\tilde{x}$.
- A *code* $S \subset \mathbb{B}^n$ is a subset of bit strings.
- *Error detecting*: if $x \in S$ and $\tilde{x} \notin S$ we've detected an error.
- *Error correcting*: $x \in S$, find $\hat{x}$ "closest" to $\tilde{x}$.

Probability of disagreement

- $\mathcal{F}_S(p)$: probability if $x \in S$ is random that $\tilde{x} \in S$.

$$F_S(t) := \sum_{i=0}^{n} A_i(S)t^i,\ A_i(S) := \#\{x, y \in S : d_H(x, y) = i\}$$

$$\mathcal{F}_S(p) := \frac{1}{|S|}(1-p)^n F_S(p/(1-p))$$

Probability of disagreement

- $\mathcal{F}_S(p)$: probability if $x \in S$ is random that $\tilde{x} \in S$.

$$F_S(t) := \sum_{i=0}^{n} A_i(S) t^i,\ A_i(S) := \#\{x, y \in S : d_H(x, y) = i\}$$

$$\mathcal{F}_S(p) := \frac{1}{|S|}(1-p)^n F_S(p/(1-p))$$

- Error detecting: probability of failure. We want to *minimize*.

Probability of disagreement

- $\mathcal{F}_S(p)$: probability if $x \in S$ is random that $\tilde{x} \in S$.

$$F_S(t) := \sum_{i=0}^{n} A_i(S) t^i, \; A_i(S) := \#\{x, y \in S : d_H(x, y) = i\}$$

$$\mathcal{F}_S(p) := \frac{1}{|S|}(1-p)^n F_S(p/(1-p))$$

- Error detecting: probability of failure. We want to *minimize*.
- $f^{-1}(b)$ is set of elements in bucket labeled by $b \in \mathbb{B}^r$.

$$P(f) := \Pr(f(X) = f(\tilde{X})) = \sum_{b \in \mathbb{B}^r} \mathcal{F}_{f^{-1}(b)}(p).$$

Probability of disagreement

- $\mathcal{F}_S(p)$: probability if $x \in S$ is random that $\tilde{x} \in S$.

$$F_S(t) := \sum_{i=0}^{n} A_i(S)t^i,\ A_i(S) := \#\{x, y \in S : d_H(x, y) = i\}$$
$$\mathcal{F}_S(p) := \frac{1}{|S|}(1-p)^n F_S(p/(1-p))$$

- Error detecting: probability of failure. We want to *minimize*.
- $f^{-1}(b)$ is set of elements in bucket labeled by $b \in \mathbb{B}^r$.

$$P(f) := \Pr(f(X) = f(\tilde{X})) = \sum_{b \in \mathbb{B}^r} \mathcal{F}_{f^{-1}(b)}(p).$$

- *Goal*: Find $S \subset \mathbb{B}^n$, $|S| = 2^{n-r}$ which *maximizes* $\mathcal{F}_S(p)$.

Equivalence

- $\sigma \in \mathfrak{S}_n$: a permutation.
- $\sigma(x)$ permutes the coordinates of x.
- Note: $F_{\sigma(S)\oplus a}(p) = F_S(p)$, where $a \in \mathbb{B}^n$, $\oplus$ is mod 2 addition of coordinates.
- We will say that S and $\sigma(S) \oplus a$ are *isomorphic*.
- Thus $P(f_{\sigma,a}) = P(f)$ where $f_{\sigma,a}(x) = f(\sigma(x) \oplus a)$.
- Note: If S is "good" we can define a hash function f from it if it's a *tile*: $\mathbb{B}^n$ is a disjoint union of translates of S using $\oplus$.
- Index translates by elements of $\mathbb{B}^r$, map x to index of translate containing it.

The question I was asked

- Projection: $\pi : \mathbb{B}^n \to \mathbb{B}^r$ be $\pi((x_1, \dots, x_n)) = (x_1, \dots, x_r)$.
- *Question*: Can we do better than using π?
- Answer: It depends on p.

The isoperimetric theorem for the Hamming Cube

Theorem (Isoperimetric Theorem (Harper))

If $S \subset \mathbb{B}^n$*, let* $e(S) = \#\{x \in S, y \notin S : d_H(x, y) = 1\}$*. Then*

$$e(S) \geq \frac{1}{2}|S| \log_2 |S|,$$

with equality if and only if S *is isomorphic to* $(*, \ldots, *, 0, \ldots, 0)$*, a subcube.*

The isoperimetric theorem for the Hamming Cube

Theorem (Isoperimetric Theorem (Harper))

If $S \subset \mathbb{B}^n$*, let* $e(S) = \#\{x \in S, y \notin S : d_H(x, y) = 1\}$*. Then*

$$e(S) \geq \frac{1}{2}|S| \log_2 |S|,$$

with equality if and only if S *is isomorphic to* $(*, \ldots, *, 0, \ldots, 0)$*, a subcube.*

Theorem

Projection is best if $p \leq 2^{-2(n-r)}$*.*

The isoperimetric theorem for the Hamming Cube

Theorem (Isoperimetric Theorem (Harper))

If $S \subset \mathbb{B}^n$*, let* $e(S) = \#\{x \in S, y \notin S : d_H(x, y) = 1\}$*. Then*

$$e(S) \geq \frac{1}{2}|S| \log_2 |S|,$$

with equality if and only if S *is isomorphic to* $(*, \ldots, *, 0, \ldots, 0)$*, a subcube.*

Theorem

Projection is best if $p \leq 2^{-2(n-r)}$*.*

Proof.

Use the isoperimetric inequality for the Hamming cube. Note that $A_0(S) = |S|, A_1(S) = n|S| - e(S)$. □

Doing better than projection

- $C \subset \mathbb{B}^n$: a *linear* subspace of dimension r.

Doing better than projection

- $C \subset \mathbb{B}^n$: a *linear* subspace of dimension r.
- *Check matrix*: A: $x \in C \Leftrightarrow Ax = 0$.

Doing better than projection

- $C \subset \mathbb{B}^n$: a *linear* subspace of dimension r.
- *Check matrix*: A: $x \in C \Leftrightarrow Ax = 0$.
- *Syndrome*: Given $\tilde{x}$ calculate $A\tilde{x}$. Gives the coset of C containing $\tilde{x}$.

Doing better than projection

- $C \subset \mathbb{B}^n$: a *linear* subspace of dimension r.
- *Check matrix*: A: $x \in C \Leftrightarrow Ax = 0$.
- *Syndrome*: Given $\tilde{x}$ calculate $A\tilde{x}$. Gives the coset of C containing $\tilde{x}$.
- For each coset $a \oplus C$ give $y \in a \oplus C$ of minimum Hamming weight: *Coset Leader*.

Doing better than projection

- $C \subset \mathbb{B}^n$: a *linear* subspace of dimension r.
- *Check matrix*: A: $x \in C \Leftrightarrow Ax = 0$.
- *Syndrome*: Given $\tilde{x}$ calculate $A\tilde{x}$. Gives the coset of C containing $\tilde{x}$.
- For each coset $a \oplus C$ give $y \in a \oplus C$ of minimum Hamming weight: *Coset Leader*.
- Use the set of coset leaders as a region S.

Doing better than projection

- $C \subset \mathbb{B}^n$: a *linear* subspace of dimension r.
- *Check matrix*: A: $x \in C \Leftrightarrow Ax = 0$.
- *Syndrome*: Given $\tilde{x}$ calculate $A\tilde{x}$. Gives the coset of C containing $\tilde{x}$.
- For each coset $a \oplus C$ give $y \in a \oplus C$ of minimum Hamming weight: *Coset Leader*.
- Use the set of coset leaders as a region S.
- Theorem: Asymptotically this beats projection for a random code of fixed rate.

Doing better than projection

- $C \subset \mathbb{B}^n$: a *linear* subspace of dimension r.
- *Check matrix*: A: $x \in C \Leftrightarrow Ax = 0$.
- *Syndrome*: Given $\tilde{x}$ calculate $A\tilde{x}$. Gives the coset of C containing $\tilde{x}$.
- For each coset $a \oplus C$ give $y \in a \oplus C$ of minimum Hamming weight: *Coset Leader*.
- Use the set of coset leaders as a region S.
- Theorem: Asymptotically this beats projection for a random code of fixed rate.

For the Golay code $\mathcal{G}$

$$F_{\mathcal{G}}(t) := 2048 + 11684t + 128524t^2 + 226688t^3,$$

Better than projection when $p \geq 0.2555$.

Optimal Regions

Definition (Optimal Region)

Let $S \subset \mathbb{B}^n$. Say that S is *optimal* at $t \in (0,1)$ if $F_S(t) \geq F_{S'}(t)$ for all $S' \subset \mathbb{B}^n, |S'| = |S|$.

S is *optimal* if it is optimal at some $t \in (0,1)$.

Optimal Regions

Definition (Optimal Region)

Let $S \subset \mathbb{B}^n$. Say that S is *optimal* at $t \in (0,1)$ if $F_S(t) \geq F_{S'}(t)$ for all $S' \subset \mathbb{B}^n, |S'| = |S|$.

S is *optimal* if it is optimal at some $t \in (0,1)$.

Theorem (Optimal Region Theorem (Gordon, Miller, Ostapenko))

An optimal subset $S \subset \mathbb{B}^n$ is isomorphic to an order ideal in the partial order $\preccurlyeq_R$ (defined below).

Proof.

Uses the "shifting" and "compression" functions of Erdős-Ko-Rado from extremal set theory. Looks at local failures to be an order ideal, and corrects them. □

A partial order

- S is a set, $\preccurlyeq$ is a *partial order* on S.

A partial order

- S is a set, $\preccurlyeq$ is a *partial order* on S.
- *Reflexive*: $x \preccurlyeq x$ for all x.
- *Antisymmetric*: $x \preccurlyeq y, y \preccurlyeq x \Rightarrow x = y$.
- *Transitive*: $x \preccurlyeq y, y \preccurlyeq z \Rightarrow x \preccurlyeq z$.
- *Note*: Not every pair $x, y \in S$ may be comparable.

A partial order

- S is a set, $\preccurlyeq$ is a *partial order* on S.
- *Reflexive*: $x \preccurlyeq x$ for all x.
- *Antisymmetric*: $x \preccurlyeq y, y \preccurlyeq x \Rightarrow x = y$.
- *Transitive*: $x \preccurlyeq y, y \preccurlyeq z \Rightarrow x \preccurlyeq z$.
- *Note*: Not every pair $x, y \in S$ may be comparable.
- $(S, \preccurlyeq)$ is called a *poset*.

A partial order

- S is a set, $\preccurlyeq$ is a *partial order* on S.
- *Reflexive*: $x \preccurlyeq x$ for all x.
- *Antisymmetric*: $x \preccurlyeq y, y \preccurlyeq x \Rightarrow x = y$.
- *Transitive*: $x \preccurlyeq y, y \preccurlyeq z \Rightarrow x \preccurlyeq z$.
- *Note*: Not every pair $x, y \in S$ may be comparable.
- $(S, \preccurlyeq)$ is called a *poset*.
- Identify bitstring $x \in \mathbb{B}^n$ with a subset of $\{0, \ldots, n-1\}$, $I(x)$ of positions of 1 bits.
- $T_{(i)} := i^{\text{th}}$ largest element of T

A partial order

- S is a set, $\preccurlyeq$ is a *partial order* on S.
- *Reflexive*: $x \preccurlyeq x$ for all x.
- *Antisymmetric*: $x \preccurlyeq y, y \preccurlyeq x \Rightarrow x = y$.
- *Transitive*: $x \preccurlyeq y, y \preccurlyeq z \Rightarrow x \preccurlyeq z$.
- *Note*: Not every pair $x, y \in S$ may be comparable.
- $(S, \preccurlyeq)$ is called a *poset*.
- Identify bitstring $x \in \mathbb{B}^n$ with a subset of $\{0, \dots, n-1\}$, $I(x)$ of positions of 1 bits.
- $T_{(i)} := i^{\text{th}}$ largest element of T
- Define: $x \preccurlyeq_R y$ if

$$I(x)_{(1)} \leq I(y)_{(1)}, \dots, I(x)_{(k)} \leq I(y)_{(k)},$$

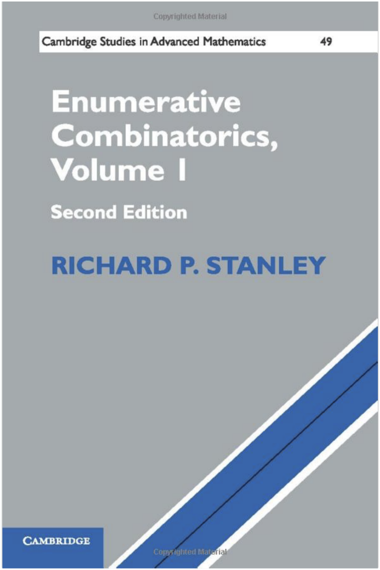
where $k = \min(|I(x)|, |I(y)|)$.

What's in a name?

The partial order $\preccurlyeq_R$ has many names.

- Kündgen: *right-shifted partial order*
- Stanley, Proctor (and others): $M(n)$ (the poset name).
- Ahlswede, Tamm: pushing order.

Has many interesting connections: partitions, Coxeter Groups.



Order Ideals

- *Order Ideal*: A subset $T \subset S$ where $x \in S, y \preccurlyeq x \Rightarrow y \in S$.
- Generators: $T \subset S$. $\langle T \rangle := \{x \in S : \exists y \in T, x \preccurlyeq y\}$.
- *Principal ideal*: $\langle \{x\} \rangle$: one generator.

Finding all order ideals of a given size

- Squire: a recursion to find all order ideals of a poset.
- Number of all ideals grows too quickly, but we're only interested those of limited size.

Finding all order ideals of a given size

- Squire: a recursion to find all order ideals of a poset.
- Number of all ideals grows too quickly, but we're only interested those of limited size.

Principal ideals of size n in $M(n)$

1, 1, 2, 1, 2, 2, 3, 1, 3, 2, 3, 2, 3, 3, 6, 1, 2, 3, 4, 2, 6, 2, 4, 3, 5, 2, 6, 3, 4, 5, 7, 1, 4, 3, 6, 4, 5, 2, 7, 3, 4, 5, 7, 3, 8, 2, 6, 2, 6, 4, 8, 3, 4, 5, 11, 4, 7, 3, 6, 5, 6, 4, 15

Finding all order ideals of a given size

- Squire: a recursion to find all order ideals of a poset.
- Number of all ideals grows too quickly, but we're only interested those of limited size.

Principal ideals of size n in $M(n)$

1, 1, 2, 1, 2, 2, 3, 1, 3, 2, 3, 2, 3, 3, 6, 1, 2, 3, 4, 2, 6, 2, 4, 3, 5, 2, 6, 3, 4, 5, 7, 1, 4, 3, 6, 4, 5, 2, 7, 3, 4, 5, 7, 3, 8, 2, 6, 2, 6, 4, 8, 3, 4, 5, 11, 4, 7, 3, 6, 5, 6, 4, 15

Order ideals of size n in $M(n)$: A274312. $\approx 2.06372 \cdot 1.259305361.29232158^n$

$\boxed{1}$, $\boxed{1}$, 1, $\boxed{2}$, 2, 3, 4, $\boxed{6}$, 7, 10, 13, 18, 23, 31, 40, $\boxed{54}$, 69, 91, 118, 155, 199, 260, 334, 433, 555, 717, 917, 1180, 1506, 1929, 2458, $\boxed{3140}$, 3990, 5081, 6445, 8185, 10361, 13125, 16581, 20956, 26424, 33322, 41940, 52782, 66312, 83293, 104467, 130979, . . . , $\boxed{4384627}$.

Finding small optimal regions

- Find all order ideals in $M(n)$ of sizes $s = 2, 4, 8, 16, 32, 64$.
- Calculate corresponding $F_S(t)$ polynomials.
- Compare all of them to find optimal regions.

Finding small optimal regions

- Find all order ideals in $M(n)$ of sizes $s = 2, 4, 8, 16, 32, 64$.
- Calculate corresponding $F_S(t)$ polynomials.
- Compare all of them to find optimal regions.
- For $s = 2, 4, 8$ only projection is optimal.
- For $s = 16$: 5 optimal besides projection.
- For $s = 32$: 20 optimal besides projection.
- For $s = 64$: 56 optimal besides projection.

Finding small optimal regions

- Find all order ideals in $M(n)$ of sizes $s = 2, 4, 8, 16, 32, 64$.
- Calculate corresponding $F_S(t)$ polynomials.
- Compare all of them to find optimal regions.
- For $s = 2, 4, 8$ only projection is optimal.
- For $s = 16$: 5 optimal besides projection.
- For $s = 32$: 20 optimal besides projection.
- For $s = 64$: 56 optimal besides projection.
- For all but 10 of the size 64, they are sets of minimal weight coset leaders of a linear code.

The terrible 10

Table: Putative tiles

k	n	generators of V
6	12	$\{11\}, \{10, 5\}, \{9, 8\}$
7	13	$\{12\}, \{10, 4\}, \{9, 8\}$
8	14	$\{13, 2\}, \{13, 1, 0\}, \{3, 2, 0\}$
9	15	$\{14, 1, 0\}, \{10, 2\}$
16	22	$\{21, 1\}$
17	23	$\{22, 0\}, \{19, 1\}$
18	24	$\{23, 0\}, \{17, 1\}$
19	25	$\{24, 0\}, \{15, 1\}$
20	26	$\{25, 0\}, \{13, 1\}$
21	27	$\{26, 0\}, \{11, 1\}$

Tiles in $\mathbb{B}^n$

Definition (Tile)

A subset $S \subseteq \mathbb{B}^n$ is a *tile* if $\mathbb{B}^n$ is covered by disjoint translates of S.

$$\exists A \subseteq \mathbb{B}^n,\ A \oplus S = \mathbb{B}^n,\ \text{uniquely.}$$

The set A is called a *complement* of S. Note: A is also a tile.

Remark

This is equivalent to $A \oplus S = \mathbb{B}^n, (A \oplus A) \cap (S \oplus S) = \{0\}$.

Deciding if a subset is a tile

- $\chi_S(x) = 1$ if $x \in S$, 0 otherwise.

Deciding if a subset is a tile

- $\chi_S(x) = 1$ if $x \in S$, 0 otherwise.
- Convolution: $g \star h(x) := \sum_{y \in \mathbb{B}^n} g(x)h(x \oplus y)$.

Deciding if a subset is a tile

- $\chi_S(x) = 1$ if $x \in S$, 0 otherwise.
- Convolution: $g \star h(x) := \sum_{y \in \mathbb{B}^n} g(x)h(x \oplus y)$.
- S is known, A is unknown.

Deciding if a subset is a tile

- $\chi_S(x) = 1$ if $x \in S$, 0 otherwise.
- Convolution: $g \star h(x) := \sum_{y \in \mathbb{B}^n} g(x)h(x \oplus y)$.
- S is known, A is unknown.
- $\chi_A \star \chi_S(x) = \#\{a \in A, s \in S : x = a \oplus s\}$

Deciding if a subset is a tile

- $\chi_S(x) = 1$ if $x \in S$, 0 otherwise.
- Convolution: $g \star h(x) := \sum_{y \in \mathbb{B}^n} g(x)h(x \oplus y)$.
- S is known, A is unknown.
- $\chi_A \star \chi_S(x) = \#\{a \in A, s \in S : x = a \oplus s\}$
- Hadamard Transform: $\widehat{g}(y) := \sum_{x \in \mathbb{B}^n} g(x)(-1)^{x \cdot y}$.

Deciding if a subset is a tile

- $\chi_S(x) = 1$ if $x \in S$, 0 otherwise.
- Convolution: $g \star h(x) := \sum_{y \in \mathbb{B}^n} g(x)h(x \oplus y)$.
- S is known, A is unknown.
- $\chi_A \star \chi_S(x) = \#\{a \in A, s \in S : x = a \oplus s\}$
- Hadamard Transform: $\widehat{g}(y) := \sum_{x \in \mathbb{B}^n} g(x)(-1)^{x \cdot y}$.
- Convolution $\Rightarrow$ Product: $\widehat{g \star h}(y) = \widehat{g}(y)\widehat{h}(y)$

Deciding if a subset is a tile

- $\chi_S(x) = 1$ if $x \in S$, 0 otherwise.
- Convolution: $g \star h(x) := \sum_{y \in \mathbb{B}^n} g(x)h(x \oplus y)$.
- S is known, A is unknown.
- $\chi_A \star \chi_S(x) = \#\{a \in A, s \in S : x = a \oplus s\}$
- Hadamard Transform: $\widehat{g}(y) := \sum_{x \in \mathbb{B}^n} g(x)(-1)^{x \cdot y}$.
- Convolution $\Rightarrow$ Product: $\widehat{g \star h}(y) = \widehat{g}(y)\widehat{h}(y)$
- Equivalent to: $\widehat{\chi_A}(y)\widehat{\chi_S}(y) = |A|\delta(y)$, ($\delta(y) = 1$ if $y = 0$, $= 0$, otherwise).

Deciding if a subset is a tile

- $\chi_S(x) = 1$ if $x \in S$, 0 otherwise.
- Convolution: $g \star h(x) := \sum_{y \in \mathbb{B}^n} g(x)h(x \oplus y)$.
- S is known, A is unknown.
- $\chi_A \star \chi_S(x) = \#\{a \in A, s \in S : x = a \oplus s\}$
- Hadamard Transform: $\widehat{g}(y) := \sum_{x \in \mathbb{B}^n} g(x)(-1)^{x \cdot y}$.
- Convolution $\Rightarrow$ Product: $\widehat{g \star h}(y) = \widehat{g}(y)\widehat{h}(y)$
- Equivalent to: $\widehat{\chi_A}(y)\widehat{\chi_S}(y) = |A|\delta(y)$, ($\delta(y) = 1$ if $y = 0$, $= 0$, otherwise).
- Integer program: Given a finite set of linear equalities and inequalities with integer variables, find values of variables satisfying all of them.

Necessary and Sufficient equations for a tile

Variables:

$$z_u = \chi_A(u), w_x = \widehat{\chi_A}(x).$$

Conditions:

$$\begin{aligned} 0 \le z_u \le & 1 \text{ and is an integer.} \\ -|A| \le w_x \le & |A| \text{ and is an integer.} \\ w_0 = & |A|. \\ w_x = & 0 \text{ if } x \neq 0 \text{ and } \widehat{\chi_S}(x) \neq 0. \\ w_x = & \sum_u (-1)^{x \cdot u} z_u \text{ for all } x. \end{aligned}$$

Unfortunately too hard for CPLEX (high quality Integer Programming solver).

A relaxation

- Use $(A \oplus A) \cap (S \oplus S) = \{0\}$.
- Use that and equation of Hadamard transform: for $n = 12, 13, 14, 15$ sought for A doesn't exist!

Necessary Equations for a tile

Variables:

$$b_u = \chi_A \star \chi_A(u), c_x = |\widehat{\chi_A}(x)|^2.$$

Conditions:

$$\begin{aligned}
0 \leq b_u \leq & |A| \text{ and is an integer.} \\
0 \leq c_x \leq & |A|^2 \text{ and is the square of an integer.} \\
b_0 = & |A|. \\
c_0 = & |A|^2. \\
b_u = & 0 \text{ if } u \neq 0 \text{ and } \chi_S \star \chi_S(u) \neq 0. \\
c_x = & 0 \text{ if } x \neq 0 \text{ and } \widehat{\chi_S}(x) \neq 0. \\
c_x = & \sum_u (-1)^{x \cdot u} b_u \text{ for all } x.
\end{aligned}$$

A useful trick

- Equations for the Hadamard Transform involve 2^{2n} nonzero coefficients.

A useful trick

- Equations for the Hadamard Transform involve 2^{2n} nonzero coefficients.
- Sparse matrix multiplication: Calculate Ax with # multiplications = # nonzero coefficients in A.

A useful trick

- Equations for the Hadamard Transform involve 2^{2n} nonzero coefficients.
- Sparse matrix multiplication: Calculate Ax with # multiplications = # nonzero coefficients in A.
- If we can write $A = B^{(1)} \cdots B^{(r)}$, and $B^{(i)}$ are sparse it's a win.

A useful trick

- Equations for the Hadamard Transform involve 2^{2n} nonzero coefficients.
- Sparse matrix multiplication: Calculate Ax with # multiplications = # nonzero coefficients in A.
- If we can write $A = B^{(1)} \cdots B^{(r)}$, and $B^{(i)}$ are sparse it's a win.
- Fast Hadamard Transform: $H = B^{(1)} \cdots B^{(n)}$, where # nonzeros in $B^{(i)}$ is only 2^n.

A useful trick

- Equations for the Hadamard Transform involve 2^{2n} nonzero coefficients.
- Sparse matrix multiplication: Calculate Ax with # multiplications = # nonzero coefficients in A.
- If we can write $A = B^{(1)} \cdots B^{(r)}$, and $B^{(i)}$ are sparse it's a win.
- Fast Hadamard Transform: $H = B^{(1)} \cdots B^{(n)}$, where # nonzeros in $B^{(i)}$ is only 2^n.
- Introduce extra variables for intermediate products.

A useful trick

- Equations for the Hadamard Transform involve 2^{2n} nonzero coefficients.
- Sparse matrix multiplication: Calculate Ax with # multiplications = # nonzero coefficients in A.
- If we can write $A = B^{(1)} \cdots B^{(r)}$, and $B^{(i)}$ are sparse it's a win.
- Fast Hadamard Transform: $H = B^{(1)} \cdots B^{(n)}$, where # nonzeros in $B^{(i)}$ is only 2^n.
- Introduce extra variables for intermediate products.
- Makes the problems for $n = 12, 13, 14, 15$ small enough for CPLEX. Others are still too big.

Pieces and Bins

- X: linear subspace of $\mathbb{B}^n$.
- Intersect S with cosets of X: pieces.
- $\#((a \oplus S) \cap (b \oplus X)) = \#(S \cap ((a \oplus b) \oplus X))$.
- Must use all pieces to cover cosets of X.
- Can't make it work for $n = 12, 13$ but can for all others.

k	r	bin size	piece census
8	3	8	10*5, 1*6, 1*8
9	3	8	4*4, 8*5, 1*8
16	2	4	20*3, 1*4
17	2	4	3*2, 18*3, 1*4
18	2	4	6*2, 16*3, 1*4
19	2	4	9*2, 14*3, 1*4
20	2	4	12*2, 12*3, 1*4
21	2	4	15*2, 10*3, 1*4

Things to do

- Prove asymptotics for # order ideals of a given size in $M(n)$.

Things to do

- Prove asymptotics for # order ideals of a given size in $M(n)$.
- Characterize those ideals yielding optimal regions.

Things to do

- Prove asymptotics for # order ideals of a given size in $M(n)$.
- Characterize those ideals yielding optimal regions.
- Better formulation for linear programming proof of non-tileability.

Things to do

- Prove asymptotics for # order ideals of a given size in $M(n)$.
- Characterize those ideals yielding optimal regions.
- Better formulation for linear programming proof of non-tileability.
- When does bin packing work?

Things to do

- Prove asymptotics for # order ideals of a given size in $M(n)$.
- Characterize those ideals yielding optimal regions.
- Better formulation for linear programming proof of non-tileability.
- When does bin packing work?
- Can we combine the two ideas?

Things to do

- Prove asymptotics for # order ideals of a given size in $M(n)$.
- Characterize those ideals yielding optimal regions.
- Better formulation for linear programming proof of non-tileability.
- When does bin packing work?
- Can we combine the two ideas?
- Ultimate goal: good characterization of those ideals yielding tiles.

References I

Bollobás, Béla and Imre Leader (1991). "Compressions and isoperimetric inequalities". In: *Journal of Combinatorial Theory, Series A* 56.1, pp. 47–62.

Cohen, Gerard, Simon Litsyn, Alexander Vardy, and Gilles Zémor (1996). "Tilings of binary spaces". In: *SIAM Journal on Discrete Mathematics* 9.3, pp. 393–412.

Coppersmith, Don and Victor S Miller (2012). "Binary Nontiles". In: *SIAM Journal on Discrete Mathematics* 26.1, pp. 30–38.

Erdos, P, Chao Ko, and R Rado (1961). "Intersection theorems for systems of finite sets". In: *Quart. J. Math. Oxford* 12, pp. 313–320.

Frankl, Peter (1987). "The shifting technique in extremal set theory". In: *London Math. Soc. Lecture Note Series*. Surveys in combinatorics 1987 123, pp. 81–110.

Gordon, Daniel M, Victor S Miller, and Peter Ostapenko (2010). "Optimal Hash Functions for Approximate Matches on the *n*-Cube". In: *IEEE transactions on information theory* 56.3, pp. 984–991.

Harper, Lawrence Hueston (1964). "Optimal assignments of numbers to vertices". In: *Journal of the Society for Industrial and Applied Mathematics* 12.1, pp. 131–135.

Katona, Gyula (1964). "Intersection theorems for systems of finite sets". In: *Acta Mathematica Academiae Scientiarum Hungaricae* 15.3-4, pp. 329–337.

Kündgen, André (2002). "Minimum average distance subsets in the Hamming cube". In: *Discrete mathematics* 249.1-3, pp. 149–165.

Squire, Matthew B (1995). "Enumerating the ideals of a poset". In: DOI: `10.1.1.22.1919`. URL: `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.22.1919`.