An Algorithmic Reduction Theory for Binary Codes

Léo Ducas, CWI.

Joint work with

Thomas Debris-Alazard & Wessel van Woerden

Overview

This work

- Propose analogues from lattices to binary codes (Defs, Algs, Bounds)

- ? Use it to speed-up Cryptanalytic algorithms (code-based cryptography)

Overview

This work

- Propose analogues from lattices to binary codes (Defs, Algs, Bounds)

? Use it to speed-up Cryptanalytic algorithms (code-based cryptography)

This talk

- Recall the LLL alg for Lattices
- Adapt it to Codes

What notion of Orthogonality for binary codes?

Reduction

Find a $\left\{\begin{array}{l}\text{unique}\\ \text{canonical}\\ \text{good}\end{array}\right\}$ representative $x \in X$

of a given class $c \in X/\sim$.

Reduction

Find a $\left\{\begin{matrix}\text{unique}\\ \text{canonical}\\ \text{good}\end{matrix}\right\}$ representative $x \in X$

of a given class $c \in X/\sim$.

Lattice Reduction

Find a good basis $B \in GL_n(\mathbb{R})$

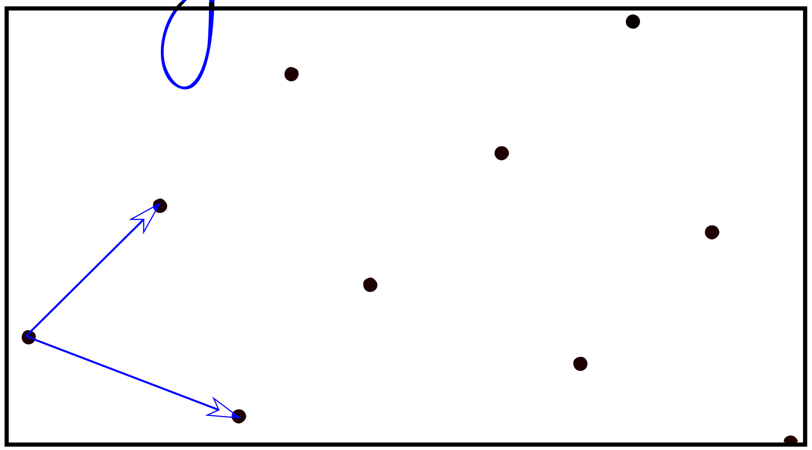
of a lattice $\mathcal{L} \in GL_n(\mathbb{R}) / GL_n(\mathbb{Z})$.

Lattices

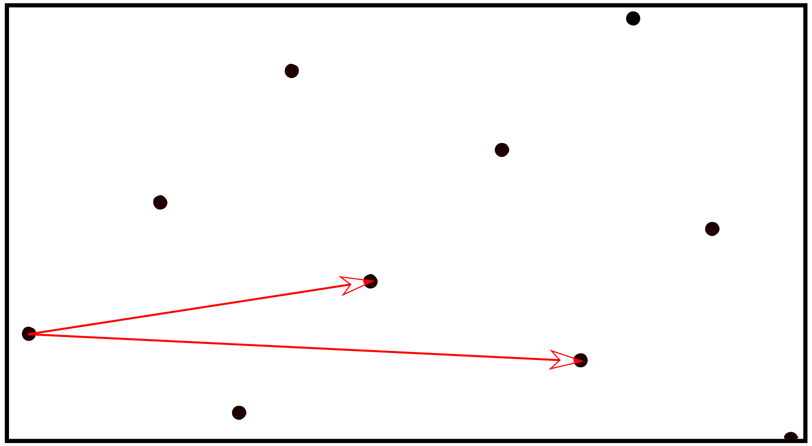
$\mathcal{L} \subseteq \mathbb{R}^n$, a discrete subgroup of a Euclidean Vector Space

Lattices

Good basis

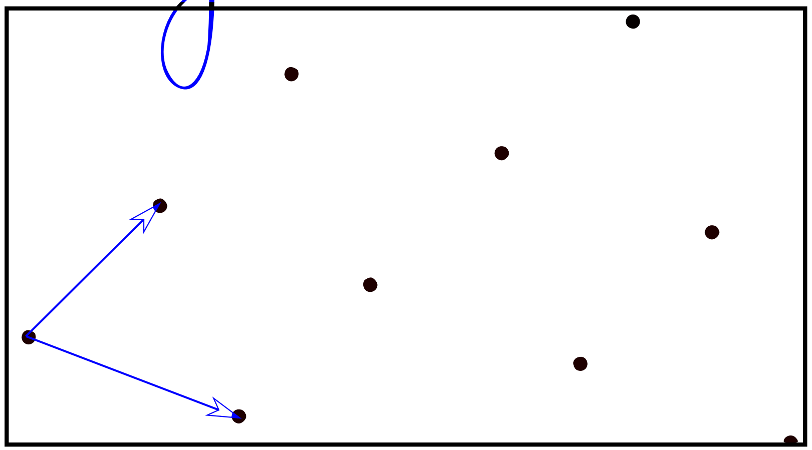


Bad basis

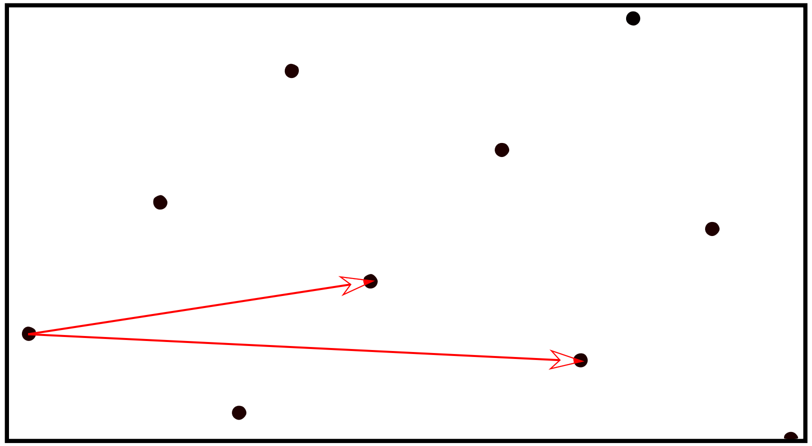


Lattices

Good basis



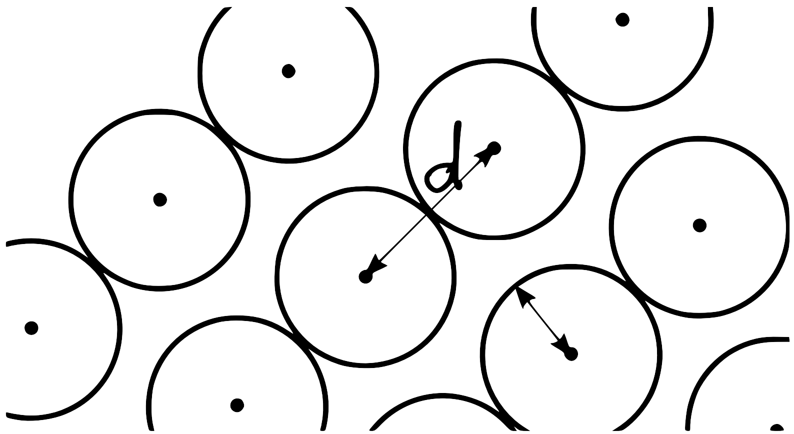
Bad basis



Minimal distance

$$d = \min_{x \in \mathcal{L} \setminus \{0\}} \|x\|$$

$\Longrightarrow$ Sphere packing



Invariants

B and B' generate the same lattice iff :

$$\exists\, U \in GL_n(\mathbb{Z}) \quad \text{st} \quad B' = B \cdot U.$$

$\Rightarrow \quad \det(\mathcal{L}) := \det(B)$ is an invariant of $\mathcal{L}$.

Invariants

B and B' generates the same lattice iff :

$$\exists\, U \in GL_n(\mathbb{Z}) \quad \text{st} \quad B' = B \cdot U.$$

$\Rightarrow \quad \det(\mathcal{L}) := \det(B)$ is an invariant of $\mathcal{L}$.

Gram-Schmidt Orthogonalisation

$$b_i^* := \pi^{\perp}_{(b_1, \dots b_{i-1})}(b_i)$$

$$= b_i - \sum_{j<i} \frac{\langle b_i, b_j^* \rangle}{\langle b_i^*, b_i^* \rangle} \cdot b_j^*$$

Invariants

B and B′ generates the same lattice iff:

$$\exists\, U \in GL_n(\mathbb{Z}) \quad \text{st} \quad B' = B \cdot U.$$

$\Rightarrow \quad \det(\mathcal{L}) := \det(B)$ is an invariant of $\mathcal{L}$.

Gram-Schmidt Orthogonalisation

$$b_i^* := \pi^{\perp}_{(b_1 \dots b_{i-1})}(b_i)$$

$$= b_i - \sum_{j<i} \frac{\langle b_i, b_j^* \rangle}{\langle b_j^*, b_j^* \rangle} \cdot b_j^*$$

Invariant

$$\det(\mathcal{L}) = \prod_i \|b_i^*\|$$

Invariants

B and B' generates the same lattice iff:

$$\exists U \in GL_n(\mathbb{Z}) \quad \text{st} \quad B' = B \cdot U.$$

$\Rightarrow \quad \det(\mathcal{L}) := \det(B)$ is an invariant of $\mathcal{L}$.

Gram-Schmidt Orthogonalisation

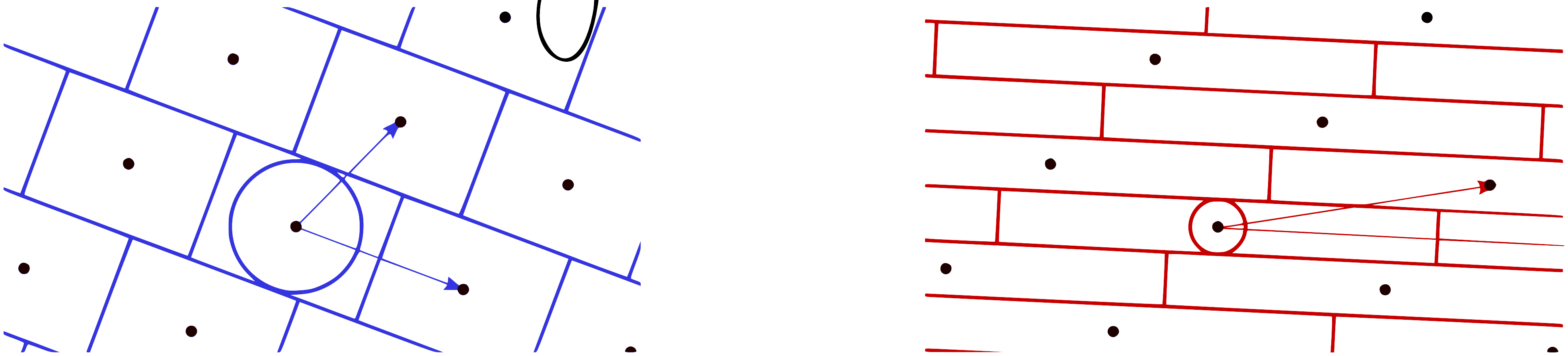
$$b_i^* := \underbrace{\pi^{\perp}_{(b_1, \ldots, b_{i-1})}}_{=:\pi_i}(b_i)$$

$$= b_i - \sum_{j<i} \frac{\langle b_i, b_j^* \rangle}{\langle b_j^*, b_j^* \rangle} \cdot b_j^*$$

Invariant

$$\det(\mathcal{L}) = \prod_i \|b_i^*\|$$

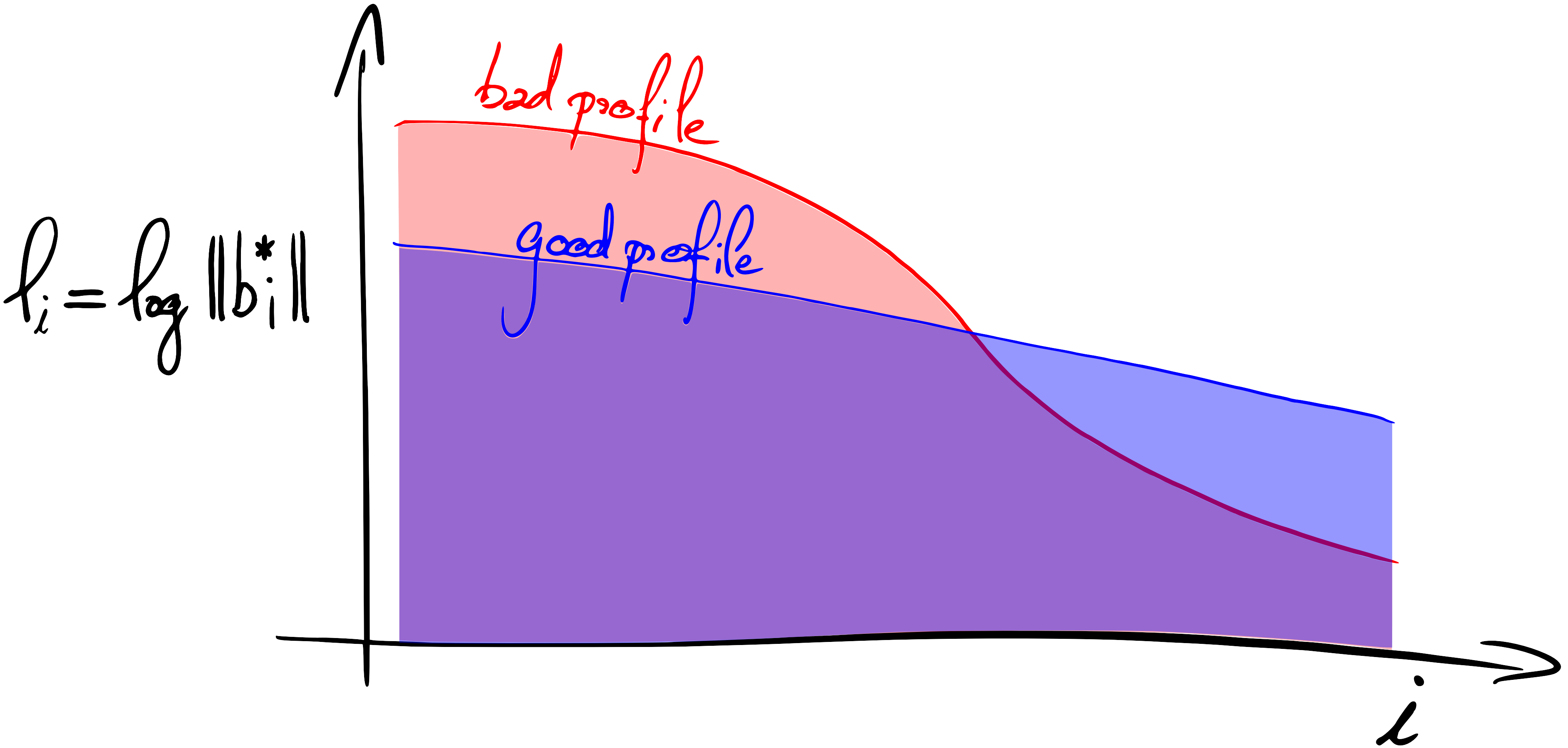
Good basis



"Good basis" $\iff$ Fundamental Parallelipiped $P(B^*)$ is "close" to a hypercube

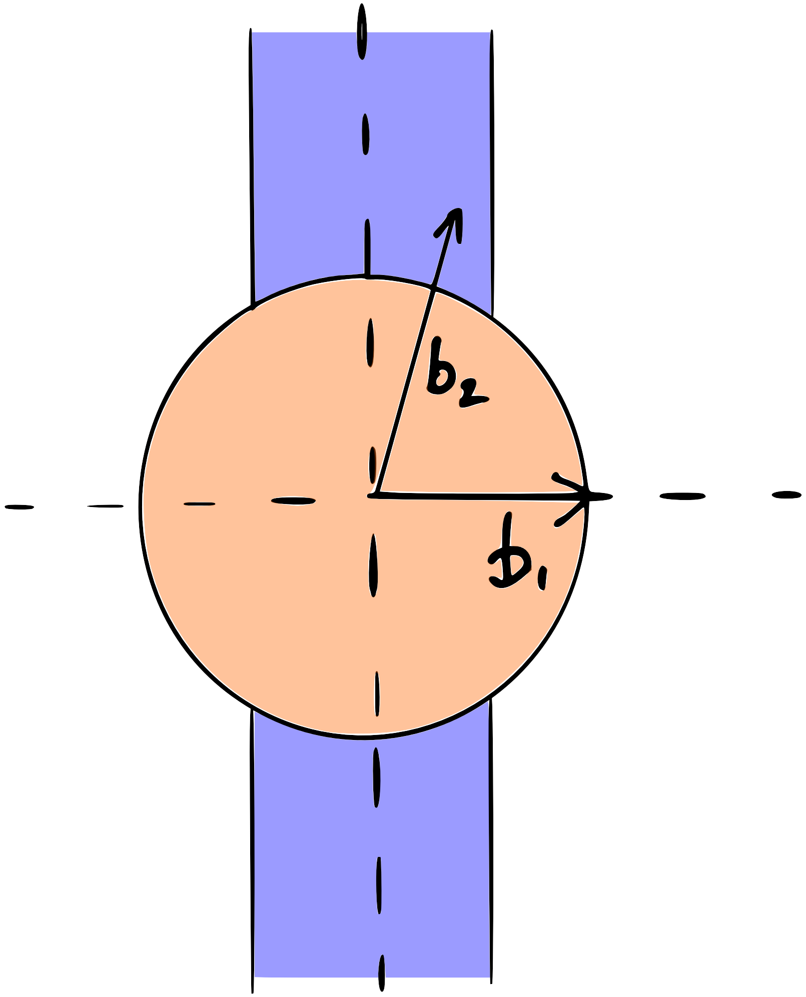
$$\iff \|b_1^*\| \approx \|b_2^*\| \approx \dots \approx \|b_n^*\|.$$

Profile



Area = Area = $\log \det(\mathcal{L})$, invariant.

$n=2$: Lagrange Reduction



Wristwatch Lemma

For any lattice $\mathcal{L}$ of dim 2
$\exists (b_1, b_2)$ a basis s.t.

$$\|b_1\| \leq \|b_2\|$$

$$|\langle b_1, b_2 \rangle| \leq 1/2 \cdot \|b_1\|$$

In particular

$$\|b_1\| \leq \sqrt{4/3} \cdot \|b_2^*\|$$

LLL Reduction

Definition

A basis B of $\mathcal{L}$ is LLL-reduced if $(\pi_i(b_i), \pi_i(b_{i+1}))$ is Lagrange-reduced for all $i < n$.

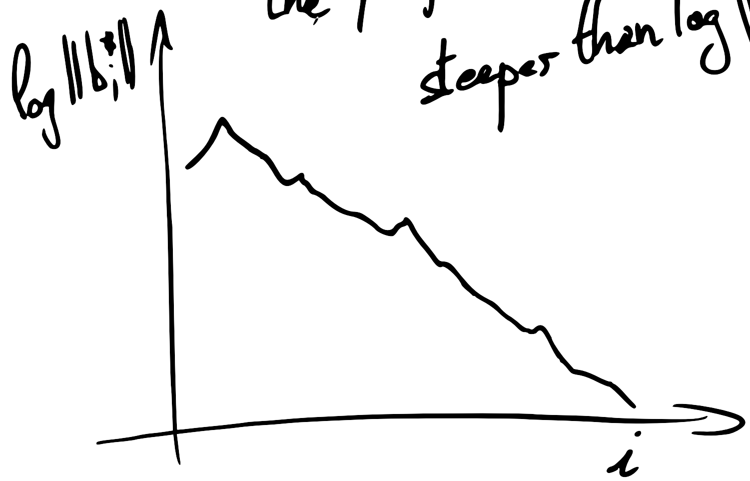
LLL Reduction

Definition

A basis B of $\mathcal{L}$ is LLL-reduced if $(\pi_i(b_i), \pi_i(b_{i+1}))$ is Lagrange-reduced for all $i<n$.

$$\Longrightarrow \forall i<n, \|b_i^*\| \leqslant \sqrt{4/3}\cdot\|b_{i+1}^*\|$$

"the profile never falls steeper than $\log\sqrt{4/3}$"



LLL Reduction

Definition

A basis B of $\mathcal{L}$ is LLL-reduced if $(\pi_i(b_i), \pi_i(b_{i+1}))$ is Lagrange-reduced for all $i<n$.

"the profile never falls steeper than $\log\sqrt{4/3}$"

$$\Longrightarrow \quad \forall i<n,\ \|b_i^*\| \leq \sqrt{4/3}\cdot\|b_{i+1}^*\|$$

Chain & collect $\Longrightarrow$

$$\|b_1\| \leq \sqrt{4/3}^{\frac{n-1}{2}}\cdot\det(\mathcal{L})^{1/n}.$$

LLL Algorithm

While $\exists i$ s.t. $(\pi_i(b_i), \pi_i(b_{i+1}))$ is not *Lagrange-reduced*

Lagrange-reduce it ...

Correctness : Trivial

LLL Algorithm

While $\exists i$ s.t. $(\pi_i(b_i), \pi_i(b_{i+1}))$ is not Lagrange-reduced
Lagrange-reduce it ...

Correctness : Trivial

Termination in poly-time :

- Requires a slight relaxation (ε-Lagrange-Reduced)
- Proved using a potential argument :

$$P = \sum_{i \leq n} \sum_{j \leq i} \log(\|b_j^*\|)$$

decreases by ε at each step and is lower-bounded.

Binary Codes

$\mathcal{C} \subseteq \mathbb{F}_2^n$ a subspace of a binary vector space, endowed with the Hamming metric.

Binary Codes

Bitstring notation

$x_1 = 0101 \in \mathbb{F}_2^4$ $\quad$ $x_2 = 0011 \in \mathbb{F}_2^4$

XOR (sum): $x_1 \oplus x_2 = 0110$ $\quad$ AND: $x_1 \wedge x_2 = 0001$ $\quad$ OR: $x_1 \vee x_2 = 0111$

Binary Codes

Bitstring notation

$x_1 = 0101 \in \mathbb{F}_2^4$ $\qquad x_2 = 0011 \in \mathbb{F}_2^4$

XOR (sum): $x_1 \oplus x_2 = 0110$ $\qquad$ AND: $x_1 \wedge x_2 = 0001$ $\qquad$ OR: $x_1 \vee x_2 = 0111$

Hamming distance

$|x| = \#\{i \mid x_i = 1\}$ $\qquad \mathrm{Supp}(x) = \{i \mid x_i = 1\}$

Minimal distance

$$d = \min_{x \in \mathcal{C} \setminus \{0\}} |x|$$

Binary Codes

Bitstring notation

$x_1 = 0101 \in \mathbb{F}_2^4$ $\quad x_2 = 0011 \in \mathbb{F}_2^4$

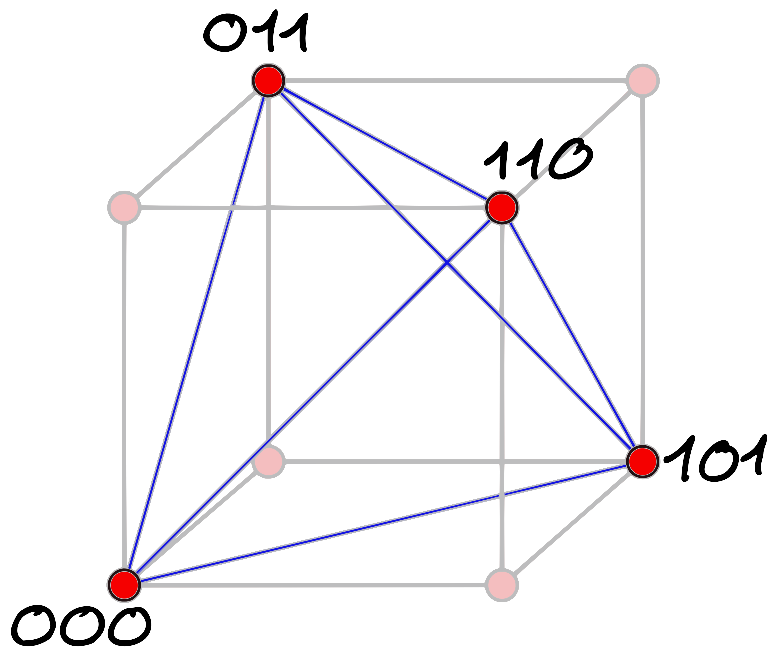
XOR (sum) : $x_1 \oplus x_2 = 0110$ $\quad$ AND : $x_1 \wedge x_2 = 0001$ $\quad$ OR : $x_1 \vee x_2 = 0111$

Hamming distance

$|x| = \#\{i \mid x_i = 1\}$ $\quad \mathrm{Supp}(x) = \{i \mid x_i = 1\}$

Minimal distance

$$d = \min_{x \in \mathcal{C} \setminus \{0\}} |x|$$



Example

$\mathcal{C} = \{000, 110, 011, 101\}$

$n = 3 \quad k = 2 \quad d = 2$

generated by $b_1 = 110$, $b_2 = 011$

Orthopodality

Inner product

$$\langle x, y\rangle = \sum x_i y_i \mod 2$$

gives no relations on $|x|, |y|$
and $|x \oplus y|$...

Definition $x \perp y$ if

$$Supp(x) \cap Supp(y) = \emptyset$$

$$(eq.\ x \wedge y = 0)$$

$$x \perp y \iff |x \oplus y| = |x| + |y|$$

Orthopodality

Inner product

$$\langle x, y\rangle = \sum x_i y_i \mod 2$$

gives no relations on $|x|, |y|$ and $|x \oplus y|$...

Definition

$x \perp y$ if

$$Supp(x) \cap Supp(y) = \emptyset$$

(eq. $x \wedge y = 0$)

$$x \perp y \iff |x \oplus y| = |x| + |y|$$

Definition

Orthopodal projection

$$\pi_x : y \longmapsto y \wedge x$$

$$\pi_x^{\perp} : y \longmapsto y \wedge \overline{x}$$

Orthopodality

Inner product

$$\langle x, y\rangle = \sum x_i y_i \mod 2$$

gives no relations on $|x|, |y|$ and $|x \oplus y|$...

Definition $x \perp y$ if

$$Supp(x) \cap Supp(y) = \emptyset$$

$$(eq.\ x \wedge y = 0)$$

$$x \perp y \iff |x \oplus y| = |x| + |y|$$

Definition

Orthopodal projection

$$\Pi_x : y \mapsto y \wedge x$$

$$\Pi_x^\perp : y \mapsto y \wedge \overline{x}$$

$$\Pi_x(y) \oplus \Pi_x^\perp(y) = y$$

$$\Pi_x(y) \perp \Pi_x^\perp(y)$$

$$|\Pi_x(y)| \leqslant |y|$$

$$\Pi_x^\perp \circ \Pi_y^\perp = \Pi_y^\perp \circ \Pi_x^\perp = \Pi_{x \vee y}^\perp$$

Epipodal matrix

Definition

For a basis $B = b_1, \dots, b_k$, the i-th epipodal vector is defined by

$$b_i^* := \pi^{\perp}_{b_1 \vee b_2 \dots \vee b_{i-1}}(b_i)$$

$$= b_i \wedge \overline{b_1 \vee b_2 \dots \vee b_{i-1}}$$

$B^* := (b_1^*, \dots, b_k^*)$ is called the epipodal matrix of B.

Epipodal matrix

Definition

For a basis $B = b_1, \dots, b_k$,
the *i-th epipodal vector*
is defined by

$$b_i^* := \pi^{\perp}_{b_1 \vee b_2 \dots \vee b_{i-1}}(b_i)$$

$$= b_i \wedge \overline{b_1 \vee b_2 \dots \vee b_{i-1}}$$

$B^* := (b_1^*, \dots, b_k^*)$ is called
the *epipodal matrix* of B.

Noting $\mathcal{C}_i := \mathcal{C}(b_1, \dots, b_{i-1})$,
$\mathrm{Supp}(b_i^*) = \mathrm{Supp}(\mathcal{C}_i) \setminus \mathrm{Supp}(\mathcal{C}_{i-1})$

Epipodal matrix

Definition

For a basis $B = b_1, \ldots, b_k$, the *i-th epipodal vector* is defined by

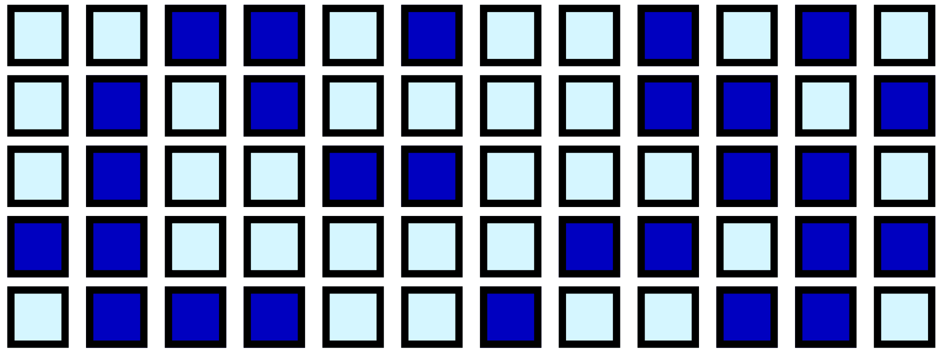
$$b_i^* := \pi^{\perp}_{b_1 \vee b_2 \ldots \vee b_{i-1}}(b_i)$$

$$= b_i \wedge \overline{b_1 \vee b_2 \ldots \vee b_{i-1}}$$

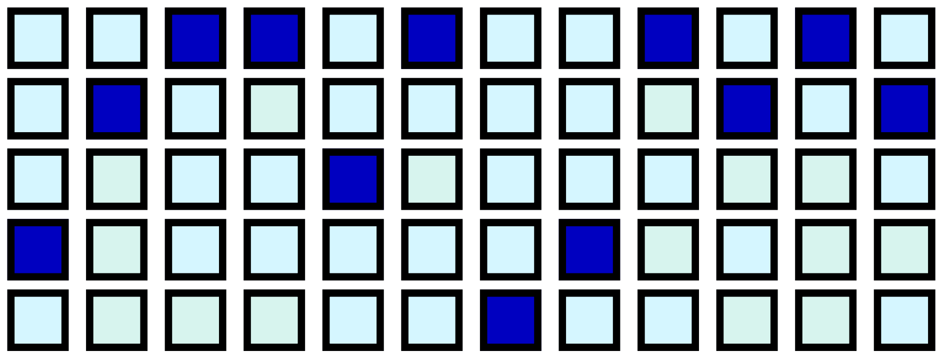
$B^* := (b_1^*, \ldots, b_k^*)$ is called the *epipodal matrix* of B.

Noting $\mathcal{C}_i := \mathcal{C}(b_1, \ldots, b_{i-1})$,
$\mathrm{Supp}(b_i^*) = \mathrm{Supp}(\mathcal{C}_i) \setminus \mathrm{Supp}(\mathcal{C}_{i-1})$

B



B^*



Epipodal matrix

Definition

For a basis $B = b_1, \dots, b_k$, the *i-th epipodal vector* is defined by

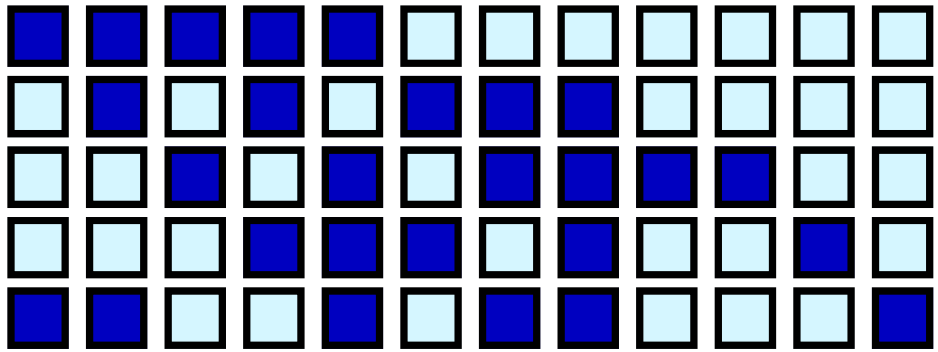
$$b_i^* := \pi^{\perp}_{b_1 \vee b_2 \dots \vee b_{i-1}}(b_i)$$

$$= b_i \wedge \overline{b_1 \vee b_2 \dots \vee b_{i-1}}$$

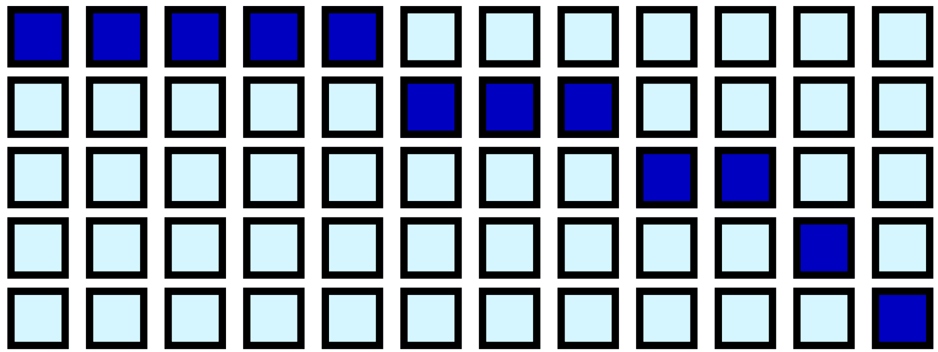
$B^* := (b_1^*, \dots, b_k^*)$ is called the *epipodal matrix* of B.

Noting $\mathcal{C}_i := \mathcal{C}(b_1, \dots, b_{i-1})$,
$\mathrm{Supp}(b_i^*) = \mathrm{Supp}(\mathcal{C}_i) \setminus \mathrm{Supp}(\mathcal{C}_{i-1})$

B



B^*



Epipodal matrix

Definition

For a basis $B = b_1, \dots, b_k$, the i-th epipodal vector is defined by

$$b_i^* := \pi^{\perp}_{b_1 \vee b_2 \dots \vee b_{i-1}}(b_i)$$

$$= b_i \wedge \overline{b_1 \vee b_2 \dots \vee b_{i-1}}$$

$B^* := (b_1^*, \dots, b_k^*)$ is called the epipodal matrix of B.

Noting $\mathcal{C}_i := \mathcal{C}(b_1, \dots, b_{i-1})$,

$$\mathrm{Supp}(b_i^*) = \mathrm{Supp}(\mathcal{C}_i) \setminus \mathrm{Supp}(\mathcal{C}_{i-1})$$

Gram-Schmidt-like properties

$$\forall i \neq j, \quad b_i^* \perp b_j^*$$

$$\forall i, \bigcup_{j \leq i} \mathrm{Supp}(b_j) = \bigcup_{j \leq i} \mathrm{Supp}(b_j^*)$$

in particular $b_1^* = b_1$

Epipodal matrix

Definition

For a basis $B = b_1, \dots, b_k$,
the i-th epipodal vector
is defined by

$$b_i^* := \pi^{\perp}_{b_1 \vee b_2 \dots \vee b_{i-1}}(b_i)$$

$$= b_i \wedge \overline{b_1 \vee b_2 \dots \vee b_{i-1}}$$

$B^* := (b_1^*, \dots, b_k^*)$ is called
the epipodal matrix of B.

Noting $\mathcal{C}_i := \mathcal{C}(b_1, \dots, b_{i-1})$,
$\mathrm{Supp}(b_i^*) = \mathrm{Supp}(\mathcal{C}_i) \setminus \mathrm{Supp}(\mathcal{C}_{i-1})$

Gram-Schmidt-like properties

$\forall i \neq j, \quad b_i^* \perp b_j^*$

$\forall i, \; \bigcup_{j \leq i} \mathrm{Supp}(b_j) = \bigcup_{j \leq i} \mathrm{Supp}(b_j^*)$

in particular $b_1^* = b_1$

Invariant

$$\#\mathrm{Supp}(\mathcal{C}) = \sum_i |b_i^*|$$

Analogue to $\det(\mathcal{L}) = \prod \|b_i^*\|$

$k=2$: Mimicking Lagrange

Lemma For any code $\mathcal{C}$ of support size $n = \#\mathrm{Supp}(\mathcal{C})$ and dimension $k=2$, there exist a basis b_1, b_2 s.t.

$$|b_1| \leqslant 2/3 \cdot n \;, \quad |b_2 \wedge b_1| \leqslant 1/2 \cdot |b_1|$$

$k=2$: Mimicking Lagrange

Lemma For any code $\mathcal{C}$ of support size $n = \#\mathrm{Supp}(\mathcal{C})$ and dimension $k=2$, there exist a basis b_1, b_2 s.t.

$$|b_1| \leqslant 2/3 \cdot n \quad , \quad |b_2 \wedge b_1| \leqslant 1/2 \cdot |b_1|$$

In particular $\quad |b_1| \leqslant 2 \cdot |b_2^*|$ $\quad$ (lattice case : $\|b_1\| \leqslant \sqrt{4/3}\, \|b_2^*\|$)

$k=2$: Mimicking Lagrange

Lemma For any code $\mathcal{C}$ of support size $n = \#\mathrm{Supp}(\mathcal{C})$ and dimension $k=2$, there exist a basis b_1, b_2 s.t.

$$|b_1| \leqslant 2/3 \cdot n \quad , \quad |b_2 \wedge b_1| \leqslant 1/2 \cdot |b_1|$$

In particular $\quad |b_1| \leqslant 2 \cdot |b_2^*|$ $\quad$ (lattice case : $\|b_1\| \leqslant \sqrt{4/3}\, \|b_2^*\|$)

Proof $\mathcal{C}$ has 3 distincts non-zero codewords:

Up to isometry (permutation)

$$\begin{array}{rl}
 & \overbrace{\qquad\qquad\quad}^{a}\ \overbrace{\qquad\qquad\quad}^{b}\ \overbrace{\qquad\quad}^{c} \\
c_1 = & \boxed{11 \cdots \quad \cdots \quad \cdots 11}\ \boxed{0 \cdots 00} \\
c_2 = & \boxed{00 \cdots 00}\ \boxed{11 \cdots \quad \cdots \quad \cdots 11} \\
\hline
c_3 = c_1 \oplus c_2 = & \boxed{11 \cdots 11}\ \boxed{00 \cdots 00}\ \boxed{11 \cdots 11}
\end{array}$$

$$\begin{aligned}
|c_1| &= a+b \\
|c_2| &= b+c \\
|c_3| &= a+c \\
n &= a+b+c
\end{aligned}$$

LLL for Codes

While $\exists i$ s.t. $(\pi_i(b_i), \pi_i(b_{i+1}))$ is not *Lagrange-reduced*

Lagrange-reduce it ...

LLL for Codes

While $\exists i$ s.t. $(\pi_i(b_i), \pi_i(b_{i+1}))$ is not Lagrange-reduced
Lagrange-reduce it ...

- No need for an ε-relaxation
- Same potential argument applies

LLL for Codes

While $\exists i$ s.t. $(\pi_i(b_i), \pi_i(b_{i+1}))$ is not Lagrange-reduced
Lagrange-reduce it ...

- No need for an ε-relaxation
- Same potential argument applies

Guarantees

$$|b_i^*| \leq 2 \cdot |b_{i+1}^*|, \qquad |b_i^*| \geq 1$$

LLL Bound

Guarentees $\quad \ell_i \leq 2 \cdot \ell_{i+1}, \quad \ell_i \geq 1 \qquad \ell_i := |b_i^*|$

Invariant $\quad \sum_i \ell_i = \# Supp(\mathcal{C}) \leq n$

LLL Bound

Guarentees $\quad \ell_i \leq 2 \cdot \ell_{i+1}, \quad \ell_i \geq 1 \qquad \ell_i := |b_i^*|$

Invariant $\quad \sum_i \ell_i = \#Supp(\mathcal{C}) \leq n$

$p := \lfloor \log_2 \ell_1 \rfloor \qquad \ell_1 \cdot \sum_{i=0}^{p} 2^{-i} + \sum_{i=p+1}^{k-1} 1 \leq n$

LLL Bound

Guarentees $\qquad \ell_i \leq 2 \cdot \ell_{i+1}, \quad \ell_i \geq 1 \qquad \ell_i := |b_i^*|$

Invariant $\qquad \sum_i \ell_i = \#Supp(\ell) \leq n$

$p := \lfloor \log_2 \ell_1 \rfloor \qquad \ell_1 \cdot \sum_{i=0}^{p} 2^{-i} + \sum_{i=p+1}^{k-1} 1 \leq n$

$$\ell_1 - \left\lfloor \frac{\log_2 \ell_1}{2} \right\rfloor \leq \frac{n-k+1}{2}$$

LLL Bound

Guarentees
$$\ell_i \leq 2 \cdot \ell_{i+1}, \quad \ell_i \geq 1 \qquad \ell_i := |b_i^*|$$

Invariant
$$\sum_i \ell_i = \#Supp(e) \leq n$$

$$p := \lfloor \log_2 \ell_1 \rfloor \qquad \ell_1 \cdot \sum_{i=0}^{p} 2^{-i} + \sum_{i=p+1}^{k-1} 1 \leq n$$

$$\ell_1 - \left\lfloor \frac{\log_2 \ell_1}{2} \right\rfloor \leq \frac{n-k+1}{2}$$

griesmer bound

Bounds

		Algorithmic?
Singleton's	$d \leq n-k+1$	Yes
Hamming's	$2^k \cdot \sum_{i=0}^{\lfloor \frac{d-1}{2} \rfloor} \binom{n}{i} \leq 2^n$	No
Griesmer's	$d - \left\lfloor \frac{\log_2 d}{2} \right\rfloor \leq \frac{n-k+1}{2}$	Yes

What's more

- An analogue to Babai's nearest plane algorithm
- A study of the associated fundamental domains
- A hybrid Lee-Brickell + Babai Algorithm
- Open-source implementation & experiments
- Many open questions, e.g.

What about Duality ?¿

The End

An Algorithmic Reduction Theory for Binary Codes

Thomas Debris-Alazard Léo Ducas Wessel van Woerden

Pre-print https://eprint.iacr.org/2020/869
Code https://github.com/lducas/CodeRed