

Linear programming for secret sharing thresholds

Iwan Duursma

Radoslav Kirov

April 6, 2008

Linear codes and
secret sharing

Linear programming

Linear codes and secret sharing

Linear programming

Linear codes and secret sharing

Linear codes and secret sharing

First description

A closer look

Summary

Notation

Composition of codes

Example

Example cont

Hermitian two-point codes

Bombieri inner product

Example of self-dual distance bound

Linear programming

Linear codes and secret sharing

First description

A closer look

Summary

Notation

Composition of codes

Example

Example cont

Hermitian two-point codes

Bombieri inner product

Example of self-dual distance bound

Let C be a code of length $n + 1$ with coordinates $\{1, \dots, n\} \cup \{0\}$. For a word (c_1, \dots, c_n, c_0) , the values c_1, \dots, c_n are shares for the secret c_0 .

Let d be the minimum distance of C and let d^\perp be the minimum distance of C^\perp .

Let C be a code of length $n + 1$ with coordinates $\{1, \dots, n\} \cup \{0\}$. For a word (c_1, \dots, c_n, c_0) , the values c_1, \dots, c_n are shares for the secret c_0 .

Let d be the minimum distance of C and let d^\perp be the minimum distance of C^\perp .

(Rejection) No party of size $\leq d^\perp - 2$ can reconstruct the secret from its shares.

(Acceptance) Every party of size $\geq n + 2 - d$ can reconstruct the secret uniquely.

$$\underbrace{0 \ \cdots \ d^\perp - 2}_{\text{Rejected}} \quad \underbrace{\cdots}_{?} \quad \underbrace{n + 2 - d \ \cdots \ n}_{\text{Accepted}}$$

Let $C_0 \subset C$ be the subcode of words that are zero in position 0. Let $D = C^\perp$, and let $D_0 \subset D$ be the subcode of words that are zero in position 0.

Words in D_0 play no role in the rejection bound, they can not be used to reconstruct the secret.

Words in C_0 in C_0 play no role in the acceptance bound, they correspond to the secret value $c_0 = 0$ and they can not cause an ambiguity about the secret value even if they contain many zeros.

Let $C_0 \subset C$ be the subcode of words that are zero in position 0. Let $D = C^\perp$, and let $D_0 \subset D$ be the subcode of words that are zero in position 0.

Words in D_0 play no role in the rejection bound, they can not be used to reconstruct the secret.

Words in C_0 in C_0 play no role in the acceptance bound, they correspond to the secret value $c_0 = 0$ and they can not cause an ambiguity about the secret value even if they contain many zeros.

(Rejection) No party of size $\leq d(D/D_0) - 2$ can reconstruct the secret from its shares.

(Acceptance) Every party of size $\geq n + 2 - d(C/C_0)$ can reconstruct the secret uniquely.

Linear codes and
secret sharing

First description

A closer look

Summary

Notation

Composition of
codes

Example

Example cont

Hermitian two-point
codes

Bombieri inner
product

Example of self-dual
distance bound

Linear programming

For secret sharing we need a pair of dual codes C and D of length $n + 1$ such that words that are nonzero in the position 0 are of high weight in both the code and its dual.

Words that are zero in the last position have no role in the reconstruction of the secret (but they could be used to recover shares of other players or to test shares for consistency).

Linear codes and
secret sharing

First description

A closer look

Summary

Notation

Composition of
codes

Example

Example cont

Hermitian two-point
codes

Bombieri inner
product

Example of self-dual
distance bound

Linear programming

For the projections of

$C = C_0 \cup (C \setminus C_0), D = D_0 \cup (D \setminus D_0)$ on

$\mathcal{P} = \{1, 2, \dots, n\}$, write

$$\begin{aligned} A &= C_0|_{\mathcal{P}}, & B &= (C \setminus C_0)|_{\mathcal{P}}. \\ X &= D_0|_{\mathcal{P}}, & Y &= (D \setminus D_0)|_{\mathcal{P}}. \end{aligned}$$

With weight enumerators $\mathbf{A}, \mathbf{B}, \mathbf{X}, \mathbf{Y}$, such that

$$\begin{aligned} W(C_0)(x, y) &= \mathbf{A}x, & W(C)(x, y) &= \mathbf{A}x + \mathbf{B}y. \\ W(D_0)(x, y) &= \mathbf{X}x, & W(D)(x, y) &= \mathbf{X}x + \mathbf{Y}y. \end{aligned}$$

$$\underbrace{0 \ \dots \ d(Y) - 1}_{\text{Rejected}} \quad \underbrace{\dots}_{?} \quad \underbrace{n + 1 - d(B) \ \dots \ n}_{\text{Accepted}}$$

Let

$$C(\Pi_1) = \left(\frac{1}{X_1} \mid \frac{1}{0} \right) \quad C(\Pi_2) = \left(\frac{1}{X_2} \mid \frac{1}{0} \right)$$

represent $\Sigma(\Pi_1)$ and $\Sigma(\Pi_2)$. Then,

$$C(\Pi) = \left(\frac{1}{X_1 \otimes 1} \mid \frac{1}{0} \right)$$
$$\left(\frac{I \otimes X_2}{0} \right)$$

represents $\Sigma(\Pi) = \Sigma(\Pi_1) \circ \Sigma(\Pi_2)$.

For a $(2, 3) \circ (3, 5)$ threshold scheme, use

$$f(x, y) \in \langle (1, x), (1, x, x^2)y, (1, x, x^2)y^2 \rangle.$$

For a $(3, 5) \circ (2, 3)$ threshold scheme, use

$$f(x, y) \in \langle (1, y, y^2), (1, y, y^2, y^3, y^4)x \rangle.$$

	b_1	b_2	b_3	b_4	b_5	\vdots		b_1	b_2	b_3	b_4	b_5
a_1	\cdot	\cdot	\cdot	\cdot	\cdot		a_1	\cdot	\cdot		\cdot	
a_2	\cdot				\cdot		a_2	\cdot			\cdot	
a_3	\cdot	\cdot					a_3	\cdot		\cdot	\cdot	\cdot

Hermitian two-point codes

Theorem (Homma - Kim 2006, reformulation Seung Kook Park 2007)

Let $G = K + aP_\infty + bP_0 \geq K + P_\infty + P_0$, and write

$$a = a_0(q + 1) - a_1, \quad 0 \leq a_1 \leq q,$$

$$b = b_0(q + 1) - b_1, \quad 0 \leq b_1 \leq q.$$

Let $d = d(C_\omega(G, D))$, $d^* = \deg(G) - (2g - 2) = a + b$.

- | | | |
|------|---|---------------------------------------|
| (1) | $a_1, b_1 \leq a_0 + b_0,$ | $d = d^*.$ |
| (2a) | $b_1 \leq a_0 + b_0 \leq a_1,$ | $d = d^* + a_1 - (a_0 + b_0).$ |
| (2b) | $a_1 \leq a_0 + b_0 \leq b_1,$ | $d = d^* + b_1 - (a_0 + b_0).$ |
| (3a) | $a_0 + b_0 \leq a_1 \leq b_1, a_1 < q,$ | $d = d^* + a_1 + b_1 - 2(a_0 + b_0).$ |
| (3b) | $a_0 + b_0 \leq b_1 \leq a_1, b_1 < q$ | $d = d^* + a_1 + b_1 - 2(a_0 + b_0).$ |
| (4) | $a_0 + b_0 \leq a_1 = b_1 = q$ | $d = d^* + q - (a_0 + b_0).$ |

Bombieri's inner product on polynomials f and g is the following:

$$[f, g] = f\left(\frac{\partial}{\partial x_1}, \dots, \frac{\partial}{\partial x_n}\right)g(x_1, \dots, x_n)$$

If σ a unitary change of variables, $[f_\sigma, g_\sigma] = [f, g]$
On homogeneous weight enumerators, $\gamma = q - 1$:

$$\sigma = \begin{pmatrix} 1 & \gamma \\ 1 & -1 \end{pmatrix}$$

Linear codes and
secret sharing

First description

A closer look

Summary

Notation

Composition of
codes

Example

Example cont

Hermitian two-point
codes

**Bombieri inner
product**

Example of self-dual
distance bound

Linear programming

Example of self-dual distance bound

Linear codes and
secret sharing

First description

A closer look

Summary

Notation

Composition of
codes

Example

Example cont

Hermitian two-point
codes

Bombieri inner
product

Example of self-dual
distance bound

Linear programming

Example for binary self-dual codes:

$$[f, g_\sigma - 2^n g] = 0$$

Choose

$$g = (x + y)^{k-2} y^{k+2} - 3(x + y)^{k-1} y^{k+1} + \\ + 6(x + y)^{k-1} y^{k+1} - 4(x + y)^{k-2} y^{k+2}.$$

Then $d < (1 - \frac{1}{q}) \frac{n}{2}$.

Linear programming

Notation

LP set up

Table for $n = n(dB, dY)$

Table for $dB = dB(dX, dY)$

Inner product

Example

Connections

Let $C_1 \subseteq C_2$ be codes of length n with codimension 1.

Codes:

$$\begin{array}{cccc} C_1 & \subseteq & C_2 & C_2 \setminus C_1 \\ C_1^\perp & \supseteq & C_2^\perp & C_1^\perp \setminus C_2^\perp \end{array}$$

Weight Enumerators:

$$\begin{array}{ccc} \mathbf{A} & \mathbf{A} + \mathbf{B} & \mathbf{B} \\ \mathbf{X} + \mathbf{Y} & \mathbf{X} & \mathbf{Y} \end{array}$$

$2(n + 1)$ variables - **A**, **B**

$$\mathbf{A} \begin{matrix} = \\ \geq \\ \geq \\ \vdots \\ \geq \end{matrix} \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

$$\mathbf{X} = P_n \mathbf{A} + P_n \mathbf{B}$$

$$= \begin{pmatrix} |A + B| \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

$$\mathbf{B} \begin{matrix} = \\ \vdots \\ = \\ \geq \\ \vdots \\ \geq \end{matrix} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \left. \vphantom{\begin{matrix} = \\ \vdots \\ = \\ \geq \\ \vdots \\ \geq \end{matrix}} \right\} d(B)$$

$$\mathbf{Y} = (q - 1)P_n \mathbf{A} - P_n \mathbf{B}$$

$$= \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \left. \vphantom{\begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}} \right\} d(Y)$$

Table for $n = n(dB, dY)$

Linear codes and
secret sharing

Linear programming

Notation

LP set up

Table for
 $n = n(dB, dY)$

Table for $dB =$
 $dB(dX, dY)$

Inner product

Example

Connections

$q = 2$

$dY \setminus dB$	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7
2	*	4	6	8	10	12	14
3	*	*	7	10	11	14	15
4	*	*	*	12	14	16	19
5	*	*	*	*	15	18	20
6	*	*	*	*	*	20	22
7	*	*	*	*	*	*	23

Table for $dB = dB(dX, dY)$

$q = 2, n = 13$

$dX \backslash dY$	1	2	3	4	5	6	7	8	9	10	11	12	13
1	13	6	5	4	3	2	1	1	1	1	1	1	1
2	13	6	5	4	3	2	1	1	1	1	1	1	1
3	7	6	5	4	3	2	1	1	1	1	1	1	1
4	7	6	5	4	3	2	1	1	1	1	1	1	1
5	5	5	5	4	3	2	1	1	1	1	1	1	1
6	5	5	5	4	3	2	1	1	1	1	1	1	1
7	4	3	3	3	3	2	1	1	1	1	1	1	1
8	3	3	3	3	3	2	1	1	1	1	1	1	1
9	3	3	3	2	2	2	1	1	1	1	1	1	1
10	3	3	3	2	2	2	1	1	1	1	1	1	1
11	3	2	2	2	2	2	1	1	1	1	1	1	1
12	3	2	2	2	2	2	1	1	1	1	1	1	1
13	2	2	2	2	2	2	1	1	1	1	1	1	1

With respect to a special dot product:

$$\frac{q^{n+1}}{AA'} ([A, A'] + \frac{[B, B']}{q-1}) = [X, X'] + \frac{[Y, Y']}{q-1}$$

Dual LP:

$$A' \geq 0, A'_0 = 1$$

$$B' \geq 0, \text{ on } [a_1, n]$$

$$X' \leq 0$$

$$Y' \leq 0, \text{ on } [a_2, n]$$

Existence of a solution to the Dual LP, means
non-existence of a a_1, a_2 LSSS.

$$[f_\sigma, g_\sigma] = q^{n+1}[f, g]$$

LSSS split weight enumerators:

$$f_{A,B}(x, y, u, v) = f_A(x, y)u + f_B(x, y)v$$

$$f_\sigma = f_X u + f_Y v$$

"Singleton Bound" Example:

$$g(x, y, u, v) = (x + \gamma y)^k x^{n-k} (u - v)$$

$$g_\sigma = g(x + \gamma y, x - y, u + \gamma v, u - v) = -q^{k+1} (x + \gamma y)^{n-k} x^k v$$

if $dB \geq k + 1$ and $dY \geq n - k + 1$, contradiction! thus

$$dB \leq n + 1 - dY$$

Linear codes and
secret sharing

Linear programming

Notation

LP set up

Table for

$n = n(dB, dY)$

Table for $dB =$

$dB(dX, dY)$

Inner product

Example

Connections

Connections with other coding theory problems:

1. $\max d = d(A + B)$ in terms of $d' = d(X + Y)$.
2. $\max \rho = d(B)$ in terms of $d' = d(X + Y)$.
3. $\max a_1 = d(B)$ in terms of $a_2 = d(Y)$.