Coding with Constraints: Different Flavors

Hoang Dau
University of Illinois at Urbana-Champaign
Email: hoangdau@uiuc.edu

DIMACS Workshop on Network Coding: the Next 15 Years
Rutgers University, NJ, 2015
Part I
Coding with Constraints: A Quick Survey
Coding with Constraints: Definition

CONVENTIONAL CODE
\[ c_j = c_j(x_1, x_2, ..., x_k) \]
Coding with Constraints: Definition

CONVENTIONAL CODE
\[ c_j = c_j(x_1, x_2, \ldots, x_k) \]

CODE WITH CONSTRAINTS
\[ c_j = c_j(\{x_i : i \in C_j\}), \quad C_j \subseteq \{1, 2, \ldots, k\} \]
Coding with Constraints: Example

\[ \begin{align*}
  x_1 & \rightarrow c_1 = c_1(x_1, x_2) \\
  x_2 & \rightarrow c_2 = c_2(x_1, x_3) \\
  x_3 & \rightarrow c_3 = c_3(x_2) \\
  & \rightarrow c_4 = c_4(x_2, x_3) \\
  & \rightarrow c_5 = c_5(x_1, x_3)
\end{align*} \]
Coding with Constraints: Example

Linear code: \((c_1, c_2, c_3, c_4, c_5) = (x_1, x_2, x_3)G\)

where the generator matrix \(G\) is

\[
G = \begin{pmatrix}
? & ? & 0 & 0 & ? \\
? & 0 & ? & ? & 0 \\
0 & ? & 0 & ? & ?
\end{pmatrix}
\]
Coding with Constraints: Main Problem

Given the constraints

\[ c_j = c_j(\{x_i : i \in C_j\}), \quad C_j \subseteq \{1, 2, \ldots, k\} \]

how to construct codes that

- achieve the \textbf{optimal minimum distance}
- over \textbf{small field size} \( q \approx \text{poly}(n) \)
Coding with Constraints: Main Problem

Given the constraints

\[ c_j = c_j(\{x_i: i \in C_j\}), \quad C_j \subseteq \{1,2, \ldots, k\} \]

how to construct codes that

– achieve the optimal minimum distance
– over small field size \( q \approx \text{poly}(n) \)

**Linear case:**

given

\[ G = \begin{pmatrix} ? & ? & 0 & 0 & ? \\ ? & 0 & ? & ? & 0 \\ 0 & ? & 0 & ? & ? \end{pmatrix} \]

how to replace “?”-entries by elements of \( F_q \) (\( q \approx \text{poly}(n) \)) so that \( G \) generate a code with optimal distance
Coding with Constraints: Upper Bound

Upper Bound (Halbawi-Thill-Hassibi’15, Song-Dau-Yuen’15)

\[ d \leq d_{\text{max}} = 1 + \min_{\emptyset \neq I \subseteq \{1,\ldots,k\}} (|\cup_{i \in I} R_i| - |I|) \]

where \( R_i = \{j : i \in C_j\} \)
Coding with Constraints: Upper Bound

Upper Bound (Halbawi-Thill-Hassibi’15, Song-Dau-Yuen’15)
\[ d \leq d_{\text{max}} = 1 + \min_{\emptyset \neq I \subseteq \{1, \ldots, k\}} \left( |\bigcup_{i \in I} R_i| - |I| \right) \]
where \( R_i = \{ j : i \in C_j \} \)

Properties
- \( d_{\text{max}} \) can be found in time \( \text{poly}(n) \)
- Codes with \( d = d_{\text{max}} \) always exists over fields of size \( \approx \binom{n}{d - 1} \)
Coding with Constraints: Upper Bound

Upper Bound (Halbawi-Thill-Hassibi’15, Song-Dau-Yuen’15)

\[ d \leq d_{\text{max}} = 1 + \min_{\emptyset \neq I \subseteq \{1,\ldots,k\}} (|\bigcup_{i \in I} R_i| - |I|) \]

where \( R_i = \{j : i \in C_j\} \)

Properties

- \( d_{\text{max}} \) can be found in time \( \text{poly}(n) \)
- codes with \( d = d_{\text{max}} \) always exists over fields of size \( \approx \binom{n}{d-1} \)

Question of interest: how about fields of size \( \text{poly}(n) \)?
Coding with Constraints: Review

(Small field)

**MDS Case:** $d_{max} = n - k + 1$

- Optimal codes exist in a few special cases (Halbawi-Ho-Yao-Duursma’14, Dau-Song-Yuen’14, Yan-Sprintson-Zelenko’14)
Coding with Constraints: Review

(Small field)

**MDS Case:** $d_{\text{max}} = n - k + 1$

- Optimal codes exist in a few special cases (Halbawi-Ho-Yao-Duursma’14, Dau-Song-Yuen’14, Yan-Sprintson-Zelenko’14)
  - $k \leq 4$ (every $n$)
Coding with Constraints: Review

(Small field)

**MDS Case:** $d_{max} = n - k + 1$

- Optimal codes exist in a few special cases (Halbawi-Ho-Yao-Duursma’14, Dau-Song-Yuen’14, Yan-Sprintson-Zelenko’14)
  - $k \leq 4$ (every $n$)
  - rows of $G$ partitioned into $\leq 3$ groups, each has same “?”-pattern
Coding with Constraints: Review

(Small field)

**MDS Case:** $d_{max} = n - k + 1$

- Optimal codes exist in a few special cases (Halbawi-Ho-Yao-Duursma’14, Dau-Song-Yuen’14, Yan-Sprintson-Zelenko’14)
  - $k \leq 4$ (every $n$)
  - rows of $G$ partitioned into $\leq 3$ groups, each has same “?”-pattern
  - rows have $k - 1$ zeros & 2 different rows share $\leq 1$ common zeros
Coding with Constraints: Review

(Small field)

**MDS Case:** $d_{max} = n - k + 1$

- Optimal codes exist in a few special cases (Halbawi-Ho-Yao-Duursma’14, Dau-Song-Yuen’14, Yan-Sprintson-Zelenko’14)
  - $k \leq 4$ (every $n$)
  - rows of $G$ partitioned into $\leq 3$ groups, each has same “?”-pattern
  - rows have $k - 1$ zeros & 2 different rows share $\leq 1$ common zeros

**General Case** (Halbawi-Thill-Hassibi’15): $d_{max} \leq n - k + 1$
Coding with Constraints: Review

(Small field)

**MDS Case:** $d_{\text{max}} = n - k + 1$

- Optimal codes exist in a few special cases (Halbawi-Ho-Yao-Duursma’14, Dau-Song-Yuen’14, Yan-Sprintson-Zelenko’14)
  - $k \leq 4$ (every $n$)
  - rows of $G$ partitioned into $\leq 3$ groups, each has same “?”-pattern
  - rows have $k - 1$ zeros & 2 different rows share $\leq 1$ common zeros

**General Case** (Halbawi-Thill-Hassibi’15): $d_{\text{max}} \leq n - k + 1$

- Optimal codes exist if there are $\geq d_{\text{max}} - 1$ indices $j$’s where $|C_j| = k$
Coding with Constraints: Review

(Small field)

MDS Case:  \( d_{\text{max}} = n - k + 1 \)
- Optimal codes exist in a few special cases (Halbawi-Ho-Yao-Duursma’14, Dau-Song-Yuen’14, Yan-Sprintson-Zelenko’14)
  - \( k \leq 4 \) (every \( n \))
  - rows of \( G \) partitioned into \( \leq 3 \) groups, each has same “?”-pattern
  - rows have \( k - 1 \) zeros & 2 different rows share \( \leq 1 \) common zeros

General Case (Halbawi-Thill-Hassibi’15):  \( d_{\text{max}} \leq n - k + 1 \)
- Optimal codes exist if there are \( \geq d_{\text{max}} - 1 \) indices \( j \)’s where \( |\mathcal{C}_j| = k \)
- Optimal systematic codes always exists (smaller bound \( d_{\text{sys}} \leq d_{\text{max}} \))
Coding with Constraints: Review

(Small field)

**MDS Case:** \( d_{\text{max}} = n - k + 1 \)
- Optimal codes exist in a few special cases (Halbawi-Ho-Yao-Duursma’14, Dau-Song-Yuen’14, Yan-Sprintson-Zelenko’14)
  - \( k \leq 4 \) (every \( n \))
  - rows of \( G \) partitioned into \( \leq 3 \) groups, each has same “?“-pattern
  - rows have \( k - 1 \) zeros & 2 different rows share \( \leq 1 \) common zeros

**General Case** (Halbawi-Thill-Hassibi’15): \( d_{\text{max}} \leq n - k + 1 \)
- Optimal codes exist if there are \( \geq d_{\text{max}} - 1 \) indices \( j \)’s where \( |C_j| = k \)
- Optimal systematic codes always exists (smaller bound \( d_{\text{sys}} \leq d_{\text{max}} \))

**Common Technique:** Reed-Solomon (sub-) code
Coding with Constraints: Review

Common Technique: Reed-Solomon (sub-) code

\[
G = \begin{pmatrix}
? & ? & 0 & 0 & ? \\
? & 0 & ? & ? & 0 \\
0 & ? & 0 & ? & ? \\
\end{pmatrix}
\]
Coding with Constraints: Review

Common Technique: Reed-Solomon (sub-) code

\[ G = \begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 \\ ? & ? & 0 & 0 & ? \\ ? & 0 & ? & ? & 0 \\ 0 & ? & 0 & ? & ? \\ 0 & ? & 0 & ? & ? \end{pmatrix} \]

\[ c_1 = c_1(x_1, x_2) \]
\[ c_2 = c_2(x_1, x_3) \]
\[ c_3 = c_3(x_2) \]
\[ c_4 = c_4(x_2, x_3) \]
\[ c_5 = c_5(x_1, x_3) \]
Coding with Constraints: Review

**Common Technique:** Reed-Solomon (sub-) code

\[ G = \begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 \\ ? & ? & 0 & 0 & ? \\ ? & 0 & ? & ? & 0 \\ 0 & ? & 0 & ? & ? \end{pmatrix} = \begin{pmatrix} f_1(\alpha_1) & f_1(\alpha_2) & 0 & 0 & f_1(\alpha_5) \\ f_2(\alpha_1) & 0 & f_2(\alpha_3) & f_2(\alpha_4) & 0 \\ 0 & f_3(\alpha_2) & 0 & f_3(\alpha_4) & f_3(\alpha_5) \end{pmatrix} \]

\[ x_1 \rightarrow \begin{array}{c} c_1 \\ c_2 \\ c_3 \end{array} = \begin{array}{c} c_1(x_1, x_2) \\ c_2(x_1, x_3) \\ c_3(x_2) \end{array} \]

\[ x_2 \rightarrow \begin{array}{c} c_4 \\ c_5 \end{array} = \begin{array}{c} c_4(x_2, x_3) \\ c_5(x_1, x_3) \end{array} \]
Coding with Constraints: Review

Common Technique: Reed-Solomon (sub-) code

\[
G = \begin{pmatrix}
\alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 \\
? & ? & 0 & 0 & ? \\
? & 0 & ? & ? & 0 \\
0 & ? & 0 & ? & ? \\
\end{pmatrix} = \begin{pmatrix}
f_1(\alpha_1) & f_1(\alpha_2) & 0 & 0 & f_1(\alpha_5) \\
f_2(\alpha_1) & 0 & f_2(\alpha_3) & f_2(\alpha_4) & 0 \\
f_2(\alpha_1) & 0 & f_2(\alpha_3) & f_2(\alpha_4) & 0 \\
0 & f_3(\alpha_2) & 0 & f_3(\alpha_4) & f_3(\alpha_5) \\
\end{pmatrix}
\]

Difficulty: G may not be full rank
Part II:
Joint Design of Different MDS Codes
(joint work with H. Kiah, W. Song, and C. Yuen)
MDS Codes for Distributed Storage

Encoded

$k$ data symbols

\[ x_1, x_2, \ldots, x_k \]

$n$ coded symbols

\[ c_1, c_2, \ldots, c_n \]

MDS: tolerate any $n - k$ node failures

Facebook: Reed-Solomon

\[ n = 14, k = 10 \]
Question of Interest

• If two (or more) independent DSS share some common data, can we jointly design the corresponding MDS codes to get a better overall failure protection?
Question of Interest

- If two (or more) independent DSS share some common data, can we jointly design the corresponding MDS codes to get a better overall failure protection?

Tolerate 1 node failure

\[ \begin{align*}
\{ x_1, x_2, x_3 \} & \quad \text{MDS} \\
\{ c_1, c_2, c_3, c_4 \} & \quad \text{Local Subcode 1}
\end{align*} \]

Local distance \( d_1 = 2 \)

Code has minimum distance \( d \): tolerate \( d - 1 \) node failures
Question of Interest

• If two (or more) independent DSS share some common data, can we jointly design the corresponding MDS codes to get a better overall failure protection?

Code has minimum distance $d$: tolerate $d - 1$ node failures
Question of Interest

- If two (or more) independent DSS share some common data, can we jointly design the corresponding MDS codes to get a better overall failure protection?

Tolerate 1 node failure

Tolerate 3 node failures

Tolerate 2 node failures

Code has minimum distance $d$: tolerate $d - 1$ node failures
Upper Bound for Global Minimum Distance

For linear code

$$(c_1, c_2, ..., c_{10}) = (x_1, x_2, ..., x_6)G$$

where

$$G = \begin{bmatrix}
? & ? & ? & ? & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}$$
Upper Bound for Global Minimum Distance

For linear code

\[(c_1, c_2, \ldots, c_{10}) = (x_1, x_2, \ldots, x_6)G\]

where

\[
G = \begin{bmatrix}
? & ? & ? & ? & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

**Goal:** replace “?”-entries with \(F_q\)-elements so that
Upper Bound for Global Minimum Distance

For linear code

\((c_1, c_2, \ldots, c_{10}) = (x_1, x_2, \ldots, x_6)G\)

where

\[
G = \begin{bmatrix}
? & ? & ? & ? & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

**Goal:** replace “?”-entries with \(F_q\)-elements so that

- two local subcodes are MDS (**additional requirement**)

\[\text{[4,3]-MDS} \quad \text{[6,4]-MDS}\]
Upper Bound for Global Minimum Distance

For linear code

\[(c_1, c_2, ..., c_{10}) = (x_1, x_2, ..., x_6)G\]

where

\[
G = \begin{bmatrix}
? & ? & ? & ? & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

Goal: replace “?”-entries with \(F_q\)-elements so that

- two local subcodes are MDS (additional requirement)
- the global code has optimal distance (same as coding with constraints)
Upper Bound for Global Minimum Distance

For linear code

\[(c_1, c_2, ..., c_{10}) = (x_1, x_2, ..., x_6)G\]

where

\[
G = \begin{bmatrix}
\end{bmatrix}
\]

[4,3]-MDS [6,4]-MDS

**Goal:** replace "?"-entries with \(F_q\)-elements so that

- two local subcodes are MDS (additional requirement)
- the global code has optimal distance (same as coding with constraints)

Same cut-set bound apply & codes over large fields achieve this bound

\[d \leq d_{max} = 1 + \min_{\emptyset \neq I \subseteq \{1, ..., k\}} (|\bigcup_{i \in I} R_i| - |I|)\]
Upper Bound for Global Minimum Distance: Two Local Subcodes

- $d \leq d_{max} = 1 + t + \min\{n_1 - k_1, n_2 - k_2\}$
- $t = \#\{\text{common } x_i\}$
Upper Bound for Global Minimum Distance: Two Local Subcodes

- \( d \leq d_{\text{max}} = 1 + t + \min\{n_1 - k_1, n_2 - k_2\} \)
- \( t = \#\{\text{common } x_i\} \)

\( n_1 = 4, k_1 = 3 \)

\( t = |\{x_2, x_3\}| = 2 \)

\( n_2 = 6, k_2 = 4 \)

In this example: \( d \leq 1 + 2 + \min\{4 - 3, 6 - 4\} = 4 \rightarrow \text{optimal code here} \)
Generator Matrix Representation

1st code: \( G_1 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 4 & 2 & 2 \end{bmatrix} = \begin{bmatrix} u \\ A \end{bmatrix} \)

Local Subcode 1

\( \begin{align*}
\mathbf{x}_1 & = x_1 + x_2 + x_3 \\
\mathbf{x}_2 & = x_1 + 2x_2 + 4x_3 \\
\mathbf{x}_3 & = x_1 + 3x_2 + 2x_3 \\
\mathbf{c}_4 & = x_1 + 4x_2 + 2x_3
\end{align*} \)

Local distance \( d_1 = 2 \)
Generator Matrix Representation

1\textsuperscript{st} code: $G_1 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 4 & 2 & 3 \end{bmatrix} = \begin{bmatrix} u \\ A \end{bmatrix}$

2\textsuperscript{nd} code: $G_2 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 4 & 2 & 2 & 4 & 1 \\ 1 & 1 & 6 & 1 & 6 & 6 \end{bmatrix} = \begin{bmatrix} B \\ V \end{bmatrix}$

Local Subcode 1
- $c_1 = x_1 + x_2 + x_3$
- $c_2 = x_1 + 2x_2 + 4x_3$
- $c_3 = x_1 + 3x_2 + 2x_3$
- $c_4 = x_1 + 4x_2 + 2x_3$

Local Subcode 2
- $c_5 = x_2 + x_3 + x_4 + x_5$
- $c_6 = x_2 + 2x_3 + 4x_4 + x_5$
- $c_7 = x_2 + 3x_3 + 2x_4 + 6x_5$
- $c_8 = x_2 + 4x_3 + 2x_4 + x_5$
- $c_9 = x_2 + 5x_3 + 4x_4 + 6x_5$
- $c_{10} = x_2 + 6x_3 + x_4 + 6x_5$

Local distance $d_1 = 2$

Local distance $d_2 = 3$
Generator Matrix Representation

1st code: \( G_1 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 4 & 2 & 3 \end{bmatrix} = [U_A] \)

2nd code: \( G_2 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 4 & 2 & 2 & 4 & 1 \\ 1 & 1 & 6 & 1 & 6 & 6 \end{bmatrix} = [B_V] \)

Global code: \( G = \begin{bmatrix} U & O \\ A & B \\ O & V \end{bmatrix} \)

Local Subcode 1
\[
\begin{aligned}
&c_1 = x_1 + x_2 + x_3 \\
&c_2 = x_1 + 2x_2 + 4x_3 \\
&c_3 = x_1 + 3x_2 + 2x_3 \\
&c_4 = x_1 + 4x_2 + 2x_3 \\
&\text{Local distance } d_1 = 2
\end{aligned}
\]

Local Subcode 2
\[
\begin{aligned}
&c_5 = x_2 + x_3 + x_4 + x_5 \\
&c_6 = x_2 + 2x_3 + 4x_4 + x_5 \\
&c_7 = x_2 + 3x_3 + 2x_4 + 6x_5 \\
&c_8 = x_2 + 4x_3 + 2x_4 + 4x_5 \\
&c_9 = x_2 + 5x_3 + 4x_4 + 6x_5 \\
&c_{10} = x_2 + 6x_3 + x_4 + 6x_5 \\
&\text{Local distance } d_2 = 3
\end{aligned}
\]

Global minimum distance \( d = 4 \)
Easy Case: Two Codes Have Few Common Data

- If few common data, i.e.
  \[ t \leq 1 + \max\{n_2 - k_2, n_1 - k_1\} \]
  using two Vandermonde matrices as \( G_1, G_2 \): optimal minimum distance
- Finite field size required: \(|F_q| \geq \max\{n_1, n_2\}\)
Easy Case: Two Codes Have Few Common Data

- If few common data, i.e.
  \[ t \leq 1 + \max\{n_2 - k_2, n_1 - k_1\} \]
  using two Vandermonde matrices as \( G_1, G_2 \) : optimal minimum distance
- Finite field size required: \(|F_q| \geq \max\{n_1, n_2\}\)

More specifically, in this case, if

- \( G_1 = \begin{bmatrix} U \\ A \end{bmatrix} \) is a nested MDS code: \( G_1, U \) are generator matrices of MDS codes
- \( G_2 = \begin{bmatrix} B \\ V \end{bmatrix} \) is a nested MDS code: \( G_2, V \) are generator matrices of MDS codes

then the global code achieves the optimal minimum distance (attains the upper bound)
Harder Case: Two Codes Have the Same Redundancy

- If same redundancy, i.e.
  \[ n_1 - k_1 = n_2 - k_2 \]
  we construct codes that have optimal global minimum distance
- Finite field size required: \( q > n = n_1 + n_2 \)

The construction uses the BCH bound
Two Codes Have the Same Redundancy: BCH Bound

- $F_q$: finite field of $q$ elements
- $\omega$: primitive element of $F_q$, i.e. $F_q = \{0,1, \omega, \omega^2, \omega^3, ... \}$
- Identify a vector $c = (c_1, ..., c_n) \in F_q^n$ with the polynomial
  \[ c(x) = c_1 + c_2x + c_3x^2 + \cdots + c_nx^{n-1} \]

**BCH Bound:** If every coded vector $c$ satisfies
\[ c(\omega^i) = 0, \text{ for every } i = 0, 1, ..., \delta - 1 \]
i.e., they all have $\delta$ consecutive powers of $\omega$ as roots, then the code has minimum distance $d \geq \delta + 1$
Two Codes Have the Same Redundancy: Construction

We construct the generator matrix of the optimal code. Example: $F_{13}$

\[
G = \begin{bmatrix}
U & O \\
A & B \\
O & V
\end{bmatrix}
\]
Two Codes Have the Same Redundancy: Construction

We construct the generator matrix of the optimal code. Example: \( F_{13} \)

\[
G = \begin{bmatrix}
U & O \\
A & B \\
O & V \\
\end{bmatrix} = 
\]

\[
\begin{array}{cccc}
5 & 1 & 6 & 1 \\
5 & 1 & 6 & 1 \\
5 & 1 & 6 & 1 \\
5 & 11 & 10 & 1 \\
9 & 4 & 7 & 5 & 1 \\
12 & 1 & 5 & 1 & 6 \\
5 & 1 & 6 & 1 \\
5 & 1 & 6 & 1 \\
5 & 1 & 6 & 1 \\
\end{array}
\]

\( k_1 = 4 \)

\( n_1 = 5 \)

\( k_2 = 5 \)

\( n_1 = 6 \)
Two Codes Have the Same Redundancy: Construction

We construct the generator matrix of the optimal code. Example: $F_{13}$

$$G = \begin{bmatrix} U & O \\ A & B \\ O & V \end{bmatrix} = \begin{bmatrix} 5 & 1 & 6 & 1 \\ 5 & 1 & 6 & 1 \\ 1 \end{bmatrix}$$

Rows of $G_1$ has root $1 = \omega^0$ -> distance 2=1+1

$k_1 = 4$, $n_1 = 5$, $k_2 = 5$, $n_1 = 6$
Two Codes Have the Same Redundancy: Construction

We construct the generator matrix of the optimal code. Example: $F_{13}$

$$G = \begin{bmatrix} U & O \\ A & B \\ O & V \end{bmatrix} =$$

Rows of $G_1$ has root $1 = \omega^0$ -> distance 2=1+1

Rows of $G_2$ has root $1 = \omega^0$ -> distance 2=1+1
Two Codes Have the Same Redundancy: Construction

We construct the generator matrix of the optimal code. Example: $F_{13}$

\[
G = \begin{bmatrix}
U & O \\
A & B \\
O & V
\end{bmatrix}
\]

Rows of $G_1$ has root $1 = \omega^0$  
\[ \Rightarrow \text{distance } 2 = 1+1 \]

Rows of $G_2$ has root $1 = \omega^0$  
\[ \Rightarrow \text{distance } 2 = 1+1 \]
Two Codes Have the Same Redundancy: Construction

Summary of this construction
• Rows: treated as polynomial having certain roots
• Solving systems of linear equations to determine rows
• BCH bound → global code & local codes have desired distances

\[
\begin{bmatrix}
5 & 1 & 6 & 1 \\
5 & 1 & 6 & 1 \\
\hline
5 & 11 & 10 \\
9 & 4 \\
\hline
\end{bmatrix}
\begin{bmatrix}
12 & 1 \\
7 & 5 & 1 \\
\hline
5 & 1 & 6 & 1 \\
5 & 1 & 6 & 1 \\
5 & 1 & 6 & 1 \\
\end{bmatrix}
\]
Conclusions

What we have done

• Introduce a new coding problem: how to jointly design 2 (or more) MDS codes to have better overall failure tolerance

• **Construct optimal codes for two cases**
  – There are few common data
  – Two codes have the same amount of redundancy

Open Questions

– Codes over small field size for 2 local codes: $n_1 - k_1 \neq n_2 - k_2$
– Codes over small field size for more than two local codes