
II W. A.
31 de março a 03 de abril de 2014
$G \times G \to G$
$(g,h) \mapsto g \cdot h$
$(a \cdot b) \cdot c = a \cdot (b \cdot c)$
$g \cdot e = g = e \cdot g$
$\iota : G \to G$
$g \mapsto g^{-1}$
$g \cdot g^{-1} = e$

Códigos Corretores de Erros

Grasiele Jorge
UNIFESP

Jorge Alencar
UNICAMP

II Workshop de Álgebra da UFG-CAC

Catalão, Brasil 31 de Março até 03 de Abril, 2014

Exemplo

Vamos construir um **código binário** de comprimento 6 de modo que as três primeiras componentes c_1, c_2 e c_3 de cada palavra do código sejam de *informação* e vamos adicionar três outros dígitos de *redundância*.

- Para isso usaremos o fato que, em $\mathbb{Z}_2 = \{\overline{0}, \overline{1}\}$ existe uma operação de soma: **a soma módulo** 2.
- Definimos então os dígitos de redundância de acordo com a seguinte regra:

$$c_4 = c_1 + c_2$$

$$c_5 = c_1 + c_3$$

$$c_6 = c_2 + c_3$$

Exemplo

Vamos construir um **código binário** de comprimento 6 de modo que as três primeiras componentes c_1, c_2 e c_3 de cada palavra do código sejam de *informação* e vamos adicionar três outros dígitos de *redundância*.

- Para isso usaremos o fato que, em $\mathbb{Z}_2 = \{\overline{0}, \overline{1}\}$ existe uma operação de soma: **a soma módulo** 2.
- Definimos então os dígitos de redundância de acordo com a seguinte regra:

$$c_4 = c_1 + c_2$$

$$c_5 = c_1 + c_3$$

$$c_6 = c_2 + c_3$$

Exemplo

Vamos construir um **código binário** de comprimento 6 de modo que as três primeiras componentes c_1, c_2 e c_3 de cada palavra do código sejam de *informação* e vamos adicionar três outros dígitos de *redundância*.

- Para isso usaremos o fato que, em $\mathbb{Z}_2 = \{\overline{0}, \overline{1}\}$ existe uma operação de soma: **a soma módulo** 2.
- Definimos então os dígitos de redundância de acordo com a seguinte regra:

$$c_4 = c_1 + c_2$$

$$c_5 = c_1 + c_3$$

$$c_6 = c_2 + c_3$$

Usando a notação vetorial para as palavras do código:

$$\boxed{c = (c_1, c_2, c_3, c_1 + c_2, c_1 + c_3, c_2 + c_3).}$$

Podemos descrever o processo que transforma a informação (c_1, c_2, c_3) na palavra do código, usando notação matricial:

$$(c_1, c_2, c_3) \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} = (c_1, c_2, c_3, c_1+c_2, c_1+c_3, c_2+c_3).$$

Desta forma, quando (c_1, c_2, c_3) percorre todos os elementos de $\mathbb{Z}_2^3$, as respectivas imagens produzem todas as palavras do código.

Usando a notação vetorial para as palavras do código:

$$\boxed{c = (c_1, c_2, c_3, c_1 + c_2, c_1 + c_3, c_2 + c_3).}$$

Podemos descrever o processo que transforma a informação (c_1, c_2, c_3) na palavra do código, usando notação matricial:

$$(c_1, c_2, c_3) \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} = (c_1, c_2, c_3, c_1+c_2, c_1+c_3, c_2+c_3).$$

Desta forma, quando (c_1, c_2, c_3) percorre todos os elementos de $\mathbb{Z}_2^3$, as respectivas imagens produzem todas as palavras do código.

Usando a notação vetorial para as palavras do código:

$$c = (c_1, c_2, c_3, c_1 + c_2, c_1 + c_3, c_2 + c_3).$$

Podemos descrever o processo que transforma a informação (c_1, c_2, c_3) na palavra do código, usando notação matricial:

$$(c_1, c_2, c_3) \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} = (c_1, c_2, c_3, c_1+c_2, c_1+c_3, c_2+c_3).$$

Desta forma, quando (c_1, c_2, c_3) percorre todos os elementos de $\mathbb{Z}_2^3$, as respectivas imagens produzem todas as palavras do código.

$$(c_1, c_2, c_3)\begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} = (c_1, c_2, c_3, c_1+c_2, c_1+c_3, c_2+c_3).$$

Definição:

A matriz acima é chamada de **matriz geradora** do código.

Podemos reescrever os dígitos de redundância da seguinte forma:

$$\begin{aligned} c_1 + c_2 + c_4 &= 0 \\ c_1 + c_3 + c_5 &= 0 \\ c_2 + c_3 + c_6 &= 0 \end{aligned}$$

Um vetor $y = (y_1, y_2, y_3, y_4, y_5, y_6) \in \mathbb{Z}_2^6$ está no código se, e somente se,

$$\begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{pmatrix} = (0, 0, 0).$$

Podemos reescrever os dígitos de redundância da seguinte forma:

$$\begin{aligned} c_1 + c_2 + c_4 &= 0 \\ c_1 + c_3 + c_5 &= 0 \\ c_2 + c_3 + c_6 &= 0 \end{aligned}$$

Um vetor $y = (y_1, y_2, y_3, y_4, y_5, y_6) \in \mathbb{Z}_2^6$ está no código se, e somente se,

$$\begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{pmatrix} = (0, 0, 0).$$

$$\begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{pmatrix} = (0, 0, 0).$$

Definição:

A matriz acima é chamada de uma **matriz de verificação de paridade** do código.

Este exemplo ilustra uma situação mais geral, que veremos no que se segue.

Para construirmos códigos de uma *maneira eficiente* e poder elaborar alguma teoria, precisamos introduzir alguma

estrutura algébrica

- Tomaremos como alfabeto A um **corpo finito** com q elementos que denotaremos por F_q.
- Neste caso, o conjunto $F_q^n = \{(x_1, \ldots, x_n); x_i \in F_q, \forall i\}$ tem uma estrutura de **espaço vetorial** de dimensão n sobre F_q.
- Tomaremos então como *códigos* C, não subconjuntos quaisquer de F_q^n, mas apenas **subespaços vetoriais** de F_q^n, de dimensão $k < n$.

Para construirmos códigos de uma *maneira eficiente* e poder elaborar alguma teoria, precisamos introduzir alguma

estrutura algébrica

- Tomaremos como alfabeto A um **corpo finito** com q elementos que denotaremos por F_q.
- Neste caso, o conjunto $F_q^n = \{(x_1, \ldots, x_n); x_i \in F_q, \forall i\}$ tem uma estrutura de **espaço vetorial** de dimensão n sobre F_q.
- Tomaremos então como *códigos* C, não subconjuntos quaisquer de F_q^n, mas apenas **subespaços vetoriais** de F_q^n, de dimensão $k < n$.

Para construirmos códigos de uma *maneira eficiente* e poder elaborar alguma teoria, precisamos introduzir alguma

estrutura algébrica

- Tomaremos como alfabeto A um **corpo finito** com q elementos que denotaremos por F_q.
- Neste caso, o conjunto $F_q^n = \{(x_1, \ldots, x_n); x_i \in F_q, \forall i\}$ tem uma estrutura de **espaço vetorial** de dimensão n sobre F_q.
- Tomaremos então como *códigos* C, não subconjuntos quaisquer de F_q^n, mas apenas **subespaços vetoriais** de F_q^n, de dimensão $k < n$.

Para construirmos códigos de uma *maneira eficiente* e poder elaborar alguma teoria, precisamos introduzir alguma

estrutura algébrica

- Tomaremos como alfabeto A um **corpo finito** com q elementos que denotaremos por F_q.
- Neste caso, o conjunto $F_q^n = \{(x_1, \ldots, x_n); x_i \in F_q, \forall i\}$ tem uma estrutura de **espaço vetorial** de dimensão n sobre F_q.
- Tomaremos então como *códigos* C, não subconjuntos quaisquer de F_q^n, mas apenas **subespaços vetoriais** de F_q^n, de dimensão $k < n$.

Para construirmos códigos de uma *maneira eficiente* e poder elaborar alguma teoria, precisamos introduzir alguma

estrutura algébrica

- Tomaremos como alfabeto A um **corpo finito** com q elementos que denotaremos por F_q.
- Neste caso, o conjunto $F_q^n = \{(x_1, \ldots, x_n); x_i \in F_q, \forall i\}$ tem uma estrutura de **espaço vetorial** de dimensão n sobre F_q.
- Tomaremos então como *códigos* C, não subconjuntos quaisquer de F_q^n, mas apenas **subespaços vetoriais** de F_q^n, de dimensão $k < n$.

Se a dimensão de C é k o número de palavras de C é $M = q^k$.

Definição: Um código C nas condições acima diz-se um (n, k)-código linear sobre F_q e, se sua distância mínima d é conhecida, então ele diz-se um (n, k, d)-código linear.

Determinar a distância de Hamming mínima de um código é um problema difícil!

Quando trabalhamos com códigos lineares podemos fazer algumas simplificações no cálculo. Se denotarmos por 0 o elemento neutro da soma no espaço vetorial, temos que $0 \in C$, pois C é sempre subespaço vetorial.

Determinar a distância de Hamming mínima de um código é um problema difícil!

Quando trabalhamos com códigos lineares podemos fazer algumas simplificações no cálculo. Se denotarmos por 0 o elemento neutro da soma no espaço vetorial, temos que $0 \in C$, pois C é sempre subespaço vetorial.

Definição: Dado um elemento em um código linear C, chama-se **peso** de x o número:

$$w(x) = d(x, 0)$$

e chama-se **peso de** C ao número

$$w(C) = \min\{w(x), x \in C, x \neq 0\}.$$

Dados $x = (x_1, x_2, \ldots, x_n), y = (y_1, y_2, \ldots, y_n) \in C$ temos:

$$\begin{aligned} d(x,y) &= |\{i;\ x_i \neq y_i, 1 \leq i \leq n\}| \\ &= |\{i;\ x_i - y_i \neq 0;\ 1 \leq i \leq n\}| \\ &= d(x-y, 0) \\ &= w(x-y) \end{aligned}$$

Portanto, a distância entre elementos do código C é também o peso de algum elemento de C. Logo

$$d(C) = w(C).$$

- Para conhecer a distância mínima de um código com M palavras precisamos avaliar $\binom{M}{2} = \frac{M(M-1)}{2}$ distâncias,
- Para conhecer seu peso, precisamos apenas avaliar $M-1$ distâncias (de cada um dos $(M-1)$ elementos não nulos ao elemento 0).

- Para conhecer a distância mínima de um código com M palavras precisamos avaliar $\binom{M}{2} = \frac{M(M-1)}{2}$ distâncias,
- Para conhecer seu peso, precisamos apenas avaliar $M-1$ distâncias (de cada um dos $(M-1)$ elementos não nulos ao elemento 0).

Exemplo

- O conjunto $C = \{(0000), (1011), (0110), (1101)\} \subseteq \mathbb{Z}_2^4$ é um subespaço vetorial de $\mathbb{Z}_2^4$.
- O conjunto $B = \{(1011), (1101)\}$ é uma base de C.
- $$w(1011) = 3,\ w(0110) = 2,\ w(1101) = 3.$$
- Portanto a distância mínima de C é 2 e C trata-se de um $(4, 2, 2)$-código.

Exemplo

- O conjunto $C = \{(0000), (1011), (0110), (1101)\} \subseteq \mathbb{Z}_2^4$ é um subespaço vetorial de $\mathbb{Z}_2^4$.
- O conjunto $B = \{(1011), (1101)\}$ é uma base de C.
- $$w(1011) = 3,\ w(0110) = 2,\ w(1101) = 3.$$
- Portanto a distância mínima de C é 2 e C trata-se de um $(4, 2, 2)$-código.

Exemplo

- O conjunto $C = \{(0000), (1011), (0110), (1101)\} \subseteq \mathbb{Z}_2^4$ é um subespaço vetorial de $\mathbb{Z}_2^4$.
- O conjunto $B = \{(1011), (1101)\}$ é uma base de C.
- $$w(1011) = 3,\ w(0110) = 2,\ w(1101) = 3.$$
- Portanto a distância mínima de C é 2 e C trata-se de um $(4, 2, 2)$-código.

Exemplo

- O conjunto $C = \{(0000), (1011), (0110), (1101)\} \subseteq \mathbb{Z}_2^4$ é um subespaço vetorial de $\mathbb{Z}_2^4$.
- O conjunto $B = \{(1011), (1101)\}$ é uma base de C.
- $$w(1011) = 3,\ w(0110) = 2,\ w(1101) = 3.$$
- Portanto a distância mínima de C é 2 e C trata-se de um $(4, 2, 2)$-código.

Matriz Geradora

- Seja $C \subseteq F_q^n$ um código linear de dimensão k.
- Seja $\{c_1, \ldots, c_k\}$ uma base de C.

Definição: Uma matriz $G \in M_{k \times n}(F_q)$ cujas linhas formam uma base para C diz-se uma **matriz geradora** de C.

Note que, para cada escolha de uma base para C obtemos uma matriz de geradora diferente, de modo que esta matriz não é única.

Matriz Geradora

- Seja $C \subseteq F_q^n$ um código linear de dimensão k.
- Seja $\{c_1, \ldots, c_k\}$ uma base de C.

Definição: Uma matriz $G \in M_{k \times n}(F_q)$ cujas linhas formam uma base para C diz-se uma **matriz geradora** de C.

Note que, para cada escolha de uma base para C obtemos uma matriz de geradora diferente, de modo que esta matriz não é única.

Matriz Geradora

- Seja $C \subseteq F_q^n$ um código linear de dimensão k.
- Seja $\{c_1, \ldots, c_k\}$ uma base de C.

Definição: Uma matriz $G \in M_{k \times n}(F_q)$ cujas linhas formam uma base para C diz-se uma **matriz geradora** de C.

Note que, para cada escolha de uma base para C obtemos uma matriz de geradora diferente, de modo que esta matriz não é única.

Matriz Geradora

- Seja $C \subseteq F_q^n$ um código linear de dimensão k.
- Seja $\{c_1, \ldots, c_k\}$ uma base de C.

Definição: Uma matriz $G \in M_{k \times n}(F_q)$ cujas linhas formam uma base para C diz-se uma **matriz geradora** de C.

Note que, para cada escolha de uma base para C obtemos uma matriz de geradora diferente, de modo que esta matriz não é única.

Código como imagem de transformação linear

- Seja $\{e_1, \ldots, e_k\}$ a base canônica de F_q^k.
- Considere a transformação linear

$$T : F_q^k \to F_q^n$$

tal que

$$T(e_i) = c_i, \text{ para } 1 \leq i \leq k.$$

- T injetora e $Im(T) = C$.

Código como imagem de transformação linear

- Seja $\{e_1, \ldots, e_k\}$ a base canônica de F_q^k.
- Considere a transformação linear

$$T : F_q^k \to F_q^n$$

tal que

$$T(e_i) = c_i, \text{ para } 1 \leq i \leq k.$$

- T injetora e $Im(T) = C$.

Código como imagem de transformação linear

- Sejam

$$\begin{aligned} c_1 &= (c_{11}, c_{12}, \ldots, c_{1n}) \\ c_2 &= (c_{21}, c_{22}, \ldots, c_{2n}) \\ &\vdots \\ c_k &= (c_{k1}, c_{k2}, \ldots, c_{kn}) \end{aligned}$$

  as coordenadas de c_i na base canônica de F_q^n.

- Considere a matriz geradora

$$G = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{k1} & c_{k2} & \cdots & c_{kn} \end{pmatrix}.$$

- A transformação linear T pode ser definida como

$$T(x) = xG.$$

Código como imagem de transformação linear

- Sejam

$$\begin{aligned} c_1 &= (c_{11}, c_{12}, \ldots, c_{1n}) \\ c_2 &= (c_{21}, c_{22}, \ldots, c_{2n}) \\ &\vdots \\ c_k &= (c_{k1}, c_{k2}, \ldots, c_{kn}) \end{aligned}$$

  as coordenadas de c_i na base canônica de F_q^n.
- Considere a matriz geradora

$$G = \begin{pmatrix} c_{11} & c_{12} & \ldots & c_{1n} \\ c_{21} & c_{22} & \ldots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{k1} & c_{k2} & \ldots & c_{kn} \end{pmatrix}.$$

- A transformação linear T pode ser definida como

$$T(x) = xG.$$

Código como imagem de transformação linear

- Sejam

$$\begin{aligned} c_1 &= (c_{11}, c_{12}, \ldots, c_{1n}) \\ c_2 &= (c_{21}, c_{22}, \ldots, c_{2n}) \\ &\vdots \\ c_k &= (c_{k1}, c_{k2}, \ldots, c_{kn}) \end{aligned}$$

  as coordenadas de c_i na base canônica de F_q^n.
- Considere a matriz geradora

$$G = \begin{pmatrix} c_{11} & c_{12} & \ldots & c_{1n} \\ c_{21} & c_{22} & \ldots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{k1} & c_{k2} & \ldots & c_{kn} \end{pmatrix}.$$

- A transformação linear T pode ser definida como

$$T(x) = xG.$$

Código como imagem de transformação linear

Os elementos de C são então todos os vetores

$$y \in F_q^n;\ y = x.G,\ x \in F_q^k.$$

- Suponha que desejamos enviar mensagens com k dígitos de informação e $n-k$ dígitos de redundância.
- Podemos considerar que o vetor de informação é um elemento do espaço vetorial F_q^k e que o vetor já codificado, é um elemento de F_q^n.

$$(x_1, \ldots, x_k) \to (y_1, y_2, \ldots, y_n).$$

- Suponha que desejamos enviar mensagens com k dígitos de informação e $n-k$ dígitos de redundância.
- Podemos considerar que o vetor de informação é um elemento do espaço vetorial F_q^k e que o vetor já codificado, é um elemento de F_q^n.

$$(x_1, \ldots, x_k) \to (y_1, y_2, \ldots, y_n).$$

Exemplo - Código de tripla repetição

- Informação: $\{(00), (10), (01), (11)\}$
- Palavras do Código:

$$C = \{(000000), (101010), (010101), (111111)\} \subseteq \mathbb{Z}_2^6.$$

- O código C pode ser visto como a imagem da transformação linear $T : \mathbb{Z}_2^2 \to \mathbb{Z}_2^6$ tal que $T(x) = xG$, onde

$$G = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

Exemplo - Código de tripla repetição

- Informação: $\{(00), (10), (01), (11)\}$
- Palavras do Código:

$$C = \{(000000), (101010), (010101), (111111)\} \subseteq \mathbb{Z}_2^6.$$

- O código C pode ser visto como a imagem da transformação linear $T : \mathbb{Z}_2^2 \to \mathbb{Z}_2^6$ tal que $T(x) = xG$, onde

$$G = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

Exemplo - Código de tripla repetição

- Informação: $\{(00), (10), (01), (11)\}$
- Palavras do Código:

$$C = \{(000000), (101010), (010101), (111111)\} \subseteq \mathbb{Z}_2^6.$$

- O código C pode ser visto como a imagem da transformação linear $T : \mathbb{Z}_2^2 \to \mathbb{Z}_2^6$ tal que $T(x) = xG$, onde

$$G = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

- Dada uma matriz geradora G de um código C, em geral é difícil decidir se um dado vetor $y \in F_q^n$ pertence ou não a C.
- Para isso, precisamos encontrar um vetor $x \in F_q^k$ tal que seja possível escrever $y = xG$, o que equivale a resolver um sistema linear com k incógnitas e n equações.

$$\begin{cases} y_1 = x_1 c_{11} + x_2 c_{21} + \ldots + x_k c_{k1} \\ y_2 = x_1 c_{12} + x_2 c_{22} + \ldots + x_k c_{k2} \\ \vdots \\ y_n = x_1 c_{1n} + x_2 c_{2n} + \ldots + x_k c_{kn} \end{cases}$$

- Dada uma matriz geradora G de um código C, em geral é difícil decidir se um dado vetor $y \in F_q^n$ pertence ou não a C.
- Para isso, precisamos encontrar um vetor $x \in F_q^k$ tal que seja possível escrever $y = xG$, o que equivale a resolver um sistema linear com k incógnitas e n equações.

$$\begin{cases} y_1 = x_1 c_{11} + x_2 c_{21} + \ldots + x_k c_{k1} \\ y_2 = x_1 c_{12} + x_2 c_{22} + \ldots + x_k c_{k2} \\ \vdots \\ y_n = x_1 c_{1n} + x_2 c_{2n} + \ldots + x_k c_{kn} \end{cases}$$

Definição: Dizemos que a matriz geradora G de um código linear C está na **forma sistemática** se

$$G = [I_{k\times k}|P_{k\times(n-k)}]$$

para alguma $k \times (n-k)$ matriz P.

Proposição: Todo código linear sobre F_q é equivalente a um código linear com uma matriz geradora na forma sistemática.

Em um código "sistemático" linear temos que o vetor informação aparece no vetor codificado

$$(x_1, x_2, \ldots, x_k)[I_{k\times k}|P_{k\times(n-k)}] = (x_1, \cdots, x_k, y_1, \ldots, y_{n-k}).$$

Definição: Dizemos que a matriz geradora G de um código linear C está na **forma sistemática** se

$$G = [I_{k\times k}|P_{k\times(n-k)}]$$

para alguma $k \times (n-k)$ matriz P.

Proposição: Todo código linear sobre F_q é equivalente a um código linear com uma matriz geradora na forma sistemática.

Em um código "sistemático" linear temos que o vetor informação aparece no vetor codificado

$$(x_1, x_2, \ldots, x_k)[I_{k\times k}|P_{k\times(n-k)}] = (x_1, \cdots, x_k, y_1, \ldots, y_{n-k}).$$

Definição: Dizemos que a matriz geradora G de um código linear C está na **forma sistemática** se

$$G = [I_{k\times k}|P_{k\times(n-k)}]$$

para alguma $k \times (n-k)$ matriz P.

Proposição: Todo código linear sobre F_q é equivalente a um código linear com uma matriz geradora na forma sistemática.

Em um código "sistemático" linear temos que o vetor informação aparece no vetor codificado

$$(x_1, x_2, \ldots, x_k)[I_{k\times k}|P_{k\times(n-k)}] = (x_1, \cdots, x_k, y_1, \ldots, y_{n-k}).$$

Definição: Sejam $u = (u_1, \ldots, u_n)$ e $v = (v_1, \ldots, v_n)$ elementos de F_q^n. Definimos o produto interno de u e v como sendo

$$\langle u, v \rangle = u_1 v_1 + \cdots + u_n v_u.$$

Definição: Seja $C \subseteq F_q^n$ um código linear, definimos

$$C^{\perp} = \{v \in F_q^n;\ \langle v, u \rangle = 0, \forall u \in C\}.$$

Proposição: $C^{\perp}$ é um espaço vetorial de F_q^n e, portanto, um código linear.

Definição: Sejam $u = (u_1, \ldots, u_n)$ e $v = (v_1, \ldots, v_n)$ elementos de F_q^n. Definimos o produto interno de u e v como sendo

$$\langle u, v \rangle = u_1 v_1 + \cdots + u_n v_u.$$

Definição: Seja $C \subseteq F_q^n$ um código linear, definimos

$$C^{\perp} = \{v \in F_q^n;\ \langle v, u \rangle = 0, \forall u \in C\}.$$

Proposição: $C^{\perp}$ é um espaço vetorial de F_q^n e, portanto, um código linear.

Definição: Sejam $u = (u_1, \ldots, u_n)$ e $v = (v_1, \ldots, v_n)$ elementos de F_q^n. Definimos o produto interno de u e v como sendo

$$\langle u, v \rangle = u_1 v_1 + \cdots + u_n v_u.$$

Definição: Seja $C \subseteq F_q^n$ um código linear, definimos

$$C^{\perp} = \{v \in F_q^n;\ \langle v, u \rangle = 0, \forall u \in C\}.$$

Proposição: $C^{\perp}$ é um espaço vetorial de F_q^n e, portanto, um código linear.

Matriz Controle de Paridade

Proposição: Seja H uma matriz geradora para $C^{\perp}$. Temos que

$$x \in C \iff Hx^t = 0$$

Definição: A matriz H como acima é chamada **matriz controle de paridade** para o código C.

Matriz Controle de Paridade

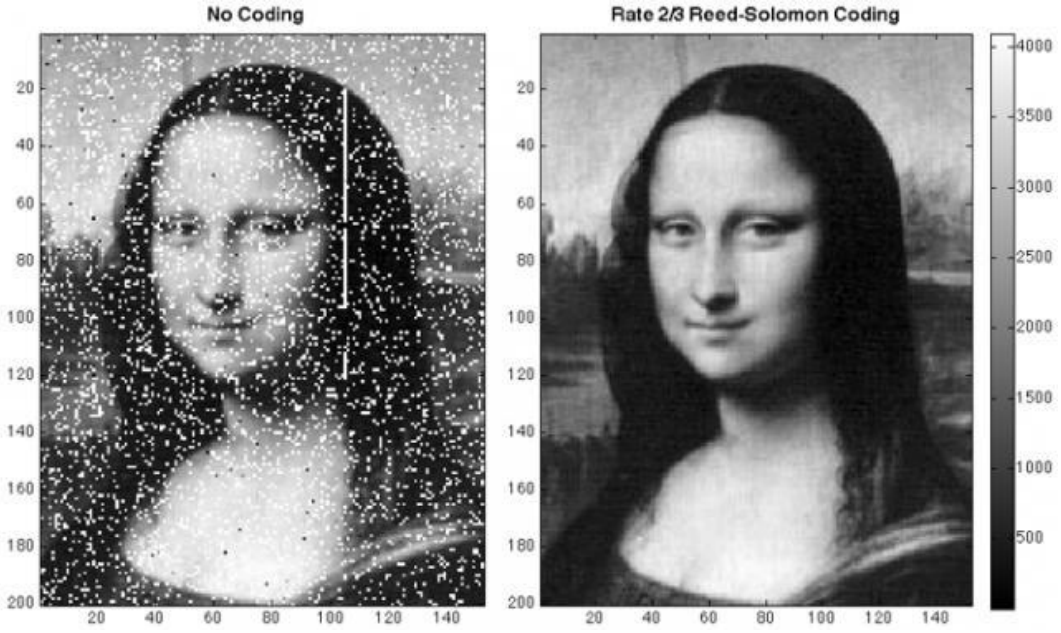
Proposição: Seja H uma matriz geradora para $C^{\perp}$. Temos que

$$x \in C \iff Hx^t = 0$$

Definição: A matriz H como acima é chamada **matriz controle de paridade** para o código C.

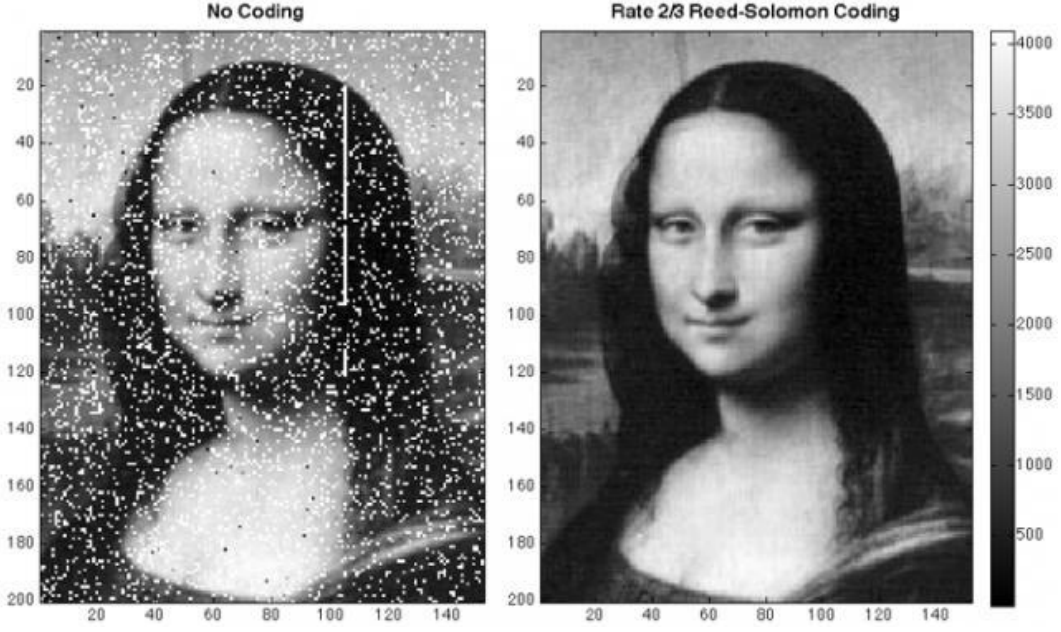
Decodificação

- Após as informações serem codificadas e passarem pelo canal, acontece o processo da decodificação que consiste em verificar se aconteceram erros e se é possível corrigí-los.
- Ocorrem erros em uma transmissão quando a palavra do código enviada é diferente do vetor recebido. Dependendo dos parâmetros de cada código existe uma quantidade de erros que o código pode detectar e corrigir.



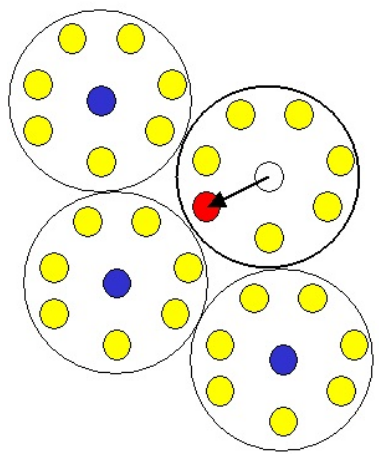
Decodificação

- Após as informações serem codificadas e passarem pelo canal, acontece o processo da decodificação que consiste em verificar se aconteceram erros e se é possível corrigí-los.
- Ocorrem erros em uma transmissão quando a palavra do código enviada é diferente do vetor recebido. Dependendo dos parâmetros de cada código existe uma quantidade de erros que o código pode detectar e corrigir.



Arranjo Padrão

Ao redor de cada palavra do código, podemos traçar uma esfera de raio $t = \lfloor (d-1)/2 \rfloor$, onde d indica a distância mínima do código. Vamos descrever todas estas esferas de raio t centralizadas nas palavras do código.



Arranjo Padrão

Toda esfera sobre cada palavra do código consiste em uma **translação** da esfera centrada na origem. Desta forma, é necessário apenas encontrar os vetores contidos na esfera centralizada na origem e transladando-os encontramos os vetores contidos em todas as respectivas esferas.

Arranjo Padrão

0	c_2	c_3	$\dots$	c_{q^k}
$0+v_1$	c_2+v_1	c_3+v_1	$\dots$	$c_{q^k}+v_1$
$0+v_2$	c_2+v_2	c_3+v_2	$\dots$	$c_{q^k}+v_2$
$\vdots$				
$0+v_j$	c_2+v_j	c_3+v_j	$\dots$	$c_{q^k}+v_j$
$0+v_{j+1}$	c_2+v_{j+1}	c_3+v_{j+1}	$\dots$	$c_{q^k}+v_{j+1}$
$\vdots$				
$0+v_l$	c_2+v_l	c_3+v_l	$\dots$	$c_{q^k}+v_l$

- Na primeira linha escrevemos todas as q^k palavras do código, sendo que a primeira deve ser o 0
- Nas linhas seguintes, abaixo da primeira coluna lista-se os vetores que distam menos de t da palavra de código toda nula.
- Se v é o primeiro vetor da segunda linha, os demais são obtidos pela soma de cada palavra de código com v.

Arranjo Padrão

0	c_2	c_3	$\dots$	c_{q^k}
$0+v_1$	c_2+v_1	c_3+v_1	$\dots$	$c_{q^k}+v_1$
$0+v_2$	c_2+v_2	c_3+v_2	$\dots$	$c_{q^k}+v_2$
$\vdots$				
$0+v_j$	c_2+v_j	c_3+v_j	$\dots$	$c_{q^k}+v_j$
$0+v_{j+1}$	c_2+v_{j+1}	c_3+v_{j+1}	$\dots$	$c_{q^k}+v_{j+1}$
$\vdots$				
$0+v_l$	c_2+v_l	c_3+v_l	$\dots$	$c_{q^k}+v_l$

- Na primeira linha escrevemos todas as q^k palavras do código, sendo que a primeira deve ser o 0
- Nas linhas seguintes, abaixo da primeira coluna lista-se os vetores que distam menos de t da palavra de código toda nula.
- Se v é o primeiro vetor da segunda linha, os demais são obtidos pela soma de cada palavra de código com v.

Arranjo Padrão

0	c_2	c_3	$\dots$	c_{q^k}
$0+v_1$	c_2+v_1	c_3+v_1	$\dots$	$c_{q^k}+v_1$
$0+v_2$	c_2+v_2	c_3+v_2	$\dots$	$c_{q^k}+v_2$
$\vdots$				
$0+v_j$	c_2+v_j	c_3+v_j	$\dots$	$c_{q^k}+v_j$
$0+v_{j+1}$	c_2+v_{j+1}	c_3+v_{j+1}	$\dots$	$c_{q^k}+v_{j+1}$
$\vdots$				
$0+v_l$	c_2+v_l	c_3+v_l	$\dots$	$c_{q^k}+v_l$

- Na primeira linha escrevemos todas as q^k palavras do código, sendo que a primeira deve ser o 0
- Nas linhas seguintes, abaixo da primeira coluna lista-se os vetores que distam menos de t da palavra de código toda nula.
- Se v é o primeiro vetor da segunda linha, os demais são obtidos pela soma de cada palavra de código com v.

- Dado um vetor recebido v, localizamos-o no arranjo-padrão,
- se estiver acima da linha horizontal, decodificamos-o como a palavra-código que está no topo da coluna, ou seja, no centro da esfera em que tal vetor se encontra.
- Caso o vetor esteja fora das esferas, apenas detectamos que houve mais do que t erros.

É impraticável realizar a decodificação de um código linear qualquer C por tal método quando k é grande.

Vamos tentar simplificar o processo.

Considere o grupo quociente

$$F_q^n / C.$$

Definição: O vetor diferença e entre um vetor recebido y e o vetor transmitido x chama-se o **vetor erro**, isto é, $e = y - x$.

Ao receber o vetor y, deve se multiplicar pela matriz H para saber se ele contém ou não erros.

Definição: O vetor diferença e entre um vetor recebido y e o vetor transmitido x chama-se o **vetor erro**, isto é, $e = y - x$.

Ao receber o vetor y, deve se multiplicar pela matriz H para saber se ele contém ou não erros.

Definição: Seja C um (n, k)-código linear, com matriz controle de paridade H. Dado um vetor $y \in F_q^n$, o vetor

$$S(y) = Hy^t$$

é chamado de síndrome de y.

- O vetor y recebido é uma palavra do código se, e somente se, sua síndrome é o vetor nulo.
- Se y é um vetor recebido, com vetor de erro e, tem-se que

$$Hy^t = H(x + e)^t = Hx^t + He^t = He^t.$$

O vetor recebido e o vetor erro têm ambos a mesma síndrome.

Definição: Seja C um (n, k)-código linear, com matriz controle de paridade H. Dado um vetor $y \in F_q^n$, o vetor

$$S(y) = Hy^t$$

é chamado de síndrome de y.

- O vetor y recebido é uma palavra do código se, e somente se, sua síndrome é o vetor nulo.
- Se y é um vetor recebido, com vetor de erro e, tem-se que

$$Hy^t = H(x + e)^t = Hx^t + He^t = He^t.$$

O vetor recebido e o vetor erro têm ambos a mesma síndrome.

Definição: Seja C um (n, k)-código linear, com matriz controle de paridade H. Dado um vetor $y \in F_q^n$, o vetor

$$S(y) = Hy^t$$

é chamado de síndrome de y.

- O vetor y recebido é uma palavra do código se, e somente se, sua síndrome é o vetor nulo.
- Se y é um vetor recebido, com vetor de erro e, tem-se que

$$Hy^t = H(x + e)^t = Hx^t + He^t = He^t.$$

O vetor recebido e o vetor erro têm ambos a mesma síndrome.

Definição: Seja C um (n, k)-código linear, com matriz controle de paridade H. Dado um vetor $y \in F_q^n$, o vetor

$$S(y) = Hy^t$$

é chamado de síndrome de y.

- O vetor y recebido é uma palavra do código se, e somente se, sua síndrome é o vetor nulo.
- Se y é um vetor recebido, com vetor de erro e, tem-se que

$$Hy^t = H(x + e)^t = Hx^t + He^t = He^t.$$

O vetor recebido e o vetor erro têm ambos a mesma síndrome.

Definição:

- O subconjunto $y + C$ de F_q^n chama-se a **classe lateral** de y determinada por C.
- Um vetor de peso de Hamming mínimo numa classe lateral diz-se um **líder da classe**.

Considere o código linear

$$C = \{000000, 100110, 010101, 001011, 011110, 101101, 110011, 111000\}$$

As classes laterais segundo C são

$000000 + C = \{000000, 100110, 010101, 001011, 011110, 101101, 110011, 111000\}$
$000001 + C = \{000001, 100111, 010100, 001010, 011111, 101100, 110010, 111001\}$
$000010 + C = \{000010, 100100, 010111, 001001, 011100, 101111, 110001, 111010\}$
$000100 + C = \{000100, 100010, 010001, 001111, 011010, 101001, 110111, 111100\}$
$001000 + C = \{001000, 101110, 011101, 000011, 010110, 100101, 111011, 110000\}$
$010000 + C = \{010000, 110110, 000101, 011011, 001110, 111101, 100011, 101000\}$
$100000 + C = \{100000, 000110, 110101, 101011, 111110, 001101, 010011, 011000\}$
$000111 + C = \{000111, 100001, 010010, 001100, 011001, 101010, 110100, 111111\}$

Neste caso temos que:
000000 é o líder de C;
000001 é o líder de $000001 + C$;
000010 é o líder de $000010 + C$;
000100 é o líder de $000100 + C$;
001000 é o líder de $001000 + C$;
010000 é o líde de $010000 + C$;
100000 é o líder de $100000 + C$;
100001, 010010, 001100 são líderes de $000111 + C$.

- Uma determinada classe lateral pode ter mais de um líder.
- Porém, podem-se demonstrar que:

Neste caso temos que:
000000 é o líder de C;
000001 é o líder de $000001 + C$;
000010 é o líder de $000010 + C$;
000100 é o líder de $000100 + C$;
001000 é o líder de $001000 + C$;
010000 é o líde de $010000 + C$;
100000 é o líder de $100000 + C$;
100001, 010010, 001100 são líderes de $000111 + C$.

- Uma determinada classe lateral pode ter mais de um líder.
- Porém, podem-se demonstrar que:

Neste caso temos que:
000000 é o líder de C;
000001 é o líder de $000001 + C$;
000010 é o líder de $000010 + C$;
000100 é o líder de $000100 + C$;
001000 é o líder de $001000 + C$;
010000 é o líde de $010000 + C$;
100000 é o líder de $100000 + C$;
100001, 010010, 001100 são líderes de $000111 + C$.

- Uma determinada classe lateral pode ter mais de um líder.
- Porém, podem-se demonstrar que:

Teorema: Seja C um código linear em F_q^n com distância mínima d. Se um vetor $x \in F_q^n$ é tal que $w(x) \leq \left\lfloor \frac{d-1}{2} \right\rfloor$, então x é o único líder de sua classe.

Proposição: Dois vetores x e y de F_q^n têm a mesma síndrome se, e somente se, $x \in y + C$.

Teorema: Seja C um código linear em F_q^n com distância mínima d. Se um vetor $x \in F_q^n$ é tal que $w(x) \leq \left\lfloor \frac{d-1}{2} \right\rfloor$, então x é o único líder de sua classe.

Proposição: Dois vetores x e y de F_q^n têm a mesma síndrome se, e somente se, $x \in y + C$.

Teorema: Seja C um código linear em F_q^n com distância mínima d. Se um vetor $x \in F_q^n$ é tal que $w(x) \leq \left\lfloor \frac{d-1}{2} \right\rfloor$, então x é o único líder de sua classe.

Proposição: Dois vetores x e y de F_q^n têm a mesma síndrome se, e somente se, $x \in y + C$.

Algoritmo

- Recebida uma palavra y, calcule sua síndrome $S(y) = Hy^t$.
- Se $S(y) = 0$, então a palavra recebida não contém erros.
- Se $S(y) \neq 0$, então a palavra y não pertence ao código.
- Neste caso, procure a classe lateral de y determinada por C e ache seu líder e.
- O vetor enviado é $x = y - e$.

Algoritmo

- Recebida uma palavra y, calcule sua síndrome $S(y) = Hy^t$.
- Se $S(y) = 0$, então a palavra recebida não contém erros.
- Se $S(y) \neq 0$, então a palavra y não pertence ao código.
- Neste caso, procure a classe lateral de y determinada por C e ache seu líder e.
- O vetor enviado é $x = y - e$.

Algoritmo

- Recebida uma palavra y, calcule sua síndrome $S(y) = Hy^t$.
- Se $S(y) = 0$, então a palavra recebida não contém erros.
- Se $S(y) \neq 0$, então a palavra y não pertence ao código.
- Neste caso, procure a classe lateral de y determinada por C e ache seu líder e.
- O vetor enviado é $x = y - e$.

Algoritmo

- Recebida uma palavra y, calcule sua síndrome $S(y) = Hy^t$.
- Se $S(y) = 0$, então a palavra recebida não contém erros.
- Se $S(y) \neq 0$, então a palavra y não pertence ao código.
- Neste caso, procure a classe lateral de y determinada por C e ache seu líder e.
- O vetor enviado é $x = y - e$.

Algoritmo

- Recebida uma palavra y, calcule sua síndrome $S(y) = Hy^t$.
- Se $S(y) = 0$, então a palavra recebida não contém erros.
- Se $S(y) \neq 0$, então a palavra y não pertence ao código.
- Neste caso, procure a classe lateral de y determinada por C e ache seu líder e.
- O vetor enviado é $x = y - e$.

Exemplo

Considere o código C com matriz verificação de paridade

$$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

Temos que $d = 3$ e, portanto, ele corrige um erro. Os vetores de peso menor ou igual a 1 com as suas respectivas síndromes estão relacionados na tabela seguinte.

Exemplo

líder	síndrome
000000	000
000001	001
000010	010
000100	100
001000	011
010000	101
100000	110

Suponhamos que a foi recebida a palavra $y = (010111)$.
Calculamos: $Hy^t = (010)$.
Verificamos então que $e = (000010)$ e a palavra enviada é, portanto: $x = y - e = (010101)$.

Exemplo

líder	síndrome
000000	000
000001	001
000010	010
000100	100
001000	011
010000	101
100000	110

Suponhamos que a foi recebida a palavra $y = (010111)$.
Calculamos: $Hy^t = (010)$.
Verificamos então que $e = (000010)$ e a palavra enviada é, portanto: $x = y - e = (010101)$.

Exemplo

líder	síndrome
000000	000
000001	001
000010	010
000100	100
001000	011
010000	101
100000	110

Suponhamos que a foi recebida a palavra $y = (010111)$.
Calculamos: $Hy^t = (010)$.
Verificamos então que $e = (000010)$ e a palavra enviada é, portanto: $x = y - e = (010101)$.

Exemplo

líder	síndrome
000000	000
000001	001
000010	010
000100	100
001000	011
010000	101
100000	110

Suponhamos que a foi recebida a palavra $y = (010111)$.
Calculamos: $Hy^t = (010)$.
Verificamos então que $e = (000010)$ e a palavra enviada é, portanto: $x = y - e = (010101)$.

Referências

- C.C. Lavor, M.M. Alvez, R.M. Siqueira, S.I.R. Costa, *Uma introdução à teoria de códigos*, SBMAC, 2006.
- A. Hefez, M.L.T. Villela, *Códigos Corretores de Erros*, IMPA, Rio De Janeiro, 2002.
- C. P. Milies, *Breve introdução à teoria dos códigos corretores de erros*, Colóquio de Matemática da Região Centro-Oeste, 2009.

Obrigada!