Suponhamos que, num determinado sistema de comunicação, necessitamos de um código com, no máximo, q^k palavras. Poderemos então usar todas as palavras $a_1a_2\cdots a_k \in \mathbb{F}_q^k$ de comprimento k. Este código será muito pouco eficiente, uma vez que a distância mínima entre palavras é igual a 1.

O Teorema da aula anterior diz-nos que, se quisermos aumentar a eficiência deste código, teremos de aumentar a distância mínima entre as suas palavras. Como poderemos fazer isso? Muito simplesmente, acrescentando a cada palavra $a_1a_2\cdots a_k$ um bloco $c_{k+1}\cdots c_n \in \mathbb{F}_q^{n-k}$ tal que, sempre que

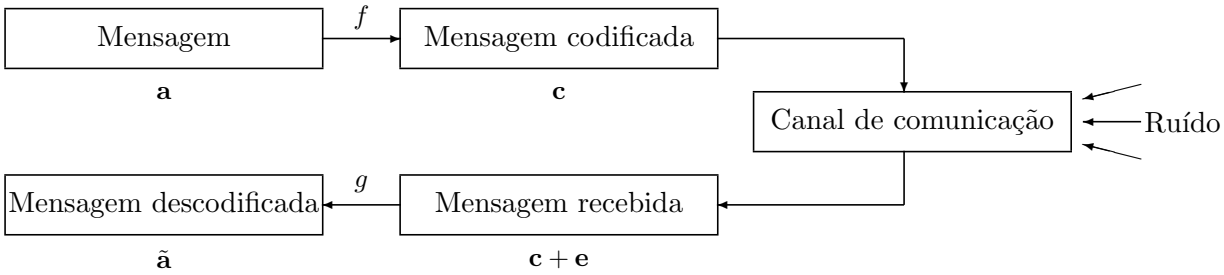
$$d(a_1a_2\cdots a_k, a'_1a'_2\cdots a'_k) = 1$$

então $d(c_{k+1}\cdots c_n, c'_{k+1}\cdots c'_n)$ é máxima, ou seja, igual a $n-k$. Se, além disso, tivermos o cuidado de garantir que $d(c_{k+1}\cdots c_n, c'_{k+1}\cdots c'_n) = n-k+1-i$ sempre que $d(a_1a_2\cdots a_k, a'_1a'_2\cdots a'_k) = i$, teremos um código $\mathcal{C}$ com distância mínima $\delta(\mathcal{C}) = n-k+1$.

Os primeiros k símbolos de cada palavra

$$\mathbf{c} = a_1a_2\cdots a_kc_{k+1}\cdots c_n$$

são a *mensagem original* e os $n-k$ símbolos adicionais são os *símbolos de controle*. A função $\mathbb{F}_q^k \to \mathbb{F}_q^n$ que aplica a palavra $a_1a_2\cdots a_k$ na palavra $a_1a_2\cdots a_kc_{k+1}\cdots c_n$ chama-se um *esquema de codificação*. Estes esquemas de codificação fazem parte de qualquer sistema de comunicação actual, que pode ser descrito do seguinte modo:



A função f é um esquema de codificação. À função $g : \mathbb{F}_q^n \to \mathbb{F}_q^k$ chama-se *esquema de descodificação*. Os esquemas de codificação podem ser apresentados do seguinte modo. Seja H uma matriz $(n-k)\times n$, com entradas em $\mathbb{F}_q$, do tipo $H = [A, I_{n-k}]$, onde A é uma matriz $(n-k)\times k$ e I_{n-k} é a matriz identidade de ordem $n-k$. Os símbolos de controle $c_{k+1}, \ldots, c_n$ podem então ser determinados a partir do sistema de equações $H\mathbf{c}^T = \mathbf{0}$, onde $\mathbf{0}$ denota o vector nulo de $\mathbb{F}_q^{n-k}$.

Exemplo: Seja H a seguinte matriz 3×7 sobre $\mathbb{F}_2$:

$$\begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

O código definido por H será constituído pelas palavras $\mathbf{c} = a_1a_2a_3a_4c_5c_6c_7$, onde os símbolos de controle c_5, c_6, c_7 podem ser calculados resolvendo o sistema $H\mathbf{c}^T = \mathbf{0}$, dados a_1, a_2, a_3, a_4:

$$\begin{cases} a_1 \quad + \quad a_3 \quad + \quad a_4 \quad + \quad c_5 \quad = \quad 0 \\ a_1 \quad + \quad a_2 \quad + \quad a_4 \quad + \quad c_6 \quad = \quad 0 \\ a_1 \quad + \quad a_2 \quad + \quad a_3 \quad + \quad c_7 \quad = \quad 0 \end{cases}$$

Portanto,

$$\begin{cases} c_5 = a_1 + a_3 + a_4 \\ c_6 = a_1 + a_2 + a_4 \\ c_7 = a_1 + a_2 + a_3 \end{cases}$$

pelo que $\mathbf{c} = (a_1, a_2, a_3, a_4, a_1 + a_3 + a_4, a_1 + a_2 + a_4, a_1 + a_2 + a_3)$. Assim, neste exemplo o esquema de codificação é a função linear de $\mathbb{F}_2^4$ em $\mathbb{F}_2^7$, definida por

$$(a_1, a_2, a_3, a_4) \mapsto (a_1, a_2, a_3, a_4, a_1 + a_3 + a_4, a_1 + a_2 + a_4, a_1 + a_2 + a_3),$$

e $\mathcal{C}$ é formado pelas 16 palavras

$$(a_1, a_2, a_3, a_4, a_1 + a_3 + a_4, a_1 + a_2 + a_4, a_1 + a_2 + a_3) \qquad a_1, a_2, a_3, a_4 \in \mathbb{F}_2.$$

Em geral, quando os esquemas de codificação são dados por aplicações lineares, usa-se a seguinte terminologia:

CÓDIGOS (n, k)-LINEARES

Seja $H = [A, I_{n-k}]$ uma matriz $(n-k) \times n$ com entradas em $\mathbb{F}_q$. O conjunto $\mathcal{C}$ dos vectores n-dimensionais $\mathbf{c} \in \mathbb{F}_q^n$ tais que $H\mathbf{c}^T = \mathbf{0}$ diz-se um *código (n, k)-linear* sobre $\mathbb{F}_q$. A matriz H diz-se a *matriz de controle* de $\mathcal{C}$. No caso $q = 2$, $\mathcal{C}$ diz-se um *código binário.*

[Note que o conjunto $\mathcal{C}$ das soluções do sistema $H\mathbf{c}^T = \mathbf{0}$ de equações lineares é um subespaço de dimensão k do espaço vectorial $\mathbb{F}_q^n$]

Exemplos: Os códigos $\mathcal{C}_2$ e $\mathcal{C}_3$ da aula anterior são exemplos de códigos lineares. $\mathcal{C}_2$ é um código $(4,2)$-linear sobre $\mathbb{F}_2$, com matriz de controle

$$H_2 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix},$$

e $\mathcal{C}_3$ é um código $(6,2)$-linear sobre $\mathbb{F}_2$, com matriz de controle

$$H_3 = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Teorema. *Um código (n,k)-linear com matriz de controle H tem distância mínima $\delta(\mathcal{C}) \geq s+1$ se e só se quaisquer s colunas de H são linearmente independentes.*

Demonstração. "$\Rightarrow$" Suponhamos, por hipótese, que $\delta(\mathcal{C}) \geq s+1$. Se H tivesse s colunas, $h_{i_1}, h_{i_2}, \ldots, h_{i_s}$, linearmente dependentes, teríamos

$$a_{i_1}h_{i_1} + a_{i_2}h_{i_2} + \cdots + a_{i_s}h_{i_s} = \mathbf{0}$$

para alguns $a_{i_1}, a_{i_2}, \ldots, a_{i_s} \in \mathbb{F}_q$ não todos nulos. Mas, então, a palavra $\mathbf{c} = c_1c_2\cdots c_n$, definida por

$$c_j = \begin{cases} 0 & \text{se } j \notin \{i_1, i_2, \ldots, i_s\} \\ a_{i_k} & \text{se existe } k \in \{1,2,\ldots,s\} \text{ tal que } j = i_k \end{cases},$$

pertenceria a $\mathcal{C}$, o que é uma contradição, pois $d(\mathbf{c}, \mathbf{0}) \leq s$.

"$\Leftarrow$" Suponhamos, por absurdo, que existem $\mathbf{c}, \mathbf{d} \in \mathcal{C}$ tais que $d(\mathbf{c}, \mathbf{d}) \leq s$. Como $\mathcal{C}$ é linear, $\mathbf{c} - \mathbf{d} \in \mathcal{C}$ e, obviamente, $d(\mathbf{c} - \mathbf{d}, \mathbf{0}) = d(\mathbf{c}, \mathbf{d}) \leq s$. Sejam $x_{i_1}, x_{i_2}, \ldots, x_{i_r}$ as r coordenadas (letras) da palavra $\mathbf{c} - \mathbf{d}$ não nulas ($r \leq s$). Como $\mathbf{c} - \mathbf{d} \in \mathcal{C}$, então $H(\mathbf{c} - \mathbf{d})^T = \mathbf{0}$, ou seja, $h_{i_1}x_{i_1} + h_{i_2}x_{i_2} + \cdots + h_{i_r}x_{i_r} = \mathbf{0}$. Mas isto significa que as r colunas $h_{i_1}, h_{i_2}, \ldots, h_{i_r}$ de H são linearmente dependentes, o que é absurdo por hipótese. ∎

Exemplos: Na matriz H_2 acima, $s = 1$, uma vez que há duas colunas linearmente dependentes (a primeira e a terceira, por exemplo). Na matriz H_3, quaisquer duas colunas são linearmente independentes mas as colunas 1, 3 e 5 são linearmente dependentes, pelo que $s = 2$.

Vimos na aula anterior que, depois de recebida uma palavra $\mathbf{y}$ pelo receptor, a sua *descodificação*, isto é, a determinação da palavra exacta $\mathbf{c}$ que lhe deu origem (isto é, a palavra enviada pelo emissor), pode ser feita determinando a palavra de $\mathcal{C}$ que está mais próxima de $\mathbf{y}$ (princípio do vizinho mais próximo). Claro que isto pode ser feito por "força bruta", determinando a distância de Hamming entre $\mathbf{y}$ e todas as palavras de $\mathcal{C}$. Mas isto é impraticável quando $|\mathcal{C}|$ é muito grande!

Em vez da força bruta, pode usar-se uma abordagem através da matriz H. Para isso, consideremos o espaço vectorial $\mathbb{F}_q^n/\mathcal{C}$ formado por todas as classes

$$\mathbf{a}+\mathcal{C} := \{\mathbf{a}+\mathbf{c} : \mathbf{c} \in \mathcal{C}\}$$

com $\mathbf{a} \in \mathbb{F}_q^n$. Cada classe contém q^k palavras e $\mathbb{F}_q^n$ pode particionar-se em $s+1 = q^{n-k}$ classes de $\mathcal{C}$:

$$\mathbb{F}_q^n = (\mathbf{0}+\mathcal{C}) \cup (\mathbf{a}^{(1)}+\mathcal{C}) \cup \cdots \cup (\mathbf{a}^{(s)}+\mathcal{C}).$$

A palavra recebida $\mathbf{y}$ tem que estar nalguma das classes, digamos $\mathbf{a}^{(i)}+\mathcal{C}$, pelo que $\mathbf{y} = \mathbf{a}^{(i)}+\mathbf{d}$ para algum $\mathbf{d} \in \mathcal{C}$. Se $\mathbf{c}$ foi a palavra transmitida, então o erro é dado por $\mathbf{e} = \mathbf{y}-\mathbf{c} = \mathbf{a}^{(i)}+\mathbf{d}-\mathbf{c} \in \mathbf{a}^{(i)}+\mathcal{C}$. Portanto, o erro $\mathbf{e}$ pertence à mesma classe da palavra $\mathbf{y}$ recebida. Assim, pelo princípio do vizinho mais próximo, para determinar o erro $\mathbf{e}$, e consequentemente a palavra original $\mathbf{y}-\mathbf{e}$, bastará determinar o *líder* da classe de $\mathbf{y}$:

PESO DE UMA PALAVRA; LÍDER DE UMA CLASSE

O *peso* (de Hamming) de $\mathbf{c} \in \mathbb{F}_q^n$ é o número de coordenadas não-nulas de $\mathbf{c}$. Por outras palavras, o peso de $\mathbf{c} \in \mathbb{F}_q^n$ é a distância $d(\mathbf{c}, \mathbf{0})$.

Um elemento de peso mínimo numa classe $\mathbf{a}+\mathcal{C}$ chama-se um *líder* de $\mathbf{a}+\mathcal{C}$.

É claro que se houver mais do que um líder na classe de $\mathbf{y}$ o erro não poderá ser corrigido, uma vez que o receptor não conseguirá decidir qual dos lideres será o vector erro $\mathbf{e}$. Por exemplo, no código $(4,2)$-linear binário $\mathcal{C}$ com matriz de controle

$$\begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix},$$

a lista das 4 classes deste código é a seguinte:

$$\begin{array}{rl} \text{classe } \mathbf{0}+\mathcal{C}=\mathcal{C}: & 0000 \quad 1010 \quad 0111 \quad 1101 \\ \text{outras classes:} & \left\{ \begin{array}{l} 1000 \quad 0010 \quad 1111 \quad 0101 \\ 0100 \quad 1110 \quad 0011 \quad 1001 \\ 0001 \quad 1011 \quad 0110 \quad 1100 \end{array} \right. \end{array}$$

A classe na segunda linha tem dois lideres: 1000 e 0010. Por exemplo, se a palavra recebida for a palavra $\mathbf{y} = 1111$ que está na segunda classe, o vector erro tanto pode ser 1000 como 0010, ou seja, a palavra original pode bem ter sido a palavra 0111 ou 1101. Isto acontece porque $\delta(\mathcal{C}) = 2$ e, portanto, o código não corrige todos os erros singulares. Se a palavra $\mathbf{y}$ recebida for a palavra 1110 na terceira classe, o erro só poderá ser igual a 0100 e, portanto, o receptor descobre imediatamente o erro: a palavra original só pode ter sido a palavra 1010.

[Se no canal de comunicação só ocorrerem no máximo t erros e $\delta(\mathcal{C}) \geq 2t+1$ (portanto $\mathcal{C}$ corrige sempre os t eventuais erros), não poderão existir dois lideres $\mathbf{e}_1$ e $\mathbf{e}_2$ na mesma classe; de facto, se tal fosse possível, $\mathbf{c}_1 := \mathbf{y} - \mathbf{e}_1$ e $\mathbf{c}_2 := \mathbf{y} - \mathbf{e}_2$ seriam palavras de $\mathcal{C}$ tais que $d(\mathbf{c}_1, \mathbf{c}_2) \leq d(\mathbf{c}_1, \mathbf{y}) + d(\mathbf{y}, \mathbf{c}_2) = d(\mathbf{e}_1, \mathbf{0}) + d(\mathbf{e}_2, \mathbf{0}) \leq t + t$, uma contradição com o facto $\delta(\mathcal{C}) \geq 2t+1$]

A classe de cada $\mathbf{y}$ pode ser determinada calculando a sua síndrome:

SÍNDROME DE UMA PALAVRA

O vector $S(\mathbf{c}) = H\mathbf{c}^T$ de comprimento $n-k$ chama-se a *síndrome* de $\mathbf{c} \in \mathbb{F}_q^n$.

Proposição.

(1) $S(\mathbf{c}) = \mathbf{0}$ *se e só se* $\mathbf{c} \in \mathcal{C}$.

(2) $S(\mathbf{c}) = S(\mathbf{d})$ *se e só se* $\mathbf{c}+\mathcal{C} = \mathbf{d}+\mathcal{C}$.

Demonstração. (1) É imediato da definição de $\mathcal{C}$ em termos de H.

(2) $S(\mathbf{c}) = S(\mathbf{d}) \Leftrightarrow H\mathbf{c}^T = H\mathbf{d}^T \Leftrightarrow H(\mathbf{c}-\mathbf{d})^T = \mathbf{0} \Leftrightarrow \mathbf{c}-\mathbf{d} \in \mathcal{C} \Leftrightarrow \mathbf{c}+\mathcal{C} = \mathbf{d}+\mathcal{C}$.

■

No exemplo anterior,

$$\begin{array}{lcccccc}
\text{palavras de } \mathcal{C}: & 0000 & 1010 & 0111 & 1101 & \binom{0}{0} \\
\text{outras classes:} & \left\{\begin{array}{cccccc} 1000 & 0010 & 1111 & 0101 & \binom{1}{0} \\ 0100 & 1110 & 0011 & 1001 & \binom{1}{1} \\ 0001 & 1011 & 0110 & 1100 & \underbrace{\binom{0}{1}}_{\text{Síndromes}} \end{array}\right.
\end{array}$$

ALGORITMO DE DESCODIFICAÇÃO

Dados: palavra $\mathbf{y}$ recebida.

(1) Calcular $S(\mathbf{y})$.

(2) Determinar o líder $\mathbf{e}$ tal que $S(\mathbf{e}) = S(\mathbf{y})$.

(3) A palavra original é a palavra $\mathbf{c} = \mathbf{y} - \mathbf{e}$.

Exemplo: Consideremos o código do exemplo anterior. Se $\mathbf{y} = 1110$ é recebida, começamos por determinar $S(\mathbf{y}) = H\mathbf{y}^T = \binom{1}{1}$. O erro $\mathbf{e}$ será então igual ao líder da respectiva classe, ou seja, a 0100. A palavra original era então igual a $\mathbf{y} - \mathbf{e} = 1010$.

Em códigos lineares muito grandes é praticamente impossível listar todas as classes e determinar os respectivos lideres; por exemplo, um código $(50, 20)$-linear binário tem aproximadamente 10^9 classes. Nesse caso, determina-se directamente o líder da classe da palavra $\mathbf{y}$, determinando a palavra $\mathbf{e}$ de menor peso tal que $H\mathbf{e}^T = S(\mathbf{y})$. No exemplo acima,

$$H\mathbf{e}^T = \binom{1}{1} \Leftrightarrow \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{pmatrix} = \binom{1}{1} \Leftrightarrow \left\{\begin{array}{l} e_1 + e_3 = e_4 \\ e_2 = 1 + e_4 \end{array}\right. \Leftrightarrow$$

$$\Leftrightarrow \mathbf{e} = (0100) \vee \mathbf{e} = (1110) \vee \mathbf{e} = (0011) \vee \mathbf{e} = (1110).$$

O vector (0100) é o que tem menor peso, pelo que $\mathbf{e} = (0100)$.