Compressão de imagem

- Uma imagem é aqui considerada como uma matriz de pequenos quadrados (*pixels*), cada um dos quais tem uma cor específica.
- Uma tal matriz pode ser vista como uma sequência de carateres, digamos por varrimento sucessivo das linhas, à qual pode ser aplicado um dos compressores anteriormente descrito.
- Por exemplo, o *GIF* (*Graphics Interchange Format*) admite 256 cores diferentes para os pixels, ao que corresponde uma palavra em binário de comprimento 8, ou seja 8 bits de informação, pelo que a imagem pode ser descrita por uma sequência destas palavras (mais as dimensões da matriz). O GIF aplica o algoritmo de compressão LZW a esta sequência.
- Uma variante mais recente é o *PNG* (*Portable Network Graphics*), na qual, além de outros melhoramentos no pré-processamento da imagem, e de uma maior variedade de cores, é aplicado o algoritmo de compressão gzip.

- Na *codificação em pirâmide* a estrutura bidimensional das imagens é aproveitada da seguinte forma:
  - a imagem é processada como uma matriz de números, correspondendo à cor de cada pixel; em geral não há variações drásticas na cor de um pixel para a dos pixels adjacentes;
  - calcula-se para cada quadrado de 2×2 pixels em que a imagem é decomposta digamos a média (arredondada para inteiros) dos quatro valores envolvidos, o que conduz a uma nova matriz e a uma imagem com um quarto da área da imagem inicial que é uma aproximação da imagem inicial; a imagem inicial é descrita pelas diferenças dos valores dos pixels originais para os valores aproximados que os substituíram;
  - não sendo em geral grandes as variações entre os valores em pixels vizinhos, as diferenças consideradas serão pequenas, e portanto necessitarão de menos bits para serem descritas;
  - por outro lado, o número de números adicionais a guardar é inferior a $\frac{1}{3}$ do número de pixels originais, sendo as aproximações sucessivas úteis para a recomposição da imagem com resolução progressivamente melhorada.

- Um outro esquema de codificação muito eficaz que explora a estrutura bidimensional das imagens é o *JPEG* (*Joint Photographic Experts Group*) que é usado nas máquinas fotográficas digitais para guardar em poucas centenas de milhar de bytes fotografias com elevada resolução.
- Trata-se do método de codificação pelo cálculo de transformadas associadas às matrizes que descrevem as imagens e que permitem concentrar a informação em regiões restritas da imagem, o que permite a sua compressão mais eficaz.
- O método mais comum utilizado aplica a chamada *transformada do cosseno discreta* (*DCT*). Aqui o termo discreto refere-se ao facto da transformada, que é um caso especial da *transformada de Fourier discreta*, ser aplicada não a funções de variável contínua mas a sequências finitas de números. Eis um exemplo de uma tal transformada:

$$Y_k = X_0 + (-1)^k X_{n-1} + 2 \sum_{j=1}^{n-2} X_j \cos \frac{\pi j k}{n-1}.$$

- A norma *JPEG 2000* fez intervir a teoria das *onduletas* enquanto que normas mais recentes lidam com imagens tridimensionais. A evolução destas normas tem um grande impacto na indústria de produção de equipamento fotográfico. Ver http://www.jpeg.org para mais detalhes.

5 Codificação do canal

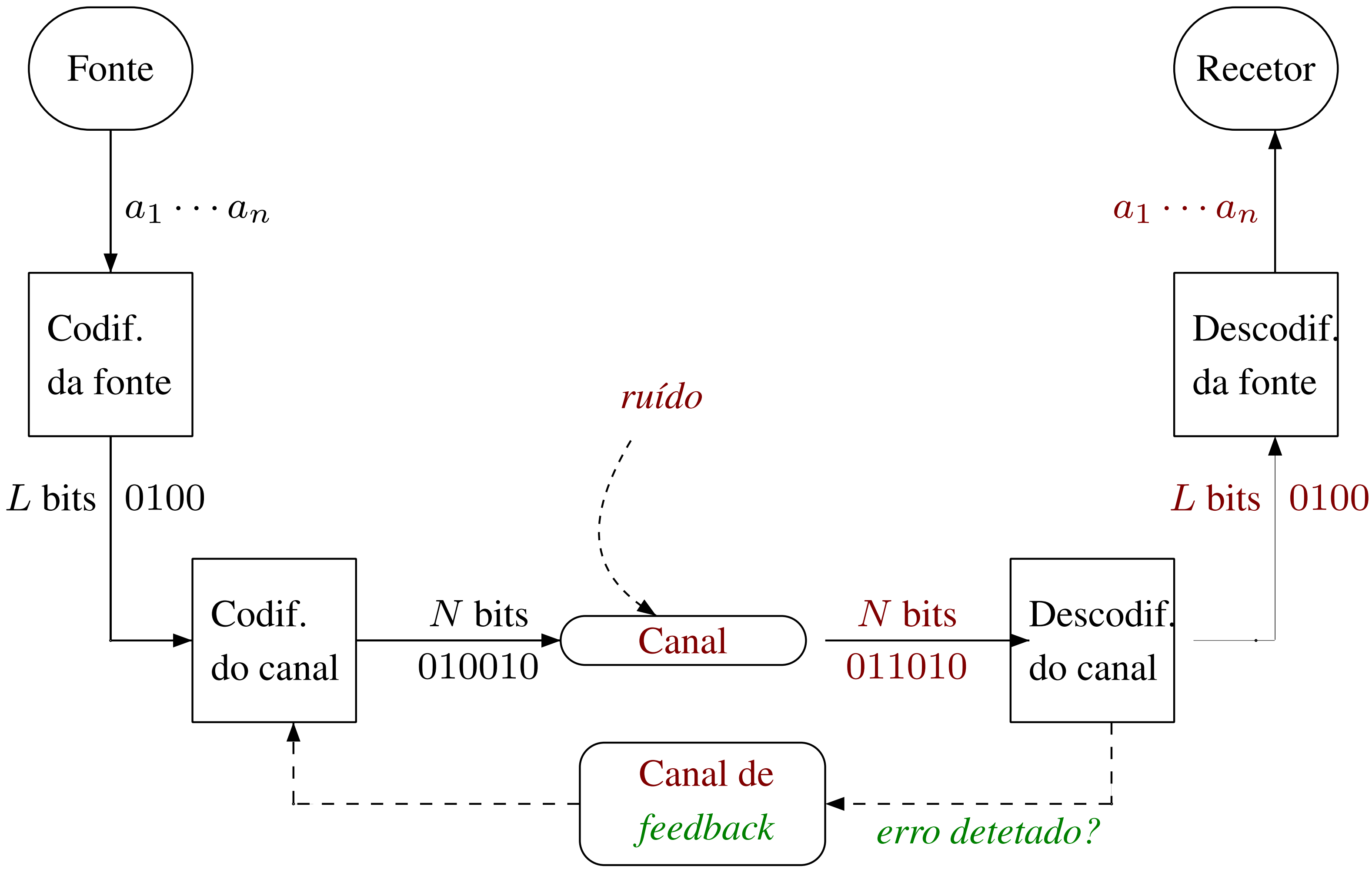
- Recorde-se que na utilização de um canal para a transmissão de informação, com entrada A e saída B, a *informação mútua* mede a quantidade de informação transmitida pelo canal, sendo dada pela diferença

$$I(A;B) = H(A) - H(A|B)$$

  entre a *quantidade de informação* na entrada e o *equívoco* da entrada em relação à saída.

- Sendo o ruído no canal significativo, digamos 1 em 10 ou mesmo 1 em 100 (em contraste com 1 em 10^6), ou se o canal tem de ser usado fidedignamente, (por exemplo se a informação tiver sido comprimida usando um código de Huffman que, reduzindo a redundância através do uso de códigos de comprimento variável, são suscetíveis ao mínimo erro, que pode alterar completamente toda a informação posterior) nesse caso deve ser encontrado algum processo de codificação do canal que permita compensar o ruído corrigindo os erros por ele produzidos.

Esquema de comunicação via um canal com ruído



- O *codificador do canal* parte a mensagem recebida do codificador da fonte em blocos de L bits, transformando cada um destes numa palavra-código com N bits, onde $N > L$. Os $N - L$ bits adicionais servem para recuperar dos erros introduzidos pelo ruído no canal.
- O número de blocos é 2^L e, portanto, o número de palavras-código é 2^L.
- O descodificador do canal recupera o bloco original a partir da palavra com N bits que o canal lhe fornece procurando a palavra-código que *mais provavelmente lhe deu origem*.
- Sendo $N > L$, o número de palavras em N bits que não são palavras-código é $2^N - 2^L$ pelo que a ideia é *separar* suficientemente bem as palavras-código no sentido de que os erros que o canal tem uma probabilidade *não negligenciável* de introduzir permitam, ainda assim, identificar de forma unívoca as palavras-código.

- O descodificador do canal deve detetar erros introduzidos pelo canal e, se possível, corrigi-los.
- Há dois tipos de sistemas que são usados neste contexto, ambos dependendo da deteção eficiente da ocorrência de erros:
  - *ARQ* (*automatic-repeat-request*): caso um erro seja detetado, o canal de *feedback* é usado para solicitar o reenvio da palavra-código até que não sejam detetados erros;
  - *FEC* (*forward-error-correction*): o descodificador do canal procede à correção de erros detetados.

- Qualquer mecanismo de controle de erros tem naturalmente custos. O simples facto de transformar blocos de comprimento L em palavras-código de comprimento $N > L$ obriga a uma utilização mais intensiva do canal e, se o tempo de transmissão da informação está em causa, a uma maior capacidade de transmissão do canal digamos em termos de bit por unidade de tempo (*bit-rate*), o que também se diz a *largura de banda* (*bandwidth*).
- Mais precisamente, suponhamos que o codificador da fonte produz informação à razão de n_s bits por segundo, i.e., 1 bit é emitido cada $T_s = \frac{1}{n_s}$ segundos. Se o codificador do canal transforma cada bloco de L bits numa palavra-código de N bits, então para que o descodificador do recetor possa receber a mensagem ao mesmo ritmo que ela é produzida, então o canal deverá transmitir $n_c = \frac{N}{L} n_s$ bits por segundo, pelo que quanto maior for a razão $\frac{N}{L}$ maior deverá ser a largura de banda do canal.

Razão do canal

- Recorde-se que, para uma fonte M com 2^L símbolos possíveis, todos igualmente prováveis, $H(M) = L$ bits. Sendo cada um deles codificado por uma palavra de comprimento N, a razão da codificação que anteriormente introduzimos é

$$R = \frac{H(M)}{N} = \frac{L}{N} = \frac{n_s}{n_c} = \frac{T_c}{T_s}$$

  onde T_c é o tempo necessário para que o canal transmita 1 bit.

- Assim, quanto maior for R, maior será a informação transmitida por bit, mas o problema é que essa informação pode estar equivocada em relação à que foi fornecida. Há portanto que encontrar um equílibrio entre os custos da largura de banda e a validade da informação.

- O *Teorema Fundamental* de Shannon afirma que, para a transmissão da informação sem erros, tem de se ter $R \leqslant C$, onde C é a capacidade do canal (máximo da informação mútua que produz para todas as fontes de informação a que possa ser aplicado).

- Note-se que, como $\frac{L}{N} = \frac{n_s}{n_c}$, fixados n_s, n_c e L, pode não existir N inteiro satisfazendo aquela equação, caso em que se toma $N = \lfloor L\frac{n_s}{n_c} \rfloor$, o que conduz a problemas de sincronização entre a entrada e a saída.

Regras de descodificação

- Seja $\mathbf{a}_i = a_{i1}a_{i2}\cdots a_{iN}$ uma palavra-código com N bits $(1 \leqslant i \leqslant M = 2^L)$ a transmitir pelo canal e seja $\mathbf{b} = b_1 b_2 \cdots b_N$ a palavra com N bits correspondente recebida à saída do canal.
- Sejam $\mathbf{B}_M$ o conjunto de todas as palavras-código válidas e $\mathbf{B}_M^c$ o conjunto das restantes palavras com N bits, pelo que $\mathbf{B}_N = \mathbf{B}_M \cup \mathbf{B}_M^c$ é o conjunto de todas a palavras com N bits.
- O descodificador do canal deve aplicar uma *regra de descodificação* $D(.)$ a $\mathbf{b}$ para recuperar $\mathbf{a}_i$.
- Seja $P_N(\mathbf{b}|\mathbf{a}_i)$ a probabilidade de ser recebido $\mathbf{b}$ ao ser transmitido $\mathbf{a}_i$. Por exemplo, se o canal não tiver memória, esta probabilidade pode ser calculada a partir das probabilidades "bit-a-bit":

$$P_N(\mathbf{b}|\mathbf{a}_i) = \prod_{t=1}^{N} P(b_t|a_{it}).$$

- Seja $P_N(\mathbf{a}_i)$ a probabilidade a priori para a mensagem correspondente à palavra-código $\mathbf{a}_i$. Então pelo Teorema de Bayes, a probabilidade de que $\mathbf{a}_i$ tenha sido transmitido sendo recebido $\mathbf{b}$ é

$$P_N(\mathbf{a}_i|\mathbf{b}) = \frac{P_N(\mathbf{b}|\mathbf{a}_i)P_N(\mathbf{a}_i)}{P_N(\mathbf{b})}.$$

Descodificação por minimização do erro

- A probabilidade de erro ao descodificar $\mathbf{b}$ na palavra-código $\mathbf{a}_i$ é, portanto, $1 - P_N(\mathbf{a}_i|\mathbf{b})$. A minimização da probabilidade do erro na descodificação é feita quando for maximizada a probabilidade $P_N(\mathbf{a}_i|\mathbf{b})$ como função da escolha $\mathbf{a}_i$ para $D(\mathbf{b})$.

- A *regra de descodificação por minimização do erro* (*minimum-error*) é dada por

$$D_{ME}(\mathbf{b}) = \mathbf{a}$$

  onde $\mathbf{a} \in \mathbf{B}_M$ é tal que

$$P_N(\mathbf{a}|\mathbf{b}) \geqslant P_N(\mathbf{a}_i|\mathbf{b}) \quad \varphi i.$$

- Usando a expressão para $P_N(\mathbf{a}_j|\mathbf{b})$ e cancelando $P_N(\mathbf{b})$, obtemos a condição de máximo equivalente

$$P_N(\mathbf{b}|\mathbf{a})P_N(\mathbf{a}) \geqslant P_N(\mathbf{b}|\mathbf{a}_i)P_N(\mathbf{a}_i) \quad \forall i.$$

Descodificação por escolha mais provável

- A *regra de descodificação por escolha mais provável* (*maximum-likelihood*) é dada por

$$D_{ML}(\mathbf{b}) = \mathbf{a}$$

onde $\mathbf{a} \in \mathbf{B}_M$ é tal que

$$P_N(\mathbf{b}|\mathbf{a}) \geqslant P_N(\mathbf{b}|\mathbf{a}_i) \quad \forall i.$$

- Note-se que tanto $D_{ME}(\mathbf{b})$ como $D_{ML}(\mathbf{b})$ não ficam necessariamente definidos de forma única, podendo haver várias escolhas para $\mathbf{a}$ que maximizem as probabilidades em causa.

- Note-se também que nenhuma das condições de máximo sobre as probabilidades implica a outra, embora no caso de todas as palavras-código $\mathbf{a}_i$ terem a mesma probabilidade, as condições sejam equivalentes.

Exemplo

- Consideremos um BSC com probabilidade 0.4 de troca do bit e um codificador do canal com $(L, N) = (2, 3)$. O codificador usa portanto $4 = 2^2$ palavras-código de comprimento 3. Suponhamos que as respetivas probabilidades de ocorrência são dadas por

palavra-código	$\mathbf{a}_1 = 000$	$\mathbf{a}_2 = 011$	$\mathbf{a}_3 = 101$	$\mathbf{a}_4 = 110$
$P_N(\mathbf{a}_i)$	0.4	0.2	0.1	0.3

- Suponhamos que uma palavra-código é transmitida pelo BSC e que é recebida a palavra $\mathbf{b} = 111$. Temos

$$P_N(\mathbf{b}|\mathbf{a}_1) = P_N(111|000) = P(1|0)P(1|0)P(1|0) = 0.064$$

$$P_N(\mathbf{b}|\mathbf{a}_2) = P_N(111|011) = P(1|0)P(1|1)P(1|1) = 0.144$$

$$P_N(\mathbf{b}|\mathbf{a}_3) = P_N(111|101) = P(1|1)P(1|0)P(1|1) = 0.144$$

$$P_N(\mathbf{b}|\mathbf{a}_4) = P_N(111|110) = P(1|1)P(1|1)P(1|0) = 0.144$$

  pelo que qualquer uma das palavras-código $\mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4$ poderia ser escolhida para $D_{ML}(\mathbf{b})$.

- Por outro lado, temos

$$P_N(\mathbf{b}|\mathbf{a}_1)P_N(\mathbf{a}_1) = 0.064 \times 0.4 = 0.0256$$
$$P_N(\mathbf{b}|\mathbf{a}_2)P_N(\mathbf{a}_2) = 0.144 \times 0.2 = 0.0288$$
$$P_N(\mathbf{b}|\mathbf{a}_3)P_N(\mathbf{a}_3) = 0.144 \times 0.1 = 0.0144$$
$$P_N(\mathbf{b}|\mathbf{a}_4)P_N(\mathbf{a}_4) = 0.144 \times 0.3 = 0.0432$$

  pelo que devemos tomar $D_{ME}(\mathbf{b}) = \mathbf{a}_4$.
- Como, para fontes arbitrárias ou, como anteriormente argumentámos, para fontes resultantes da codificação eficiente, as mensagens deverão ser igualmente prováveis, é razoável assumir que esta propriedade se verifique para as mensagens que chegam ao codificador do canal. Neste caso, os dois esquemas de descodificação do canal D_{ME} e D_{ML} são equivalentes.

Distância de Hamming

- Dadas duas palavras binárias do mesmo comprimento $\mathbf{a} = a_1 a_2 \cdots a_N$ e $\mathbf{b} = b_1 b_2 \cdots b_N$, a *distância de Hamming* entre elas é o número de posições em que diferem:

$$d(\mathbf{a}, \mathbf{b}) = |i \in \{1, \ldots, N\} : a_i \neq b_i|.$$

Proposição 5.1 *A distância de Hamming é uma métrica no conjunto de todas as palavras binárias de comprimento N: para quaisquer palavras $\mathbf{a}, \mathbf{b}, \mathbf{c}$ deste tipo,*

1. $d(\mathbf{a}, \mathbf{b}) \geqslant 0$, *verificando-se a igualdade se e só se* $\mathbf{a} = \mathbf{b}$;
2. $d(\mathbf{a}, \mathbf{b}) = d(\mathbf{b}, \mathbf{a})$;
3. $d(\mathbf{a}, \mathbf{b}) + d(\mathbf{b}, \mathbf{c}) \geqslant d(\mathbf{a}, \mathbf{c})$.$\square$

- Por exemplo, $d(11010001, 00010010) = 4$, $d(00010010, 01010011) = 2$ e $d(11010001, 01010011) = 2$.

Regra de descodificação de Hamming para BSC's

- Consideremos um BSC com probabilidade q de troca de bit. Seja $D = d(\mathbf{b}, \mathbf{a_i})$ a distância de Hamming entre a mensagem recebida e uma palavra-código $\mathbf{a}_i$. Recorde-se que, na *descodificação por escolha mais provável*, pretendemos maximizar a probabilidade $P_N(\mathbf{b}|\mathbf{a}_i)$.
- Ora, pela definição da distância de Hamming, temos

$$P_N(\mathbf{b}|\mathbf{a}_i) = q^D(1-q)^{N-D}.$$

Assim, caso $q < 0.5$, $P_N(\mathbf{b}|\mathbf{a}_i)$ é maximizado quando $D = d(\mathbf{b}, \mathbf{a}_i)$ for minimizado.

- Seja $\mathbf{b}$ uma palavra binária de comprimento N recebida à saída de um BSC para uma entrada $\mathbf{a}_i \in \mathbf{B}_M$, de M possíveis palavras binárias de comprimento N.
- A *regra de descodificação de Hamming* consiste no seguinte:
  - caso exista um *único* $\mathbf{a} \in \mathbf{B}_M$ tal que

    $$d(\mathbf{a}, \mathbf{b}) \leqslant d(\mathbf{a}_i, \mathbf{b}) \quad \forall i,$$

    tomar para a descodificação de $\mathbf{b}$ a palavra-código $\mathbf{a}$ tal que a condição acima se verifica, detetando um erro de pelo menos $d(\mathbf{a}, \mathbf{b})$ bits caso esta distância seja positiva (i.e., se $\mathbf{b} \notin \mathbf{B}_M$);
  - caso contrário a regra limita-se a detetar um erro de pelo menos $d(\mathbf{a}, \mathbf{b})$ bits para qualquer $\mathbf{a}$ que minimize a distância de Hamming a $\mathbf{b}$.

Exemplo

Consideremos o seguinte código de canal:

Mensagem $(L = 2)$	palavra-código $(N = 3)$
00	000
01	001
10	011
11	111

Temos $M = 2^L = 4$ e $R = \frac{L}{N} = \frac{2}{3}$. Eis como a regra de descodificação de Hamming procede:

$\mathbf{b} = b_1 b_2 b_3$	palavras-código mais próximas	descodificação	erro detetado
011	011	011	$-$
010	$000, 011$	$-$	1bit
100	000	000	1bit
101	$001, 111$	$-$	1bit
110	111	111	1bit

Deteção de erros

- Para um canal mal comportado, não podemos esperar corrigir todos os erros.
- Mas, seria bom podermos corrigir com segurança todos os erros que não envolvam a troca de mais do que um certo número t de bits.
- Dado um código por blocos $\mathcal{K}$, a *distância mínima de* $\mathcal{K}$ é dada por

$$d(\mathcal{K}) = \min\{d(\mathbf{a}, \mathbf{b}) : \mathbf{a}, \mathbf{b} \in \mathcal{K} \text{ e } \mathbf{a} \neq \mathbf{b}\}.$$

- Dizemos que um código por blocos $\mathcal{K}$ *deteta (todas as combinações de) até t erros* se, alterando entre 1 e t bits em qualquer palavra-código, não obtivermos nenhuma outra palavra-código.

Limiar da deteção de erros

Proposição 5.2 *O código por blocos $\mathcal{K}$ deteta até t erros se e só se $d(\mathcal{K}) > t$.*

Prova. Seja $\mathbf{a}$ uma palavra-código de $\mathcal{K}$ e seja $\mathbf{b}$ uma palavra obtida de $\mathbf{a}$ por troca de t bits. Para qualquer outra palavra-código $\mathbf{a}' \neq \mathbf{a}$, pela desigualdade triangular temos

$$d(\mathbf{b}, \mathbf{a}') \geqslant d(\mathbf{a}, \mathbf{a}') - d(\mathbf{a}, \mathbf{b}) \geqslant d(\mathcal{K}) - t.$$

Logo, se $d(\mathcal{K}) > t$, então $d(\mathbf{b}, \mathbf{a}') > 0$ para toda a palavra-código $\mathbf{a}'$, pelo que a regra de descodificação de Hamming deteta um erro na tentativa de descodificação de $\mathbf{b}$, ou seja esta regra de descodificação deteta até t erros. Reciprocamente, se $d(\mathcal{K}) \leqslant t$, então existem palavras-código $\mathbf{a}$ e $\mathbf{a}'$ tais que $d(\mathbf{a}, \mathbf{a}') \leqslant t$, pelo que a regra de descodificação de Hamming não é capaz de detetar $d(\mathbf{a}, \mathbf{a}') \leqslant t$ erros. $\square$

Correção de erros

- Dizemos que o código por blocos $\mathcal{K}$ *corrige (todas as combinações de) até t erros* se a partir de uma palavra obtida duma palavra-código $\mathbf{b}$ pela alteração de até t bits, a regra de descodificação de Hamming conduz unicamente à palavra-código $\mathbf{a}$.

Limiar da correção de erros

Proposição 5.3 *Um código por blocos $\mathcal{K}$ corrige até t erros se e só se $d(\mathcal{K}) > 2\,t$.*

Prova. Seja $\mathbf{a}$ uma palavra-código de $\mathcal{K}$ e seja $\mathbf{b}$ uma palavra obtida de $\mathbf{a}$ por troca de t bits. Para qualquer outra palavra-código $\mathbf{a}' \neq \mathbf{a}$, pela desigualdade triangular temos, como acima,

$$d(\mathbf{b}, \mathbf{a}') \geqslant d(\mathbf{a}, \mathbf{a}') - d(\mathbf{a}, \mathbf{b}) \geqslant d(\mathcal{K}) - t.$$

Logo, se $d(\mathcal{K}) > 2t$, então $d(\mathcal{K}) - t > t$ e $d(\mathbf{b}, \mathbf{a}') > t$ para toda a palavra-código $\mathbf{a}'$, pelo que a regra de descodificação de Hamming descodifica $\mathbf{b}$ em $\mathbf{a}$, ou seja esta regra de descodificação corrige até t erros. Reciprocamente, se $d(\mathcal{K}) \leqslant 2t$, então existem palavras-código $\mathbf{a}$ e $\mathbf{a}'$ que diferem em r bits com $r \leqslant 2t$. Trocando $s = \lceil \frac{r}{2} \rceil$ desses bits em $\mathbf{a}$ obtemos uma palavra $\mathbf{b}$ cuja distância de Hamming a $\mathbf{a}$ e a $\mathbf{a}'$ é no máximo t, pelo que a regra de descodificação de Hamming não permite recuperar das s alterações efetuadas em $\mathbf{a}$. $\square$

Exemplo

- O *código controle de paridade* para blocos de comprimento L acrescenta um bit a cada bloco que indica a paridade do número de 1's no bloco. Por exemplo, para $L = 2$, temos o seguinte código:

mensagem	palavra-código
00	000
01	011
10	101
11	110

  Note-se que $d(\mathcal{K}) = 2$, pelo que este código permite detetar erros num só bit.

- ***Exercício:*** mostre que $d(\mathcal{K}) = 2$ para qualquer $L \geqslant 2$.

Exemplo

- Um *código por repetição* é definido para mensagens de um só bit ($L = 1$) pela transformação do bit a em N cópias $\underbrace{aa\cdots a}_{N}$ de a, com N ímpar.
  A regra de descodificação de Hamming, aplicada a uma mensagem b, produz o bit que aparecer maioritariamente em b.
  Note-se que $d(\mathcal{K}) = N$, pelo que um tal código permite corrigir até $\frac{N-1}{2}$ erros e detetar até $N - 1$ erros.

Exemplo

- Considere-se o código $\mathcal{K}$ para mensagens de $L = 3$ bits dado pela seguinte tabela:

mensagem $(L = 3)$	palavra-código $(N = 6)$
000	000000
001	001110
010	010101
011	011011
100	100011
101	101101
110	110110
111	111000

Por cálculo direto, obtém-se $d(\mathcal{K}) = 3$, donde se conclui que este código corrige erros de um só bit.
No entanto, esta correção só é garantidamente possível a partir de mensagens resultantes da alteração até um bit das palavras-código. Por exemplo, a mensagem 111111 está igualmente próxima de $011011, 101101, 110110$, o que deteta um erro de 2 bits mas não o permite corrigir.

- O valor de $d(\mathcal{K})$ determina até que ponto podemos detetar e corrigir erros.
- Surge assim naturalmente a questão de saber, para valores dados d e N, qual é o número máximo M de mensagens (blocos) que podemos codificar.
- Ou, dados N e L, qual é a melhor correção de erros (i.e., máximo valor de $d(\mathcal{K})$) que é possível obter para códigos por blocos de comprimento N codificando todos os blocos de comprimento de L?
- Ou ainda, dado que se pretende um código para blocos de comprimento L com correção de erros até t bits, qual é o valor mínimo do comprimento das palavras-código?

- O número máximo de palavras-código num código $\mathcal{K}$ por blocos de comprimento N com $d(\mathcal{K}) = d$ representa-se por $B(N, d)$.
- ***Exercício:*** Mostre que:
  - $B(N, 1) = 2^N$;
  - $B(N, 2) = 2^{N-1}$;
  - $B(N, 2t + 1) = B(N + 1, 2t + 2)$;
  - $B(N, N) = 2$.

O majorante de Hamming

Teorema 5.4 (Majorante de Hamming) *Se o código por blocos de comprimento N corrige t erros, então o número M de palavras-código satisfaz a seguinte desigualdade:*

$$M \leqslant \frac{2^N}{\sum_{i=0}^{t} \binom{N}{i}}.$$

Prova. Seja $V(N,t)$ o número de palavras de comprimento N cuja distância a uma dada palavra-código $\mathbf{a}_j$ é no máximo t, ou seja o número de palavras na *bola (fechada)* de raio t centrada em $\mathbf{a}_j$. Sendo $\binom{N}{i}$ o número de palavras a distância i de $\mathbf{a}_j$, temos $V(N,t) = \sum_{i=0}^{t} \binom{N}{i}$.
Sendo o nosso código $\mathcal{K}$ corretor de t erros, nenhuma palavra pode estar a distância menor ou igual a t de mais do que uma palavra-código. Logo as bolas de raio t centradas nas palavras-código não partilham pontos e portanto o número de elementos na sua união é $M \cdot V(N,t)$.
Havendo somente 2^N palavras de comprimento N, temos
$M \cdot \sum_{i=0}^{t} \binom{N}{i} \leqslant 2^N$, donde segue a desigualdade de Hamming. $\square$

Casos particulares

- Em particular, o número máximo M de palavras-código num código corretor de t erros usando palavras de comprimento N satisfaz a desigualdade

$$M = B(N, 2t+1) \leqslant \frac{2^N}{\sum_{i=0}^{t} \binom{N}{i}}.$$

- No caso particular de $t = 1$, obtemos

$$M = B(N, 3) \leqslant \frac{2^N}{\binom{N}{0} + \binom{N}{1}} = \frac{2^N}{N+1}.$$

- Aplicando a igualdade $B(N, 2t+1) = B(N+1, 2t+2)$, resulta que

$$B(N, 4) \leqslant \frac{2^{N-1}}{N}.$$

- O majorante de Hamming dá $B(4,3) \leqslant \lfloor \frac{16}{5} \rfloor = 3$ mas $B(4,3) = 2$: