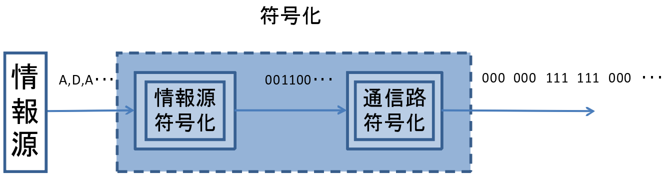
情報理論
第 13 回 通信路符号化における
パリティ検査とハミング符号

堀田 政二
工学部 情報工学科

符号化 (encoding) のモデル (再掲)



情報源符号
{A,B,C,D}

情報源符号
A:00, B:01
C:10, D:11
（2元符号）

通信路符号
0:000
1:111

- **通信路符号化**
  - **送信側で情報を担うビット (情報ビット) に加えて，情報を持たない余分なビット (冗長ビット) を符号に付加**
  - **いかに少ない冗長度で誤りを検出・訂正できるかが通信路符号化の課題**

誤り検出・訂正の原理 (再掲)

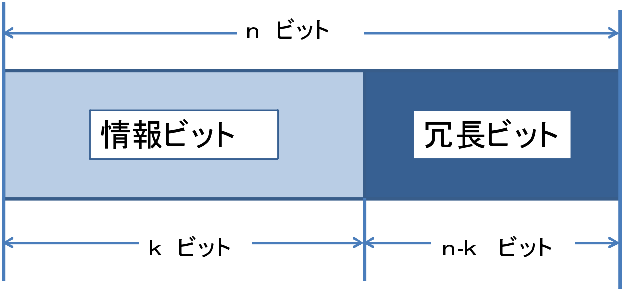
- 符号語と非符号語
  - 符号語: 情報源の記号に対応する符号．送信側から送られる符号はすべて符号語
  - 非符号語: 符号語以外の符号

- 誤りを検出できる条件
  - 誤りがあれば受信符号は符号語に一致せず，必ず非符号語になること
- 誤りを訂正するためには
  - 誤りビットの位置を特定する必要．検出よりもさらに大きな冗長度

誤りを検出・訂正するには，非符号語が必要．非符号語は記号に対応しておらず，情報伝送の観点からは冗長

- 誤り検出のための符号: 誤り検出符号 (EDC, error detecting code)
- 訂正のための符号: 誤り訂正符号 (ECC, error correcting code)

通信路符号の構成，(n,k) 符号



情報源記号の種類: 2^k **個，記号は** k **ビット長の符号．送信に用いる符号はこれよりも長い全長が** n **ビット** $(n>k)$ **の符号**

- **記号に割り当てられた符号語は** 2^k **個．それ以外の** 2^n-2^k **個の符号は非符号語**
- **全ビット長に含まれる情報ビットの割合 (符号化率，または情報速度) は** $\eta = k/n$ **と表される．そのため，冗長度は** $\rho = 1-\eta$

情報源が二元符号の場合の例

全ビット長n	1		2		3	
情報源記号		非符号語		非符号語		非符号語
A	0	なし	00	01, 10	000	100, 110等
B	1	なし	11	01, 10	111	011, 001等
誤り検出	×		○		○	
誤り訂正	×		×		○	

符号語に含まれる 0 と 1 の個数の多数決で 0 か 1 かを判定

- **情報源記号の種類 $2^1 (k = 1)$ 個**
- $n = 1$ **の時，符号化率** $\eta = 1$**，冗長度** 0**．誤りを検出，訂正できない**
- $n = 2$ **の時，符号化率** $\eta = 1/2$**，冗長度** $1/2$**．誤りが一つであれば検出できるが，訂正はできない**
- $n = 3$ **の時，符号化率** $\eta = 1/3$**，冗長度** $2/3$**．誤りが一つであれば，誤り検出・訂正が可能**

シャノンの第 2 基本定理 (Shannon's 2nd fundamental theorem)

誤りを検出，訂正するためには冗長を際限なく増大させればよいが，そうすると伝送速度が著しく低下してしまう，というジレンマに陥ってしまう

通信路符号化定理

通信路容量 C の通信路において，伝送速度 $R(< C)$ で情報を伝送するとき，ある $\delta(> 0)$ が存在し，$R < C - \delta$ ならば，誤り確率をいくらでも小さくできる

すなわち，雑音があっても $R < C$ ならば，情報伝送の信頼性は 100% にできる $\rightarrow$ ジレンマの解消

- どのような通信路であっても通信路容量 C というものが求まる
- C よりも小さい伝送速度 R で通信すれば，符号化部と復号化部を工夫することにより誤り確率をいくらでも小さくできる

ハミング距離 (誤り検出符号)

長さ k **の** 2 **元符号列** $\boldsymbol{x} = x_1 x_2 \cdots x_k$**，** $\boldsymbol{y} = y_1 y_2 \cdots y_k$ **を考える．**
ハミング距離 (Hamming distance) **は次式によって定義される**:

- $d_H(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{k} x_i \oplus y_i$
- **ただし，** $x_i \in \{0, 1\}$, $y_i \in \{0, 1\}$

ここで演算子 $\oplus$ **は排他的論理和** (exclusive OR) **であり，次のような演算を表す**:

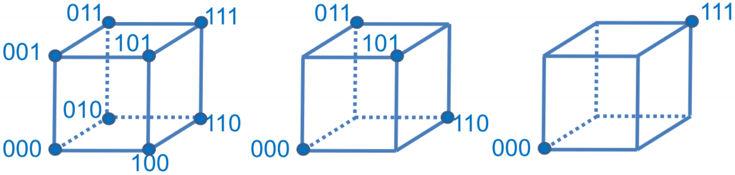
$$0 \oplus 0 = 0,\ 0 \oplus 1 = 1,\ 1 \oplus 0 = 1,\ 1 \oplus 1 = 0$$

ハミング距離が d **であるということは，** k **桁のビット列のうち** d **箇所が異なることを意味する**
【例】 $k = 12$ **の場合の例**

$$\begin{cases} \boldsymbol{x} & = 001011010010 \\ \boldsymbol{y} & = 000011111010 \end{cases}$$

$\boldsymbol{x} \oplus \boldsymbol{y} = 001000101000$ **となりハミング距離は** 3

ハミング距離の幾何学的な解釈



ハミング距離 1 ハミング距離 2 ハミング距離 3

- $k = 3$ の場合，各符号は 3 次元空間上の立方体の頂点と見做せる (左図)
- ある頂点から他の頂点へ至る辺の数がハミング距離 d_H となる．d_H が互いに 2 だけ離れた 4 つの頂点を符号とすると，1 か所の誤り検出が可能 (中央図)
- d が互いに 3 だけ離れた 2 つの頂点を符号とすると，1 か所の誤りであれば検出と誤り訂正が可能 (右図)

パリティ検査法 (誤り検出符号)

パリティ検査法 (parity check procedure)

ビット列中の 0，または 1 の合計が奇数か偶数かということを予め決めておき，それによって誤りの発生を検出する方法

- **長さ k の 2 元符号列 (情報ビット) を $x = x_1 x_2 \cdots x_k$ とする**
- **x に 1 ビットの検査ビット (チェックビット) を最後に付与した長さ $n = k + 1$ の符号列を $w = x_1 x_2 \cdots x_k c$ とする**
- **$s = x_1 \oplus x_2 \oplus \cdots \oplus x_k \oplus c$ を計算する**
- **偶数パリティ検査法: $s = 0$ となるように検査ビット c を決めて送信する方法**
- **奇数パリティ検査法: $s = 1$ となるように検査ビット c を決めて送信する方法**

パリティ検査法の例

長さ $k = 4$ の 2 元符号列 0111，1010，0011，1000 に対して，偶数パリティ検査法で誤りを検出できるようにチェックビットを付与してみる:

- w_1 =01111
- w_2 =10100
- w_3 =00110
- w_4 =10001

赤い部分がチェックビット．これらの w を送信し，受信側が以下のような符号 y を受け取ったとする:

- y_1 =01111，$s = 0$:**偶数**
- y_2 =10100，$s = 0$:**偶数**
- y_3 =10110，$s = 1$:**奇数** $\rightarrow$ **誤りが発生！**
- y_4 =10001，$s = 0$:**偶数**

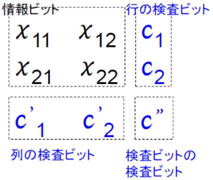
この方法では偶数個の誤りは検出できないことに注意

垂直水平パリティ符号 (誤り訂正符号)

パリティ検査法では 1 ビットの誤りを検出できるが訂正はできない．そこでパリティ検査法を拡張して，1 ビットの誤りを訂正できるようにすることを考える

【例】

- 長さ 4 の 2 元符号列を $x = x_1x_2x_3x_4$ とする
- 各ビット列を 2×2 の行列に並び替え，各行・各列に 1 ビットつずつチェックビットを付与



ここで，$x_1 = x_{11}, x_2 = x_{12}, x_3 = x_{21}, x_4 = x_{22}$ であり，
$c_1 = x_{11} \oplus x_{12}, c_2 = x_{21} \oplus x_{22}$ は行の検査ビット，
$c'_1 = x_{11} \oplus x_{21}, c'_2 = x_{12} \oplus x_{22}$ は列の検査ビット，
$c'' = c_1 \oplus c_2 \oplus c'_1 \oplus c'_2$ は検査ビットの検査ビットである

垂直水平パリティ符号の具体例

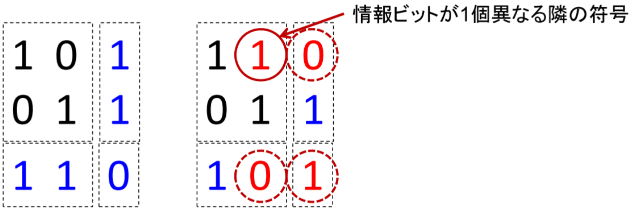
- 2 元符号列を $X = 1001$ とする
- $c_1 = x_{11} \oplus x_{12} = 1$
- $c_2 = x_{21} \oplus x_{22} = 1$
- $c_1' = x_{11} \oplus x_{21} = 1$
- $c_2' = x_{12} \oplus x_{22} = 1$
- $c'' = c_1 \oplus c_2 = c_1' \oplus c_2' = x_{11} \oplus x_{12} \oplus x_{21} \oplus x_{22} = 0$

1	0	1
0	1	1
1	1	0

符号を $(x_{11}x_{12}x_{21}x_{22}c_1c_2c_1'c_2'c'')$ として送るとすると，この例の場合，送信符号は w =100111110

誤り検出の例と訂正の方法

- $x_2 = x_{12}$ が 0 から 1 に誤ったとする．すなわち，受信符号が $y = 110111110$ であったとする
- 検査ビット c_1，c_2' により 1 行 2 列に誤りが生じたことが分かる
- 訂正は誤りが生じた部分が 1 である誤りパターン $e = 0100000$ を用いて $w = y \oplus e$ により実現できる



ハミング符号 (誤り訂正符号)

- 垂直水平パリティ符号よりも効率的な誤り訂正符号
- 単一誤りを訂正できる符号の中で最も効率が良い

具体例として，情報ビット数が 4，検査ビット数が 3 からなる 7 ビットのハミング符号を

$$\boldsymbol{w} = (w_1, w_2, \ldots, w_7) = (x_1, x_2, x_3, x_4, c_1, c_2, c_3) = (\boldsymbol{x}, \boldsymbol{c})$$

と表記する．ハミング符号では検査ビットを

$$\begin{aligned} c_1 &= x_1 \oplus x_2 \oplus x_3 \\ c_2 &= x_2 \oplus x_3 \oplus x_4 \\ c_3 &= x_1 \oplus x_2 \oplus x_4 \end{aligned}$$

と定める (理由は後述)．すなわち，各 c_i は三つの x_i の排他的論理和によって定める

パリティ検査方程式 (parity check equation)

$\boldsymbol{w} = (w_1, w_2, ..., w_7) = (x_1, x_2, x_3, x_4, c_1, c_2, c_3)$ **に対して，検査ビットの決め方から次の関係式が成り立つ**:

$$\begin{cases} w_1 \oplus w_2 \oplus w_3 \oplus w_5 = 0 \ (x_1 \oplus x_2 \oplus x_3 \oplus c_1 = 0) \\ w_2 \oplus w_3 \oplus w_4 \oplus w_6 = 0 \ (x_2 \oplus x_3 \oplus x_4 \oplus c_2 = 0) \\ w_1 \oplus w_2 \oplus w_4 \oplus w_5 = 0 \ (x_1 \oplus x_2 \oplus x_4 \oplus c_3 = 0) \end{cases}$$

このように，情報ビットと検査ビットの関係を $= 0$ **の形で表した式の組をパリティ検査方程式と呼ぶ**

シンドローム (syndrome)

受信符号 $\boldsymbol{y} = (y_1, y_2, \ldots, y_7)$ を受け取った場合，誤りがなければ受信符号はパリティ検査方程式を満たすはずである．すなわち

$$\begin{cases} s_1 = y_1 \oplus y_2 \oplus y_3 \oplus y_5 \\ s_2 = y_2 \oplus y_3 \oplus y_4 \oplus y_6 \\ s_3 = y_1 \oplus y_2 \oplus y_4 \oplus y_7 \end{cases}$$

を計算し，誤りが生じていなければ，すべての s_i が 0 となる．しかし，1 ビットの誤りが発生した場合には，少なくとも一つの s_i が 1 となるはずである．しかも，ハミング符号では，この 3 つの s_i の組合せで誤りの位置も特定できる．このように誤りの位置を特定するための s_i をシンドロームと呼ぶ

$(7,4)$ ハミング符号の誤り箇所とシンドローム

誤り箇所	誤りパターン							シンドローム		
	x1	x2	x3	x4	c1	c2	c3	s1	s2	s3
なし	0	0	0	0	0	0	0	0	0	0
左から1桁	1	0	0	0	0	0	0	1	0	1
左から2桁	0	1	0	0	0	0	0	1	1	1
左から3桁	0	0	1	0	0	0	0	1	1	0
左から4桁	0	0	0	1	0	0	0	0	1	1
左から5桁	0	0	0	0	1	0	0	1	0	0
左から6桁	0	0	0	0	0	1	0	0	1	0
左から7桁	0	0	0	0	0	0	1	0	0	1

この表 (エラーテーブル) を参照すれば，誤り箇所がわかる

問題

- 【13.1】$x = 0110101$ と $y = 0001110$ のハミング距離を求めよ．また，$x \oplus y$ を求めよ
- 【13.2】長さが 4 の情報ビット $x = 0011$ に対する垂直水平パリティ符号を求めよ
- 【13.3】次のような $(9, 4)$ の垂直水平パリティ符号が受信された．誤りを訂正せよ

$$y = 100101011$$

- 【13.4】長さが 4 の情報ビット $x = 0101$ に対するハミング符号を求めよ
- 【13.5】次のような $(7, 4)$ のハミング符号が受信された．誤りを訂正せよ

$$y = 0111100$$