

# 講義「情報理論」

## 第13回 通信路符号化法(2)

情報理工学部門 情報知識ネットワーク研究室  
喜田拓也

# 組織符号, 線形符号 (おさらい)

$k$ 個の情報記号に対して検査記号を求め, それを末尾に追加することで信頼性を高める符号長 $n$ の等長符号を**組織符号**という

$$W = x_1 x_2 \cdots x_k c_1 c_2 \cdots c_{n-k}$$

符号長 $n$ , 情報記号数 $k$ の組織符号を **$(n, k)$ 符号**と書く.

$(n, k)$ 符号の効率 $\eta$ は,  $\eta = k/n$

検査記号が情報記号の線形の式で与えられる符号を**線形符号**と呼ぶ. 線形符号では, 任意の二つの符号語について, 成分ごとの和をとったものも符号語になっている. これは**線形符号となるための必要十分条件**である

**単一パリティ検査符号**は, 一つの誤りを検出できる**誤り検出符号**

**水平垂直パリティ検査符号**は, 1個の誤りが訂正でき, 2個の誤りを検出することができる**誤り検出訂正符号**

# (7,4)ハミング符号 (おさらい)

情報ビット  $x_1, x_2, x_3, x_4$  に対し, 検査ビットを

$$c_1 = x_1 + x_3 + x_4$$

$$c_2 = x_1 + x_2 + x_3$$

$$c_3 = x_2 + x_3 + x_4$$

と計算し,  $\mathbf{w} = (x_1, x_2, x_3, x_4, c_1, c_2, c_3)$  という符号語に組織符号化する符号を(7,4)ハミング符号という

このとき, 受信語  $\mathbf{y} = (y_1, y_2, \dots, y_7)$  に対するシンδροーム  $\mathbf{s} = (s_1, s_2, s_3)$  は

$$s_1 = y_1 + y_3 + y_4 + y_5$$

$$s_2 = y_1 + y_2 + y_3 + y_6$$

$$s_3 = y_2 + y_3 + y_4 + y_7$$

であり, これを計算することで誤りを検出・訂正することができる

(7,4)ハミング符号

$x_1$	$x_2$	$x_3$	$x_4$	$c_1$	$c_2$	$c_3$
0	0	0	0	0	0	0
1	0	0	0	1	1	0
0	1	0	0	0	1	1
1	1	0	0	1	0	1
0	0	1	0	1	1	1
1	0	1	0	0	0	1
0	1	1	0	1	0	0
1	1	1	0	0	1	0
0	0	0	1	1	0	1
1	0	0	1	0	1	1
0	1	0	1	1	1	0
1	1	0	1	0	0	0
0	0	1	1	0	1	0
1	0	1	1	1	0	0
0	1	1	1	0	0	1
1	1	1	1	1	1	1

# 今日の内容

8.2.2 生成行列と検査行列

8.2.3 一般のハミング符号

8.2.5 ハミング距離と誤り訂正能力

# 線形符号の生成行列

線形符号の場合、符号化と復号の計算を行列の式で記述できる  
たとえば、(7,4)ハミング符号の符号語 $w$ は、

$$w = (x_1, x_2, x_3, x_4, x_1 + x_3 + x_4, x_1 + x_2 + x_3, x_2 + x_3 + x_4)$$

と書くことができる.  $x = (x_1, x_2, x_3, x_4)$ とし、

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \begin{array}{l} \leftarrow x_1 \\ \leftarrow x_2 \\ \leftarrow x_3 \\ \leftarrow x_4 \end{array}$$

という行列 $G$ を考えると、 $w = xG$ として符号化できる

このように、 $k$ 個の情報記号からなるベクトル $x$ に掛けたとき、それに対応する符号語が生成されるような行列 $G$ を生成行列という

$(n, k)$ 線形符号の生成行列は  $k \times n$  行列となる

# 検査行列とシンδροーム

(7,4)ハミング符号のパリティ検査方程式の係数を並べた行列を

$$\mathbf{H} = \begin{array}{cccccc} & w_1 & w_2 & w_3 & w_4 & w_5 & w_6 & w_7 \\ \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} & \leftarrow \text{1つ目} & & & & & & \\ & & \leftarrow \text{2つ目} & & & & & \\ & & & \leftarrow \text{3つ目} & & & & \end{array} \quad \mathbf{H}^T = \begin{array}{ccc} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} & \begin{array}{l} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ w_7 \end{array} \end{array}$$

とおく. これを用いるとパリティ検査方程式は

$$\mathbf{wH}^T = \mathbf{0}$$

と書ける.  $\mathbf{H}^T$ は $\mathbf{H}$ の転地行列,  $\mathbf{0}$ は全成分が0のベクトルを表す.

この行列 $\mathbf{H}$ をパリティ検査行列, または単に検査行列と呼ぶ

$(n, k)$ 線形符号のパリティ検査方程式の数は, 検査記号数 $n - k$ に等しいので, 検査行列は $(n - k) \times n$ 行列となる

検査行列 $\mathbf{H}$ を用いると, シンδροームの計算は受信語 $\mathbf{y}$ に対して,

$$\mathbf{s} = (s_1, s_2, s_3) = \mathbf{yH}^T.$$

よって, シンδροームは誤りパターンを $\mathbf{e}$ とすると次が成り立つ.

$$\mathbf{s} = (\mathbf{w} + \mathbf{e})\mathbf{H}^T = \mathbf{wH}^T + \mathbf{eH}^T = \mathbf{eH}^T.$$

Try 問8.5 6

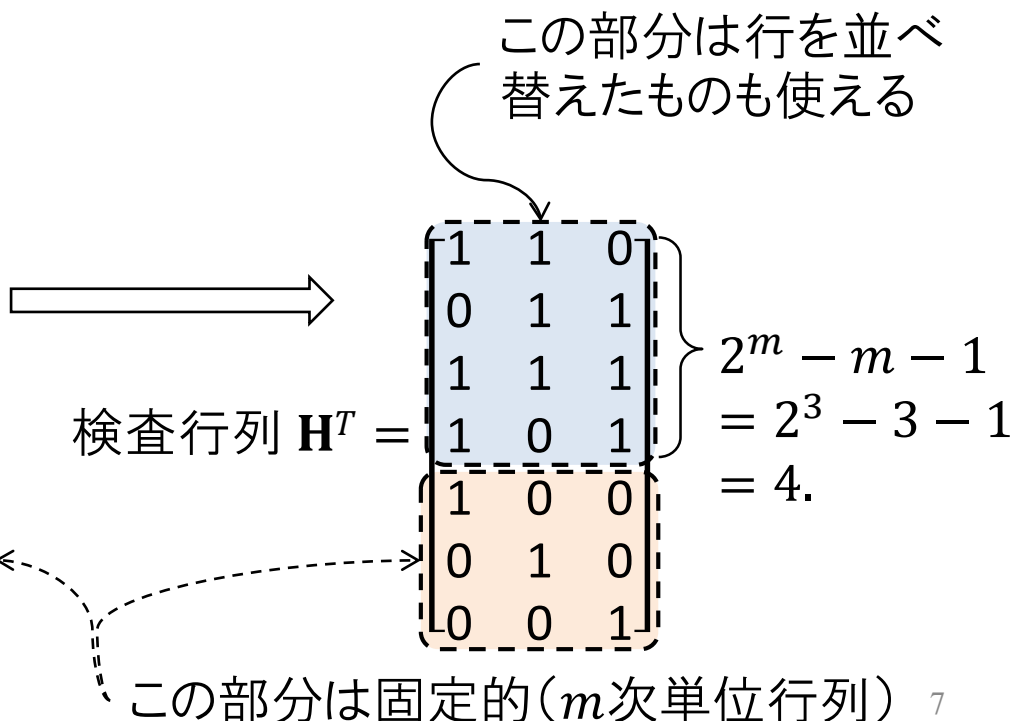
# 検査行列 $H^T$ の構成

(7,4)ハミング符号では, 3ビットのシンδροーム( $2^3 = 8$ 通りある)で受信語に含まれる単一の誤りを見分けられた

全ゼロ(000)のパターンは誤りがないことを表すので, **残り7つのパターンを使って, ちょうど7ビットの符号語の誤りを識別している**

単一誤りに対するシンδροーム

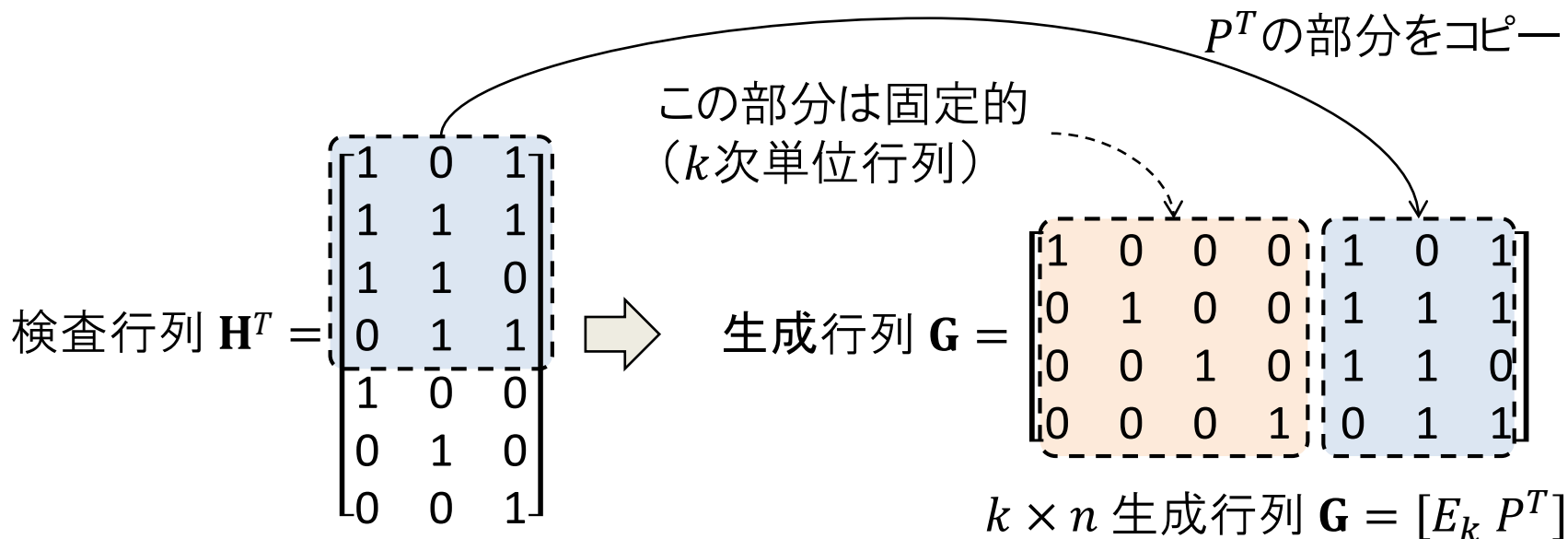
誤りパターン							シンδροーム		
$e_1$	$e_2$	$e_3$	$e_4$	$e_5$	$e_6$	$e_7$	$s_1$	$s_2$	$s_3$
1	0	0	0	0	0	0	1	1	0
0	1	0	0	0	0	0	0	1	1
0	0	1	0	0	0	0	1	1	1
0	0	0	1	0	0	0	1	0	1
0	0	0	0	1	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0
0	0	0	0	0	0	1	0	0	1
0	0	0	0	0	0	0	0	0	0



# 生成行列Gの構成

検査行列 $\mathbf{H}^T$ から、対応する生成行列 $\mathbf{G}$ を機械的に構成できる

1.  $\mathbf{G}$ の左の部分は常に $k \times k$ 単位行列
2.  $\mathbf{G}$ の右の部分に検査行列の転置 $\mathbf{H}^T$ の上部 $k$ 行を配置する



$m \times n$  検査行列  $\mathbf{H} = [P \ E_m]$

$P$ :  $m \times k$  行列,  $E_m$ :  $m \times m$  単位行列

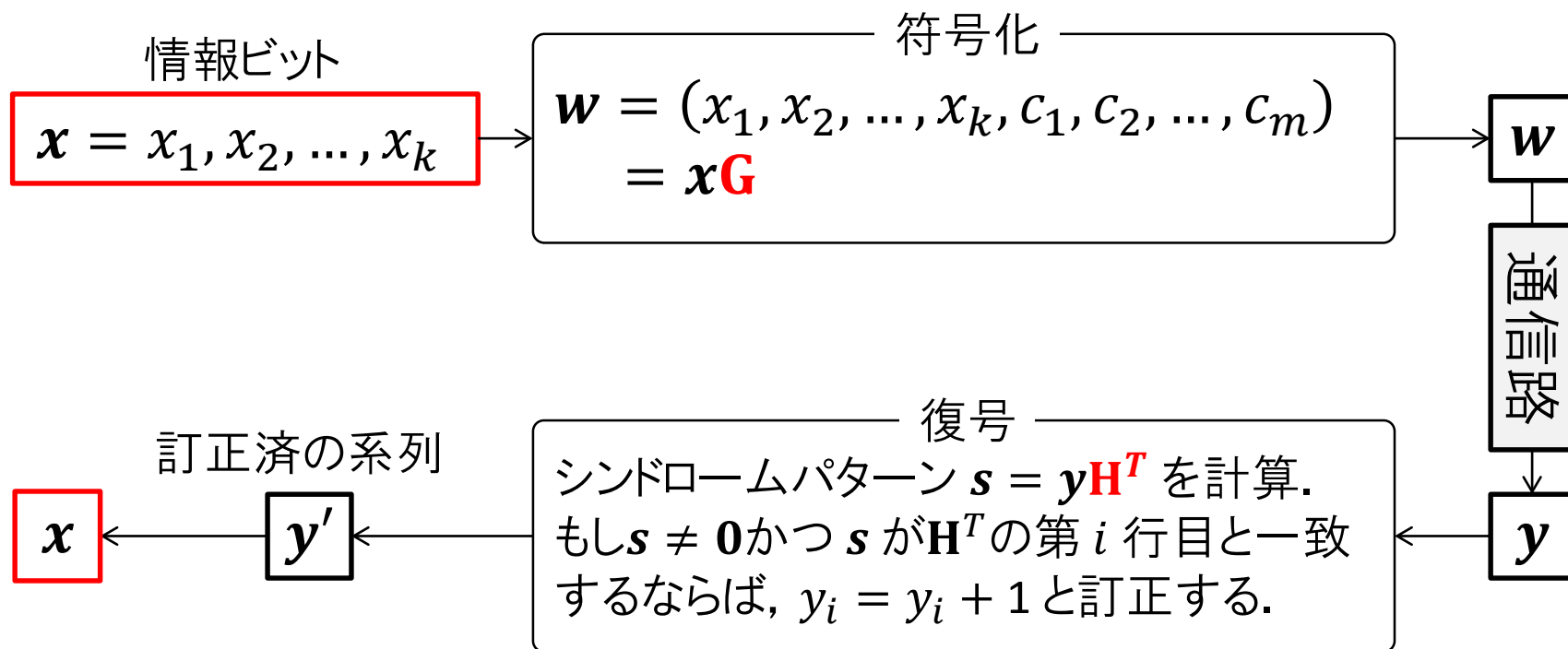
[Try 問8.6~8.9](#)



# 一般のハミング符号

任意の  $m \geq 2$  について、符号語長が  $n = 2^m - 1$ 、情報ビットが  $k = 2^m - m - 1$  となるハミング符号を構成することができる

(与えられた  $m$  に対して、 $m \times n$  の検査行列  $\mathbf{H} = [P \ E_m]$  および  $k \times n$  の生成行列  $\mathbf{G} = [E_k \ P^T]$  を構成して、下図のように符号化・復号すればよい)



※ ハードウェア化すれば、各ビットを並列に処理することが可能(8.2.4節)

# ちょっと休憩

# ハミング距離

2つの $n$ 次元ベクトル  $\mathbf{u} = (u_1, u_2, \dots, u_n)$ ,  $\mathbf{v} = (v_1, v_2, \dots, v_n)$  に対して, **ハミング距離**  $d_H(\mathbf{u}, \mathbf{v})$  を次のように定義する.

$$d_H(\mathbf{u}, \mathbf{v}) \triangleq \sum_{i=1}^n \delta(u_i, v_i). \quad \text{ただし, } \delta(u, v) = \begin{cases} 0 & (u = v), \\ 1 & (u \neq v). \end{cases}$$

すなわち,  $d_H(\mathbf{u}, \mathbf{v})$  は,  $\mathbf{u}$  と  $\mathbf{v}$  の **互いに異なる成分の数** である  
ハミング距離は **距離の3公理** を満たす

## 距離の3公理

任意の  $n$  次元ベクトル  $v_1, v_2, v_3$  に対して以下のことが成り立つ.

- i.  $d_H(v_1, v_2) \geq 0$  であり, 等号が成立するのは  $v_1 = v_2$  のときに限る
- ii.  $d_H(v_1, v_2) = d_H(v_2, v_1)$
- iii.  $d_H(v_1, v_2) + d_H(v_2, v_3) \geq d_H(v_1, v_3)$  (三角不等式)

# ハミング重み

$n$ 次元ベクトル $\boldsymbol{v}$ に対し, **ハミング重み** $w_H(\boldsymbol{v})$ を次のように定義する.

$$w_H(\boldsymbol{v}) \triangleq d_H(\boldsymbol{v}, \mathbf{0}) \quad (\mathbf{0} \text{ はすべての成分が0のベクトル})$$

すなわち,  $w_H(\boldsymbol{v})$  は $\boldsymbol{v}$ の**0でない成分の数**である

たとえば,  $\boldsymbol{v} = (1, 1, 0, 0)$  に対して,  $w_H(\boldsymbol{v}) = 2$  となる

ハミング距離はハミング重みを用いて次のように表せる.

$$d_H(\boldsymbol{u}, \boldsymbol{v}) = w_H(\boldsymbol{u} - \boldsymbol{v}).$$

たとえば,  $\boldsymbol{u} = (1, 1, 0, 0)$ ,  $\boldsymbol{v} = (0, 1, 1, 0)$  に対して,

$$d_H(\boldsymbol{u}, \boldsymbol{v}) = w_H(\boldsymbol{u} - \boldsymbol{v}) = w_H((1, 0, 1, 0)) = 2.$$

符号語 $\boldsymbol{w}$ を送り $t$ 個の誤りが生じて $\boldsymbol{y} = \boldsymbol{w} + \boldsymbol{e}$ が受信されたとする.  
このとき, 次が成り立つ.

$$d_H(\boldsymbol{w}, \boldsymbol{y}) = w_H(\boldsymbol{e}) = t.$$

# 符号の最小距離と最小ハミング重み

## 定義8.12

与えられた2元符号Cに関して、任意の2つの異なる符号語の間のハミング距離の最小値、すなわち、

$$d_{\min} \triangleq \min_{u, v \in C, u \neq v} d_H(u, v)$$

計算が面倒

を、符号Cの**最小ハミング距離**(または単に**最小距離**)と呼ぶ

## 定義8.13

与えられた2元符号Cに関して、全ゼロを除くすべての符号語のハミング重みの最小値

$$w_{\min} \triangleq \min_{w \in C, w \neq 0} w_H(w)$$

計算が楽!

を、符号Cの**最小ハミング重み**と呼ぶ

# 線形符号の最小ハミング距離

## 定理8.5

線形符号では、最小ハミング重みと最小距離が一致する

証明：定義より.

$$d_{\min} = \min_{u, v \in C, u \neq v} d_H(u, v)$$

$$= \min_{u, v \in C, u \neq v} w_H(u - v)$$

$d_H(u, v) = w_H(u - v)$  だから

$$= \min_{w \in C, w \neq 0} w_H(w)$$

$$= W_{\min}$$

線形符号だから、符号語どうしの和(差)も符号語になる



いったい何の意味があるの・・・？

符号の誤り訂正能力と関係があります！

# 最小距離と誤り訂正能力の関係

## 限界距離復号法:

$t_1$ は選べる

$d_{\min} \geq 2t_1 + 1$  を満たす整数  $t_1$  を定め,  $t_1$  以下の誤り訂正を行う

$t_1$  の最大値  $t_0 = \lfloor (d_{\min} - 1)/2 \rfloor$  を符号Cの誤り訂正能力という

$t_2 = d_{\min} - 2t_1 - 1$  とおくと,  $t_1 + 1 \leq t \leq t_1 + t_2$  個の誤りについて, 訂正はできないが検出は可能

$t_1$  を大きくする

正しく復号される確率は増大

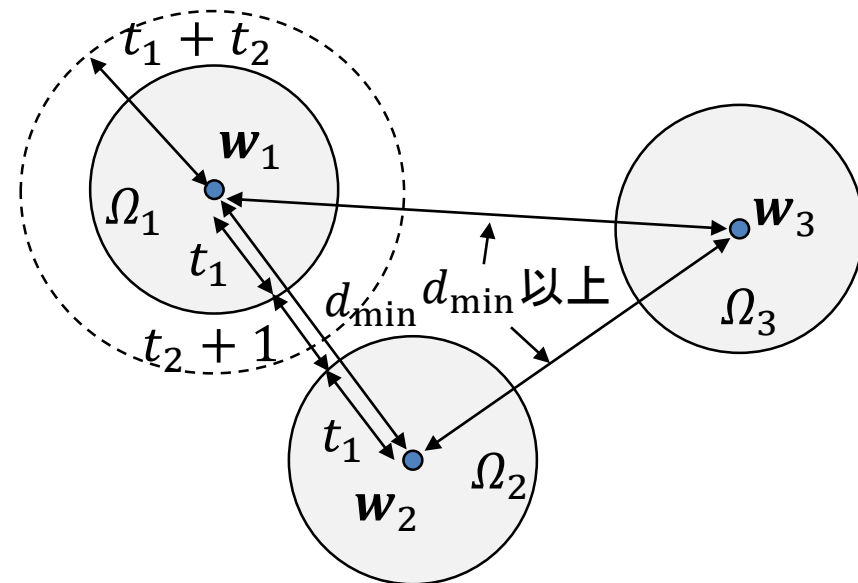
誤って復号される確率も増大

$t_1$  を小さくする

正しく復号される確率は減少する

検出可能な誤りの個数が増える

(検出できれば, 再送要求などの救済措置ができる)



# 誤り訂正能力の例

【例】  $d_{\min} = 5$  の符号による誤りの訂正と検出

$t_1$	訂正可能な誤り	訂正できないが検出可能な誤り
0	—	1～4個
1	1個	2～3個
2	2個	—

## ハミング符号

符号語長に関係なくすべて、最小距離  $d_{\min} = 3$  (定理8.6)

誤り訂正能力  $t_0 = 1$

( (7,4)ハミング符号の場合, 最小距離  $d_{\min} =$  最小ハミング重み  $= 3$  )

## 水平垂直パリティ検査符号

最小距離  $d_{\min} = 4$ , 誤り訂正能力  $t_0 = 1$

単一誤り訂正・2重誤り検出符号



# 限界距離復号法と最尤復号法の違い

ビット誤り率  $p < 0.5$  の2元対称通信路(BSC)を考える

間違いの数

符号語  $\mathbf{w}$  を送って,  $\mathbf{y}$  が受信される確率は,  $t = d_H(\mathbf{w}, \mathbf{y})$  とすると,

$$P(\mathbf{y}|\mathbf{w}) = p^t(1-p)^{n-t}.$$

$p < 0.5$  では,  $t$  について単調減少

## 最尤復号法の場合

$\mathbf{y}$  が受信されたとき,  $P(\mathbf{y}|\mathbf{w})$  が最大となる符号語  $\mathbf{w}$  に復号

$p < 0.5$  のBSCでは,  $t = d_H(\mathbf{w}, \mathbf{y})$  が最小となる  $\mathbf{w}$  に復号

$\mathbf{y}$  とハミング距離が一番近い符号語  $\mathbf{w}$  が送られたと常に推定

## 限界距離復号法の場合

$\mathbf{y}$  が受信されたとき, ハミング距離  $t = d_H(\mathbf{w}, \mathbf{y})$  が  $t_1$  以下の最も近い符号語  $\mathbf{w}$  に復号

どの  $\mathbf{w}$  についても  $d_H(\mathbf{w}, \mathbf{y}) > t_1$  となった場合は推定を放棄

正しく復号される確率  $P_c$

誤って復号される確率  $P_e$

実現の容易さ

○ 最尤復号法 > 限界距離復号法

最尤復号法 > 限界距離復号法 ○

最尤復号法 < 限界距離復号法 ○

# 今日のまとめ

8.2.2 生成行列と検査行列

8.2.3 一般のハミング符号

8.2.5 ハミング距離と誤り訂正能力

次回

巡回符号