

高信頼システム 04 ～誤り訂正符号～

張山昌論

誤り訂正符号の分類 (誤りの性質の観点)

- **ブロック符号**
 - ランダム誤り訂正
 - ハミング符号 (磁気テープで使用) , BCH符号
 - バースト誤り訂正
 - Reed-Solomon(RS)符号
(CD,DVDなど光ディスクで使用)
- **畳み込み符号**
 - ランダム誤り訂正
 - ビタビ(Viterbi)符号 (放送で使用)
 - バースト誤り訂正
 - Hagelberger符号

ブロック符号と畳み込み符号

- ブロック符号：
現在送られる1区切りのデータから誤り訂正。
過去のデータが不要なので短時間で訂正ができるので通信の他にコンピュータのデータ処理等にも使用される。
- 畳み込み符号：
過去と現在のデータを用いて誤り訂正をするので通信状態の悪い移動体通信や遠距離通信等に多く使用される。処理時間がかかる。

ランダム誤りとバースト誤り

- **ランダム誤り**

- 送信した各ビットごとに独立に発生する誤り.
- 例) 通信や放送での伝送路雑音, 装置中の熱雑音, 磁気/光記録の媒体ノイズなど

- **バースト誤り**

- 部分的に集中して連続的に発生する誤り.
- 例) 通信路の断線, 光ディスクのキズやゴミ, 磁気記録の磁性体の欠陥
- バースト誤りにおいて, 最初に誤りが発生したビットから最後の誤りまでのビットまでの長さのことを「バースト長」という.

パリティ検査とは？

情報ビット	送信ビット
0 0 0	0 0 0 0
0 0 1	0 0 1 1
0 1 0	0 1 0 1
0 1 1	0 1 1 0
1 0 0	1 0 0 1
1 0 1	1 0 1 0
1 1 0	1 1 0 0
1 1 1	1 1 1 1

「パリティビット」を追加→1の個数を偶数にする

垂直水平パリティ符号の構成

									水平パリティ
データ1	0	0	0	1	0	1	1	0	1
データ2	0	1	1	1	0	1	1	1	0
データ3	0	0	1	1	1	0	1	0	0
	0	1	0	1	1	0	1	1	1
									垂直パリティ

パリティは排他的論理和で求める

排他的論理和

$$p \oplus p = 0$$

$$p \oplus 0 = 0 \oplus p = p$$

$$p_1 \oplus p_2 = p_2 \oplus p_1 \quad \text{交換法則}$$

$$(p_1 \oplus p_2) \oplus p_3 = p_1 \oplus (p_2 \oplus p_3) \quad \text{結合法則}$$

1の数の偶数/奇数を決定するために使うことができる

排他的論理和の例

偶数？ 奇数？

$$1 \oplus 1 =$$

$$1 \oplus 1 \oplus 1 =$$

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

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

パリティ検査を数式で表現すると

水平パリティ

データ1	0	0	0	1	0	1	1	0	1
データ2	0	1	1	0	0	1	1	1	0
データ3	0	0	1	1	1	0	1	0	0
	0	1	0	1	1	0	1	1	1

垂直パリティ

(データiの水平パリティ) = (データiの全ビットの排他的論理和)

(ビットiの垂直パリティ) = (ビットiの全ビットの排他的論理和)

垂直水平パリティ符号による誤り訂正

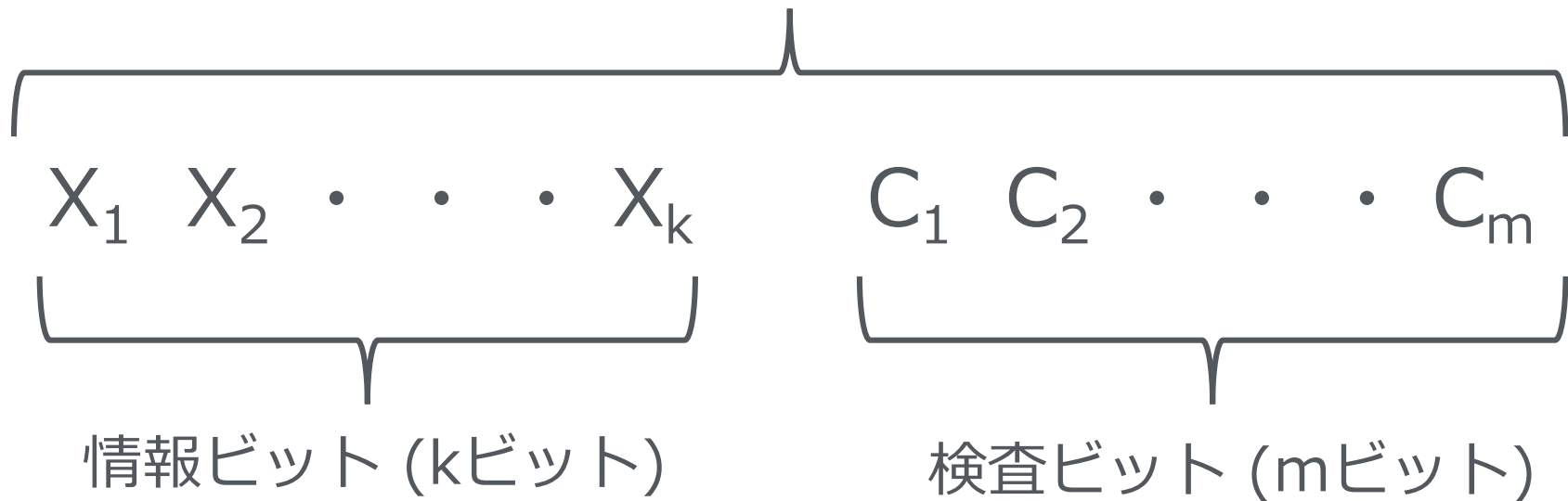
下記のデータには1箇所のみ誤りがある。
その誤りを見つけよ

									水平パリティ
データ1	0	0	0	1	0	1	1	0	1
データ2	0	1	1	0	0	1	1	1	0
データ3	0	0	1	1	1	0	1	0	0
	0	1	0	1	1	0	1	1	1
									垂直パリティ

2個以上の誤りには対応できない

ハミング符号 (パリティ検査の拡張)

全体の符号ビット ($n = k + m$ ビット)



誤りの位置を特定できるように設計

目的： N ビットの符号の中に1ビット以下の誤り
→ 誤り訂正可能とする

(n,k)符号

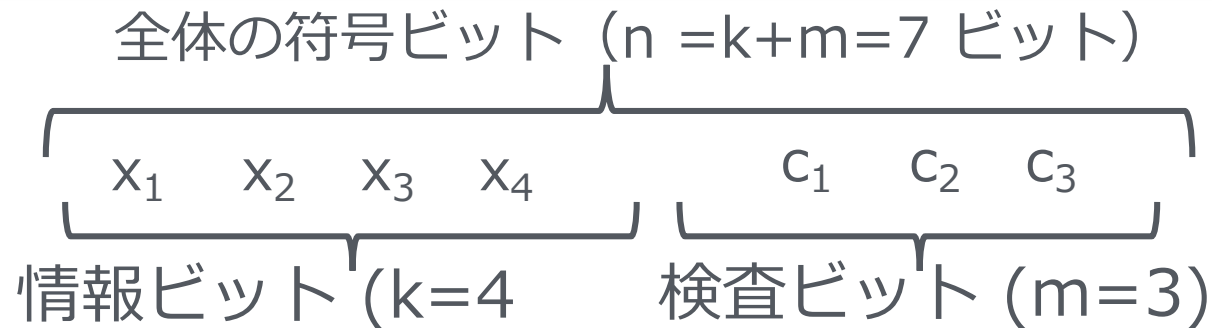
nビットの符号で情報ビットがkビットである符号

(n, k)ハミング符号の例：

符号全体のビット数(n)	3	7	15	31	63
情報ビット数(k)	1	4	11	26	57
検査ビット数(m)	2	3	4	5	6

$2^m - m = k + 1$ の関係が成立する

(7,4)ハミング符号の検査ビット



s_1	s_2	s_3	誤りの位置
0	0	0	誤りなし
0	0	1	x_1
0	1	0	x_2
0	1	1	x_3
1	0	0	x_4
1	0	1	c_1
1	1	0	c_2
1	1	1	c_3

パリティ s_1, s_2, s_3 を2進数とみなし, 計算した値が「誤りが発生した」ビットを表すようにしてみよう

x4, c1, c2, c3に誤りがある場合

s ₁	s ₂	s ₃	誤りの位置
0	0	0	誤りなし
0	0	1	x ₁
0	1	0	x ₂
0	1	1	x ₃
1	0	0	x ₄
1	0	1	c ₁
1	1	0	c ₂
1	1	1	c ₃

$$s_1 = x_4 \oplus c_1 \oplus c_2 \oplus c_3$$

排他的論理和

$$p \oplus p = 0$$

$$p \oplus 0 = 0 \oplus p = p$$

続き

s_1	s_2	s_3	誤りの位置
0	0	0	誤りなし
0	0	1	x_1
0	1	0	x_2
0	1	1	x_3
1	0	0	x_4
1	0	1	c_1
1	1	0	c_2
1	1	1	c_3

$$s_2 = x_2 \oplus x_3 \oplus c_2 \oplus c_3$$

$$s_3 = x_1 \oplus x_3 \oplus c_1 \oplus c_3$$

続き

誤りが無い場合：

$$s_1 = s_2 = s_3 = 0$$

$$0 = x_4 \oplus c_1 \oplus c_2 \oplus c_3$$

$$0 = x_2 \oplus x_3 \oplus c_2 \oplus c_3$$

$$0 = x_1 \oplus x_3 \oplus c_1 \oplus c_3$$

連立方程式を解けば良い

$$c_1 = x_2 \oplus x_3 \oplus x_4$$

$$c_2 = x_1 \oplus x_3 \oplus x_4$$

$$c_3 = x_1 \oplus x_2 \oplus x_4$$

(補足) 連立方程式の解き方

$$0 = x_4 \oplus c_1 \oplus c_2 \oplus c_3 \quad \text{式(1)}$$

$$0 = x_2 \oplus x_3 \oplus c_2 \oplus c_3 \quad \text{式(2)}$$

$$0 = x_1 \oplus x_3 \oplus c_1 \oplus c_3 \quad \text{式(3)}$$

式(1)+(2)

$$0 = x_2 \oplus x_3 \oplus x_4 \oplus c_1 \oplus c_2 \oplus c_2 \oplus c_3 \oplus c_3$$

$$0 = x_2 \oplus x_3 \oplus x_4 \oplus c_1$$

両辺に c_1 を足して

$$c_1 = x_2 \oplus x_3 \oplus x_4$$

c_2, c_3 も同様に求める.

得られた(7,4)ハミング符号

符号	x_1	x_2	x_3	x_4	c_1	c_2	c_3
u_1	0	0	0	0	0	0	0
u_2	0	0	0	1	1	1	1
u_3	0	0	1	0	1	1	0
u_4	0	0	1	1	0	0	1
u_5	0	1	0	0	1	0	1
u_6	0	1	0	1	0	1	0
u_7	0	1	1	0	0	1	1
u_8	0	1	1	1	1	0	0

符号	x_1	x_2	x_3	x_4	c_1	c_2	c_3
u_9	1	0	0	0	0	1	1
u_{10}	1	0	0	1	1	0	0
u_{11}	1	0	1	0	1	0	1
u_{12}	1	0	1	1	0	1	0
u_{13}	1	1	0	0	1	1	0
u_{14}	1	1	0	1	0	0	1
u_{15}	1	1	1	0	0	0	0
u_{16}	1	1	1	1	1	1	1

例題: (7,4)ハミング符号で誤り訂正

あ, い, う, え, お, か, き, く, け, こ,
さ, し, す, せ, そ, た



16個のデータ → 4ビットの符号

あ → 0,0,0,0 (u_1)

い → 0,0,0,1 (u_2)

う → 0,0,1,0 (u_3)

...

た → 1,1,1,1 (u_{16})

送信側での処理：データ送信してみよう

「う」を送信

符号割り当てより

$$x_1=0, x_2=0, x_3=1, x_4=0$$

検査ビット c_1, c_2, c_3 は

$$c_1 = x_2 \oplus x_3 \oplus x_4 = 0 \oplus 1 \oplus 0 = 1$$

$$c_2 = x_1 \oplus x_3 \oplus x_4 = 0 \oplus 1 \oplus 0 = 1$$

$$c_3 = x_1 \oplus x_2 \oplus x_4 = 0 \oplus 0 \oplus 0 = 0$$

検査ビットは
式から計算!!

送信される符号は0010110

受信側での処理

$$s_1 = x_4 \oplus c_1 \oplus c_2 \oplus c_3$$

$$s_2 = x_2 \oplus x_3 \oplus c_2 \oplus c_3$$

$$s_3 = x_1 \oplus x_3 \oplus c_1 \oplus c_3$$

誤りの位置	誤りパターン							シンドローム		
	x_1	x_2	x_3	x_4	c_1	c_2	c_3	s_1	s_2	s_3
誤りなし	0	0	0	0	0	0	0	0	0	0
左から1ビット目	1	0	0	0	0	0	0	0	0	1
左から2ビット目	0	1	0	0	0	0	0	0	1	0
左から3ビット目	0	0	1	0	0	0	0	0	1	1
左から4ビット目	0	0	0	1	0	0	0	1	0	0
左から5ビット目	0	0	0	0	1	0	0	1	0	1
左から6ビット目	0	0	0	0	0	1	0	1	1	0
左から7ビット目	0	0	0	0	0	0	1	1	1	1

(s_1, s_2, s_3): シンドローム. 誤りの位置情報を表す

受信処理の例

「0010010」を受信

$$s_1 = x_4 \oplus c_1 \oplus c_2 \oplus c_3 = 1$$

$$s_2 = x_2 \oplus x_3 \oplus c_2 \oplus c_3 = 0$$

$$s_3 = x_1 \oplus x_3 \oplus c_1 \oplus c_3 = 1$$

$s_1, s_2, s_3 = 1, 0, 1 \rightarrow$ 2進数の5 \rightarrow 5ビット目に誤り
 \rightarrow 5ビット目のデータを反転 (0 \rightarrow 1)

正しくは「0010**1**10」が送信された

例題

「あ」～「た」の16文字のひらがなデータを用いるとき、次の系列を受信した。このとき、送信された情報を求めよ。

ただし、 $(7,4)$ ハミング符号を用いるとする。

00011110111010101010011010