

9. 通信路符号化手法1 (誤り検出と誤り訂正の原理)

- 誤り検出と誤り訂正の原理
 - 符号空間とハイパーキューブ
 - ハミング距離
 - 情報ビットと検査ビット

準備：符号の空間的取扱い

ハイパーキューブ(2元符号の符号空間)

(定義)ハイパーキューブ

次のように再帰的に定義される構造をハイパーキューブという。 n 次元のハイパーキューブを $HC(n)$ と書く。

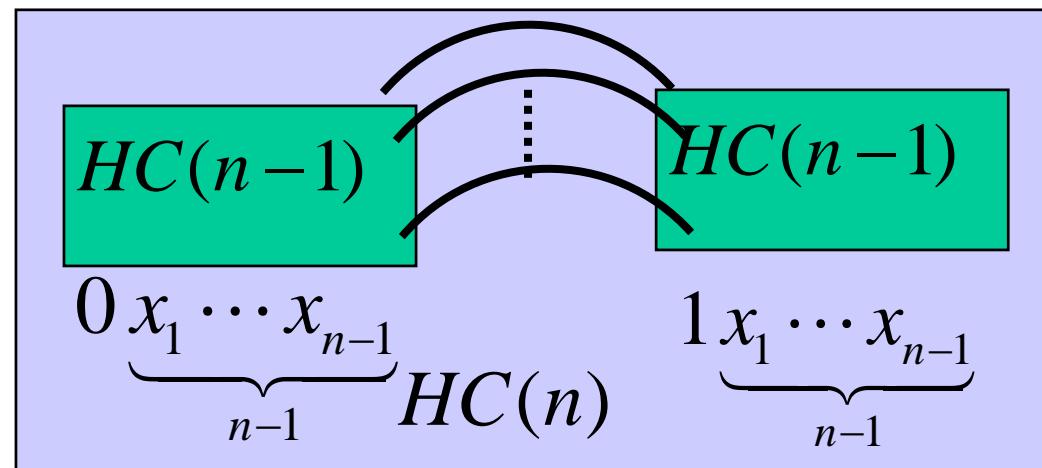
(1) 1次元のハイパーキューブ(基礎)

線分 $HC(1)$

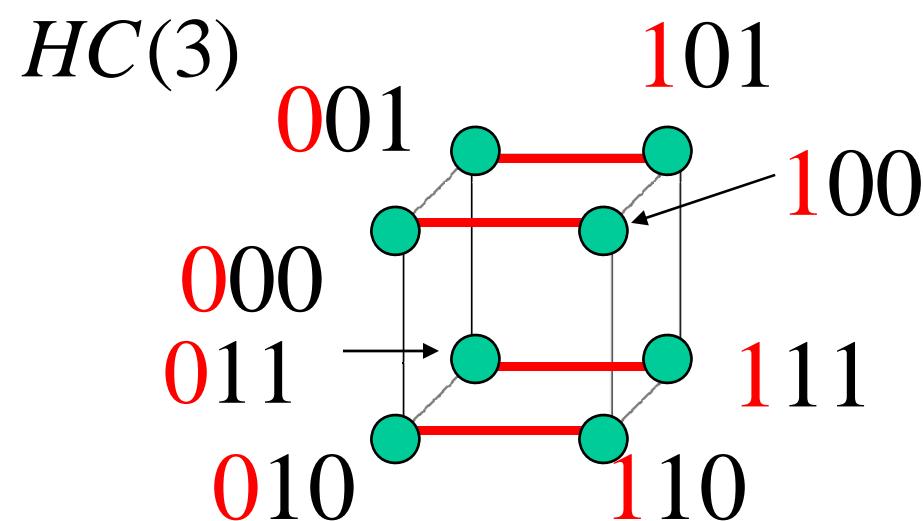
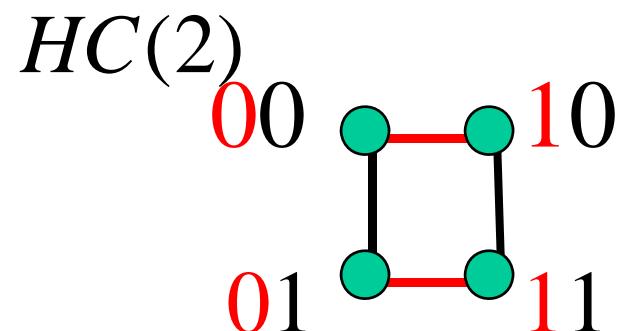
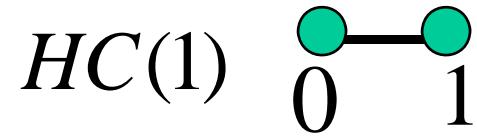


(2) 次元のハイパーキューブ(帰納)

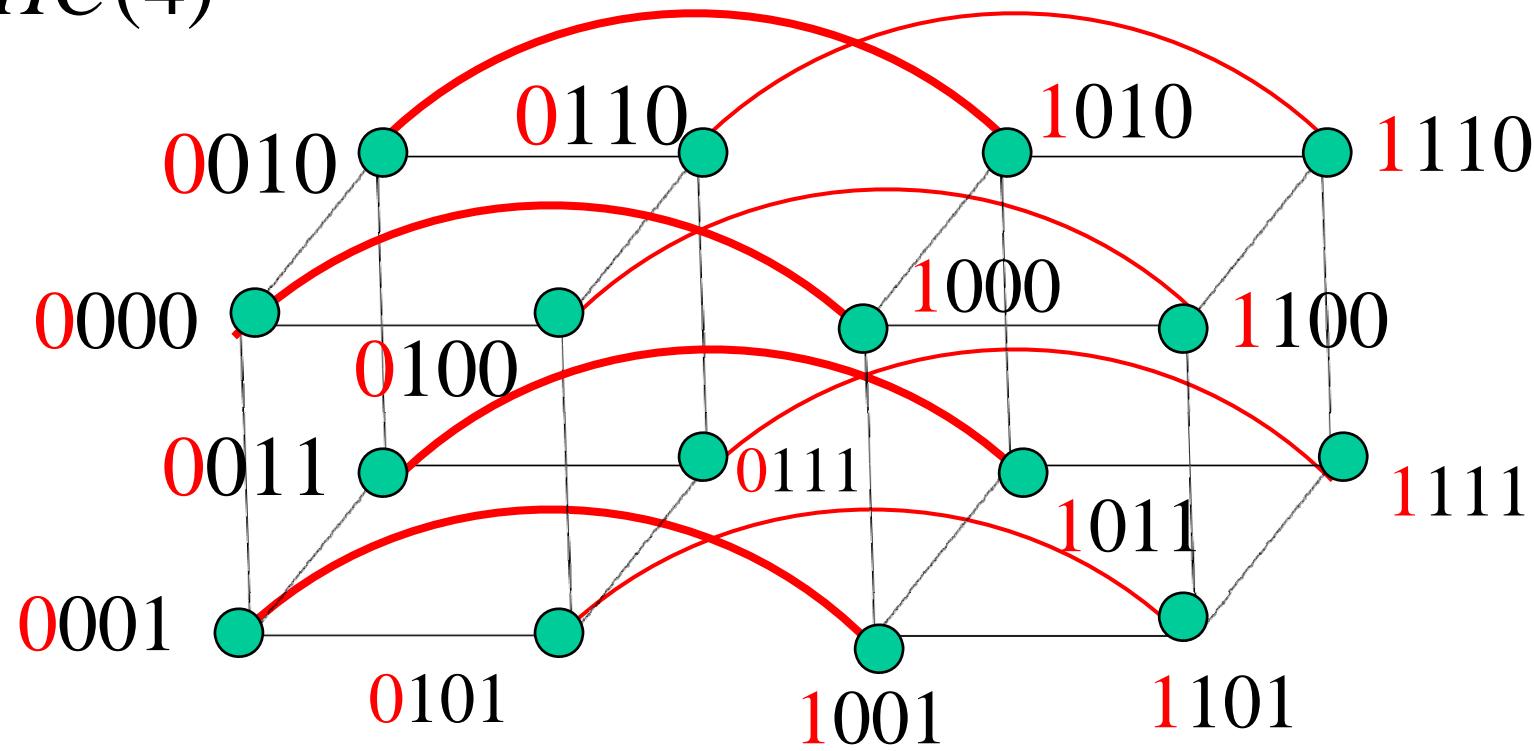
$HC(n-1)$ と $HC(n-1)$ を対応する頂点同士、辺で結び、最上位ビットだけが異なるように 0, 1 を割り振る。



例



$HC(4)$



練習

$HC(5)$ を図示せよ。

ハミング距離(符号語間の“近さ”の尺度)

(定義)ハミング距離

長さ n の2元符号(n 次元2値ベクトル(ブールベクトル))

$$x = (x_1, \dots, x_n) \in B^n$$

教科書では $h(x, y)$

$$y = (y_1, \dots, y_n) \in B^n$$

に対して、次式で定義される距離 $d_h(x, y)$ を**ハミング距離**(Hamming Distance)という。

$$d_h(x, y) = \sum_{i=1}^n \delta(x_i, y_i)$$

n ビット中、異なるビットの総数

ただし、

$$\delta(x_i, y_i) = \begin{cases} 0 & x_i = y_i \\ 1 & x_i \neq y_i \end{cases}$$

ビットが異なるとき
に1で等しいとき0。

ハミング距離は次のようにも書ける。

$$d_h(x, y) = \sum_{i=1}^n |x_i - y_i|$$

$$d_h(x, y) = \sum_{i=1}^n (x_i - y_i)^2$$

例

次の2つの系列のハミング距離を求めよ。

$$x = 100101 \in B^6$$

$$y = 101110 \in B^6$$

解

$$\begin{array}{r} 100101 \\ \downarrow \quad \downarrow \quad \downarrow \\ 101110 \end{array}$$

ビットが異なる位置
の総数。

$$d_h(x, y) = 3$$

練習

次のハミング距離を求めよ。

(1)

$$d_h(011,110)$$

(2)

$$d_h(1100,0110)$$

(3)

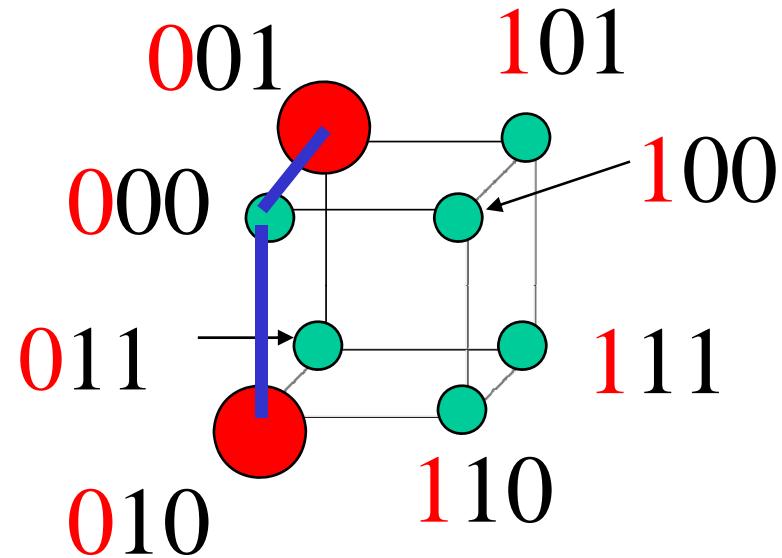
$$d_h(011001,101010)$$

(4)

$$d_h(00011011,10010110)$$

ハミング距離とハイパーキューブ

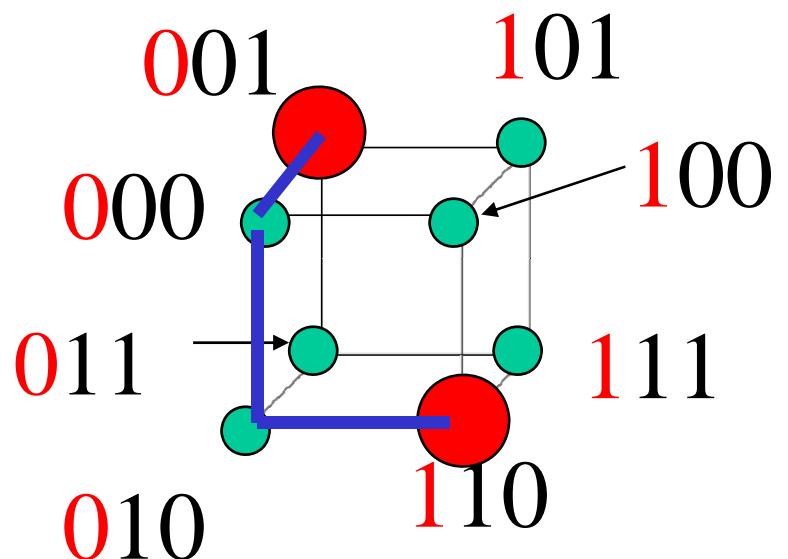
$HC(3)$



$$d_h(001, 010) = 2$$

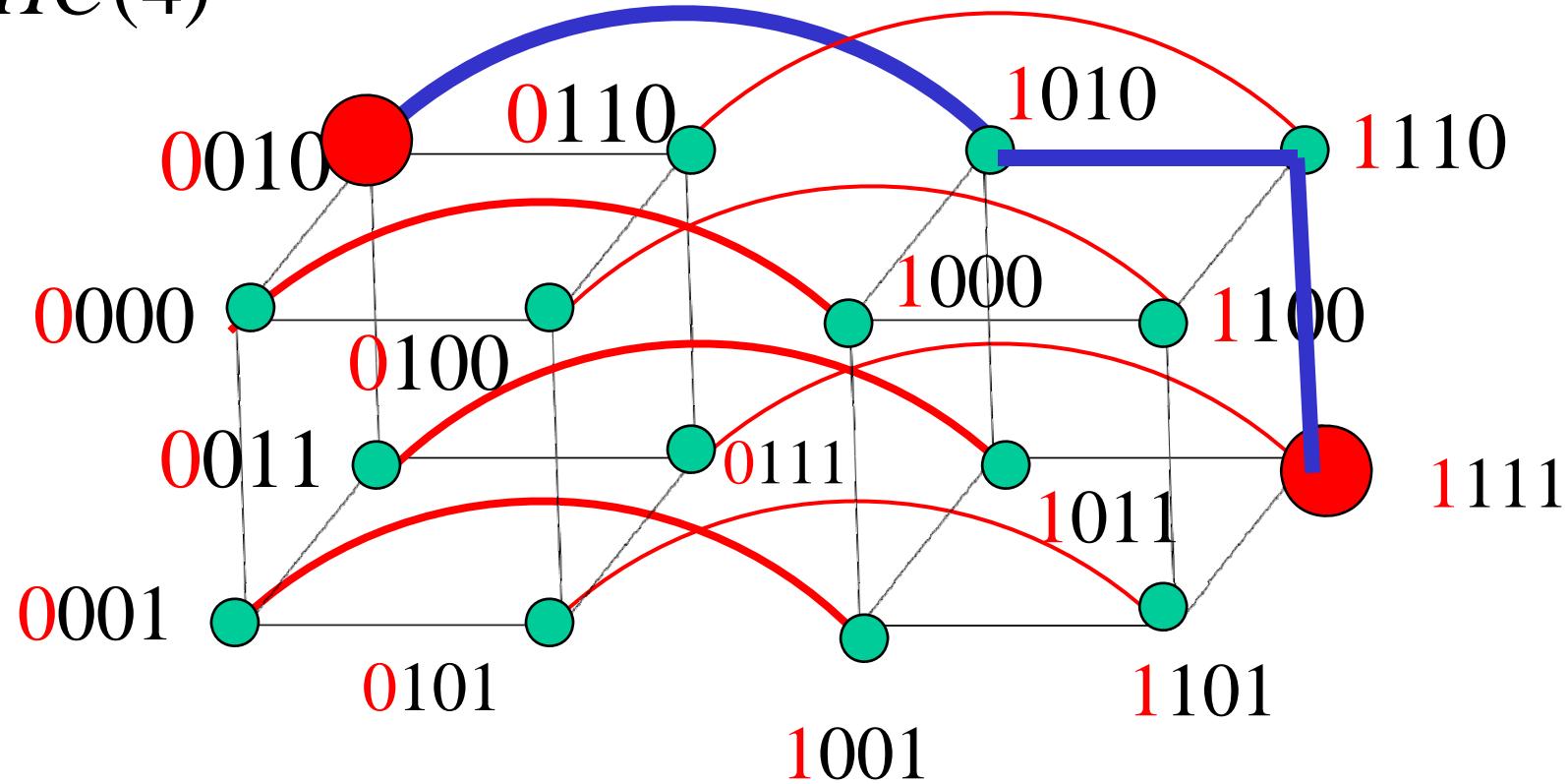
ハイパーキューブ上での
経路長がハミング距離

$HC(3)$



$$d_h(001, 110) = 3$$

$HC(4)$



$$d_h(0010, 1111) = 3$$

練習

次の2つの系列

$$x = 11001 \in B^5, y = 10110 \in B^5$$

をハイパーキューブ $HC(5)$ 上に図示し、
ハミング距離 $d_h(x, y)$ を求めよ。

参考：距離の3公理

(公理) 距離の公理

任意の n 次元ベクトル $x, y, z \in K^n$, ($K = B$ や \mathbb{R} や \mathbb{C}) に対して、次の3つの性質を満たす関係 $d : K^n \times K^n \rightarrow \mathbb{R}$ を **距離** という。

(1) $d(x, x) = 0$ 逆に、 $d(x, y) = 0$ なら $x = y$

同一地点間の距離は0

(2) $d(x, y) = d(y, x)$

距離は対称的

(3) $d(x, y) + d(y, z) \geq d(x, z)$

3角不等式

ユークリッド距離だけが距離なのではない。様々な距離の概念がある。

様々な距離1(ユークリッド距離)

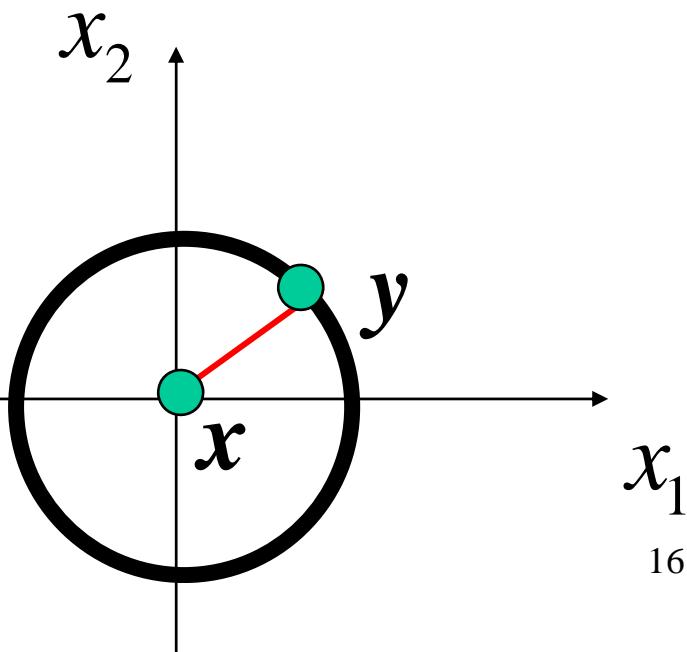
$$x = (x_1, \dots, x_n), y = (y_1, \dots, y_n) \in \mathbb{R}^n$$

$$d_2(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

最もよく用いられる
距離

$$= \left\{ \sum_{i=1}^n |x_i - y_i|^2 \right\}^{\frac{1}{2}}$$

2次元でのユークリッド距離での、等距離線は円になる。 $n = 2$



様々な距離2(マンハッタン距離)

$$x = (x_1, \dots, x_n), y = (y_1, \dots, y_n) \in \mathbb{R}^n$$

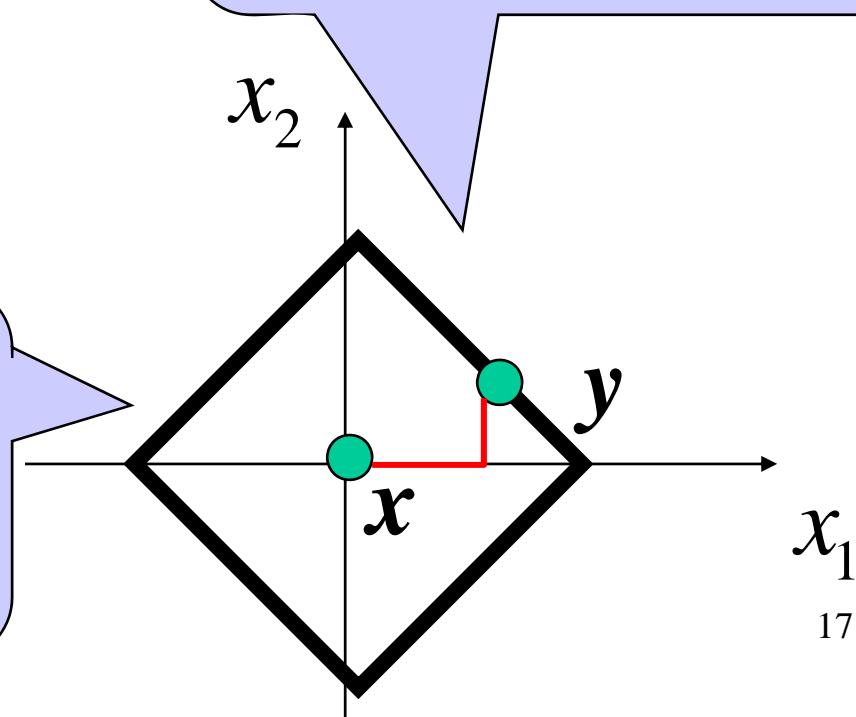
$$d_1(x, y) = \sum_{i=1}^n |x_i - y_i|$$

$$= \left\{ \sum_{i=1}^n |x_i - y_i|^1 \right\}^{\frac{1}{1}}$$

2次元でのマンハッタン距離での、等距離線は菱形（回転した正方形）になる。

$$n = 2$$

水平あるいは垂直な方向にしか進めないと仮定したときの最短経路長。



様々な距離3(ハミング距離)

$$x = (x_1, \dots, x_n), y = (y_1, \dots, y_n) \in \{0,1\}^n$$

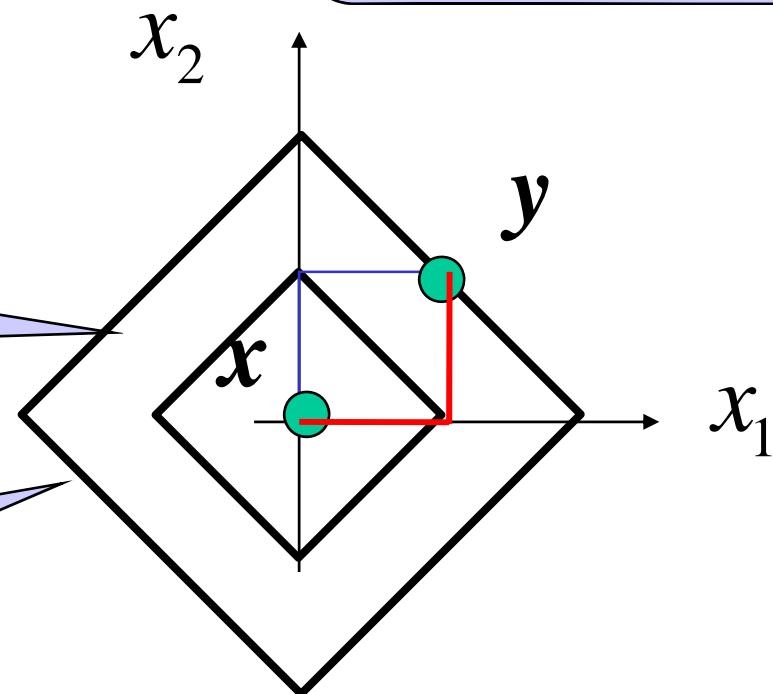
$$d_h(x, y) = \sum_{i=1}^n (x_i - y_i)^2$$

$$= \sum_{i=1}^n |x_i - y_i|$$

0,1しか認めない世界でのマンハッタン距離

半径1の“円”

半径2の“円”



様々な距離3(L_∞ 距離)

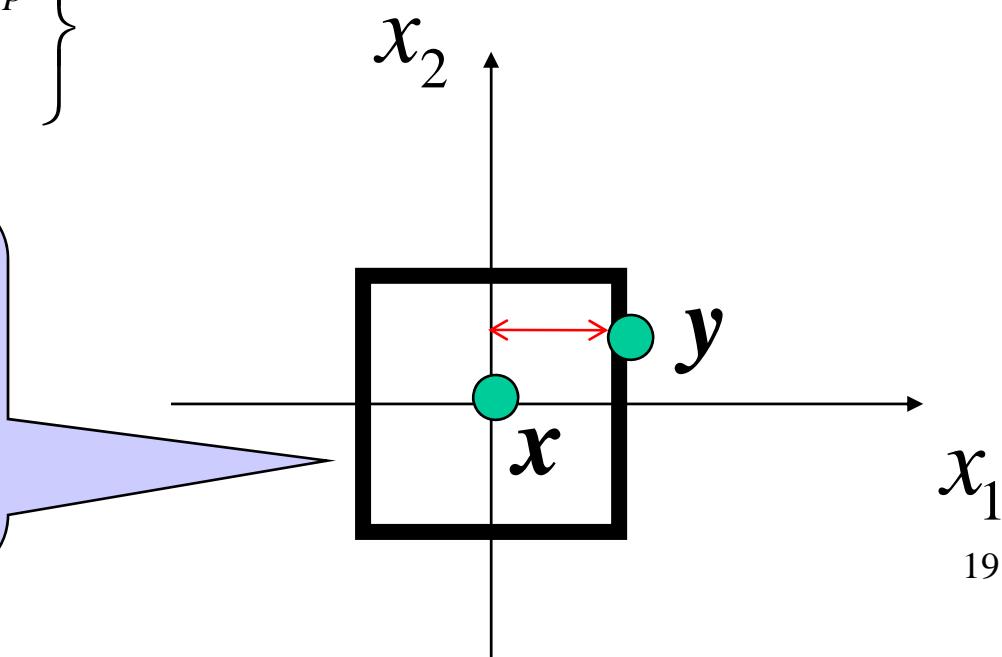
$$x = (x_1, \dots, x_n), y = (y_1, \dots, y_n) \in \mathbb{R}^n$$

$$d_\infty(x, y) = \max_{i=1}^n \{|x_i - y_i|\}$$

ある軸での、
座標の差の最大値

$$= \lim_{p \rightarrow \infty} \left\{ \sum_{i=1}^n |x_i - y_i|^p \right\}^{\frac{1}{p}}$$

2次元での等
距離線は正方
形になる。
 $n = 2$



様々な距離4(L_p 距離)

$$x = (x_1, \dots, x_n), y = (y_1, \dots, y_n) \in \mathbb{R}^n$$

$$d_p(x, y) = \left\{ \sum_{i=1}^n |x_i - y_i|^p \right\}^{\frac{1}{p}}$$

マンハッタン距離は
別名 L_1 距離と
もいう。

ユークリッド距離は
別名 L_2 距離と
もいう。

練習

$A = (0, 0), B = (1, 2), C = (3, 2)$ としたとき、次の距離を求めよ。

(1)

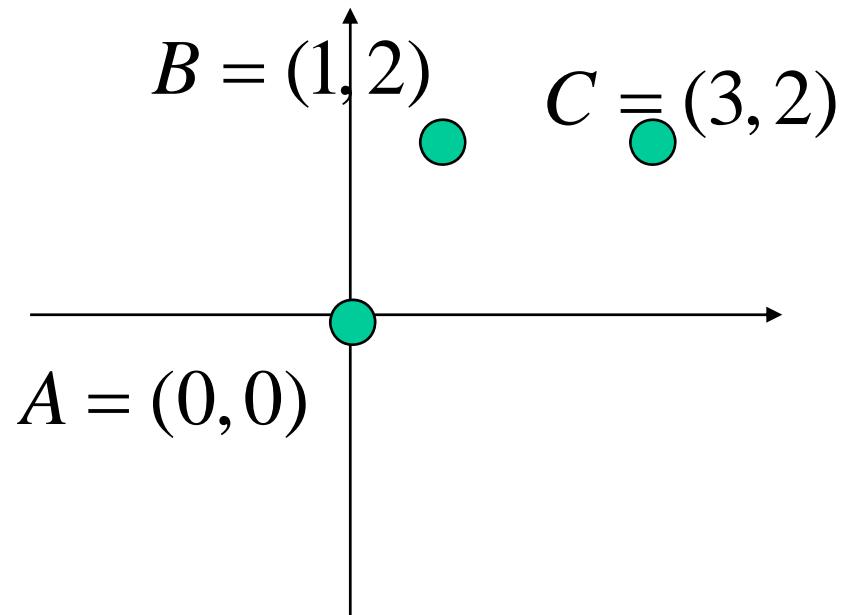
$$d_2(A, B), d_2(B, C), d_2(A, C)$$

(2)

$$d_1(A, B), d_1(B, C), d_1(A, C)$$

(3)

$$d_\infty(A, B), d_\infty(B, C), d_\infty(A, C)$$



誤り検出と誤り訂正の原理

排他的論理和

(定義) 排他的論理和 —————

次の真理値で定められる論理関数を**排他的論理和**という。

論理変数 x と論理変数 y の排他的論理和を $x \oplus y$ と書く。

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

排他的論理和は2進数の加算、減算に対応する。

2を法とする演算

$X \pmod{2}$ で X を2で割った余りを意味する。(この計算を2を法とする計算という。)

$x + y \pmod{2}$ を考える。これは、以下のように計算できる。

$x \pmod{2}$	$y \pmod{2}$	$x + y$	$x + y \pmod{2}$
0	0	0	0
0	1	1	1
1	0	1	1
1	1	2	0

排他的論理和と同一

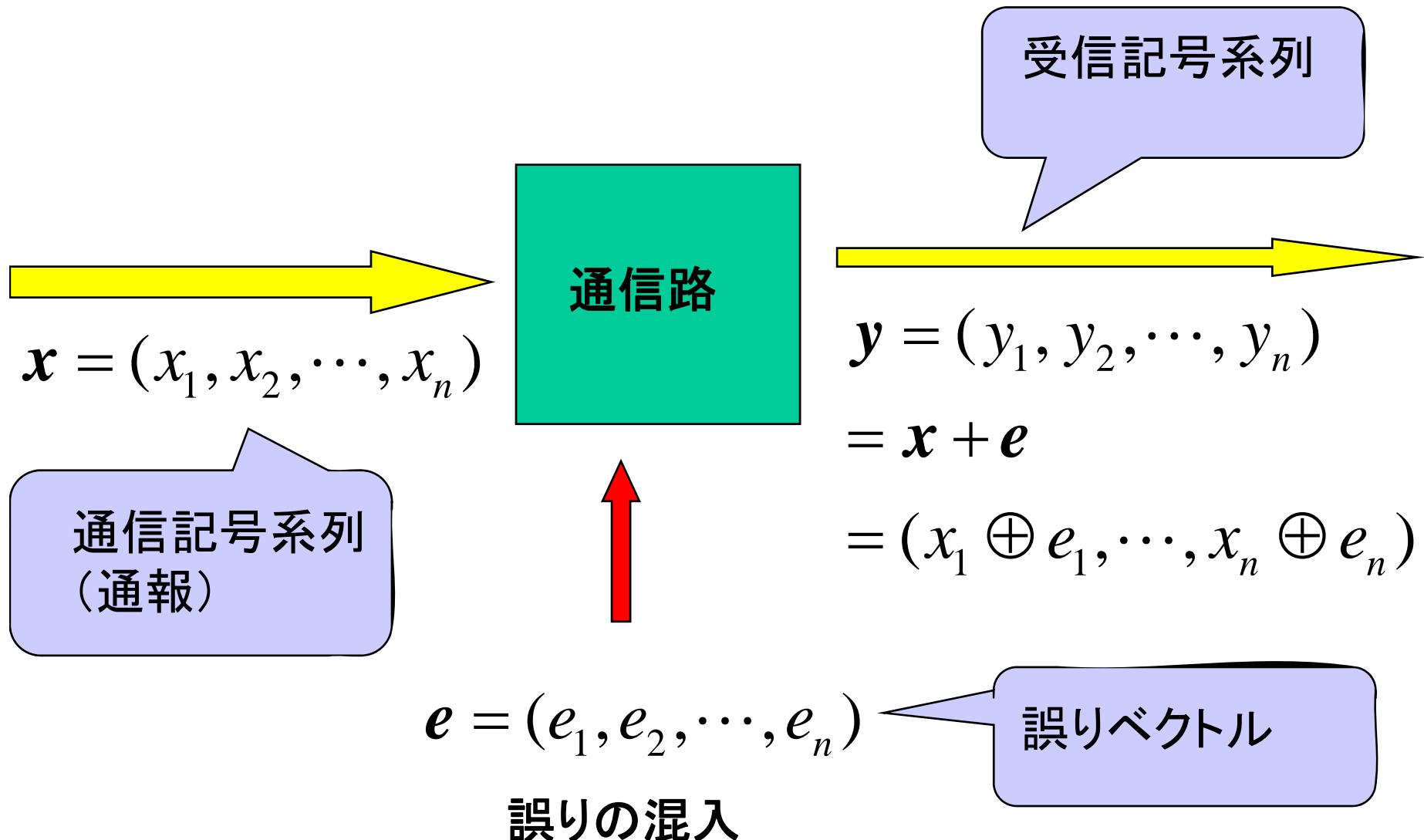
$x - y \pmod{2}$ を考える。これは、以下のように計算できる。

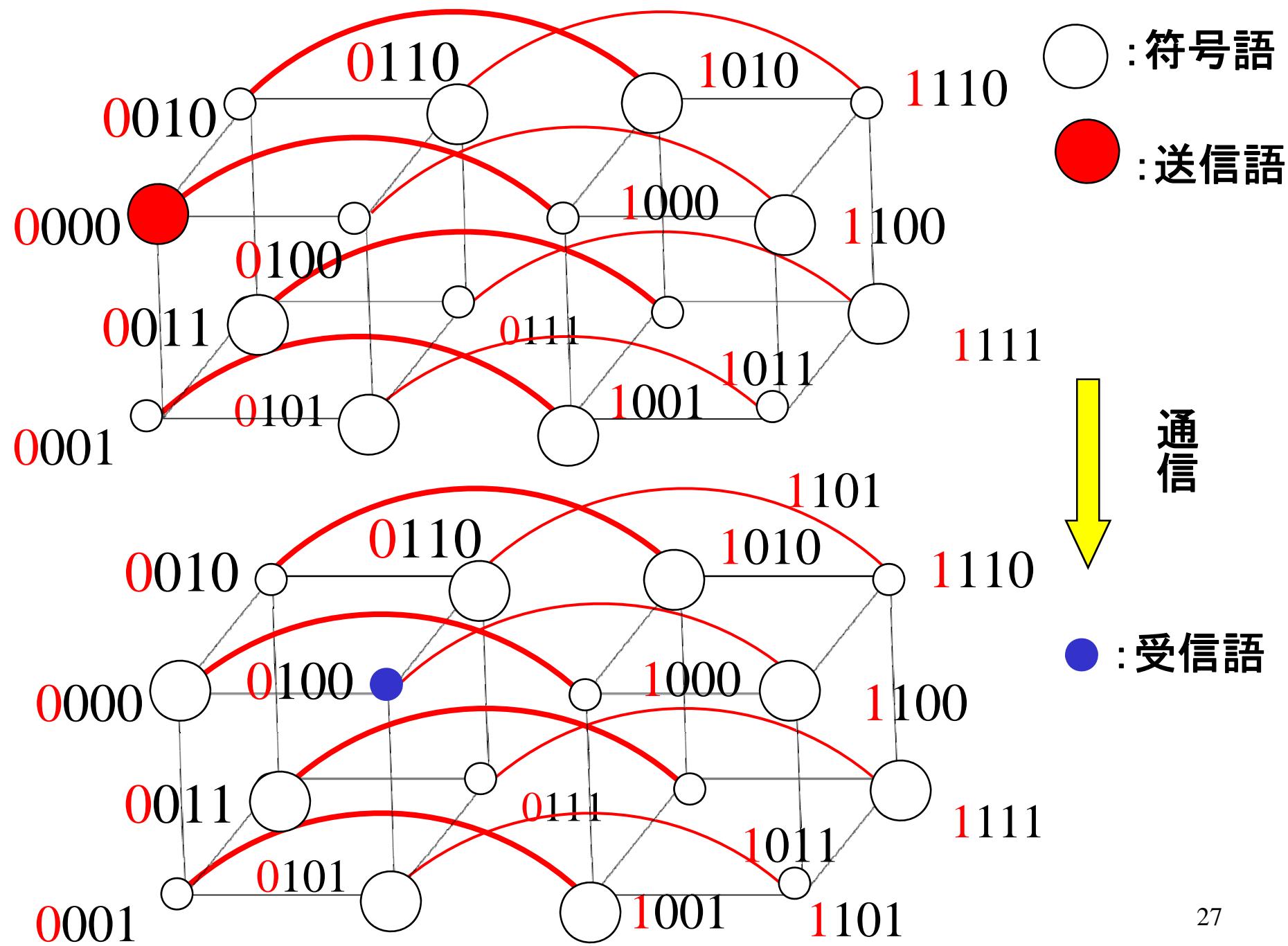
$x \pmod{2}$	$y \pmod{2}$	$x - y$	$x - y \pmod{2}$
0	0	0	0
0	1	-1	1
1	0	1	1
1	1	0	0

排他的論理和と同一。、
2を法とする加算とも同一。

$$-1 \pmod{2} = 1$$

通信路モデルにおける誤りの混入





前のスライドにおいては、
通信路符号は

$$W = \{0000, 0011, 0101, 0110, 1001, 1010, 1100, 1111\}$$

である。

このとき符号語 $w_i = 0000 \in W$ を通信し、

誤り $e_i = 0100$ が混入されると、

$y_i = 0100 \notin W$ が受信される。

一方、誤りに制限を加えなければ、誤りの検出、訂正はできない。

$$0100 =$$

$$y_i = w_i + e_i$$

$$= 0000 + 0100 ?$$

$$= 0011 + 0111 ?$$

$$= 0110 + 0010 ?$$

誤り空間と複号領域

(定義)誤り空間と複号領域

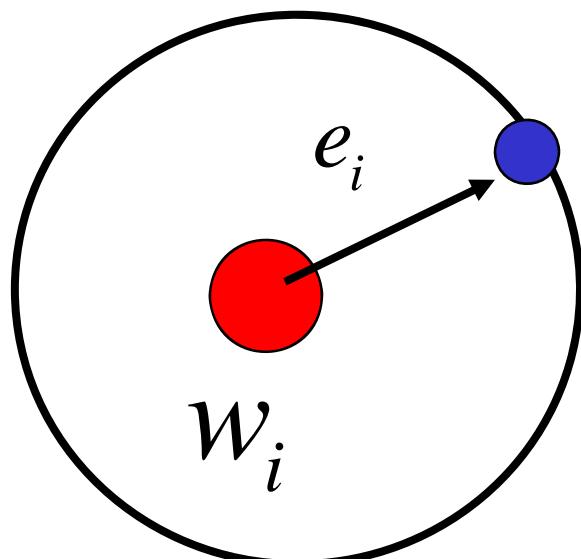
混入される可能性のある誤りベクトルの集合を**誤り空間**といい、

$$E_i = \{e_i \mid w_i \text{に混入される誤りベクトル}\}$$

で表わす。また、次式で表わされる空間を**複号領域**という。

$$\Omega_i = \{y_i \in B^n \mid y_i = w_i + e_i, e_i \in E_i\}$$

(mod 2) は省略



$$\begin{aligned} d_h(w_i, y_i) &= d_h(w_i, w_i + e_i) \\ &= d_h(w_i - w_i, e_i) \\ &= d_h(0, e_i) \end{aligned}$$

0ベクトルからのハミング距離を考えると便利なことが多い。

誤り空間と複号領域例

符号 $W = \{w_1, w_2\} = \{1010, 0101\}$ を通信する際に、1ビットまで誤りが起こるとする。すなわち、

$$d_h(0, e) \leq 1$$

とする。このとき、誤り空間 E および、各符号語の複号領域 Ω_i を求めよ。

解答例)

誤り空間: $E = \{0000, 0001, 0010, 0100, 1000\}$
符号語 $w_1 = 1010$ の複号領域:

$$\Omega_1 = \{1010, 1011, 1000, 1110, 0010\}$$

符号語 $w_2 = 0101$ の複号領域:

$$\Omega_2 = \{0101, 0100, 0111, 0001, 1101\}$$

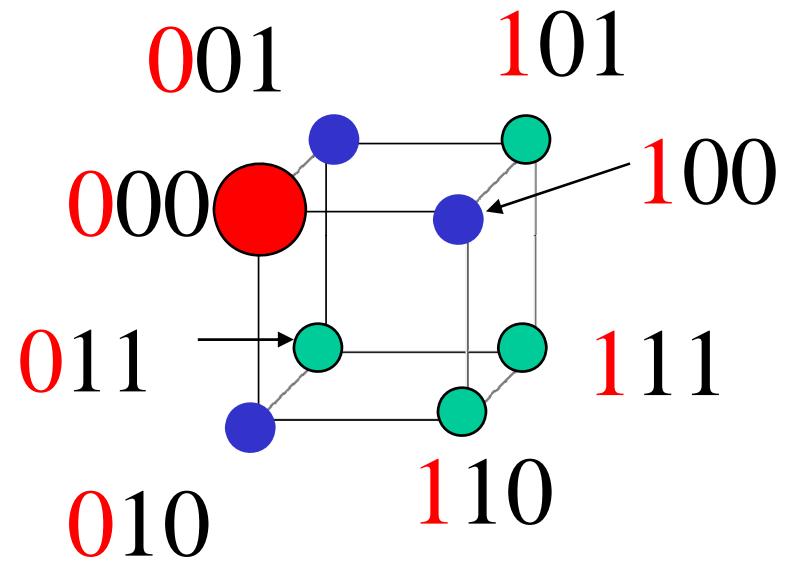
練習

符号 $W = \{w_1, w_2\} = \{0000, 1111\}$ を通信する際に、1ビットまで誤りが起こるとする。すなわち、

$$d_h(0, e) \leq 2$$

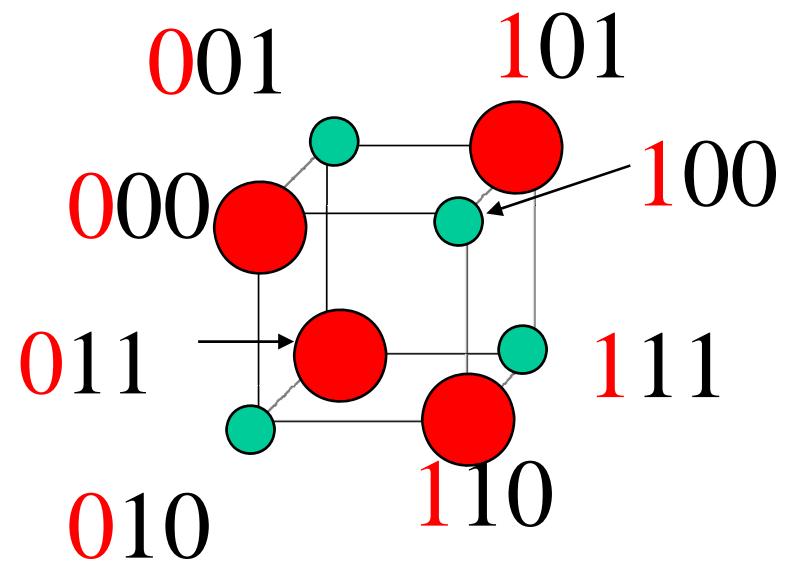
とする。このとき、誤り空間 E および、各符号語の複号領域 Ω_i を求めよ。

複号領域と誤り検出



$$w_1 = 000$$

$$\Omega_1 = \{000, 001, 010, 100\}$$



$$W = \{000, 011, 101, 110\}$$

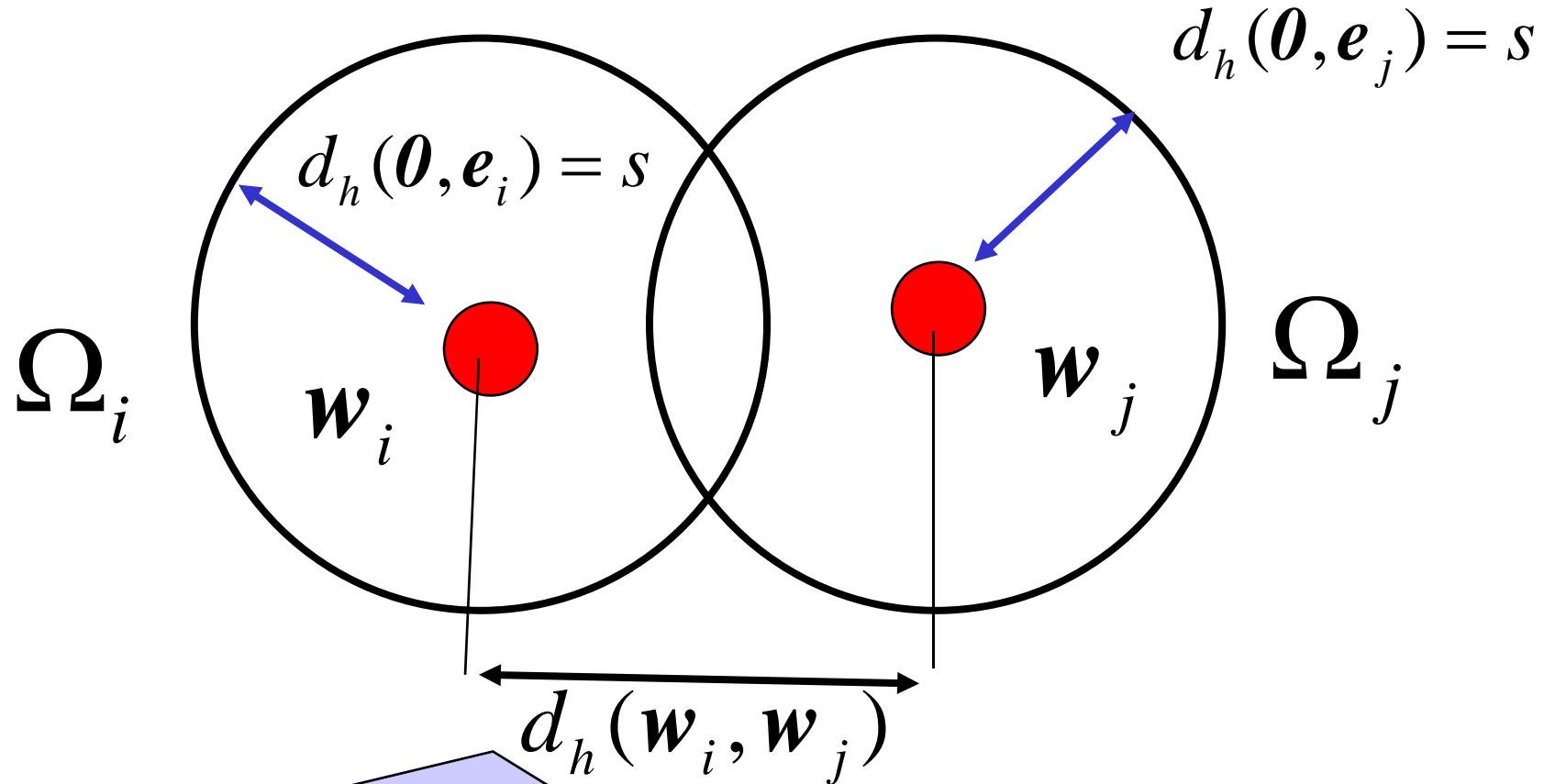
$$w_2 = 011 \notin \Omega_1, w_3 = 101 \notin \Omega_1,$$

$$w_4 = 110 \notin \Omega_1$$

$$\forall i, j, i \neq j \quad w_j \notin \Omega_i$$

誤り検出を可能とする関係式。

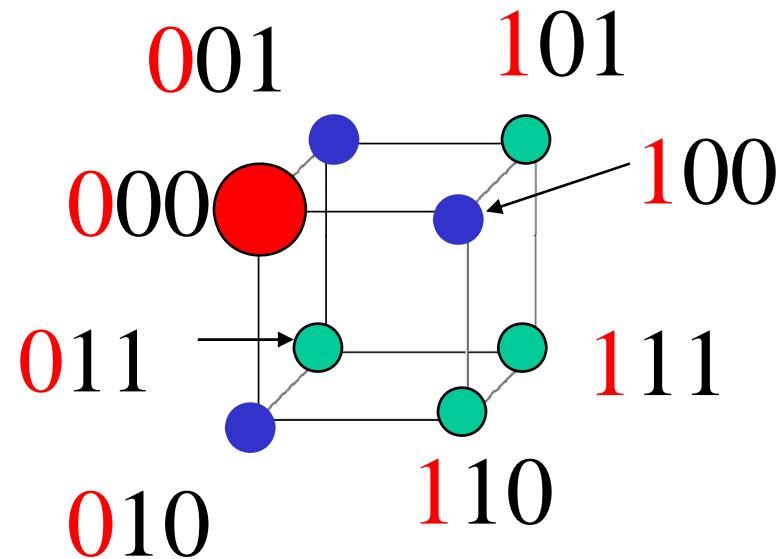
ハミング距離と誤り検出



s ビット誤りを検出するためには次式が成り立つ
必要がある。

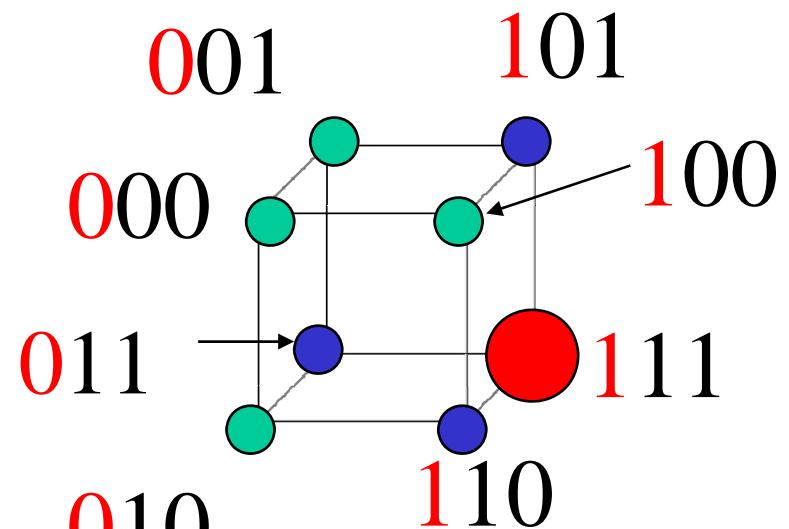
$$\forall i, j, i \neq j \quad d_h(w_i, w_j) \geq s + 1$$

複号領域と誤り訂正



$$w_1 = 000$$

$$\Omega_1 = \{000, 001, 010, 100\}$$



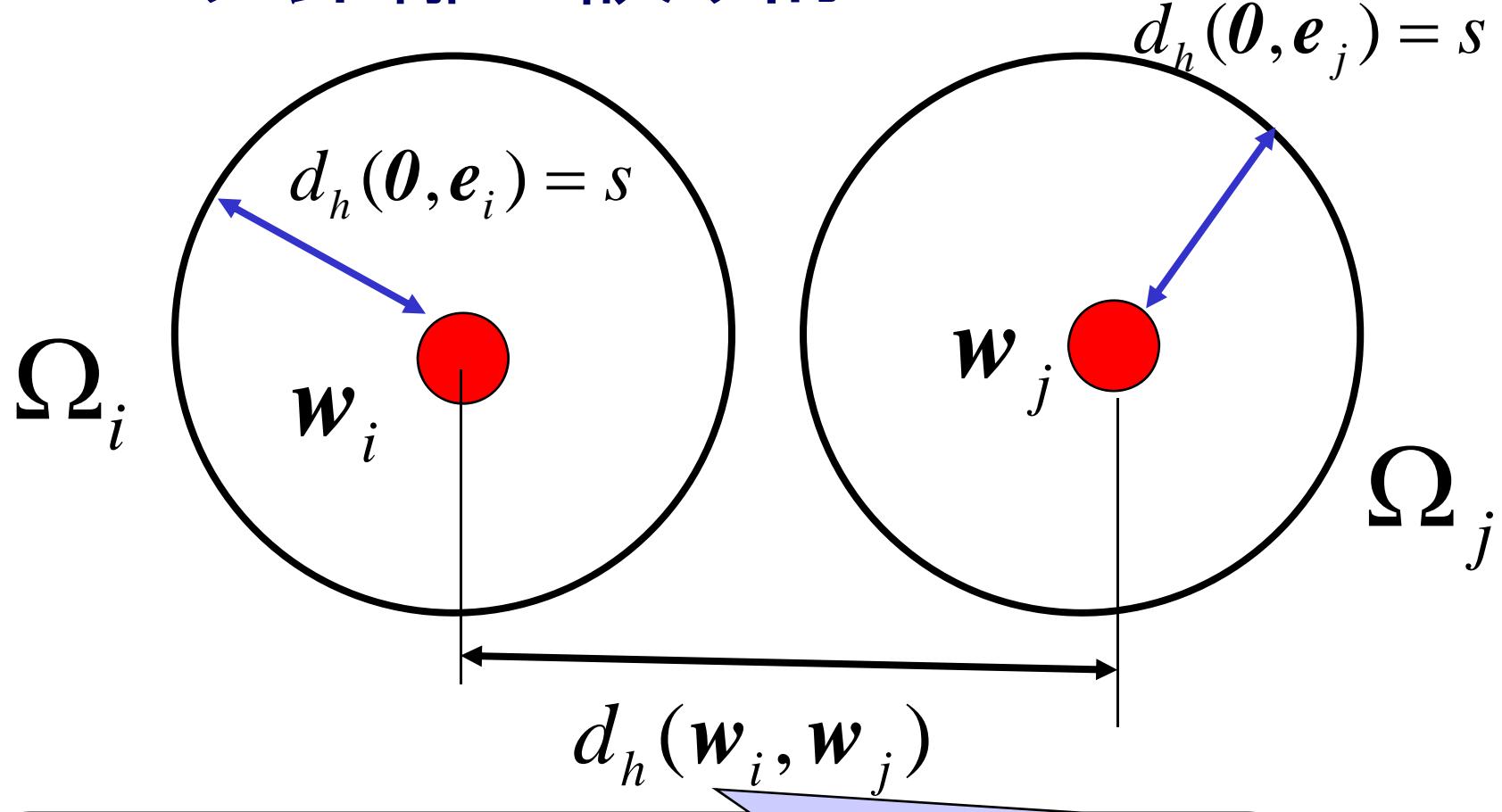
$$w_2 = 111$$

$$\Omega_2 = \{111, 110, 101, 011\}$$

$$\forall i, j, i \neq j \quad \Omega_i \cap \Omega_j = \emptyset$$

誤り訂正を可能とする関係式。

ハミング距離と誤り訂正



s ビット誤りを訂正するためには次式が成り立つ
必要がある。

$$\forall i, j, i \neq j \quad d_h(w_i, w_j) \geq 2s + 1$$