http://goo.gl/4jF0Kl

KYOTO UNIVERSITY

情報理論
誤り訂正符号①

鹿島久嗣
京都大学 情報学研究科 知能情報学専攻

DEPARTMENT OF INTELLIGENCE SCIENCE AND TECHNOLOGY

後半の予定：
「誤り訂正符号」と「アナログ情報源」について学習

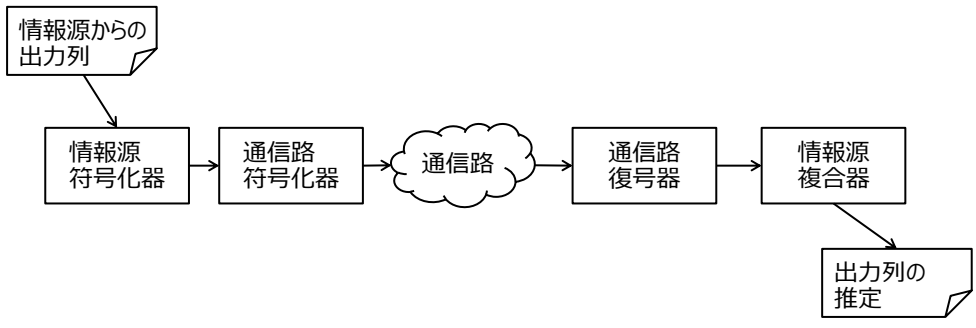
- 第9回　6月16日　誤り訂正符号①
- 第10回　6月23日　誤り訂正符号②
- 第11回　6月30日　誤り訂正符号③
- 第12回　7月 7日　アナログ情報源 ①
- 第13回　7月14日　アナログ情報源②
- 第14回　7月**17**日（振替）　到達度確認テスト

- 講義Webページ（資料）：http://goo.gl/PXyycQ

通信路おさらい

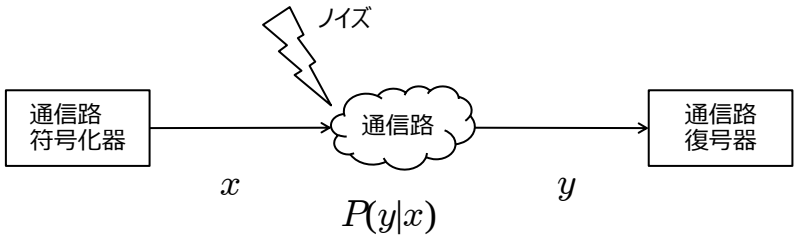
情報通信のモデル：情報源符号化と通信路符号化

- 講義前半では情報源の符号化を扱ってきた
- 後半は通信路の符号化を扱う



通信路のモデル：
通信によって送った信号にノイズが載る

- 通信路：
  - 記号 x を送信すると、記号 y が届く （雑音によって変化する）
  - 通信路の振舞を条件付き確率$P(y|x)$で表す
    - 典型例： 2元対称通信路



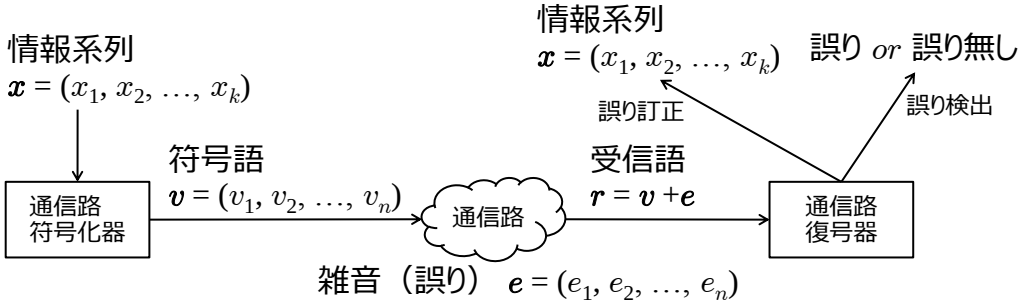
通信路符号化定理：
情報源符号化定理と並ぶ重要定理

- 通信路容量： $C = \max_{P(X)} I(X; Y)$
  - 入力と出力の相互情報量の（入力分布について）最大値
  - 通信路の性能限界
- 通信路符号化定理：
  （伝送速度が通信路容量よりも小さければ）
  符号長を大きくすることで複合誤り確率をいくらでも小さくできる
  - 伝送速度： $1/n \cdot \log |C|$ （ C は符号語の数）
  - 符号誤り率を下げるために、伝送速度を下げる必要は無い

誤り訂正符号

誤り訂正符号：
通信路で起こるあやまりを検出・修正する仕組み

- 具体的にどのように通信路符号を構成するか？
- 通信路を通ることで、符号語 $\boldsymbol{v}$ に雑音 $\boldsymbol{e}$ が加わる
- 誤り検出： $\boldsymbol{e} = \boldsymbol{0}$ かどうかを判定
- 誤り訂正： $\boldsymbol{e} \neq \boldsymbol{0}$ のときに $\boldsymbol{x}$ を復元 （ $\boldsymbol{e}$ を同定する）



（参考）2元体の計算：
以下では2元体上での計算を前提にする

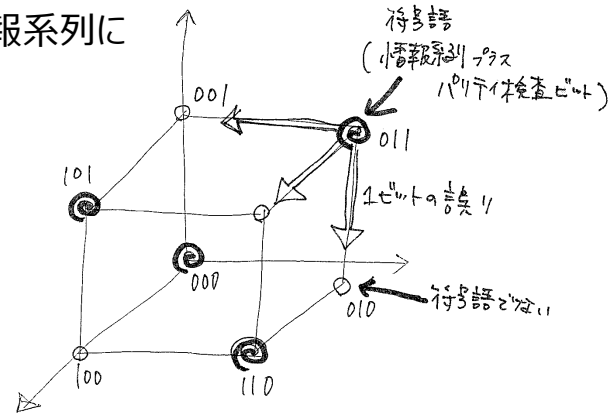
- アルファベットは{0, 1}
- 演算は mod 2演算とする ：
  - 加：0+0 = 1+1 = 0, 0+1 = 1+0 = 1
  - 減：0-0 = 1-1 = 0, 1-0 = 0-1 = 1
  - 乗：0×0 = 1×0 = 0×1 = 0, 1×1 = 1
  - 除：0÷1 = 0, 1÷1 = 1
  - 複数桁の場合、ビットごとの演算 （例： 110+010 = 100 ）

単一パリティ検査符号：
1ビットの誤りを検出する簡単な誤り検出符号

- 基本的なアイディア：
  - 情報系列中の1の数が偶数になるように、最後に1ビット付け加える
    - $(x_1, x_2, \ldots, x_k)$
      $\rightarrow (v_1=x_1, v_1=x_2, \ldots, v_k=x_k, v_{k+1}=\Sigma_{i=1,\ldots,k}\, x_i)$
    - 例： (1, 0, 0) → (1, 0, 0, 1)、(1, 1, 0) → (1, 1, 0, 0)
  - 誤りが（どこか1ビットで）起こると 1の数が奇数になる
    → 誤り検出

単一パリティ検査符号の誤り検出イメージ：格子状で使用されない頂点に移動することで検出される

- (情報系列, パリティ検査ビット) は3次元の格子上に配置される
- 1ビットの誤りは隣の頂点に移動すること
- 2ビット誤ると別の情報系列に



水平垂直パリティ検査符号：1ビットの誤りを訂正可能

- 情報系列とパリティ検査ビットを2次元に配置する
- パリティ検査ビット：
  $v_5=x_1+x_2$, $v_6=x_3+x_4$, $v_7=x_1+x_3$, $v_8=x_2+x_4$,
- パリティ検査ビットの検査ビット：
  $v_9=v_5+v_6=v_7+v_8$
- 1ビットの誤りを訂正可能
- 2ビットの誤りを検出可能

x_1	x_2	v_5
x_3	x_4	v_6
v_7	v_8	v_9

線形符号

線形符号：
検査記号が情報記号の線形式で与えられる扱い易い符号

- (n, k) 符号：k 個の情報ビットから n-k 個の検査ビットを計算して付加し、n ビットの符号語をつくる
  - 単一パリティ検査符号：k 個の情報ビットから1個の検査ビットを計算して付加し、k+1 ビットの符号語に → $(k+1, k)$ 符号
- 線形符号：検査ビットが情報記号の線形式で与えられる符号
  - 符号語の線形和もまた符号語になっているようなもの
  - 確認：パリティ検査符号

パリティ検査方程式とシンドローム：誤りの訂正と検出のための式

- 通信路では、符号語 $\boldsymbol{v} = (v_1, v_2, \ldots, v_n)$ に対して、誤りパタン $\boldsymbol{e} = (e_1, e_2, \ldots, e_n)$ が加わり、受信語 $\boldsymbol{r} = \boldsymbol{v} + \boldsymbol{e} = (v_1+e_1, v_2+e_2, \ldots, v_n+e_n)$ として観測される
- パリティ検査方程式：線形符号の必要十分条件を与える式 f
  - $\boldsymbol{v}$ が符号語なら $f(\boldsymbol{v}) = 0$、そうでないなら $\neq 0$
  - 単一パリティ検査符号では $f(\boldsymbol{v}) = x_1 + x_2 + \ldots + x_k + x_{k+1} = 0$ がみたされている
- シンドローム： $f(\boldsymbol{r}) = f(\boldsymbol{v}+\boldsymbol{e}) = f(\boldsymbol{v}) + f(\boldsymbol{e}) = f(\boldsymbol{e})$
  - 検査方程式の式 f に受信語を代入したもの
  - 誤りベクトルにのみ依存しており、誤りの検出と修正に利用する

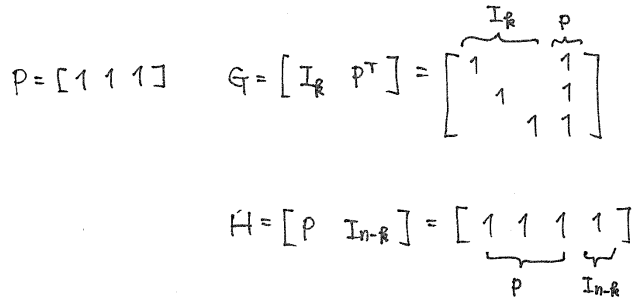
生成行列とパリティ検査行列：行列を用いた表現の検出と修正のための式

- 生成行列 $\boldsymbol{G}$：符号化の行列表現 $\boldsymbol{v} = \boldsymbol{x}\boldsymbol{G}$
- パリティ検査行列 $\boldsymbol{H}$：パリティ検査方程式の行列表現 $f(\boldsymbol{r}) = \boldsymbol{r}\boldsymbol{H}^{\top} = (\boldsymbol{v}+\boldsymbol{e})\,\boldsymbol{H}^{\top} = \boldsymbol{v}\boldsymbol{H}^{\top} + \boldsymbol{e}\boldsymbol{H}^{\top}$
- 情報系列が k=3 ビットの時の単一パリティ検査符号の例：

$$x = (x_1, x_2, x_3) \qquad G = \begin{bmatrix} 1 & & 1 \\ & 1 & 1 \\ & & 1 \end{bmatrix} \qquad H = [1\ 1\ 1\ 1]$$

生成行列とパリティ検査行列の関係：パリティ検査行列から生成行列を導ける

- ある行列 P に関して次の関係：
  生成行列　$G = [I_k \; P^\top]$, パリティ検査行列 $H = [P \;\; I_{n-k}]$
- 任意の k 次元情報ベクトル u にたいして $(uG)H^\top = \mathbf{0}$
- シンドローム計算 $(uG + e)H^\top = eH^\top$



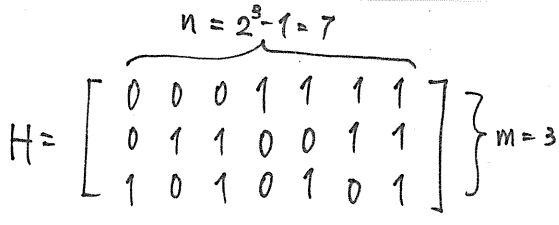
ハミング符号

基本的な考え方：
1ビット誤りを訂正できるパリティ検査行列のデザイン

- パリティ検査行列 $\boldsymbol{H} = [\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_n]$ があるとする
- 誤り位置が i であるような誤りベクトル $\boldsymbol{e}_i = (0, \ldots, 0, 1, 0, \ldots, 0)$ に対し、シンドローム $\boldsymbol{e}_i\boldsymbol{H}^\top = \boldsymbol{h}_i^\top$ となる
- すべての i について $\boldsymbol{h}_i \neq \boldsymbol{0}$ かつ、
  すべての $i \neq j$ に対して $\boldsymbol{h}_i \neq \boldsymbol{h}_j$ であれば、
  シンドロームを見れば誤り位置が特定できる

ハミング符号：
1ビット誤りを訂正できる

- 誤り検出のためには n ビットの符号語に対して、
  列数 n のパリティ検査行列 $\boldsymbol{H}$ が必要
- m ビットベクトルからは 2^m-1 個の異なる非ゼロベクトルが作れる
  - 2^m-1 $\geqq$ n → Hの行数を $m = \lceil \log n+1 \rceil$ でとれば十分
- $m = 3$ のとき：

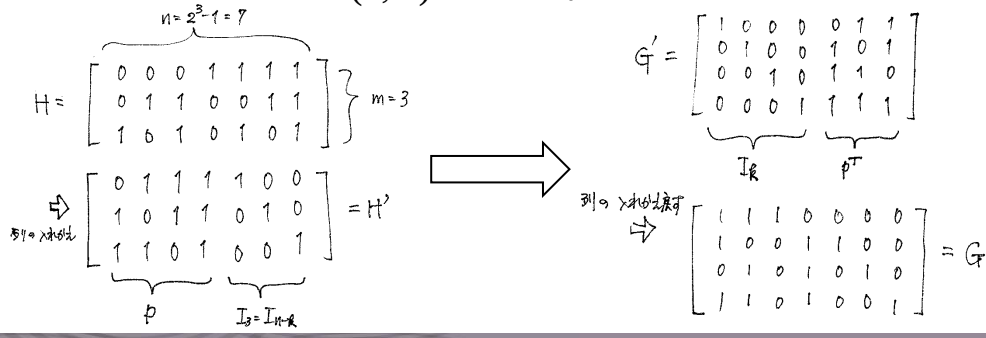


- たとえば $\boldsymbol{e}_i = (0, 0, 1, 0, 0, 0, 0)$ に対して $\boldsymbol{e}_i\boldsymbol{H}^\top = \boldsymbol{h}_i^\top = (0, 1, 1)$

ハミング符号の生成行列：検査行列から導ける

- 生成行列と検査行列の関係より、生成行列を導ける
  - 生成行列　$G = [I_k \; P^{\top}]$
  - パリティ検査行列 $H = [P \;\; I_{n-k}]$

(7, 4)- ハミング符号



一般のハミング符号：1ビット誤りを訂正できる

- 前述のハミング符号は(7, 4)- ハミング符号
- 一般の場合　(m)
  - 符号長 $n = 2^m - 1$
  - 情報ビット数 $k = 2^m - 1 - m$
  - 検査ビット数 $n - k = m$

符号の誤り訂正能力

最小距離：符号の性能評価尺度

- ハミング距離 $d_H(\boldsymbol{u},\boldsymbol{v})$： 2つのベクトル $\boldsymbol{u}, \boldsymbol{v}$ の異なり成分数
- ハミング重み $w_H(\boldsymbol{u}) = d_H(\boldsymbol{u}, \boldsymbol{0})$ ：ベクトル $\boldsymbol{u}$ における1の数
  - 逆に $d_H(\boldsymbol{u},\boldsymbol{v}) = w_H(\boldsymbol{u} - \boldsymbol{v})$
- 最小距離：符号Cにおける2つの符号語の最小ハミング距離

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

  - 線形符号の最小距離は**0**でない符号の重みの最小値になる
    - 線形符号では2つの符号語の差はやはり符号になる
      - 確認： ハミング符号の最小距離は3

最小距離にもとづく誤り訂正・検出能力評価：符号の性能評価尺度

- 最小距離：$d_{\min} = \min_{u \neq v,\ u,v \in C} d_H(u,v)$
- $d_{\min} \geq 2t_1 + 1$ であれば、t_1個以下の誤りを訂正できる
  - さらに、t_1+t_2個以下の誤りを検出できる （$t_2+1 = d_{\min} - 2t_1$）
    - 誤り検出だけを目的とすれば $2t_1+t_2$ 個
- 誤り訂正可能ビット数と誤り検出可能ビット数はトレードオフ
  - t_1 大で誤り訂正力大 → 誤り検出力小
  - 確認：$d_{\min}$ =5のとき