符号の例

ここではいくつかの単純な符号の例について考える。

これらは簡単に理解できるが、伝送速度または誤り確率の点で有効ではない。

より効率的な例については後の章で考える。

例 6.3(反復符号)

体 F 上の反復符号 $\mathcal{R}_n$ は $u \in F$ に対して $\mathbf{u} = uu \dots u \in \mathcal{V} = \mathbf{F}n$ なる符号語から構成される。

よって、$M = \mid F \mid = q$ である。

もし F が体ならば $\mathcal{R}_n$ は $11 \dots 1$ なる符号語 (あるいはベクトル) の張る次元 1 の線形符号になる。

図 6.1 は $\mathcal{V} = \mathbf{F}^3$ の部分空間として $\mathcal{R}_3$ なる二元符号を示している。黒い点が $\mathcal{R}_3$ の符号語である。

5 章では $q = 2$ で n が奇数の場合、$\mathcal{R}_n$ は $(n-1)/2$ 個の誤りを訂正したのを見た。

このことから、$\mathbf{u} \in \mathcal{R}_n$ が伝送され、それらの n 個のシンボルのうち、多くとも $(n-1)/2$ 個が誤って伝送されたとき、最近傍復号は常に正しくなる。

似たような議論によって、どんな q と n に対しても、

$\lfloor x \rfloor = \max \{m \in \mathbf{Z} \mid m \leq x\}$

として、最近傍復号を使えば $\mathcal{R}_n$ は $\lfloor (n-1)/2 \rfloor$ 個の誤りを訂正することが言える。

これは素晴らしいが、運の悪いことに式 (6.2) によって $n \to \infty$ のとき伝送速度 $R = 1/n = 0$ となり、良くないことが言える。

例 6.4(パリティ検査符号)

体 $F = F_q$ 上のパリティ検査符号 $\mathcal{P}_n$ は、$\mathbf{u} = u_1 u_2 \dots u_n \in \mathcal{V}; \sum_i u_i = 0$ なる全てのベクトルから構成される。$u_1, \dots, u_{n-1}$ を情報桁、u_n を $u_n = -u_1 - \cdots - u_{n-1}$ として定義される検査桁としてみなすことができる。

例えば $n = 3$ かつ $q = 2$ のとき、図 6.2 に示されるように $\mathcal{P} = \{000, 011, 101, 110\}$ となる。

一次方程式によって定義されるので、$\mathcal{P}_n$ は線形符号である。

これは次元 $k = n - 1$ で、$\mathcal{V}$ の標準基底ベクトル $\mathbf{e}_i = 000 \dots 010 \dots 0$ の下で、$\mathbf{u}_1 = \mathbf{e}_1 - \mathbf{e}_n, \dots, \mathbf{u}_{n-1} = \mathbf{e}_{n-1} - \mathbf{e}_n$ なる基底を持つ。(これを理解するためには、それぞれのベクトル $\mathbf{u} = u_1 \dots u_n \in \mathcal{P}_n$ が $u_1 \mathbf{u}_1 + \cdots + u_{n-1} \mathbf{u}_{n-1}$ なる唯一の $\mathbf{u}_i$ の線型結合で書けることに注意すると良い。)

このようにして、$M = q^{n-1}$ で、$n \to \infty$ のとき、$R = (n-1)/n$ は $R \to 1$ となり、伝送速度が良いことがわかる。

不幸なことに、この符号は誤り訂正の目的ではほとんど使えない。これは 1 個の誤りを検出するが、それを訂正できないからである。

$\mathbf{u} = u_1 \dots u_n \in \mathcal{P}_n$ が伝送され、$\mathbf{v} = v_1 \dots v_n \in \mathcal{V}$ が受信された場合を考えよう。

受信者は $\sum_i v_i$ を F 上で計算する。一つの誤りがあれば、$\mathbf{v}$ の 1 桁 v_i のみが対応する $\mathbf{u}$ の u_i の桁と異なり、$\sum_i u_i = 0$ から $\sum_i v_i \neq 0$ である。

このとき受信者は $\mathbf{v}$ が符号語でないことと 1 つの誤りがあることを知っている。しかし、どの 1 桁を変更することによっても符号語を得ることができるので、どの桁が間違っているかを判断する方法はない。

更に悪いことに、$\mathbf{v}$ の 2 つ及びそれ以上の相殺される誤りは検知できない。

例 6.5(2 元ハミング符号)

2 元ハミング符号 $\mathcal{H}_7$ は符号長 7、F_2 上の線形符号である。

これは最初に発見された誤り訂正符号の一つで、1947 年にベル研究所の計算機がしばしばクラッシュすることに腹を立てていたハミングという技術者によって導入された。(Ha47, Ha50, Sh48) (初期の誤り訂正の歴史に関する報告については [Th83] を見よ)

この符号を構成するには、3 つの集合 A, B, C のベン図を表した図 6.3 を用いる。$\bar{A} \cap \bar{B} \cap C, \bar{A} \cap B \cap \bar{C}, \bar{A} \cap B \cap C, A \cap \bar{B} \cap \bar{C}, A \cap \bar{B} \cap C, A \cap B \cap \bar{C}, A \cap B \cap C$ に対応する領域に $1, 2, \ldots, 7$ と番号をつける。このように領域の番号 i は A, B, C が含まれる場合それぞれ a, b, c が 1 になるとして、整数 $i = 4a + 2b + c$ の 2 進数表現 abc に対応する。

例えば 5 は 2 進数表現で 101 と書ける、よってこれは $A \cap \bar{B} \cap C$ に対応し、一方で $\bar{A} \cap B \cap \bar{C}$ は 2 の二進数表現 010 に対応している。

4 桁の情報桁 $\mathbf{a} = a_1 a_2 a_3 a_4$ を 7 桁の符号語 $\mathbf{u} = u_1 \ldots u_7$ に符号化する。

まず、$u_3 = a_1, u_5 = a_2, u_6 = a_3, u_7 = u_7$ と定義する。そしてこれらの桁を、図 6.3 の中でそれぞれ 3,5,6,7 の番号の領域に書く。そして、領域 A にある 4 桁の数字の二進数の和が 0 になるように、つまり F_2 上で

$u_4 + u_5 + u_6 + u_7 = 0$

となるように $u_4 = 0$ or 1 を定義し、それを領域 4 に書く。

領域 B と C を用いて、u_2 と u_1 を似たように定義する。つまり、

$u_2 + u_3 + u_6 + u_7 = 0$

$u_1 + u_3 + u_5 + u_7 = 0$

となるように u_2, u_1 を定義する。

(3 つの方程式のそれぞれの添字は、それぞれ二進数表現 a, b, c の 1 桁目、2 桁目、3 桁目に 1 を含んでいるものであることに注意しよう)

符号 $\mathcal{H}_7$ はこの方法で構成された全ての $\mathbf{u} \in \mathcal{V} = \mathbf{F}_2^7$ からなる。

$\mathcal{H}_7$ は変数 u_i の線形な方程式で定義されているので、線形符号である。

a_1, a_2, a_3, a_4 には $2^4 = 16$ 通りの選択があり、それらは $u_1, \ldots, u_7$ を一意に決定するので、$M = \mid \mathcal{H}_7 \mid = 16$ である。

これは更に $\mathcal{H}_7$ が次元 $k = 4$ であり、a_1, a_2, a_3, a_4 のそれぞれの桁を 1、それ以外の桁 a_i を $a_i = 0$ としてを取ることで得られる基底 $\mathbf{u}_1 = 1110000, \mathbf{u}_2 = 1001100, \mathbf{u}_3 = 0101010, \mathbf{u}_4 = 1101001$ を持つことを示す。

この符号は符号語 $\mathbf{u}$ のどんな 1 つの誤りも訂正する。

$\mathbf{u} \in \mathcal{H}_7$ が送信され、そして $\mathbf{v} \in \mathcal{V}$ が受信され、$\mathbf{v}$ は $\mathbf{u}$ と比べて i 桁目 v_i のみ異なるとする。

受信者は F_2 上で

$s_1 = v_4 + v_5 + v_6 + v_7$

$s_2 = v_2 + v_3 + v_6 + v_7$

$s_3 = v_1 + v_3 + v_5 + v_7$

を計算する。$\mathcal{H}_7$ の定義からこれらは全て 0 になるべきであるが、誤っている桁 v_1 はそれらの少なくとも 1 つを 1 にする。

今 v_1 は s_j $(j = 1, 2, 3)$ の式に現れ、i の 2 進数表現の j 桁目は 1 である。よって、これの j 桁目が 0 or 1 であるように $s_j = 0$ or 1 である。

このことを簡単に言うと、s_j はこれの j 桁目なので、$\mathbf{s} = s_1 s_2 s_3$ は i の 2 進数表現である。

受信者は s_1, s_2, s_3 を計算したことで、誤った桁 v_i を見つけ、そして $u_i = u_i + 1$ として定義することでこれを訂正できる。

図のように、$\mathbf{a} = 0110$ を符号化するとしよう。

まず情報桁を $u_3 = 0, u_5 = u_6 = 1, u_7 = 0$ と定義する。そして検査桁 $u_4 = 0, u_2 = 1, u_1 = 1$ を得るため、3 つの一次方程式を解く。よって、送信される符号語は $\mathbf{u} = 1100110$ である。

今、3 桁目で誤りがある、すなわち $\mathbf{v} = 1110110$ が受信された場合を考える。受信者は $s_1 = 0+1+1+0 = 0, s_2 = 1+1+1+0 = 1, s_3 = 1+1+1+0 = 1$ を計算し、$i = 3$ の二進数表現である $\mathbf{s} = s_1 s_2 s_3$ を得る。

こうして $\Delta(\mathbf{v}) = 1100110$ という正しい符号語を得るため、$\mathbf{v}$ の 3 桁目は変更され、この符号語の $3, 5, 6, 7$ 番目の位置から、情報桁 $0, 1, 1, 0$ を展開できる。

この符号はどんな 1 つの誤りでも訂正できるが、2 つ以上の誤りでは訂正に失敗する。例えば上の例では、誤りが u_3 と u_4 にあるとすると、つまり $\mathbf{v}' = 1111110$ が受信されたとする。受信者は $s'_1 = 1, s'_2 = 1, s'_3 = 1$ を計算し、$\mathbf{s}' = s'_1 s'_2 s'_3 = 111$ を与える。これは誤りが $i = 7$ 桁目にあることを示し、$\mathbf{v}'$ は $\Delta(\mathbf{v}) = 1111111$ に復号され、これは誤っている。

練習 6.2　情報桁 1101 を表現する $\mathcal{H}_7$ の符号語を見つけよ、そしてどのようにして 6 桁目の誤りを訂正するかを示せ。4 桁目と 6 桁目に誤りがある場合どうなるか？

2 元符号 $\mathcal{R}_3$ と $\mathcal{H}_7$ は共に 1 桁の誤りを訂正するが、$\mathcal{H}_7$ の伝送速度 $R = 4/7$ は $\mathcal{R}_3$ の $1/3$ よりも明らかに良い。7 章では $n \to \infty$ のとき $R \to 1$ となるような 1 誤り訂正 2 元符号 $\mathcal{H}_n$ $(n = 2^c - 1)$ の列を与えるため、$\mathcal{H}_7$ の構成を一般化する。先立ってどのように出来るかを考えたい場合、3 つの集合 A, B, C を $A_1, \ldots, A_c$ に置き換えてみよ。

例 6.6

C を体 F 上の符号長 n の符号とする。このとき、それぞれの符号語 $\mathbf{u} = u_1 \ldots u_n \in C$ に追加の桁 u_{n+1} を $u_1 + \cdots + u_{n+1} = 0$ となるように添加することで、F 上で符号長 $n+1$ の符号、いわゆる拡大符号 $\bar{C}$ を構成できる。

明らかに $\mid \bar{C} \mid = \mid C \mid$ で、C が線形であれば $\bar{C}$ も同様に同じ次元で線形である。例えば、$C = \mathcal{V} = F^n$ なら $\bar{C} = \mathcal{P}_{n+1} \subset F^{n+1}$ である。

例 6.7

C の符号長が n のとき、それぞれの符号語 $u_1 \ldots u_n \in C$ から i 桁目をを選んで u_i を削除することでパンクチャド符号 C° を構成することが出来る。一般に C° の構造は i の選択に依存する。