離散無記憶通信路

送信側から受信側に文字列を送るときに、文字列が変化する場合の扱いを見ていきます。
前半は単語を定義し、後半は誤り訂正符号を扱っています。

アルファベット $\mathcal{X}$ があり、$\mathcal{X}$ の元による文字列を何かを経由して受け取ることを考えます。何かの部分を通信路 (channel) と呼び、通信路においてもとの文字列から変化するとき雑音 (noise) があると言います。情報理論では通信路が現実の物体としては具体的に何であるかは指定せずに、確率のみを使います。

送信側では $\mathcal{X}$ の元 x_i $(i = 1, 2, \ldots, |\mathcal{X}|$、$|\mathcal{X}|$ は元の数) が確率 p_i で現れ、その確率で通信路に送るとし、送信側の情報源を $(\mathcal{X}, \boldsymbol{p})$ とします。雑音によって受信側では x_i だったものがアルファベット $\mathcal{Y}$ の元 y_j $(j = 1, 2, \ldots, |\mathcal{Y}|)$ になって現れるとし、その情報源を $(\mathcal{Y}, \boldsymbol{q})$ とします（q_j は受信側で見て y_j が現れる確率)。そうすると、通信路に x_i が入り、通信路で変化して y_j になる確率が必要になります。これは x_i が現れる前提で y_j になる確率なので、条件付き確率 $P(y_j|x_i)$ です。つまり、条件付き確率によって雑音のある通信路が特徴づけられます。

アルファベットを $\mathcal{X}, \mathcal{Y}$ と区別して書いていますが、大抵は同じ文字の集まりとします。例えば、$\mathcal{X} = \{0, 1\}$ なら $\mathcal{Y} = \{0, 1\}$ とします。送信側のアルファベットを入力アルファベット、受信側のアルファベットを出力アルファベットと言ったりもします。

というわけで、通信路を使った送受信は、送信側の情報源 $(\mathcal{X}, \boldsymbol{p})$、受信側の情報源 $(\mathcal{Y}, \boldsymbol{q})$、条件付き確率 $P(y_j|x_i)$ によって構成されます。このような通信路は離散的 (discrete) と呼ばれます。

条件付き確率に性質を加えます。例えば、a を送り、その結果として a' を受け取り、次に b を送り b' を受け取ったとします。このとき、a' の確率には a のみ、b' の確率には b のみに依存するとします。このように、送受信の関係がその 1 回ごとで完結している通信路は無記憶 (memoryless) と言われます。無記憶であれば、a, b が起きた前提で a', b' になる条件付き確率は、それぞれが無関係になっているので

$$P(a', b'|a, b) = P(a'|a)P(b'|b)$$

となります。ここでは、離散的で無記憶な通信路である離散無記憶通信路 (discrete memoryless channel) を見ていきます。

単語の定義を与えていきます。送信側と受信側のエントロピーは

$$H(X) = -\sum_i P(x_i) \log P(x_i)\ ,\ H(Y) = -\sum_j P(y_j) \log P(y_j)$$

$\mathcal{X}, \mathcal{Y}$ に含まれる元の数を $|\mathcal{X}|, |\mathcal{Y}|$ として、和の範囲を書いていないときは、$\mathcal{X}$ の元では 1 から $|\mathcal{X}|$、$\mathcal{Y}$ の元では 1 から $|\mathcal{Y}|$ の範囲とします。結合エントロピーと条件付きエントロピーは

$$H(X, Y) = -\sum_{i,j} P(x_i, y_j) \log P(x_i, y_j)$$

$$H(X|Y) = -\sum_{i,j} P(x_i, y_j) \log P(x_i|y_j)\ ,\ H(Y|X) = -\sum_{i,j} P(x_i, y_j) \log P(y_j|x_i)$$

とします。

- 通信路容量

  相互情報量 $I(X;Y)$ は「エントロピー」で触れたように、X と Y が関係しているときに 0 でないので、通信路に関する量と言えます。相互情報量は

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) = \sum_{i,j} P(x_i, y_j) \log \frac{P(x_i, y_j)}{P(x_i)P(y_j)}$$

と定義されます。対数部分は

$$\log \frac{P(x_i, y_j)}{P(x_i)P(y_j)} = \log P(x_i, y_j) - \log P(x_i) - \log P(y_j) = I(x_i) + I(y_i) - I(x_i, y_j) \geq 0$$

$I(x_i, y_j)$ を全体の情報量のように解釈すれば、送信側の情報量と受信側の情報量の和と全体の情報量の差と言えるので、この差は通信路において発生していると考えられます。このようなことから、相互情報量は通信路が持てる情報量と言え、その最大の値を通信路容量 (channel capacity) と定義します。最大値の記号 max を使えば

$$\gamma = \max I(X;Y)$$

max は確率の値に依存します。雑に言えば、$I(X;Y) \geq 0$ で、対数は連続関数 ($\log x$ $(x \neq 0)$)、変数である確率は実数の有限の範囲に収まっている (0 から 1 の値で、それらを全て足せば 1) ということから、取れる範囲内に最大値が存在します。

相互情報量は

$$0 \leq I(X;Y) \leq H(X) \quad (H(X) < H(Y))$$
$$0 \leq I(X;Y) \leq H(Y) \quad (H(Y) < H(X))$$

という関係なので

$$\gamma = \max I(X;Y) \leq \max H(X) \quad (H(X) < H(Y))$$
$$\gamma = \max I(X;Y) \leq \max H(X) \quad (H(Y) < H(X))$$

$H(X), H(Y)$ は全ての確率が等しいときに最大なので

$$\max H(X) = -\sum \frac{1}{|\mathcal{X}|} \log \frac{1}{|\mathcal{X}|} = \log |\mathcal{X}|$$
$$\max H(Y) = \log |\mathcal{Y}|$$

よって

$$\gamma \leq \log |\mathcal{X}| \quad (H(X) < H(Y))$$
$$\gamma \leq \log |\mathcal{Y}| \quad (H(Y) < H(X))$$

- ハミング距離

  送信側が送った文字列 s と受信側が受け取った文字列 s' の長さが同じで

$$s = a_1 a_2 \cdots a_n \ , \ s' = b_1 b_2 \cdots b_n$$

このとき、$a_i \neq b_i$ となっている数をハミング距離 (Hamming distance) と言います。例えば

$$0001101 \ , \ 0000100$$

でのハミング距離は 2 です (右から 1 番目と 4 番目の 2 個)。ハミング距離は $d(s, s')$ と表記されます。ハミング距離は異なる文字の個数なので、1 文字に対するハミング距離 $d(a_i, b_i)$ を $a_i = b_i$ なら 0、$a_i \neq b_i$ なら 1 と定義すれば

$$d(s, s') = d(a_1, b_1) + d(a_2, b_2) + \cdots + d(a_n, b_n)$$

と書けます。

ハミング距離は u, v, w を文字列として

(i) $d(u, u) = 0$.

(ii) $d(u, v) > 0 \quad (u \neq v)$.

(iii) $d(u, v) = d(v, u)$.

(iv) $d(u, v) \leq d(u, w) + d(w, v)$.

これらは距離の定義と同じです。(i),(ii),(iii) はハミング距離の定義からそのままです。(iv) を示します。u, v, w の文字列を

$$u = u_1 u_2 \cdots u_n$$

$$v = v_1 v_2 \cdots v_n$$

$$w = w_1 w_2 \cdots w_n$$

とします。単純に示すなら、1 文字でのハミング距離の場合を見ればいいです。$u = u_1, \ v = v_1, \ w = w_1$ のとき、$u_1 = v_1$ なら $d(u_1, v_1) = 0$ となり、これは最小値なので不等式は成立します。$u_1 \neq v_1$ なら $d(u_1, v_1) = 1$ なので

$$d(u_1, v_1) = 1 \leq d(u_1, w_1) + d(w_1, v_1)$$

$w_1 = u_1$ なら $w_1 \neq v_1$ なので $d(u_1, w_1) + d(w_1, v_1) = 1$、$w_1 \neq u_1$ なら $w_1 \neq v_1$ か $w_1 = v_1$ なので 1 か 2 です。よって、1 文字でのハミング距離の不等式は成立し、文字列のハミング距離は 1 文字のハミング距離の和なので、(iv) は成立します。

文字列のままからも示せます。$A(u, v)$ を $u_i \neq v_i$ となる i の集合とします。例えば

$$u = u_1 u_2 \cdots u_{10} \ , \ v = u_1 v_2 \cdots v_5 u_6 v_7 u_8 v_9 v_{10}$$

となっているなら、$A(u,v)=\{2,3,4,5,7,9,10\}$ です。w が

$$w = v_1u_2w_3v_4w_5\cdots w_8u_9u_{10}$$

となっていると、$A(u,w)=\{1,3,4,5,6,7,8\},\ A(w,v)=\{2,3,5,6,7,8,9,10\}$ となります。この例から、$A(u,v)$ の元は $A(u,w)$ か $A(w,v)$ に含まれています。これは単純で、$A(u,v)$ の元を k として、$A(u,w)$ と $A(w,v)$ は k を含んでいないとすると

$$A(u,v):\ u_k \neq v_k$$

$$A(u,w):\ u_k = w_k$$

$$A(w,v):\ w_k = v_k$$

なので、$u_k \neq v_k,\ u_k = w_k = v_k$ となり矛盾します。このため、$A(u,v)$ の元は $A(u,w)$ か $A(w,v)$ に含まれます。言い換えれば、$A(u,v)$ は和集合 $A(u,w)\cup A(w,v)$ の部分集合です。そうすると、$A(u,v)$ の元の個数は $A(u,w)\cup A(w,v)$ の元の数以下でないといけないと分かり、元の数であるハミング距離は $d(u,v)\leq d(u,w)+d(w,v)$ という関係になります。

- 通信路行列

  通信路は条件付き確率によって特徴づけられており、条件付き確率を行列にしたものを通信路行列 (channel matrix, channel transition matrix) と言います。なので、定義は単純で、条件付き確率を $P(y_j|x_i)$ とすれば、通信路行列 Γ の成分は

$$\Gamma_{ji} = P(y_j|x_i)$$

となります。結合確率 $P(x_i,y_j)$ の関係

$$P(y_j) = \sum_i P(x_i,y_j)$$

$$P(x_i,y_j) = P(y_j|x_i)P(x_i)$$

を使うと

$$\begin{aligned}\sum_i P(x_i,y_j) &= \sum_i P(y_j|x_i)P(x_i)\\ P(y_j) &= \sum_i P(y_j|x_i)P(x_i)\\ &= \sum_i \Gamma_{ji}P(x_i)\end{aligned} \tag{1}$$

このことから、通信路行列は $P(x_i)$ から $P(y_j)$ への変換行列になっています。

ここでは $\Gamma_{ji}=P(y_j|x_i)$ として添え字の並びを与えていますが、ひっくり返して $\Gamma_{ij}=P(y_j|x_i)$ としていることが多いです。これは $\Gamma_{ij}=P(y_j|x_i)$ とすれば、Γ_{ij} の成分を左から読むことで送信側から受信側の並

びになるからです。このようにしない理由は、行列計算を (1) のように書くことや右から左に読むことに個人的に慣れているというだけです。なので、他の本とかを読むときは行列の成分がどちらの定義になっているのかに注意してください。

- n-th extension

  通信路に文字列を入れることは、1 文字ずつ入れていくことと同じです。そして、無記憶であるなら 1 文字ずつで完結しているので、文字列を送るときの通信路の条件付き確率は 1 文字での条件付き確率の積で表せます。長さ n の文字列 $s = x^{(1)}x^{(2)}\cdots x^{(n)}$ から $s' = y^{(1)}y^{(2)}\cdots y^{(n)}$ となる条件付き確率は

$$P(s'|s) = P(y^{(1)}|x^{(1)})P(y^{(2)}|x^{(2)})\cdots P(y^{(n)}|x^{(n)}) = \Gamma^{(1)}\Gamma^{(2)}\cdots\Gamma^{(n)}$$

  この条件付き確率を離散無記憶通信路の n-th extension と言います。

  n-th extension での通信路容量は 1 個の通信路容量の n 倍になることを示します。n-th extension ではエントロピーの変数が n 個になるので、そのときの表記を与えます。1 文字のときは

$$H(X) = -\sum_i P(x_i)\log P(x_i)$$

  例えば、2 文字では $x_i^{(1)}, x_j^{(2)}$ ($x_i^{(k)}$ での (k) は文字の区別、i は元の区別) として

$$H(X_1, X_2) = -\sum_{i,j} P(x_i^{(1)}, x_j^{(2)})\log P(x_i^{(1)}, x_j^{(2)})$$

  となりますが、これを省略して

$$H(\boldsymbol{X}) = -\sum_{\boldsymbol{x}} P(\boldsymbol{x})\log P(\boldsymbol{x})$$

  と書くことにします。$\boldsymbol{x}$ は $\boldsymbol{x} = (x_i^{(1)}, x_j^{(2)})$ のようにしており、和での $\boldsymbol{x}$ は $x_i^{(1)}, x_j^{(2)}$ での i, j に対して取るという意味です。n 個でも同様にします。条件付きエントロピーでも

$$\begin{aligned} H(\boldsymbol{Y}|\boldsymbol{X}) =& -\sum_{\boldsymbol{x},\boldsymbol{y}} P(\boldsymbol{x},\boldsymbol{y})\log P(\boldsymbol{y}|\boldsymbol{x}) \\ =& -\sum_{i_1,i_2,j_1,j_2} P(x_{i_1}^{(1)}, x_{i_2}^{(2)}, y_{j_1}^{(1)}, y_{j_2}^{(2)})\log P(y_{j_1}^{(1)}, y_{j_2}^{(2)}|x_{i_1}^{(1)}, x_{i_2}^{(2)}) \end{aligned}$$

  と表記します。無記憶では X_1, Y_1 と X_2, Y_2 は無関係なので

$$\begin{aligned} H(\boldsymbol{Y}|\boldsymbol{X}) =& -\sum_{i_1,i_2,j_1,j_2} P(x_{i_1}^{(1)}, x_{i_2}^{(2)}, y_{j_1}^{(1)}, y_{j_2}^{(2)})\log P(y_{j_1}^{(1)}|x_{i_1}^{(1)}) \\ & -\sum_{i_1,i_2,j_1,j_2} P(x_{i_1}^{(1)}, x_{i_2}^{(2)}, y_{j_1}^{(1)}, y_{j_2}^{(2)})\log P(y_{j_2}^{(2)}|x_{i_2}^{(2)}) \\ =& -\sum_{i_1,j_1} P(x_{i_1}^{(1)}, y_{j_1}^{(1)})\log P(y_{j_1}^{(1)}|x_{i_1}^{(1)}) - \sum_{i_2,j_2} P(x_{i_2}^{(2)}, y_{j_2}^{(2)})\log P(y_{j_2}^{(2)}|x_{i_2}^{(2)}) \\ =& H(Y_1|X_1) + H(Y_2|X_2) \end{aligned}$$

n 個でも同様です。

n-th extention での通信路容量の定義は

$$\gamma = \max I(\boldsymbol{X};\boldsymbol{Y})$$

相互情報量の定義は変わらないので

$$I(\boldsymbol{X};\boldsymbol{Y}) = H(\boldsymbol{Y}) - H(\boldsymbol{Y}|\boldsymbol{X}) = H(\boldsymbol{X}) - H(\boldsymbol{X}|\boldsymbol{Y})$$

無記憶での条件付きエントロピーは

$$H(\boldsymbol{Y}|\boldsymbol{X}) = H(Y_1|X_1) + \cdots + H(Y_n|X_n)$$

と書けることと

$$H(X_1, X_2, \ldots, X_N) \leq H(X_1) + H(X_2) + \cdots + H(X_N)$$

から

$$\begin{aligned} H(\boldsymbol{X};\boldsymbol{Y}) = H(\boldsymbol{Y}) - H(\boldsymbol{Y}|\boldsymbol{X}) &\leq H(Y_1) + \cdots + H(Y_n) - (H(Y_1|X_1) + \cdots + H(Y_n|X_n)) \\ &= H(Y_1) - H(Y_1|X_1) + \cdots + H(Y_n) - H(Y_n|X_n) \\ &= n\gamma \end{aligned}$$

となるので、n-th extension での通信路容量は 1 個の通信路容量の n 倍です。

- 2 元対称通信路

  情報源のアルファベットを $\{0, 1\}$ とします。送信側から 0 を通信路に送ると受信側では 0 か 1 になり、1 を送ると受信側では 0 か 1 になる離散無記憶通信路のことを、2 元対称通信路 (binary symmetry channel) と言います。2 元対称通信路では通信路の条件付き確率は 4 個だけで、送信側で 0 のときは $P(0|0), P(1|0)$、1 のときは $P(0|1), P(1|1)$ となります。これらは確率の規則から

$$P(0|0) + P(1|0) = 1\ ,\ P(0|1) + P(1|1) = 1$$

となります。2 元対称通信路は、送信側から受信側に行くときに変化が起きる確率を ϵ とすれば

$$P(0|0) = 1 - \epsilon\ ,\ P(0|1) = \epsilon$$

$$P(0|1) = \epsilon\ ,\ P(1|1) = 1 - \epsilon$$

となり、これによって 2 元対称通信路での通信路行列 Γ は

$$\Gamma_{00} = P(0|0) = 1 - \epsilon \ , \ \Gamma_{01} = P(0|1) = \epsilon$$

$$\Gamma_{10} = P(1|0) = \epsilon \ , \ \Gamma_{11} = P(1|1) = 1 - \epsilon$$

$\epsilon = 0$ なら対角行列になります。続きの話は「2 元対称通信路」でしていきます。

ここからは誤り訂正符号（error-correcting code）を見ていきます。通信路で文字が変化する可能性がある場合、受信側は送られてきた文字列をもとの文字列に再現する方法を持っている必要があり、その方法が誤り訂正符号です。

使うアルファベットは $\mathcal{B} = \{0, 1\}$ とします。なので、送信側は $\{0, 1\}$ から作られる prefix-free の符号 $\mathcal{C}$ による文字列を受信側に送るとします。今は通信路で変化する場合を扱いたいので、長さ n の符号語 c を送って受信側では長さ n の文字列 z になったとします。受信側では符号語ではなくなっている可能性がありますが、$\mathcal{S}_B^n$ の文字列 (長さ n の文字列) にはなっていることから、$z \in \mathcal{S}_B^n$ から $c \in \mathcal{C}$ への変換が存在しているとし、それを関数 σ とします ($\mathcal{S}_B^n$ の文字列を符号語に変換する関数)。このように、受信側で受け取った文字列と符号語を対応させる方法は decision rule と言います。日本語だと復号法と言ったりしています。decision rule は細かく言えば、受信した文字列を適当な長さに分割し、分割されたものを符号語に戻すことです。

具体的な例を見ておきます。離れた人に前後左右に動くように伝えるとします。もとの単語は前、後、左、右の 4 個なので、必要な符号語は 4 個です。$\{0, 1\}$ から prefix-free に作るなら

$$\mathcal{C} = \{00, 01, 10, 11\}$$

これを前、後、左、右に対応させます。この符号において通信路で例えば 00 が 01 に変化してしまうと、前と伝えたいのに後と伝わってしまいます。これを防ぐために、符号語の長さを増やすという手段が取れます。

1 文字増やして符号語の長さを 3 にしたとして

$$\mathcal{C} = \{000, 110, 101, 011\}$$

と作ります。000 を送ったとき通信路で 1 文字だけ変化するなら、$001, 010, 100$ になります。このどれかを受信側が受け取ったなら、この 3 個はどれも他の符号語になっていないので

$$\sigma(000) = 000 \ , \ \sigma(001) = 000 \ , \ \sigma(010) = 000 \ , \ \sigma(100) = 000$$

とすれば、もとの符号語が 000 と判別できます。しかし、001 でも同様にしてみると、001 は $000, 011, 001$ と変化するために、受信側において符号語 011 として現れてしまう可能性があります。このため、符号語 001 と 011 を区別できなく、もとの文字列に戻せないです。さらに言えば、符号語 101 では 1 文字の変化で 001 が現れるので $\sigma(001) = 000$ も有効ではなくなります。というわけで、符号語の長さを 3 にしても解決していません。

解決していませんが改善はしています。なので、符号語の長さをさらに増やして、どの符号語の変化においても重複がなくなれば、通信路で変化が起きても受信者はもとの文字列を再現できるようになります。しかし、符号語を何も考えずにただ長くするというだけでは効率が悪すぎるので、文字列からもとの符号語を再現できるための方法を考えます。

変化した文字列ともとの符号語に関連する確率は通信路を特徴づける $P(z|c)$ です。なので、逆向きに見て (受信側から見て)$P(c|z)$ が最も大きくなる c を $\sigma(z) = c$ として選ぶ方法を最適復号法 (ideal observer rule) と呼びます (z が与えられたときに最も高い確率で現れる c を選ぶ方法)。つまり、z を受け取ったとき、符号語 $c_1, c_2, \ldots, c_n$

の中の c_i で最も高い確率になるなら、$\sigma(z) = c_i$ と復号させるということです。しかし、$P(c|z)$ は送信側の符号に依存しており、受信側は $P(c|z)$ を直接知っていることはないので、$P(c|z)$ を改めて求める必要があります。符号の数が増えるほどこの手間は増えていきます。

受信側は $P(c|z)$ を知らなくても通信路を特徴づける $P(z|c)$ は知っているとして、$P(z|c)$ が最も高くなる符号語を $\sigma(z)$ に選ぶ方法を最尤復号法 (maximum likelihood rule) と呼びます。最尤 (さいゆう、maximum likelihood) は統計学で対象に対して最も確率の高い分布を選ぶという意味で使われている単語です。

2 元対称通信路とします。ϵ を文字が変化する確率として、長さ n の文字列 z と符号語 c, c' による条件付き確率は

$$P(z|c) = \epsilon^d (1-\epsilon)^{n-d} \ , \ P(z|c') = \epsilon^{d'} (1-\epsilon)^{n-d'}$$

1 個の 2 元対称通信路を n-th にしているだけです (「2 元対称通信路」参照)。d は z と c、d' は z と c' とのハミング距離です。簡単に言えば、無記憶なので 1 文字ずつの条件付き確率の積になり、変化した文字の数はハミング距離 $d(z, c)$ なので ϵ^d、変化しない文字の数は $n - d$ なので $(1-\epsilon)^{n-d}$ になるというだけです。

これらの割合は

$$\frac{P(z|c)}{P(z|c')} = \frac{\epsilon^d (1-\epsilon)^{n-d}}{\epsilon^{d'} (1-\epsilon)^{n-d'}} = \frac{\epsilon^d (1-\epsilon)^{-d}}{\epsilon^{d'} (1-\epsilon)^{-d'}} = \epsilon^{-(d'-d)} (1-\epsilon)^{d'-d} = (\frac{1-\epsilon}{\epsilon})^{d'-d}$$

変化する確率が少なくとも $1/2$ よりは小さくなっている通信路を使うべきなので (変化しないことより変化することが多い通信路は使いたくない)、$\epsilon < 1/2$ とします。そうすると

$$\begin{aligned} \frac{1}{2} &< 1 - \epsilon \\ 1 &< 2(1-\epsilon) \\ &< \frac{1-\epsilon}{\epsilon} \quad (\frac{1}{\epsilon} > 2) \end{aligned}$$

となっているので、$d < d'$ では

$$\begin{aligned} \frac{P(z|c)}{P(z|c')} &> 1 \\ P(z|c) &> P(z|c') \end{aligned}$$

よって、2 元対称通信路において最尤復号法 ($P(z|c) > P(z|c')$ での c を選ぶ方法) は、$\epsilon < 1/2$ のとき、z とのハミング距離が最も小さくなる符号語を選ぶと言い換えられます。

ハミング距離が最も小さい符号語を選ぶ方法を最小距離復号法 (minimum distance decoding rule, nearest neighbour decoding rule) と言います。これも感覚的に有効と思える方法で、異なる文字が少ないもの同士を対応させます。これをさらに見ていきます。

最小距離復号法で受信側の文字列をもとの符号語に戻せる条件を求めます。まず、異なる文字の数が $\sigma(z)$ と c の関係にどう関わるのかを簡単な例を使って見ておきます。符号語の長さを 3 とし、通信路では最大で 1 文字しか変化しないとします。例えば、送信側が 000 を送ったとすれば、受信側は

$$000 \ \Rightarrow \ 000 \ , \ 001 \ , \ 010 \ , \ 100 \tag{2}$$

のどれかを受け取ります。001 を送ったとすれば

$$001 \ \Rightarrow \ 001\ ,\ 000\ ,\ 011\ ,\ 101 \tag{3}$$

(2) と (3) では 001 が重複してしまっているので、000 と 001 は判別できません。しかし、111 を送ると

$$111 \ \Rightarrow \ 111\ ,\ 110\ ,\ 101\ ,\ 011 \tag{4}$$

となるために (2) と (4) には重複がなく、000 と 111 は判別できます。このため符号 $\mathcal{C} = \{000, 111\}$ では 1 文字までしか変化しない通信路を使うなら、受信側は送られてきた符号語を再現できます。他にも

$$001 \ \Rightarrow \ 001\ ,\ 000\ ,\ 011\ ,\ 101 \ \Rightarrow \ \sigma(001) = \sigma(000) = \sigma(011) = \sigma(101) = 001$$

$$110 \ \Rightarrow \ 110\ ,\ 111\ ,\ 100\ ,\ 010 \ \Rightarrow \ \sigma(110) = \sigma(111) = \sigma(100) = \sigma(010) = 110$$

とできるので、$\mathcal{C} = \{001, 110\}$ でも再現できます。

このようになるのは、通信路の 1 文字の変化に対して重複を起こさせないためには、3 文字異なっている符号語でないといけないからと言えます。

今度は符号語の長さを 4 にし、通信路では 1 文字までしか変化しないとします。0000 を送ると、0000 と重複がないのは例えば 0111 と 1111 で、0000 は 0111 とは 3 文字、1111 とは 4 文字異なっています。ただし、0111 と 1111 には通信路の変化に対して重複があるので同時に選べなく、$\mathcal{C} = \{0000, 0111\}$ か $\mathcal{C} = \{0000, 1111\}$ として選ぶ必要があります。　長さが 4 でも 2 個の符号語しか選べない理由は簡単です。0000(他の場合でも同様) でない符号語 c_1, c_2 があるとし、c_1, c_2 を 0000 から 3 文字異なっているように作ろうとします。そうすると、c_1 は 1 を 3 個含ませればよく、c_2 は c_1 で残っている 0 を 1 にし c_1 での 3 個の 1 の中から 1 個を 0 にすれば 0000 とは 3 文字異なるように作れます。しかし、c_1 と c_2 の異なる文字は 2 個になるので、通信路での 1 文字の変化に対して重複が起きます。実際に、c_1, c_2 として与えられるのは

$$1110\ ,\ 1101\ ,\ 1011\ ,\ 0111$$

となっており、これらは 1 文字の変化に対して重複します (お互いに 2 文字しか異なってないため)。さらに、これらは 0000 から 4 文字異なっている 1111 とは 1 文字しか違わないので、1111 と同時に使えないです。

このように、符号語のお互いに異なっている文字の個数と通信路で起きる文字の変化の個数とが、受信側がもとの符号語を再現できるかに関わっています。今の例からの推測として、符号に含まれる符号語は全て同じ長さとし、通信路における文字の変化が最大で r、送った符号語 c と他の符号語 c_j の差が $c - c_j \geq 2r + 1$ であるなら、受信者が受け取った文字列 z は $\sigma(z) = c$ と判別できると言えます (z は符号語の長さと同じ)。実際にそうなっていることを示していきます。

ハミング距離が最も小さくなる符号語を選ぶとどうなるのか見ていきます。ハミング距離と関係して、ある文字列 v に対して $d(v, u) \leq r$ となる u の集合を $N_r(v)$ と書くことにします。これは閉球体と同じ定義です (中心を v、半径を r とする球)。

2 個以上の符号語を含む符号 $\mathcal{C}$ において最も小さなハミング距離を δ_c で表します。例えば、符号が $\{0001, 0011, 1111\}$ であるなら

$$d(0001, 0011) = 1\ ,\ d(0001, 1111) = 3\ ,\ d(0011, 1111) = 2$$

なので、$\delta_c = 1$ です。また、符号語 c_1, c_2, c_3, c_4 があり、$d(c_1, c_3) = 2,\ d(c_3, c_4) = 2$ のようになっていても $\delta_c = 2$ とします。

- 符号 $\mathcal{C}$ において $\delta_c \geq 2r+1$ のとき、$N_r(c_1), N_r(c_2)$ $(c_1, c_2 \in \mathcal{C})$ は共通部分を持たない。

  符号 $\mathcal{C}$ において、最も小さなハミング距離は r_0 以上になっているとします ($\delta_c \geq r_0$)。このとき、ある文字列 z があり、この z は符号語 c_1, c_2 での $N_r(c_1), N_r(c_2)$ の両方に含まれているとします (ハミング距離を与えるために c_1, c_2, z の長さは同じ)。そうすると、ハミング距離の三角不等式から

$$d(c_1, c_2) \leq d(c_1, z) + d(z, c_2)$$

  z は $N_r(c_1), N_r(c_2)$ に含まれているので

$$d(c_1, z) \leq r \ , \ d(z, c_2) \leq r$$

  これを入れれば

$$d(c_1, c_2) \leq 2r \tag{5}$$

  これから、$\mathcal{C}$ の符号語のハミング距離は $2r$ を超えないことになります。一方で、符号においてハミング距離の最も小さな値 δ_c が存在し、それを $\delta_c \geq 2r+1$ と仮定すると (5) と矛盾します (最も小さなハミング距離は $2r+1$ 以上と仮定すると、$2r$ を超えることはないと言っている (5) と矛盾)。今の z は $N_r(c_1), N_r(c_2)$ の共通部分にいるということなので、$N_r(c_1), N_r(c_2)$ が共通部分を持つとき、ハミング距離の最も小さな値 δ_c は $\delta_c \geq 2r+1$ とはならないです。これの対偶は、ハミング距離の最も小さな値 δ_c が $\delta_c \geq 2r+1$ のとき、$N_r(c_1), N_r(c_2)$ は共通部分を持たないとなります。

- 符号 $\mathcal{C}$ において $\delta_c \geq 2r+1$ なら $d(c, z) < d(c', z)$ $(c, c' \in \mathcal{C},\ c' \neq c)$。

  符号 $\mathcal{C}$ が $\delta_c \geq 2r+1$ になっているとします。もし文字列 z が c での $N_r(c)$ に含まれているなら (c との文字の違いが r 個以下)、今の結果から $c' \neq c$ の符号語での $N_r(c')$ には含まれません (c' との文字の違いが r 個を超えている)。つまり

$$d(c, z) \leq r \ , \ d(c', z) > r \tag{6}$$

  なので、$d(c, z) < d(c', z)$ となります。これは、c との違いが r 個以下である文字列 z は、c' $(c' \neq c)$ とでは違いはより多くなるということです。

これで何が言えるのかをまとめます。$\delta_c \geq 2r+1$ の符号を使うとし、通信路で起こる文字の変化が r 個以下と決まっているとします。この通信路を使って、送信側が符号語 c を送り、受信側が文字列 z として受け取ったとします。そうすると、(6) から、受け取った z との文字数の違いが r 以下になっている符号語とそれ以外の r を超えている符号語が現れます。通信路の設定から、r を超えて文字が変化することはないので、違いが r 以下の符号語が z のもとの符号語になります。簡単に言えば、受け取った文字列とのハミング距離が最小になる符号語がもとの符号語になります。

この結果は、最小距離復号法で受信側の文字列をもとの符号語に戻せる符号を $\delta_c \geq 2r+1$ なら構成できると言っています。このような、通信路での変化が r 個以下のとき、最小距離復号法によって文字列をもとの符号語に戻せる符号を r-誤り訂正符号 (r error-correcting code) と呼びます。

単純な例としては、通信路では最大で 1 文字しか変化しないとき、$\{000, 111\}$ は 1-誤り訂正符号です。1 文字しか変化しないなら

$$000 \ \Rightarrow \ 000\ ,\ 001\ ,\ 010\ ,\ 100 \ \Rightarrow \ \sigma(000)=\sigma(001)=\sigma(010)=\sigma(100)=000$$

$$111 \ \Rightarrow \ 111\ ,\ 110\ ,\ 101\ ,\ 011 \ \Rightarrow \ \sigma(111)=\sigma(110)=\sigma(101)=\sigma(011)=111$$

ハミング距離での判別ができるので、受信側は最小距離復号法でもとの符号語に戻せます。

誤り訂正符号に使える符号語の数には制限があることも分かります。今は符号語 c_i による $N_r(c_i)$ はそれぞれ重なっていないという条件があります。言い換えれば、長さ n の文字列の個数 2^n の中に $N_r(c_i)$ $(i=1,2,\ldots,M)$ が重なることなく置かれているということです。そして、$N_r(c_i)$ に含まれる文字列 z は $d(c_i,z)\leq r$ という条件を満たすものです。

$d(c_i,z)=k$ は、n 個の文字のうち k 個が c_i,z で異なっているということなので、そのような文字列の個数は

$$ {}_nC_k=\frac{n!}{k!(n-k)!}$$

と求まります。$k\leq r$ となるものが $N_r(c_i)$ に含まれている文字列の総数になるので、c_i もそれに含めれば

$$1+{}_nC_1+{}_nC_2+\cdots+{}_nC_r$$

2^n をこれで割った個数だけ $N_r(c_i)$ が置けるので

$$M\leq\frac{2^n}{1+{}_nC_1+{}_nC_2+\cdots+{}_nC_r}$$

これをハミングの不等式と言い、誤り訂正符号の符号語の数には上限があることが分かります。ただし、例えば $n=4,r=1$ では $M\leq 3.2$ となるので、符号語は 3 個まで使えるように思えますが、上での例で見たように実際には 2 個までしか使えないです。このように、ハミングの不等式は使える符号語の最大の数を教えてはくれません。