

2016年度 通信理論

第5回 データ圧縮 (ZL符号, ランレングス符号)

2016. 4.21

植之原 裕行 uenohara.h.aa@m.titech.ac.jp

講義スケジュール(前半)

	日付	教科書	内容
第1回	4/7	第1章・第2章	通信システム概論: 通信伝送のモデルと具体例
第2回	4/11	第3章	情報源符号化 (情報量, エントロピー, 情報源符号化定理)
第3回	4/14	第4章2節	データ圧縮 (情報源符号化, 語頭符号, ハフマン符号)
第4回	4/18	第4章3節～4節	データ圧縮(イライアス符号)
第5回	4/21	第4章5節～6節	データ圧縮(ZL符号, ランレンジス符号)
第6回	4/25	第5章	通信路容量 (通信路の確率モデル, 相互情報量, 通信路容量)
第7回	4/28	第6章1節～2節	通信路符号化 (通信路符号化, 通信路符号化定理, 繰返し符号)
第8回	5/2	第1章～ 第6章2節	通信理論に関する演習

講義スケジュール(後半)

	日付	教科書	内容
第9回	5/ 9	第6章3、4節	誤り訂正符号(線形ブロック符号)
第10回	5/12	第6章5節	誤り訂正符号(巡回符号, 畳み込み符号)
第11回	5/16	第6章6節	誤り訂正符号(LDPC符号)
第12回	5/19	第7章	連続情報と連続通信路
第13回	5/23	第8章1節～2節	暗号の基礎理論
第14回	5/26	第8章3節～7節	暗号の例(共通鍵暗号、ハッシュ関数)
第15回	5/30	第8章3節～7節	暗号の例(公開鍵暗号)
	6/ 2		期末試験(予定)

ジブ・レンペル (Ziv-Lempel) 符号 (ZL符号)

- Unixの圧縮ツール「compress」の基本アルゴリズム
- (特徴) 文字列の増分分解
セグメントと呼ばれる部分文字列の接続に分解。
各セグメントに対して符号化を行い、無歪圧縮を実現。
- (特徴)
 - ・情報源シンボルの出現確率が既知でなくてよい。
 - ・漸近的最良性を備える。
 - ・計算量は情報シンボル長さの多項式オーダーで済む。

増分分解

- ・情報源アルファベット A の文字集合を

$$A = \{0, 1, 2, \dots, M - 1\}$$

で与えられる整数の部分集合とする。

- ・文字系列を以下で表す。

$$x_i^j = \begin{cases} x_i x_{i+1} \cdots x_j & (\text{if } i \leq j) \\ \emptyset & (\text{if } i > j) \end{cases} \quad \leftarrow \text{長さ0の文字列(空文字列)}$$

● 増分分解(符号化する文字列への分解)

文字列 $x_1^n = x_1 x_2 \cdots x_n$ を6・7ページ①②③によりセグメント(部分文字列)

$$X_k = x_{s(k-1)+1}^{s(k)}$$

(X_k の最後の文字を $x_{s(k)}$ と表記, また $s(0) = 0$)

の連接 $x_1^n = X_0 X_1 \cdots X_p$ に分解することを文字列 x_1^n の増分分解という。

増分分解の手順(1)

● 増分分解の手順

① $X_0 = \emptyset$

② $X_0X_1X_2 \cdots X_{i-1}$ までが得られ、

$$x_1^n = \underbrace{X_0X_1X_2 \cdots X_{i-1}}_{\phi \text{ (途中まで) 分解された文字列}} \overline{X_{i-1}} \quad \text{残りの文字列}$$

$$\overline{X_{i-1}} = x_{s(i-1)+1}^n$$

とする。

$s(i-1) = n$ ならば、 $\overline{X_{i-1}} = \emptyset$ であり、分解を終了。

$\overline{X_{i-1}} \neq \emptyset$ ならば③に進む。

増分分解の手順(2)

③ $\overline{X_{i-1}}$ ($\neq \emptyset$)の語頭になっている既出のセグメントの中で最長のセグメントを $X_{i'}$ ($i' < i$)とし、次のセグメント X_i を

$$X_i = \begin{cases} \overline{X_{i-1}} & (\text{if } X_{i'} = \overline{X_{i-1}}) \\ \underbrace{X_{i'}} \cdot \underbrace{x_{s(i-1)+1+|X_{i'}|}} & (\text{if } X_{i'} \neq \overline{X_{i-1}}) \end{cases}$$

既出語頭の次の1文字でセグメントを区切る

既出のセグメントのうち最長のもの ($\overline{X_{i-1}}$ の語頭になっている)

で定め、 $i \leftarrow i + 1$ として②に戻る。

符号化アルゴリズム

ブロック長 n を定めて、以下を行う。

① 情報源文字列 $x_1^n \in A^n$ の増分分解を求め、それを $x_1^n = X_0 X_1 \cdots X_p$ とする。

② 各セグメント X_i ($i = 1, 2, \dots, p$)に対して、 i' を以下で与えられる正整数として、 $c(X_i)$ を整数の和

$$c(X_i) = \begin{cases} Mi' + x_{s(i)}, & \text{if } X_i \neq X_{i'} \\ Mi', & \text{if } X_i = X_{i'} \end{cases}$$

M : アルファベットの文字数

語頭となっているセグメントの右下数字

セグメント最終文字の数字

によって定める。

ただし、 $x_{s(i)} \in A = \{0, 1, \dots, M-1\}$

→ $c(X_i)$ を $L_i = \lceil \log_2(Mi) \rceil$ ビットの2進整数 b_i で表して、 $b_1, b_2, \dots, b_p$ の接続 $b = b_1 b_2 \cdots b_p$ を x_1^n に対する符号語とする。

(注) 符号長 $L(x_1^n) = \sum_{i=1}^p L_i = \sum_{i=1}^p \lceil \log_2(Mi) \rceil$

復号化アルゴリズム(1)

$i = 1, 2, \dots, p$ に対して下記を行う。

① 2進系列 b を $L_i = \lceil \log_2(Mi) \rceil$ ビットごとに区切って b_i を求める。

② $b_i = Mi' + x_{s(i)}$

$$x_{s(i)} \in A = \{0, 1, \dots, M-1\}$$

b_i を M で割った余り $\langle b_i \rangle_M = x_{s(i)}$ とすると、

$i' = (b_i - \langle b_i \rangle_M) / M$ として $i' (\leq i - 1)$ が定まる。

$\rightarrow X_i = X_{i'} \cdot x_{s(i)}$ として定まる。

ただし、 $s(i) (i \leq p - 1)$ は

$s(i) = |X_{i'}| + 1 + s(i - 1)$ により定まる。

復号化アルゴリズム(2)

- ③ $b_1, b_2, \dots$ に対して復号を行っていくと、 b_k を復号して $X_k = X_{k'} \cdot x_{s(k)}$ を得た段階で復号された情報源系列 $X = X_1 X_2 \cdots X_k$ の長さが n あるいは $n+1$ になる k が存在。
→ この k が p である。
- $|X| = n$ であれば X が元の情報系列である。
 - $|X| = n + 1$ であれば $x_{s(k)} = 0$ となり、 X から $x_{s(k)}$ を除いた $X' = X_1 X_2 \cdots X_{k'}$ が元の情報系列である。

符号化例(1)

• $A = \{a_1 = 0, a_2 = 1\}$

・情報源系列を $n = 5, x_1^5 = x_1x_2x_3x_4x_5 = a_1a_1a_2a_1a_1$ とする。

● 符号化

・増分分解を行う。

① $X_0 = \emptyset$

② $i = 1$ とする。

$$\overline{X_0} = x_{s(0)+1}^5 = x_1^5 = a_1a_1a_2a_1a_1 \rightarrow s(0) = 0$$

$\overline{X_0} \neq \emptyset$ なので、 $\overline{X_0}$ の語頭になっている既出のセグメント中、
最長のセグメント $X_{i'} = X_0 = \emptyset$ ($i' = 0$)

$X_{i'} = X_0 \neq \overline{X_0}$ なので、

つまり先頭の文字

$$X_1 = X_0 \cdot x_{s(0)+1+|X_0|} = X_0 \cdot x_1 = a_1 \rightarrow i = 2 \text{ とする。}$$

符号化例(2)

• $i = 2$ に対して、

$$\overline{X_1} = x_{s(1)+1}^5$$

X_1 の最後の文字は a_1 で、 $x_{s(1)} = a_1$ である。

$$X_1 = x_{s(0)+1}^{s(1)} = x_1^{s(1)} = a_1 \rightarrow s(1) = 1$$

$$\overline{X_1} = x_2^5 = a_1 a_2 a_1 a_1$$

$\overline{X_1} \neq \emptyset$ なので、 $\overline{X_1}$ の語頭になっている既出のセグメント中、
最長のセグメント $X_{i'} = a_1 \rightarrow i' = 1$

$$X_2 = X_1 \cdot x_{s(1)+1+|X_1|} = X_1 \cdot x_3 = X_1 \cdot a_2 \rightarrow i = 3 \text{とする。}$$

$\overline{X_1}$ の先頭が X_1 であり、その
次の文字が a_2 である。

符号化例(3)

• $i = 3$ に対して、

$$\overline{X_2} = x_{s(2)+1}^5$$

X_2 の最後の文字は a_2 で、 $x_{s(2)} = a_2$ である。 $\rightarrow s(2) = 3$

$$X_2 = x_{s(1)+1}^{s(2)} = x_2^3 = a_1 a_2$$

$$\overline{X_2} = x_4^5 = a_1 a_1$$

$\overline{X_2} \neq \emptyset$ なので、 $\overline{X_2}$ の語頭になっている既出のセグメント中、
最長のセグメント $X_{i'} = X_1 = a_1 \rightarrow i' = 1$

$$X_3 = X_1 \cdot x_{s(2)+1+|X_1|} = X_1 \cdot x_5 = X_1 \cdot a_1 \rightarrow i = 4 \text{とする。}$$

$\overline{X_2}$ の先頭が X_1 であり、その
次の文字が a_1 である。

符号化例(4)

- $i = 4$ に対して、

$$\overline{X_3} = x_{s(3)+1}^5$$

X_3 の最後の文字は a_1 で、 $x_{s(3)} = a_1$ である。

$$X_3 = x_{s(2)+1}^{s(3)} = x_4^{s(3)} = a_1 a_1 \rightarrow s(3) = 5$$

$$\overline{X_3} = \emptyset \rightarrow \text{分解終了}$$

符号化例(5)

- ・符号化に移る。

$$L_1 = \lceil \log_2(M \times 1) \rceil = \lceil \log_2(2 \times 1) \rceil = 1$$

$$c(X_1) = 2 \times 0 + x_{s(1)} = 0 + a_1 = 0 \rightarrow 2\text{進数は} "0"$$

$i' = 0$

$$L_2 = \lceil \log_2(M \times 2) \rceil = \lceil \log_2(2 \times 2) \rceil = 2$$

$$c(X_2) = 2 \times 1 + x_{s(2)} = 2 + x_3 = 2 + a_2 = 3 \rightarrow 2\text{進数は} "11"$$

$i' = 1$

$$L_3 = \lceil \log_2(M \times 3) \rceil = \lceil \log_2(2 \times 3) \rceil = 3$$

$$c(X_3) = 2 \times 1 + x_{s(3)} = 2 + x_5 = 2 + a_1 = 2 \rightarrow 2\text{進数は} "010"$$

$i' = 1$

$$\therefore c(a_1 a_1 a_2 a_1 a_1) = (0, 11, 010) \leftarrow \text{実際はカンマはなし。}$$

復号化例(1)

- 受信系列 $b = (011010 \dots)$ を長さ $L_1 = 1, L_2 = 2, L_3 = 3, \dots$ のビット列に分割。

$$\rightarrow (b_1, b_2, b_3, \dots) = (0, 11, 010, \dots)$$

- b_1, b_2, b_3 を復号化アルゴリズム②で復号化。

$i = 1$ に対して、

$$\langle b_1 \rangle_M = \langle 0 \rangle_2 = 0 \rightarrow x_{s(1)} = 0$$

$$i' = \frac{(b_1 - \langle b_1 \rangle_M)}{M} = 0$$

$$s(1) = |X_0| + 1 + s(0) = 1$$

$$X_1 = X_0 \cdot x_1 = X_0 \cdot a_1$$

復号化例(2)

$i = 2$ に対して、

$$\langle b_2 \rangle_M = \langle 11 \rangle_2 = 1 \rightarrow x_{s(2)} = 1$$

$$i' = \frac{(b_2 - \langle b_2 \rangle_M)}{M} = 1$$

$$s(2) = |X_1| + 1 + s(1) = 3$$

$$X_2 = X_1 \cdot x_3 = a_1 x_3 = a_1 a_2 = 01$$

$i = 3$ に対して、

$$\langle b_3 \rangle_M = \langle 010 \rangle_2 = 0 \rightarrow x_{s(3)} = 0$$

$$i' = \frac{(b_3 - \langle b_3 \rangle_M)}{M} = 1$$

$$s(3) = |X_1| + 1 + s(2) = 5$$

$$X_3 = X_1 \cdot x_5 = a_1 a_1 = 00$$



$$(a_1, a_1 a_2, a_1 a_1) = x_1^5$$

ワイル(Wyle)符号

- ランレンクス(run length)符号の代表例
スキャナ・FAXなどに利用

(例) 文書(白黒)をスキャンした信号
→ 白=0, 黒=1の情報として表す。

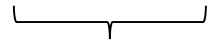
時間的に1行目、2行目、...の順にそれぞれ左から右に扱われる。

(特徴)

白=0の割合が圧倒的に大きく、それが連続して発生。

符号化(1)

- 2値情報源系列を0のラン(1つのシンボルの連続)のパターン $x_n = 00 \cdots 01$ の連続に分解し、 x_n を符号化。



n ビット

- ① $1 \leq n \leq 4$ のとき、

$n-1$ の2進表現を $d_1 d_2 = [n-1]_2$ として、

x_n の符号語 $y(x_n)$ を

$$y(x_n) = 0d_1d_2$$

とする。

(注) $0 \leq n-1 \leq 3$ なので、2ビットで表現できる。

符号化(2)

② $n \geq 5$ のとき、

n は $k \geq 2$ を適当に選ぶと、

$$n = 2^k + l \quad (1 \leq l \leq 2^k)$$

のように一意に表される。

→ $l - 1 = n - 2^k - 1$ の2進表現を

$$d_1 d_2 \cdots d_k = [l - 1]_2 \text{ として}$$

$$y(x_n) = \underbrace{11 \cdots 10}_{k-1 \text{ ビット}} \underbrace{d_1 d_2 \cdots d_k}_{k \text{ ビット}}$$

(注) $0 \leq l - 1 \leq 2^k - 1$ なので、 k ビットで表現できる。

符号化例

n	$y(x_n)$
1	0 0 0
2	0 0 1
3	0 1 0
4	0 1 1

n	$y(x_n)$
$17 = 2^4 + 1$	1 1 1 0 0 0 0 0
$18 = 2^4 + 2$	1 1 1 0 0 0 0 1
.....	
$32 = 2^4 + 16$	1 1 1 0 1 1 1 1

$5 = 2^2 + 1$	1 0 0 0
$6 = 2^2 + 2$	1 0 0 1
$7 = 2^2 + 3$	1 0 1 0
$8 = 2^2 + 4$	1 0 1 1

情報系列 x

0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1
 x_2 x_3 x_{10} x_1 x_6

$9 = 2^3 + 1$	1 1 0 0 0 0
$10 = 2^3 + 2$	1 1 0 0 0 1
.....	
$16 = 2^3 + 8$	1 1 0 1 1 1

符号化系列 y

0 0 1 0 1 0 1 1 0 0 0 1 0 0 0 1 0 0 1
 | | | | | | | | | | | | | | | |

復号化例

● 符号化系列 y

0010101100010001001 の復号

- ① y の最初のビットが0なので、前ページ符号対応表より
符号長3 ($n = 1 \sim 4$)であることがわかる。

→ “001” → 表より、 $x_2 = 01$

- ② y から最初の001を除いた0101100010001001
を見ると、最初のビットが0なので、符号長3。

→ “010” → 表より、 $x_3 = 001$

- ③ 残った系列 1100010001001
コンマ符号の部分として “110” が得られる。

→ 符号語長さ6

→ “110001” → x_{10}

- ④ 残った系列 0001001 → コンマ符号の部分として “0”

→ 符号語長さ3 → “000” → x_1

まとめ

ジブ・レンペル(ZL符号)符号

- (特徴)文字列の増分分解
セグメントと呼ばれる部分文字列の接続に分解。
各セグメントに対して符号化を行い、無歪圧縮を実現。

ワイル(Wyle)符号

- ランレンクス(run length)符号の代表例