## 1.1 定義と例

情報を出している元を**情報源** (source) といい, $\mathcal{S}$ で表される. 例えば, 計算機や文字を出力するディスプレイなどが情報源となる. 情報源は何らかのシンボル $X_n$(例えばバイナリ値なら 0 や 1) の列 $\boldsymbol{s} = X_1X_2\cdots$ ($X_n$ がバイナリ値なら $\boldsymbol{s} = 01$ や $\boldsymbol{s} = 0011000\cdots$ など) を出力する. このシンボルの列はバイナリ列などのように何らかの列であっても良いし, $n$ 回目の試行結果でも良い. また, 列は長さが有限であっても, 無限であってもかまわない. 情報源 $\mathcal{S}$ の出力するシンボルの有限集合を**情報源 $\mathcal{S}$ のアルファベット** (source alphabet) といい, その集合は $S = \{s_1, \cdots, s_q\}$ と表される. すなわち, $X_n \in S$ である.

シンボル列 $\boldsymbol{s}$ の $n$ 番目のシンボル $X_n \in S$ が $s_i \in S$ である確率は出力される順番に依らないと仮定する. つまり,

$$P(X_n = s_i) = p_i \ \ (i = 1, \cdots, q)$$

である. 式を見ればわかるように, $p_i$ はシンボル毎に異っていてもよいが, ここでは時間に依存せず一定であると仮定する. このように確率が時間に依らずに一定となる情報源を**定常情報源** (stationary source) という. また, 情報源が出力するシンボルは過去のシンボル $X_m$ $(m < n)$ に依存しないと仮定しているが, このようなシンボルの出力が過去のシンボルに依存しないような情報源を**記憶のない情報源** (memoryless source) という.

**例 1.** どの目もまったく同じ確率で出るサイコロを情報源 $\mathcal{S}$ とすると, そのアルファベットは $S = \{1, \cdots, 6\}$ となる. サイコロを $n$ 回目に振った時に各々の目がでる確率は $1/6$.

**例 2.** 情報源を *1* 冊の英語で書かれた本とし, アルファベットを英字のアルファベットとする. この時, 単語に出現するアルファベットは前のアルファベットに依存しているので, 本は記憶のない情報源とはならない *(*たとえば *"queen"* などのように *"q"* の後ろには *"u"* が続いて出ることが多い*).*

情報源から出力されるシンボルを符号化したシンボル $t_j$ を**符号シンボル** (code symbol) という. この符号シンボルから構成される有限集合を**符号アルファベット** (code alphabet) といい, $T = \{t_1, \cdots, t_r\}$ と表される. 符号化対象のアルファベットと符号アルファベットの個数が一致するとは限らないことに注意. つまり, 多く一部の情報を同じ符号シンボルに割り当てても良いし, 符号アルファベットの濃度の方が符号化対象のアルファベットよりもおおくても構わない ($r > q$). これは符号アルファベットは情報源よりもむしろ伝送する通信路の都合に依存することに依るためである. 符号アルファベットの要素数 $r$ を

**基数** (radix) といい, $r$ 個のシンボル $t_j$ からなる符号アルファベット $T$ を $r$ 元符号という. 特に $r = 2$ を **2 元符号** (binary code) という. ASCII コードなどは 0 と 1 を符号シンボルとした 2 元符号で英字を符号化している. また, モールス信号はトンとツー, それに空白を合わせた **3 元符号** (ternary code) となる.

情報源 $\mathcal{S}$ を符号化するには**符号語** (符号シンボルの有限列) を各シンボル $s_i \in S$ に割り当てることで行われる. たとえば, $\boldsymbol{s} = X_1X_2\cdots$ の各 $X_n = s_i$ に符号語 $w_i$ を割り当てて, 符号アルファベット $T$ のシンボル列 $\boldsymbol{t}$ を得るというようにする.

**例 3.** 情報源 $\mathcal{S}$ として, 出目の確率が全て同じとなるサイコロを考える. 符号アルファベットを $T = \{0, 1\}$ として, $w_i$ を $S = \{1, \cdots, 6\}$ の元 $s_i \in S$ を 2 進数で表記したものとする. すなわち, $w_1 = 1, w_1 = 10, \cdots, w_6 = 110$ というように符号化する. すると, サイコロを 5 回振ったときに得られたシンボル列が $\boldsymbol{s} = 53214$ であれば, $\boldsymbol{t} = 10111101100$ と符号化される.

なお, この本ではわかりやすさのために, 符号語の間に "."(ドット) を入れることがある. たとえば, 上記の例の符号化列では $\boldsymbol{t} = 101.11.10.1.100$ のように表記されることもあることに注意. ただし, "." そのものは符号アルファベットに加えられていないので, あくまで見易さのために挿入されている.

さて, もう少し厳密に符号を定義しなおす. まず, **語** (word) $w$ とは符号アルファベット $T$ の有限シンボル列をさし, その $w$ を構成しているシンボルの数を $w$ の**長さ**あるいは**語長** (word length) $|w|$ という. 上のサイコロの例であれば, $|w_6| = 3$ である. 語全体の集合を $T^*$ であらわす. この集合は長さが 0 である**空語** (empty word)$\epsilon$ を含むことに注意. 空語を除いたものは $T^+$ と表される. すなわち, 長さ $n$ の語の集合を $T^n = T \times \cdots \times T$($T$ の $n$ 個の直積) として,

$$T^* = \bigcup_{n\geq 0} T^n, T^+ = \bigcup_{n>0} T^n$$

と表される. 情報源の出力するアルファベットを符号化するには**情報源符号** (source code, または単に**符号** (code)) $\mathcal{C} : S \to T^+$ という関数を利用して行われる. すなわち, シンボル $s_i \in S$ に対して, $w_i = \mathcal{C}(s_i)$ のように語 $w_i \in T^+$ が割り当てられる. 符号の性質の多くは符号語のみに依存し, シンボルと符号語の特定の対応のさせ方に依存しないので, しばしば関数 $\mathcal{C}$ を単に $T^+$ の元である語の有限集合 $\{w_1, \cdots w_q\}$ と見做す. 同様の手順で $S^*$ と $T^*$ にたいしても拡張可能である. すなわち,

$$\boldsymbol{s} = s_{i_1}s_{i_2}\cdots s_{i_n} \mapsto \boldsymbol{t} = w_{i_1}w_{i_2}\cdots w_{i_n}$$

のようになる．この関数の像は

$$\mathcal{C}^* = \{w_{i_1}w_{i_2}\cdots w_{i_n} \in T^* : w_{i_j} \in \mathcal{C}, n \geq 0\}$$

となる．シンボル $s_i \in S$ の生起確率 $p_i$ を用いて, $\mathcal{C}$ の**平均符号長** (average code-word-length) は

$$L(\mathcal{C}) = \sum_{i=1}^{q} p_i|w_i|$$

とあらわされる．

**例 4.** 上のサイコロの例は符号が $\mathcal{C} = \{1, 10, 11, 100, 101, 110\}$ となるので, 平均符号長は

$$L(\mathcal{C}) = \frac{1}{6}(1 + 2 \times 2 + 3 \times 3) = \frac{7}{3}$$

第 1 章の続きでは, 条件

1. 容易で曖昧さのない複合 $\boldsymbol{t} \mapsto \boldsymbol{s}$ がある
2. 平均符号長 $L(\mathcal{C})$ がなるべく小さい

を満たす符号 $\mathcal{C}$ を構成することを目指す．

## 1.2 一意複合可能な符号

関数 $\mathcal{C}$ が単射のとき, $\mathcal{C}$ は**一意復号可能** (uniquely decodable; u.d.) であるという．こうすれば, 復号時に複合先が被るということはなくなる．

**例 5.** 上の例のサイコロの出目を *2* 進数で符号化する方法は一意復号可能ではない．たとえば, $\boldsymbol{t} = 11$ は 1.1 $(\boldsymbol{s} = 11)$ にも 11 $(\boldsymbol{s} = 3)$ にも復号できる．これを解決するには, 情報源シンボルを *3* 桁の *2* 進数であらわすようにすればよい．すなわち,

$$1 \mapsto 001, 2 \mapsto 010, \cdots, 6 \mapsto 110$$

のようにすればよい．

より一般には次のような定理が成立する．

**定理 1.** 符号 $\mathcal{C}$ に含まれる符号語 $w_i$ の長さが全て同じであれば, $\mathcal{C}$ は一意復号可能．

*Proof.* $l = |w_1| = \cdots = |w_q|$ とする. ある $\boldsymbol{t} \in C^*$ が, $u_1 \cdots u_m = v_1 \cdots v_n \ (u_i, v_j \in \mathcal{C})$ と分解されるならば, $lm = |\boldsymbol{t}| = ln \iff m = n$. $u_1$ と $v_1$ は $\boldsymbol{t}$ の最初の $l$ 個のシンボルのうち最初の語のため, $u_1 = v_1$. 以下同様にして $u_i = v_i \ (i = 1, \cdots, n)$ □

このような $\mathcal{C}$ の符号語の長さが全て同じ $|w|$ であるとき, $\mathcal{C}$ を長さ $|w|$ の**ブロック符号** (block code) という. これについては第 5 章以降触れられる.

実はブロック符号だけでなくとも一意復号可能な符号は存在する.

**例 6.** *2* 元符号 $\mathcal{C}$ が

$$s_1 \mapsto w_1 = 0, s_2 \mapsto w_2 = 01, s_3 \mapsto w_3 = 011$$

となっているとき, 一意復号可能となる. これは $0 \in T$ が区切り文字として機能していることに起因している.

このように符号長は一定でなくても良いので, 以降は一意復号可能性の必要十分条件について考えていく.

まず, 準備のためにいくつか定義を行う. 全ての $n \in \mathcal{N}$ に対し, $C_n \subseteq T^+$ となるように, 空語でない語からなる集合の列 $\mathcal{C}_0, \mathcal{C}_1, \cdots$ を

$$\mathcal{C}_n = \begin{cases} \mathcal{C} & \text{if } n = 0 \\ \{w \in T^+ : uw = v \ (u \in \mathcal{C}, v \in \mathcal{C}_{n-1} \text{または} u \in \mathcal{C}_{n-1}, v \in \mathcal{C})\} & \text{if } n \geq 1 \end{cases}$$

とする. また,

$$\mathcal{C}_\infty = \bigcup_{n=1}^{\infty} C_n$$

とする. この集合は $\mathcal{C}_{n-1} = \emptyset$ ならば, $\mathcal{C}_n = \emptyset$ となること,

$$\mathcal{C}_1 = \{w \in T^+ : uw = v \ (u, w \in \mathcal{C})\}$$

となることと, 符号長が一定である必要はないというところに注意すればわかりやすいかもしれない.

**例 7.** $\mathcal{C} = \{0, 01, 011\}$ とする. すると, 上の $C_1$ の式のうち, $u = 0$ とすれば, $\mathcal{C}_1 = \{1, 11\}$ となることがわかる. $n = 2$ とすると,

$$\mathcal{C}_2 = \{w \in T^+ : uw = v \ (u \in \mathcal{C}, v \in \mathcal{C}_1 \text{または} u \in \mathcal{C}_1, v \in \mathcal{C})\}$$

となるが, 制約式のどちらか一方には 0 が含まれ, どちらか一方には 0 が含まれていないことから, $\mathcal{C}_2 = \emptyset$ となる. よって, $\mathcal{C}_\infty = \mathcal{C}_1 = \{1, 11\}$ である.

この $\mathcal{C}_\infty$ は, 実は有限回の操作で求めることができる.

**補題 1.** $\mathcal{C} = \{w_1, \cdots, w_q\}$ とする. ある $n$ に対し, $w \in \mathcal{C}_n$ であるとき, $|w| \leq \max\{|w_1|, \cdots, |w_q|\}$. さらに, ここから, $\mathcal{C}_n$ は有限集合であり, 集合 $\mathcal{C}_0, \mathcal{C}_1, \cdots$ は, 最終的に周期的となる.

*Proof.* まず, $w \in \mathcal{C}_n$ に対し, $|w| \leq \max\{|w_1|, \cdots, |w_q|\}$ となることを帰納法を用いて示す.

**$n = 0$ のとき** $\mathcal{C}_0 = \mathcal{C}$ より, 明らかに $|w| \leq \max\{|w_1|, \cdots, |w_q|\}$.

**$n > 0$ のとき** $k \geq 0$ で命題が成立すると仮定する. $k+1$ のとき,

$$\mathcal{C}_{k+1} = \{w \in T^+ : uw = v \ (u \in \mathcal{C}, v \in \mathcal{C}_k \text{または} u \in \mathcal{C}_k, v \in \mathcal{C}\}$$

となる. $v \in \mathcal{C}$ の場合は

$$|uw| = |u| + |w| = |v| \leq \max\{|w_1|, \cdots, |w_q|\}$$

となる. また, $v \in \mathcal{C}_k$ のときは $\mathcal{C}_k$ でも命題が成立しているのでことに注意すれば, $|v| \leq \max\{|w_1|, \cdots, |w_q|\}$. あとは同様にして $|w| \leq \max\{|w_1|, \cdots, |w_q|\}$ を証明できる.

次に, $\mathcal{C}_n$ が有限集合であることを示す. $\mathcal{C}_n$ の語の長さは最大で $l := \max\{|w_1|, \cdots, |w_q|\}$ であるので, 符号アルファベットの数が $|T| = r$ であれば, その並びかたの分だけある. したがって, 空語を除くと, 最大で

$$N := \sum_{i=1}^{l} r^i = \frac{r(r^l - 1)}{r - 1}$$

の濃度しかない. よって, $\mathcal{C}_n$ は有限集合となる.

最後に周期的になるということの証明を行う. 符号シンボルの並べ方は $2^N$ 通りしかないことから, $\mathcal{C}_0$ から $\mathcal{C}_{2^N}$ の間に必ず $C_i = C_j \ (0 \leq i < j \leq 2^N)$ となるところがある. また, $\mathcal{C}_n$ の定義は再帰的になっているので, かならず周期的となることがわかる. □

この補題と, $\mathcal{C}_\infty$ の定義から, 周期的になった部分は $\mathcal{C}_n$ が集合であることから, 周期の 2 週目が始まるようなある $N > 1$ があって, その $N$ 以降は $\mathcal{C}_\infty$ への語の追加は起こらないので,

$$\mathcal{C}_\infty = \bigcup_{n=0}^{\infty} \mathcal{C}_1 = \bigcup_{n=1}^{N-1} \mathcal{C}_n$$

となる. したがって, $\mathcal{C}_\infty$ は有限回の操作で求められることがわかる.

この $\mathcal{C}_\infty$ を用いると, 次のような定理が示せる.

サーディナス-パターソンの定理

**定理 2.** 符号 $\mathcal{C}$ が一意復号可能であるための必要十分条件は, $\mathcal{C} \cap \mathcal{C}_\infty = \emptyset$ である.

証明は次回.

**例 8.** *2* 元符号 $\mathcal{C}$ が

$$s_1 \mapsto w_1 = 0, s_2 \mapsto w_2 = 01, s_3 \mapsto w_3 = 011$$

の場合を考える. これは上の例で見たが, 一意復号可能であった. また, $\mathcal{C}_1 = \{1, 11\}$, $\mathcal{C}_2 = \emptyset$ だったとを思いだそう. よって, $\mathcal{C}_\infty = \{1, 11\}$ から $\mathcal{C} \cap \mathcal{C}_\infty = \emptyset$.

**例 9.** *3* 元符号 $\mathcal{C} = \{01, 1, 2, 210\}$ を考える. すると,

$$\mathcal{C}_1 = \{10\}, \mathcal{C}_2 = \{0\}, \mathcal{C}_3 = \{1\}$$

であって, $\mathcal{C} \cap \mathcal{C}_\infty = \{1\}$. よってこの符号は一意復号可能ではない. 実際, 符号列 $\boldsymbol{t} = 2101$ という列は一意に復号できない. なぜならば, 210.1 と 2.1.01 の *2* つの意味にとることができてしまう.

さらに, 無限列に対しても一意復号可能性を考えたいが, これに対するより強い条件を考えることができる. 実際, Even[1] や Levenshtein[2], Riley[3] による定理はある $n \leq 1$ に対し, $\mathcal{C}_n = \emptyset$ かつ $\mathcal{C} \cap \mathcal{C}_\infty = \emptyset$ となるとき, かつ, そのときのみ一意復号可能であることを示している (この条件は 2 つの符号列が最初の $d$ 個のシンボルが一致するならば, 最初の符号語が同じであるということが成立するような定数 $d$ が存在するということを意味する**有限幅遅延で一意復号可能** (uniquely decodable with bounded delay) のための必要条件でもある. よって復号自体は多くとも $d$ 個のシンボル分の遅延のあとから始めることができるということが示されている).

# 参考文献


[1] Shimon Even. “Tests for Unique Decipherability”. In: *IEEE Transactions on Information Theory* 9 (2 1963), pp. 109–112 (cit. on p. 6).

[2] V. I. Levenshtein. “Some Properties of Coding and Self-Adjusting Automata for Decoding Messages”. In: *Problemy Kibernet* 11 (1964), pp. 63–121 (cit. on p. 6).

[3] J. A. Riley. “The Sardinas-Patterson and Levenshtein theorems”. In: *Information and Control* 10 (1967), pp. 120–136 (cit. on p. 6).