■1 群（信号・システム）- 1 編（情報理論）

6 章 ユニバーサル符号

【本章の構成】

本章では，以下について解説する．

6-1 統計的手法
6-2 統計量を用いないユニバーサル符号
6-3 整数の符号化
6-4 コルモゴロフ複雑度とユニバーサル符号

■1 群 - 1 編 - 6 章

6-1 統計的手法

（執筆者：有村光晴）[2012 年 10 月 受領]

本節では統計的推定に基づくユニバーサル符号について述べる．これらの符号は，情報源を定常無記憶情報源もしくはマルコフ情報源と仮定し，各シンボルの出現頻度から統計的に推定した確率を用いて算術符号化を行う．よって，確率を如何に推定するかが重要である．

本章で述べる符号では，(1) 2 段階符号，(2) ベイズ符号及び重み付け法，そして (3) 最尤符号と呼ばれる各種の符号が用いられている．

6-1-1 KT（Krichevsky-Trofimov）推定

KT 推定は，2 値の i.i.d. 系列の確率をパラメータに関する持つみます。重み付けで求めるものである．逐次算術符号と組み合わせると，定常無記憶情報源全体のクラスに対する逐次的な強ユニバーサル符号を実現できる．

2 値 i.i.d. 系列において 0 の出現する確率を θ と書き，このパラメータ θ が $\theta \in \Theta = [0,1]$ の範囲をとるような，定常無記憶情報源全体のクラスを考える．0 を $n(0)$ 個，1 を $n(1)$ 個 $(n(0)+n(1)=n)$ 含む系列 x^n が，パラメータが θ の情報源から生成される確率は $P_\theta(x^n) = \theta^{n(0)}(1-\theta)^{n(1)}$ と書ける．$\theta \in \Theta$ の確率分布を $(1/2,1/2)$-Dicichlet 分布としてこの確率を加重平均すると，KT 推定量[1, 2]

$$P_{KT}(n(0),n(1)) = \int_0^1 \frac{1}{\pi\sqrt{(1-\theta)\theta}}(1-\theta)^{n(0)}\theta^{n(1)}d\theta$$

が得られる．

まず，$P_{KT}(n(0),n(1))$ を用いた符号化アルゴリズムを述べる．確率 $P_{KT}(n(0),n(1))$ は $n(0) \geq 0, n(1) \geq 0$ を満たす任意の $n(0), n(1)$ に対して

$$P_{KT}(n(0)+1,n(1)) = \frac{n(0)+\frac{1}{2}}{n(0)+n(1)+1}P_{KT}(n(0),n(1)), \tag{6・1}$$

$$P_{KT}(n(0),n(1)+1) = \frac{n(1)+\frac{1}{2}}{n(0)+n(1)+1}P_{KT}(n(0),n(1)), \tag{6・2}$$

を満たし，また $P_{KT}(0,0)=1$ となるので，系列 x^n を x_1 から順に見ながら $P_{KT}(n(0),n(1))$ を逐次的に計算することが可能である．よって，$P_{KT}(n(0),n(1))$ を符号化確率に用いる逐次型算術符号を構成することが可能である．

次に，確率 $P_{KT}(n(0),n(1))$ を用いた Shannon 符号の性能を述べる．確率 $P_{KT}(n(0),n(1))$ は

$$P_{KT}(n(0),n(1)) \geq \frac{1}{2\sqrt{n(0)+n(1)}}\left(\frac{n(0)}{n(0)+n(1)}\right)^{n(0)}\left(\frac{n(1)}{n(0)+n(1)}\right)^{n(1)}$$

のように θ によらない下界を持つことから

$$\log\frac{(1-\theta)^{n(0)}\theta^{n(1)}}{P_{KT}(n(0),n(1))} \leq \frac{1}{2}\log n + 1$$

が成立する．系列の確率に KT 推定量を用いて Shannon 符号で x^n を符号化したときの符号語長は $\ell_{KT}(x^n) = \lceil -\log P_{KT}(n(0), n(1)) \rceil$ と書けるので，1 シンボル当たりの冗長度は，

$$\frac{1}{n}\left\{\ell_{KT}(x^n) - \log\frac{1}{P_\theta(x^n)}\right\} = \frac{1}{n}\left\{\left\lceil \log\frac{1}{P_{KT}(n(0), n(1))}\right\rceil - \log\frac{1}{(1-\theta)^{n(0)}\theta^{n(1)}}\right\}$$
$$\leq \frac{1}{2n}\log n + 2$$

のように，θ にも x^n にもよらない上界で押さえられる．以上で，KT 推定量を用いることで定常無記憶情報源のクラスに対するユニバーサル符号を構成することができた．

6-1-2 MDL 原理

Rissanen[3]によって，ユニバーサル符号化だけでなく情報源のモデル推定も対象とする方法論として提案されたのが MDL（Minimum Description Length；最小記述長）原理である．

Rissanen[4]は次の定理を示した．系列 x^n の確率が，k 次元のパラメータ $\theta = (\theta_1, \theta_2, \cdots, \theta_k) \in \Omega^k$ によって $P_\theta(x^n)$ と指定されるパラメトリックな定常情報源 X_θ^n の集合を考える．ただし，Ω^k は k 次元実数 R^k の部分集合のうち適当な条件を満たすものであり，更にパラメータ θ の最尤推定量 $\theta^*(x^n)$ が中心極限定理を満たすと仮定する．すると，各 n で Kraft の不等式を満たすような x^n の符号語長関数 $L(x^n)$ の確率分布 P_θ に関する期待値 $E_\theta[L(X_\theta^n)]$ は，すべての k と任意の $\varepsilon > 0$，及び $\theta \in \Omega^k \backslash A_\varepsilon(n)$ に対して

$$\frac{E_\theta[L(X_\theta^n)]}{n} \geq \frac{H(X_\theta^n)}{n} + \left(\frac{1}{2} - \varepsilon\right)\frac{k}{n}\log n$$

を満たす．ただし，$A_\varepsilon(n)$ は $n \to \infty$ のとき測度が 0 に収束するような θ の集合である．また，任意の $\varepsilon > 0$ と十分大きい $n \geq n(\varepsilon)$，すべての $\theta \in \Omega = \cup_k \Omega_k$ に対して

$$\frac{E_\theta[L(X_\theta^n)]}{n} < \frac{H(X_\theta^n)}{n} + \left(\frac{1}{2} + \varepsilon\right)\frac{k}{n}\log n$$

を満たすような，最適な符号語長関数 $L(x^n)$ が存在する．

これにより，系列 x^n を生成した情報源 X_θ^n を符号化の際に知らない場合でも，1 シンボル当たりの符号語長の期待値が，情報源のエントロピーレートに約 $(k/2n)\log n$ を加えた量となるような符号が存在することになる．Rissanen[4]はこの符号語長を達成する符号として次のような 2 段階符号を構築した．まず，x^n から k 及び θ を最尤推定する．次に k と θ を符号化すると，これらの値で指定される確率分布によって x^n が生成されたものと仮定して x^n を Shannon 符号で符号化することができる．これより，

$$\min_{\theta,k}\left\{\log\frac{1}{P_\theta(x^n)} + \frac{k}{2}\log n\right\} = \log\frac{1}{P_{\theta^*(x^n)}(x^n)} + \frac{k^*(x^n)}{2}\log n \qquad (6\cdot3)$$

を，情報源クラス Ω に関して系列 x^n に含まれる情報量として定義する．第 2 項 $(k^*(x^n)/2)\log n$ は，パラメータ $\theta \in \Omega^k$ の最尤推定量を記述するために必要なコストと考えることができるので，これを最適なモデルコストと呼ぶことにする．以上で，系列 x^n によってパラメータの次数 k 及びパラメータ θ を推定することができる．これが MDL 原理である．

6-1-3 Stochastic Complexity

系列 x^n のコルモゴロフ複雑度は，ユニバーサルなチューリング機械によって x^n を表現するためのプログラムの最短の長さで定義される．一方，式 (6･3) は x^n からそのモデルを推定するための基準であるが，これを系列 x^n を記述するための最小の長さとして解釈すると，系列 x^n の複雑度としてみることができる．これを，上記の 2 段階符号のみならずすべての符号にわたって最小化したものが確率的複雑度である．この値は Rissanen[5]によって次のように定義された．x^n が与えられたもとでの θ の最尤推定値を $\theta^*(x^n)$ とすると，x^n の最大尤度は $P_{\theta^*(x^n)}(x^n)$ で与えられる．この値を規格化した

$$\hat{m}_n(x^n) = \frac{P_{\theta^*(x^n)}(x^n)}{\sum_{x^n} P_{\theta^*(x^n)}(x^n)}$$

を符号化確率として用いる Shannon 符号を最尤符号と呼ぶ．その記述長は切り上げを除くと $\log(1/\hat{m}_n(x^n))$ と書ける．これを確率的複雑度として定義すると，

$$\log \frac{1}{\hat{m}_n(x^n)} = \log \frac{1}{P_{\theta^*(x^n)}(x^n)} + \frac{k}{2} \log \frac{n}{2\pi} + \log \int_\Theta \sqrt{\det I(\theta)} d\theta + o(1)$$

が成立する．ただし，$I(\theta)$ は θ の Fisher 情報行列であり，$o(1)$ は $n \to \infty$ のとき 0 に収束する量である．

6-1-4 CTW，PPM

文脈長に上界が存在するような情報源クラスにおいて，その上界が既知である場合のユニバーサル符号として代表的なものに CTW 法[2] と PPM 法[6] が存在する．両者ともその後，文脈長に上界が存在しないような情報源クラスに対応できるよう拡張されている．

(1) CTW 法

Context Tree Weighting（CTW）法は，Willems ら[2]によって 1995 年に提案された．2 値系列に対する深さ D の文脈 2 分木 $\mathcal{T}_D$ を考える．節点に対するラベル s を，文脈木の根から節点までの長さ $|s|$ の経路とする（節点 s とそのラベル s を同一視する）．深さ m の節点のラベルは長さ m の文脈に対応する．木の根に対するラベルは長さ 0 の文脈 λ であり，節点 s の 2 つの子節点は文脈 $0s$ 及び $1s$ に対応する．

文脈 s に対して，x^n 内で $s0$ 及び $s1$ が出現した（すなわち，文脈 s の次にシンボル 0 及び 1 が出現した）頻度をそれぞれ a_s, b_s と書くと，s の子節点 $0s$ における頻度 a_{0s}, b_{0s} と $1s$ における頻度 a_{1s}, b_{1s} は $a_{0s} + a_{1s} = a_s$ 及び $b_{0s} + b_{1s} = b_s$ を満たす．

節点 s において，頻度 a_s, b_s から KT 推定確率 $P_{KT}(a_s, b_s)$ を求める．この値を

$$P_w^s = \begin{cases} \dfrac{P_{KT}(a_s, b_s) + P_w^{0s} P_w^{1s}}{2}, & \text{if } |s| < D, \\ P_{KT}(a_s, b_s), & \text{if } |s| = D, \end{cases}$$

のように再帰的に重み付けし，木の根において求まる重み付き確率 P_w^λ を系列の出現確率の推定値 $P^*(x^n)$ とし，$P^*(x^n)$ を用いて算術符号を行う．

x_1 に対する文脈 x_{-D+1}^0 が与えられているという条件のもとでの x^n に対するこの符号の符

号語長を $L(x^n|x_{-D+1}^0)$ と書くと，個別冗長度は

$$L(x^n|x_{-D+1}^0) - \log \frac{1}{P(x^n|x_{-D+1}^0)} < \Gamma_D(\mathcal{S}) + |\mathcal{S}| \cdot \gamma(\frac{n}{|\mathcal{S}|}) + 2$$

のように，個別の系列 x^n にも情報源の確率にも依存しない一様な上界で押さえられる．ただし，$\Gamma_D(\mathcal{S})$ は $\Gamma_D(\mathcal{S}) = |\mathcal{S}| - 1 + |\{s : s \in \mathcal{S}, |s| < D\}|$ で，$\gamma(\cdot)$ は

$$\gamma(z) = \begin{cases} z, & \text{for } 0 \le z < 1, \\ \frac{1}{2}\log z + 1, & \text{for } z \ge 1 \end{cases}$$

で与えられる．

(2) PPM 法

Prediction by Partial Matching（PPM）法は，Cleary と Witten[6]によって 1984 年に提案された．系列 $x^n = x_1 x_2 \cdots x_n$ のうち x^{i-1} の符号化が終了した時点で，$x_i = \varphi$ を符号化する際の確率 $p(x_i) = p(\varphi)$ の推定を，x^{i-1} を用いて行うことにする．このとき PPM 法では，過去に x_i と同じシンボルが出現した，なるべく長い文脈を用いて，今回の x_i の確率を推定する．

マルコフモデルの最大次数を o とする．$0 \le m \le o$ に対して，x^{i-1} の中で長さ m の文脈の後に文字 φ が出現した回数を $c_m(\varphi)$ とする．文脈に関わらずシンボル自体が出現しない（すなわち，すべての m に対して $c_m(\varphi) = 0$ となる) 場合のために $m = -1$ を準備する．$m = -1$ ではすべての φ に対して $c_{-1}(\varphi) = 1$ と設定しておく．

m 次のモデルで予測されるが，それより高い次数では予測されない文字の集合を A_m とする．集合 A_m に含まれる文字は $A_m = \{\varphi : c_m(\varphi) > 0\} - \bigcup_{l=m+1}^{o} A_l$ と書かれる．文字 φ に対して m 次で推定される確率は $p_m(\varphi) = c_m(\varphi)/(1 + C_m)$ で与えられる．ただし，C_m は m 次のモデルで予測される文字の出現頻度の合計 $C_m = \sum_{\varphi \in A_m} c_m(\varphi)$ である．よって，次数 m において初めて出現する文字に対するエスケープ確率は $e_m = 1 - \sum_{\varphi \in A_m} p_m(\varphi) = 1/(1 + C_m)$ で与えられる．最終的に，PPM におけるシンボルの推定確率は $p(\varphi) = p_m(\varphi) \prod_{l=m+1}^{o} e_l$ で与えられる．

$p(\varphi)$ の計算の手続きは最高次数 o で開始される．文字 φ に対する $p_m(\varphi)$ が 0 である間，次数を減らしながら各次数のエスケープ確率 e_m を掛け合わせる．最後に，$p_m(\varphi) > 0$ となる最初の m における $p_m(\varphi)$ を掛けたものが $p(\varphi)$ となる．

$p_m(\varphi)$ 及び e_m の計算方法は，上の式以外にも様々なバリエーションが存在する．

■参考文献

1) Raphail E. Krichevsky and Victor K. Trofimov：“The Performance of Universal Encoding,” IEEE Transactions on Information Theory, vol.IT-27, no.2, pp.199–207, Mar. 1981.
2) Frans M. J. Willems, Yuri M. Shtarkov, and Tjalling J. Tjalkens：“The Context-Tree Weighting Method: Basic Properties,” IEEE Transactions on Information Theory, vol.41, no.3, pp.653–664, May 1995.
3) Jorma Rissanen：“Modeling by Shortest Data Description,” Automatica, vol.14, pp.465–471, 1978.
4) Jorma Rissanen：“Universal Coding, Information, Prediction and Estimation,” IEEE Transactions on Information Theory, vol.IT-30, no.4, pp.629–636, Jul. 1984.
5) Jorma Rissanen：“Fisher Information and Stochastic Complexity,” IEEE Transactions on Information Theory, vol.42, no.1, pp.40–47, Jan. 1996.

6) John G. Cleary and Ian H. Witten：“Data Compression Using Adaptive Coding and Partial String Matching,” IEEE Transactions on Communications, vol.COM-32, no.4, pp.396–402, Apr. 1984.

■1 群 - 1 編 - 6 章

6-2 統計量を用いないユニバーサル符号

（執筆者：有村光晴）[2012 年 10 月 受領]

6-2-1 辞書法及びその符号化定理

(1) LZ77 符号

LZ77 符号は，1977 年に Ziv と Lempel[1]によって提案されたアルゴリズムであり，系列をブロックに分割して符号化する．パラメータとして 1 以上の整数 w が存在する．

系列 x^n のうち x^i $(i < n)$ の分割が終了していると仮定し，未分割部 x_{i+1}^n の先頭からブロックを切り出すことを考える．x^i の最後 w シンボル x_{i-w+1}^i をスライド窓と呼ぶ．スライド窓の中に先頭シンボルが存在する系列 x_{i-j+1}^n $(1 \le j \le w)$ と，未分割部 x_{i+1}^n の語頭が何シンボル一致しているかを調べる．ここで，最も長い一致となった j が j_{max} だったとし，一致長が k シンボルであるとする．このとき，未分割部からその $k+1$ シンボルの語頭 x_{i+1}^{i+k+1} を切り出して次のブロックとする．

切り出された各ブロックは，最大一致の先頭インデックス j_{max} と一致長 k，不一致シンボル x_{i+k+1} という 3 つの情報が符号化される．

この符号化アルゴリズムの若干異なるバリエーションについて，スライド窓サイズを大きくしていったときに平均符号語長がエントロピーに収束することが証明されている[2]．定常エルゴード情報源 $\boldsymbol{X} = X_1X_2\cdots$ のエントロピーレートを $H(\boldsymbol{X})$ とし，この情報源から生成された系列 x^n を窓サイズ w で符号化したときの符号語長を $\ell_{LZ77}(x^n)$ とすると，

$$\lim_{w\to\infty}\lim_{n\to\infty}\frac{E_{X^n}[\ell_{LZ77}(X^n)]}{n} = H(\boldsymbol{X})$$

が成立する．

(2) LZ78 符号

LZ78 符号は，1978 年に Ziv と Lempel[3]によって提案された．

この符号化では，単語が登録された外部辞書 D を保持し，辞書と系列の未分割部の語頭との間で最大一致する単語を探し，未分割部の語頭を切り出して符号化を行う．

符号化に先立って，$D = \{\}$ としておく．系列 x^n のうち x^i $(i < n)$ の分割が終了していると仮定し，未分割部 x_{i+1}^n の先頭からブロックを切り出すことを考える．辞書に含まれる単語 d_j と未分割部 x_{i+1}^n の語頭が何シンボル一致しているか調べる．ここで，最も長い一致となった j が j_{max} であるとし，一致長が k シンボルであったとする．このとき，未分割部 x_{i+1}^n からその $k+1$ シンボルの語頭 x_{i+1}^{i+k+1} を切り出して次のブロックとする．また，x_{i+1}^{i+k+1} を辞書 D に追加する．

切り出された各ブロックに対して，最大一致のインデックス j_{max} と不一致シンボル x_{i+k+1} が符号化される．

この符号化アルゴリズムについては，オリジナルの論文[3]において，その符号語長がエントロピーレートに概収束することが証明されている．エントロピーレートが $H(\boldsymbol{X})$ であるような定常エルゴード情報源 $\boldsymbol{X} = X_1X_2\cdots$ から生成された系列 x^∞ の先頭 n シンボルを符号化したときの符号語長を $\ell_{LZ78}(x^\infty, n)$ とすると，

$$\Pr\left\{x^{\infty} : \lim_{n\to\infty} \frac{\ell_{LZ78}(x^{\infty}, n)}{n} = H(\boldsymbol{X})\right\} = 1$$

が成立する．

6-2-2 ソート法及びその符号化定理

(1) ブロックソート法

ブロックソート法は，Burrows と Wheeler[4]によって 1994 年に提案された．系列をブロックに区切った後，まず Burrows-Wheeler（BW）変換と呼ばれる変換で同じ長さのブロックに変換する．更に，ブロックを Recency Rank 符号化[5, 6, 7] によって同じ長さの正整数列に変換する．これを算術符号によって符号化するアルゴリズムである．

なお，bzip2 に代表されるこのアルゴリズムの実装の多くでは，Recency Rank 符号と算術符号の間にランレングス符号化が挟まっている．ここではランレングス符号を用いない符号化法について述べる．

ブロックソート法によって系列 x^n を符号化する際には，まず Burrows-Wheeler 変換（BW 変換）と呼ばれる変換によって系列 x^n を別の系列 y^n と整数 p に変換する．更に，系列 y^n を Recency Rank 変換（RR 変換）によって同じ長さの整数列 z^n に変換する．最後に，p と z^n を算術符号によってエントロピー符号化したものが x^n に対する符号となる．

まず BW 変換について述べる．系列 x^n が与えられたとするとき，その先頭シンボル x_1 を系列の最後に移動した $x_2^n x_1$ を x^n の左巡回シフトと呼び，$R(x^n)$ と書く．x^n に左巡回シフトを k 回繰り返して適用したものを $R^k(x^n)$ と書く．$n \times n$ の行列を作成し，k 行目に $R^{k-1}(x^n)$ を書く．次に，この行列の行集合を辞書順に整列する．整列された行列の一番右の列を上から順に取り出したものを y^n とし，整列後に x^n が存在する行を p とする．

次に，RR 変換によって系列 y^n を整数列 z^n に変換する．系列 x^n のアルファベットを $\mathcal{A}$，そのサイズを A とする．変換に先立って $\mathcal{A}$ に含まれるシンボルをアルファベット順に並べたものを L_0 とする．各 $i = 1, \ldots, n$ に対して，L_{i-1} の中でシンボル y_i が先頭から z_i 番目にあったとする．このシンボルを先頭に移動したものを L_i とする．以上で，系列 y^n が整数列 z^n に符号化される．

p は 1 以上 A 以下の整数，z^n は同じく 1 以上 A 以下の整数列なので，これを連接した pz^n を算術符号化すればよい．

(2) 文脈ソート法

文脈ソート法は，横尾と高橋[8]によって 1996 年に提案された．ブロックソート法の逐次アルゴリズム版として見ることが可能である．

この符号化アルゴリズムでは，各シンボルに対して逐次的に整数値のランクを割り当て，ランクの値を符号化する．例えば，x^n =abrabr のうち x^j =abrab の符号化が終了していると仮定する．まず，各シンボルとその文脈の組 (x_j, x_1^{j-1}) を $j = 1, 2, \ldots, i$ に対して作成したものと，次に符号化するシンボルを？とし，現在の文脈と組にしたもの (?, abrabr) をすべて書き出すと

$$(\mathrm{a}, \lambda), (\mathrm{b}, \mathrm{a}), (\mathrm{r}, \mathrm{ab}), (\mathrm{a}, \mathrm{abr}), (\mathrm{b}, \mathrm{abra}), (?, \mathrm{abrab})$$

となる．ただし，λ は空列を表す．次に，文脈を右から見ながら辞書順にソートすると，

$$1:(\mathrm{a},\lambda), 2:(\mathrm{b},\mathrm{a}), 3:(\mathrm{b},\mathrm{abra}), 4:(\mathrm{r},\mathrm{ab}), 5:(?,\mathrm{abrab}), 6:(\mathrm{a},\mathrm{abr}),$$

となる．次に，現在の文脈の位置が p であるとき，文脈の類似度順を反映するように $p+1$, $p-1$, $p+2$, $p-2$, ... の順に並べ換えると

$$1:(\mathrm{a},\mathrm{abr}), 2:(\mathrm{r},\mathrm{ab}), 3:(\mathrm{b},\mathrm{abra}), 4:(\mathrm{b},\mathrm{a}), 5:(\mathrm{a},\lambda)$$

となる．最後に，シンボル x_j の出現順にランクを振ると，シンボル a, r, b のランクがそれぞれ 1, 2, 3 となる．実際に次に出現するシンボル x_{i+1} は r なので，2 という値に変換される．

このような手続きにより，類似した文脈の次に出現するシンボルは偏っているという性質を用いて系列をシンボル単位で符号化する．

6-2-3 文法法及びその符号化定理

(1) MPM

MPM は，Kieffer ら[12]によって 2000 年に提案された．系列を 2 のべき乗のブロックに再帰的に分割して，各レベルでパターン一致を見つけて符号化を行う．

最大レベルを L とし，系列長 n が 2^L の倍数であると仮定する．系列 x^n を長さ 2^L ずつのブロック $w_L(1), w_L(2), \ldots$, に区切って並べたものをレベル L のブロック列とする．まず，レベル L の各ブロック $w_L(i)$ において，左側に同じブロック $w_L(j)$ $(j<i)$ が存在しないもの $w_L^1, w_L^2, \ldots$ を取り出す．次に，各ブロック $w_L(i)$ で，左側に同じブロック w_L^j が存在するものは，インデックス j をこのブロックに対する符号とする．

次に，初出ブロック列 $w_L^1, w_L^2, \ldots$ のそれぞれを 2 分割したものをレベル $L-1$ のブロック列 $w_{L-1}(1), w_{L-1}(2), \ldots$ とする．レベル $L-1$ に対してもレベル L と同様，初出ブロックとそれ以外のブロックに分けて，初出ブロック以外のブロックは初出ブロックに対するインデックスを符号とし，初出ブロックは更に 2 分割する．この操作をレベル 1 まで繰り返すと，レベル 1 の各ブロックの長さは 1 となる．

各ブロックに対するインデックスの符号化は，逆にレベル 1 からレベル L まで順に行うと効率的に符号化できる．

(2) 文脈自由文法を用いる符号

文脈自由文法を用いる符号，は Kieffer と Yang[10, 11] によって提案された符号のクラスである．MPM 法や LZ78 法は，このクラスの符号にてブロック分割の形に制限をかけたものとしてみることができる．また，Nevill-Manning と Witten[9]によって類似のアルゴリズムが提案されている．

文脈自由文法 $G=(V,T,P,S)$ は非終端記号（変数）V，終端記号の集合（アルファベット）T，生成規則 $P=\{p_i\}, p_i: V \to (V \cup T)^*$，開始記号 $S \in V$ から成る．符号化する系列 x^n のみを生成するような文法規則を作成する．

例えば，x^n = ababbabbb を生成するような文法規則は以下のようにして作ることができる．

1. 初期化として，与えられた x^n を右辺とする規則 $S \to$ ab<u>abb</u> <u>abbb</u> を作成する．

2. abb が S の右辺の 2 箇所に存在するので，これを変数 A に置き換えて，新しく規則 A を作成すると $S \rightarrow \mathsf{ab}A A\mathsf{b}$, $A \rightarrow \mathsf{abb}$ となる．

3. S と A の右辺に ab が存在するので，これを変数 B に置き換えて，新しく規則 B を作成すると $S \rightarrow BAA\mathsf{b}$, $A \rightarrow B\mathsf{b}$, $B \rightarrow \mathsf{ab}$ となる．

4. 右辺の変数が出現順に A_1, A_2 となるように変数の付け替えを行い，生成規則の順序を変えると，生成規則の集合は $S \rightarrow A_1A_2A_2\mathsf{b}$, $A_1 \rightarrow \mathsf{ab}$, $A_2 \rightarrow A_1\mathsf{b}$ となる．

以上で，系列 x^n を生成する文脈自由文法の生成規則ができた．この生成規則の右辺を上から順に並べると $A_1A_2A_2\mathsf{b}, \mathsf{ab}, A_1\mathsf{b}$ となるので，x^n の代わりにこの系列を（区切り文字のカンマも含めて）ベルヌーイ系列として算術符号化する．

■参考文献

1) Jacob Ziv and Abraham Lempel：“A Universal Algorithm for Sequential Data Compression,” IEEE Transactions on Information Theory, vol.IT-23, no.3, pp.337–343, May 1977.
2) Aaron D. Wyner and Jacob Ziv：“The Sliding-Window Lempel-Ziv Algorithm is Asymptotically Optimal,” Proceedings of the IEEE, vol.82, no.6, pp.872–877, Jun. 1994.
3) Jacob Ziv and Abraham Lempel：“Compression of Individual Sequences via Variable-Rate Coding,” IEEE Transactions on Information Theory, vol.IT-24, no.5, pp.530–536, Sep. 1978.
4) M. Burrows and D. J. Wheeler：“A Block-Sorting Lossless Data Compression Algorithm,” SRC Research Report 124, digital Systems Research Center, May 1994.
5) B. Ya. Ryabko：“Data Compression by Means of a “Book Stack”,” Problems of Information Transmission, pp.265–269, 1981, translated from Problemy Peredachi Informatsii, vol.16, no.4, pp.16–21, Oct.–Dec. 1980.
6) Jon Louis Bentley, Daniel D. Sleator, Robert E. Tarjan, and Victor K. Wei：“A Locally Adaptive Data Compression Scheme,” Communications of the ACM, vol.29, no.4, pp.320–330, Apr. 1986.
7) Peter Elias：“Interval and Recency-Rank Source Coding: Two On-Line Adaptive Variable-Length Schemes,” IEEE Transactions on Information Theory, vol.IT-33, no.1, pp.3–10, Jan. 1987.
8) Hidetoshi Yokoo and Masaharu Takahashi：“Data Compression by Context Sorting,” IEICE Transactions on Fundamentals, vol.E79-A, no.5, pp.681–686, May 1996.
9) Craig G. Nevill-Manning and Ian H. Witten：“Compression and Explanation Using Hierarchical Grammars,” Computer Journal, vol.40, no.2/3, pp.103–116, 1997.
10) John C. Kieffer and En-hui Yang：“Grammar-Based Codes: A New Class of Universal Lossless Source Codes,” IEEE Transactions on Information Theory, vol.46, no.3, pp.737–754, May 2000.
11) En-hui Yang and John C. Kieffer：“Efficient Universal Lossless Data Compression Algorithm Based on a Greedy Sequential Grammar Transform–Part I: Without Context Models,” IEEE Transactions on Information Theory, vol.46, no.3, pp.755–777, May 2000.
12) John C. Kieffer, En-hui Yang, Gregory J. Nelson, and Pamela Cosman：“Universal Lossless Compression via Multilevel Pattern Matching,” IEEE Transactions on Information Theory, vol.46, no.4, pp.1227–1245, Jul. 2000.

■1 群 - 1 編 - 6 章

6-3 整数の符号化

■1 群 - 1 編 - 6 章

6-4 コルモゴロフ複雑度とユニバーサル符号

（執筆者：葛岡成晃）[2009 年 11 月 受領]

コルモゴロフ複雑度は，系列の複雑さを測る一つの尺度である．ある系列 x のコルモゴロフ複雑度は，計算機が x を出力するために必要な最小の入力プログラムの長さとして定義される．コルモゴロフ複雑度に基づいた情報理論は，1960 年代にコルモゴロフ（Kolmogorov）[1]やソロモノフ（Solomonoff）[2]，チャイティン（Chaitin）[3]らによって創始され，現在，アルゴリズム情報理論と呼ばれている．

本節では，コルモゴロフ複雑度の定義と基本的な性質について説明した後，コルモゴロフ複雑度とエントロピーとの関係について述べる．そして，コルモゴロフ複雑度が，系列のランダム性を判定する問題やチューリング機械の停止判定問題と関係していることを説明する．最後に，与えられたデータが持つ構造をコルモゴロフ複雑度に基づいて抽出する，アルゴリズム情報理論的な考え方を紹介し，MDL（Minimum Description Length）原理などとの関係を説明する．なお，本節の内容の詳細については文献 4) を参照のこと．

6-4-1 コルモゴロフ複雑度の定義と性質

(1) 計算モデル

空列 ϵ を含む有限長の 2 進列全体の集合を $\{0,1\}^*$ で表す．なお，$\{0,1\}^*$ と非負整数の集合 $\mathcal{N}$ との間に自然な一対一対応を定めることができるので，これ以降 $\{0,1\}^*$ と $\mathcal{N}$ を同一視する．$\ell(x)$ で 2 進列 $x \in \{0,1\}^*$ の系列長を表す．2 つの 2 進列 $x, y \in \{0,1\}^*$ に対して xy で x と y の連接を表す．また，$\langle x,y\rangle$ で $\ell(x)$ 個の 1 の後に 0，x 及び y を連接した系列 $1^{\ell(x)}0xy$ を表す．

チューリング機械で計算可能な関数を帰納的関数（Recursive Function）という．また，実数値関数 f に対して，任意の $x \in \{0,1\}^*$ と $k \in \mathcal{N}$ ついて $|f(x) - g(x,k)| < 1/k$ となる帰納的関数 g が存在するとき，f は計算可能（Computable）という．ただし，ここでは $g(\langle x,k\rangle) = \langle p,q\rangle$ のとき g は $g(x,k) = p/q$ なる有理数関数であると考える．同様に，k に関して単調非減少でかつ $\lim_{k\to\infty} g(x,k) = f(x)$ となる帰納的関数 g が存在するとき f は下から半計算可能（Semi-computable from Below）という．$-f$ が下から半計算可能なとき f は上から半計算可能（Semi-computable from Above）という．

以下では，作業用テープのほかに一方向にのみ動く読み込み専用の入力用テープと書き込み専用の出力用テープとを持つチューリング機械のみを考える．すると，あるチューリング機械 T に対して，入力用テープから $p \in \{0,1\}^*$ を読み込んだ状態で T が停止するような 2 進列 p の全体は，語頭条件を満足することになる．T が停止した段階で，入力用テープから $p \in \{0,1\}^*$ を読み込んでおり，出力用テープに $x \in \{0,1\}^*$ を書き込んだ状態になっているとき，$T(p) = x$ と表す．

(2) 複雑度の定義

チューリング機械 T と 2 進列 $x, y \in \{0,1\}^*$ に対して $K_T(x|y) = \min\{\ell(p) : T(y,p) = x\}$ と定義する．ただし，$T(y,p) = T(\langle y,p\rangle)$ である．また，$T(y,p) = x$ となる $p \in \{0,1\}^*$ が存在しない場合には $K_T(x|y) = \infty$ と定義しておく．このとき，任意のチューリング機械 T と任意の $x, y \in \{0,1\}^*$ に対して $K_U(x|y) \le K_T(x|y) + c_T$（$c_T$ は x と y には依存しない定数）と

なる万能チューリング機械 U が存在する．そこで，万能チューリング機械 U を 1 つ固定し，条件付きコルモゴロフ複雑度 $K(x|y)$ を $K(x|y) = K_U(x|y)$ と定める．なお，2 つの万能チューリング機械 U と U' に対して，定数 $c_{UU'}$ が存在して任意の $x, y \in \{0,1\}^*$ に対して $|K_U(x|y) - K_{U'}(x|y)| \le c_{UU'}$ となるので，どの万能チューリング機械を用いて複雑度を定義しても，その差は高々定数で抑えられる．

2 進列 $x \in \{0,1\}^*$ のコルモゴロフ複雑度 $K(x)$ は，空列 ϵ を条件としたときの条件付きコルモゴロフ複雑度 $K(x|\epsilon)$ によって定義される．また，$K(x,y) = K(\langle x,y\rangle)$ とする．

(3) 複雑度の性質

以下では，x に依存しない定数 c が存在して $f(x) \le g(x) + c$ となるとき $f(x) \lesssim g(x)$ と表し，$|f(x) - g(x)| \le c$ となるとき $f(x) \approx g(x)$ と表す．すると，複雑度と系列長について $K(x|\ell(x)) \lesssim \ell(x)$ 及び $K(x) \lesssim \ell(x) + 2\log \ell(x)$ が成り立つ．また，$K(x|y) \lesssim K(x) \lesssim K(x,y)$ や，劣加法性 $K(x,y) \lesssim K(x) + K(y|x) \lesssim K(x) + K(y)$ 及び対称性 $K(x,y) \approx K(x) + K(y|x, K(x)) \approx K(x) + K(y|x^*) \approx K(y,x)$ などが成り立つ．ただし，x^* は x を出力する最短の入力（つまり $U(x^*) = x$ かつ $\ell(x^*) = K(x)$ なる 2 進列）である．更に，万能チューリング機械 U を停止させる入力の全体が語頭条件を満足することから，クラフトの不等式 $\sum_{x\in\{0,1\}^*} 2^{-K(x)} \le 1$ 及び $|\{x \in \{0,1\}^* : K(x) < k\}| < 2^k$ が成り立つ．

複雑度を $\mathcal{N}$ 上の関数として考えた場合，$K(n) \lesssim \log^* n = \log n + \log^{(2)}(n) + \cdots + \log^{(m(n))}(n)$ が成り立つ．ただし，$\log^{(k)}$ は log の k 回の合成関数であり，$m(n)$ は $\log^{(m)}(n) \ge 0$ となる最大の正整数 m を表す．また，$\sum_n 2^{-f(n)} = \infty$ なる関数 $f(n)$ に対して，$K(n) > f(n)$ となる n が無限個存在する．例えば，$K(n) > \log n$ が無限個の n に対して成り立つ．

なお，コルモゴロフ複雑度は計算不可能である．更に，無限個の $x \in \{0,1\}^*$ に対して $\phi(x)$ が定義され，かつ，$\phi(x)$ が定義されている x については $\phi(x) = K(x)$ となる帰納的部分関数 ϕ は存在しない．ただし，$K(x)$ は上から半計算可能ではある．

6-4-2 コルモゴロフ複雑度とエントロピー

コルモゴロフ複雑度の定義で用いた万能チューリング機械 U を用いて，$\{0,1\}^*$ 上の普遍確率（Universal Probability）P_U を $P_U(x) = \sum_{p:U(p)=x} 2^{-\ell(p)}$ と定義する．このとき，ある定数 c が存在して $2^{-K(x)} \le P_U(x) \le c2^{-K(x)}$ が成り立つ．すなわち，$K(x) \approx -\log P_U(x)$ となる．この関係は，コルモゴロフ複雑度 $K(x)$ が，分布 P_U に対するシャノン・ファノ符号の符号長 $-\log P_U(x)$ に対応することを示している．同様に，条件付き普遍確率（Conditional Universal Probability）を $P_U(x|y) = \sum_{p:U(y,p)=x} 2^{-\ell(p)}$ と定義すれば，$K(x|y) \approx -\log P_U(x|y)$ が成り立つ．

$\sum_{x\in\{0,1\}^*} P(x) = 1$ なる関数 $P\colon \{0,1\}^* \to [0,1]$ を $\{0,1\}^*$ 上の確率測度と呼ぶ．任意の計算可能な確率測度 P に対して，ある定数 c が存在して任意の $x \in \{0,1\}^*$ について $P_U(x) \ge cP(x)$ が成り立つ．また，P のエントロピーを $H(P) = -\sum_x P(x)\log P(x)$ と定めると，任意の計算可能な確率測度 P に対して，ある定数 c が存在して $H(P) \le \sum_x P(x)K(x) \le H(P) + c$ が成り立つ．すなわち，高々定数の差を除けば，コルモゴロフ複雑度の平均値はエントロピーと等しくなる．

同様の結果は確率過程に対しても示される．$f(x) = \mu(\{x\omega : \omega \in \{0,1\}^\infty\})$ $(x \in \{0,1\}^*)$ が計算可能である μ を確率測度として持つ確率過程 $(X_1, X_2, \dots)$ に対して，ある定数 c が存在して任意の正整数 n について $H(X_1, X_2, \dots, X_n) \le E_\mu[K(X_1, X_2, \dots, X_n|n)] \le H(X_1, X_2 \dots, X_n) + c$

が成り立つ[6]．ただし，E_μ は確率測度 μ に関する平均を表す．

6-4-3 コルモゴロフ複雑度とランダム性

(1) ランダム性と圧縮不可能性

計算可能な $\{0,1\}^\infty$ 上の確率測度 μ に対して，次の条件を満たす $\{0,1\}^\infty$ 上の関数 δ 考える．(1) 下から半計算可能な関数 γ を用いて，$\delta(\omega) = \sup_{n\in\mathcal{N}} \gamma(\omega_{1:n})$ と表せる．ただし，$\omega_{1:n}$ は $\omega \in \{0,1\}^\infty$ の先頭 n 文字を表す．(2) 任意の $m \geq 0$ に対して $\mu(\{\omega : \delta(\omega) \geq m\}) \leq 2^{-m}$．これら 2 つの条件を満たす関数 δ を逐次的 μ-検定（Sequential μ-test）と呼ぶ（$\delta(\omega) = \infty$ のとき検定 δ は ω を棄却するという）．このとき，ある逐次的 μ-検定 f が存在して，任意の逐次的 μ-検定 δ に対して $f(\omega) \geq \delta(\omega) - c_\delta$ となる（c_δ は ω によらない定数）．この f を万能逐次的 μ-検定（Universal Sequential μ-test）と呼び $\delta_0(\cdot|\mu)$ で表す．無限長の 2 進列 ω が万能逐次的 μ-検定に棄却されないとき，すなわち $\delta_0(\omega|\mu) < \infty$ であるとき，ω は μ-ランダムであるという．簡単に述べれば，μ に対する計算可能なすべての検定に合格する系列を μ-ランダムな系列と呼ぶということである．なお，μ-ランダムな系列の集合は確率測度 1 を持つ，すなわち $\mu(\{\omega : \delta_0(\omega|\mu) < \infty\}) = 1$ となる．特に μ が一様分布のとき，μ-ランダムな系列を（アルゴリズム情報理論的に）ランダムな系列と呼ぶ．

上のようにランダム性を定義すると，ランダム性と圧縮不可能性とのつながりを示す次の事実が成り立つ：無限長の 2 進列 $\omega \in \{0,1\}^\infty$ がランダムであるのは，ある定数 c が存在してすべての n に対して $K(\omega_{1:n}) \geq n - c$ であるとき，かつ，そのときに限る．すなわち，ω がランダムであることは，ω が漸近的に圧縮できないことと等しい．また，この事実より，$\rho(\omega) = \sup_{n\in\mathcal{N}}\{n - K(\omega_{1:n})\}$ と定めれば，$\rho(\omega) < \infty$ となるのは ω がランダムであるとき，かつ，そのときに限られる．すなわち，コルモゴロフ複雑度を用いて一様分布に対する万能検定を構成できる．

(2) 停止確率

実数 $\Omega \in [0,1)$ を $\Omega = \sum_{p:U(p)\,\mathrm{halts}} 2^{-\ell(p)}$ によって定義する．ここで和は万能チューリング機械 U を停止させるプログラム全体にわたってとられる．Ω は，一様分布からランダムに発生させた 2 進列を入力として与えたときに万能チューリング機械 U が停止する確率であり，停止確率（Halting Probability） と呼ばれる．

停止確率 Ω は以下の性質を持つ．(1)Ω は計算不可能である．(2) Ω を 2 進数展開したときの小数点以下 1 桁目から n 桁目までのビット列 $\Omega_{1:n} \in \{0,1\}^n$ が与えられたとすると，長さ n の任意の 2 進列 p について，p を入力したときにチューリング機械 U が停止するか否かを判定することができる．(3) ある定数 c が存在して，任意の n に対して $\Omega_{1:n}$ は $K(\Omega_{1:n}) \geq n - c$ となる．すなわち，Ω はランダムである．

6-4-4 コルモゴロフ複雑度とデータの構造

(1) コルモゴロフ最小十分統計量

長さ n の 2 進列 $x \in \{0,1\}^n$ を次のように符号化することを考える．まず x を含むある集合 $S \subseteq \{0,1\}^n$ を符号化し，次に x が S のどの要素かを $\log|S|$ ビットで符号化する．このような 2 段階符号化が最適，すなわち $K(S|n) + \log|S| \approx K(x|n)$ となるならば，集合 S は x が持つ構造をすべて捉えており，S が与えられたもとで x はランダムになっていると考え

ることができる．このような考えに基づき，コルモゴロフ構造関数（(Kolmogorov Structure Function）を $K_k(x|n) = \min\{\log|S| : \ell(p) \le k, U(p,n) = S, x \in S \subseteq \{0,1\}^n\}$ と定める．そして，定数 c を任意に固定して $k^* = \min\{k : k + K_k(x|n) \le K(x|n) + c\}$ とし，k^* に対応する集合 S と入力 p をそれぞれ S^* と p^* で表す（つまり S^* と p^* は $\log|S^*| = K_{k^*}(x|n)$，$\ell(p^*) \le k$ 及び $x \in S^* = U(p^*,n)$ を満たす）．このとき p^*（あるいは S^*）を x のコルモゴロフ最小十分統計量（Kolmogorov Minimum Sufficient Statistics）という[5]．コルモゴロフ最小十分統計量 p^* は，x に含まれる構造の最もコンパクトな表現になっていると考えることができる．

（2）コルモゴロフ複雑度に基づくモデル選択

あるデータ $x \in \{0,1\}^*$ が与えられたとき，そのデータを生成した情報源の構造を推定するという問題を考える．ここで，データは何らかの確率分布に従って生成されると仮定する．つまり，あるパラメータ $\theta \in \Theta$ によって P_θ と指定される確率分布の族 $\{P_\theta\}_{\theta\in\Theta}$ があって，その中のいずれかの確率分布 P_θ に従って x が生成されたと考える．すると問題は，与えられたデータ x に対してどのパラメータ $\theta \in \Theta$ を選ぶかという問題になる．なお，一般的に，パラメータ空間 Θ は，幾つかの異なる構造を持つ部分空間から構成される（例えば，$\Theta = \Theta_1 \cup \Theta_2 \cup \cdots$ で，各 n について Θ_n が次数 n のパラメータの集合である場合など）．

この問題に対して，ここではデータの記述量の観点から選択基準を考える．いま，最初にパラメータ θ を $K(\theta)$ ビットで記述し，その後 P_θ に対するシャノン・ファノ符号を用いて x を $-\log P_\theta(x)$ で符号化するという 2 段階符号化を考える．このような符号化を行うと，x は $-\log P_\theta(x) + K(\theta)$ ビットで符号化されることになる．この「x を符号化するのに必要なビット数」を最小にする θ は，x を記述するのに最も適したパラメータであると言える．このように考えると，与えられたデータ $x \in \{0,1\}^*$ に対して $-\log P_\theta(x) + K(\theta)$ を最小にする $\theta \in \Theta$ を選べばよいという選択基準を得ることができる．

上で述べた選択基準は，$K(\theta)$ が計算不可能なので実際に利用することはできない．しかしながら，モデルを符号化するための記述量 $(K(\theta))$ とそのモデルのもとでのデータの記述量 $(-\log P_\theta(x))$ との和を最小にするモデルを選択するという考え方は，MDL（Minimum Description Length）原理[7]や MML（Minimum Message Length）原理[8] の基本的な指針を与えている．

■参考文献

1) A.N. Kolmogorov：“Three approaches to the quantitative definition of information,” Problems Inform. Transmission, vol.1, no.1, pp.4–7, 1965.
2) R.J. Solomonoff：“A formal theory of inductive inference,” Inf. Control, vol.7, pp.1–22, 224–254, 1964.
3) G.J. Chaitin：“On the length of programs for computing finite binary sequences,” J. Assoc. Comput. Mach, vol.13, pp.574–569, 1966.
4) M. Li and P. Vitányi：“An Introduction to Kolmogorov Complexity and Its Applications, 2nd Edition,” Springer-Verlag, 1997.
5) T.M. Cover and J. Thomas：“Elements of Information Theory, 2nd Edition,” John Wiley & Sons, Inc., 2006.
6) S.K. Leung-Yan-Cheong and T.M. Cover：“Some equivalences between Shannon entropy and Kolmogorov complexity,” IEEE Trans. Inf. Theory, vol.IT-24, no.3, pp.331–338, May 1978.
7) J. Rissanen：“Modeling by the shortest data description,” Automatica, vol.14, pp.465–471, 1978.

8) C.S. Wallace and D.M. Boulton：“An information measure for classification,” Comput. J., vol.11, no.2, pp.185–195, 1968.