

単純マルコフ情報源に対する シンδροーム情報源符号化について

准員 鈴木 輝暁† 正員 今井 秀樹†

On Syndrome-Source-Coding for Simple Markov Sources

Teruaki SUZUKI†, Associate Member and Hideki IMAI†, Regular Member

あらまし 誤り訂正符号を用いてデータ圧縮を行う方式に、シンδροーム情報源符号化と呼ばれる方式がある。この方式に対し、情報源モデルとして無記憶情報源を仮定した場合に、RL (ランレングス) 符号化などの比較が行われ、シンδροーム情報源符号化の方が、広い範囲の無記憶情報源に対して効率が良いという意味で、より一般的であるという結果が得られている。本論文では、情報源モデルとして実際の情報源により近い単純マルコフ情報源を仮定し、(I)RL 符号化、(II)差分処理後にRL 符号化、(III)バースト誤り訂正符号を用いたNUSSC (Noiseless Universal Syndrome-Source-Coding)、(IV)差分処理後にランダム誤り訂正符号を用いたNUSSC、(V)情報源分離後にランダム誤り訂正符号を用いたNUSSC の5種類の方式を適用した場合に、各方式の効率の比較を行った。この結果、ファクシミリなどへの応用を考える場合には、従来良い方式と考えられてきた方式(III)は必ずしも良い方式ではなく、方式(IV)、(V)の方が、広い範囲で効率が良いという意味で、より一般的であるという結論が得られた。

1. ま え が き

最近、電話回線の解放や画像通信に対する社会的需要を背景として、ファクシミリなどにおけるデータ圧縮が盛んに研究され、実用に供されるに至っている。データ圧縮は、情報の蓄積という問題においても、記憶容量低減のための重要な技術である。

いままでに提案されてきたデータ圧縮方式の大部分はランレングス符号化(以下、RL 符号化と略記する)を基本とする種々の変形であり、これらの幾つかは実用化されている。データ圧縮方式には、誤り訂正符号を用いる方式もあり、理論的な面から種々の研究がなされている。この方式には、符号語法(図1)、シンδροーム情報源符号化法(図2)の二つの方式があり、文献(1)、(2)に両者の比較が行われている(表1)。

この二つの方式を比べると、シンδροーム情報源符号化法の方が、符号語法よりも、実際の情報源に対して適していると考えられる。その理由は次のとおりである。符号語法には、低いR(高い圧縮率)の符号語

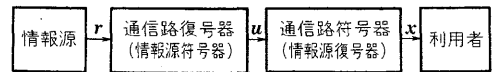


図1 符号語法
Fig.1-The codeword method.

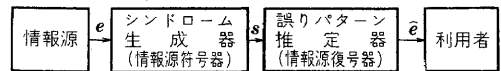


図2 シンδροーム情報源符号化法
Fig.2-The syndrome-source-coding method.

表1 二つの方式の比較

	符号語法	シンδροーム情報源符号化法
情報源出力	受信語 r	誤りパターン e
情報源符号器	通信路復号器 (複雑)	シンδροーム生成器 (簡単)
圧縮データ	情報系列 u	シンδροーム s
圧縮率	$1/R^*$	$1/(1-R)$
情報源復号器	通信路符号器 (簡単)	誤りパターン推定器 (複雑)
推定情報	符号語 x	誤りパターン推定値 \hat{e}

* R (符号化率) = k (情報点数) / n (符号長)

† 横浜国立大学工学部情報工学科, 横浜市
Faculty of Engineering, Yokohama National University,
Yokohama-shi, 240 Japan
論文番号: 昭 54-472[A-129]

によって代表的な情報源出力系列をうまく近似できないということ、及びデータ圧縮の際に用いる符号をど

んな符号にすればよいのかという評価基準がはっきりとしていないという短所がある。これに対し、シンドローム情報源符号化法には、パースト誤り訂正符号などを用いることによりかなりの記憶がある情報源出力系列をも良く近似できるということ、及び用いる符号の評価基準として誤り訂正能力を用いることができ、評価基準がはっきりとしているという長所がある。又、簡単な装置が情報源側にあり、宇宙実験などに適している。

文献(2)において、情報源モデルとして無記憶情報源を仮定した場合に、(i)RL 符号化と(ii)ランダム誤り訂正符号を用いた NUSC(Noiseless Universal Syndrome-Source-Coding) を適用して効率の比較を行い、方式(ii)の方が、広い範囲の無記憶情報源に対して効率が良いという意味で、より一般的であるという結果が得られている。そして、記憶のある情報源に対しても、シンドローム情報源符号化が有効であろうという推測がなされている。

本論文では、情報源モデルとして実際の情報源により近い単純マルコフ情報源を仮定し、5種類の方式を適用した場合に、各方式の効率の比較を行った。5種類の方式は、次のとおりである。

- (I) RL 符号化
- (II) 差分処理後に RL 符号化
- (III) パースト誤り訂正符号を用いた NUSC
- (IV) 差分処理後にランダム誤り訂正符号を用いた NUSC
- (V) 情報源分離後にランダム誤り訂正符号を用いた NUSC

以下、それぞれ RL 方式、差分 RL 方式、パースト NUSC 方式、差分 NUSC 方式、情報源分離 NUSC 方式と略記する。このうち、差分 NUSC 方式および情報源分離 NUSC 方式は、本論文で新たに提案する方式である。

この結果、従来文献(2)などにおいて良いであろうとされてきたパースト NUSC 方式は、ファクシミリなどへの応用を考える場合には、簡単な方式である差分 RL 方式よりも効率が悪くなってしまいうということが分かった。しかし、差分 NUSC 方式および情報源分離 NUSC 方式のように前処理後に NUSC を行うことにより、大幅な効率の向上が得られ、これらは、広い範囲で効率が良いという意味で、より一般的な方式であるという結論を得ることができた。

なお、データ圧縮方式として、ランダム誤り訂正符

号を用いた NUSC も考えることができる。しかし、本論文では、ファクシミリなどのように情報がパースト的に発生する情報源への応用を考えており、ランダム誤り訂正符号の誤り訂正能力を有効に用いることができないので、考察の対象から除外した。

2. 各方式の定式化

本章では、単純マルコフ情報源についての簡単な説明を行った後に、各方式について1情報源記号を表すのに必要なディジット数を導出することにしてしよう。

2.1 単純マルコフ情報源

情報源モデルとして、図3に示す単純マルコフ情報源を用いる。本論文では、主としてファクシミリへの応用を考えているので、二つの状態を状態 W (white)及び状態 B (black)とした。ここで、 P は W から B への遷移確率を、 p は逆に B から W への遷移確率を表す。又、状態 W 及び状態 B の定常確率 $P(W), P(B)$ は明らかに、 $P(W)=p/(p+P), P(B)=P/(p+P)$ となる。

単純マルコフ情報源のエントロピー(1次エントロピー) H_1 は、

$$H_1 = P(W) \cdot h(P) + P(B) \cdot h(p) \text{ ビット / 記号} \quad (1)$$

となる⁽³⁾。但し、 $h(x) = -x \log x - (1-x) \log(1-x)$ である。

2.2 RL 方式

RL 符号化にはいろいろな方式が考えられるが、本論文では解析しやすい次の方式⁽²⁾を用いることにする。なお、他の RL 符号化方式を用いても、効率はあまり変わらないと考えられる。

符号器は情報源から出力された各々の“1”の間の、連続した“0”の数(この数は0であることもある)を m ビットの2進数で表現して伝送する。但し、この RL が $2^m - 1$ 以上の場合は、符号器は2進数で $2^m - 1$ を伝送し、続いてラン中の残りの“0”の数(この数は0であることもある)を表す m ビット単位の符号を送る。従って、RL が $(i-1) \cdot 2^m - (i-1)$ から $i \cdot 2^m - (i+1)$ までの範囲のとき、必要なビット数は im ビットとなる。

ラン当りの平均ディジット数 d_{RL} は容易に求めるこ

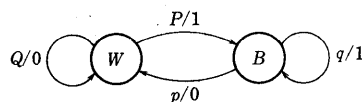


図3 単純マルコフ情報源
Fig.3-Simple Markov source.

とができ、次式のようになる。

$$d_{RL} = m \left[1 + \frac{pA}{Q(1-A)} \right] \quad (2)$$

但し、 $A=Q^{2^m}-1$ である。

出力された全記号数を N とすると、 N が十分大きい場合には“1”の発生個数は $N \cdot P(B)$ となり、ランの総数も $N \cdot P(B)$ となる。

以上より、RL方式において1情報源記号を表すのに必要なディジット数 ν_{RL} は次式となる。

$$\nu_{RL} = P(B) \cdot d_{RL} \quad (3)$$

2.3 差分RL方式

通常、ファクシミリなどでは隣接した記号間の相関が強いので、予測を行い“0”のRLを長くしてRL符号化を行う場合が多い。そこで、本節ではこの方式について検討を行うことにしよう。

ここでの予測は差分処理⁽³⁾とする。すなわち、既に出力された情報源出力 X_{i-1} を用いて現時点の出力 X_i の値を

$$\hat{X}_i = X_{i-1} \quad (4)$$

で予測し、予測値 \hat{X}_i と現時点の出力 X_i が一致すれば“0”、一致しないときは“1”なる値をとる系列 $\{Y_i\}$ に変換する。系列 $\{Y_i\}$ ($Y_i = X_i \oplus X_{i-1}$) は元の系列 $\{X_i\}$ の0,1の変化点を表す。Yが“1”となる確率を $P_d(B)$ とすれば、単純マルコフ情報源の場合、

$$P_d(B) = P(W) \cdot P + P(B) \cdot p \quad (5)$$

となる。系列 $\{Y_i\}$ における記号の発生を表すモデルは、図3の $p/0, q/1$ を、それぞれ $p/1, q/0$ に置換えればよい。

状態 W におけるラン当りの平均ディジット数 d_W 、及び状態 B におけるラン当りの平均ディジット数 d_B は容易に求めることができ、次式のようになる。

$$d_W = m / (1-A) \quad (6)$$

$$d_B = m / (1-C) \quad (7)$$

但し、 $A=Q^{2^m}-1, C=q^{2^m}-1$ である。

出力された全記号数 N が十分大きい場合には、“1”の発生個数は $N \cdot P_d(B)$ となり、ランの総数も $N \cdot P_d(B)$ となる。そして、ランの総数が十分大きい場合には、状態 W におけるラン数と状態 B におけるラン数はほぼ等しく、 $N \cdot P_d(B) / 2$ で与えられる。

以上より、差分RL方式において1情報源記号を表すのに必要なディジット数 ν_{dRL} は、次式となる。

$$\nu_{dRL} = P_d(B) \cdot (d_W + d_B) / 2 \quad (8)$$

2.4 NUSSC⁽²⁾

ここで簡単にNUSSCについて説明しよう。 $V_1, V_2,$

\dots, V_M ($M=2^m p$) を符号化率 $R_1 \geq R_2 \geq \dots \geq R_M$ であるような符号長 n の2元線形符号とする。又、 E_i を符号 V_i の訂正可能な誤りパターンの集合とする ($i=1, 2, \dots, M$)。通常、 $V_M = \{0\}$ として E_M を n 次元2元ベクトルのすべての集合とする。これは無ひずみ (noiseless) とするためである。

NUSSCは2元情報源に対して次のように符号化する方式である。まず、長さ n の情報源出力 e に対して、 $e \in E_i$ となる最小の i を見出し、 i を m_p ビットのプレフィクスで表す。そして、符号 V_i を用いて、情報源出力 e から $n(1-R_i)$ ビットのシンδροーム $s = eH_i^T$ を生成する。但し、 H_i は V_i の検査行列である。このようにして、情報源出力 e は図4のようにプレフィクスとシンδροームに対応づけて伝送されることになる。

利用者側においては、まずプレフィクス i から V_i を識別し、次に V_i の復号を行うことにより e を求める。以上より、情報がひずみなしに再生されることになる。

一般に、NUSSCにおいて1情報源記号を表すのに必要なディジット数 ν は、

$$\nu = m_p / n + (1-R_1)P_1 + \dots + (1-R_M)P_M \quad (9)$$

で与えられる。但し、 P_i は情報源出力 e が誤りパターンの集合 E_i に含まれる確率とする。

2.5 バーストNUSSC方式

単純マルコフ情報源において、長さ n の出力内に長さ b の単一バーストが発生する確率 $P(b, n)$ は、文献(4)より次のように与えられる。

$$P(0, n) = p \cdot Q^{n-1} / (p+P) \quad (10 \cdot a)$$

$$P(1, n) = \frac{pPQ^{n-3}}{p+P} [(n-2)p+2Q] \quad (10 \cdot b)$$

$$P(b, n) = \frac{[P+p(Q-p)^{b-1}]pP}{(p+P)^2} \cdot Q^{n-b-2} \cdot [(n-b-1)p+2Q] \quad (b \geq 2) \quad (10 \cdot c)$$

これを用いれば、バーストNUSSC方式において1情報源記号を表すのに必要なディジット数は、式(9)から計算できる。

2.6 差分NUSSC方式

まず初めに、この節および2.7で必要となる確率式を導出することにしよう。

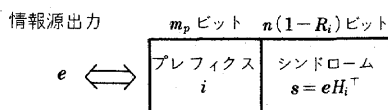


図4 情報源出力の伝送形式
Fig.4-Transmission format of source output.

差分処理を行う場合、初期状態を決めておく必要がある。ここでは、これを W としよう。本論文では、主としてファクシミリへの応用を考えており、これらにおいては通常 $P(W)$ が大きいので、初期状態を W とするのは自然な設定である。

ここで、長さ n の情報源出力を考えることにしよう。情報源出力における最後のディジットが B であるバーストを“タイプ-1”のバースト、最後のディジットが W であるバーストを“タイプ-2”のバーストと呼ぶことにする。又、図5のように長さ n の出力内に、 r 個のソリッドバースト(すべての要素が B のバースト)が発生しているバーストを“ r 重ソリッドバースト”と呼び、全体として B を k 個含む r 重ソリッドバーストを“重み k の r 重ソリッドバースト”と呼ぶことにしよう。

重み k のタイプ- j ($j=1,2$)の r 重ソリッドバーストが発生する確率を $P_{r,j}(k,n)$ で表すことにしよう。又、重み k の r 重ソリッドバーストが発生する確率を $P_r(k,n)$ 、長さ n の出力内に B が発生しない確率を $P_0(n)$ としよう。これらは、次のようになる(導出は付録参照)。

$$P_0(n) = Q^n \tag{11}$$

$$P_{1,1}(k,n) = Q^{n-k} P_q^{k-1} \tag{12}$$

$$P_{1,2}(k,n) = (n-k) P_q^{k-1} p Q^{n-k-1} \tag{13}$$

$$P_{r,j}(k,n) = \frac{k-1}{r-1} \sum_{i=0}^{n-k-r+1} Q^i P_r \cdot P_{r-1,j}(k-1, n-i-2) \quad (j=1,2; r \geq 2) \tag{14}$$

$$P_r(k,n) = P_{r,1}(k,n) + P_{r,2}(k,n) \tag{15}$$

k の取り得る範囲は、明らかに

$$r \leq k \leq n-r+1 \tag{16}$$

であり、 r の取り得る範囲は次のようになる。

$$1 \leq r \leq \lfloor (n+1)/2 \rfloor \tag{17}$$

ここに、 $\lfloor x \rfloor$ は x 以下の最大整数を示す。

さて、差分NUSSC方式は、図6のような予測誤差系列にそのままランダム誤り訂正符号を用いたNUSSCを適用する方式である。この節で導出した確率式を用いることにより、予測誤差系列における予測誤りの発生確率を求めることができる。 s 個の予測誤りが発生する確率を $P_D(s,n)$ で表すと、これは次式で与えられ

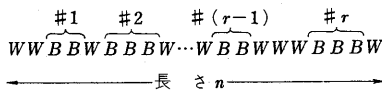


図5 r 重ソリッドバースト
Fig.5- r -tuple solid bursts.

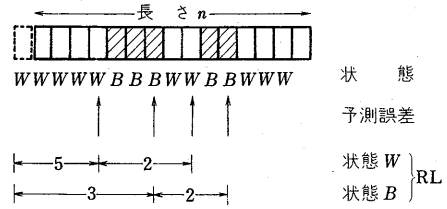


図6 ランが統計的に独立なモデル
Fig.6-The model of statistical independent runs.

る。

$$\left. \begin{aligned} P_D(0,n) &= Q^n \\ P_D(2t-1,n) &= \sum_{k=t}^{n-t+1} P_{t,1}(k,n) \quad (t \geq 1) \\ P_D(2t,n) &= \sum_{k=t}^{n-t+1} P_{t,2}(k,n) \quad (t \geq 1) \end{aligned} \right\} \tag{18}$$

これを用いれば、差分NUSSC方式において1情報源記号を表すのに必要なディジット数は、式(9)から計算できる。

2.7 情報源分離NUSSC方式

差分処理後の系列、すなわち予測誤差系列 $\{Y_i\}$ における予測誤りはランダムに発生するものではない。これは、系列 $\{Y_i\}$ の0次エントロピー $H_0(Y) = h(P_D(B))$ が単純マルコフ情報源のエントロピー H_1 よりも大きくなることから分かる⁽⁵⁾。このため、差分NUSSC方式はランダム誤り訂正符号の誤り訂正能力を有効に用いているとはいえない。そこで、情報源を分離するという概念を導入しよう⁽⁶⁾。すなわち、図6のように、状態 W と状態 B で別々に予測誤りの間のRLを求めるのである。例えば、状態 W で最初の予測誤りが発生するまでに、状態 W が5個生じていればRLは5となる。このようにすれば、各状態のランの発生はランダムとなる。このことは、式(1)より明らかである。

さて、情報源分離NUSSC方式では、情報源出力に対応する系列を求めるアルゴリズムとして次のようなものを採用する。

- (a) 情報源から n ディジット取ってくる。
- (b) 図6に従い、各状態のRLを求める。
- (c) 各状態のRLから、各状態に対応する系列を求める。但し、RLが j のとき、これは

$$\underbrace{00 \cdots 01}_{j-1 \text{ 個}}$$

に対応させる。

- (d) 状態 W に対応する系列の長さが n となるように、後にダミーの“0”を付加する(この長さ n の系列

を e_W とする)。

(e) 状態 B に対応する系列は、系列 e_W の後にそのまま続ける。

このようなアルゴリズムを用いれば、情報を一意的に再生できることは明らかである。

例えば、図6の情報源出力は、図7の系列と対応づけられる。

2.6で導出した確率式を用いることにより、系列 e_W における“1”の発生確率を求めることができる。 r 個の“1”が発生する確率 $P_r(n)$ は、 r 重ソリッドバーストが発生する確率に等しく、次式で与えられる。

$$\left. \begin{aligned} P_0(n) &= Q^n \\ P_r(n) &= \sum_{k=r}^{n-r+1} P_r(k, n) \quad (r \geq 1) \end{aligned} \right\} \quad (19)$$

このようなアルゴリズムを用いるので、情報源出力 e は図8と対応づけられて伝送されることになる。なお、情報源出力の最後のディジットが W の場合、状態 B に対応する系列の長さは k であるが、最後のディジットが B の場合には $k-1$ となる。

式(19)を用いれば、情報源分離NUSSC方式において1情報源記号を表すのに必要なディジット数は、次式から計算できる。

$$\begin{aligned} \nu &= m_p/n + (1-R_1)P_1 + \dots + (1-R_M)P_M \\ &+ \sum_{r=1}^{\lfloor (n+1)/2 \rfloor} \sum_{k=r}^{n-r+1} [(k-1)P_{r,1}(k, n) \\ &+ kP_{r,2}(k, n)]/n \end{aligned} \quad (20)$$

3. 計算結果

各方式において1情報源記号を表すのに必要なディジット数を ν とすれば、情報源符号化方式の効率 η は、

$$\eta = H_1 / \nu \quad (21)$$

で定義される。但し、 H_1 は式(1)で与えられる。この式は、理論上最少の必要ディジット数 H_1 の ν に対する比率を表すものである。

3.1 $n=15$ の場合

NUSSCを用いる三つの方式では、符号長 $n=15$ 、 $M=2^m=4$ とする。これに対応するRL方式と差分RL

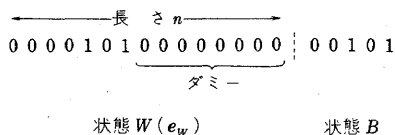


図7 情報源出力に対応する系列

Fig.7-The sequence corresponding to source output.

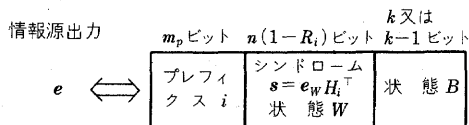


図8 情報源出力の伝送形式

Fig.8-Transmission format of source output.

方式では、 $m=4$ となる。

3.1.1 バースト誤り訂正符号を用いる場合

ファクシミリなどへの応用を考える場合、 W の方が B よりも出やすいため、単純マルコフ情報源からの長さ n の出力内に長さ b の単一バーストが発生する確率 $P(b, n)$ は、 b に関して単調減少であるという性質がある。そこで、誤りパターンの集合は、

- (1) $E_1 = \{0\}$
- (2) E_2 : 長さ1のバースト
- (3) E_3 : 長さ2の単一バースト
- (4) E_4 : 長さ3から15までの単一バースト

と設定した。バースト誤り訂正符号として、実現可能な最適符号⁽⁶⁾を用いることにすると、

- (1) V_1 : (15, 15) 符号
- (2) V_2 : (15, 11) ハミング符号
- (3) V_3 : (15, 10) 巡回符号
- (4) $V_4 = \{0\}$

となる。(1)では、シンδροーム s は空となり、(4)では、 $s=e$ となる。

R_i, P_i は容易に求めることができる。そして、式(9)を用いてバーストNUSSC方式の効率を計算することができる。

3.1.2 ランダム誤り訂正符号を用いる場合

誤りパターンの集合は、文献(2)と同じく、

- (1) $E_1 = \{0\}$
- (2) E_2 : 重み1の誤り
- (3) E_3 : 重み2の誤り
- (4) E_4 : 重み3から15までの誤り

と設定した。符号は次のようなものを用いる。

- (1) V_1 : (15, 15) 符号
- (2) V_2 : (15, 11) ハミング符号
- (3) V_3 : (15, 7) BCH 符号
- (4) $V_4 = \{0\}$

差分NUSSC方式と情報源分離NUSSC方式のそれぞれの R_i, P_i は容易に求めることができる。そして、式(9)、(20)を用いてそれぞれの方式の効率を計算することができる。

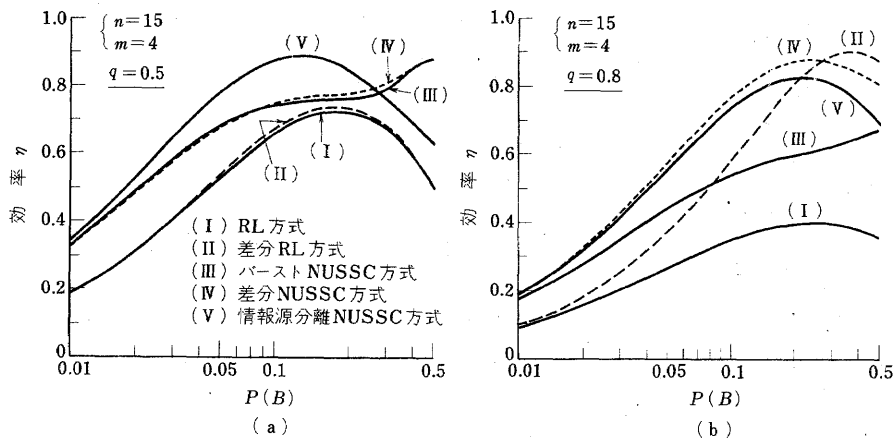


図9 5種類の方式の効率
Fig.9-Efficiencies of the five schemes.

3.1.3 計算結果

計算結果のグラフとして、図3における遷移確率 q が0.5と0.8の場合を図9(a), (b)に示す。

3.2 $n=31$ の場合

NUSSCを用いる三つの方式では、符号長 $n=31$, $M=2^m=8$ とする。これに対応するRL方式と差分RL方式では、 $m=5$ となる。

3.2.1 バースト誤り訂正符号を用いる場合

3.1.1と同様な理由により、誤りパターンの集合は、次のように設定した。

- (1) $E_1 = \{0\}$
- (2) \sim (7) E_i : 長さ $(i-1)$ の単一バースト ($i=2, 3, \dots, 7$)
- (8) E_8 : 長さ7から31までの単一バースト符号は次のようなものを用いる⁽⁶⁾。

- (1) V_1 : (31, 31) 符号
- (2) V_2 : (31, 26) ハミング符号
- (3) V_3 : (31, 25) 巡回符号
- (4) V_4 : (31, 23) 短縮化巡回符号
- (5) V_5 : (31, 22) 短縮化巡回符号
- (6) V_6 : (31, 20) 短縮化巡回符号
- (7) V_7 : (31, 19) 短縮化巡回符号
- (8) $V_8 = \{0\}$

R_i, P_i は容易に求めることができる。そして、式(9)を用いてバーストNUSSC方式の効率を計算することができる。

3.2.2 ランダム誤り訂正符号を用いる場合

誤りパターンの集合は、文献(2)と同じく、

- (1) $E_1 = \{0\}$
 - (2) \sim (4) E_i : 重み $(i-1)$ の誤り ($i=2, 3, 4$)
 - (5) E_5 : 重み4と5の誤り
 - (6) E_6 : 重み6と7の誤り
 - (7) E_7 : 重み8から15までの誤り
 - (8) E_8 : 重み16以上の誤り
- と設定した。符号は次のようなものを用いる。
- (1) V_1 : (31, 31) 符号
 - (2) \sim (7) V_i : (31, $36-5i$) BCH 符号 ($i=2, 3, \dots, 7$)
 - (8) $V_8 = \{0\}$

差分NUSSC方式と情報源分離NUSSC方式のそれぞれの R_i, P_i は容易に求めることができる。そして、式(9), (10)を用いてそれぞれの方式の効率を計算することができる。

3.2.3 計算結果

遷移確率 q が0.5と0.8の場合の計算結果を図10(a), (b)に示す。

4. 比較検討

$q=0.5$ の場合、バーストNUSSC方式の効率はかなり様な特性を持ち、広い範囲で効率が良いという意味で一般的であることが分かる。ここで、ファクシミリなどへの応用を考えてみる。これらにおいては通常 $P(B)$ は0.1以下である。図10(a)の場合、 $0.05 < P(B) < 0.2$ の範囲で、RL方式と差分RL方式の方がまさっており、バーストNUSSC方式の方が優れているとは一概にはいえない。これに対し、差分NUSSC

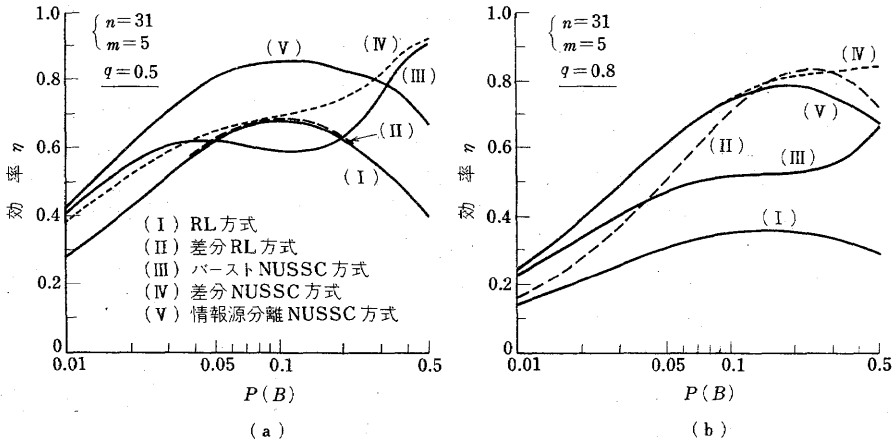


図10 5種類の方式の効率
Fig.10-Efficiencies of the five schemes.

方式、情報源分離NUSSC方式の方が、広い範囲で効率が良いという意味で、より一般的であるといえる。特に $q=0.5$ のように比較的小さい q に対する情報源分離NUSSC方式の特性には目を見はるものがある。

次に、ファクシミリにおける q の平均値として文献(3)に0.77という数値があがっており、 $q=0.8$ の方がより実際に近い場合を表していることになる。この場合、差分RL方式は非常に鋭いピークを持ち、大部分の範囲でパーストNUSSC方式を上回っていることが分かる。従って、ファクシミリへの応用を考える際には、パーストNUSSC方式は適しているとはいえない。しかし、 $q=0.5$ の場合と同様、差分NUSSC方式と情報源分離NUSSC方式は、考慮すべき範囲において他の三つの方式よりも効率が大きくなっている。従って、差分NUSSC方式と情報源分離NUSSC方式は、広い範囲で効率が良いという意味で、より一般的であるという結論が得られる。

5. むすび

本論文では、情報源モデルとして実際の情報源により近い単純マルコフ情報源を仮定し、シンドローム情報源符号化に関する検討を行った。適用した方式は、(I)RL符号化、(II)差分処理後にRL符号化、(III)パースト誤り訂正符号を用いたNUSSC、(IV)差分処理後にランダム誤り訂正符号を用いたNUSSC、(V)情報源分離後にランダム誤り訂正符号を用いたNUSSCの5種類である。

その結果、ファクシミリなどへの応用を考える場合

には、従来良い方式と考えられてきた方式(III)は、簡単な方式である方式(II)に劣ってしまうという結論が得られた。しかし、方式(IV)、(V)のように何らかの前処理の後にNUSSCを適用すれば効率が大幅に改善され、広い範囲で効率が良いという意味で、より一般的となることが分かった。

今後残された問題として、NUSSCにおける誤りパターンの集合をどのように設定したらよいか、又、ファクシミリで用いられている2ライン一括処理モードRL符号化と対応して、情報源を4個の状態からなるマルコフ情報源と考えた場合、本論文と同様な議論ができないか、などがある。

文 献

- (1) Massey, J.L.: "The codeword and syndrome methods for data compression with error-correcting codes", in "New directions in signal processing in communication and control" (J.K. Skwirzynski ed.), p.3, Noordhoff International Publishing, Leyden, Netherlands (1975).
- (2) Anчета, T.C. Jr.: "Syndrome-source-coding and its universal generalization", IEEE Trans. Inf. Theory, IT-22, p.432 (July 1976).
- (3) 電子通信学会編, 安田靖彦: "ファクシミリの基礎と応用", 電子通信学会(昭52).
- (4) 藤原, 中西, 笠原(正), 手塚, 笠原(秀): "パースト誤りを生ずる通信路とその改善度の改善に関する考察", 信学論(A), 51-A, 8, p.311(昭43-08).
- (5) Preuß, D.: "Twodimensional facsimile source encoding based on a Markov model", Nachrichtentech. Z., 28, 10, p.358 (1975).
- (6) Kasami, T.: "Optimum shortened cyclic codes

for burst-error-correction", IEEE Trans. Inf. Theory, IT-10, p.105 (April 1964).

付 録

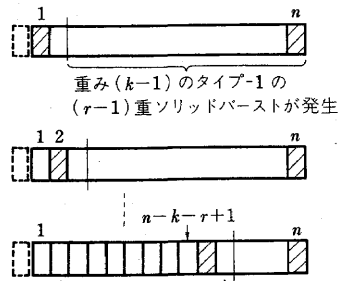
式(11)~(15)の導出

$P_0(n)$ は図3より明らかである。重み k のタイプ-1の1重ソリッドバーストが発生する確率 $P_{1,1}(k, n)$ が、式(12)で与えられることは容易に分かる。次に、重み k のタイプ-2の1重ソリッドバーストについて考えてみる。このようなバーストは、 $n-k$ とおり存在し、それぞれの発生確率は $Pq^{k-1}p \cdot Q^{n-k-1}$ で与えられるので、式(13)が導かれる。

次に、重み k のタイプ-1の r 重ソリッドバースト ($r \geq 2$)が発生する確率を考えてみる。まず、最初のソリッドバーストの長さが1である場合の確率は、図A.1より

$$\sum_{i=0}^{n-k-r+1} Q^i Pp \cdot P_{r-1,1}(k-1, n-i-2) \quad (A.1)$$

となる。初期状態が W で、最後のディジットが B であるバーストのみを考えているから、重み k の r 重ソリッドバースト(このようなバーストは ${}_{k-1}C_{r-1}$ とおり存在する)のそれぞれが発生する確率は、すべて等しくなることが分かる。最初のソリッドバーストの長さが1である場合の数は、 ${}_{k-2}C_{r-2}$ とおりであるから、式(A.1)の値を ${}_{k-2}C_{r-2}$ で割った値が、一つの重み k の



図A.1 $P_{r,1}(k, n)$ の説明
Fig.A.1-The illustration for $P_{r,1}(k, n)$.

r 重ソリッドバーストが発生する確率となる。従って、重み k のタイプ-1の r 重ソリッドバーストが発生する確率 $P_{r,1}(k, n)$ は、

$$P_{r,1}(k, n) = \frac{{}_{k-1}C_{r-1}}{{}_{k-2}C_{r-2}} \times \text{式(A.1)} \quad (A.2)$$

となる。ここで、

$${}_{k-1}C_{r-1} / {}_{k-2}C_{r-2} = (k-1) / (r-1) \quad (A.3)$$

であり、又、タイプ-2の r 重ソリッドバーストに対しても全く同様な議論が成立するから、結局式(14)が成立することが分かる。

重み k の r 重ソリッドバーストが発生する確率 $P_r(k, n)$ は $P_{r,1}(k, n)$ と $P_{r,2}(k, n)$ の和で与えられるから、式(15)が成立つ。

(昭和54年2月26日受付, 5月17日再受付)