電子情報通信学会誌 Vol.86 No.2 pp.120-126 2003年2月

Tutorial Series

IT講座

データ圧縮における最新アルゴリズム［I］

——無ひずみデータ圧縮アルゴリズムの変遷——

山本 博資

山本博資 正員 東京大学大学院情報理工学系研究科
E-mail Hirosuke@icee.org

New Algorithms in Data Compression［I］: The History of Lossless Data Compression Algorithms. By Hirosuke YAMAMOTO, Member (School of Information Science and Technology, The University of Tokyo, Tokyo, 113-8656 Japan).

1. はじめに

携帯電話や無線LANなどの発達をはじめとするIT技術の進歩に伴いユビキタスネットワーク社会が到来しようとしている．そのようなネットワーク社会では，伝送／記録される情報量がますます増加し，それとともに，より効率良く伝送／記録するためのデータ圧縮技術も，その重要性が増していくものと思われる．

データ圧縮に関する研究は，1948年の有名なシャノンの論文[1]に始まるが，その論文において確率分布$\{p(x)\}$を持つ定常無記憶な情報源系列の圧縮限界がエントロピー$H(X)=-\sum_x p(x)\log p(x)$であるという情報源符号化定理が示された．更にシャノンは，定常エルゴード情報源系列の圧縮限界がエントロピーレート$H(X)=\lim_{K\to\infty} H(X_K \mid X_1X_2\cdots X_{K-1})=\lim_{K\to\infty}(1/K)H(X_1X_2\cdots X_K)$であることを述べているが，1953年にマクミラン[2]がそのことを詳しく証明したことにより，データ圧縮の究極の目的がエントロピーレートを達成する符号の実現であることが明らかとなった．更に，1966年にはユニバーサル符号（あるクラスに属するすべての情報源系列を同一の符号でエントロピーレートまで圧縮できる符号）を理論的に構成できることがFitingof[3]，Lynch[4]，Davisson[5]により示され，その後，理論的な研究とともに，より広いクラスの情報源系列をより効率良く，より少ない計算量とメモリ量で符号化するための実用的なユニバーサルデータ圧縮アルゴリズムについて，数多くの研究がなされている．

本講座では，それらのデータ圧縮アルゴリズムのうちから，幾つか重要なものを取り上げ5回にわたって紹介する．2.で，まずデータ圧縮符号の分類を行った後，本講座の概要を紹介する．

データ圧縮アルゴリズムは，単にデータ圧縮の用途だけでなく，様々な分野に応用可能である．例えば，木符号はデータ検索や故障診断などの最適な探索木として利用できる．また，高性能なユニバーサルデータ圧縮符号は，情報源系列からそれらが持つデータ構造を自動的に抽出し，過去の系列から次のシンボルを精度良く推定する機能を本質的に有している．これらの性能をうまく利用すれば，データ圧縮アルゴリズムを他分野へ応用することも可能となる．このような意味から，本講座はデータ圧縮の研究者や技術者だけでなく，広く他分野の人にも読んで頂けることを期待している．

予定目次

2. データ圧縮符号の分類と本講座の概要

データ圧縮符号は，大きく「無ひずみデータ圧縮符号」と「有ひずみデータ圧縮符号」に分類される．無ひずみデータ圧縮符号は，元の情報源系列が1ビットの誤りもなく（あるいは無視できるほど小さい誤り確率で）復号できる符号である．これに対して，有ひずみデータ圧縮符号は，復号したデータにある程度のひずみを許す代りに，無ひずみ圧縮よりも更に小さく圧縮する方式である．有ひずみ圧縮は，音声や画像など，ある程度ひずみが許されるデータに使用されるが，誤りの許されない一般のファイル圧縮などには，無ひずみ圧縮が用いられる．

無ひずみデータ圧縮符号は表1のように分類できる．

エントロピー符号化に分類される符号は，符号の構成あるいは符号化に情報源系列の確率分布を陽に利用する符号である．ハフマン符号[6]やTunstall符号[7]，算術符号[8],[9]などがその代表的な符号である．情報源の確率分

表1　主なデータ圧縮符号の分類

I．エントロピー符号化
i．静的符号化（ハフマン符号，Tunstall 符号，算術符号など） ii．動的符号化（動的ハフマン符号，動的 Tunstall 符号，算術符号など）
II．ユニバーサル符号
II-A　確率分布のユニバーサルな推定と算術符号の組合せ（MDL 符号，CTW 符号，PPM 符号など） II-B　符号化アルゴリズムに確率を陽に含まない符号 i．構成要素として使用される符号（MTF 符号，正整数のユニバーサル符号など） ii．辞書法（LZ77 符号，LZ78 符号，LZW 符号，LZFG 符号など） iii．ソート法（BS 符号，CS 符号，ACB 符号など） iv．文法法（SEQUITUR 符号，MPM 符号など）

布が定常であると仮定し，固定した符号を用いる場合を静的符号化という．これに対して，実際符号化している情報源系列の頻度分布に基づき，適応的に符号を構成しながら符号化する場合を動的符号化という．

エントロピー符号化では，実際に符号化したい情報源系列の確率分布 $P=\{P(x), x\in\chi\}$ が符号構成時に仮定した確率分布 $Q=\{Q(x), x\in\chi\}$ と一致する場合は性能良く圧縮できるが，P と Q が異なる場合は少なくともダイバージェンス $D(P\|Q)=\sum_{x\in\chi}P(x)\log\frac{P(x)}{Q(x)}$ 分のロスを生じる．実用的には，事前に情報源の確率分布を完全に知ることは困難なため，エントロピー符号化が単独で用いられることは少なく，ユニバーサル符号の一部として利用される場合が多い．

ユニバーサル符号は，あるクラスに属する任意の情報源系列を効率良く圧縮できる（狭義には，漸近的にエントロピーレートまで圧縮できる）符号である．ユニバーサル符号の構成方法は，ユニバーサルな推定法を用いて情報源系列の確率分布を推定し，その推定した確率分布を用いてエントロピー符号化（特に算術符号化）を行う方式（II-A）と，符号化／復号化に確率を陽に含まないアルゴリズムを用いる方式（II-B）に，大きく分類できる．なお，適応的なエントロピー符号化である動的ハフマン符号[10],[11]や動的 Tunstall 符号[12],[13]は，任意の無記憶情報源系列に対して適応的に最適な符号を構成できるので，一種のユニバーサル符号と考えることもできるが，対象が無記憶情報源系列という非常に狭いクラスに限られているため，一般にはユニバーサル符号とは呼ばれていない．

II-A の方式は，符号化を2段階で行う2パス方式（全部の情報源系列を先に調べてその確率分布を推定し，その分布に基づいて算術符号化を行う方式）と，1段階で行う1パス方式（情報源系列を逐次的に読み込みながら，確率分布の推定と算術符号化を逐次的に繰り返す方式）とに分類できる．後者の実用的な符号としては，MDL 符号[14],[15]や PPM 符号[16]がある．MDL 符号は，推定した情報源の確率モデルを符号化するための記述長と，そのモデルの下で情報源系列を符号化した符号語長のトータルの長さを最小にする情報源モデルを用いて符号化する方式である．これに対して，情報源モデルを一つに固定せずに複数の情報源モデルを用い，それらをある事前確率分布で混合して情報源の確率分布を推定するベイズ符号があるが，そのような効率の良い符号として CTW 符号[17]がある．

II-B の方式は，符号の特徴により，ii．辞書法，iii．ソート法，iv．文法法に大きく分類できる．辞書法は，情報源系列の部分系列が辞書中のどの系列と一致しているかを示すポインタ情報を用いて符号化する符号化法であり，符号化とともに辞書を適応的に更新していくことでユニバーサル性を実現している．この方式には，LZ77 符号[18]や LZ78 符号[19]をはじめとする，いわゆるレンペル・ジブ符号（LZ 符号）系の符号が含まれている．ソート法では，辞書中の情報源系列をソートすることにより，更に効率良く符号化できるように工夫したものである．ブロックソート（BS）符号[20]，文脈ソート（CS）符号[21]，ACB 符号[22]などがある．ivの文法法は，符号化したい情報源系列を生成できる文法規則を作り，その文法規則を符号語とする符号化法である．SEQUITUR 符号[23],[24]や MPM 符号[25]などがこの方式の代表的な符号である．

辞書法など多くのユニバーサルデータ圧縮符号では，情報源系列の部分系列を正整数値のポインタ情報に符号化した後，その整数値を2値系列に符号化する場合が多い．このとき，正整数の2値符号化には，I のエントロピー符号化または II-B-i の正整数のユニバーサル符号が用いられる．正整数のユニバーサル符号は，正整数 n の生起確率が $p(n)\geq p(n+1)$ と，単調に減少するような特性を持つ場合に，ユニバーサルに効率の良い2値符号化を与えるものであるが，0や1の連なりの長さを整数値で符号化するランレングス符号[26]もその一種と考えることができる．また，同じ文字が連続して出現しやすい系列を，MTF（Move-To-Front）符号[27],[28]を用いて正整数に変換すると，小さい正整数が出現しやすくなり，正整数のユニバーサル符号で効率良く符号化できるようになる．MTF 符号のその特性は，BS 符号の中で利用され，BS 符号の高圧縮性能の一因となっている．なお，正整数のユニバーサル符号については，文献(29),(30)などに詳しく紹介されている．

本講座では，まず，第1回（本稿）では，データ圧縮

符号の分類と変遷を紹介した後，木（tree）構造を利用したエントロピー符号化について幾つかの符号化法を説明する．次に第2回（著者：植松友彦）でⅡ-B-iiの辞書法，第3回（著者：横尾英俊）でⅡ-B-iiiのソート法，第4回（著者：古賀弘樹）でⅡ-B-ivの文法法を取り上げる．また，第5回（著者：星　守）では，算術符号化と，その拡張として，ある確率分布を持つ系列から異なる確率分布を持つ系列を作成する乱数生成法を紹介する．

3. データ圧縮符号の変遷

1948年に，シャノンが情報源符号化定理の証明に用いたシャノン符号[1]以来，多くのデータ圧縮符号が提案されているが，その主なものを表2に示す．1950年～1960年代の初期のころに，定常無記憶情報源に対して最適なエントロピー符号化が行えるハフマン符号[6]やTunstall符号[7]が見つけられている．

ユニバーサルデータ圧縮符号の研究は，1966年の理論的な研究[3],[4],[5]に始まるが，実用的なユニバーサルデータ圧縮符号に関して，1970年代以降10年ごとに大きな進歩が生まれている．

まず，1970年代半ばに算術符号[8],[9]が考案され，情報源の確率分布が既知な場合に，実用的な計算量とメモリ量で，圧縮限界のエントロピーレートに十分近くまで容易に圧縮できるようになった．また，実用的なユニバーサル符号の原形となるレンペル・ジブ符号（LZ77符号[18]，LZ78符号[19]）も，ほぼ同じ時期に提案されている．

1980年代は，ユニバーサル符号の実用化が進んだ年代である．ユニバーサルな確率分布推定と算術符号を組み合わせるⅡ-Aの方式として，MDL符号[14],[15]，PPM符号[16]などの実用的な符号が提案された．また，LZ78符号を改良したLZW符号[34]が提案され，Unixのcompressコマンドとしてインプリメントされたことにより，ユニバーサル符号が広く使用されるようになった．また，LZSS符号[35]などLZ77符号を改良したユニバーサル符号も数多く提案され，LHaやgzipなどの圧縮ツールとしてパソコンなどでも広く用いられるようになった．

1990年代後半には，ソートを用いて符号化効率を良くするブロックソート（BS）符号[20]や文脈ソート（CS）符号[21]などが考案された．特に，ブロックソート法はレンペル・ジブ系の符号より圧縮率が優れているため，研究者の注目を集めるとともに，bzipなどの圧縮ツールとしてすぐに使用され始めた．また，情報源系列をそれが生成できる文法規則に符号化するという新しい概念に基づいたSEQUITUR符号[23],[24]やMPM符号[25]などが提案され，研究者の注目を集めている．

表2　主なデータ圧縮符号
（｛　｝内は関連してよく知られているもの）

1948	シャノン符号[1]
1952	ハフマン符号[6]
1959	ランレングス符号化[26]
1962	Tunstall符号[7]
1966	ヒストグラム符号化[3]
1968 ｛1975｝	正整数のユニバーサル符号[31]｛[32]｝
1973	数え上げ符号[33]
1973 ｛1978｝	動的ハフマン符号[10]｛[11]｝
1976	算術符号[8],[9]
1977	LZ77符号[18]
1978	LZ78符号[19]
1983	MDL符号化[14],[15]
1984	Welch符号（LZW符号）[34]
1984	PPM符号[16]
1986	Bell符号（LZSS符号）[35]
1986	Move-to-Front符号（MTF符号）[27]
｛1987｝	｛Recency-Rank符号[28]｝
1989	Fiala-Greene符号（LZFG符号）[36]
1992 ｛1996｝	動的Tunstall符号[13]｛[12]｝
1994	ブロックソート符号（BS符号）[20]
1994	ACB符号[22]
1994 ｛1997｝	SEQUITUR符号[23]｛[24]｝
1995	CTW符号[17]
1996	文脈ソート符号（CS符号）[21]
2000	MPM符号[25]

4. 符号木と分解木を用いるエントロピー符号化

3.で説明したように，1970年代以降ユニバーサルデータ圧縮符号は大きな発展を遂げてきているが，ハフマン符号などの木構造を用いたエントロピー符号化についても，様々な研究が続けられている[37]．4.では，その中の幾つかの重要な符号化アルゴリズムについて紹介する．

木構造を利用したエントロピー符号化では，情報源系列を部分系列に分解し，その部分系列を{0,1}からなる2値系列の符号語に符号化する．そのとき，情報源系列を固定長の部分系列に分解し，それを可変長の符号語に符号化する場合を固定長可変長符号（FV code：Fixed-to-Variable length code）といい，逆にデータ系列を可変長で部分系列に分解し，それを固定長の符号語に符号化する場合を可変長固定長符号（VF code：Variable- to-Fixed length code）という．また，FV符号とVF符号で用いられる木はそれぞれ符号木（code tree）

及び分解木（parse tree）と呼ばれる．

4.1 FV符号と符号木構成アルゴリズム

符号木の例を図1に示す．葉が情報源文字xに対応し，根から葉までの枝に振られた系列がxの符号語となる．符号語長$L(x)$は根から葉までの枝の個数で与えられる．各文字xが生起確率$p(x)$を持つとき，平均符号語長

$$L=\sum_x L(x)p(x) \qquad (1)$$

を最小にする符号木はハフマン符号木であり，次のハフマンアルゴリズムにより構成できる．

（1） ハフマンアルゴリズム

H-1 すべての文字xと葉を対応させ，各葉に確率$p(x)$を割り振る．$\chi=\{$葉の集合$\}$とする．

H-2 χの中で最も確率の小さい葉または節点x_1, x_2を選び，それらの親節点$\hat{x}$を作る．$\hat{x}$とx_1及びx_2をそれぞれ枝で結び，一方に0，他方に1を割り振る．親節点に確率$p(x_1)+p(x_2)$を割り振り，$\{x_1, x_2\}$をχから除き，代りに親節点$\hat{x}$をχに追加する．

H-3 χの要素数が1のとき，終了，さもなければH-2に戻る．

なお，文字xの頻度$n(x)$と出現文字総数nで決まる頻度分布$n(x)/n$を用いてハフマン木を作る場合は，すべてをn倍することにより，確率分布$p(x)$の代りに頻度$n(x)$を使って符号木を作れる．

次にハフマンアルゴリズムの拡張を幾つか紹介する．情報源の確率分布が既知でない場合や定常でない場合は，情報源系列の頻度を適応的に調べ，その頻度に基づいて，符号木が常に最適となるように逐次的に更新していく必要がある．

ハフマン木では，節点xの確率重み$p(x)$は，木の深さが深いほど小さくなる．したがってある節点xの頻度がそれより浅い節点$\hat{x}$の頻度より大きくなると，節点xと$\hat{x}$をそれぞれ根とする部分木を交換することで最適な

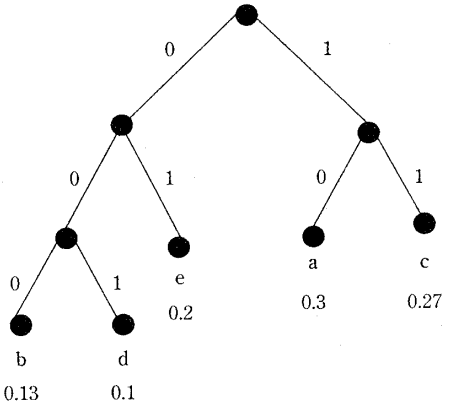


図1 ハフマン符号木　文字xの下の数字は，xの生起確率$p(x)$である．

符号木を適応的に作っていくことができる．そのような符号を動的ハフマン符号[10],[11]という．

ハフマン符号では，情報源文字xのアルファベット順序とその符号語$B(x)$のアルファベット順序は一般には一致していない．そのため，データ検索をするために情報源系列の比較を行いたい場合は，符号語系列を一度復号してから比較しなければならない．しかし，符号語$B(x)$のアルファベット順序がxのアルファベット順序に一致していると，復号せずに，符号語系列のままで比較することが可能となる．このような，xのアルファベット順序と符号語$B(x)$のアルファベット順序が一致するFV符号をアルファベット符号（Alphabetic code）というが，その符号木は次のHu-Tuckerアルゴリズムにより構成できる．

（2） Hu-Tuckerアルゴリズム

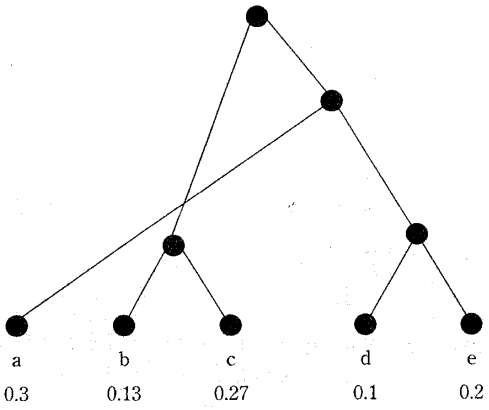
HT-1 すべての文字xをアルファベット順序に並べる．

HT-2 ハフマンアルゴリズムと同様のアルゴリズムで符号木を作る．ただし，ハフマンアルゴリズムのステップH-2において，最も小さい確率を持つ二つの節点x_1, x_2を選ぶときに，それらの間に，まだ親節点の作られていない葉が存在する場合は，その組を除いて考える．

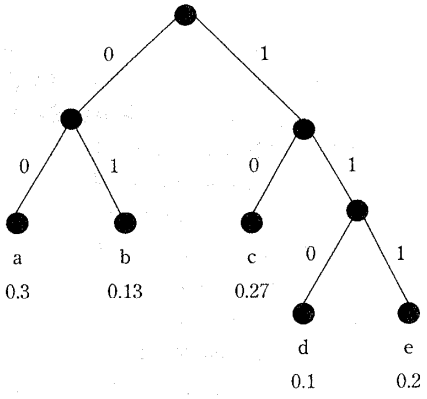
HT-3 全部の木が構成された後に，深さが同じレベルの枝を左から順に付け替え，アルファベット符号木を作る．

図1の場合と同じ確率分布に対するアルファベット符号木を図2に示す．符号語が情報源文字と同じ順序を保持していることが分かる．ただし，平均符号語長はハフマン符号に比べて，0.07だけ長くなっている．なお，ハフマン符号の平均符号語長Lはエントロピー$H(X)$に対して，$L\leq H(X)+1-h(p_{\min})$を満たすのに対して，アルファベット符号では$L\leq H(X)+2-p_{\max}-p_{\min}$となることが知られている[38]．ここで，$p_{\min}=\min_x p(x)$，$p_{\max}=\max_x p(x)$であり，$h(\cdot)$は2値エントロピー関数である．

FV符号の符号木は，故障診断や検索などの探索木として使用することができる．探索対象xが確率$p(x)$で生起するものとし，符号木の各節点を，探索対象が右側の集合に含まれているか左側の集合に含まれているかを調べる一つのテストに対応させれば，式(1)のLは平均探索回数となる．したがって，ハフマン符号は平均探索回数を最小とする探索木を与える．しかし，故障診断のような場合，任意の二つの集合に対して，テストを自由に設定できない場合が多い．診断対象が$a_1 \to a_2 \to \cdots \to a_A$と縦続接続された装置や処理システムの場合，$a_i$と$a_{i+1}$をつなぐ間でテストを行えば，$\{a_j, j\leq i\}$の集



(a) HT-2 実行後



(b) HT-3 実行後

図2　アルファベット符号木の構成

合に故障が含まれているか，$\{a_j, j \geq i+1\}$ に故障があるかのテストが容易に行える．そのような場合には探索木はアルファベット符号木になっている必要があるが，上記の Hu-Tucker アルゴリズムを用いれば，平均探索回数が最小のアルファベット探索木を作ることができる．

4.2 VF 符号と分解木構成アルゴリズム

VF 符号では，図3のような分解木を用いて，情報源系列を符号化する．根から葉までの系列が情報源系列に対応し，葉に割り振られた番号が符号語である．例えば，系列 $a_1a_1a_3$ は符号語番号2に符号化される．符号語の番号は葉の総数が N のとき，$\lceil \log_2 N \rceil$ ビットの2進数で表現できる．

VF 符号の平均分解長を S とすると，情報源系列1文字を表すのに必要な平均ビット長は，レート $R=\lceil \log_2 N \rceil / S$ で与えられ，N が一定の場合には，平均分解長 S が大きいほど効率が良い符号となる．情報源アルファベット $\{a_1, a_2, \cdots, a_A\}$ のサイズが A で，N がある自然

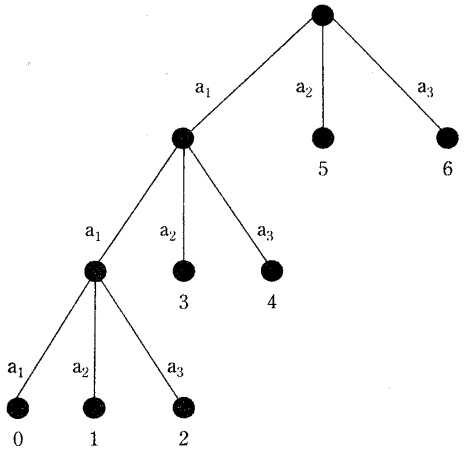


図3　Tunstall 分解木　$P(a_1)=0.6$，$P(a_2)=0.3$，$P(a_3)=0.1$，$N=7$，$A=3$ の場合

数 m に対して $N=(m-1)A$ という関係を満たしている場合は，最大の平均分解長 S を与える分解木は次の Tunstall アルゴリズムにより作ることができる．

（3）Tunstall アルゴリズム

T-1　分解木の根から A 本の枝と葉を作り，各枝に情報源文字 $a_1, a_2, \cdots, a_A$ を割り振る．また対応する葉に確率重み $P(a_i)$ を割り振る．$\hat{N}=A$ とする．

T-2　$\hat{N}=N$ のとき，すべての葉に符号語 $0 \sim N-1$ を割り振り，終了する．

T-3　分解木の葉の中で最も確率重みの大きいものを t とする．t を内部節点として A 本枝を延ばし，葉を A 個作る．各枝に $a_1, a_2, \cdots, a_A$ を割り振り，新しい葉に確率重み $P(t) \times P(a_i)$ を対応付ける．T-2 に戻る．

FV 符号で平均符号語長を最小にするハフマンアルゴリズムでは，最も確率重みの小さい葉をまとめて親節点を作っていたのに対して，平均分解長を最大にする Tunstall アルゴリズムでは，最も大きな確率重みを持つ節点から子の葉を延ばしており，互いに双対なアルゴリズムになっていることが分かる．

Tunstall の分解木は，その構成アルゴリズムより，葉の確率重みが内部節点の確率重みより大きくならない特徴を有している．情報源文字の頻度に基づいて，適応的に分解木を更新するためには，この性質を満たさない葉と内部節点を根とする部分木とを入れ替えることにより，動的 Tunstall 符号を作ることができる[10],[11]．

動的 Tunstall 符号は，葉の数 N が固定されている場合に分解木を最適なものに適応的に更新する符号であるが，情報源系列を分解木で分解するごとに，葉を一つずつ増加させても構わない場合は，最適な分解木は増分分

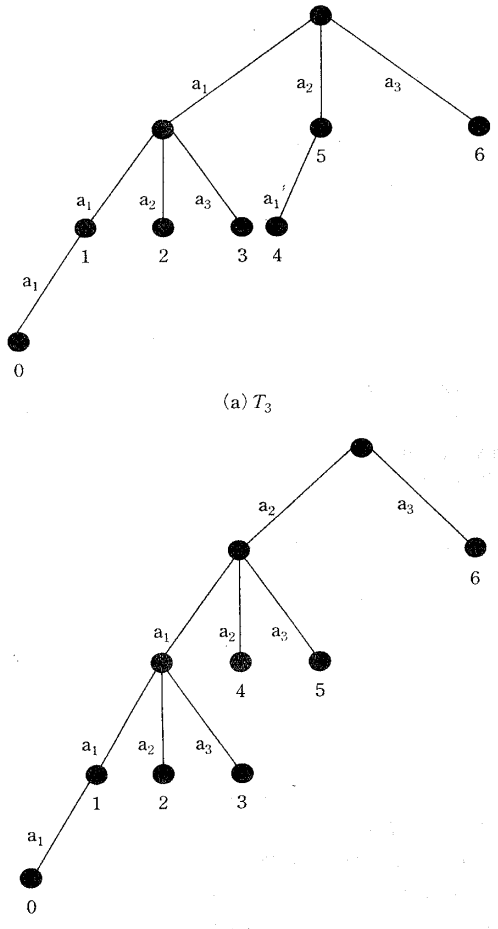


図4　AIVF 符号木

解木となる[39].

上記の Tunstall 符号は，情報源アルファベットサイズ A に対して，葉の総数 N が $N=(m-1)A$ を満たす自然数 m が存在するとき，構成することができる．したがって，$A=2$ の場合は任意の自然数 N に対して構成できるが，$A\geq 3$ では希望の N を構成できるとは限らない．葉に割り振られた符号語番号を $\lceil \log_2 N \rceil$ ビットで固定長符号化するには，N は 2 のべき乗になると都合がよい．しかし，例えば $A=3$ のとき，$N=2^4=16$ を持つ Tunstall 符号を作ることができない．これは Tunstall 符号では分解木として完全木だけを用いているためである．そこで，分解木として不完全木も許し，葉及び不完全節点（子供が A 未満の節点）にすべて符号語を割り振るように拡張した AIVF（Almost Instantaneous VF）符号[40]が提案されている．図 4 に図 3 と同じ確率分布に対する AIVF 符号の分解木を示す．深さの 1 段目に $k=2, 3, \cdots, A$ の葉を持つ $A-1$ 個の AIVF 分解木 T_k を用意し，j 個の子供を持つ節点（葉の場合は $j=0$）で符号化されたときには，次の符号化は分解木 T_{A-j} で行う．図3 の確率分布に対して，Tunstall 符号の平均分解長 S_T は $S_T=1.96$ であるが，AIVF 符号の平均分解長 S_A は $S_A=2.08$ となり，5 %以上改善されている．

5. ま　と　め

本稿では，データ圧縮アルゴリズムの分類とその変遷を紹介するとともに，木符号化について幾つかのアルゴリズムを紹介した．データ圧縮以外の木符号の応用として，4. では，探索木への応用を紹介したが，それ以外にもいろいろな応用がある．例えば，べき乗計算を乗算を用いて（あるいは乗算計算を加算を用いて）効率良く計算する方法に VF 符号やランレングス符号が利用できる[41],[42]．また，暗号システムのグループ鍵の更新を効率良く行う方法に FV 符号木が利用されている[43]．

これら以外にも，木を用いたデータ構造はいろいろな目的で利用されているが，それらをデータ圧縮の符号木や分解木などに関係付けることができれば，データ圧縮符号の研究成果をそれらに応用できる可能性がある．

なお，文献(29)，(30)，(37)，(44)，(45)，(46) などに様々なデータ圧縮符号のアルゴリズムが紹介されている．より深く知りたい人は，本講座と合わせてそれらの文献を参考にして頂きたい．

文　献

（1） C.E. Shannon, "A mathematical theory of communications," Bell Syst. Tech. J., vol.27, pp.379-423, vol.28, pp.623-625, 1948.
（2） B. MacMillan, "The basic theorems of information theory," Ann. Math. Statistics, vol.24, pp.196-219, 1953.
（3） B.M. Fitingof, "Optimal coding in the case of unknown and changing message statistics," Probl. Inf. Transm., vol.2, no.2, pp.3-11 (in Russian), pp.1-7 (English Trans.), 1966.
（4） T.J. Lynch, "Sequence time coding for data compression," Proc. IEEE, vol.54, pp.1490-1491, Oct. 1966.
（5） L.D. Davisson, "Comments on sequence time coding for data compression," Proc. IEEE, vol.54, p.2010, Dec. 1966.
（6） D.A. Huffman, "A method for the construction of minimum-redundancy codes," Proc. of IRE, vol.40, pp.1098-1101, Sept. 1952.
（7） B.P. Tunstall, "Synthesis of noiseless compression codes," Ph.D. Thesis, Georgia Institute of Technology, 1968.
（8） R. Pasco, "Source coding algorithms for fast data compression," Ph.D. Thesis, Stanford University, 1976.
（9） J. Rissanen, "General kraft inequality and arithmetic cod-

ing," IBM J. Res. Dev., vol.20, pp.198-203, 1976.
(10) N. Faller, "An adaptive system for data compression," Record of the 7-th Asilomar Conference on Circuits, Systems and Computers, pp.593-597, 1973.
(11) R.G. Gallager, "Variations on a theme by huffman," IEEE Trans. Inf. Theory, vol.IT-24, no.6, pp.668-675, Nov. 1978.
(12) F. Fabris, A. Sgarro, and R. Pauletti, "Tunstall adaptive coding and miscoding," IEEE Trans. Inf. Theory, vol.42, no.6, pp.2167-2180, Nov. 1996.
(13) P.R. Stubley, "Adaptive data compression using tree codes," Univ. of Waterloo, Dep. of Elec. Eng., PhD thesis, 1992.
(14) J. Rissanen, "A universal prior for integers and estimation by minimum description length," An. Statist., vol.11, no.2, pp.416-431, June 1983.
(15) J. Rissanen, "Universal coding, information prediction, and estimation," IEEE Trans. Inf. Theory, vol.IT-30, no.4, pp.629-636, July 1984.
(16) J.G. Cleary and I.H. Witten, "Data compression using adaptive coding and partial string matching," IEEE Trans. Commun., vol.COM-32, no.4, pp.396-402, April 1984.
(17) F.M.J. Willems, Y.M. Shtarkov, and T.J. Tjalkens, "The context tree weighting method : basic properties," IEEE Trans. Inf. Theory, vol.41, no.3, pp.653-664, May 1995.
(18) J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," IEEE Trans. Inf. Theory, vol.IT-23, no.3, pp.337-343, May 1977.
(19) J. Ziv and A. Lempel, "Compression of individual sequences via variable-rate coding," IEEE Trans. Inf. Theory, vol.IT-24, no.5, pp.530-536, Sept. 1978. (U.S. patent 4,464,650).
(20) M. Burrows and D.J. Wheeler, "A block-sorting lossless data compression algorithm," SRC Research Report 124, Digital Systems Research Center, Palo Alto, CA., May 1994.
(21) H. Yokoo and M. Takahashi, "Data compression by context sorting," IEICE Trans. Fundamentals, vol.E79-A, no.5, pp.681-686, May 1996.
(22) G. Buyanovski "Associative coding," translated by Z. Agazarian Monitor, Moscow, pp.10-19, Aug. 1994.
(23) C.G. Nevil-Manning, I.H. Witten, and D.L. Maulsby, "Compression by induction hierachical grammars," Proc. IEEE DCC'94, pp.244-253, Snowbird, Utah, USA, March 1994.
(24) C.G. Nevil-Manning and I.H. Witten, "Compression and explanation using hierachical grammars," The Comput. J., vol.40, no.2/3, pp.104-116, 1997.
(25) J.C. Kieffer, E. Yang, G. Nelson, and P. Cosman, "Universal lossless compression via multilevel pattern matching," IEEE Trans. Inf. Theory, vol.46, no.4, pp.1227-1245, July 2000.
(26) J. Capon, "A probabilistic model for run-length coding of pictures," IRE Trans. Inf. Theory, vol.5, no.5, pp.157-163, Dec. 1959.
(27) J.L. Bentley, D.D. Sleator, R.E. Tarjan, and V.K. Wei, "A locally adaptive compression scheme," Commun. ACM, vol.29, no.4, pp.320-330, April 1986.
(28) P. Elias, "Interval and recency rank source coding : Two on-line adaptive variable-length schemes," IEEE Trans. Inf. Theory, vol.IT-33, no.1, pp.3-10, Jan. 1987.
(29) 韓 太舜, 小林欣吾, "情報と符号化の数理," 培風館, 1999.
(30) 山本博資, "データ圧縮と正整数のユニバーサル表現," 情報源符号化, 無歪みデータ圧縮, 情報理論とその応用学会 (編), 4章, pp.53-78, 1998.
(31) V.I. Levenshtein, "On the redundancy and delay of decodable coding of natural numbers," Problem of Cybernetics, vol.20, pp.149-155, 1968.
(32) P. Elias, "Universal codeword sets and representations of the integers," IEEE Trans. Inf. Theory, vol.IT-21, no.2, pp.194-203, March 1975.
(33) T. Cover, "Enumerative source coding," IEEE Trans. Inf. Theory, vol.IT-19, pp.73-76, Jan. 1973.
(34) T.A. Welch, "A technique for high-performance data compression," IEEE Comput., vol.17, no.6, pp.8-19, June 1984.
(35) T.C. Bell, "Better OPM/L text compression," IEEE Trans. Commun., vol.COM-34, no.12, pp.1176-1182, Dec. 1986.
(36) E.R. Fiala and D.H. Greene, "Data compression with finite windows," Commun. ACM, vol.32, no.4, pp.490-505, April 1989.
(37) J. Abrahams, "ハフマン符号木に関連した話題," 山本博資 (訳), 応用数理, vol.8, no.2 (データ圧縮特集号), pp.4-20, June 1998.
(38) R.W. Yeung, "Alphabetic codes revisited," IEEE Trans. Inf. Theory, vol.37, no.3, pp.564-572, May 1991.
(39) 横尾英俊, "ユニバーサル情報源符号化のための修正 Ziv-Lempel 符号," 信学論(A), vol.J68-A, no.7, pp.664-671, July 1985.
(40) H. Yamamoto and H. Yokoo, "Average-sense optimality and competitive optimality for almost instantaneous VF codes," IEEE Trans. Inf. Theory, vol.47, no.6, pp.2174-2184, Sept. 2001.
(41) N. Kunihiro and H. Yamamoto, "Window and extended window methods for addition chain and addition-subtraction chain," IEICE Trans. Fundamentals, vol.E81-A, no.1, pp.72-81, Jan. 1998.
(42) N. Kunihiro and H. Yamamoto, "New methods for generating short addition chains," IEICE Trans. Fundamentals, vol.E83-A, no.1, pp.60-67, Jan. 2000.
(43) A. Selçuk and D. Sidhu, "Probabilistic optimization techniques for multicast key management," Comput. Netw., vol.40, pp.219-234, 2002.
(44) D. Salomon, "Data compression, the complete reference," (2nd Ed.), Springer-Verlag, New York, 2000.
(45) 植松友彦, 文書データ圧縮アルゴリズム入門, CQ出版, 1994.
(46) 山本博資, "ユニバーサルデータ圧縮アルゴリズム : 原理と手法," 情報処理, vol.35, no.7, pp.600-608, July 1994.

山本 博資 (正員)

昭50静岡大・工・電気卒. 昭55東大大学院博士課程了. 工博. 同年徳島大・工・電子助手. 同講師, 助教授を経て, 昭62電通大・電子情報助教授. 平5東大・工・計数助教授, 平11同教授, 平13東大・情報理工・数理情報教授, 現在に至る. 平6年度情報処理学会 Best Author 賞受賞. 情報理論, 通信理論, 暗号理論, データ圧縮アルゴリズムなどの研究に従事. IEEE, 情報理論とその応用学会, 日本応用数理学会各会員.