算術符号化

(正会員) 山 下 幸 彦†

† 東京工業大学 大学院理工学研究科

"Arithmetic Coder" by Yukihiko Yamashita (Graduate School of Science and Engineering, Tokyo Institute of Technlogy, Tokyo)
キーワード：算術符号化，情報源符号化定理，エントロピー符号化，画像符号化，ハフマン符号化

ま え が き

算術符号化[1)]は，情報源から送られるデータを平均的に少ないデータ量のデータに変換する情報源符号化法の一つである．算術符号化は，JPEG200のMQ-coder[2)]，MPEG-4 AVCのCABAC (Context-based Adaptive Binary Arithmetic Coding) [3)]に採用され，画像符号化の効率の向上に大きく貢献している．今回はその算術符号化に関して説明する．

算術符号化の目的

本稿では，算術符号化器の入力と出力を0/1のビット列とする．これは画像符号化では一般的である．算術符号化の目的は，入力されるビット列の確率構造を用いて，入力ビット列を平均的に短いビット列に変換して出力することである．シャノンの情報源符号化定理により，符号化法を工夫するとき，出力平均ビット長を入力情報源のエントロピーに近づけることができる．算術符号化でも，出力ビット長がそのエントロピーに近くなるように符号化することが必要である．

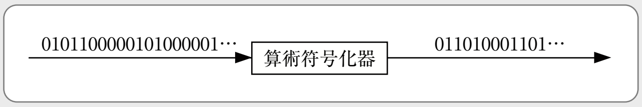

図1　算術符号化器の役割

復習：2進数小数

算術符号化器の出力は，2進数小数を表していると考えることができる．2進数では，小数点第1位で1=1/2を，小数点第2位で1/2²を表す．例えば，2進数で0.101101を10進数に変換すると，

$$1\times\frac{1}{2}+0\times\frac{1}{2^2}+1\times\frac{1}{2^3}+1\times\frac{1}{2^4}+0\times\frac{1}{2^5}+1\times\frac{1}{2^6}=\frac{45}{64}$$

となる．一般にビット列の先頭に0.を付け加えれば，ビット列は区間 [0.0, 1.0) に含まれる2進数小数を表していると考えることができる．

算術符号化の原理

算術符号化の原理を説明する．nビットの入力を符号化するとき，入力のビット列は2^n通り存在する．そして，それぞれのビット列に対して，[0, 1)に含まれる区間を重複しないように割当てる．例えば，n=3とすれば，シンボル列は000, 001, 010, 011, 100, 101, 110, 111の8通りで，$0<a_{001}<a_{010}<...<a_{011}<1$として，それぞれのビット列に，$[0, a_{001})$，$[a_{001}, a_{010})$，$[a_{010}, a_{011})$，...，$[a_{110}, a_{111})$，$[a_{111}, 1)$を対応させる．この区間に関する情報は，符号化器と復号器の両方で共有している必要がある．

算術符号化器は，入力ビット列が対応する区間に含まれる2進数小数を一つ出力する．復号器ではその小数から区間がわかるので，入力されたビット列を知ることができる．出力のビット長を短くするためには，区間に含まれる小数の中で，できるだけ小数点以下の桁数が少ない小数を出力すれば良い．一般に，区間長が2^{-m}以上である場合，その区間は高々桁数がmである小数を含んでいる（それより少ない桁数の小数を含むこともある）．したがって，高々mビットでその区間，すなわち，入力ビット列を表すことができる．

各入力ビット列の区間長をその出現確率p_iになるようにすれば，それを$\lfloor \log_2 p_i \rfloor$（$\log_2 p_i$以上の最小整数）のビット長で表すことが可能である．したがって，平均の出力ビット長は，入力のエントロピーに近くなる．0と1が入力される確率が，それぞれ0.7と0.3と一定で，0に対する区間を下側から割当てるものとする．3ビットの入力に対するそれぞれの区間は図2のようになる．例えば，101に対応する区間は，[0.847, 0.91)になる．

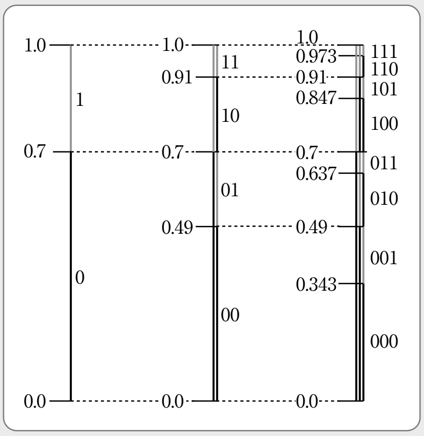

図2　算術符号化の原理
（無限精度での区間分割）

区間の再分割

上の例において，もう1ビットの入力が加わり，4ビットの入力になったとする．ハフマン符号化では，4ビットの入力に最適化した符号を出力するためには，符号化テーブルを初めから作り直すことが必要である．算術符号化ではその必要はなく，例えば，010の後に入力があった場合，区間$[a_{010}, a_{011})$を再分割すれば良い．bを$a_{010}<b<a_{011}$となる数として，入力0100と0101に，それぞれ区間$[a_{010}, b)$と$[b, a_{011})$を割当てる．このとき，この分割を4ビット目のシンボルの出現確率に比例して分割すれば，最適な符号化が可能になる．

有限精度の問題

入力ビット長が増加すると，区間を表すために多数の桁の小数点数の保存と計算が必要になり，実際に符号化することが困難になる．したがって，近似により固定長の整数で実現できるようにする必要がある．そのために複数の方法が提案されているが，ここではレンジコーダ[4]の基本部分を使って説明する．通常は16ビット程度で計算を行うが，ここでは[0.0, 1.0)を[0, 32)の5ビットで表した場合において説明する（**図3**）．

初めの入力に対して，全区間長32を0と1の出現確率0.7と0.3の比で分割して，0と1の区間をそれぞれ[0, 22)と[22, 32)とする．次に0が入力されたとき，[0, 22)を ☞

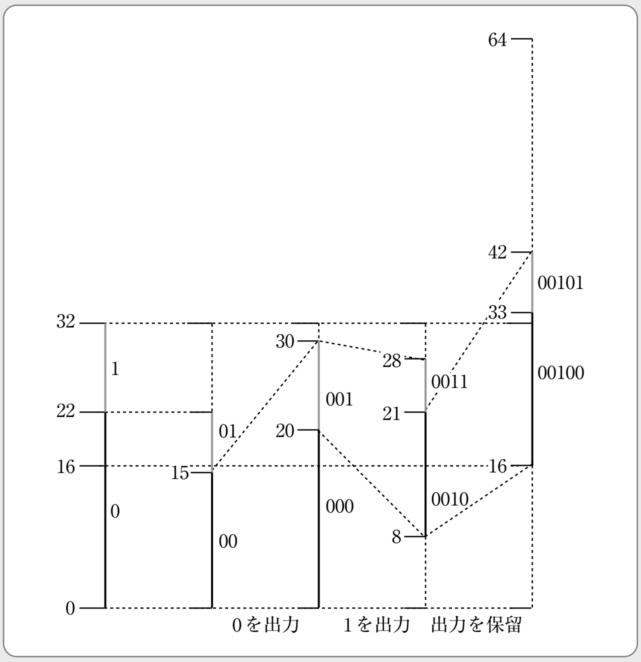

図3　算術符号化の実装（有限精度での区間分割）

☞ 0.7対0.3の比で分割して，00と01の区間をそれぞれ［0, 15）と［15, 22）とする．このように分割していくと，区間長が短くなってしまい，区間長が1になった時点で，それ以上の分割が不可能になる．そのため，区間長が16以下になったときに，区間を2倍に拡大する．

例えば，00と入力された場合の区間は，［0, 16）に含まれている．その後のどのような入力に対しても，入力に対応する区間は，小数で表せば［0.0, 0.5）に含まれるため，先頭ビットは0であることが確定している．したがって，0を出力し［0.0, 0.5）を［0, 32）で表すようにする．その結果，先の整数値を2倍にした区間［0, 30］が，次の入力のために分割する区間となる．

逆に，001の次に1が入力されたときは，区間が［20, 30）となり，小数で表した区間［0.5, 1.0）に含まれ，先頭のビットは1であることが確定する．この場合は1を出力し，16を引いて2倍した［8, 28）が次の入力のために分割する区間となる．

問題となるのは，区間長が16以下であるが，その区間が16を含むときである．このときは先頭ビットが0か1が確定しないため，出力を保留する．そして，その区間を2倍する．出力の保留中は区間［0, 64）で計算が行われる．その後の計算で分割した区間が

(a)［0, 32）に含まる場合：保留した出力ビットが0であることがわかり，0を出力し保留を解除する．逆に，

(b)［32, 64）に含まる場合：それが1であることがわかり，1を出力し区間から32を引いて保留を解除する．

区間を分割し長さが16以下になったとき，その区間が32を含む場合は，はじめに保留した出力を確定することができないため，再度保留する．この場合には，区間から16を引いてから2倍し，区間［0, 64）に収める．保留が複数回続くときも，保留される出力は，01…1または，10…0（ビット長は保留された回数に等しい）のどちらかである．したがって，(a)または(b)の条件で保留を解除したときに，それぞれを出力すればよい．

Qコーダ

レンジコーダでは，入力1ビットごとに区間と確率の乗算が必要であり，計算負荷が高い．Qコーダ[5]ではこの乗算を省略する．上の例では，確率1に対応する区間長は17から32の間である．そのため，確率1に対応する区間長を，その中間程度の（32/1.5）と決めてしまう．そして，入力1に対応する区間長を，0.3×32/1.5≒6に固定する．区間長の比が確率の比とはならないため，出力ビット長が長くなることが予想されるが，実際には大きな問題にならないと言われている．また，Qコーダは，適応符号化を採用し，符号化効率が高く，実装面でも工夫し，計算負荷が小さくなっている．JPEG 2000で使われているMQコーダは，Qコーダの改良版である．

む す び

近年の画像符号化器に採用されている算術符号化に関して，その原理と実装を説明した．さらに詳しく知りたい場合は，参考文献等を読むとよい．また，東京工業大学山下研究室内で作成した，画像符号化を説明するためのスライドと教育用のレンジコーダのJavaプログラムを，http://www.ide.titech.ac.jp/~yamasita/yylab/ITE2011.zipに置く．

（2011年10月18日受付）

参 考 文 献

1) G. G. Langdon: "An Introduction to arithmetic coding", IBM Journal of Research and Development, **28**, 2, pp.135-149 (Mar. 1984)

2) 小野定康，鈴木純司：“わかりやすいJPEG2000の技術”，オーム社（Mar. 2003）

3) 角野眞也，菊池義浩，鈴木輝彦，大久保榮：“H.264/AVC教科書（改訂三版）”，インプレスR&D（Dec. 2008）

4) G.N.N. Martin: "Range encoder range encoding: An algorithm for removing redundancy from a digitized message", Video & Data Recording Conference, 43, Southampton, UK, July 2427（1979）

5) W.B. Pennebaker, J.L. Mitchell, G.G. Langdon, R.B, Arps: "An overview of the basic principles of the Q-Coder adaptive binary arithmetic coder", IBM Journal of Research and Development, **32**, 6, pp.717-726（Nov. 1988）

山下（やました） 幸彦（ゆきひこ） 1985年，東京工業大学大学院理工学研究科修士課程修了．同年，日本原子力研究所入所．1988年，(株)アイザック入社．現在，東京工業大学大学院理工学研究科にて，画像処理，パターン認識，集合知活用型システムの研究に従事．博士(工学)．正会員．