目次

★4　パターン情報の表現 (3) — 情報量とエントロピー符号化

★4.1　情報量の概念

★4.1.1　情報量の定義

ある事象が起こったことを知らせる際に伝達される「情報の量」を考えてみよう．「1 億枚中 1 枚しか当たりのないくじではずれをひいた」という知らせよりも「そのくじで当たりをひいた」という知らせの方が情報が多い気がしないだろうか (☆ 1)．「犬が人を噛んだ」というニュースよりも「人が犬を噛んだ」という方が情報が多い気がしないだろうか．実は，このような素朴な議論から出発して，**情報量**というものを次のように定義するとよいことが知られている．

☆ 1) 「びっくり度」が高い，と言ってもよいかも．

ある事象 E の生起確率が $P(E)$ であるとき，E が起こったことを知らせる際に伝達される情報量 $I(E)$ を次式で定義する：

$$I(E) = \log\left(\frac{1}{P(E)}\right) = -\log P(E) \tag{1}$$

対数の底は任意に定めればよい (☆ 2) が，bit との対応づけができるため，2 を選ぶことが多い．その場合，情報量の単位は [bit] となる．

☆ 2) ある二つのことの情報量に注目すると，底の選び方を変えてもそれらの大小関係は変化しない．ただし底の選び方によって情報量の値そのものは変化する．

例： 確率 $\frac{1}{2}$ で赤旗か白旗のいずれかが上がるのを観測する場合，「赤（白）の旗が揚がった」という知らせの情報量は $-\log_2 \frac{1}{2} = 1$[bit] となる (☆ 3)．

☆ 3)「赤旗が揚がった」と「白旗が〜」の 2 つに 2 進数を対応づけるためには 1 桁 (bit) の 2 進数が必要であるということに対応している．たとえば，0 が「赤旗が〜」で 1 が「白旗が〜」．

例： 天気が「晴れ」か「雨」のどちらかにしかならない地域があったとする．晴れの確率は 0.9 だという．この場合，この地域の天気が晴れだということ，および雨だということを知らせる情報の情報量は次のようになる．

$$I(\text{晴れ}) = -\log_2 0.9 = -\frac{\log 0.9}{\log 2} = \text{約 } 0.152\text{[bit]}$$

$$I(\text{雨}) = -\log_2 0.1 = -\frac{\log 0.1}{\log 2} = \text{約 } 3.32\text{[bit]}$$

例： 4 つの事象 A, B, C, D のどれかが起こるとして，それらの生起確率が等しい（つまり $\frac{1}{4}$ である）場合，どの事象が起こったかを知らせる情報の情報量は 2[bit] となる (☆ 4)．

☆ 4) A, B, C, D に対応させるには 2 桁 (bit) の 2 進数が必要．

Q1. 10 通りの事象のいずれかが同じ確率で起こる場合，どの事象が起こったかを知らせる情報の情報量は何 bit か．必要ならば $\log_2 5 = 2.32$ を用いたらよい．

Q2. 「晴れ」，「曇り」，「雨」，「雪」の 4 通りの天気のいずれかが起こる地域があったとする．この地域の雪の確率が $\frac{1}{1024}$ だとすると，この地域の天気が雪だということを知らせる情報の情報量は何 bit になるか．

★ 4.1.2 情報量の性質

例： $4 \times 13 = 52$ 枚のカードから成るトランプから無作為に 1 枚引くなら

- 「それが ♡ だった」という知らせの情報量は $-\log \frac{1}{4} = \log 4$
- 「エース (1) だった」という知らせの情報量は $-\log \frac{1}{13} = \log 13$
- 「♡ のエースだった」という知らせの情報量は $-\log \frac{1}{52} = \log 52$

となる．つまり，$I(♡) + I(\text{エース}) = I(♡ \text{ かつエース})$ である (☆ 5)．

ここから予想できるように，情報量には次のように**加法性**が成り立つ (☆ 6)：

事象 A の情報量が $I(A)$, 事象 B の情報量が $I(B)$ であるとき，A, B が独立ならば，事象 $A \cap B$ の情報量（A も B も起こったことを知らせる）は

$$I(A \cap B) = I(A) + I(B) \tag{2}$$

となる (☆ 7)．

☆ 5) ここでは対数の底を e としているが，他の値にしても結論は変わらない．

☆ 6) 実際の理論の成り立ちは逆で，「素朴に考えると情報量というのは『生起確率の単調減少関数』でありかつ『加法性が成り立つ』ものでなければならない」というのを出発点にして，そのような性質を満たすのは式 (1) のような対数の形でなければならないことが導出されている．

☆ 7) 「ハートだった」を A，「赤いのだった」を B としてみると，どうなるだろう．

Q3. さいころを振って出た目が「1 か 2 だった」，「奇数だった」，「1 だった」ということを知らされることで得られる情報量をそれぞれ求めなさい．これらの間にはどのような関係があるだろう（ヒント：「1 か 2 だった」かつ「奇数だった」というのはどんな場合？）．

★ 4.1.3 エントロピー

n 個の事象 $E_1, E_2, \ldots, E_n$ がそれぞれ確率 $p_1, p_2, \ldots, p_n$ で生起する場合を考える．$\sum_{i=1}^{n} p_i = 1$ とする．このとき，事象 E_i が「起こった」という知らせの情報量 $I(E_i)$ は $I(E_i) = -\log p_i$ である．それでは，「$E_1, E_2, \ldots, E_n$ のどれが起こったかまだ知らない状況でどれが起こったかを知らせてもらう場合，その知らせは平均としてどれだけの情報量をもつと期待できるか」を考えてみよう．その値を H とおくと，H は得られる情報量の期待値であるから，

$$H = \sum_{i=1}^{n} p_i I(E_i) = -\sum_{i=1}^{n} p_i \log p_i \tag{3}$$

となる．これは，「どれが起こったか知らない不確定な状況を確定させることで得られる情報量の平均値」を表しており，**平均情報量**あるいは**エントロピー**と呼ばれる．「知ろうとしている状況の不確かさの度合い」を表していると考えてもよい．

エントロピーについてはいろいろ興味深い話があるが，この授業では割愛する．

ただし，$\lim_{p \to +0} p \log p = 0$ より，$p = 0$ の場合の $p \log p$ の値は 0 として扱う．

エントロピー: entropy. 熱力学でててくるエントロピーと同じようなものと考えられる．

例: 確率 p で表が，確率 $1-p$ で裏が出るコインがある．このコインを投げたときに得られるエントロピーは，(底を 2 とすると)

$$H = p \times I(\text{表}) + (1-p) \times I(\text{裏}) = -p \log_2 p - (1-p) \log_2 (1-p) \quad [\text{bit}] \tag{4}$$

となる．右図からわかるように，このエントロピーは $p = 0$ または $p = 1$ のときに 0（投げる前から結果がわかっている）となり，$p = \frac{1}{2}$ のときに最大 (1[bit]) となる（最も不確かな状況）．これは，「確率に偏りがある ⇔ エントロピーが小さい ⇔ 不確かさが小さい」ということを意味している．

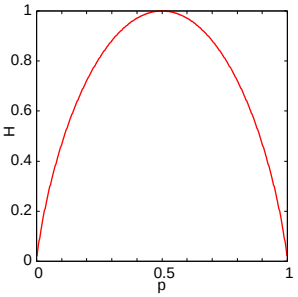

図: コイン投げのエントロピー

★ 4.2 エントロピー符号化

データ圧縮の手法として，エントロピーの概念に基づく**エントロピー符号化**という符号化法が知られている．

★ 4.2.1 エントロピー符号化の考え方

表に示す確率で 6 つの記号のいずれかが出現する (☆ 8) 記号の並びを 0,1 で符号化したいとする．

☆ 8) 記号の出現の仕方は記号列中の位置や並び方に無関係と想定している．

記号	D	E	F	G	H	O
確率	$\frac{10}{100}$	$\frac{2}{100}$	$\frac{15}{100}$	$\frac{13}{100}$	$\frac{20}{100}$	$\frac{40}{100}$

最も単純な方法は，各記号を 3bit の符号で表現する（記号 D に 000，E に 001，等），というものである．この場合，たとえば `HOGEHOHOOO` という 10 文字の並びは，$10 \times 3 = 30$bit のビット列となる．この例では，どの記号にも 3bit を割り当てることになるから，「1 つの記号を平均何 bit で表せるか」を表す**平均符号語長**を考えると，その値は 3bit となる．

しかし，これらの記号の出現頻度には偏りがあるのでエントロピーすなわち平均情報量はもっと小さいと考えられる (☆ 9)．この例の場合に実際にエントロピーを計算してみると，

☆ 9) 6 通りの記号が等確率で出現する場合でもエントロピーは $-6 \times \frac{1}{6}\log_2\frac{1}{6} = \log_2 6 =$ 約 2.58[bit] である．

$$H = -\frac{10}{100}\log_2\frac{10}{100} - \frac{2}{100}\log_2\frac{2}{100} - \cdots - \frac{40}{100}\log_2\frac{40}{100} = \text{約 } 2.23[\text{bit}] \quad (5)$$

となる．このことは，符号の割り当て方を工夫すれば，平均符号語長をここまで小さくできる可能性があることを意味している．たとえば，右表のように出現確率の高いものに短い符号を割り当てるようにしてやると，平均符号語長を

記号	符号
D	1001
E	1000
F	110
G	101
H	111
O	0

$$\frac{10}{100}\times 4 + \frac{2}{100}\times 4 + \frac{15}{100}\times 3 + \frac{13}{100}\times 3 + \frac{20}{100}\times 3 + \frac{40}{100}\times 1 = 2.32[\text{bit}] \quad (6)$$

にすることができる．実際に上記の 10 文字の並びの例でやってみると，

$$11101011000011101111000 \quad (7)$$

となり，30bit だったものを 21bit に減らせている．

このように，エントロピー符号化とは，記号の出現頻度（確率）に偏りがある，すなわちエントロピーが小さい場合に，その偏りを利用して効率的な（平均符号語長の短い）符号化を実現しようとするものである．上記の例ではエントロピーが約 2.23bit であるから，理想的なエントロピー符号化ができれば平均符号長をそこまで減らせることを意味している．

Q4. 式 (7) の符号を表に従って先頭から順に復号していくとちゃんと元通りになることを確認してみよう．

★ 4.2.2 ハフマン符号化

エントロピー符号化の一種である**ハフマン符号化**は，記号毎に算出した出現頻度を用いて，次のような手順で木（ハフマン木）を作ることで，それぞれの記号に割り当てる符号を求める（授業中にもう少し詳しく説明します）．

ハフマン符号化: Huffman coding．1950 年代前半に D. A. Huffman が提案した符号化法．画像圧縮方式 JPEG でも，離散コサイン変換（そのうち紹介するかも）したデータの符号化に用いられている．

1. 個々の記号に対応した葉節点をつくる．出現確率をそれぞれの値とする．
2. 親のいない節点の中で最も小さい値をもつもの 2 つを選び，それらの親となる節点をつくる．親節点の値は，2 つの子節点の値の和とする．
3. 親を持たない節点が一つになるまで 2. を繰り返す．
4. 「左」を 0,「右」を 1 として，できた木の各節点から左右に接続した枝のそれぞれに 0,1 を割り当てる（「左」を 1 としてもよい）．
5. 根から葉までたどる際に現れる 0,1 のならびをそれぞれの葉の記号に対応した符号とする．

ハフマン符号化では，符号化時だけでなく復号時にもハフマン木が必要となる．したがって，符号化したデータにハフマン木そのものも添えてやらねばならない．長いデータの場合には問題にならないが，あまり短いデータに適用すると，そのために元よりデータ量が増えてしまうこともある．

Q5. 上の（授業中に説明した）Huffman 木を用いて，次のデータを復号しなさい．1011000111010101110

Q6. 64 種類の記号が等確率で出現する長い記号列があったとする．この記号列にハフマン符号化を適用しても，有効なデータ圧縮はできない．その理由を述べなさい．

★ 4.2.3 よだんだよん

- エントロピー符号化は記号の出現頻度の偏りを利用する手法であるから，記号の並び方（同じ記号が連続するなど）は圧縮の性能に影響しない．
- エントロピー符号化の方法としては，Huffman 符号化よりも高効率な算術符号化という方法もある．
- 符号化は何もデータ圧縮のためだけに行なうものではない．雑音等のせいで送信データに誤りが生じてしまう状況で誤りを見つける（誤り検出）／誤りを訂正する（誤り訂正）ためにも用いられる (☆ 10)．最も原始的な誤り検出の符号化法は，ビット列の最後に，ビット列中の 1 の数が常に偶数になるように 0 か 1 を付け加える，というものである．興味のある人はいろいろ調べてみるとおもしろいかも．
- 符号化のさらに別の目的として，暗号化がある．

☆ 10) 情報通信機器では当たり前に使われている．図書の ISBN，様々な商品の JAN コード (バーコードの元になってるコード）などにも使われている．

情報理論の参考書: 高橋の手元には「情報理論」甘利俊一，ダイヤモンド社，ISBN4-478-82000-7，といういい本があるのですが，残念ながら現在は手に入らないようです (☆ 11)．他にいい本ないか探索中．
データ圧縮の参考書:「圧縮アルゴリズム 符号化の原理と C 言語による実装」昌達 K'z，ソフトバンクパブリッシング，ISBN4-7973-2552-6

☆ 11) 最近，ちくま学芸文庫で復活しました．

宿題

ある記号列を調べたところ，6 通りの記号が次表のような頻度で出現することがわかった．このような記号列をハフマン符号化法でビット列に符号化するとして，次の問に答えなさい（答えのみでよい）．

記号	0	1	2	3	い	か
頻度	0.55	0.03	0.2	0.01	0.07	0.14

(1)–(6) 上記の表にもとづいてハフマン木を求めると，右図のようになった．(1) から (6) までの各葉節点に割り当てられる記号を答えなさい．ただし，このハフマン木は，授業中に説明したものと同様に，子をもつ全ての節点について，(左の子の値) < (右の子の値) となるように描かれているものとする．

(7) 右図のハフマン木の子をもつ全ての節点について，左の子にむかってのびた枝に符号 0 を，右の子にむかってのびた枝に符号 1 を割り当てたとする．このハフマン木を用いて，次のビット列を記号列に復号しなさい．

100011101000

(8) ハフマン符号化法はデータ圧縮のためによく用いられるが，データの性質によっては有効でない（圧縮率が高くならない）ことがある．それはどのような場合か，理由も含めて述べなさい．