目次

★ 3　パターン情報の表現 (2) — データ圧縮

3.1　パターンのデータ量を見積もってみよう

パターンを表現するディジタルデータのデータ量がどれくらいになるのか見積もってみよう．CD に記録された音楽データの例，画像の例（授業中に示します）．

3.2　データ圧縮とは

上で説明した例からもわかるように，パターンを表現するディジタルデータのデータ量は非常に大きくなりがちである．そのようなデータをそのまま記録したり通信したりするのでは記憶容量や通信帯域が勿体ないので，**データ圧縮**を行うことが一般的である．

データ圧縮とは，あるデータが与えられたときに，それが含む情報の性質を損なわないでよりデータ量の小さいデータに変換する処理のことである (☆ 1)．あるデータをより小さいデータ量のデータに変換することを「**圧縮**する」といい，圧縮されたデータを元のデータに戻すことを「**伸長**する（あるいは**展開**する）」という．「**符号化**する／**復号**する」ということもある (☆ 2)．元のデータと圧縮したデータのデータ量を比べて，前者より後者が十分小さければ「**圧縮率**が高い」といい，そうでなければ（あまり変わらなければ）「圧縮率が低い」という．

勿体ない: mottainai.
データ圧縮: data compression

☆ 1) **データ圧縮の対象はパターンに限らないことに注意．多様な種類の情報に適用される，非常に重要な情報処理の方法である．**

☆ 2) 本来，「符号化」には標本化・量子化のような処理も含む広い意味がある（第 3 回「よだんだよん」も参照）が，「圧縮」の意味で使われることも多い．ちなみに，「復号」は動詞なので「化」はつけない．俗に「解凍する」ということも．

3.3　データ圧縮手法の分類

データ圧縮の手法には様々なものがあり，いろいろな観点から分類することができる．例えば，全てのデータ圧縮手法は次の二つに分けられる．

可逆圧縮　逆が可能，つまり，圧縮したデータを伸長したら元のデータと完全に同じものが得られるようなデータ圧縮の方法．プログラムやテキストデータを対象とするなら一般にこちらを用いる．

不可逆圧縮　伸長しても元のデータを完全に再現できるとは限らない方法．その分，可逆圧縮よりも高い圧縮率を実現できる．音声や画像といったパターン等，情報の欠落や変化が多少あっても許容できる性質のデータに用いられる．

身近なところでは，コンピュータ上のファイルを圧縮するツール (☆ 3) は可逆圧縮を行なっている．静止画像のファイル形式では，PNG は可逆圧縮であるが，JPEG は不可逆圧縮である (☆ 4)．映像データの形式である MPEG や，オーディオデータの形式である MP3 なども不可逆なデータ圧縮方式を採用している (☆ 5) .

可逆圧縮: lossless compression, 不可逆圧縮: lossy compression

☆ 3) `compress`,`gzip`,`bzip2` といった UNIX 系 OS のコマンドや，MS-DOS や Windows で良く用いられる LHA などのソフトウェア，ZIP 形式を扱うツールなど．

☆ 4) PNG: Portable Network Graphics. JPEG: Joint Photographic Experts Group（この方式を作った組織の名前でもある）．JPEG は可逆手法と不可逆な手法を組み合わせているので全体として不可逆．ただし，可逆圧縮のみ行うことも可能．

可逆圧縮の手法は，次の二種類に分けることができる．

- **データ値の並び方を利用する手法**: 同じ値が連続している，近い値が連続している，文章中の単語のように一定の並びが繰り返し出現する，などといった性質を利用する．後述のランレングス圧縮や，LZ 符号化 (☆ 6) などが代表的．
- **データ値の現れ方の偏りを利用する手法**: 次回登場予定のエントロピー符号化法（ハフマン符号化や算術符号化が代表的）など．

いずれの手法も，データの種類（画像であるとか音声であるとかテキストであるとか）によらず幅広く用いられる．不可逆圧縮したデータをさらに圧縮するのに用いることも多い．

一方，不可逆圧縮の手法は，対象とするデータの種類を（静止画像向け，音響信号向け，などのように）限定し，データの性質を利用して高い圧縮率を実現しているものが多い．可逆圧縮手法のように一般的な性質でもってそれらの手法を分類することもできるが，割愛する．

☆ 5) MPEG: Moving Picture Experts Group. MPEG-1, MPEG-2 など様々な規格がある．DVD-Video は MPEG-2 をベースとしている．MP3 はもともと MPEG-1 のオーディオデータの規格.

☆ 6) 1970 年代後半に Ziv と Lempel が提案した手法．様々に改良され，現在も幅広く使われている．有名な LHA もこれを用いる圧縮ツールの一つ.

Q1. 自分の持っている画像データがどんな形式で保存されているか調べてみよう．ウェブ上の画像がどんな形式か調べてみよう．単純には，そのファイルの拡張子を見ればよい (☆ 7)．おそらく，JPEG や PNG が多いだろう．それぞれどのような画像に使われることが多いか，どのような画像に向いているかを調べてみよう．

☆ 7) 拡張子とファイル形式との対応は，ネットや書籍で調べてみよう．ただし，拡張子が間違っている／わざと変えてある（初歩的なコンピュータウィルスなどの手口）こともあるが.

Q2. 計算機室の Linux 環境で，適当なファイルを圧縮してみよう．

```
$ ls -l hoge      ← 圧縮前のサイズ（バイト単位）を調べる
$ gzip hoge       ← hoge を圧縮．hoge.gz というファイルができる
$ ls -l hoge.gz   ← 圧縮後のサイズを調べる
```

伸長の仕方は，

```
$ man gzip
```

して調べよう．ファイルの種類（プログラムのソース，実行形式，JPEG 画像，etc.）によって圧縮率が違ったりするだろうか．

Q3. 画像を扱える適当なソフトウェア（フリーで手に入るものも多い）を用いて，適当な画像を様々な形式で保存してみよう．圧縮率をいじれるならば，ファイルサイズと見た目がどのように変化するかいろいろ試してみよう．

計算機室の Linux 環境なら，ImageMagick (画像処理ツール群) が使える．ImageMagick の `display` コマンドを使って

```
$ display hoge.jpg
```

とすれば画像 `hoge.jpg` を画面に表示できるし，そこからマウスクリックしてメニューを開いて画像形式を選択して保存することもできる．ImageMagick には他にも画像の大きさや形式を変換する `convert` などがある．

3.4 データ圧縮の例—ランレングス圧縮

ある地域の天気を 1 日毎に「晴」「曇」「雨」「雪」「霙」(☆ 8) などの 8 種類に分類したデータがあったとしよう．例えば，

晴 晴 晴 晴 晴 晴 晴 曇 曇 雨

といったものである．このように同じ値が連続して現れることの多いデータの場合，値とその連続する数をペアにして

晴 7 曇 2 雨 1

のように表すと，データ量を減らせそうである．このようなアイデアに基づくデータ圧縮の手法を，**ランレングス圧縮**（ランレングス符号化）という．

「減らせそう」でごまかさないで具体的に考えてみよう．この例では，1 日の天気の値を 3bit で表現することができる．その場合，元のデータを表すためのデータ量は $3 \times 10 = 30$[bit] となる．一方，ランレングス圧縮したデータの方は，値だけでなくその連長も符号化する必要がある．例えば長さも 3bit で表現するとしたなら (☆ 9)，データ量は $(3+3) \times 3 = 18$[bit] ということになる．したがって，この例ではランレングス圧縮を行なうことでデータ量を元の 6 割にまで削減できている．

ランレングス圧縮は，**二値画像**（白と黒の二通りの画素で構成された画像）を扱うのに特に適しているので，ファクシミリ等に応用されている（例えば文書などを読み取って二値画像にすると，白画素ばかりになるから）．例えば右図の画像 の画素値を左上から右に向かって順に符号化していくと「白 4 黒 1 白 7 黒 1 白 1 黒 7 白 5 黒 1 白 5 黒 1 白 2」のように表せるが，二値かつ白から数えると仮定すれば，さらに省略して「4 1 7 1 1 7 5 1 5 1 2」としてもちゃんと復号できる．

上述のことからわかるように，ランレングス圧縮のアルゴリズムは非常に単純である (☆ 10) しかし，単純なランレングス圧縮の方法には次のような問題点がある

- 同じ値があまり続かない場合には逆にデータ量を増やしてしまう
- 同じ値が長く続くからといってその長さをそのまま符号化すると，長さを表すために大きな bit 数を割り当てる必要が生じて圧縮率が落ちてしまう

そのため，実用的にはもう少し凝ったアルゴリズムを考える必要がある (☆ 11).

Q4. 以下は，7×7 の格子状に並んだ画素の値（白か黒）をランレングス圧縮して得られるデータである．ただし，左上の画素から右に向かって符号化してある（最初は白の数）とする．これを伸長するとどんなパターンが得られるか．(^_-)
815111137151158

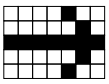
3.5 データ圧縮の参考書

「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

☆ 8) 霙：みぞれと読むそうです．

ランレングス: run-length. 連長ともいう．

☆ 9) 3bit なら 1 から 8 までの長さを表せる．



☆ 10) プログラミングの好きな人は，例えば，整数のならびをキーボードから入力するとそれをランレングス圧縮したものを表示するプログラムを考えてみると楽しいでしょう．手頃な練習問題になります．

☆ 11) 興味のある人は，データ圧縮関連の文献などを調べてみるとよい．下記の参考文献には，これらの問題点に対応するための改良版ランレングス圧縮を含め各種データ圧縮手法の C プログラムが載っている．

3.6 情報量の概念

3.6.1 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 (☆ 12)．「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．

☆ 12) 「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 (☆ 13) 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．

☆ 13) 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 (☆ 14)．

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．

$$I(\text{晴れ}) = -\log_2 0.9 = -\frac{\log 0.9}{\log 2} = \text{約 } 0.152\text{[bit]}$$
$$I(\text{雨}) = -\log_2 0.1 = -\frac{\log 0.1}{\log 2} = \text{約 } 3.32\text{[bit]}$$

☆ 14) 「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： 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 (☆ 15)．

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

Q5. 20 通りの事象のいずれかが起こるとする．それらの生起確率が等しい場合，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．

Q6. 「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 $\frac{1}{512}$ 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．

3.6.2 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 (☆ 16)．

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 (☆ 17)：

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 (☆ 18)．

Q7. さいころを振って出た目が「3 の倍数だった」,「偶数だった」,「6 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．ただし，単位は bit とすること．これらの間にはどのような関係があるだろう（ヒント：「3 の倍数」かつ「偶数だった」というのはどんな場合？）．

☆ 16) 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．

☆ 17) 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．

☆ 18) 「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．

宿題

この資料の 3.6 を読み，Q5, Q6, Q7 をやりなさい．