

情報理論の基礎

- 自己情報量 (self information)
- 平均情報量 (average information)
- エントロピー (entropy)
- 最大エントロピー (maximum entropy)
- 冗長度 (redundancy)

可逆圧縮の原理と実践

- ランレングス符号化
- ハフマン符号化 (エントロピー符号化)

そもそも情報とはなに？

「ある事柄に関して知識を得たり判断のより所としたりするために不可欠な、何らかの手段で伝達（入手）された種々の事項（の内容）」

（新明解国語辞典 第6版、三省堂より一部抜粋）

コンピュータで取り扱う情報の定義

・Claud Elwood Shannon（情報理論の発案者）

「変化するパターンの中から選択できるもの」

例：T子さんが結婚する（未婚→既婚への変化）

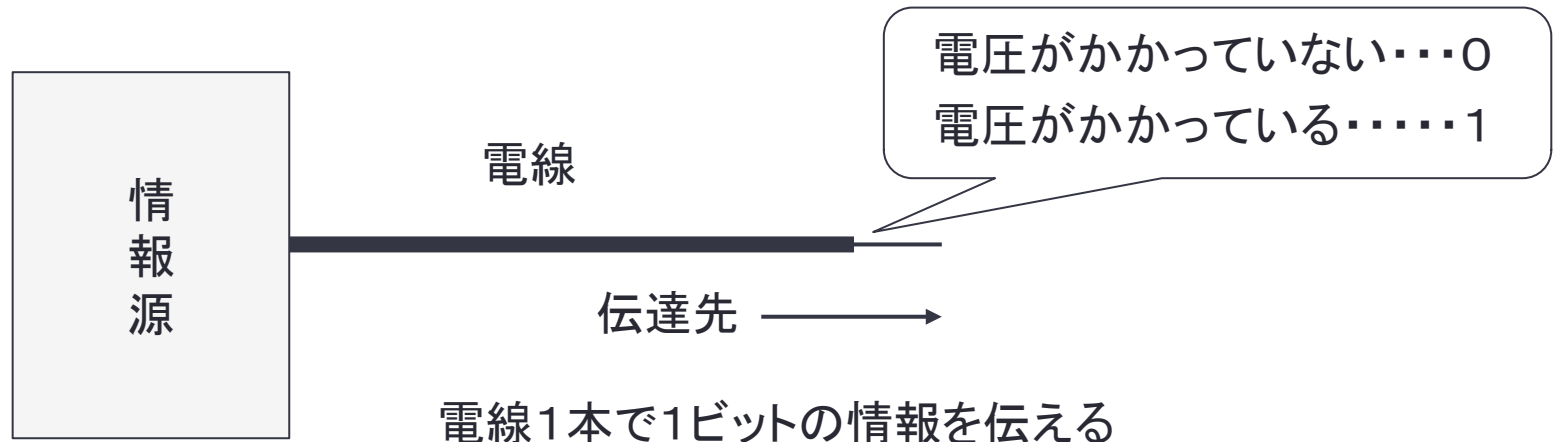
ここで問題です①

Q. 「オバケのQ太郎」という漫画には、毎日三食とも必ずラーメンを食べている小池さんというキャラクターが登場します。小池さんが今日何を食べたかは、情報と呼べるでしょうか？

A. 小池さんは、いつでも必ずラーメンを食べているのですから、まったく変化がありません。したがって、小池さんが今日何を食べたかは、情報とは呼べません。もしも、小池さんの食事が、「ラーメンを食べる／カレーを食べる」のように変化するなら情報と呼べます。

情報の最小単位

天気という情報は、晴れ、曇り、雨、雪の4通りに変化します。道路信号は、青、黄、赤の3通りに変化します。それでは、最も少ない変化は、何通りでしょう。YES/NO、男/女、前/後のような2通りの変化です。すなわち、2通りの変化が情報の最小単位であり、これを「ビット」と呼びます。ビット(bit)はbinary digit(2進数)を略した言葉です。



情報の基本単位

コンピュータは、8本の電線をセットで使った 8 ビットを、情報の基本単位としています。これを「バイト」と呼びます。8 ビット = 1 バイト (byte) です。



最近のコンピュータは
64ビットCPU (central
processing unit: 中央
処理装置) が搭載され
ている → 1800京通り

8本 = 1バイトで表せるパターンは256通り、すなわち256通りの変化を処理できる

1024 byte (B) = 1 KB (キロバイト)

→ 2^{10} byte

1024 KB = 1 MB (メガバイト)

→ 2^{20} byte

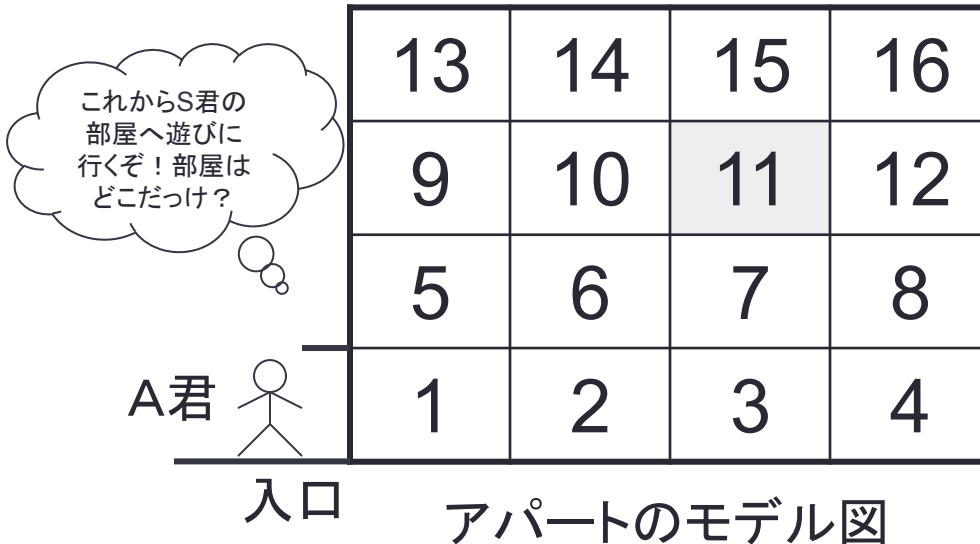
1024 MB = 1 GB (ギガバイト)

→ 2^{30} byte

1024 GB = 1 TB (テラバイト)

→ 2^{40} byte

情報の量とは、その情報を得た人にとっての内容の豊富さのことであると考えてよい。



A君は入口で次の情報を教えてもらった

情報: S君は11号室に住んでいる

この情報はA君にとって果たしてどれぐらいの情報の量であったのか。

Q. ケース1とケース2では、どちらが入口で得た情報の量が多い(豊富)と考えられるか？

→ 1/16

Case 1: A君はS君が何階に住んでいるのか知らなかった。

Case 2: A君はS君が3階に住んでいるのを知っていた。 → 1/4

- 情報の量は、情報を得る前の可能性の数（前例では部屋の数）に関係し、その数が増すにつれ情報の量も増すようであればならない（単調増加関数）。
- 情報の加法性が成り立たなければならない。
 - 情報1：S君の部屋は16室の11号室である。
 - 情報2：S君の部屋は3階にある。
 - 情報3：S君の部屋は左から3番目の部屋である。

「情報1」の量＝「情報2」の量＋「情報3」の量



- 情報の量は、可能性の数の対数で定義するのが便利
- 対数の底を2にとると、最小の情報（二択）の量は $\log_2 2 = 1$ となり、情報量の単位として都合がよい。

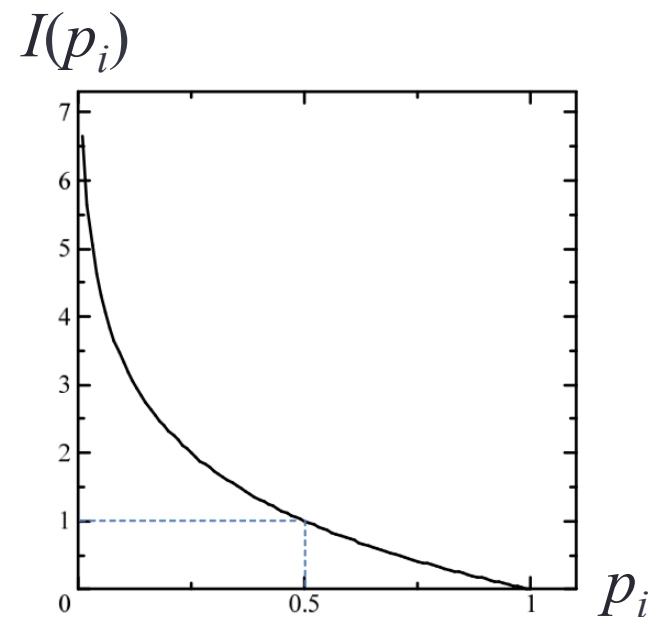
情報量の定義

p_i : ある出来事が発生する確率(事前確率)

$$\begin{aligned} I(p_i) &= \log_2 \frac{\text{事後確率}}{\text{事前確率}} \\ &= \log_2 \frac{1}{p_i} = -\log_2 p_i \end{aligned}$$

自己情報量 or 選択情報量

情報量の単位は ビット[bit]



自己情報量のグラフ

$p_i = 1/2$ のとき、 $I(p_i) = 1$

$p_i = 1$ のとき、 $I(p_i) = 0$

自己情報量の例

[例1] 赤ん坊が生まれたとき、その男女比が1:1とする。男が生まれる事象をboy、女が生まれる事象をgirlとすると、それぞれの自己情報量は次のようになる。

$$I(\text{boy}) = -\log_2 \frac{1}{2} = 1 \text{ bit}$$

$$I(\text{girl}) = -\log_2 \frac{1}{2} = 1 \text{ bit}$$

[例2] ある試験では、合格する可能性が1/8である。この試験に合格した場合の自己情報量は

$$I(\text{合格}) = -\log_2 \frac{1}{8} = 3 \text{ bit}$$

となる。一方、不合格になったときの自己情報量は

$$I(\text{不合格}) = -\log_2 \frac{7}{8} = -\log_2 7 + \log_2 8 = -2.807 + 3 = 0.193 \text{ bit}$$

情報の加法性

情報1: S君の部屋は16室の11号室である. $\longrightarrow \log_2 16 = 4$

情報2: S君の部屋は3階にある. $\longrightarrow \log_2 4 = 2$

情報3: S君の部屋は左から3番目の部屋である. $\searrow \log_2 4 = 2$

「情報1」の量 = 「情報2」の量 + 「情報3」の量

[例] ジョーカーを除いた52枚のトランプを相手に引いて貰い、その内容を教えてもらうことを考える。

- ・引いたカードが、ハートのAであることを知ったときの情報量は

$$I(\text{ハートのA}) = -\log_2 \frac{1}{52} \approx 5.7 \quad \text{bit}$$

- ・引いたカードが、ハートであることのみを知ったときの情報量は

$$I(\text{ハート}) = -\log_2 \frac{1}{4} = 2 \quad \text{bit}$$

- ・引いたカードが、Aであることのみを知ったときの情報量は

$$I(A) = -\log_2 \frac{1}{13} \approx 3.7 \quad \text{bit}$$

平均情報量 (average information)

情報量の平均(期待値)について考えよう。いま、ある事象系を $\{a_1, a_2, \dots, a_N\}$ とする。これら N 個の事象は互いに排反で、その生起確率 p_i の総和は1とする(完全事象系)。このとき、情報量 $I(p_i)$ の平均、すなわち平均情報量 H は次のように示される。

$$\begin{aligned} H &= p_1(-\log_2 p_1) + p_2(-\log_2 p_2) + \dots + p_N(-\log_2 p_N) \\ &= -\sum_{i=1}^N p_i \log_2 p_i \quad \text{ただし、} \quad \sum_{i=1}^N p_i = 1.0 \end{aligned}$$

- ・平均情報量の取りうる値は $0 \leq H \leq \log_2 N$ bit
- ・事象系 $\{a_1, a_2, \dots, a_N\}$ において、一つの事象 a_i の生起確率が $p_i = 1$ で、その他の事象の生起確率がすべて0のとき $H = 0$ 、これは結果を聞く前から結果が既知なので情報としては価値がない。
- ・事象系 $\{a_1, a_2, \dots, a_N\}$ において、すべての事象の生起確率が $p_i = 1/N$ と一様の場合、平均情報量は最大の $H = \log_2 N$ bit となる。これはどれが起きるか全く予想できない状態。

平均情報量の例

[例1]ある家の本日の晩ご飯の生起確率が以下の通りだとする。

$$\begin{aligned} p(\text{カレー}) &= \frac{1}{4}, & p(\text{ハンバーグ}) &= \frac{1}{4}, & p(\text{焼き魚}) &= \frac{1}{8} \\ p(\text{からあげ}) &= \frac{1}{8}, & p(\text{とんかつ}) &= \frac{1}{8}, & p(\text{さしみ}) &= \frac{1}{16} \\ p(\text{ステーキ}) &= \frac{1}{16}, & p(\text{すき焼き}) &= 0 \end{aligned}$$

$$\begin{aligned} H &= -\sum_{i=1}^8 p_i \log_2 p_i \\ &= -\frac{1}{4} \log_2 \frac{1}{4} \times 2 - \frac{1}{8} \log_2 \frac{1}{8} \times 3 - \frac{1}{16} \log_2 \frac{1}{16} \times 2 - 0 \log_2 0 \\ &= \frac{4}{4} + \frac{9}{8} + \frac{8}{16} + 0 = 1 + 1.125 + 0.5 = 2.625 \quad \text{bit} \end{aligned}$$

ただし、 $x \rightarrow 0$ のとき $x \log_2 x \rightarrow 0$

エントロピー (entropy) 無秩序, 混乱

熱力学における分子の無秩序さを表す尺度

熱力学における
エントロピー

$$H = -K \sum_k n_k \ln n_k$$

K はボルツマン定数, n_k は気体分子の k 番目のエネルギー状態にある確率

情報理論における
エントロピー

$$H = - \sum_k p_i \log_2 p_i$$

熱力学におけるエントロピーと平均情報量は定数倍, 対数の底を除いて一致する. そのため, 平均情報量を情報理論におけるエントロピーと呼ぶことにする.

最大エントロピー (maximum entropy)

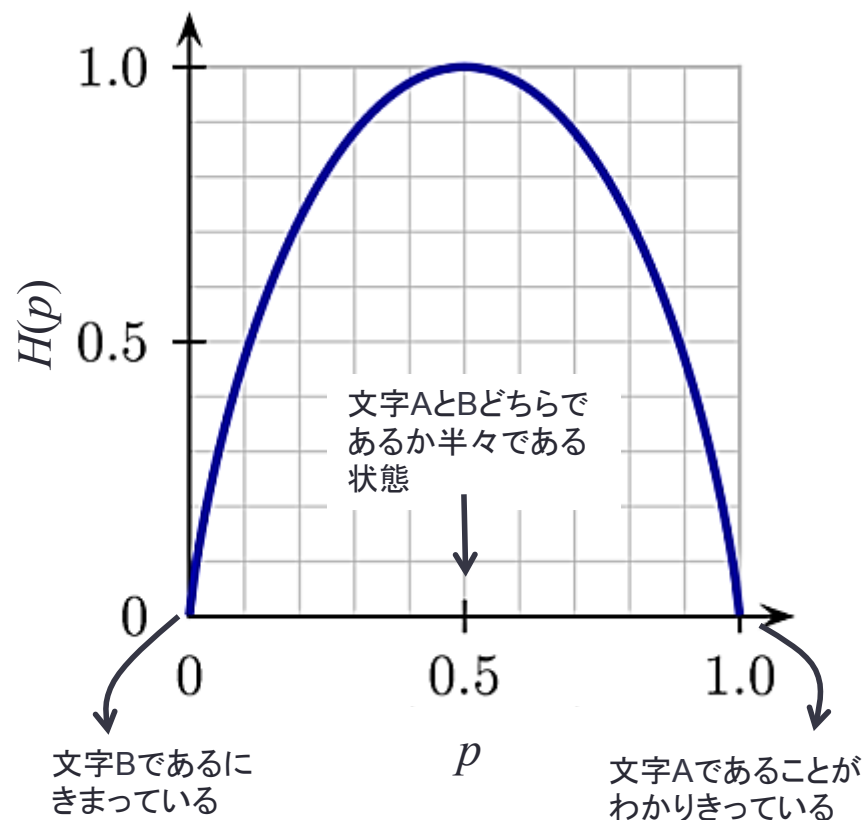
2種類の文字A, Bが, それぞれ確率 p , $1-p$ で生起する情報源のエントロピーは次のように表される.

$$H = -\{p \log p + (1-p) \log(1-p)\}$$

H を確率 p の関数として示すと右のようなグラフ(エントロピー関数)になる.

H が零になるのは, $p = 1$ か $p = 0$ のとき.

H が最大になるのは, $p = 0.5$ のとき.



すべての出来事が等確率で発生すると仮定した場合のエントロピーを最大エントロピー (maximum entropy) という.

最大エントロピーの例

以下の例はいずれも各事象の生起確率が等確率と仮定する.

- サイコロを一回振る時の最大エントロピー

$$H_{\max} = - \sum_{i=1}^6 \frac{1}{6} \log_2 \frac{1}{6} = \log_2 6 = 2.585 \quad \text{bit}$$

- 英数字 (A~Zと空白, 計27文字) の最大エントロピー

$$H_{\max} = - \sum_{i=1}^{27} \frac{1}{27} \log_2 \frac{1}{27} = \log_2 27 = 4.755 \quad \text{bit}$$

- 常用漢字 (1945文字) の最大エントロピー

$$H_{\max} = - \sum_{i=1}^{1945} \frac{1}{1945} \log_2 \frac{1}{1945} = \log_2 1945 = 10.925 \quad \text{bit}$$

冗長度 (redundancy)

最大エントロピー: $H_{\max}$ とエントロピー: H の違いは情報源に含まれる「無駄さ」である。与えられた情報源がどれだけ無駄なものを含むかの度合いを**冗長度 (Redundancy)**という。これは情報の中で実際の情報以外のものの割合とも言える。

冗長度: r は次のように定義される

$$r = 1 - \frac{H}{H_{\max}} = 1 - \frac{\text{エントロピー}}{\text{最大エントロピー}}$$

冗長性があるおかげで. . . .

- ・データの圧縮が可能
- ・正確な情報通信が可能

情報そのものは減らすことなく、情報以外の冗長な部分を切り詰める

あえて冗長な部分を付加・利用して、情報の正確さを判定・担保する

例： 4つの文字 (A, B, C, D) からなるデータ

- ・ 出現率が均等である場合

$$p_A = p_B = p_C = p_D = 1/4$$

冗長度がなく、
最大エントロピー $H_{\max}$ に一致する

$$\text{平均情報量 } H = \log_2 4 = 2.0 \text{ bit}$$

- ・ 出現率に偏りがある場合

$$p_A = 1/2, p_B = 1/4, p_C = p_D = 1/8$$

冗長度があり、得られる
情報量は見かけより少ない

$$\begin{aligned} \text{平均情報量 } H &= 1/2 \times \log_2 2 + 1/4 \times \log_2 4 + 1/4 \times \log_2 8 \\ &= 0.5 + 0.5 + 0.75 \\ &= 1.75 \text{ bit} \end{aligned}$$

$$\text{冗長度 } r = 1 - 1.75 / 2.0 = 0.125$$

情報量を、情報を表現できるビット数であると考えれば、出現率に偏りがある情報のほうが情報量が小さくなる。すなわち、少ないビット数で表現できることを意味する。これは、データ圧縮、特に**可逆圧縮**の基本
原理である。可逆圧縮は元のデータに完全に復元できる。それに対し
て**非可逆圧縮**は元のデータに完全には復元できない。

最小符号長

平均情報量(エントロピー) = 冗長性を廃した場合のデータ量
= **最小符号長**

(情報を表現するのに必要な最低限の符号の長さ)

例: 4つの値からなるデータ(A, B, C, D)

- ・出現率が均等である場合

$$p_A = p_B = p_C = p_D = 1/4$$

平均情報量 = 2.0 ビット



4つの値を表現するのに
必要なビット数に一致する

- ・出現率に偏りがある場合

$$p_A = 1/2, p_B = 1/4, p_C = p_D = 1/8$$

平均情報量 = 1.75 ビット



4つの値を表現するのに
必要なビット数より少ない



「統計的な偏り」があれば、情報量を保持したままデータを圧縮
することができる

データ量と情報量

例: 100種類の文字からなる100文字の文字列

データ量

1文字=8ビット → $8 \times 100 = 800$ ビット (アルファベットなど)

1文字=16ビット → $16 \times 100 = 1600$ ビット (ひらがな, 漢字など)

情報量

データ量は変わっても, 情報の質は変わらない



「100文字の文字列である」

情報量は同じでも, データ化によってデータ量に差が生じる

情報量 < データ量

→ データの冗長性



ファックスでどのように画像を送るのかは、規格で決まっているのね



FAX

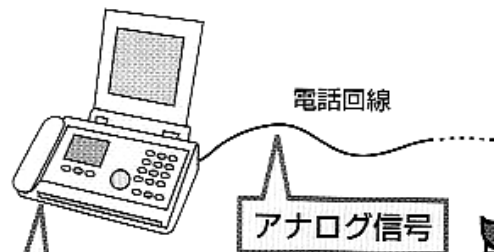
FAXには基本的なデータ圧縮処理が使われている

ファックスの通信規格

G1	A4サイズを6分で送れる
G2	A4サイズを3分で送れる
G3	A4サイズを1分で送れる
G4	デジタル回線で高速

G3の規格がもっともよく使われている

G3ファックス



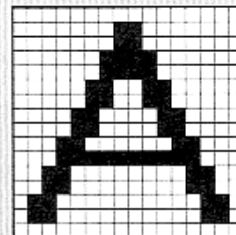
ファックスに内蔵されている「モデム」でデジタル信号をアナログ信号に変換して送っているんだ



FAXで使われる 圧縮技術①

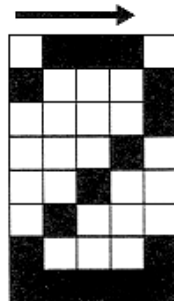


ファックス



ファックスのスキヤナで画像が読み込まれる

ランレングス符号化



左上から順に読み取っていくと……

「白黒黒黒白黒白白黒白白白……」

白ラン 黒ラン 白ラン 黒ラン 白ラン

同じデータが並んだものをランというんだね

ランの長さで書くと……

「白1黒3白1黒1白3黒1白4……」

同じデータがたくさん並んでいると、データ量を少なくできるのね

Run Length encoding
ランレングス符号化
(連長符号化)

FAXで使われる 圧縮技術②

Haffman encoding
ハフマン符号化

エントロピー符号化
の一種

MH符号化

よく出てくるランレングス

「白3」→「10」

「黒2」→「11」

⋮

よく出てくるパターン
を短い記号にあては
めて送れば、全体のデ
ータを短くできるね

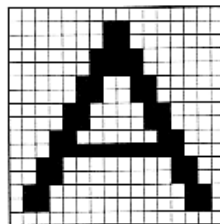
あまり出ないランレングス

「白20」→「00001101」

「黒15」→「00001100」

⋮

MR符号化



ある1行とその下の行と
では似たような画素の並
び方となっている

上の行と異なっている部分のデ
ータだけを送ることにすればさ
らにデータの量を減らせるのね

ランレングス符号化

データ内に同じ値が並んでいる場合,
その並びの数を記録していく方法

元データ

1	1	1	1	0	0	1	1	1	1	2	2	2	1	1	1	3	3	3
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

圧縮データ

4	1	2	0	4	1	3	2	3	1	3	3
---	---	---	---	---	---	---	---	---	---	---	---

ランレングス符号化のバリエーション

元データ

1	1	1	1	0	1	1	1	1	2	2	2	1	1	1	1	1	3
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

方法① 基本的なランレングス符号化

4	1	1	0	4	1	3	2	5	1	1	3
---	---	---	---	---	---	---	---	---	---	---	---

方法② ランレングスを示すコードを挿入する

0xFF	4	1	0	0xFF	4	1	0xFF	3	2	0xFF	5	1	3
------	---	---	---	------	---	---	------	---	---	------	---	---	---

方法③ ラン長部分をランレングスを示すコードとする

0x84	1	0	0x84	1	0x82	2	0x85	1	3
------	---	---	------	---	------	---	------	---	---

エントロピー符号化

データ値の出現頻度に応じてビット長の違う
符号を割り当てる方法

- ・モールス符号と基本的には同じ考え方

普通の符号化

データ値	符号
A	00
B	01
C	10
D	11

平均符号長

$$0.25 \times 2 + 0.25 \times 2 + 0.25 \times 2 + 0.25 \times 2 \\ = 2.0 \quad (\text{ビット})$$

出現頻度に基づく符号化

データ値	出現頻度	符号
A	0.8	0
B	0.1	10
C	0.05	110
D	0.05	111

平均符号長

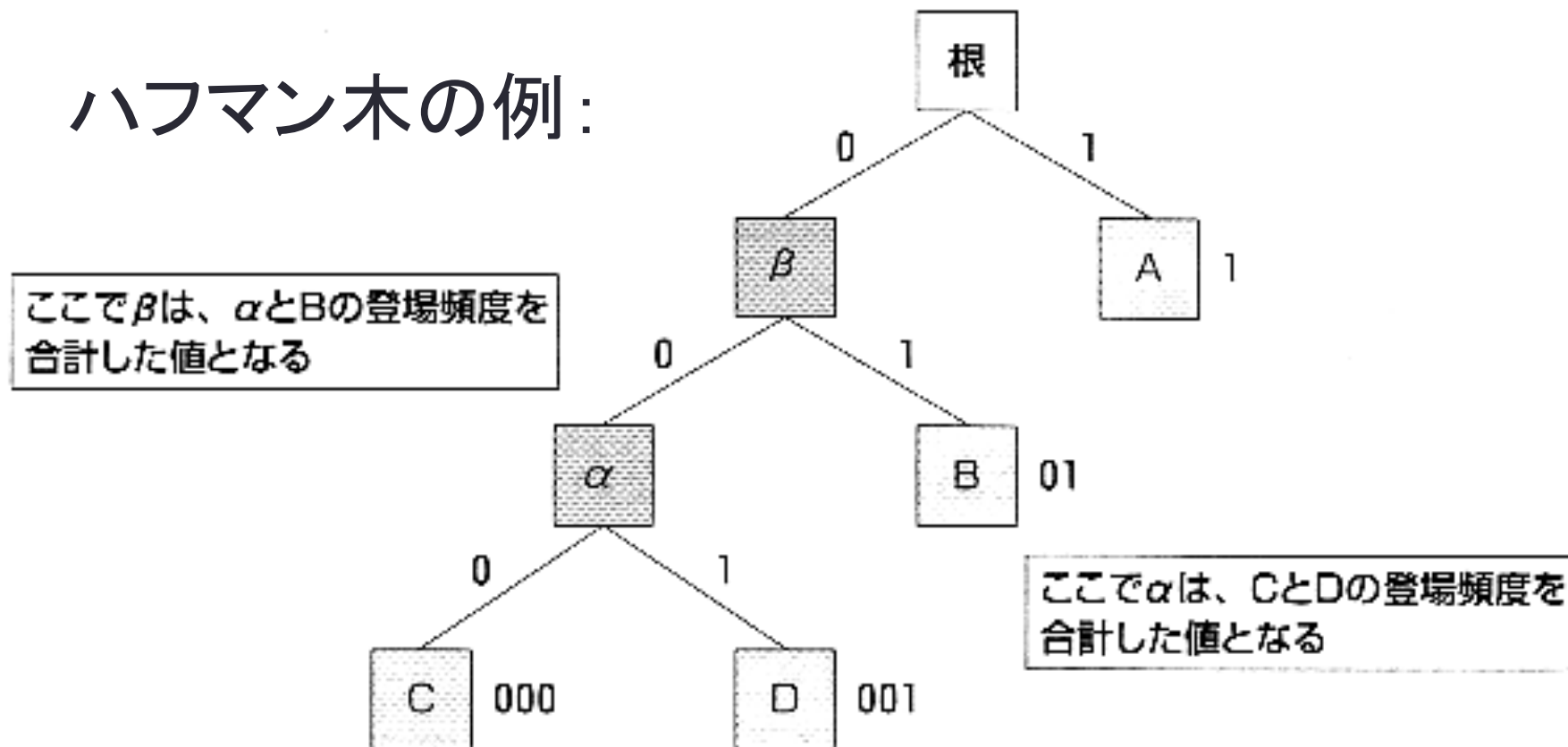
$$0.8 \times 1 + 0.1 \times 2 + 0.05 \times 3 + 0.05 \times 3 \\ = 1.3 \quad (\text{ビット})$$

代表的なもの： シャノン・ファノ符号化, **ハフマン符号化**

ハフマン符号化

各データを重みを持った葉と捉え、出現頻度の低いものをまとめて「ハフマン木」と呼ばれる木構造のデータを構築し、ハフマン木から各データに割り当てるビット列を決定する

ハフマン木の例:

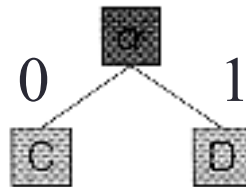


ハフマン木の作成法

- ① データのなかで出現頻度の低いものの2つをまとめ、ツリー状のデータ構造で表現する。そして、2つの出現頻度を合計したものを新たなデータ値の出現頻度とする。
- ② ①で作成したデータ値と次に出現頻度の低いデータとで①の処理を行う。これをハフマン木が1つにまとまるまで繰り返す。

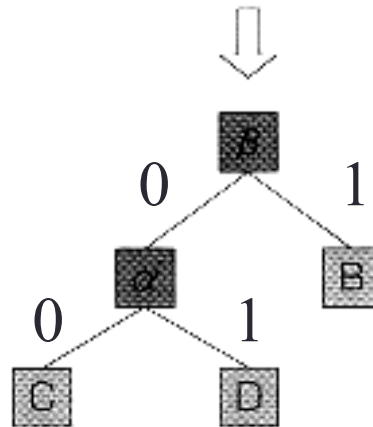
ハフマン木の作成例

データ値	登場頻度
A	0.8
B	0.1
C	0.05
D	0.05



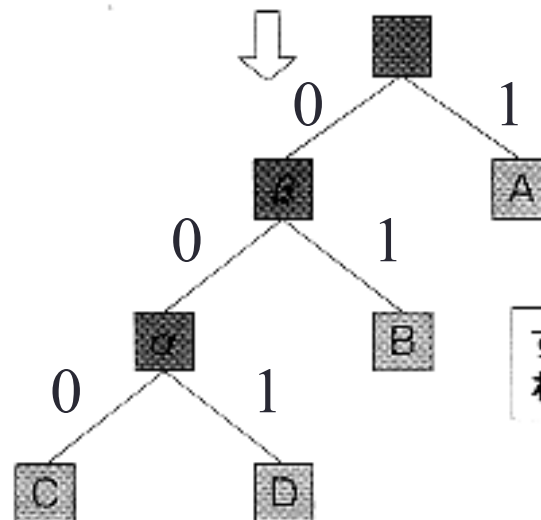
登場頻度の低い2つをまとめる

データ値	登場頻度
A	0.8
B	0.1
α	$0.05+0.05$



登場頻度の低い2つをまとめたものと、次に登場頻度の低いものをまとめる

データ値	登場頻度
A	0.8
β	$0.1+0.1$



すべてのデータが木に結び付けられるまで処理を繰り返す

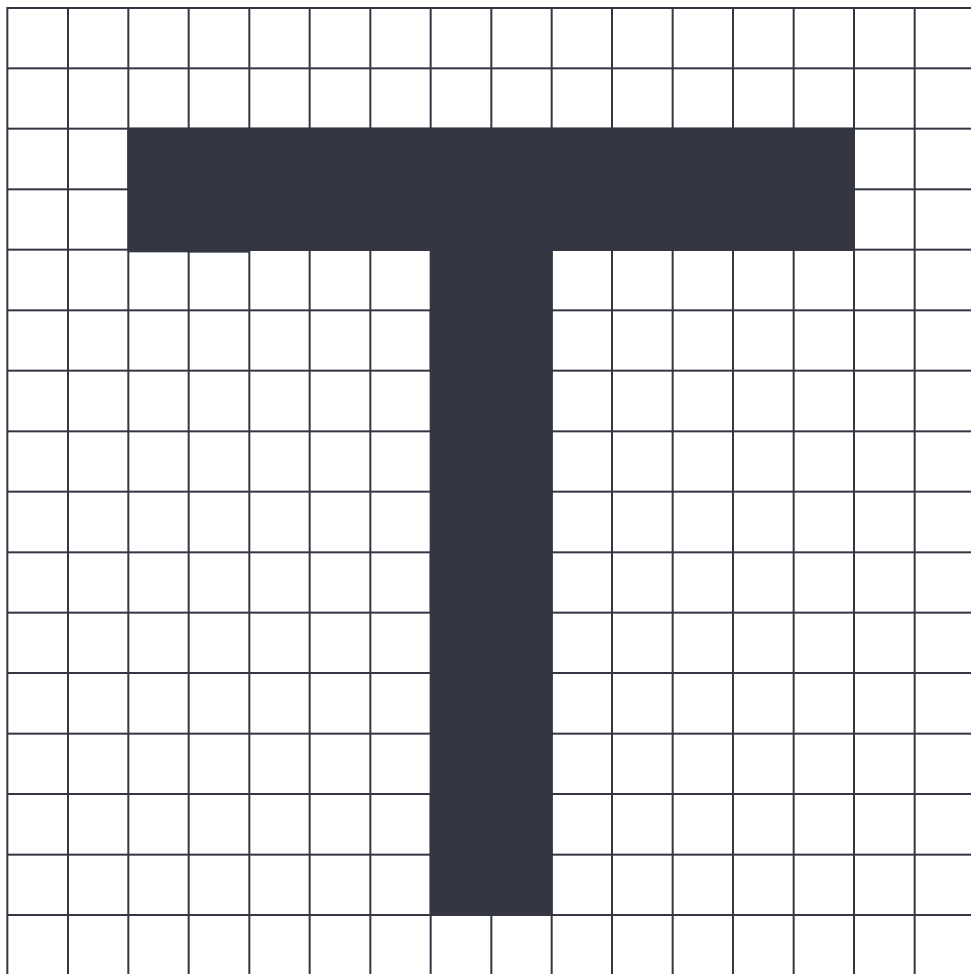
ハフマン符号化されたデータの復号

- ・復号にもハフマン木が必要
- ・元のデータの出現頻度情報を付加しておく

データ値		出現頻度		出現頻度情報の 付加によるデータ量 の増加分 
A	1 バイト	0.8	4 バイト	
B	1 バイト	0.1	4 バイト	
C	1 バイト	0.05	4 バイト	
D	1 バイト	0.05	4 バイト	
		+		= 20 バイト
4 バイト			16 バイト	

- ・ハフマン木のための付加情報によってデータ量が多くなって
しまうこともあり得る

圧縮してみよう！



1画素を 1 bitで表現した
2値画像: (16×16画素)

- ・データ量はいくつかな？
- ・平均情報量はいくつかな？
- ・ランレングス符号化後のデータ量はいくつかな？
- ・ランレングス符号化＋ハフマン符号化後のデータ量はいくつかな？