

7 データの取扱い

7.1 コンピュータが扱うデータ

7.2 数値計算における誤差

7.3 符号

7.4 データの圧縮

7.5 暗号

1

7.1 コンピュータが扱うデータ

整数： k ビット固定、2の補数表現

- $k = 4$ のとき、

0111	$2^3 - 1$
↓	↓
0000	0
1111	-1
↓	↓
1000	-2^3

2

浮動小数： 仮数 $\times 10^{\text{指数}}$ のような方式

- 32ビット浮動小数表現:

31	30	...	23	22	...	0
符		指数			仮数	

$\pm 0.\overline{\text{仮数}} \times 2^{\overline{\text{指数}} - 127}$ を表す。ここで、

- 符号：0が正、1が負
 - 仮数：左のビットから順に重み $1/2, 1/4, \dots$
 - 指数：符号なし整数(8ビット)で、 $127 = 2^8 - 1 - 1$
- 丸め誤差：0.1を10000回加算しても1000にならない

3

7.2 数値計算における誤差

文字, 文字列： ASCIIコード、UTF-8コード など

アナログデータ： 音声、画像、動画：離散化が必要

- 標本化：時間軸の離散化
- 量子化：大きさ軸の離散化

圧縮の種類 可逆圧縮(LZ, 連長圧縮)、非可逆圧縮(jpeg)

4

桁落ち： 引き算などで、有効数字が減る場合あり

123.4567
- 123.4566

0.0001

有効数字が6桁減る

情報落ち： 加減算で小さい数字の桁が損失

123.4567
+ 0.01234567

123.4444

繰り返し計算すると大きな誤差になるが、小さい順で加えれば誤差はほとんどない

5

例：分散 (ここで、 m は x_1, \dots, x_n の平均)

- 桁落ちが誤差を大きくする例

$$\frac{1}{n} \left(\sum_{i=1}^n x_i^2 \right) - m^2$$

- 桁落ちが誤差に影響しない例

$$\frac{1}{n} \sum_{i=1}^n (x_i - m)^2$$

近似式： 精度と計算量はトレードオフ

コラム： $\sum_{n=1}^{\infty} \frac{1}{n} = \infty$ の計算

$\sum_{n=1}^{2^k} \frac{1}{n} \geq 1 + \frac{k}{2}$ であることが知られているが、左辺を単精度(有効数字8桁程度)で計算すると k を大きくしても情報落ちのため15.5を超えられない

6

7.3 符号 (7.3.1 符号化)

符号化 : アルファベット Γ の文字列をアルファベット Σ の文字列で表すこと (1対1の準同型写像 $C : \Gamma^* \rightarrow \Sigma^*$)

- C が準同型写像 $\stackrel{\text{def}}{\iff}$

$$\forall a_1, \dots, a_n \in \Gamma \quad C(a_1 \dots a_n) = C(a_1) \dots C(a_n)$$
- 準同型写像を定めるには、 $\forall a \in \Gamma \quad C(a)$ の値を決めればよい

問7.8 : $C(a) = 10, C(b) = 1$ で定まる準同型写像 C は (たまたま) 1対1であり、逆変換可能

- $C(ab) = C(a)C(b) = 101$
- $C(bba) = C(b)C(b)C(a) = 1110$
- $C^{-1}(101101) = abab$

7

符号 : $C(\Gamma) = \{C(a) \mid a \in \Gamma\}$

- 前ページの間では、 $\{10, 1\}$ が符号

接頭符号 : 接頭条件 ($X\Sigma^+ \cap X = \emptyset$) を満たす符号 X
 ここで、 $X\Sigma^+ = \{ww' \mid w \in X, w' \in \Sigma^+\}$

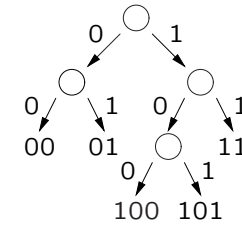
- 符号 $\{10, 1\}$ は接頭符号でない

問7.9 : 接頭条件を満たす文字列の集合は符号であるか?

解 : **YES.** X が接頭条件を満たすが符号でないとは仮定
 - $\exists x_1 \dots x_n, y_1 \dots y_m \in X \quad x_1 \dots x_n = y_1 \dots y_m$ かつある i で $x_i \neq y_i$ (i を最小のものとする)
 - x_i と y_i は接頭の関係であり、接頭条件を満たすことに矛盾する

8

- 木を利用した接頭符号の生成



符号 $\{00, 01, 100, 101, 11\}$

- 木を利用すると復号も容易

9

7.3 符号 (7.3.2 誤検出、訂正符号)

パリティ : 1の個数が偶数になるよう、**パリティビット**を付け加える

0110111**1**, 0110011**0**

ここで、青字がパリティビット

- 1が奇数個のとき誤りを検知
- 偶数個の誤りは検知できない

10

ハミング符号 : どの符号間も3以上のハミング距離をもつ符号

- ハミング距離: 異なるビットの数 (以下の例では3)

0001110
 $\uparrow \quad \uparrow \uparrow$
 0101101

- 1ビットの誤りなら訂正可能
 ハミング距離1の符号がひとつ定まるはず

11

6.4 データ圧縮 (6.4.1 ハフマン符号)

ハフマン符号 : 圧縮したい文字列における (各文字の) 生起確率の高い記号を短い符号に割り当てる方法

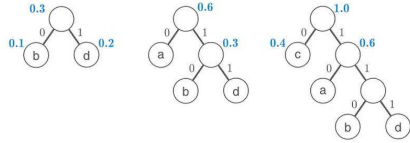
例 : モールス符号

長さ	1	3	3	5	5	5	...
符号	.	..	-	...	-.	..-	...
記号	e	i	t	s	c	n	

12

ハフマン符号の構成：生起確率の小さい順に木を作成

記号 x	出現確率 p_x	ハフマン符号
a	0.3	10
b	0.2	110
c	0.4	0
d	0.1	111



符号長の期待値: $0.3 \times 2 + 0.2 \times 3 + 0.4 \times 1 + 0.1 \times 3 = 1.9$

問7.18: *abracadabra*に対するハフマン符号を求めよ

6.4 データ圧縮 (6.4.2 情報量)

情報量：データがもつ本質的な情報の量：圧縮可能な最小のビット数

定理7.2：どんな n ビットデータでも $n-1$ ビット以下に可逆に圧縮するアルゴリズムは存在しない

証明 n ビットデータは 2^n 種類あるが、 $n-1$ ビットデータの種類は真に小さいため

シャノンの情報量 基本的な考え方

- 事象の確率が高ければ、その事象の情報量は少ない
- 確率 p の事象 P の情報量を $f(p)$ と書く
- 独立事象 P_1, P_2 について、 $f(p_1 \times p_2) = f(p_1) + f(p_2)$

情報量の定義: $f(p) = -\log_2 p$

定理7.3：独立事象の確率 p_1, \dots, p_m について、その情報を表す符号を平均して

$$\sum_{i=1}^m (-p_i) \log_2 p_i$$

ビットより縮めることは出来ない

問7.19: 13ページの生起確率における情報量を求め、

ハフマン符号の平均長1.9と比較せよ

$$-p_a \log_2 p_a + \dots + -p_d \log_2 p_d = 1.847$$

6.5 暗号 (6.5.1 暗号の基本)

送信者: 平文 $\xrightarrow{\text{暗号化}}$ 暗号

受信者: 暗号 $\xrightarrow{\text{復号}}$ 平文

- 暗号関数=アルゴリズム + 鍵
- 復号関数=アルゴリズム + 鍵

共通鍵暗号系：暗号関数と復号関数で共通の秘密鍵を用いる

公開鍵暗号系：一方の鍵を秘密とし、もう一方の(異なる)鍵を公開とする

Diffie-Hellman鍵交換：秘密鍵を共有するためのプロトコルのひとつ

前提：(秘密でない)数 x と素数 p を**A**と**B**で共有

a : **A**の秘密の数(ナンス)、 b : **B**の秘密の数(ナンス)

A	通信	B
$A := x^a \pmod p$	\xrightarrow{A}	
	\xleftarrow{B}	$B := x^b \pmod p$
$K_A := B^a \pmod p$		$K_B := A^b \pmod p$

ここで、鍵は、 $K_A = K_B = x^{ab} \pmod p$ 。通信の盗聴者は、 a, b を知らないため鍵を求めることは困難

6.5 暗号 (6.5.2 公開鍵暗号)

暗号 受信者の公開鍵で暗号化... 受信者の秘密鍵でのみ復号可能

署名 送信者の秘密鍵で暗号化... 送信者の公開鍵でのみ復号可能

RSA暗号系 : Rivest-Shamir-Adleman 提案の公開鍵暗号系

基礎事実 : p, q を素数とするとき

$$\forall x, k \in \mathbb{N}, x \equiv_{pq} x^{1+k(p-1)(q-1)}$$

ここで、 \equiv_{pq} は mod pq のもとの等価性を表す
基礎事実の証明は教科書 6.5.3 を見よ

19

鍵生成 : 公開鍵 (e, n) 、秘密鍵 (d, n) の生成

- 大きな素数 p, q を選ぶ ($n = pq$ とする)

- $de \equiv_{(p-1)(q-1)} 1$ を満たす $d, e \in \mathbb{N}_+$ を定める

- 基礎事実より $x \equiv_n x^{de} (= (x^d)^e = (x^e)^d)$

- $f(x) = x^e \bmod n, g(x) = x^d \bmod n$ とさだめれば、

$$g(f(x)) = f(g(x)) = x \quad (0 \leq x < n)$$

p, q が知られると e から d が計算できるので、鍵を生成したら p, q を忘れることが必要。基礎事実の証明は教科書 6.5.3 を見よ

20

素数の候補を探す : フェルマーテストを使って素数候補を決める

$$p \text{ が素数ならば } 2^{p-1} \equiv_p 1$$

この他、いろいろな (決定的、あるいは、確率的) 素数判定法がある

鍵の対を探す : 拡張ユークリッド法

$$mx + ny = \gcd(m, n) \text{ を満たす } x, y \text{ を計算する手法}$$

21