情報の表現—記号・符号化 (その 2)

山本昌志*

2007 年 10 月 24 日

概 要

ここでの主な内容は，情報の符号化と誤り検出・訂正についてである．

1 本日の学習内容

情報の表現の方法とデジタル化の基礎を学ぶ

- ハミング距離の概念が分かる．
- デジタル符号の圧縮の方法が分かる．
- デジタル符号の検出と誤りの訂正が分かる．

教科書 [1] の pp.27–35 が本日の範囲である．

2 デジタル符号化

一般に，情報を記号によって表現することを符号化 (コード化) と言う．表現されたものを符号 (コード) と呼ぶ．情報を表す記号は，何を使っても良いが，整数を使うのが最も簡単であるし，コンピューターとの相性も良い．特に，コンピューターでは 2 進数がもっとも扱いやすい．

2.1 デジタル符号化の事例

2.1.1 2 進数符号

10 進数 2 進数で表す話である．しかし，講義ではこのこの辺の話は，省略する．10 進数と 2 進数，16 進数の間の変換については，諸君はしつこく学習したであろう．私も，3 年生の電子計算機の第 2 回の講義「位取り基数法 (2 進数，10 進数，16 進数)[1]」で，説明した．よもや，これらの変換が分からない者など，いないはず．

ただし，ハミング距離 (Hamming distance)[2]については，述べておくべきであろう．

*独立行政法人　秋田工業高等専門学校　電気情報工学科

[1] http://www.akita-nct.jp/yamamoto/lecture/2005/3E/2nd/html/index.html

[2] ハミングとは人の名前．初期のコンピューター (偶数パリティを使っていた) は誤りが多く，しばしば計算途中で止まった．これに，腹を立てたハミングは，誤り訂正を考えたとか．

二つのデータで，同じ位置にあるビットで異なるビットの個数をハミング距離と呼ぶ．たとえば，(01000110) と (01000001) のハミング距離は 3 である．

- 8 ビット整数 (0–255) のうち，隣り合う整数の組でハミング距離が最大の組を示せ．

2.1.2 グレイ符号

グレイ符号は，値が一つ異なる整数のハミング距離がいつでも 1 の符号である．教科書 [1] の pp.28 の表 2.2 を見れば，このことが分かるだろう．詳細については，応用上，重要な話もあるが，今後の講義にあまり関係なので，説明しない．

グレイ符号については，私の講義ノート「グレイコード　誤りの検出[3]」に書いてあるので見ると良いだろう．また，参考文献 [2] も詳しい．

2.2 デジタル符号の圧縮

p の確率 (probability) で生じる事実が発生した時，それを観測することにより得られる情報量は $-\log_2 P$[bit] である．たとえば，X 先生はテストの出題には癖があり，練習問題 A と B が出題される確率は表 1 のとおりである．両方の問題が出題された場合，それにより得られる情報量は 1[bit] である．

- 問題 B のみ出題されたとき，得られる情報量は何 [bit] か?

表を見て分かるとおり，これらの事実は 4 通りある．そのため，2 桁の 2 進数で符号化可能である．しかし，得られる情報量は 2 ビットとは限らない．

表 1: テストに練習問題 A と B が出題されるときの符号化と確率

問題 A と B の出題	どちらも出題されない	問題 B のみ	問題 A のみ	両方出題
符号化	00	01	10	11
確率	1/8	1/8	1/4	1/2

すなわちデジタル符号 (データ) の圧縮は，情報量 [bit] と 2 進数の桁数が異なることを利用する．すなわち，情報を 2 進数で符号化すると，その桁数と情報量は一般には異なる．先に示したように，符号化された 2 進数の 0 と 1 の出現確率が異なるためである．もし，0 と 1 の出現確率が同じ 1/2 ならば，桁数と情報量は同じになり，デジタル符号の圧縮はできない．

実際の圧縮の例として，教科書 [1] の pp.30–31 ではハフマン符号化とラングレス圧縮の例が記述されている．

- **ハフマン符号化**は，出現確率の大きな記号に短いビット列を，小さいに記号に長いビット列を当てる方法である．すべての記号に同じ長さのビット列を与えるよりも，短いビット列で符号化できる確率が高い．

[3] http://akita-nct.jp/yamamoto/lecture/2003/2E/pdf_files_2E/numerical_code.pdf

- **ラングレス圧縮**は，ビットの値とその繰り返しで符号化する．

人間の感覚では気がつかない部分の情報を減らすことにより，デジタル符号を圧縮することも可能である．たとえば，画像データフォーマットで使われる JPEG などがその例である．教科書 [1] の pp.27 の図 2.10 のように，周波数の高い成分の情報を削除しても人間は気がつかない．人間が気がつかない周波数の高い成分を圧縮しても，人間への伝達は問題がない．

符号の圧縮には，可逆圧縮と非可逆圧縮がある．圧縮されたデジタル符号が元の状態に戻せるものを可逆圧縮，戻せないものを非可逆圧縮と呼ぶ．ハフマン符号化とラングレス圧縮は可逆圧縮で，JPEG 圧縮は非可逆圧縮である．

コーヒーブレイク

TEX 関係で有名な奥村晴彦先生が，**圧縮のジレンマ**として，おもしろいことを書いている [3] ので引用しておく．

> **定理** 最悪の場合でも圧縮データが元データより大きくならない圧縮ソフトは，どんなデータも圧縮できない．
>
> **証明** 元データのサイズが 0 ビットなら，圧縮して大きくなることはないのだから，圧縮データのサイズも 0 ビットでなくてはならい．元データのサイズが 1 ビットなら，もう 0 ビットは使用済みなので，圧縮データは 1 ビットでなくてはならない．1 ビットの元データ (2 通り) が，1:1 に対応する．以下同様に続けていけば，元データのサイズが n ビットなら，圧縮しても n ビットでなければ成らないことがわかる．

要するに，元データと圧縮データは 1:1 に対応するため，全く圧縮できないことになる．また，もしこの定理が成り立たなければ，この圧縮ソフトを繰り返し使うことにより，どんなデータも 0 ビットに圧縮できてしまう．これは，矛盾である．

3 符号の誤り検出・訂正

この辺の話は，参考文献 [4] にかなりわかりやすく書かれている．余裕があると呼んでみると良い．

3.1 誤り検出と訂正がなぜ必要か?

コンピューターでは大量のデータ，すなわち気の遠くなるような 0 と 1 のビット列が取り扱われており，一つの間違いも許されない．諸君が使っているパソコンのメインメモリーが 1G[byte] とすると，どれだけのビットがあるだろうか? $8 \times 2^{30} = 2^{33} \simeq 10^{10}$ 個のビットがある．100 億個である．ハードディスクになると，その数百倍のビットを扱うことになる．

データのエラーは 2 通りの方法で生じる．一つは，メモリーやハードディスクの製造時における欠陥である．もう一つは，稼働時のおける書き込みあるいは読み込みの間違い，あるいはデータ保存時にビットが変化してしまうことによるエラーである．

前者のエラー，製造時のエラーは欠陥部分を使わないようにしてしまうことにより回避できる．いつも，同じ場所でエラーが生じるので，検査によりそのビットを使わないようにする．

ここで問題とするのは後者のエラーである．すなわち，装置に欠陥が無いものの，書き込んだものと異なる値が読み出される場合である．これは熱や宇宙線の作用により，記憶装置のビットが変化する場合に起きる．このようなビットの変化は，ランダムに生じ，時や場所を特定することができない．

ランダムにビットが変化するようなエラーは，情報の保存のみならず，情報の伝達の時も生じる．情報を伝達のケーブルの電圧を下げると熱振動によるノイズの影響によりビットが変化するかもしれない．また，通信速度を上げると，パルス幅が短くなり，トランジスターが誤動作するかもしれない．

熱や宇宙線によるこれらのビットの変化が生じる確率は，ゼロでは無いが非常に少ない．問題の無いレベルになるように，記憶装置や通信装置を慎重に設計を行わなければならない．ただ，ゼロにすることは不可能なので，ビットが変化しても情報が失われないように，データ蓄積と転送のとき工夫する．

データ蓄積と転送は全く異なる技術に見えるが，ランダムにビットが変化するエラーを防ぐためには同じような技術が使える．

3.2 誤りの検出

パリティビットを使うと，蓄えた情報あるいは転送して送られてきた情報の誤りの有無が分かる．たとえば，0110010 と符号化された情報があるとする．これの最後に 1 ビット付け加えて，列の全体の 1 の数を偶数[4]にする．すなわち，01100101 とする．このようにパリティビットを加えて，符号を記憶，あるいは転送する．

データを読み出すときに，1 の数を数えれば誤りの有無が分かる．もし，何らかの原因でどれか一つのビットが反転した場合，全体の 1 の数は奇数となるので，おかしいと気付く．

この方法だと，二つのビットの誤り (二重誤り) には気付かない．どのようにすれば，二重あるいは 3 重の誤りが分かるだろうか?

偶数パリティを加えることにより，正しい符号のハミング距離は 2 以上になる．パリティビットを含んだ符号全体の 1 の数は，いつも偶数であるからである．従って，一つの誤りが生じると，正しい場合に比べてハミング距離が 1 変化し，不当な符号になる．もし，2 重の誤りが生じると，正しい符号と区別がつかなくなる．従って，誤りの検出ができない．

このことを一般化すると，t 個の誤りを検出するためには，正しい符号のハミング距離を $t+1$ 以上にする必要がある (図 1).

[4]偶数にする場合を偶数パリティ，奇数にする場合を奇数パリティと呼ぶ.

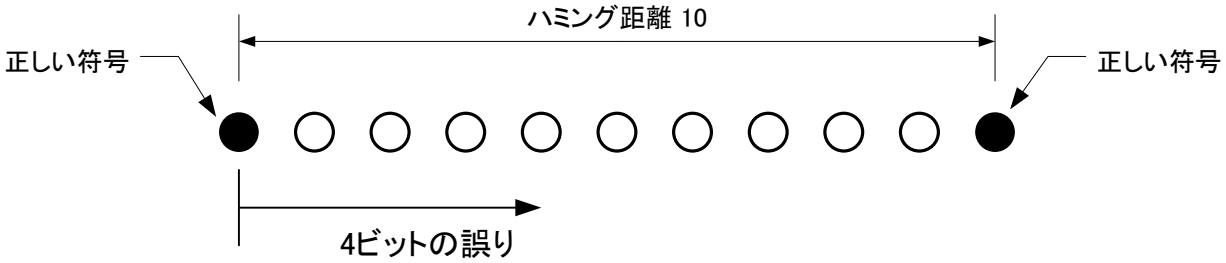

図 1: 誤りの検出限界とハミング距離の関係.

3.3 誤りの訂正

記憶装置の場合，誤りが分かっただけでは困る．データの損失になるからである．それに対して，通信の場合は，再送信することにより，正しい情報を得ることができる．しかし，いずれの場合でも，誤りを検出すると同時に得られた誤りのある符号から，正しい符号に訂正できれば，ハッピーである．これは，技術的に可能である．

t 個のビットの誤りがある場合，ハミング距離が $2t+1$ 以上ならば，誤りを検出して訂正までできる．このことを，図 2 を用いて説明する．この図，次のように理解する．

- 誤り検出・訂正用のビットを付加して，正しい符号のハミング距離を 11 としている．
- もし，一個のビットに誤りがあると，元の正しい符号からハミング距離が 1 だけ離れた場所に符号が移動する．この場合，最近接の正しい符号に戻せば，誤りの訂正を行ったことになる．
- 二個，三個，四個，五個までならば，最近接の正しい符号に戻すことができる．このように最近接の正しい符号に戻すことにより誤りの訂正ができる．
- もし，誤りが 6 個以上になると，最近接の正しい符号に戻しても，元の符号に戻らない．誤った訂正を行ったことになり，情報が失われる．

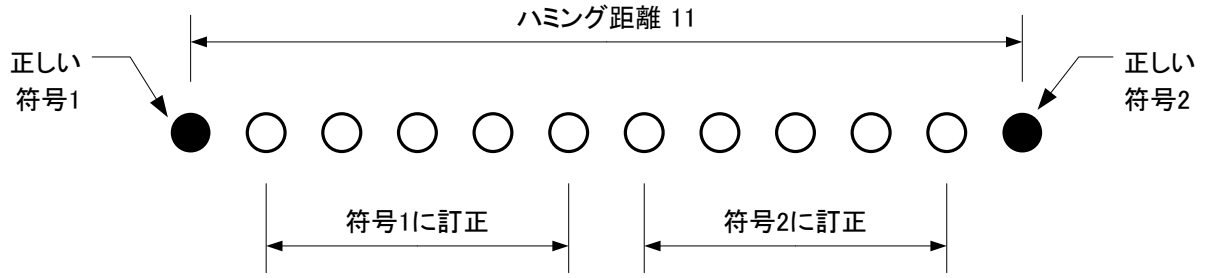

図 2: 誤りの訂正限界とハミング距離の関係.

実際に，誤りの訂正ができることを教科書のハミング符号で示す (図 3)．この教科書のハミング符号によると，以下のようなことが分かる．

- もし，得られた符号を検査して，z_1 と z_2 が間違いで，z_3 が間違いの場合，教科書の表 2.3 より，x_1 が間違いだと分かる．
- もし，得られた符号を検査して，z_1 が間違いで，z_1 と z_3 が間違いの場合，教科書の表 2.3 より，y_1 が間違いだと分かる．

ハミング符号による誤り訂正のおもしろいところは，付加したビットそのものの誤りも分かるところである．元の符号と訂正用に付加した符号は同等に取り扱われる．

実際に使われているハミング符号は，もっと高度なもので，高速に処理ができるようになっている．教科書のハミング符号は，分かりやすく書くために書かれており，処理の効率は悪い．

	x_1	x_2	x_3	x_4	y_1	y_2	y_3
0 ⟶	0	0	0	0	0	0	0
1 ⟶	0	0	0	1	0	1	1
2 ⟶	0	0	1	0	1	0	1
3 ⟶	0	0	1	1	1	1	0
4 ⟶	0	1	0	0	1	1	0
5 ⟶	0	1	0	1	1	0	1
6 ⟶	0	1	1	0	0	1	1
7 ⟶	0	1	1	1	0	0	0
8 ⟶	1	0	0	0	1	1	1
9 ⟶	1	0	0	1	1	0	0
A ⟶	1	0	1	0	0	1	0
B ⟶	1	0	1	1	0	0	1
C ⟶	1	1	0	0	0	0	1
D ⟶	1	1	0	1	0	1	0
E ⟶	1	1	1	0	1	0	0
F ⟶	1	1	1	1	1	1	1

図 3: 教科書のハミング符号.

4 課題

4.1 課題内容

以下の課題を実施し，レポートとして提出すること．

[問 1] (復予) 教科書 [1]pp.27–51 を 2 回読み，重要な部分には赤線でアンダーラインを入れよ．レポートには「2 回読んだ」と書け．

[問 2] (復) 整数 $(0)_{10}$〜整数 $(15)_{10}$ を 4 ビットの 2 進数で符号化するとき，隣り合う整数のハミング距離をすべて示せ．

[問 3] (復) プリントの表 1 のすべての現象についての情報量を示せ.

[問 4] (復) 以下の符号の最後の桁に偶数パリティを追加する. 追加された符号を書け.

00100111001101　　10101010101010

4.2 レポート提出要領

期限　11 月 9 日 (金) AM 8:45
用紙　A4 のレポート用紙. 左上をホッチキスで綴じて, 提出のこと.
提出場所　山本研究室の入口のポスト
表紙　表紙を 1 枚つけて, 以下の項目を分かりやすく記述すること.
　授業科目名「情報理論」
　課題名「課題　情報の表現—記号・符号化 (その 2)」
　提出日
　5E　学籍番号　氏名
内容　2 ページ以降に問いに対する答えを分かりやすく記述すること.

参考文献

[1] 河合慧（編）. 情報. 東京大学出版会, 2006.

[2] 立木秀樹. グレイコードと実数. http://www.i.h.kyoto-u.ac.jp/ tsuiki/bit/gray.html.

[3] 奥村晴彦. データ圧縮. 数学セミナー, pp. 28–32. 日本評論社, 10 月号, 2006. この 10 月号の特集は「符号の数理」である. 符号に関に関する様々な側面を分かりやすく記述している.

[4] A. ヘイ, R. アレン（編）. ファインマン 計算機科学. 岩波書店, 2002. 物理学者が情報科学に関して, 講義を行った時の講義ノート. 非常にユニークな視点で書かれており, 一読を勧める.