

最小ハミング距離法のファイル構成への応用

筑波大学社会工学系

高橋磐郎

§0 序

リレーショナルデータベース等におけるファイルは、表 2 (の $x_1 \sim x_7$ 列) のようなものと考えられる。これは $m = 7$ 項目, 各項目の カテゴリ数 $q = 2$, レコード数 $n = 16$ ($\leq q^k = 2^4$) のものである。 q が素数 p のとき, 各項目のカテゴリを $GF(q)$ の元と考えると, ファイル R は $GF(q)^m$ の大至小の部分集合である。

Bose 等は, $GF(q)^m$ の中に強 s の t 次元線形部分空間 S (強 s の直交表) を考え, S の各点をブロック (バッチの集り) とみなすファイルによって, t 項目直内に答え得るファイリングシステムを提案し, これが普通の転置方式より冗長度が少いことを証明している。しかしこの方法は R の点の分布に無関係に S を構成しているため, R の点の分布が一様性をとつ場合にしか, その有用性が謳われぬ。実際に

R の基底は $GF(q)^m$ の中にきわめて疎にしかもたれり備って分布している。[1][2]

ここで与えられた R にある意味で最もよく適合する線形部分空間 S にとがくファイリングシステムを提案する。与えられた R に対して S を求めるのに、実数の世界での最小二乗法に匹敵する、最小ハミング距離法なるものを利用する。

このアルゴリズムは本質的に、コード理論における リードランナーコード のデコード方式と相当のものである。[0]

§1 最小ハミング距離法によるあてはめ

$GF(q)$ ($= GF(2)$) 上の k ($= 3$) 変数

関数 $y = f(x_1, x_2, x_3)$ の値を表 1 のよう

に与えられたとき、これに 1 次式

$$(1) y_p = a_0 + a_1 x_{p1} + a_2 x_{p2} + a_3 x_{p3} + e_p$$

$$p = 1, \dots, q^k = n, \quad a_i \in GF(q)$$

をあてはめる問題を考える。その原則は

誤差ベクトル $e = (e_p : p = 1, \dots, n)$ のハ

ミングウェイトが最小になるように、

つまり $y = (y_p : p = 1, \dots, n)$ と $(a_0 + a_1 x_{p1}$

$+ a_2 x_{p2} + a_3 x_{p3} : p = 1, \dots, n)$ とのハミ

表 1

p	x_1	x_2	x_3	$y = f(x_1, x_2, x_3)$
1	0	0	0	0
2	0	0	1	1
3	0	1	0	0
4	0	1	1	1
5	1	0	0	1
6	1	0	1	1
7	1	1	0	1
8	1	1	1	0

とグ距離が最小になるように, a_0, a_1, a_2, a_3 を定めるものとする.

このような a_i を最小ハミング距離推定値と呼ぶ. とすると, GF(2) の場合は, これほつきの形式で決められることかわかる.

$$(2) \quad \begin{cases} \hat{a}_1 = \text{maj} \left\{ \sum_{x_1 \in \text{GF}(2)} f(x_1, 0, 0), \sum_{x_1} f(x_1, 0, 1), \sum_{x_1} f(x_1, 1, 0), \sum_{x_1} f(x_1, 1, 1) \right\} \\ \hat{a}_2 = \text{maj} \left\{ \sum_{x_2} f(0, x_2, 0), \sum_{x_2} f(0, x_2, 1), \sum_{x_2} f(1, x_2, 0), \sum_{x_2} f(1, x_2, 1) \right\} \\ \hat{a}_3 = \text{maj} \left\{ \sum_{x_3} f(0, 0, x_3), \sum_{x_3} f(0, 1, x_3), \sum_{x_3} f(1, 0, x_3), \sum_{x_3} f(1, 1, x_3) \right\} \end{cases}$$

$$(3) \quad \hat{a}_0 = \text{maj} \{ y'_p \mid p=1, \dots, n \},$$

$$y'_p = y_p - \hat{a}_1 x_{p1} - \hat{a}_2 x_{p2} - \hat{a}_3 x_{p3}, \quad p=1, \dots, n$$

ここで maj は多数決関数, つまり $\text{maj} \{ 0, 0, 1 \} = 0$, $\text{maj} \{ 1, 0, 1 \} = 1$ など. tie が起こる場合おまじ $d(y, \hat{y}) > 2^{m-2}$ の場合は [4] 参照. ただし $\hat{y} = \hat{a}_0 + \hat{a}_1 x_1 + \hat{a}_2 x_2 + \hat{a}_3 x_3$ ($x_i = (x_{pi} : p=1, \dots, n)$) で $d(y, \hat{y})$ は y と \hat{y} とのハミング距離.

表 1 に対して (2), (3) を適用すると,

$$(4) \quad \hat{y} = x_1 + x_3, \quad d(y, \hat{y}) = 1$$

である. この公式は本質的には, 1 次のリード-マラーコード

にあけるデコード方式と同等である。一般に k 次式のあてはめも同様な方法で可能であり、これは k 次のリードマウーコードの場合に対応する [0]。

§2 リレーショナルファイルにあてはめられた

線形部分空間

表2 ($x_1 \sim x_7$ 列) のようなファイル R が与えられているとしよう。これにキー (t_1, t_2, t_3, t_4) (表2 $t_1 \sim t_4$ 列) を対応させ、各 x_i を t_1, t_2, t_3, t_4 の関数とせず最小ハミング距離法で1次式のあてはめを行くと、下記のようになる

$$(5) \quad \left\{ \begin{array}{ll} \hat{x}_1 = t_1 + t_2 & (2) \\ \hat{x}_2 = t_2 + t_3 & (2) \\ \hat{x}_3 = t_2 + t_4 & (2) \\ \hat{x}_4 = t_1 & (2) \\ \hat{x}_5 = t_1 + t_3 & (1) \\ \hat{x}_6 = t_2 & (2) \\ \hat{x}_7 = t_1 + t_3 + t_4 & (2) \end{array} \right.$$

(カッコ内はあてはめの誤差 $d(x_i, \hat{x}_i)$ と示す)

これは $GF(2)^m = GF(2)^7$ の $k=4$ 次元線形部分空間 S を表現する1次式である。 S の各点かブロックあるいはベクトルでありこれを表すに示す。

record

表 2 (R)

record	t_1	t_2	t_3	t_4	x_1	x_2	x_3	x_4	x_5	x_6	x_7
0	0	0	0	0	0	1	0	0	0	0	0
1	0	0	0	1	0	0	1	0	0	0	1
2	0	0	1	0	1	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0	1	0	1
4	0	1	0	0	1	1	0	0	0	1	1
5	0	1	0	1	1	1	0	0	0	1	1
6	0	1	1	0	1	1	0	0	1	1	0
7	0	1	1	1	1	0	0	0	0	0	0
8	1	0	0	0	1	0	1	1	1	0	0
9	1	0	0	1	0	0	0	1	1	0	0
10	1	0	1	0	1	1	0	1	0	1	0
11	1	0	1	1	1	1	1	0	1	0	1
12	1	1	0	0	0	1	1	0	1	1	1
13	1	1	1	0	0	1	0	1	1	1	0
14	1	1	1	1	0	0	1	1	0	1	1
15	1	1	1	1	0	0	0	1	0	1	1

表 3 (S)

stored record

	\hat{x}_1	\hat{x}_2	\hat{x}_3	\hat{x}_4	\hat{x}_5	\hat{x}_6	\hat{x}_7
0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	1
2	0	1	0	0	1	0	1
3	0	1	0	0	1	0	1
4	1	1	1	0	0	1	1
5	1	1	1	0	0	1	1
6	1	1	0	0	1	1	0
7	1	1	0	0	1	1	0
8	1	0	1	1	1	0	0
9	1	0	1	1	1	0	0
10	1	1	0	1	0	0	1
11	1	1	1	1	0	0	1
12	0	1	1	1	1	1	0
13	0	1	0	1	1	1	0
14	0	0	1	1	0	1	0
15	0	0	0	1	0	1	1

stored record

0	1	2	3
1	3		
2	1	2	3
3	1	2	3
4			
5			
6	1	2	3
7	2	3	
8	2	3	
9	2	3	
10	2	3	
11	2	3	
12	0	2	3
13	0	2	3
14	0	2	3
15	0	2	3

§3 ストアアルゴリズム

(5)の一般形を

$$(6) \quad \hat{x}_i = b_{i1}t_1 + \dots + b_{ik}t_k, \quad i=1, \dots, m$$

としておくと、レコード $x = (x_1, \dots, x_m)$ をキー $t_p = (t_{p1}, \dots, t_{pk})$ に対応するバケツト $\hat{x}_p = (\hat{x}_{p1}, \dots, \hat{x}_{pm})$ にストアするか否かを定めるアルゴリズムは;

$$(7) \quad I = \{ i : x_i = \hat{x}_{pi}, i=1, \dots, m \}$$

とし、 $I = \phi$ (空集合) ならストアしない、 $I \neq \phi$ なら

$$(8) \quad x_i = b_{i1}t_1 + \dots + b_{ik}t_k, \quad i \in I \quad \left(\begin{array}{l} \text{この番号の若い順} \\ \text{に並べる} \end{array} \right)$$

なる方程式の解 t_1, \dots, t_k (一般には不定解となるが、 t_1, t_2, \dots の順に掃き出し、ピボットは(8)式の若い順番に選ぶという方法で、基準形に変換したときの基底解をとるというルール*に従うものとする) が t_p に一致すればストアし、そうでなければストアしない。

たとえば表2のレコード② $x = (1100101)$ を $\hat{x}_{13} = (0101110)$ にストアすべきかを判定しよう。まず $I = \{2, 3, 5\}$ となり方程式(8)は

$$(9) \quad \begin{cases} t_2 + t_3 & = 1 \\ t_2 + t_4 & = 0 \\ t_1 + t_3 & = 1 \end{cases}$$

となるが、これにルール*による掃き出しをほどこして変形すると、

$$(10) \quad \begin{cases} t_2 + t_4 = 0 \\ t_3 + t_4 = 1 \\ t_1 + t_4 = 0 \end{cases}$$

なる基準形が得られる。この基底解は非基底変数 t_4 を 0 とおいて得られるもので、 $(t_1, t_2, t_3, t_4) = (0, 1, 0, 0)$ となるが、これは $x_{13} = (1, 1, 0, 1)$ と一致しないから、②は \hat{x}_{13} にはストアしない。

この方法で①, ①, ②, ③を S の各点にストアした結果が表 3 の右の部分である。(8) の解を一意に決めるためには、上記のような方法でもよいし、解 \mathcal{S} の全体のうちこれを 2 進数とみて一番若いものを優先するというルールでもよい。以上のものは [1] のものと本質的に同等である。ただし [1] では検索をより容易にするため、上のバケット \hat{x}_p をさらに細かなサブバケットに分割する方法がとられている。

§4 検索アルゴリズム

本項目の負内に対する検索とは、 a_{i_1}, \dots, a_{i_k} が与えられたとき、

$$(11) \quad x_{ij} = a_{ij}, \quad j=1, \dots, k$$

となるような, ストアされている, Lコード $\alpha = (\alpha_1, \dots, \alpha_m)$ (のLコード番号) をすべて取り出すことである. [1] などでは t 項目以下の箇門に即答できるために, S を決定する (5) に対応する 1 次式などの t 個も 1 次独立であること (強士 t の条件) を要請している. こうすると (11) に対応する方程式

$$(12) \quad a_{ij} = b_{ij,1} t_1 + \dots + b_{ij,t} t_t, \quad j \in \{1, \dots, t\}$$

かつねに解をもつから, この解 (ストアの場合と同一のルール*で基準形の基底解として一意に決める) (t_1, \dots, t_t) に対応するバケットの中に, (11) をみたすすべての Lコードが含まれている. これが [1] の検索アルゴリズムである.

われわれの S は R に適合しているかわり, 強士 t の条件をみたしていない. したがって任意に与えられた t 個の値 a_{ij} ($j=1, \dots, t$) に対して (12) の解が必ずしも存在しない. しかしその場合は, $\{i_1, \dots, i_t\}$ を適当にいくつかのグループ G_1, \dots, G_s にわけ ($\{i_1, \dots, i_t\} = G_1 \cup \dots \cup G_s$, G_i, G_j は必ずしも disjoint でなくてもよい), 各 G_i に対して上と同様に検索して (12) の $j \in G_i$ に対する式を解をもつようになるまで細分する) 得た Lコードの集合 R_i の共通部分

$$(13) \quad R_1 \cap \dots \cap R_s$$

とすれば、これが求めるものであることは容易にわかる。

このように検索ルールを拡張すれば、尤項目と言わず任意数項目に対する検索可能な（必ずしも即答でないが）ファイリングシステムが得られる。

§5 今後の問題

ここでのべた最小ハミング距離法は、 $GF(2)$ の場合は、リードマラーコードのデコード方式と同等でうまく行くが、一般の $GF(q)$ の場合には必ずしも確立されていらないようである。今後の研究に待つ。

またここでは、与えられたデータ R に S をあてはめるのに、表2に示すようなキー (t_1, \dots, t_k) を勝手に与えたが、キーの順序によつて当てはめられた式(5)は大きく変る。実際にはキーはファイル R の中のいくつかの項目 $(x_{j_1}, \dots, x_{j_k})$ によつて決まることが多いので、 (t_1, \dots, t_k) のかわりに、このようなそのファイルに固有のキー $(x_{j_1}, \dots, x_{j_k})$ を用いることが望ましい。こうして残りの x_j をキーの1次式として、最小ハミング距離法を用いて、表現すればよいのである。

このようなキーが R の中にない場合はどうすべきか。それには、 x_1, \dots, x_m の1次式としていくつかの t_i を構成することが考えられる。つまり

$$(14) \quad t_{\nu i} = w_{i1} x_{\nu 1} + \dots + w_{im} x_{\nu m} + e_{\nu i}$$

$$\nu = 1, \dots, n, \quad i = 1, \dots, k$$

とし、 $\{e_{\nu i} : \nu = 1, \dots, n, i = 1, \dots, k\}$ のハミルトンウェイトが最小になるように w_{ij} を決めて、 x_1, \dots, x_m の情報 t_1, \dots, t_k に集約すればよい。これは丁度実数場での主成分分析に相当するものであろうが、^[5]この効率よいアルゴリズムも未完成である。

またたとえ上のような意味で R に適合した S が得られたとしても、これが R に無関係に決めた S より低い冗長度を与えるかどうかは、直観的にはうなづけるとしても、理論的な証明はまだなされていない。さし当りては、いくつかの例でシミュレーションによって実験的に確かめてみるのが手始めであらう。

参考文献

- [0] F. J. Mac Williams & N. J. A. Sloane "The Theory of Error Correcting Codes" North Holland (1977)
- [1] R. C. Bose, C. T. Abraham & S. P. Gosh
File Organization of Records for Multiple valued Attributes for Multiattribute

- Queries" IBM Research Paper RC-1886 (1967)
- [2] R.C. Bose & Gary. G. Koch "The Design of Combinatorial Information on Retrieval Systems for Files with Multivalued Attributes" SIAM J. App. Math. (1969)
- [3] 高橋磐郎 "ガロア体の離散データ処理への応用"
数埋科学 (1979年11月)
- [4] I. Takahashi "Filing Schema by Galois Functions" Information & Control (投稿予定)
- [5] 奥野忠一他 "多変量解析法" 日科技連出版 (1971)