数理解析研究所講究録
第 404 巻 1980 年 90-100

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 $n=16$ $(\leqq q^k = 2^4)$ のものである．q が素数べきのとき，各項目のカテゴリを $GF(q)$ の元と考えると，ファイル R は $GF(q)^m$ の大きさ n の部分集合である．

Bose 等は，$GF(q)^m$ の中に強さ t の k 次元線形部分空間 S（強さ t の直交表）を考え，S の各点をブロック（バケットの集り）とみなすファイルによって，t 項目質問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もつ場合にしか，その有用性が謳われない．実際に

Rの点は $GF(q)^m$ の中にきわめて疎にしかもかなり偏って分布している. [1][2]

ここでは与えられたRにある意味で最もよく適合する線形部分空間Sにもとづ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マラーコードのデコード方式と同当のものである. [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 $y=f(x_1,x_2,x_3)$ の値が表1のように与えられたとき, これに1次式

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

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

をあてはめる問題を考える. その原則は誤差ベクトル $\boldsymbol{e}=(e_p : p=1,\cdots,n)$ のハミングウェイトが最小になるように, つまり $\boldsymbol{y}=(y_p : p=1,\cdots,n)$ と $(a_0+a_1x_{p1}+a_2x_{p2}+a_3x_{p3} : p=1,\cdots,n)$ 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

ング距離が最小になるように、a_0, a_1, a_2, a_3 をきめるものとする。

このような a_i を最小ハミング距離推定 $\hat{a}_i$ と呼ぶとすると、GF(2) の場は、これはつぎのような式で決められることがわかる。

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

$$
(3)\quad \hat{a}_0 = \mathrm{maj}\{ y'_\nu ; \nu = 1, \cdots, n \},
$$

$$
y'_\nu = y_\nu - \hat{a}_1 x_{\nu 1} - \hat{a}_2 x_{\nu 2} - \hat{a}_3 x_{\nu 3},\quad \nu = 1, \cdots, n
$$

ここで maj は多数決関数、つまり $\mathrm{maj}\{0, 0, 1\} = 0$, $\mathrm{maj}\{1, 0, 1\} = 1$ など。tie が起る場合および $d(\boldsymbol{y}, \hat{\boldsymbol{y}}) > 2^{m-2}$ の場合は [4] 参照。ただし $\hat{\boldsymbol{y}} = \hat{a}_0 + \hat{a}_1 \boldsymbol{x}_1 + \hat{a}_2 \boldsymbol{x}_2 + \hat{a}_3 \boldsymbol{x}_3$　$(\boldsymbol{x}_i = (x_{\nu i} : \nu = 1, \cdots, n))$ で $d(\boldsymbol{y}, \hat{\boldsymbol{y}})$ は $\boldsymbol{y}$ と $\hat{\boldsymbol{y}}$ とのハミング距離。

表1に対して (2), (3) を適用すると、

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

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

におけるデコード方式と同等である．一般にl次式のあてはめも同様な方法で可能であり，これはl次のリードマラ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の関数とみて最小ハミング距離法で1次式のあてはめを行うと，つぎのようになる

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

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

これは$GF(q)^m = GF(2)^7$の$k=4$次元線形部分空間Sを表現する1次式である．Sの各点がブロックあるいはベクトルであり，これを表3に示す．

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	1	1	0	1	0
5	0	1	0	1	1	1	0	0	0	1	1
6	0	1	1	0	1	1	1	0	1	1	1
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	1	0	0	1
12	1	1	0	0	0	1	1	0	1	1	1
13	1	1	0	1	0	1	0	1	1	1	0
14	1	1	1	0	0	0	1	1	0	1	0
15	1	1	1	1	0	0	0	1	0	1	1

表3 (S)

$\hat{x}_1$	$\hat{x}_2$	$\hat{x}_3$	$\hat{x}_4$	$\hat{x}_5$	$\hat{x}_6$	$\hat{x}_7$	stored record			
0	0	0	0	0	0	0	0	1	2	3
0	0	1	0	0	0	1	1	3		
0	1	0	0	1	0	1	0	1	2	3
0	1	1	0	1	0	0	0	3		
1	1	1	0	0	1	0	0	1	2	3
1	1	0	0	0	1	1	2			
1	0	1	0	1	1	1	1	2	3	
1	0	0	0	1	1	0	2			
1	0	0	1	1	0	1	1	2	3	
1	0	1	1	1	0	0	2	3		
1	1	0	1	0	0	0	0	2		
1	1	1	1	0	0	1	2	3		
0	1	1	1	1	1	1	0	1	2	3
0	1	0	1	1	1	0	0			
0	0	1	1	0	1	0	1			
0	0	0	1	0	1	1				

§3 ストアアルゴリズム

(5)の一般形を

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

としておくと，レコード $\boldsymbol{x} = (x_1, \cdots, x_m)$ をキー $\boldsymbol{t}_\nu = (t_{\nu 1}, \cdots, t_{\nu k})$ に対応するバケット $\hat{\boldsymbol{x}}_\nu = (\hat{x}_{\nu 1}, \cdots, \hat{x}_{\nu m})$ にストアするか否かを決めるアルゴリズムは；

(7) $I = \{ i : \ x_i = \hat{x}_{\nu i}, \ i=1,\cdots,m \}$

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

(8) $x_i = b_{i1}t_1 + \cdots + b_{ik}t_k \quad , i \in I$（$i$の番号の若い順に並べる）

なる方程式の解 $t_1, \cdots, t_k$（一般には不定解となるが，$t_1, t_2 \cdots$ 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 $\boldsymbol{t}_\nu$ 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 $\boldsymbol{x} = (1\ 1\ 0\ 0\ 1\ 0\ 1)$ を $\hat{\boldsymbol{x}}_{13} = (0\ 1\ 0\ 1\ 1\ 1\ 0)$ 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

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

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，

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

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 $\boldsymbol{t}_{13} = (1, 1, 0, 1)$ 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⓪，①，②，③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 $\boldsymbol{t}$ 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

t 項目の質問に対する検索とは，$a_{i_1}, \cdots, a_{i_t}$ が与えられたとき，

$$(11)\qquad x_{i_j} = a_{i_j} \quad , \; j = 1, \cdots, t$$

となるような，ストアされている，レコード $\boldsymbol{x}=(x_1,\cdots,x_m)$（のレコード番号）をすべて取り出すことである．[1] などでは t 項目以下の質問に即答できるために，S を決定する (5) に対応する 1 次式がどの t 個も 1 次独立であること（強さ 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) $\quad a_{i_j}=b_{i_j 1}t_1+\cdots+b_{i_j k}t_k \quad, j\in\{1,\cdots,t\}$

が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）$(t_1,\cdots,t_k)$ に対応するバケットの中に，(11) をみたすすべてのレ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，強さ 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_{i_j} $(j=1,\cdots,t)$ 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，$\{i_1,\cdots,i_t\}$ を適当にいくつかのグループ $G_1,\cdots,G_s$ にわけ（$\{i_1,\cdots,i_t\}=G_1\cup\cdots\cup G_s$，$G_i, G_j$ は必ずしも disjoint でなくてよい），各 G_i に対して上と同様に検索して（(12) の $j\in G_i$ に対する式が解をもつようになるまで細分する）得たレコードの集合 R_i の共通部分

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

をとれば，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，t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　今后の問題

ここでのべた最小ハミング距離法は，GF(2) の場は，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ないようである。今后の研究に待つ。

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（$t_1, \cdots, t_k$）を勝手に与えたが，キーの順序によっ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（$x_{j_1}, \cdots, x_{j_k}$）によって決まることが多いので，（$t_1, \cdots, t_k$）のかわりに，このようなそのファイルに固有のキー（$x_{j_1}, \cdots, x_{j_k}$）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，$x_1, \cdots, x_m$ 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) $t_{\nu i} = w_{i1} x_{\nu 1} + \cdots + w_{im} x_{\nu m} + e_{\nu i}$

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

とし，$\{ e_{\nu i} : \nu = 1, \cdots, n, \ i = 1, \cdots, k \}$ のハミングウェイトが最小になるように w_{ij} を決めて，$x_1, \cdots, x_m$ の情報を $t_1, \cdots, t_k$ に集約すればよい．これは2元体の場での主成分分析に相当するものであるが，[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．さし当っては，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ろう．

参考文献

[0] F. J. MacWilliams & 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" 数理科学 (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)