

シグネチャファイルの新しい構成方法の テストデータからの評価

二神常爾（聖学院大学非常勤講師）

文献検索で、転置ファイルとともに、シグネチャファイルは古くから用いられてきた。シグネチャファイルはフォルスドロップを生じる欠点がある。本研究ではフォルスドロップが 0 になるようなシグネチャファイルの新しい構成方法を提案し、テストデータから評価した。文献数 1793、キーワード数 1360 の転置ファイルを用いて転置ファイルとシグネチャファイルの性能を比較した。シグネチャファイルをつくるプログラムと転置ファイルを Golomb 符号により圧縮するプログラムを比較したところ、メモリ量、処理時間の面で転置ファイルの方が優れていることが分かった。また、両者の検索プログラムを比較したところ、メモリ量、検索時間の面で転置ファイルの方が優れていた。一方、転置ファイルでは、各文献が含むキーワードのリストから各キーワードを含む文献のリストを作らねばならず、その分メモリ量と処理時間がかかることが分かった。また、プログラムの行数はシグネチャファイルの方が小さく、簡潔なプログラムを書けることが分かった。以上からいくつかの点ではシグネチャファイルの方が転置ファイルより優れていることが確認された。

Evaluation on a new construction method of a signature file

Tsuneji Futagami (Seigakuin University)

In document retrieval, an inverted file and a signature file have been used. Retrieval by a signature file has a drawback that it causes false drops. In this study, a new construction method of a signature file causing no false drops is proposed. It is evaluated from test data. Construction and retrieval by using an inverted file and a signature file were evaluated from a viewpoint of memory amount and transaction time. It was proved that an inverted file is superior from these viewpoints. On the other hand, a signature file is superior in that a list of keywords each document contains must be created in inverted file and such task is not necessary in a signature file. Also length of programs is shorter in a signature file which means easier coding. In some points, a signature file is superior than an inverted file.

1. はじめに

我々に入手できるデータの量の増大とともに、大量のデータの中から必要な情報を探す情報検索の技術はますます重要になっている。図書館等でキーワードを入力し、そのキーワードに関連する文献をオンラインで探す文献検索は情報検索の一分野であり、古くから研究されてきた。キーワードによる文献検索にはいくつかの方法がある[1]。第 1 は文献全体を走査してキーワードを文献が含むかどうかを調べる方法である。この方法ではメモリへの負荷は小さくなるが、検索時間は長くなる。第 2 はキーワードと文献の対応表を利用する方法である。この対応表としてよく知られたものに転置ファイルやシグネチャファイルがある。転置ファイルを利用すると高速な検索が可能だが、メモリへの負荷が大きくなる難点がある。そこで、転置ファイルを圧縮し、転置ファイルを格納するのに必要なメモリ量を低減する研究が多くなされてきた[2-4]。一方、シグネチャファイルを用いる検索ではフォルスドロップが生じる欠点がある。つまり、検索の結果得られる文献集合（ドロップ）は検索条件に適合する適合文献（アクチュアルドロップ）だけでなく、検索条件に適合しない不適合文献（フォルスドロップ）を含む[5,6]。本論文では、シグネチャファイルの新しい構成方法を提案し評価する。この方法ではパラメータを適切に選ぶことによりフォルスドロップが 0 になるように、文献検索システムを構築することが可能である。このような構成方法はこれまで明らかにされてこなかった。また、転置ファイルの例を用いた評価により提案した構成方法が有効なことを示す。なお、最近ではシグネチャファイルは XML 文書のキーワード検索などでも用いられている[7]。

2. シグネチャファイル

文献とキーワードの総数をそれぞれ N_0 と M とおく。キーワード j ($1 \leq j \leq M$) に対してワードシグネチャ $\bar{B}_j$ を割り当てる。ワードシグネチャは各ビットが 0 か 1 の長さ n ($n < M$) の二元列である。ワードシグネチャの重み(値 1 をもつビット数)を w とする。 n と w はすべてのキーワード j ($1 \leq j \leq M$) に対して同じ値をとるとする。

ワードシグネチャ 1	1 1 1 0 0 0 0 0 0	文献のシグネチャ $\bar{C}_i$ ($1 \leq i \leq N_0$) は、 文献が含む全ての キーワードのワー ドシグネチャを重 ね合わせることに
ワードシグネチャ 2	1 0 0 1 0 0 1 0 0	
ワードシグネチャ 3	1 0 0 0 1 0 0 0 1	
文献のシグネチャ	1 1 1 1 1 0 1 0 1 (論理和)	

図1 文献のシグネチャの生成の例 ($li=3, w=3, n=9$)

より得られる(図1)。すなわち、 i ($1 \leq i \leq N_0$) 番目の文献が l_i 個のキーワードを含むとき、文献のシグネチャ $\bar{C}_i$ は次の重ね合わせにより与えられる。

$$\bar{C}_i = \bar{B}_{a_{i,1}} \cup \bar{B}_{a_{i,2}} \cup \dots \cup \bar{B}_{a_{i,l_i}}, \quad (1)$$

$a_{i,j}$ は i 番目の文献が含む j ($1 \leq j \leq l_i$) 番目のキーワードである. $\bar{B}_{a_{i,j}}$ はそのキーワードに対するワードシグネチャである. あるビットについて, l_i 個のワードシグネチャのビットがすべて 0 以外の場合は 1 になる. シグネチャファイル C はその行ベクトルが文献のシグネチャ $\bar{C}_i$ ($1 \leq i \leq N_0$) となっている行列である. すなわち, シグネチャファイルの大きさは $N_0 \times n$ である. 簡単のために 1 つの質問キーワード q を含む文献を求める検索を考える. 質問キーワードに対するワードシグネチャを $\bar{B}_q$ とする. 文献検索は

$$\bar{C}_i \supseteq \bar{B}_q, \quad (2)$$

を満たす文献 i ($1 \leq i \leq N_0$) を見出すことである. これは文献のシグネチャ $\bar{C}_i$ が, 質問キーワード q のシグネチャ $\bar{B}_q$ が値 1 をもつ全てのビットに値 1 をもつことを意味する. ただし, 条件式(2)は文献が質問キーワード q を含むための必要条件であるが十分条件でない. フォルスドロップが生じる場合には, 条件式(2)が満たされるが文献は実際には質問キーワード q を含んでいない. すなわち,

$$q \notin \{a_{i,1}, a_{i,2}, \dots, a_{i,l_i}\}, \quad (3)$$

が成り立つ. $\lambda_{i,j}$ を二つのワードシグネチャ $\bar{B}_i$ と $\bar{B}_j$ の間のオーバーラップとして定義する. つまり, $\lambda_{i,j}$ は 2 つのワードシグネチャ $\bar{B}_i$ と $\bar{B}_j$ がともに値 1 をもつビットの数である. $\lambda_{i,j}$ の最大値を $\lambda_{\max}$ とする (ただし, $i \neq j$). すなわち,

$$\lambda_{\max} = \max \lambda_{i,j}, \quad (4)$$

である. 長さ n , 重み w , 最大のオーバーラップ $\lambda_{\max}$ のシグネチャの集合の異なる要素の数の最大値を $M_{\max}(n, w, \lambda_{\max})$ とする. すると, 次の関係式が成り立つ[8].

$$M_{\max}(n, w, \lambda_{\max}) \leq \binom{n}{\lambda_{\max} + 1} \binom{w}{\lambda_{\max} + 1}, \quad (5)$$

3. 提案方法

式(1)より文献のシグネチャ $\bar{C}_i$ は l_i 個のワードシグネチャの重ね合わせからなるので, 1 つの文献に含まれるキーワード数の最大値 $l_{\max}$ ($l_{\max} = \max l_i$) に対して

$$l_{\max} < (w/\lambda_{\max}), \quad (6)$$

となるようにパラメータ w , $\lambda_{\max}$ の値を選べば, いかなる質問キーワードに対してもこ

の文献検索システムではフォルスドロップが起こらない. 以下では $\lambda_{\max} = 1$ を仮定する. このとき, 式(5)は

$$M_{\max}(n, w, 1) \leq \frac{n(n-1)}{w(w-1)}, \quad (7)$$

となる.

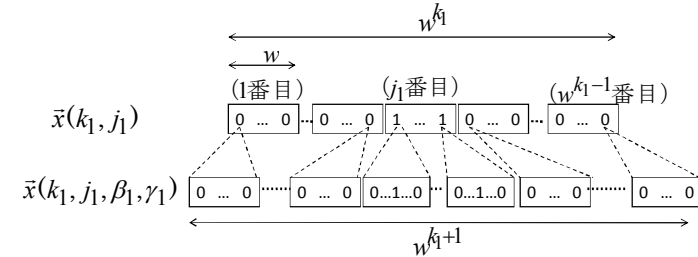


図2 シグネチャファイルの構築方法

次に, シグネチャファイルの構成方法を考える (図2). 1つの長さ w , 重み w のシグネチャ (すべてのビットが 1) と $(w^{k_1-1} - 1)$ 個の長さ w , 重み 0 のシグネチャ (すべてのビットが 0) をつなげて長さ w^{k_1} , 重み w のシグネチャをつくる ($1 \leq k_1 \leq k$). この際に長さ w , 重み w のシグネチャを先頭から j_1 ($1 \leq j_1 \leq w^{k_1-1}$) 番目に置くとして, この長さ w^{k_1} , 重み w のシグネチャを $\bar{x}(k_1, j_1)$ によって表わす. すると, $\bar{x}(k_1, j_1)$ の値 1 をもつ w 個のビットのうち i 番目のものの先頭からの位置を $u_{0,i}$ ($1 \leq i \leq w$) とすると, 次が成り立つ.

$$u_{0,i} = (j_1 - 1)w + i, \quad (8)$$

次に, シグネチャ $\bar{x}(k_1, j_1)$ の各ビットに長さ w のシグネチャを割り当てて, これをつなげて長さ w^{k_1+1} のシグネチャをつくる. シグネチャ $\bar{x}(k_1, j_1)$ の値 0 をもつビットには長さ w , 重み 0 のシグネチャを割り当てる. 一方で, シグネチャ $\bar{x}(k_1, j_1)$ の $u_{0,i}$ ($1 \leq i \leq w$) 番目のビットは 1 なので, このビットに長さ w , 重み 1 のシグネチャを割り当てる. この長さ w のシグネチャ内の値 1 のビットの位置 $v_{1,i}$ ($1 \leq i \leq w$) を次のように与える.

$$v_{1,i} = [\{(\beta_1 + \gamma_1 i) \bmod w\} + 1], \quad (9)$$

ここで, β_1, γ_1 は $0 \leq \beta_1, \gamma_1 \leq w-1$ を満足する整数である. こうしてつくられる長さ w^{k_1+1} , 重み w のシグネチャを $\bar{x}(k_1, j_1, \beta_1, \gamma_1)$ により表わす. シグネチャ $\bar{x}(k_1, j_1, \beta_1, \gamma_1)$ 内での i ($1 \leq i \leq w$) 個目の値 1 の位置を $u_{1,i}(k_1, j_1, \beta_1, \gamma_1)$ により表わす. すると,

$$u_{1,i} = w(u_{0,i} - 1) + v_{1,i}, \quad (10)$$

が成り立つ。

以下、同様の手順で各ビットに長さ w のシグネチャを割り当ててつなげることを繰り返す。値 1 をもつビットには重み 1 のシグネチャを割り当て、値 0 をもつビットには重み 0 のシグネチャを割り当てる。最終的に $2(k-k_1)$ 個のパラメータ

$\beta_1, \gamma_1, \dots, \beta_{k-k_1}, \gamma_{k-k_1}$ ($0 \leq \beta_1, \gamma_1, \dots, \beta_{k-k_1}, \gamma_{k-k_1} \leq w-1$) を用いて長さ w^k 、重み w のシグネチャ $\bar{x}(k_1, j_1, \beta_1, \gamma_1, \dots, \beta_{k-k_1}, \gamma_{k-k_1})$ を得る。このシグネチャ内での i ($1 \leq i \leq w$) 番目の値 1 の位置を $u_{k-k_1,i}(k_1, j_1, \beta_1, \gamma_1, \dots, \beta_{k-k_1}, \gamma_{k-k_1})$ とすると

$$u_{k-k_1,i} = w(u_{k-k_1-1,i} - 1) + v_{k-k_1,i} \quad (1 \leq i \leq w), \quad (11)$$

が成り立つ。ここで $u_{k-k_1-1,i}$ は長さ w^{k-1} のシグネチャ $\bar{x}(k_1, j_1, \beta_1, \gamma_1, \dots, \beta_{k-k_1-1}, \gamma_{k-k_1-1})$ の w 個の 1 をもつビットのうち、 i 番目のものの位置であり、 $v_{k-k_1,i}$ は次式で与えられる。

$$v_{k-k_1,i} = [\{(\beta_{k-k_1} + \gamma_{k-k_1} i) \bmod w\} + 1], \quad (12)$$

上記のようにつくられる長さ w^k 、重み w のシグネチャの個数を求める。 $2(k-k_1)$ 個のパラメータ $(\beta_1, \gamma_1, \dots, \beta_{k-k_1}, \gamma_{k-k_1})$ は各々 w 個の整数のいずれかの値をとる。従って、パラメータ (k_1, j_1) の組に対して、シグネチャ $\bar{x}(k_1, j_1, \beta_1, \gamma_1, \dots, \beta_{k-k_1}, \gamma_{k-k_1})$ の個数は最大 $w^{2(k-k_1)}$ 個存在する。パラメータ k_1, j_1 は $1 \leq k_1 \leq k$ 、 $1 \leq j_1 \leq w^{k-1}$ を満足する整数のいずれかなので、シグネチャの総数は重複がなければ

$$\sum_{k_1=1}^k w^{k_1-1} \cdot w^{2(k-k_1)} = \frac{w^k(w^k - 1)}{w(w-1)}, \quad (13)$$

となる。 w が素数の場合には、このようにつくられたシグネチャのオーバーラップは 0 または 1 である (付録)。この場合、上記のようにつくられる長さ $n(=w^k)$ 、重み w のシグネチャは全て異なり (重複はない)、その総数が式(13)によって与えられる。これは式(7)で等号を満たす場合に相当する。

4. テストデータによる評価

4.1 フォルスドロップ数と期待値

二神(2002)[2]で作成した転置ファイルを用いて提案アルゴリズムを評価した。転置ファイルのパラメータは文献数 $N_0 = 1793$ 、異なりキーワード数 $M = 1360$ 、延ベキーワード数 7789 である。この転置ファイルを元に提案するアルゴリズムに従って、シグネチャファイルを作成した。作成のパラメータとして、 $w = 7, k = 3, n = 343$ を選んだ。シグネチャの長さの異なりキーワード数に対する割合 (n/M) は 0.25 である。

このシグネチャファイルを用いて、1 個の質問キーワードに対する検索を行い、ドロ

ップ数を得た。1360 個の異なりキーワードに対するドロップの合計は 7790 個であり、延ベキーワード数 (アクチュアルドロップ数) 7789 より 1 個多くなった。これはフォルスドロップが 1 件生じたことを意味する。フォルスドロップとなった 1 件の文献は 11 個のキーワードを含んでいた。

3 章で述べたように、提案するシグネチャファイルの構成方法に従えば、 $(w-1)$ 個 (今の場合、 $w-1=6$) 以下のキーワードを含む文献はフォルスドロップになりえない。6 個以下のキーワードを含む文献がフォルス・ドロップとならないことが実際確認された。

従来のシグネチャファイルの構成方法によれば、フォルスドロップが生じる確率は

$$p(n, w, t) = (1 - (1 - 1/n)^w)^w, \quad (14)$$

により与えられる[9]。これは、長さ n のシグネチャ内の値 1 をもつビット位置を、ハッシュ技法などを用いてランダムに決定する場合に対応する。ここで、 t は 1 つの文献が含むキーワード数である。評価に用いた転置ファイルの t の最小値は 1、最大値は 20 であった。 t 個のキーワードを含む文献の数を l_t とする。すると、フォルス・ドロップ数の期待値は

$$\sum_{t=1}^{20} (M-t) l_t p(n, w, t) \quad (15)$$

によって与えられる。ここで、上に示したパラメータを用いて計算すると、1.34 を得る。従って、提案方法によって生じたフォルス・ドロップ数(1)は期待値(1.34)よりも若干小さい。

4.2 シグネチャファイルと転置ファイルの比較

シグネチャフ

ァイルまたは転

置ファイルを作

成し、検索を行

う過程を 3 つの

プログラムに分

けて評価した

(図 3)。プロ

グラム 1 及びプ

ログラム 2 は、

各文献が含むキーワード名のリストからシグネチャファイルまたは圧縮した転置ファイルを作成する過程に対応する。プログラム 3 は、シグネチャファイルまたは圧縮した転置ファイルを用いて、1 つの検索キーワードに対して検索を行う過程に対応する。

プログラミングは C 言語により行った。それぞれのプログラムのメモリ量と処理時間とプログラムの行数を評価した。メモリ量はプログラムで用いる変数のビット長の総和により求めた。また、各プログラムから出力したファイル (ファイル 2~4) の大きさを

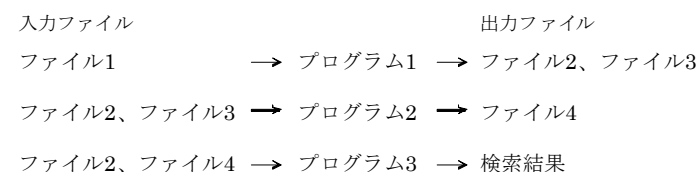


図3 プログラムと入出力ファイル

出力に用いた変数のビット長の総和により求めた。

プログラムの処理時間は、プログラムを何回か繰り返して実行した際実行にかかる時間を繰り返し回数で割って求めた。ただし、ファイルからの入力、ファイルへの出力は処理時間に含めなかった。

4.2.1 シグネチャファイルの場合

4.2.1.a キーワードリストの作成 (プログラム 1)

ファイル1: 文献番号0: キーワード名 $a_{1,1}$, キーワード名 $a_{1,2}$,...
文献番号1: キーワード名 $a_{2,1}$, キーワード名 $a_{2,2}$,...
:

ファイル2: キーワード番号0: キーワード名 b_1 アルファベット順に
キーワード番号1: キーワード名 b_2 各キーワード名に
: キーワード番号を
割り振る。

ファイル3: 文献番号0: キーワード番号 $n_0(a_{1,1})$, キーワード番号 $n_0(a_{1,2})$,...
文献番号1: キーワード番号 $n_0(a_{2,1})$, キーワード番号 $n_0(a_{2,2})$,...
:

図4 シグネチャファイル作成のためのファイル

をアルファベット順に並べたものである。アルファベット順に0から1359までのキーワード番号を順に割り当てた。ファイル3は、ファイル1をキーワード名に代わってキーワード番号を用いて書き直したものである。

4.2.1.b シグネチャファイルの作成 (プログラム 2)

ファイル3からファイル4 (シグネチャファイル) を作成した。シグネチャファイルは、各キーワードに対するシグネチャ (ワードシグネチャ) と各文献について文献が含むキーワードのワードシグネチャを重ね合わせて作成した文献シグネチャからなる。シグネチャは、文字変数の配列により表現した。長さ n の文字変数の配列により長さ $8n$ ビットのシグネチャを表現した。

4.2.1.c 検索 (プログラム 3)

ファイル2及びファイル4を用いて検索を行った。キーワード名 (質問キーワード) を入力し、そのキーワードに対応するキーワード番号を求めた。これはファイル2を参照して、2分探索法で求めた。次に、得られたキーワード番号に対応するワードシグネチャと各文献に対応する文献シグネチャ (ファイル4) を用いて、検索を行った。図5

ファイル1は、各文献が含むキーワード名のリストである。このファイル1からファイル2 (キーワードリスト) を作成した (図4)。これは、ファイル1に出現する全てのキーワード名 (全1360個)

に示すように質問キーワードに対応するワードシグネチャが値1をもつビット位置に対応するビットスライスの論理積 (AND) をとった。論理積で値1をもつビット位置の集合がドロップに対応する。

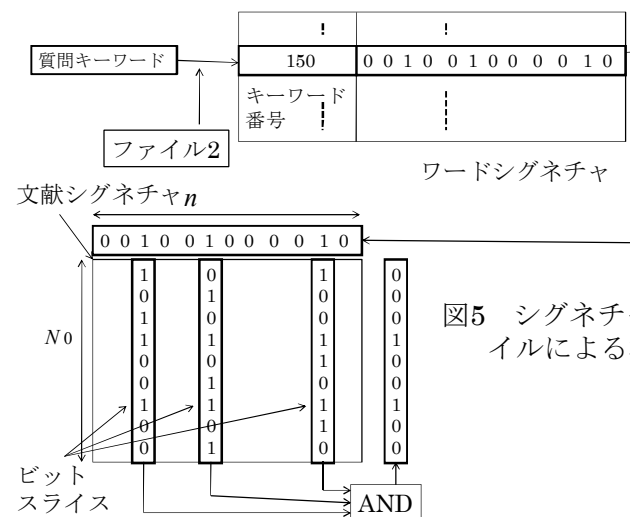


図5 シグネチャファイルによる検索

4.2.2 転置ファイルの場合

4.2.2.a 転置ファイルの作成 (プログラム 1)

ファイル1: 文献番号0: キーワード名 $a_{1,1}$, キーワード名 $a_{1,2}$,...
文献番号1: キーワード名 $a_{2,1}$, キーワード名 $a_{2,2}$,...
:

ファイル2: キーワード番号0: キーワード名 b_1 アルファベット順に
キーワード番号1: キーワード名 b_2 各キーワード名に
: キーワード番号を
割り振る。

ファイル3: キーワード番号0: 文献番号 $n_1(b_1)$, 文献番号 $n_1(b_1)$,...
(転置ファイル) キーワード番号1: 文献番号 $n_1(b_2)$, 文献番号 $n_1(b_2)$,...
:

図6 転置ファイル作成のためのファイル

ファイル1はシグネチャファイルの場合と同様に、各文献が含むキーワード名のリストを並べたものである (図6)。このファイルから4.2.1.aと同じキーワードリストを作成した。ファイル3は転置

ファイルであり、各キーワード番号とそのキーワードを含む文献番号のリストである。

4.2.2.b 転置ファイルの圧縮（プログラム2）

x	Golomb符号
1	00
2	010
3	011
4	100
5	1010
6	1011
7	1100
8	11010
9	11011
10	11100

表1 Golomb符号 ($b=3$)

ファイル3からファイル4（圧縮ファイル）を作成した。各キーワードを含む文献番号の差分値をGolomb符号により圧縮した。整数 x を符号化する方法は以下の通りである。ある整数 $b(>0)$ に対して、 $q = \lfloor (x-1)/b \rfloor$ ビットの1を並べて、次に1ビットの0を付ける。次に、整数 $r = (x-1) - qb$ を $\lfloor \log b \rfloor$ ビットあるいは $\lceil \log b \rceil$ ビットで表現したものを付ける。例えば、 $b=3$ の場合、 $x=1\sim10$ に対するGolomb符号は表1の通りである。 b は $b = 0.69N_0 / f_i$ によって計算される。 f_i はあるキーワードを含む文献の数である。キーワードごとに異なる f_i 及び b を用いて符号化するのが一般的である。しかし、ここでは

プログラムを簡潔に記述するために、 $b = 0.69N_0 / \bar{f}_i$ として、全てのキーワードについて同じ b 値を用いて符号化した。ここで、 $\bar{f}_i$ は1キーワードを含む文献数の平均値であり、延べキーワード数(7789)と異なりキーワード数 $M=1360$ の比により与えられる。

4.2.2.c 検索（プログラム3）

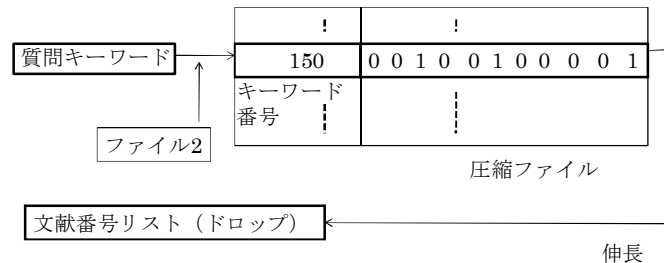


図7 転置ファイルによる検索

ファイル2及びファイル4を用いて検索を行った。キーワード名（質問キーワード）を入力し、そのキーワードに対応するキーワード番号を求めた。これは4.2.1.cのシグネチャファイルの場合と同じく、ファイル2を参照して、2分探索法で求めた。次に、得られたキーワード番号に対応する圧縮ベクトルを伸長し、質問キーワードを含む文献番号を得た。

5. 評価結果

プログラム2、プログラム3で必要なメモリ量、処理時間ともに、シグネチャファイルより転置ファイルが優れている（表2）。とくに、プログラム3の処理時間（検索にかかる時間）は転置ファイルの方が約80倍速い。一方で、プログラム1の処理時間、メモリ量はわずかな差であるが、シグネチャファイルの方が優れている。また、プログラム1～プログラム3の行数はシグネチャファイルの方が短く、簡潔にプログラムを書けることが分かる。とくに、シグネチャファイル作成のプログラム2の行数は圧縮した転置ファイル作成のプログラム2の行数の38%である。

	プログラム1		プログラム2		プログラム3	
	処理時間	メモリ量	処理時間	メモリ量	処理時間	メモリ量
シグネチャファイル	0.118sec	814kbytes	7.27msec	155kbytes	210μsec	277kbytes
転置ファイル	0.120sec	835kbytes	3.68msec	8.5kbytes	2.78μsec	157kbytes

	プログラム1 行数	プログラム2 行数	プログラム3 行数	ファイル2 大きさ	ファイル3 大きさ	ファイル4 大きさ
シグネチャファイル	76	65	61	139kbytes	23kbytes	136kbytes
転置ファイル	106	171	91	139kbytes	21kbytes	21kbytes

表2 アルゴリズムの評価結果

6. まとめ

本論文でフォルスドロップを0にする文献検索用のシグネチャファイルを構成できることを示した。これは理論及びテストデータという2つの面から確認された。また、提案方法によるシグネチャファイルの構成・検索のプログラムは、転置ファイルの構成・検索のプログラムより簡潔に書くことができる。さらに、転置ファイルでは各キーワードを含む文献のリストをつくる必要があるが、シグネチャファイルではこのリストを作る必要がない。その分メモリ量や処理時間を改善できる。

参考文献

- 1) Faloutsos, C.: Access Methods for Text, *Computing Surveys*, Vol.17, No.1, pp.49-74 (1985).

- 2) 二神常爾： 検索効率を考慮した転置ファイルの圧縮， 情報処理学会論文誌， Vol.43， No.5， pp.1432-1445 (2002)．
- 3) Choueka, Y., Fraenkel, A.S., Klein, S.T. and Segal, E.: Improved Hierarchical Bit-vector Compression in Document Retrieval Systems, *Proc. 9th ACM-SIGIR International Conference on Research and Development in Information Retrieval*, pp.88-96 (1986).
- 4) Moffat, A. and Zobel, J.: Parameterized Compression for Sparse Bitmaps, *Proc. 15th ACM-SIGIR International Conference on Research and Development in Information Retrieval*, pp.274-285(1992).
- 5) 有川節夫，篠原武，松本一教，張裕民：重ね合わせ符号を用いた文献検索システムについて—キーワードのための重ね合わせ符号—，データベース・システム，54-2，pp.1-8 (1986)．
- 6) Faloutsos, C. and Christodoulakis, S.: Description and Performance Analysis of Signature File Methods for Office Filing, *ACM Transactions on Office Information Systems*, Vol.5, No.3, pp.237-257 (1987).
- 7) 三木健士，横田治夫：スーパーインボーズコーディングを用いたXML文書キーワード索引手法，情報処理学会研究報告，DBS-140(I)，pp.179-185 (2006)．
- 8) Kautz, W.H. and Singleton R.C.: Nonrandom Binary Superimposed Codes, *IEEE Transaction on Information Theory*, IT-10, pp.363-377 (1964).
- 9) Zobel, J., Moffat, A. and Ramamohanarao, K.: Inverted Files Versus Signature Files for Text Indexing, *ACM Transactions on Database Systems*, Vol.23, No.4, pp.453-490 (1998).

付録

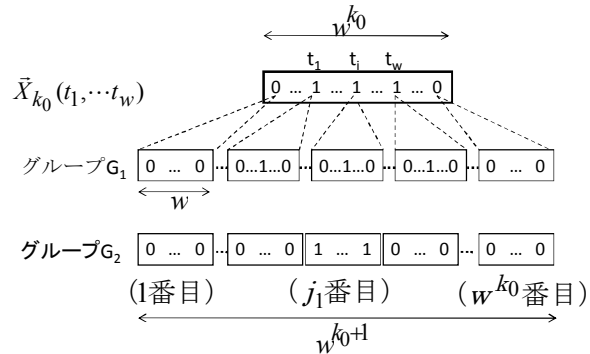


図8 シグネチャファイルの拡張方法

3章で与えた方法によってつくられるシグネチャの集合のオーバーラップが0または1になることを k についての帰納法により示す。 $k=1$ のときには題意が成り立つ。このとき、 $n=w$ であり、つくられるシグネチャは全てのビットが1であるベクトル1つだけ

である。次に、 $k=k_0$ ($k_0 \geq 1$) の場合に題意が成り立つと仮定する。 $k=k_0$ の場合につくられるシグネチャを $\bar{X}_{k_0}(t_1, t_2, \dots, t_w)$ で表す。 $t_1, t_2, \dots, t_w$ を値1をもつビットの位置とする ($1 \leq t_1 < t_2 < \dots < t_w \leq w$)。 $k=k_0+1$ の場合にはシグネチャ $\bar{X}_{k_0+1}(t_1, t_2, \dots, t_w)$ の各ビッ

トが値0をもつとき長さ w ，重み0のシグネチャを割り当てる。値1をもつ場合には長さ w ，重み1のシグネチャを割り当てる。これらをつなげて長さ w^{k_0+1} のシグネチャをつくる (図8)。このシグネチャの集合を G_1 とする。このシグネチャ内での値1をもつビットのうち、 i 番目のものの位置は

$$u_i = (t_i - 1)w + \{(\beta + \gamma i) \bmod w\} + 1, \quad (16)$$

で与えられる。

集合 G_1 の要素はパラメータ $(t_1, t_2, \dots, t_w, \beta, \gamma)$ ($0 \leq \beta, \gamma \leq w-1$) によって互いに区別される。

次に、 $\bar{x}(k_0+1, j_1)$ ($1 \leq j_1 \leq w^{k_0}$) から成る w^{k_0} 個のシグネチャの集合を G_2 とする。提案方法によって与えられる長さ w^{k_0+1} のシグネチャの集合は G_1 と G_2 の和集合 $G(G_1 \cup G_2)$ に等しい。集合 G に属する任意の2つのシグネチャのオーバーラップは0か1になることを示せばよい。

① 集合 G_2 に属する2つの異なるシグネチャ

$\bar{x}(k_0+1, i)$ ， $\bar{x}(k_0+1, j)$ ($1 \leq i, j \leq w^{k_0}, i \neq j$) のオーバーラップは0である。

② 集合 G_1 に属するシグネチャと集合 G_2 に属するシグネチャのオーバーラップは0か1である。

③ 集合 G_1 に属する2つの異なるシグネチャのパラメータを $(t_1, t_2, \dots, t_w, \beta, \gamma)$ と $(t'_1, t'_2, \dots, t'_w, \beta', \gamma')$ とする。

(i) 集合 $\{t_1, t_2, \dots, t_w\}$ と集合 $\{t'_1, t'_2, \dots, t'_w\}$ が異なっている場合 $[\{t_1, t_2, \dots, t_w\} \neq \{t'_1, t'_2, \dots, t'_w\}]$ には、2つの集合の共通の要素が高々1つしかない (帰納法の仮定より)。共通の要素がない場合には2つのシグネチャのオーバーラップは0になる。共通の要素が1つだけある場合には、その要素を $\hat{i}$ とする。 $\hat{i}$ 番目の長さ w のシグネチャの重みはともに1である。従って、2つのシグネチャを $\bar{X}(t_1, t_2, \dots, t_w, \beta, \gamma)$ と $\bar{X}(t'_1, t'_2, \dots, t'_w, \beta', \gamma')$ とすると、これらのオーバーラップは0か1である。

(ii) t_i の集合が同じで $[t_i = t'_i, 1 \leq i \leq w]$ ， $\beta \neq \beta'$ か $\gamma \neq \gamma'$ の少なくともどちらかが成り立つ場合

2つのシグネチャ $\bar{X}(t_1, t_2, \dots, t_w, \beta, \gamma)$ と $\bar{X}(t_1, t_2, \dots, t_w, \beta', \gamma')$ を考える。シグネチャ $\bar{X}(t_1, t_2, \dots, t_w, \beta', \gamma')$ の値1をもつビットのうち i 番目のものの位置 u'_i は、式(16)と同様の式が成り立つ。

$$u'_i = (t_i - 1)w + \{(\beta' + \gamma' i) \bmod w\} + 1, \quad (17)$$

式(16)，(17)より適当な整数 α ， α' を用いて

$$u_i = (t_i - 1)w + \alpha w + \beta + \gamma i + 1, \quad (18)$$

$$u'_i = (t_i - 1)w + \alpha'w + \beta' + \gamma'i + 1, \quad (19)$$

が成り立つ. 式(18)から式(19)を引くと

$$u_i - u'_i = (\alpha - \alpha')w + \beta - \beta' + (\gamma - \gamma')i, \quad (20)$$

を得る.

(a) $\gamma = \gamma'$ かつ $\beta \neq \beta'$ の場合

関係式

$$1 \leq |\beta - \beta'| \leq w - 1, \quad (21)$$

が成り立つ. 従って, 式(20)より

$$\begin{aligned} \theta &\equiv \{(u_i - u'_i) \bmod w\} \\ &= [\{(\alpha - \alpha')w + \beta - \beta'\} \bmod w] > 0, \end{aligned} \quad (22)$$

が成り立つ. 全ての異なる i の値に対して, $u_i \neq u'_i$ が成り立つ. 従って, シグネチャ

$\bar{X}(t_1, t_2, \dots, t_w, \beta, \gamma)$ と $\bar{X}(t_1, t_2, \dots, t_w, \beta', \gamma')$ のオーバーラップは 0 である.

(b) $\gamma \neq \gamma'$ の場合

2つの異なる i の値 i_1, i_2 ($1 \leq i_1, i_2 \leq w, i_1 \neq i_2$) に対して

$$1 \leq |i_1 - i_2|, |\gamma - \gamma'| \leq w - 1, \quad (23)$$

であり, w は素数なので, $(\gamma - \gamma')(i_1 - i_2)$ の値は w で割り切れない. 従って, 次式が成り立つ.

$$[\{(\gamma - \gamma')i_1\} \bmod w] \neq [\{(\gamma - \gamma')i_2\} \bmod w], \quad (24)$$

これは異なる i ($i = 1, 2, \dots, w$) に対して $[\{(\gamma - \gamma')i\} \bmod w]$ の値は全て異なることを意味する. 言い換えると異なる w 個の i ($i = 1, 2, \dots, w$) について, $[\{(\gamma - \gamma')i\} \bmod w]$ は w 個の異なる値 $(0, 1, \dots, w-1)$ をとる.

$$\theta = [\{(\alpha - \alpha')w + \beta - \beta' + (\gamma - \gamma')i\} \bmod w], \quad (25)$$

なので, θ の値も異なる i ($i = 1, 2, \dots, w$) の値に対して, w 個の異なる値 $(0, 1, \dots, w-1)$ をとる. 唯一の i の値に対して $\theta = 0$, つまり $u_i = u'_i$ が成り立つ. 従って, シグネチャ $\bar{X}(t_1, t_2, \dots, t_w, \beta, \gamma)$ とシグネチャ $\bar{X}(t_1, t_2, \dots, t_w, \beta', \gamma')$ のオーバーラップは 1 になる. 以上の①～③の考察から, シグネチャの集合 G に属する任意の 2 つのシグネチャのオーバーラップは 0 か 1 である.