フォルスドロップを0にするシグネチャファイルの構成に関する一考察

二神常爾（聖学院大学非常勤講師）

1.はじめに

シグネチャファイルは、転置ファイルとともに文献検索で古くから研究されてきた．シグネチャファイルを用いる長所として，格納するのに必要なメモリ量を転置ファイルのメモリ量より小さくできる可能性がある．一方で，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は検索条件に適合する適合文献だけでなく，検索条件に適合しない非適合文献（フォルスドロップ）を含む[1]．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。この構成方法はこれまで明らかにされなかった．

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 $j(1\le j\le M)$ に対してワードシグネチャ $\vec{B}_j$ 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 $n(<M)$ 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 $j(1\le j\le M)$ に対して同じ値をとるとする．

ワードシグネチャ1	111000000
ワードシグネチャ2	100100100
ワードシグネチャ3	100010001
文献のシグネチャ	111110101（論理和）

図1 文献のシグネチャの生成の例 $(l_i=3, w=3, n=9)$

文献のシグネチャ $\vec{C}_i\,(1\le i\le N_0)$ 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より得られる（図 1）．すなわち，$i(1\le i\le N_0)$ 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 $\vec{C}_i$ は次の重ね合わせにより与えられる．

$$\vec{C}_i=\vec{B}_{a_{i,1}}\cup\vec{B}_{a_{i,2}}\cup\cdots\cup\vec{B}_{a_{i,l_i}}, \qquad (1)$$

$a_{i,j}$ は i 番目の文献が含む $j(1\le j\le l_i)$ 番目のキーワードである． $\vec{B}_{a_{i,j}}$ 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 $\vec{C}_i\,(1\le i\le N_0)$ 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 $\vec{B}_q$ とする．文献検索は

$$\vec{C}_i\supseteq\vec{B}_q, \qquad (2)$$

を満たす文献 $i(1\le i\le N_0)$ を見出すことである．これは文献のシグネチャ $\vec{C}_i$ が，質問キーワード q のシグネチャ $\vec{B}_q$ 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，

$$q\notin\{a_{i,1},a_{i,2},\cdots,a_{i,l_i}\}, \qquad (3)$$

が成り立つ． $\lambda_{i,j}$ を二つのワードシグネチャ $\vec{B}_i$ と $\vec{B}_j$ 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 $\vec{B}_i$ と $\vec{B}_j$ 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，

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

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[2]．

$$M_{\max}(n,w,\lambda_{\max})\le\binom{n}{\lambda_{\max}+1}\Big/\binom{w}{\lambda_{\max}+1}, \qquad (5)$$

3.提案方法

式(1)より文献のシグネチャ $\vec{C}_i$ 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

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

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

$$M_{\max}(n,w,1) \le \frac{n(n-1)}{w(w-1)}, \qquad (7)$$

となる.

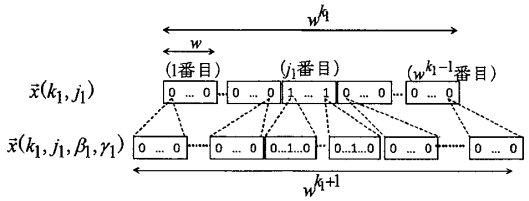

図2 シグネチャファイルの構築方法

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）を連接して長さw^{k_1}，重みwのシグネチャをつくる（$1 \le k_1 \le k$）．連接する際に長さw，重みwのシグネチャを先頭から$j_1\,(1 \le j_1 \le w^{k_1-1})$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$\vec{x}(k_1,j_1)$によって表わす．すると，$\vec{x}(k_1,j_1)$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$u_{0,i}\,(1 \le i \le w)$とすると、次が成り立つ．

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

次に，シグネチャ$\vec{x}(k_1,j_1)$の各ビットに長さwのシグネチャを割り当てて、これを連接して長さw^{k_1+1}のシグネチャをつくる．シグネチャ$\vec{x}(k_1,j_1)$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$\vec{x}(k_1,j_1)$の$u_{0,i}\,(1 \le i \le w)$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$v_{1,i}\,(1 \le i \le w)$を次のように与える．

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

ここで，β_1，γ_1,は$0 \le \beta_1, \gamma_1 \le w-1$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$\vec{x}(k_1,j_1,\beta_1,\gamma_1)$により表わす．シグネチャ$\vec{x}(k_1,j_1,\beta_1,\gamma_1)$内での$i\,(1 \le i \le w)$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，

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

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連接することを繰り返す．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

$$\beta_1, \gamma_1, \cdots, \beta_{k-k_1}, \gamma_{k-k_1} \quad (0 \le \beta_1, \gamma_1, \cdots, \beta_{k-k_1}, \gamma_{k-k_1} \le w-1)$$

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

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

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$\vec{x}(k_1,j_1,\beta_1,\gamma_1,\cdots,\beta_{k-k_1-1},\gamma_{k-k_1-1})$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．

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

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$(\beta_1,\gamma_1,\cdots,\beta_{k-k_1},\gamma_{k-k_1})$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$\vec{x}(k_1,j_1,\beta_1,\gamma_1,\cdots,\beta_{k-k_1},\gamma_{k-k_1})$の個数は$w^{2(k-k_1)}$個存在する．パラメータ$k_1,j_1$は$1 \le k_1 \le k$，$1 \le j_1 \le w^{k_1-1}$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

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

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（証明略）．従って，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.結び

今後、メモリ量と検索効率の視点から、シグネチャファイルの構成についての本提案方法が転置ファイルを用いる場合より改善されるか否かを調べたい。

参考文献

1) 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).

2) Kautz, W.H. and Singletonm R.C.: Nonrandom Binary Superimposed Codes, *IEEE Transaction on Information Theory*, IT-10, pp.363-377 (1964).