計算機ファジー検索方法の一考察

A Study of the Information Retrieval by using the Fuzzy Method

陳漢武＊　近藤高司＊＊　鈴木達夫＊＊
Chen Hanwu＊, Takashi KONDOH＊＊ and Tatsuo SUSUKI＊＊

Abstract　Recently, the information retrieval system for the books library is treated a massive data-base therefore finding out efficiency of requested books data is essential function of it. We studied concern with the evaluation of information retrieval system by using the fuzzy methods. And, we proposed some ideas of the computer algorithm included the fuzzy calculations.

１．はじめに

文献情報の検索とは，文献データベースのデータファイル中に存在するデータを逐次検索し，必要とするデータを得るということである。１９７０年前後，当時中国の国内で利用された，計算機文献情報検索というソフトウェアシステムの多くは，最も一般的であった情報検索方法の一つである―文字匹配（比較）法（文字を見比べる方法）―を採用していた。即ちユーザが希望する検索関鍵字 (keywords，ｷｰﾎﾟｲﾝﾄ的な字）と分類された文献データファイル中のレコードのデータとを逐次比較検討するという方法である。この方法ではユーザの検索語を明確に入力しなければならず，検索処理結果の言葉も明確であることが要求される。これは良いことは良い，しかし，もしユーザが自分が欲しい文献の関鍵字がはっきり分からないし時，あるいはより大きい範囲の文献を欲しっている時，この方法による検索システムでは，ユーザの要求を充分満足させることはできないと考える。

１９７０年代後半から，中国が外国から文献庫検索システムを導入したとともに，ネットワークで国内の主要な図書館や文献情報検索センターと外国の著名な文献庫検索システムと接続することができるようになった。これらを利用して，一般のユーザは国内文献はもとより，外国の文献までを即座に検索できるようになった。多種多様な文献検索方法が提案され実用化されているが，特別なユーザの要求を満足させるために，主題字 (Index Terms) 検索法が脚光をあびるようになってきた。これは検索方法の一つとして多くの人々が注目し興味をそそり始めた，この原因で主題字検索の研究が盛んになり発展してきた。

主題字検索とは，実際最も本質的な基本はやはり文字匹配法の一種類であることである。しかし，この検索方法にファジー測度を融合させるので，検索の目標がより深くはっきりしていない意味を含んでいる。選択された文献が持っている Index Terms と検索者の申し込みの主題字と一致する程度はｄより大きくないべきである。（このｄは，システムに設定されて文献データ集合をファジー的に割る閥値 (keypoint) である）。この検索を次の，ある一種類様な問題に概括することができる．文献庫中のデータ集合がユーザの検索主題字によってある種類な模糊的に割られるということである。

計算機情報検索効果の評価は正確率と完全率を使うことができる。主題字検索の正確率と完全率は多くの要素の影響を受け易い。主要な客観的要素と主観的要素を次に示します。

①　客観的な要素は文献庫の品質である．

品質が良いかどうか：収集された相互に関連する文献データが完全かどうかと，時間を越える長さと，文献に含められた Index Terms と専門技術用語がぴったり一致するかどうかという関係である。

②　客観的な要素は文献キーポイント的な品質

即ち，文献キーポイント品質は専門分野を目指しことが強いかどうか，及びより高い文献の特色をまとめることが高いかどうかなどということの影響を受ける。専門分野を目指したことが強いかどうかということが検索正確率に影響を掛ける，文献の特色をまとめることが高いかどうかということが検索完全率に影響を掛ける。より文献検索完璧一致の域に達するために，先ず文献検索データに付与する Index Terms の品質を高めなければならない。

③　主観的な要素は，検索主題字によって導入されて確定された文献集合が，検索の数学的モデルや検索経路や検索アルゴリズムなどの要素によって支配される

検索ユーザは簡単的に検索目標の正確率と完全率を追求するのではなく，ある検索カテゴリーに対して的確な範疇にあり，より多くの文献情報を入手できるということである。しかし，より多いということはそれ自体にはっきりしていないと言う意味を伴うことがある。

２．文献情報ファジー検索の伝統な方法

文献情報主題詞ファジー検索伝統な方法はおおよそ次の順序に帰結できる。

＊東南大学計算機学科（中国南京市）
＊＊愛知工業大学経営工学科

2・1　検索対象集合と対象特徴性質確定：検索対象はある文献集

$$A=\{A_1, A_2, \ldots\ldots, A_n\}$$

である。またA全体の中の任意の一つA が一つ特徴性質集合

$$P=\{P_1, P_2, \ldots\ldots, P_m\}$$

を備えている。P定義について二つの方法があり，即ち，定量化と定性化と呼ばれる。二つ方法がそれぞれ関数μとχで定義されている。

①定量化：

$$A_i \mid\rightarrow \mu_{pj}(A_i)\ ;\ \mu_{pj}(A_i)\in[0, 1]$$

$$(i=1, 2, \ldots, n;\ j=1, 2, \ldots, m)$$

μはPに対応するAの従属関数である。

②定性化：

$$A_i \mid\rightarrow \chi_{pj}(A_i)\ ;\ \chi_{pj}(A_i)\in(0, 1)$$

$$(i=1, 2, \ldots, n;\ j=1, 2, \ldots, m)$$

χはPに対応するAの特徴関数である。

2・2　検索対象と特徴性質とのファジー関係生成：
われわれはただ一つ

$$\text{Matrix } R_{nxm}=(\gamma_{ij})_{nxm}$$

でAとPとのファジー関係を表す可能性がある，このγ　が

$P_j \in P$に関する$A_i \in A$ (i=1, 2, ……, n; j=1, 2, ……, m)

の従属数率或いは特徴値。

2・3　検索キーポイント詞集合確定：検索キーポイント集合を

$$B=\bigvee_{k=1}^{m} B_k \quad \text{で表す。}$$

$$B_k = \begin{bmatrix} \nu_{1k} \\ \nu_{2k} \\ \vdots \\ \vdots \\ \nu \\ \nu_{mk} \end{bmatrix} \qquad (k=1, 2, \ldots\ldots, l)$$

その中ν_{jk}が，B中第k番分詞B_kて，P_jに関連するA中任意の要素に対応する特徴な要求或いは従属数率であるというものを表す。

2・4　検索キーポイント詞 Matrix Bに対応するA集合の解を求めて得る Matrix を組み立てる。求める解 Matrix $T_{nxl}=(\sigma_{ik})$，その中σ_{ik}がγ_{ij}とν_{jk}の線形関数です，

$$(1\leqq i\leqq n)\ \ (1\leqq j\leqq m)\ \ (1\leqq k\leqq l)。$$

たとえ，われわれは

$$\sigma_{ik}=f(\gamma_{ij},\ \nu_{jk}) \underset{=}{\triangle} \sum_{j=1}^{m} \min(\gamma_{ij},\ \nu_{jk}) \Big/ \sum_{j=1}^{m} \max(\gamma_{ij},\ \nu_{jk})$$

を命ずることができます，こうしたら，さらに検索キーポイント詞Bに対応する集合A全体の中の任意のA_i (i=1, 2, ……, n) の従属度を打診すると，打診した結果が数量化 f になって明らかに，　$f\in[0,\ 1]$ となる。

2・5　命題Bに対応するAの従属度が高まって区別できるようになるため，有限範囲内で拡大したσ_{ik}，ξ_{ik}になる：

$$\sigma_{ik}\rightarrow\xi_{ik}$$

検索キーポイント詞Bに対するA全体の中の任意の一つ要素A_iの可能な参考度を弁別するために，一度Aの内容を分けることが重要で，分けられる集合Aが四つの子集合

$A_{\lambda 1}$，$A_{\lambda 2}$，$A_{\lambda 3}$，$A_{\lambda 4}$　のようになり，

$$A_{\lambda i} \underset{=}{\triangle} (A)_{\lambda i} \underset{=}{\triangle} \{A_j \mid \mu_{\lambda i}(A_j)\geqq\lambda_i\} \quad (i=1, 2, 3, 4)$$

$(A)_{\lambda i}$ はλについて一つAの分割集合です，

$$\lambda_i = f(\sigma_{ik},\ C,\ S)$$

しかし（その中，Cが拡大係数で，Sはある常数）(0，1) で更に1に近づく値を選択できる，そして

$$0<\lambda_4<\lambda_3<\lambda_2<\lambda_1<C$$

ということになる。このλ i(i=1, 2, 3, 4) はAの分割集合の各閾値と呼ばれる。例えば，

$$M\underset{=}{\triangle}\max_{\substack{1\leqq i\leqq n\\ 1\leqq k\leqq l}}\sigma_{ik} \quad \text{と} \quad m\underset{=}{\triangle}\min_{\substack{1\leqq i\leqq n\\ 1\leqq k\leqq l}}\sigma_{ik}$$

と仮定すると，さらに

$$K\triangle m+C*(M-m)$$
$$=$$
$$k\triangle m+C*(M-m)*S$$
$$=$$
$$\xi_{ik}=\frac{\sigma_{ik}-m}{M-m}*C$$

を命ずる，その上，

$\lambda_1 = C$，$\lambda_2 = 1$，$\lambda_3 = S$，$\lambda_4 = 0.5$

になったら，任意一つξ　次の通りになるに違いない：

Ⅰ．$1 \leqq \xi_{ik} \leqq C$　優解
Ⅱ．$1 \leqq \xi_{ik} < 1$　拡解
Ⅲ．$1 \leqq \xi_{ik} < S$　備解
Ⅳ．$1 \leqq \xi_{ik} < 0.5$ 不満足

Ⅰ～Ⅳを踏まえて，

matrix $T'_{n\times 1} = (\xi_{ik})$

の処理を行われると，条件が満足するξをτに保存する。

2・6　検索キーポイント詞Bに対応する集合Aの予測解

matrix $T''_{n\times 1} = (\tau_{ik})$，

その中

$\tau_{ik} = (A_i, \xi_{ik})$，

$$a_i = \sum_{i=1}^{m} \tau_{ik}$$

を求めるによって，解ベクトル

$((A_1, a_1), (A_2, a_2), \ldots, (A_n, a_n))$

を得る。

2・7　もう一度数値によってmatrix Aを配列して，新しいA′になる，即ち

$A \mid\rightarrow A' ((A_1', a_1'), (A_2', a_2'), \ldots, (A_n', a_n'))$

その中

$a_1' \geqq a_2' \geqq \ldots \geqq a_n'$

で有る。検索管理者はa′の同じでない従属区域値を踏まえて，ユーザに優解集とか，拡解集とかと備解を提供する。

3．一種類主題詞アルゴリズム：

3・1　約定：

①．文献主題詞を文献データベースのレコートに書き込むとき，分詞が順列の順序によって配列する，そして一定な引き寄せる網羅度を満足させる。

②．ユーザが検索キーポイントを入力するとき，次のⅠとⅡを守るはずである。
Ⅰ．規範標準で主題詞を入力する。
Ⅱ．規範標準検索表示できないとき，ユーザが，検索管理者正しく誘導によって問題が規範標準表示になったら，システムに入力する。

③．主題詞入力格式は

$P = (m, \rho_1, \rho_2, \ldots, \rho_m)$，

その中，mが文献資料検索主題ρ，の数です。

④．検索システムに提出する検索キーポイントはこの様

$V = (l, \upsilon_1, \upsilon_2, \ldots, \upsilon_l)$

になります，その中，lが入力されたυ，の数です。

⑤．制御解の分類閥値を選択する

$\lambda_1 = (\sqrt{5}+1)/2 \triangle C$；
$=$
$\lambda_2 = C - S/2$；
$\lambda_3 = S$；
$\lambda_4 = 0.45C$；

このSが特定な0.5より大きいパーセント数て，優選閥値増量パーセント数と呼ばれる。

⑥．拡大係数matrix $W = (\omega_{ij})$を構造する，Wはファジー検索中に，検索キーポイントと文献主題詞との互いに関連することや，キーポイントと主題詞との差額と比率値などを反応できます。そして，手で主題詞検索する体験を踏まえてω_{ij}を選別できます。ω_{ij}は第ｉ番目検索キーポイント集合と文献主題詞の第ｊ番目分詞との匹配の成功率数値の拡大係数である。

$\omega_{ij} \triangle g(l, m, |i-j|, i/j)$
$=$

⑦．設置ベクトルY，

$Y \triangle (\zeta_1, \zeta_2, \ldots, \zeta_k)$
$=$
$k = \min(l, m)$

Ｙ［ｉ］には，第ｉ番目検索キーポイントと文献主題分詞の匹配成功の分詞序次号をおいている。匹配が失敗する時，Ｙ［ｉ］＝100＋ｉ

⑧．任意の文献Ａ，と検索Ｖとのファジー匹配選択結果が数値化Ｔ　になります。

3・2　アルゴリズム：

```
read   V;    /* V△(l, υ1, υ2, , . . . , υl, )
                  は検索キーポイントです)*/
open   A°;   /* A°は文献データベース分
                類庫に関連する主題詞目録で
                す)*/
WHILE ¬A°  DO
  BEGIN
  read  P;
    /*  P△(m, ρ1, ρ2. . . , ρm, )は次
```

```
    の検索したい文献主題詞 ) です */
z := max(m, l);
define Y[z];        /* 定義線形ベクトルY */
  FOR  I := 1 TO  l
   FOR  J := 1 TO m
        検索キーポイント分詞と主題分詞定性
        匹配する過程 ;
        /* 結果が Y[i] に書き込む */

   IF 一度匹配成功
        THEN  Y[I] := J;
        ELSE  Y[I] := 100 + I;
  END (FOR J);
  Y[I] := 1 - | Y[I] - I | / 100;
    Case
      m ≧ l:
      BEGIN

            x := 1 / l * Σ(i=1 to l) Y[i];

      Case

          m=l. and. 0. 91≦ x ≦1:
                        T := x * C;
          m>l. and. 0. 91≦ l ≦1:
                        T := x * C * S;
          ELSE          T := x * C * ω   ;
          END (Case);
        END;
      m < l:
       BEGIN

      x := 1 / ( m +( l - m ) * H ) * Σ(i=1 to m) Y[i];

          /* H は 1 より大きい常数です */
      IF ( 0. 91*m/(m+(l+m)*H) ≦ x ≦
                                m/(m+(l-m)*H) )
            THEN  T := x * C;
            ELSE  T := x * C * ω   ;
        END;
    END (Case);
現在文献Aと検索結果Tを線形数組Aに書き込む
/* A△ (U1, U2, ..., Un),  Ui△ (Ai,  TAi )*/
     =                          =
   END ( FOR I );
 END ( WHILE );
```

Aの中の要素のU_iを値の大小順序によって配列する，結果によって検索結果をユーザに渡す。

4．まとめ

下記の表1，$W_{10\times10}$は拡大係数 matrix です，しかしこれがただ文献主題標引数10個と検索キーポイントも10個の範囲で考慮しているものである。各ω_{ij} はl, l-「x」,「x」/l関数であるべきで，即ちω_{ij}= f(l ,「x」)。この matrix $W_{10\times10}$ は，この原則を踏まえて手検システムから取った数値を処理して取ったものである。

（受理　平成8年3月19日）

表1．マトリックスW（１０×１０）の計算例

1	1.00	1.75								
2	1.60	1.00	1.37	1.71						
3		1.22	1.00	1.24	1.48	1.65				
4		1.56	1.09	1.00	1.20	1.37	1.48	1.61		
5			1.33	1.05	1.00	1.17	1.30	1.40	1.48	1.55
6			1.50	1.22	1.02	1.00	1.15	1.26	1.34	1.41
7				1.33	1.15	1.00	1.00	1.15	1.23	1.30
8				1.46	1.25	1.11	1.00	1.00	1.15	1.21
9					1.33	1.19	1.08	1.00	1.00	1.15
10					1.40	1.26	1.15	1.06	1.00	1.00
	1	2	3	4	5	6	7	8	9	10