論理積項を利用した高速文献検索法とそのファジィ検索への応用†

3G−8

鈴木 知明*　向殿 政男**

明治大学理工学部情報科学科‡‡

1 はじめに

現在、大量の文献がデータベース化され、利用されている．しかし従来の文献に付加されたキーワードと検索要求とのAND/ORによる検索だけでは希望する文献をうまく捜し出せない場合があった．そこで文献と検索要求との包含度を用いて，より自然な検索を行なう場合が考えられるがこれは検索に時間がかかる．そこで分離加法形式（論理式を加法形式で表現し，各積項が分離的である加法形式表現）を用いたクラスタリングを行ない，検索の高速化を試みる手法を提案する．さらにその場合において，要求に曖昧さを持たせたファジィ検索への拡張も考察する．

2 ファジィ文献検索

文献データベースにおいて、文献はその文献の内容を表現するキーワードをつけて蓄えられる．検索は検索要求に含まれるキーワード（以下要求キーワード）と文献に含まれるキーワード（以下文献キーワード）を用いて行なう．従来の方式では，要求キーワードの一つが文献キーワードに含まれるか，否か，のみで検索を行なっていた．しかしこの方式では，要求キーワードが複数個の場合，文献キーワードが要求キーワードを一つだけ含んでいる文献と，文献キーワードがすべての要求キーワードを含んでいる文献とが同等に扱われてしまう．

そこで，この不自然さを解消するために，次のようなファジィ文献検索が提案されている．[1]ここで以下の説明を行なうために，次の集合を定義する．

定義 2.1

文献の集合 D = $\{d_1, \cdots, d_l\}$
様々な検索要求の集合 Q = $\{q_1, \cdots, q_m\}$
キーワードの集合 K = $\{k_1, \cdots, k_s\}$ ■

さらに h_d をD×K上，及び h_q をQ×K上のファジィ関係で，$h_d(d, k_j)$ は文献 d がキーワード k_j に相当する度合いを表し，$h_q(q, k_j)$ は検索要求 q がキーワード k_j に相当する度合いを表すとする．この関係 (h_d) は，D上で定義され，Kのファジィ集合に値をとる関数 $h_d(d), d \in$ D，とみなすこともできる．この関数とファジィ関係との関連は

$$h_d(d) = \sum h_d(d, k_j)/k_j \tag{2.1}$$

となる．（関係 h_q についても同様）

ここで包含度 t を次のように定義する．

定義 2.2 (包含度 t)

$$t(q,d) = \frac{|h_d(d) \cap h_q(q)|}{|h_q(q)|} = \frac{\sum_{i=1}^{s} min(h_d(d,k_i), h_q(q,k_i))}{\sum_{i=1}^{s} h_q(q,k_i)} \tag{2.2}$$

但し ここでは $\sum$ は算術和
$s = |K|$(すなわちキーワードの総数) ■

但し 本稿においては，関数 $h_d(d, k_i)$ の値は閉区間 [0,1] ではなく，0,1 の2値のみをとるとしている．これは文献のクラスタリングに論理式を用いるためである．

3 分離加法形式

定義 3.1 (分離加法形式) [2]

論理関数 f は，通常，加法形式と呼ばれる論理式で表現される．

$$f = \alpha_1 \vee \cdots \vee \alpha_m \tag{3.1}$$

ここで，$\alpha_i (i = 1, \cdots, m)$ は積項である．式 (3.1) において，各積項が分離的である時，すなわち，$\alpha_i \cdot \alpha_j = 0$ がすべての $i, j (i \neq j)$ について成立している時，式 (3.1) は，分離加法形式による表現であるという． ■

4 分離加法形式にもとづいた検索高速化

4.1 文献集合 D の分離加法形式表現

定義 4.1

$$f^m(d) \triangleq x'_1 \cdots x'_s \tag{4.1}$$

但し $s = |K|$ (すなわちキーワードの総数)
x_i は論理変数．
$d \in D$

論理変数 x'_i の値は

$$x'_i \triangleq x_i \ (h_d(d, k_i) = 1\text{の時}) \triangleq \bar{x}_i \ (h_d(d, k_i) = 0\text{の時})$$

とする． ■

定義4.1の変数 x'_i は文献 d にキーワード k_i が含まれているか否かを表している．そして $f^m(d)$ は文献 d をn個のリテラルの論理積で表現される最小項に対応させることを意味している．これを文献集合 D のすべての要素に対して行なうことにより，

$$f = f^m(d_1) \vee f^m(d_2) \vee \cdots \vee f^m(d_l) \tag{4.2}$$

但し $l = |D|$

まず，文献集合 D の論理和標準形（最小項の和）f を得る．次に f の分離加法形式 f^d を求める．

4.2 分離加法形式を用いた検索手法

本稿で提案する検索手法は文献一つ一つを検索対象として検索する代りに分離加法形式 f^d の積項 $f_i^d (i$ は f_d の i 番目の積項を意味する.) を用いて検索を行なう．ここで，定義 2.2で定義した包含度 t を次のように再定義する．

定義 4.2

文献集合 D を分離加法形式

$$f^d = f_1^d \vee f_2^d \vee \cdots \vee f_t^d$$

†A fast document retrieval method based on product terms and it's application to fuzzy retrieval

*Chiaki Suzuki

**Masao Mukaidono

‡‡Department of Computer Science,Meiji University

で表現した時, 分離加法形式の i 番目の積項 f_i^d と要求 $q(q \in Q)$ との包含度は

$$t^*(q, f_i^d) = \frac{\sum_{j=1}^{s} min(c_{ij}, h_q(q, k_j))}{\sum_{j=1}^{s} h_q(q, k_j)} \qquad (4.3)$$

但し　$s = |K|$(すなわちキーワードの総数)

$$c_{ij} = \begin{cases} 0 & \overline{x_j} が f_i^d に存在する時 (j = 1 \cdots s) \\ 1 & \begin{cases} x_j が f_i^d に存在する時 (j = 1 \cdots s) \\ x_j 及び \overline{x_j} が f_i^d に存在しない時 (ドント \\ \quad ケアの時)(j = 1 \cdots s) \end{cases} \end{cases}$$

■

ドントケアを 1 として扱うと, 包含度 t^* は f_i^d に含まれる文献の包含度 t のもっとも値の大きなものと等しくなる. これは, 検索要求に「包含度 0.5 以上」(以下要求レベル) のような指定があった場合に, 有効になる. つまり積項 f_i^d の包含度 t^* が要求レベル未満の場合, その項に含まれているすべての文献はまとめて検索対象から外すことができ, その結果, 検索が高速化される. つまり, 積項 f_i^d の包含度 t^* が要求レベルより高い時にのみ, f_i^d に含まれる各文献の包含度 t を計算する.

例 4.1

表 4.1: 文献データ

文献	キーワード
document1	k_1
document2	k_1, k_2, k_4
document3	k_1, k_2
document4	k_1, k_2, k_3, k_4
document5	k_1, k_2, k_3
document6	k_1, k_4

表 4.1の文献データベースを仮定する. まず この文献データベースを分離加法形式にする.

$$\begin{aligned} f_m(document1) &= x_1\bar{x}_2\bar{x}_3\bar{x}_4 \\ f_m(document2) &= x_1x_2\bar{x}_3x_4 \\ f_m(document3) &= x_1x_2\bar{x}_3\bar{x}_4 \\ f_m(document4) &= x_1x_2x_3x_4 \\ f_m(document5) &= x_1x_2x_3\bar{x}_4 \\ f_m(document6) &= x_1\bar{x}_2\bar{x}_3x_4 \end{aligned}$$

ゆえに

$$f_d = x_1x_2 \vee x_1\bar{x}_2\bar{x}_3 \qquad (4.4)$$

この文献データベースに対し, 次の要求を与える.(ここで 要求キーワードの後ろの数値は その要求キーワードの度合とする.)

要求 q=k_1 0.8, k_2 0.6
要求レベルは 0.8

式 (4.4) の分離加法表現に対し, 定義 4.2の包含度 t^* を用いて, 検索を行なう.
$x_1x_2(= f_1^d)$ と要求 q の包含度 t^*

$$\begin{aligned} t^*(q, f_1^d) &= \frac{\sum_{j=1}^{4} min(c_{1j}, h_q(q, k_j))}{\sum_{j=1}^{4} h_q(q, k_j)} \\ &= \frac{0.8 + 0.6 + 0 + 0}{0.8 + 0.6 + 0 + 0} \\ &= 1.0 \end{aligned} \qquad (4.5)$$

同様に $x_1\bar{x}_2\bar{x}_3(= f_2^d)$ と要求 q の包含度 t^*

$$t^*(q, f_2^d) = \frac{0.8}{1.4} = 0.57 \qquad (4.6)$$

これらの包含度 t^* は, 前述したように, 項 $x_1x_2, x_1\bar{x}_2\bar{x}_3$ に含まれる文献の中で要求との一致度がもっとも高い文献の包含度 t と等しい. 今 x_1x_2 の包含度 t^* は要求レベル以上なので, この積項に含まれる文献の包含度 t をすべて計算, 検索の結果に加える.

$x_1\bar{x}_2\bar{x}_3$ の包含度 t^* は要求レベル以下なので何も操作を行なわない.

5 実験結果

実験は架空の文献データベースを用いて行なった. 各文献は 15 個のキーワードをランダムに組み合わせた文献キーワードを持たせている. 但しこのデータベース内においては, まったく同一の組合せのキーワードを持つ文献は存在しないとしている.

この文献データベースに対して, 1000 個の要求をランダムに発生させ検索を行なった. この要求は, まったく同一の組合せのキーワードを持つ要求も存在する. これは現実の検索においても十分あり得ることであり特に重複をさける必要はない. また要求キーワードの持つ度合はすべて 1.0 としている. 実験方法は文献一つづつに対して包含度 t をとる方式と, 分離加法形式を用いた検索による検索時間を比較する. 表 5.1の横軸は要求レベルであり, 縦軸は文献数である. 表の中の値は式 (5.1) による.

表 5.1: 実験結果 : 基本的な検索

文献数:積項数 *1	0.0	0.2	0.4	0.6	0.8	1.0
1638 : 1280	1.30	1.26	1.18	0.97	0.85	0.81
3276 : 2094	1.37	1.32	1.20	0.91	0.73	0.68
6553 : 3606	1.40	1.34	1.22	0.88	0.67	0.61
9830 : 5018	1.37	1.31	1.19	0.85	0.64	0.57

*1: 分離加法形式の積項数

$$\text{表 5.1中の値} = \frac{\text{分離加法形式を用いた検索時間}}{\text{文献を一つずつ調べた検索時間}} \qquad (5.1)$$

上の実験結果より, 要求レベルが高いと検索が高速化され, 本手法が有効であることがわかる.

6 まとめ

本稿では文献検索を行なう際, 分離加法形式を用いてクラスタリングすることにより 高速な検索法が可能であることを示した. 更に, この方法はファジィ文献検索にも応用できることを示した.

参考文献

[1] 宮本 定明, 三宅 郷久: “ファジィ情報検索について”, 日本ファジィ学会誌 Vol.3 No.1(1991)

[2] 高橋 直哉, 向殿 政男: “論理関数の分離加法形式による表現とその応用”, 電子通信学会論文誌 Vol.J69-D No.8(1986/8)