数理解析研究所講究録
1041 巻 1998 年 175-182

部分語相関ルール発見のための高速アルゴリズム

渡木 厚, 有村 博紀, 藤野 亮一, 有川 節夫
九州大学 大学院システム情報科学研究科, 情報理学専攻
〒816 春日市春日公園 6-1

概要: 与えられた文字列集合において, 与えられた値以上の頻度で出現し, 確信度を最大化するような最適パタンを見つける問題を考察する. 二語相関パタンとよぶ単純なパタンを仮説としたとき, 接尾辞木技法と直交領域木技法を用いて, 最適パタンを $O(mn^2 \log^2 n)$ 時間および $O(kmn \log n)$ 領域で計算するアルゴリズムを与える. ここに, m と n は, それぞれ入力例の個数と総サイズであり, k は距離パラメータと呼ばれる小さな定数である.

1 はじめに

データマイニング (データ発掘, data mining) は, データベースに蓄積された一見無意味に見える大量のデータから, 自明でない規則性やパターンを半自動的にとりだす方法についての科学的研究である. データ発掘は, 1990 年代初めからその研究が顕在化し, 現在, ビジネス分野や科学分野で盛んに適用されている.

従来のデータ発掘研究は, 明示的な構造をもつ関係データベースを対象としている. この一方で, 近年, Web ページや SGML 文書, ファイルシステムに大量に蓄積された電子メールや電子化文書などのテキストデータベースの利用がここ数年で急速に広まりつつある. しかし, これらのテキストデータベース [1, 7, 10] に関しては,

- 関係データベースのような明示的な構造を持たない,
- 数ギガバイトから数テラバイトにおよぶ膨大なデータの集積である

などの理由から, 従来の方法をそのまま適用することができない. そのため, 文字列データベースを対象とした新しいデータ発掘手法が必要である.

本稿では, 与えられたテキスト集合を特徴づけるパタンを高速に見つける問題を考察する. 具体的には, 正例と負例の集合が与えられたとき, 正例中に与えられた値以上の頻度で出現し, 確信度を最大化するような (負例に関する誤差を最小化するような) 最適パタンを見つける見つけるという最大確信度パタン発見問題を考察する.

この問題は, 仮説として 2 語相関パタンと呼ぶ単純なパタンを考えた場合, 自明なアルゴリズムによって $O(n^5)$ 時間で解くことができる. しかし, 実際の応用を考えたとき, このアルゴリズムはひどく効率が悪い. そこで, テキスト索引のための接尾辞木技法と計算幾何学における直交領域質問技法を組み合わせて, より高速なアルゴリズムを与える.

A fast algorithm for finding optimum word-association patterns
Hiroki Arimura, Atsushi Wataki, Ryoichi Fujino, Setsuo Arikawa, Department of Informatics, Kyushu University, Kasuga Koen 6-1, Kasuga, 816 Japan,
TEL: 092-583-7635, FAX: 092-583-7635, {wataki,arim,fujino,arikawa}@i.kyushu-u.ac.jp

2 準備

2.1 文字列

記号のアルファベットを Σ に対して，Σ の要素の有限列 $w = a_1 a_2 \cdots a_n$ を文字列とよび，Σ 上の文字列全体を Σ^* で表す．文字列 w の長さ n を $|w|$ で表す．文字列 w と正整数 i, j $(i \leq j)$ に対して，$w[i]$ で i 番目の文字 a_i を表し，$w[i..j]$ で**部分語** (subword) $a_i a_{i+1} \cdots a_j$ を表す．正整数 $1 \leq i \leq n$ に対して，部分語 $w[j..n]$ を w の i 番目の**接尾語** (suffix) とよぶ．

文字列 u, w に対して，u が w の部分文字列であるとき，つまり，ある正整数 i に対して $u = w[i\,..\,i+|u|-1]$ となるとき，u は w に**出現する**という．このとき，i を u の**出現位置** (occurrence) という．

2.2 二語相関パタン

二語相関パタン (two-word association pattern), または**近接パタン** (proximity pattern) とは，2 つの文字列 α，$\beta \in \Sigma^*$ と距離パラメータとよばれる非負整数 $k \geq 0$ の三つ組 $\langle \alpha, k, \beta \rangle$ である．以下は二語相関パタンの例である．

⟨TATA, 30, AGGAGGT⟩.

⟨knowledge, 50, databases⟩.

直感的には，パタン ⟨TATA, 30, AGGAGGT⟩ は，「テキスト中の "TATA" の出現から 30 文字以内に "AGGAGGT" が出現する」ことを意味する．形式的には，二語相関パタン $P = \langle \alpha, k, \beta \rangle$ がテキスト S に**照合する**とは，テキスト S において，α のある出現位置 i と β のある出現位置 j が存在して，$0 \leq i - j \leq k$ となることと定義する．言いかえると，P は，部分語 α の出現位置から k 文字以内の距離に部分語 β が出現することを表わしている．出現位置の組 $\langle i, j \rangle$ を，パタン P の出現位置という．

2.3 最適パタン発見問題

テキストとは Σ 上の文字列をいう．テキストデータベースは，テキストの集合 $T = \{t_1, t_2, \cdots, t_m\} \subseteq \Sigma^*$ である．**サンプル** (sample) または**分類例集合**とは，T のテキスト x に，それが目標の集合に含まれるか否か (正例か負例か) を表す 2 値のラベルをつけた組 $\langle x, b \rangle \in T \times \{0, 1\}$ の集合 S である．各組 $\langle x, b \rangle$ を**分類例** (labeled example) という．S を分類例集合としたとき，正例の集合を $S_+ = \{x | \langle x, 1 \rangle \in S\}$，負例の集合を $S_- = \{x |\, \langle x, 0 \rangle \in S\}$ と定義する．S のサイズを $size(S) = \sum_{<x,d> \in S} |x|$ と定義する．

以下では，テキストデータマイニングを，分類例集合が与えられたとき，あらかじめ定められた仮説空間から，ラベル付けされたテキストを最もうまく分類する最適なパタンを見つける問題として定義する．まず，**仮説**となるパタンの族を $\mathcal{P}$ とおく．$P \in \mathcal{P}$ がテキスト $x \in \Sigma^*$ に照合するならば，$P(x) = 1$ と定義し，照合しなければ $P(x) = 0$ と定義する．

つぎにテキストデータマイニングを，福田等 [6] の最適規則発見の枠組を用いて定式化する．パタンの評価基準として，支持率 (support) と確信度 (confidence) を導入する．分類例集合 S とパタン P に対して，パタンに照合する正負例数を $match(P) = \sum_{\langle x,b \rangle \in S} P(x)$ とし，パタンに照合する正例数を $hit(P) = \sum_{x \in S_+} P(x)$ とする．このとき，P の支持度 $supp(P)$ を，正例全体に対する P が照合する正例の割合とし，確信度 $conf(P)$ を，P が照合する例全体に対する P が照合する正例の割合と定義する:

$$supp(P) = hit(P)/|S_+|, \qquad conf(P) = hit(P)/match(P)$$

ここでの問題は，与えられた値以上の支持度をもち，確信度を最大化するパタンを見つけることである．確信度を最大化するパタンを，以降では単に**最適パタン** (optimum pattern) とよぶ:

最大確信度パタン発見問題:

入力： 分類例集合 $S \subseteq \Sigma^* \times \{0,1\}$ と正数 $\sigma > 0$.

問題： S において支持度 $supp(P) > \sigma$ をみたすパタン $P \in \mathcal{P}$ で，$conf(P)$ を最大にするパタンものを見つけよ.

2.4 接尾語木

テキストを $A = a_1 a_2 \cdots a_{n-1}\$$ とし，$n = |A|$ とおく．ここに，$\$ \notin \Sigma$ は，任意の記号 $a \in \Sigma$ に対して $a \neq \$$ と定義される特別な区切り記号である．正整数 $1 \leq p \leq n$ に対して，A の位置 p からはじまる**接尾語** (suffix) を，

$$A_p = a_p \cdots a_{n-1}\$$$

とおく．**接尾語木** (suffix tree) は，与えられたテキストの接尾語すべてを $O(n)$ 領域で保持するためのデータ構造である [2, 8]．テキスト集合 A の**接尾語木** $Tree_A$ とは，つぎの性質をみたす有向順序木 $Tree_A$ である：

(1) 各辺は，A の部分語 α をラベルとしてもつ．ラベル α は，A 中のその出現における開始位置と終了位置の組で符号化される．

(2) 各節点 v は，根からその節点までの経路上のラベルを連結して得られる語 $W(v)$ として，A の部分語を表すものとする．さらに，$Tree_A$ はちょうど n 個の葉をもち，それぞれの葉は A の空でない接尾語を表している．これら葉でが表す接尾語は，左から右へ $\Sigma \cup \{\$\}$ 上の辞書式順序に並んでいる．

(3) 任意の節点から出ていく 2 本の辺のラベルは，異なる先頭文字をもつ.

接尾語木 $Tree_A$ は，n 個の葉をもち，すべての内部節点が 2 つ以上の子をもつ．このことから，$Tree_A$ はたかだか $2n-1$ 個の節点しかもたないことがわかる．よって，$Tree_A$ は線形領域しか使わない．さらに，$Tree_A$ は線形時間で計算する巧妙なアルゴリズムが知られている [2, 8].

2.5 直交領域探索

2 次元離散平面上の点集合を $X \subseteq [1..n] \times [1..n]$ とする．直交領域探索 (orthogonal region query) とは，入力として与えられた長方形 $[x_1..x_2] \times [y_1..y_2]$ に含まれるすべての点の集合を求める問題である.

この問題は，領域木 (range tree) と呼ばれるデータ構造を用いて点集合 X を時間 $O(n)$ で前処理しておくことで，$O(\log^2 n)$ 時間と $O(n \log n)$ 領域で解くことができる [11]．ここでは，平面上の各点が正整数 $l \in [1..m]$ でラベル付けされており，長方形 $[x_1..x_2] \times [y_1..y_2]$ に含まれるすべての点のラベルの集合を求めるよう拡張された場合を考える．このときも，以下の時間領域で直交領域質問が可能であることが容易にわかる.

補題 1 上記の仮定の下で，与えられた長方形に含まれる点のラベルの集合を求める問題は，前処理時間 $O(n)$ を用いて，$O(m \log^2 n)$ 時間と $O(mn \log n)$ 領域で解ける．ここに，m は異なるラベル数の最大値である.

3 基本的アルゴリズム

本節では，最適確信度パタン発見問題を $O(mn^2 \log^2 n)$ 時間と $O(kmn \log n)$ 領域で解くアルゴリズムを与える.

分類例集合を $S \subseteq D \times \{0,1\}$ とする．ここで，任意のパタン $P, Q \in \mathcal{P}$ に対して，もし $P \equiv_S Q$ ならば $supp(P) = supp(Q)$ かつ $conf(P) = conf(Q)$ をみたすような同値関係 $\equiv_S$ が存在すると仮定しよう．このとき，最適パタンを見つけるには，$\mathcal{P}$ のパタンのうち $\equiv_S$ に関する代表元だけを調べ

ればいい．したがって，この条件をみたす同値関係 $\equiv_S$ が存在する場合，つぎのようなアルゴリズムで最適パタン発見問題を解くことができる．

Algorithm Find_Optimized_Patterns

Input: 分類例集合 S および，最小支持度 $\sigma \geq 0$, 距離パラメータ $k \geq 0$.

Output: $\equiv_S$ に関するすべての最適確信度パタンの代表元.

- 各代表元パタン P について以下を実行する：
  - 支持度 $supp(P)$ と確信度 $conf(P)$ を計算する.
  - もし $supp(P) \geq \sigma$ ならば P を優先度付待ち行列 Q に挿入する.

    P は，$conf(P)$ に関してソートされる.
- Q 中のパタン P のうち $conf(P)$ を最大化するものを出力する.

高速なアルゴリズムを実現するための鍵は，上で述べた同値関係 $\equiv_S$ をうまく与えることと，この同値関係に関する代表元パタンについて，支持度と確信度を高速に計算することである．この2つの点の詳細を，以下の節で述べる．

3.1 入力テキスト

入力として与えられた分類例集合を $S = \{\langle t_1, d_1\rangle, \ldots, \langle t_m, d_m\rangle\} \subseteq \Sigma^* \times \{0,1\}$ とする $(m \geq 1)$. 分類例集合 S が与えられると，アルゴリズムは S 中のすべての例を順に結合して，1本の長いテキスト

$$A = t_1\$ \cdots \$t_m\$$$

に変換する．ここに，$\$ \notin \Sigma$ は，任意の $a \in \Sigma$ に対し $\$ \neq a$ であり，$\$ \neq \$$ と定義される特別な区切り記号である．A の長さを $n = |A|$ とおく．A 中の位置 $1 \leq p \leq n$ に対して，$\chi(p)$ を位置 p を含む，すなわち，p が区間 t_i 中の位置となるような，例の番号 i $(1 \leq i \leq m)$ と定義する．

3.2 代表元パタン

文字列 α に対して，$Occ_A(\alpha)$ を，テキスト A 中の α の出現位置すべてのなす集合と定義する．ここで，区切り記号 $\$$ は Σ のいかなる記号とも同一でないので，α の任意の出現はある $1 \leq i \leq m$ に対して区間 t_i に完全に含まれ，区切り記号をまたがることはない．まず，任意の $\alpha, \beta \in \Sigma^*$ に対して，$\alpha \equiv_A \beta$ iff $Occ(\alpha) = Occ(\beta)$ と定義する．さらに，これをつぎのように二語相関パタン間の関係に拡張する：$(\alpha_1, k, \beta_1) \equiv_A (\alpha_2, k, \beta_2)$ iff $\alpha_1 \equiv_A \alpha_2$ and $\beta_1 \equiv_A \beta_2$. 明らかに，次が成り立つ．

補題 2 二語相関パタンを P と Q とする．このとき，$P \equiv_A Q$ ならば $supp(P) = supp(Q)$ かつ $conf(P) = conf(Q)$ である．

次に，A の接尾語木 $Tree_A$ を考える．A の部分語 α の**実節点** (locus) とは，$Tree_A$ の節点 $Locus(\alpha)$ で，$W(Locus(\alpha))$ が α を接頭語として含み，α が $W(Parent(Locus(\alpha)))$ を真の接頭語として含むものである．α に対して，その実節点が表す語を $Rep(\alpha) = W(Locus(\alpha))$ と定義する．ここに，$Parent(v)$ は節点 v の親を表す．実節点は必ず存在し，一意に決まる．

補題 3 (Amir et al.[2], McCreight[8]) 接尾語木 $Tree_A$ の節点を u とする．このとき，$W(u)$ がテキスト S 中の位置 p に出現することと，接尾語 A_p を表す $Tree_A$ の葉が u の子孫となることは同値である．

補題 4 任意の部分語 α に対して，$Rep(\alpha) \equiv_A \alpha$ が成立する．

証明: 今，接尾語木 $Tree_A$ を（辺のラベルに文字でなく文字列を許した）トライ (trie) [2] とみなして，α を根からの経路のラベルでちょうど表すような仮想的な節点 u を，$Tree_A$ に一時的に挿入したと仮定する．α の実接点を v とすると，u から v への経路上には分岐がないので，u と v の子孫である葉の集合は一致する．このとき，u に対しても補題3は依然として成立する．したがって，$Occ(W(u)) = Occ(W(v))$ である．定義より，$\alpha = W(u)$ かつ $Rep(\alpha) = W(v)$ なので，題意が示された． ■

これより，次の補題が成立する．

補題 5 任意の二語相関パタン $\langle \alpha, k, \beta \rangle$ に対して，$\langle \alpha, k, \beta \rangle \equiv_A \langle Rep(\alpha), k, Rep(\beta) \rangle$.

補題 5 から最適パタンの候補として，$Tree_A$ の任意の節点 u, v に対して，$\langle W(u), k, W(v) \rangle$ の形をしたパタン (以後**正規形**と呼ぶ) だけを調べればいいことがわかる．

3.3 支持度と確信度の高速な計算

本節では，Manber, Baeza-Yates [9] の手法を用いて，支持度と確信度を高速に計算する方法を与える．まず，入力テキスト A のすべての接尾語を考え，これらを辞書式順序で小さい順に並べた列を $A_{p_1}, A_{p_2}, \ldots, A_{p_n}$ とする．ここに A_p は，位置 p から始まる A の接尾語である．

つぎに，これらの接尾語の添字 $p_1, p_2, \ldots, p_n$ をこの順序で長さ n の 1 次元配列配列 $suf : [1..n] \to [1..n]$ に格納し，同時に，配列 suf の逆関数として 1 次元配列 $pos : [1..n] \to [1..n]$ を定義する．定義より，$suf(i)$ は順位 i の接尾語の S 中の出現位置であり，$pos(p)$ は接尾語 A_p の順位となる．

このとき，A の任意の部分語 α に対して，α を接頭語として含むような A の接尾語の出現位置全体は，配列 suf の連続したある部分区間 $I(\alpha)$ を占めることがわかる．

この観察を，二語相関パタンの出現位置に拡張しよう．位置空間 $[1..n] \times [1..n]$ の幅 k の対角成分を $INTER_k$ とおく．すなわち，

$$INTER_k = \{ (p, q) \mid 0 \leq (q - p) \leq k,\ \chi(p) = \chi(q) \}$$

である．ここに，$1 \leq \chi(p) \leq m$ は位置 p を含む例の番号である．つぎに，$INTER_k$ に含まれる点を，配列 pos を用いて位置空間から順位空間に変換する：

$$RANK_k = \{ (pos(p), pos(q)) \mid (p, q) \in INTER_k \}.$$

ここで，二語相関パタン $\langle \alpha, k, \beta \rangle$ に順位空間の軸並行な長方形 $I(\alpha) \times I(\beta)$ を対応させる．

補題 6 任意の組 $(p, q) \in [1..n] \times [1..n]$ に対して，以下の 2 つは同値である：

- パタン $\langle \alpha, k, \beta \rangle$ が A 中の位置 (p, q) に出現する．
- $(pos(p), pos(q)) \in I(\alpha) \times I(\beta) \cap RANK_k$.

補題 1と補題6から，与えられた二語相関パタンが出現する例の番号の集合 X を，直交領域質問をもちいて効率よく計算可能である．ことなる例の総数を m としたとき，集合 X から $supp(P)$ と $conf(P)$ は $O(m)$ 時間で容易に計算できる．

3.4 計算量

以上の補題から，つぎの定理を示すことができる．

定理 7 正整数を $k \geq 0$ とし，最小確信度を $0 \leq \sigma \leq 1$，S を分類例集合とする．をもつ二語相関パタンを仮説空間としたとき，アルゴリズム $Find\_Optimized\_Patterns$ は，距離パラメータ k と最小確信度 σ のもとで，最適確信度パタンのすべての代表元を時間 $O(mn^2 \log^2 n + K)$ と領域 $O(kmn \log n)$ で計算する．ここに，m は例の数であり，$n = size(S)$, K は最適確信度パタンの異なる代表元の数である．さらに，$k \leq n$ とする．

Algorithm Find_Optimized_Patterns1

Input: 分類例集合 S および，最小支持度 $\sigma \geq 0$, 距離パラメータ $k \geq 0$.

Output: $\equiv_S$ に関するすべての最適確信度パタンの代表元.

- $A = t_1\$\cdots\$t_m\$$ の接尾語木 $Tree_A$ と配列 suf と配列 pos を計算する.
- $R = \{\, \langle\langle pos(p), pos(q)\rangle, \chi(p)\rangle \mid \langle p, q\rangle \in INTER_k \,\}$ を計算する.
- $Tree_A$ の $O(n)$ 個の節点を葉から根へとボトムアップに走査しながら，以下の計算を行う：
  - はじめに，$Tree_A$ の葉を左から右へ走査し，第 x 番目の葉 l_x $(1 \leq x \leq n)$ に対して，$B(l_x) := \{\, \langle y, d\rangle \mid \langle\langle x, y\rangle, d\rangle \in R \,\}$ とし，y に関してソートして重複を除く.
  - 次に，葉から根に向かって $Tree_A$ の内部節点 u を走査しながら，再帰的に $B(u)$ を計算する．節点 u において，そのすべての子供 $u_1, \ldots, u_h$ のリスト $B(u_1), \ldots, B(u_h)$ を x 座標 x について重複を除いてソートしながら併合し，リスト $B(u) = \cup_{1\leq i\leq h} B(u_i)$ を計算する.
  - 同時に，内部節点 u において，u とリスト $B(u)$ を引数として，以下の手続きを実行する．これは u を固定したまま，$Tree_A$ の $O(n)$ 個の節点を葉から根へと再びボトムアップに走査しながら，以下の計算を行う：：
    - はじめに，$Tree_A$ の葉を左から右へ走査し，第 y 番目の葉 l_y $(1 \leq y \leq n)$ に対して，$C(l_y) := \{\, d \mid \langle y, d\rangle \in B(u) \,\}$ とし，d に関してソートして重複を除く.
    - 次に，葉から根に向かって $Tree_A$ の内部節点 v を走査しながら，再帰的に $C(v)$ を計算する．節点 v において，そのすべての子供 $v_1, \ldots, v_h$ のリスト $C(v_1), \ldots, C(v_h)$ を d に関して重複を除いてソートしながら併合し，リスト $C(v) = \cup_{1\leq i\leq h} C(v_i)$ を計算する.
    - 内部節点 v において，対応する二語相関パタン $P = \langle W(u), k, W(v)\rangle$ の支持度と確信度を評価する．リスト $C(v)$ は，P が出現する例番号のソート済みリストになっている．そこで，$C(v)$ から $supp(P)$ と $conf(P)$ を $O(m)$ 時間で計算する．もし $supp(P) \geq \sigma$ ならば，P を $conf(P)$ の値とともに優先度付待ち行列 Q に挿入する.
- Q 中のパタン P のうち，$conf(P)$ の値を最大化するものを解として出力する.

図 1: 改良版最適確信度パタン発見プログラム

証明: 入力テキストが与えられると，アルゴリズムは $O(n)$ 時間と $O(n)$ 領域で接尾語木 $Tree_A$ を計算する．次に，接尾語木の節点 v を葉から根へとボトムアップに走査しながら，区間 $I(v)$ を $O(n)$ 時間で順に計算する．補題 5から，$Tree_A$ の節点の組 (p, q) を枚挙することで，高々 $O(n^2)$ 個の $P = (Word(u), k, Word(v))$ の形の代表元パタンをすべて生成することができる．ここで，補題 1と補題 6から，$supp(P)$ と $conf(P)$ を $O(kmn \log n)$ 時間の前処理を用いて，$O(mn^2 \log^2 n)$ 時間と $O(kmn \log n)$ 領域で計算可能である．よって，定理が示された. ■

この結果は，計算時間 $O(n^5)$ の自明なアルゴリズムに比べると，より現実的な計算時間を与えている.

4 より高速なアルゴリズム

副手続きとして毎回新たに直交質問をするかわりに，接尾語木自体を区間木として使い，直交領域木を実現することができる．一般の直交領域木の場合と異なり，この場合，接尾語木は一般に平衡木ではないが，探索対象となる区間の数が線形個しかないので，Maass [4] の手法を利用して，効率よいアルゴリズムを実現できる．図 1に，このアイディアを用いた改良版アルゴリズムを示す.

接尾語木 $Tree_A$ の各節点 v には，ラベルとして，順位空間の y 座標と例番号の組 $\langle y, d\rangle$ を保持するリスト $B(v)$ と，例番号 d のリストを保持するリスト $C(v)$ が関連づけられているとする．アルゴリズムは，葉から根の方向へ接尾語木を走査しながら，節点のラベルを動的計画法を用いて計算する.

表 1: 最小支持度 σ に関する枝刈りで残ったパタン数の比較.

σ %	データサイズ (bytes)	初期仮説数	局所支持仮説数	大域支持仮説数
90%	2,701	7,295,401	121	40
50%	2,754	7,584,516	1,444	64
30%	2,764	7,639,696	6,241	88

補題 8 二語相関パタンに対する最大確信度パタン問題は, $O(mn^2)$ 時間と $O(\max\{k,m\}n)$ 領域で計算可能.

証明: アルゴリズムの構成からほぼ明らか. ただし, 領域量を $O(\max\{k,m\}n)$ におさえるために, アルゴリズムの上から5行目で, 要素数 kn の集合 $INTER_k$ から集合 R を一度に計算する代わりに, はじめは x 座標と例番号だけの組 $\langle p,d\rangle$ を用いて計算を行い, y 座標が要求される毎に x 座標を共有する組 $\{\,\langle\langle p,q\rangle,d\rangle \mid p \le q \le x+k\,\}$ を動的に生成して計算を継続する. また, 節点 u (v) における計算が終わり, その親に戻るときに, リスト $B(u)$ (C(v)) を破棄するよう領域を管理する. 以上と定理7を合わせて, 結果が示される. ■

5 支持度を用いた探索の枝刈り

3節の基本アルゴリズムで最適パタンの計算をする場合, 与えられた最小支持度 σ を用いて, 探索の枝刈りが可能である. これは, 実際的な効率化に使える.

まず, アルゴリズムは入力テキストを読み込みと, その接尾語木 $Tree_A$ を作成する. 次に, アルゴリズムは枝刈りの前処理として, 木全体を葉から根へとボトムアップに走査して, $O(mn)$ 時間ですべての節点の支持度を計算する. ここで, u の支持度 $supp(u)$ とは, 正例全体に対する $W(u)$ を含む正例の割合と定義する.

つぎに, アルゴリズムは, $Tree_A$ を根から葉の方向に深さ優先探索して, 前置順序 (preorder) で木を巡回し, 2つの節点の組を枚挙するものとする. 支持度は例数の増加に対して単調に増加するので, 次の補題が成り立つ.

補題 9 任意の節点 u,v において, $supp(\langle W(u),k,W(v)\rangle) \le \min(supp(u),supp(v))$.

補題 10 節点 u の任意の子孫 w に対して, $supp(w) \le supp(u)$.

これらの補題から, ある節点 u で $supp(u) \le \sigma$ ならば, $Tree_A$ の探索において u の子孫をそれ以上探索する必要がないことがわかる. これは, v についても同様である (局所枝刈りと呼ぶ). さらに, u を固定して v を枚挙している場合に, $supp(\langle W(u),k,W(v)\rangle) \le \sigma$ ならば, v の子孫をそれ以上探索する必要がないことがわかる (大域枝刈りと呼ぶ).

表 1に, 実際のテキストデータにおける枝刈りの効果を示す. データは, GenBank データベースのシグナル領域アミノ酸配列データを用いた. 495 個の正例と 7330 個の負例から, サンプリングによって正負それぞれ 50 個ずつ, 約 2.7KB 程度の例を抜き出して分類例とした. 第1番目と2番目の欄は, 最小支持度とデータサイズを表し, 3番目, 4番目, 5番目の欄は, それぞれ枝刈りを行わないときの候補パタン数, 局所枝刈り後の候補パタン数, 大域枝刈り後の候補パタン数を示している. この表から, 単純な局所枝刈りであっても, 候補パタン数を大きく減少させる効果があることがわかる.

つぎに, 大域枝刈りで残るパタン数の上限を見積もる. 分類例集合から作られる接尾語木の深さを $d \ge 0$ とすると, 次の補題が得られる.

補題 11 距離パラメータを $k \ge 0$ とし, 入力テキスト A の長さを n, A の接尾語木 $Tree_A$ の深さを d とする. このとき, 正の支持度をもつ正規形の二語相関パタンの総数は高々 $O(kdn)$ 個である.

証明: 入力テキスト A に対して，次の式をみたす可能な部分語の組 $\langle \alpha, \beta \rangle$ の総数を見積もる：$\exists p \in Occ(\alpha), \exists q \in Occ(\beta)[0 \leq q - p \leq k]$. このために，$Tree_A$ 中のすべての部分語 α について，その出現位置の総和 $F = \sum_{\alpha \in Tree_A} |Occ(\alpha)|$ を求める．まず，$Tree_A$ の全節点を葉からの高さで高々 d 個の層に分割する．補題 3から，α が i 番目の層に含まれる節点 $Locus(\alpha)$ で表されるとき，$Occ(\alpha)$ は $Locus(\alpha)$ の子孫となる葉の数に等しい．また，同じ層に含まれる節点同士は子孫を共有しない．したがって，各層の α について，$|Occ(\alpha)|$ の和は葉数 n でおさえられる．仮定より $Tree_A$ の深さは d なので，実節点で表現されるすべての部分語についてその異なる出現数の総和は，$O(dn)$ となる．二語相関パタンの定義より，支持率がゼロでないならば，先行する α の出現位置から k 文字以内に β が出現する．したがって，可能な組 $\langle \alpha, \beta \rangle$ の総数は高々 $O(kdn)$ でおさえられる． ■

一様分布で生成されたランダムテキストでは，接尾語木の平均深さが $O(\log n)$ 程度でおさえられることが知られている [9]．アミノ酸配列データも良く似たふるまいを示す．このとき，上の補題より，解の数は $O(n^2)$ よりも小さくなり，平均的には $O(kn \log n)$ 程度になることがわかる．したがって，この種のテキストでは最小支持率による枝刈りの効果も大きいと期待される．

6 おわりに

本稿では，テキストデータからのパタン発見問題について考察した．とくに，二語相関パタンとよぶ単純なパタンを仮説としたとき，最大確信度パタン問題を高速に解くアルゴリズムを与えた．

本稿で導入した二語相関パタンは，たいへん制限された正規表現とみなせる．Baeza-Yates, Gonnet [3, 7] は，接尾語木を用いて制限された正規表現のパタン照合をきわめて高速におこなう方法を提案している．本稿のパタン発見手法を拡張して，正規表現の部分族を用いたデータマイニング手法を開発することも今後の課題の一つである．

参考文献

[1] S. Abiteboul, Querying semi-structured data. Proc. ICDT'97 (1997).

[2] A. Amir, M. Farach, Z. Galil, R. Giancarlo, K. Park, Dynamic Dictionary Matching. *JCSS*, 49 (1994), 208–222.

[3] R. Baeza-Yates, and G. H. Gonnet, Fast text searching for regular expressions or automatons searcing on tries. J. ACM, **43**(6), 915–936, 1996.

[4] W. Maass, Efficient agnostic PAC-learning with simple hypothesis, In Proc. COLT94 (1994), 67–75.

[5] R. Feldman and I. Dagan, Knowledge Discovery in Textual Databases (KDT). In Proc. KDD-95 (1995).

[6] T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama, Data mining using two-dimensional optimized association rules, In Proc. the ACM SIGMOD Conference on Management of Data, 13–23, 1996.

[7] G. Gonnet, PAT 3.1: An efficient text searching system. User's manual. UW Center for the New OED, University of Waterloo (1987).

[8] E. M. McCreight, A space-echonomical suffix tree constructiooon algorithm. JACM 23 (1976), 262–272.

[9] U. Manber and E. W. Myers, Suffix arrays: A new method for on-line string searches, In *Proc. 1st Annual ACM-SIAM Symp. Discrete Algorithms*, 319–327, 1990.

[10] OPENTEXT Index, http://index.opentext.net (1997).

[11] F. P. Preparata, M. I. Shamos, Computational Geometry, Springer-Verlag (1985).