数理モデル化と問題解決 26－11
（1999．9．22）

概念階層を持つパターン言語の帰納学習による遺伝子クラスタ解析への応用

遠里 由佳子† 松田 秀雄† 橋本 昭洋†

正例として与えられた文字列集合からそれらの特徴を表す極小でかつ既約なパターンの組を求める問題をMINL問題と呼び，パターンの組の数を予め制限することにより，この問題を解く多項式時間アルゴリズムが知られている．同じ機能を持つ遺伝子を同じ文字で表すと，このアルゴリズムは遺伝子クラスタの機能分類に応用できるが，遺伝子の機能は階層的に分類されているため，どの階層の機能分類を使うか指定しないとこのアルゴリズムを適用することができない．そこで，本研究では，パターン間に概念階層を導入し，情報量というパターンの評価基準を使ってMINL問題を近似的に解く多項式時間アルゴリズムを提案し，実際に遺伝子クラスタの機能分類に適用した結果をもとに性能評価を行う．

Application to gene cluster analysis of inductive inference of languages over patterns with conceptual hierarchy

YUKAKO TOSATO,† HIDEO MATSUDA† and AKIHIRO HASHIMOTO†

The MINL problem is a problem that finds a minimum and reduced set of patterns explaining a given set of positive example strings. By restricting the number of patterns to be a fixed constant in advance, a polynomial time algorithm that solves this problem is known. This algorithm is applicable to determining gene clusters based on functional classification if genes having the same function are expressed with the same character. However, since gene function is typically classified hierarchically, the above algorithm can only be applied on a single level of the classification hierarchy. In this paper, we extend the MINL problem to cover hierarchical classifications, and propose a novel polynomial time algorithm utilizing entropy to solve the extended problem. In an experiment, we applied our method to gene cluster analysis of actual gene data.

1. はじめに

正例として与えられた単語列集合からそれらの単語列の特徴を表現する極小でかつ既約な規則の集合を求める問題をMINL問題と呼ぶ．ここでいう規則とはAngluin[1]が提案したパターンと呼ぶ記号で定式化されたものであり，パターンの組の数を予め決められた個数に制限することにより多項式時間でこの問題を解くアルゴリズムが存在する．[2]

本研究では，パターンの組の数を制限する代わりに，パターンの集合の評価基準に情報量を使うことによって，MINL問題を近似的に解くアルゴリズムを提案する．また，単語列集合の特徴をより明らかとするために，単語間に概念階層と呼ぶ概念の上位・下位関係を表す順序を導入することを考えた．従来のアルゴリズムではそのパターンの定義から単語が階層的に分類されているなら，どの階層を使ってパターンを生成するかを指定しないと適用できない．そこで，概念階層を単語の半順序集合としてとらえ，概念が上位の単語は下位の単語を包含できるようにパターンの定義を拡張した．ここで紹介するアルゴリズムは，その拡張されたパターンを用いることによる操作を含めても，従来のアルゴリズムより少ない計算量となっている．

このアルゴリズムを遺伝子データの解析に適用する．現在，各生物のゲノム上での遺伝子の位置を比較することにより，複数のゲノムで機能的に関連した遺伝子が同じかまたは部分的な入れ替えだけの類似した順番で並んでいる箇所があることが知られている．[3] この領域は遺伝子クラスタと呼ばれ，これを見つけることが分子生物学上で重要な課題の一つとなっている．[4][5]

遺伝子は，A, T, C, Gという4種類のアルファベットの要素からなる文字列で表現できる．したがって，1つの遺伝子は単語と言いかえることができる．遺伝子

† 大阪大学 大学院基礎工学研究科 情報数理系専攻
Department of Informatics and Mathematical Science, Graduate School of Engineering Science, Osaka University

間の概念階層として, 本研究では酵素をコードしている遺伝子のみに対象を限定し, 酵素の機能分類に基づいて階層を定義する. 具体的には酵素には EC(Enzyme Commission) 番号と呼ばれる 4 段階の階層を表す番号が付けられており, 本研究ではこの機能分類階層を利用する. すると, すでに分かっている酵素遺伝子からなる遺伝子クラスタのデータを正例として本アルゴリズムに与えることで, その違いや共通の規則を説明するパターンの組を取り出すことができる. 実際に, 概念階層を使った本アルゴリズムの有効性を確かめるため, 遺伝子クラスタの機能分類に適用し, 取り出されたパターンの組の精度を確かめる実験を行った.

2. 概念階層を持つパターン言語と情報量

本節では, 概念階層を定式化し, 本研究における一般化の問題を扱う上での最も重要な要素である概念階層を持つパターン言語のクラスを定義する. そして, パターン集合の評価基準として用いる情報量を導入する.

2.1 諸 定 義

概念階層を記述するため定数記号間の二項関係を Angluin[1)] の定式化したパターンに導入することにより, 概念階層を持つ正則パターン言語を定義する.

定数記号の有限集合を Σ で表す. 定数記号間には概念階層における上位, 下位を表す半順序関係 $\succeq_\Sigma$ があるものとする. $a \succeq_\Sigma b$ のとき, a は b より**一般的**, または, b は a より**具体的**であるという. 定数記号を**単語**と呼ぶこともある.

集合 Σ に対して $\#\Sigma$ は Σ における要素の数を表す. Σ における極小元の集合は, $b \succ_\Sigma a$ であるような $a \in \Sigma$ が存在しない定数記号 b の集合をさし, Σ_{min} で表す.

$X = \{x_1, x_2, x_3, \cdots\}$ を Σ と交わらない**変数**の可算無限集合とする. このとき, **正則パターン** (regular pattern) とは, 定数記号と変数記号からなる文字列のことで, 各変数は 1 回しか文字列中に出現しないものである. $\Sigma \cup X$ 上の正則パターン全体の集合を RP_Σ と書く. 以下では, 正則パターンのみ扱うので, 正則パターンのことを単にパターンと呼ぶことがある. 任意の正則パターンは正則パターン中の変数が連続した出現を持たない形に定義できる.[6)] したがって, すべての変数の一回以上の繰り返しを "*" で統一し表す.

定数記号間の階層関係 $\succ_\Sigma$ をパターン間の二項関係に拡張するために代入を定義しておく. 代入とはパターン中の変数を別のパターンか空文字で置き換えたり, 定数記号をそれより具体的な定数記号で置き換えることをいう. 以下では, 代入 θ の列 $\theta_1, \cdots, \theta_n$ を置換の集合として表すことがある. そのとき, 定数記号または変数からそれ自身への置換を省略しておく.

例 2.1 $\Sigma = \{a, b, c\}$ $(a \succ_\Sigma b \succ_\Sigma c)$, $\theta = \{* := a,\ b := c\}$ とする. このとき, $p = *b$ はパターンであり, $\theta(p) = ac$ となる.

パターン p, q に対して, $\theta(p) = q$ が成り立つ代入 θ が存在するとき $p \succeq_{RP} q$ と書く. この関係 $\succeq_{RP}$ によって概念階層を持つパターン集合 $(RP_\Sigma, \succeq_{RP})$ が定義される. 定数記号の場合と同様に, $p \succeq_{RP} q$ のとき, p は q よりも**一般的**, q は p より**具体的**であるという. $p \succeq_{RP} q$ かつ $q \succeq_{RP} p$ が成立する同値なパターンを同一視するとき, パターン集合は半順序集合とみなすことができる. パターン p の言語 $L(p)$ を $L(p) = \{w \in \Sigma^+ \mid p \succeq_{RP} w\}$ と定義する.

例 2.2 $\Sigma = \{a, b, c\}$ $(a \succ_\Sigma b, a \succ_\Sigma c)$ とする. このとき, パターン $p = *a$ の言語 $L(p)$ とは $L(p) = \{a, b, c, aa, ab, ac, ba, \ldots\}$ である.

2 つのパターン集合 P と Q の間に $(\forall p \in P, \exists q \in Q) p \succeq_{RP} q$ という関係が成り立つとき, これを $P \sqsupseteq_{RP} Q$ と表す. $P \sqsupseteq_{RP} Q$ が成り立つとき, P は Q より**一般的**, Q は P より**具体的**であるという. パターン集合 P に対して**和言語** $L(P) = \cup_{p \in P} L(p)$ を定義する. 和言語 L_i が正例 S に対して**極小**であるとは, $S \subseteq L_i$ かつ任意の j に対して $S \subseteq L_j$ ならば $L_j \not\subset L_i$ が成り立つことをいう. パターン集合 P が正例 S に対して**既約**であるとは, $S \subseteq L(P)$ かつ P の任意の真部分集合 P' に対して, $S \not\subseteq L(P')$ であることをいう.

扱う対象がパターンもしくはパターン集合であることが明らかな場合に, X が Y よりも具体的ならば Y は X を**説明する**という. また, **汎化**とは 2 つのパターンに対してそれを説明する最も具体的なパターンを求めることをいう.

2.2 パターン集合の評価基準

単語 $w \in \Sigma_{min}$ の情報量 $I(w)$ とは, 正例 S における w の出現頻度が p_w のとき, $-\log_2 p_w$ であるとする. 極小元でない単語 w については, w より具体的なすべての極小元が $w_1, w_2, \ldots, w_n$ ならば, 情報量 $I(w)$ は, 各単語の正例における出現頻度を $p_1, p_2, \ldots, p_n$ としたとき, $-\log_2(p_1 + p_2 + \ldots + p_n)$ となる. 変数 "*" の情報量は 0 とする. このとき, パターン $p = w_1 w_2 \ldots w_n (w_i \in \Sigma \cup X, 1 \leq i \leq n)$ の情報量 $I(p) = I(w_1) + I(w_2) + \ldots + I(w_n)$ とする.

パターン集合 P の情報量 $I(P)$ は, P 中の各パターン p が正例集合中の p_s 個の正例を説明するとき, $k = \#P, n = \#S$ とするならば, 以下のように計算する.

$$I(P) = -\log_2 \frac{k}{n} \sum_{p \in P} \frac{p_s}{n} I(p) \qquad (1)$$

例 2.3 $\Sigma = \{a, b, b_1, b_2, c, c_1, c_2\}$ $(b \succ_\Sigma b_1, b \succ_\Sigma b_2, c \succ_\Sigma c_1, c \succ_\Sigma c_2)$, 正例 $S1 = \{ab_1c_1, b_2c_2, ab_2c_2\}$ とする. このとき, $S1$ を説明するパターン集合 $P0 = S1$, $P1 = \{abc, b_2c_2\}$, $P2 = \{*b_2c_2, ab_1c_1\}$, $P3 = \{*bc, ab_2c_2\}$, $P4 = \{*bc\}$ において, 既約ではない $P3$ 以外の各情報量は $I(P1) = 3.60$, $I(P2) = 3.11$, $I(P4) = 4.48$ となる. また, 正例

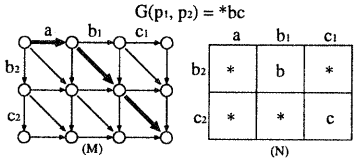

図1 動的プログラミング法による汎化
Fig. 1 Generalization of patterns by dynamic programming

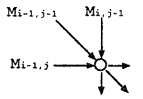

図2 評価関数の計算法
Fig. 2 Calculation method of an evaluation function

$S2 = \{ab_1c_1b_1b_2b_1, b_2c_2, ab_2c_2b_1b_2b_2\}$ ならば, $S2$ を説明するパターン集合 $P1 = \{*bc*\}$ と $P2 = \{abcb_1b_2b, b_2c_2\}$ において, 各情報量は $I(P1) = 4.53$, $I(P2) = 4.57$ となる. (小数点以下3桁で切捨て)

3. 情報量最大アルゴリズム

3.1 汎　　化

正例集合からそれを説明するパターン集合を見つけるための基本は, まず2つの正例を並べて比較してみることである. この並べる操作をアライメント (alignment) という. 汎化の出発点は, 動的プログラミング法による2つの正例のアライメントとなる. $\Sigma = \{a, b, b_1, b_2, c, c_1, c_2\}$ ($b \succ_\Sigma b_1, b \succ_\Sigma b_2, c \succ_\Sigma c_1, c \succ_\Sigma c_2$) のとき, 正例 ab_1c_2 と b_2c_2 が与えられた場合を例にして, 動的プログラミング法の原理を**図1**に示す. 図1の右は2つのパターンを説明する最も具体的なパターンを求めるための経路を表す行列 M の状態, 左は各単語の上位概念を記録するための行列 N の状態を表す. 上の $G(p_1, p_2) = *bc$ は p_1 と p_2 の汎化でパターン $*bc$ が得られることを表している.

動的プログラミング法による解法は, 比較する正例を縦方向と横方向に並べた行列の形で説明される. そして行列 M での各段階のスコア $M_{i,j}$ は, 斜め方向, 縦方向, 横方向の3つの経路のなかから最適なものを選ぶことにより決定される (**図2**参照). これは式で書くと次のようになる.

$$M_{i,j} = max \begin{cases} M_{i-1,j-1} + I(w_{1i}, w_{2j}), \\ M_{i-1,j}, \\ M_{i,j-1} \end{cases} \quad (2)$$

$I(w_{1i}, w_{2j})$ は, 単語 w_{1i} と単語 w_{2j} の最も具体的な共通の共通の上位概念となる単語を探し, 行列 N の成分 $N_{i,j}$ に記録した後, その情報量を計算することを表す. 行列 M は, 初期値を $M_{0,0} = M_{i,0} = M_{0,j} = 0$ とし, 行列の左上端から順番に $M_{i,j}$ を計算していき, 右下端に到達したときの値が最適な評価関数値である.

procedure $MG(S, \Sigma)$
　$n := \sharp S$; $P := S$; $P_k := S$;
　$max := 0$;
　for $k := n-1$ **downto** 1 **do**
　　foreach $p_i \in P$ **do**
　　　foreach $p_j \in P$ $(p_i \neq p_j)$ **do**
　　　　$p := G(p_i, p_j)$;
　　　　$P' := P - \{p_i, p_j\} \cup \{p\}$;
　　　　if P'が既約かつ $max < I(P')$ **then**
　　　　　$P_{n-k} := P'$;
　　　　　$max := I(P')$;
　　　endforeach
　　endforeach
　　if $I(P_{n-k}) < I(P_{n-k+1})$ **then**
　　　return P_{n-k+1};
　　$P := P_{n-k}$;
　endfor
　return P_1;

図3 多重汎化アルゴリズム
Fig. 3 Multiple generalization algorithm

る. この計算の後, 最適な評価関数値からそれが求められた最適経路を逆向きにたどり, 縦か横を通るならギャップ "-", 斜めを通るなら N の対応する成分の単語を当てはめるだけで s_1 と s_2 のアライメントを求めることができる. 求められるアライメントを, パターンと対応させるには, パターン中にギャップ "-" が連続する場合に, 一つの変数 "*" に置き換えるだけでよい. そして, 2つの正例を被覆する最も具体的なパターン $*bc$ を求めることができる. この手続きにかかる計算量は, 正例の最大長を l としたとき, $O(l^2)$ となる.

3.2 多重汎化への拡張

前述の2個の正例で定義された汎化を n 個の正例での汎化に拡張することを考える. このとき得られるのは k 個のパターン $(1 \leq k \leq n-1)$ となる. これを厳密に求めると多次元のアライメントが必要になり計算量が $O(\sum_{k=1}^{n-1} \binom{n}{n-k+1} l^{n-k+1})$ となるため, 大きな n の値に対しては現実的ではない. そこで, 以下で述べるように近似的に多重汎化を行う多項式時間アルゴリズムを開発した. (**図3**参照)

$k = n-1$ 個のパターンを得る多重汎化を n 個の正例から任意の2個を選んで汎化を行ったときの最も情報量の大きいパターンを選ぶ操作とし, 次の $k = n-2$ 個のパターンを得る多重汎化では, 前の操作で得られた結果 (n 個の中から2個を除いて1個のパターンを加えたもの) を新たな正例として同じ操作を行う. 以下これを $k = 1$ となるまで繰り返し, その過程で得られた最も情報量の大きいパターンを解とする.

この手続きの計算量は, 正例の最大長を l とするとき $O(n^3 l^2)$ となる. また, 極小性と情報量最大とは関連する. そして, 情報量を使う方が汎化の対象となる

Archaeoglobus fulgidus	4.2.1.20	4.2.1.20	2.4.2.18	5.3.1.24	4.1.1.48	4.1.3.27	2.4.2.18	5.3.1.24	4.1.3.27
Methanobacterium thermoautotrophicum	4.2.1.20	4.2.1.20	4.1.1.48		5.3.1.24	4.1.3.27	2.4.2.18		4.1.3.27
Methanococcus jannaschii	4.2.1.20	4.2.1.20							
Mycobacterium tuberculosis	4.2.1.20	4.2.1.20	4.1.1.48						4.1.3.27
Bacillus subtilis	4.2.1.20	4.2.1.20	4.1.1.48		5.3.1.24		2.4.2.18		4.1.3.27
Chlamydia trachomatis	4.2.1.20	4.2.1.20							
Helicobacter pylori J99	4.2.1.20	4.2.1.20		5.3.1.24		2.4.2.18			4.1.3.27
Haemophilus influenzae		4.1.3.27	4.1.1.48	5.3.1.24		4.1.3.27	2.4.2.18		4.1.3.27
	4.2.1.20	4.2.1.20							
Escherichia coli	4.2.1.20	4.2.1.20	4.1.1.48	5.3.1.24		2.4.2.18	4.1.3.27		4.1.3.27

図4　トリプトファン合成系の遺伝子クラスタ
Fig. 4　Structure of gene cluster of tryptophan in different organisms

```
H1 = { [4.2.1.20] [4.2.1.20] * ,
       * [4] * [4.1.1.48] * [4.1.3.27] [2.4.2.18] * [4.1.3.27] }
H2 = { [4.2.1.20] [4.2.1.20],
       * [4.1.1.48] [5.3.1.24] [4.1.3.27] [2.4.2.18] [4.1.3.27],
       [4.2.1.20] [4.1.1.20] * [4.1.1.48] * [2.4.2.18] [4.1.3.27] }
```

図5　実行結果
Fig. 5　Execution result

パターンを選択するときにより多くの情報をもとに判定を行い，分割数を指定せずにより少ない計算量で正例を説明するパターンの集合を求めるという理由で望ましい．

4. 実　　験

集合 Pos を正例の集合とする．このとき，Pos から真部分集合を無作為に選択し，この部分集合に本アルゴリズムを適用し，仮説として概念階層を持つパターンの集合 H を作り出す．仮説の Pos に対する精度は，Pos に含まれる例のうち正しく正例であると認識された割合とする．実験では，正例として図4のようなトリプトファン合成系の遺伝子クラスタのデータ[3)5)]を与えた．表の各列は各生物においてトリプトファン合成を担う遺伝子の列のデータを表し，生物種名以外の枠で囲まれた部分は1つの遺伝子を，その中に書かれた番号はその遺伝子が産生する酵素のEC番号を表す．また，EC番号として，4.1.1.20が与えられると概念階層としての要素に { 4, 4.1, 4.1.1, 4.1.1.20 } を追加し，$4 \succeq_\Sigma 4.1 \succeq_\Sigma 4.1.1 \succeq_\Sigma 4.1.1.20$ とした．概念階層を用いた場合と用いない場合の結果を比較する．

仮説の数を9としたとき，概念階層を用いた場合に図5のようなパターン集合 H_1 が求められ，精度100%となった．同様に，概念階層を用いない場合ではパターン集合 H_2 が求められ，精度が80%に留まった．図5中の "[" と "]" で囲まれた部分は定数記号を表している．H_1 の情報量は12.47，H_2 の情報量は8.29である．

5. おわりに

MINL問題を情報量というパターンの評価基準を使って近似的に解く多項式時間アルゴリズムを提案した．本手続きは，従来の方法より少ない計算量で，より多くの情報を基にパターン集合の良し悪しを評価する．また，背景知識として概念階層を使うことにより，その精度が向上することを確かめた．今後の課題として，極小性との関係を確かめることや，遺伝子の連接や，繰り返しといった現象を踏まえて，パターンの定義を拡張することが上げられる．

参考文献

1) Angluin, D.: Finding patterns common to a set of strings, *J. Comput. System Sci.*, Vol. 21, pp. 46–62 (1980).

2) Arimura, H., Shinohara, T. and Otsuki, S.: Finding minimal generalizations for unions of pattern languages and its application to inductive inference from positive data, In *Proc. the 11th STACS,* LNCS 775, Springer-Verlang, pp. 649–660 (1994).

3) Dandekar, T., Snel, B., Huynen, M. and Bork, P.: Conservation of gene order: a fingerprint of proteins that physically interact, *Trends in Biochemical Sciences*, Vol. 23, No. 9, pp. 324–328 (1998).

4) Bono, H., Goto, S., Fujibuchi, W., Ogata, H. and Kanehisa, M.: Systematic Prediction of Orthologous Units of Genes in the Complete Genome, *Genome Informatics 1998*, pp. 32–40 (1998).

5) Fujibuchi, W., Ogata, H., Matsuda, H. and Kanehisa, M.: Automatic Detection of Gene Clusters by P-Quasi Complete Linkage Grouping, *Genome Informatics 1998*, pp. 300–301 (1998).

6) Shinohara, T.: Polynomial time inference of pattern languages and its applications, In *Proc. the 7th IBM symposium on Mathematical Foundations of Computer Science*, pp. 191–209 (1982).